MICROMECHANICAL FATIGUE TESTING

by

JOHN A. CONNALLY

B.S. Mechanical Engineering
Tulane University
(1981)

M.S. Mechanical Engineering
Massachusetts Institute of Technology
(1987)

SUBMITTED TO THE DEPARTMENT OF
MECHANICAL ENGINEERING
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1992

Signature of Author_______________________________________
Department of Mechanical Engineering

Approved by____
Technical Supervisor, Draper Laboratory

Certified by____
Dr. Stuart B. Brown
Thesis Supervisor

Accepted by____
Dr. Ain A. Sonin
Chairman, Mechanical Engineering Department

ACKNOWLEDGEMENTS
(May 1, 1992)

This thesis is dedicated to Paul Greiff, Burton Boxenhorn, and Professor Stuart Brown: without their assistance, encouragement, and advice, this work would not have been possible. For my parents and sister, I thank them for their unwavering support and confidence through these years. Special thanks to Matt Ferber, Oak Ridge National Laboratory, for the use of the nanoindenter; and Steve Findberg, for his assistance with the design of the interface electronics.

This thesis was prepared at the Charles Stark Draper Laboratory, Inc., under Contract number 18668.

Publication of this thesis does not constitute approval by Draper or the sponsoring agency of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

I hereby assign my copyright of this thesis to The Charles Stark Draper Laboratory, Inc., Cambridge, Massachusetts.

(authors signature)

Permission is hereby granted by The Charles Stark Draper Laboratory, Inc., to the Massachusetts Institute of Technology to reproduce any or all of this thesis.

Contents

List of Tables

List of Figures

Abstract

This thesis documents an experimental and theoretical investigation of nano scale fracture and fatigue of a resonant single crystal silicon micro electromechanical device. Nano scale time dependent crack growth has been observed: crack extensions of 10 to 300 nm have been measured with an accuracy of approximately 2.5 nm; crack velocities on the order of 2.0×10^{-14} m/s have also been accurately measured. It is postulated the 10 to 40 $\mathring{A}$ silica layer that forms on the surface of freshly cleaved silicon is continuously attacked by water vapor in the environment. As the crack propagates through this layer, fresh silicon is continuously exposed, oxidized, and statically fatigued. The fatigue mechanism is rate limiting. Starter pre-cracks do not exponentially grow with stress intensity, as observed with macro scale static fatigue specimens, ultimately causing failure. The measurements were made by detecting the shift in natural frequency of a specially designed micromechanical fatigue test structure constructed from boron doped single crystal silicon. The structure is a cantilever beam with an end plate and gold inertial mass. Bridge electrodes extend over the plate, and with interface electronics and a phase sensitive control system, drive the structure continuously at resonance and sense its motion. Fatigue crack propagation is measured by detecting shifts in the natural frequency caused by the extension of a pre-crack introduced near the base of the cantilever. The pre-crack is formed by a series of nanoindentations that traverse the full width of the unetched beam.

Chapter 1

Introduction and Literature Review

1.1 Introduction

Micromechanical fabrication techniques have revolutionized the ability of designers to miniaturize electro-mechanical systems for use in compact low cost sensors, actuators, and transducers. Micromechanical devices have found many diverse applications in fields such as inertial guidance and control, fluid sensing, acoustics, robotics, and biomechanics to name a few. More exotic and novel components such as springs, gears, motors, and linkages are being fabricated and tested[13][42]. As the field emerges, other applications will be discovered opening new vistas of research and development.

A high level of engineering sophistication in several disciplines is required to successfully design micromechanical devices, and integrate associated electronic systems, subsystems, and hardware. Consequently, much effort is focused on fabrication processes and systems engineering. However, little attention is given to investigating failure processes and structural reliability. The commercial use of micromechanical devices will lead to a greater emphasis on structural reliability as questions arise dealing with the ability of a particular device to meet performance specifications under various operating conditions.

Tackling such an issue in a unified way that would reveal the nature and interaction of different failure mechanisms presents a variety of problems.

Micromechanical devices vary greatly in size, shape, and function. The materials and processes used to fabricate the devices vary as well. All these factors will in some way affect the relative susceptibility of the device to differing modes of failure, and will dictate to an extent the type of testing methodology that is appropriate. For example, testing a micromechanical gas sensor for environmental stress corrosion cracking may be more critical than testing for shock loading capability, while the reverse would be true of a micromechanical accelerometer[19].

It is necessary to establish fatigue and fracture testing techniques that have broad appeal and application so that failure analyses of micromechanical devices can evolve to yield useful information for micromechanical designers and scientists. Information gained from such tests could be readily shared with others, thereby building a common body of knowledge. The aim of this research is to introduce and demonstrate a novel fatigue testing technique particularly suited for micromechanical testing, and to utilize this technique in an analytical and experimental investigation of the fatigue effects in a specially designed micromechanical test structure.

1.2 Literature Review

1.2.1 Current Research on Micromechanical Devices

The micromechanical test structure considered in this thesis is constructed of single crystal silicon through a step by step fabrication process that involves diffusing boron into an n-type silicon wafer in a pattern identical to the device. The heavy concentration of boron makes the diffused silicon resistant to chemical etching. Therefore, after all other processing steps are complete, the wafer chip can be immersed in an etch solution that dissolves the undiffused portion of the wafer leaving a "released" structure free to move and vibrate. The etch is anisotropic and proceeds along the $< 111 >$ planes of the undiffused silicon until the planes intersect at which time no further etching occurs. The boron diffusion occurs at high temperature, $1150°C$, with the diffusion time controlled by the desired thickness of the device: the longer the time, the thicker the device. A consequence of diffusing boron into silicon is a high state of residual tensile stress which is a result of the silicon atoms being substitutionally replaced by the smaller boron atoms causing the lattice to contract. A further consequence of this diffusion is a high dislocation density caused by the displaced silicon atoms. The concentration of the boron at the surface is approximately 1.0×10^{20} atoms/cm^{-3}. This fabrication method differs from polysilicon processing in that single crystal silicon is removed by selective anisotropic etching as opposed to chemically depositing polysilicon on a wafer substrate.

Currently, no effort has begun in analyzing fatigue in heavily boron-doped single crystal silicon micromechanical devices, nor has fatigue data been reported for polysilicon structures. Within the micromechanical field, polysilicon is the primary fabrication material, and as a result, polysilicon structures and their mechanical properties are a subject of intense research. However, efforts directed toward understanding the mechanical behavior of polysilicon on a fundamental level tend to be focused on characterizing the properties of the material, particularly Young's modulus. Such emphasis is a result of the large degree of variability in the growth mechanisms of polysilicon. Grain sizes and surface roughness vary considerably from one production run to another, and from one laboratory to the next. Small differences in temperature, humidity, oxygen content, and so on have profound effects on the makeup and quality of the polysilicon. Making quantifiable and generalized statements about fatigue with such variability would be difficult. However, extremely high quality single crystal silicon, grown almost to perfection, is universally available and often used in devices requiring a high degree of mechanical stability and repeatability in material behavior. For example, open loop resonant micromechanical structures used as inertial sensors must meet drift specifications on the order of a few ppm. Such properties remove many uncertainties, making single crystal silicon an ideal material for materials research on a fundamental level; including, fatigue testing on a micromechanical scale. It can be said with reasonable assurance that single crystal silicon devices constructed by the boron diffusion process will have virtually identical mechanical properties from device to device regardless of where the device is fabricated.

The work of Gabriel and Behi is a good example of the kind of empirical experimental research being conducted on polysilicon structures [14]. In their experiment, friction and wear are characterized using a polysilicon micromechanical turbine that rotates about a concentric shaft fixed to the silicon substrate. The turbine is spun using an air jet exiting from a micropipette, and a helium/neon laser with associated optics records the angular velocity of the turbine wheel. Measurements were made of the increase in diameter of the center hole of the wheel after rotating for a specified period of time, and of the deceleration profile of the turbine when the gas stream was stopped. Such techniques and measurements provide empirical data, but the next step is correlating such data to quantitative models that predict wear lifetimes and friction properties given appropriate attributes of the material such as surface roughness, grain size, and hardness.

Tai and Muller developed a technique for determining Young's modulus by measuring the deflection of a polysilicon microbridge using a force measuring stylus [41]. The accuracy of such a measurement, however, is dependent on how well one knows the geometry of the bridge structure and the residual stresses, if any, which could affect the stiffness.

1.2.2 Fracture of Silicon Micromechanical Devices

Fan and Howe developed a structure to measure the fracture toughness, K_{IC}, of polysilicon microbridges subjected to a high residual tensile stress [12]. At the center of the structure, the configuration of the bridge resembles a compact tension specimen with a deep notch. When the microbridge is etched the residual stress either fractures the bridge at the notch, or does not, depending on the specific geometry used. Fan and Howe claim that performing a number of tests with different geometries establishes an upper and lower bound for K_{IC}. In conclusion, Fan and Howe determine a "critical geometry" defined as the ratio of fracture toughness to an assumed residual stress. K_{IC} values or "critical geometry" factors obtained from such tests, however, must be scrutinized carefully for two reasons. First, the notch is a stress concentration, not a crack; therefore, when the device is etched, it may be that given a high prestress the state of stress at the root of the notch may exceed the cohesive strength of the polysilicon initiating a crack which in turn gives a K_I well in excess of the actual K_{IC}. Second, the residual stress can only be inferred, it is not measured directly, which introduces some error.

A potentially promising approach to fracture toughness testing of microdevices was conducted by Johansson and Schweitz [23]. In their experiment, single crystal silicon micro-cantilevers, electrochemically etched, were deflected by a force and displacement sensing diamond tipped stylus until fracture. The thickness, length, and width of the cantilevers varied from $8-16\mu m$, $75-500\mu m$, and $75-240\mu m$, respectively. The test was performed *in situ* in a scanning electron microscope giving immediate observations of the fracture surface without handling or exposure to the environment. The advantage of this approach lies in the visual resolution and depth of field possible with the SEM which allows for accurate positioning of the stylus over the cantilever. Clearly, without the depth resolution of a SEM, positioning the stylus over the cantilever while viewing it at an angle with an optical microscope would be very difficult. Otherwise, a method for placing the stylus over the cantilever must be devised for the stylus would hide the cantilever from view when the microscope objective is above the cantilever. Johansson and Schweitz, however, did not compute a K_{IC} based on a known crack initiation site and corresponding K_I factor. Rather, they computed the fracture stress from beam theory, and correlated the result to random surface flaws generated during fabrication. The fracture stresses averaged about 2% of Young's modulus which is very close to the 2.06% reported by Eisner [8]. Eisner fractured silicon whiskers in tension with diameters on the order of $1\mu m$. Tensile loading was accomplished using micromanipulators and a damped pendulum bob. Peterson and Read also fractured silicon whiskers in bending and reported fracture stress values as high as 2.6% of E [35].

Experiments similar to that of Johansson and Schweitz were conducted with a nanoindenter by Nix and Weihs [21][22]. The nanoindenter was used to measure the bending stiffness of gold and LPCVD (Low Pressure Chemical Vapor Deposition) SiN_X micro-cantilevers deflected at the tip. Young's modulus was then computed using beam theory equations. Although fracture experiments were not conducted, it is possible to use the nanoindenter to deflect micro-cantilevers until fracture with a known stress intensity factor associated with a pre-crack.

1.2.3 Fracture Toughness of Single Crystal Silicon

Fracture toughness experiments on silicon have been performed and documented by several researchers. Chen and Leipold determined the fracture toughness of single crystal silicon as a function of the orientation of the crystal axes. K_{IC} values were determined for the $< 111 >$, $< 110 >$, and $< 100 >$ planes to be 0.82 $MN/m^{3/2}$, 0.90 $MN/m^{3/2}$, 0.95 $MN/m^{3/2}$, respectively [5]. In their experiment, a Knoop diamond indentation initiated a pre-crack on the top surface of a beam which was then fractured in four point bending.

Later, Chen and Leipold determined the K_{IC} of single crystal silicon in the $< 111 >$ plane to be 1 $MN/m^{3/2}$ [4]. A double torsion test specimen was used where pre-cracking was accomplished by driving a wedge into a notch machined into the edge of the silicon specimen. Chen and Hsu reported that the lower K_{IC} value determined in [5] was due to residual compressive stress in the region of the indentation site [3].

1.2.4 Stress Corrosion Fatigue in Ceramics and Silicon

Subcritical crack growth in glass and ceramics is well documented with a large body of literature available for review. A.G. Evans and S.M. Wiederhorn have published many informative articles about stress corrosion cracking in silica, alumina, glass, and other ceramics. Evans gives a good review of the various test configurations and procedures used to determine K_{IC} values for brittle materials [10]. In this article, Evans discusses single edge cracked tension specimens, three point bending, compact tension, double cantilever, tapered double cantilever, double torsion, constant moment, and wedge loaded double cantilever tests; equations for the corresponding K_I are also given. Also discussed is the relation between the crack growth rate and compliance of the specimen. Wiederhorn gives a good review and discussion of the characterization of subcritical crack growth in ceramics [45]. In Wiederhorn's article, crack propagation data for silica is presented showing the dependence of humidity and stress on crack velocity and the three characteristic regions of subcritical crack growth on $\frac{da}{dt}$ versus K_I plots. Region I, or the slow growth region, is attributable to a stress corrosion reaction at the crack tip enhanced by the presence of stress. In region I, crack growth depends on the rate of the chemical reaction at the crack tip, and exponentially on the applied stress. Region II, or the stress independent region, is a flat region where the crack velocity is independent of stress but depends strongly on the rate of water transport to the crack tip. Region III, or the fast growth region, is independent of water content and exponentially dependent on stress. It is generally accepted that crack growth in Region III is related to the structure of the material itself rather than environmental factors. Generally, these curves are determined by applying a static stress to a test specimen and measuring directly or indirectly the crack growth. Evans has shown that these three regions also characterize subcritical crack growth in ceramic materials that are dynamically loaded either through constant strain rate or stress rate conditions [11]. Magida *et al.* developed lifetime predictions and design criteria for static and dynamic loading of a machinable glass ceramic used in the construction of NASA's Gamma Ray Observatory [29]. Ritter *et al.* studied the effect of temperature and humidity on delayed failure of optical glass fibers [36]. The fatigue model used in the analysis exponentially relates crack velocity to an activation energy that is overcome by the presence of a stress intensity. Temperature and the partial pressure of water are included in the model directly. A comparison is made between the activation energy calculated and other values previously reported in the literature.

Fatigue data on silicon is less well documented and understood. Chen and Knapp performed a stress corrosion experiment with single crystal silicon bars that were pre-cracked by a Knoop microhardness tester and statically loaded in four point bending [6]. The surface of the beam was wetted with a wick with various liquids including distilled water, and the time to fracture recorded. The loading was on the order of 95% of the fracture load of those specimens which failed instantly. The beams which survived the initial loading were monitored for a period of up to two weeks. None failed, therefore Chen and Knapp concluded that stress corrosion cracking does not occur in the liquid environments they tested. However, it should be noted that direct measurements of the extension of the pre-crack were not made, and that it is possible the load was not sufficient to give a stress intensity above the stress corrosion threshold value. Calculations of the K_I factor for those beams which did not fail were not made. Wong and Holbrook performed a similar experiment where single crystal silicon wafers were indented and pre-cracked with a Leitz Miniload microhardness tester equipped with an optical microscope [49]. The radial cracks were measured as a function of time in both ambient air and deionized water. Wong and Holbrook reported they did not observe crack healing, nor crack extension, and ultimately concluded that stress corrosion cracking does not occur in silicon because of the formation of a protective oxide layer on the surface of the silicon. The explanation given is that the molar mismatch between the oxide and silicon causes a compressive stress at the crack tip which cancels the residual tensile field of the indentation thereby removing the driving force for crack extension. Although this explanation is entirely valid for indentation experiments, external loads need to be applied to maintain a K_I above the threshold for stress corrosion cracking to occur. It should also be noted that the silicon dioxide layer which forms on the surface of silicon exposed to air or water is subject to stress corrosion cracking. Bhaduri and Wang measured subcritical crack growth in single crystal silicon using load relaxation methods with pre-cracked double torsion specimens [2]. Plots of crack velocity versus stress intensity are presented with corresponding slopes and intercepts. Their results show region III behavior in ambient air at 10% humidity and liquid water. Interestingly, region II behavior was observed when an electric field was applied to the specimen. The slope and intercept of the fatigue curves associated with these cases appeared constant when voltages of 15 and 50 volts were applied. No definitive explanation was given for electrically enhanced subcritical cracking, although Bhaduri and Wang indicated that other researchers had found an electromechanical effect in silicon where hardness values decreased when a potential of a few volts was applied [44][1]. Such experiments would be of interest to the micromechanical community because micromechanical devices are electrically excited, and small currents do flow through the structure including flexural members that provide the necessary degrees of freedom. Bhaduri and Wang's findings for silicon in air and water are reasonable considering that silicon is instantly oxidized to form a thin layer of silicon dioxide, or silica, when exposed to a moist environment. It is the silica which is attacked by moisture. As a crack propagates through the material pure silicon is continuously exposed which is then oxidized and subjected to the stress corrosion reaction. Chen and Leipold used acoustic emission monitoring and load relaxation methods to determine if subcritical cracking was present in single crystal silicon [4]. Pre-cracked double torsion specimens were used. No load relaxation was observed during the hold times suggesting the absence of subcritical cracking. However, these hold times were on the order of 5-10 minutes which may not have been long enough to detect any changes. Acoustic emission monitoring also revealed no subcritical cracking during the hold period. However, it is unlikely a stress corrosion reaction in single crystal silicon would be detected by such a technique because of the absence of grain boundaries and discontinuities in the material that would produce a measurable emission.

1.2.5 Plasticity in Silicon

Plasticity in silicon has been investigated on a fundamental level by a number of researchers. Hirsch *et al.* studied the effect of doping on dislocation mobility in single crystal silicon using a microhardness indenter on a heated stage [37]. He found doping enhances dislocation motion, n doping being more effective than p doping, at temperatures of approximately $400°C$. Although this temperature is higher than the practical limits of a micromechanical device, the highest p type doping concentration used in the test was 2×10^{18} atoms/cm^{-3} which is considerably less than the concentrations used to fabricate microdevices. The typical surface concentration of a micromechanical device produced by boron diff''sion is on the order of 1×10^{20} atoms/cm^{-3}. Furthermore, silicon diffused with boron has a substantial dislocation density, which is not the case with wafers doped during the melt. Such wafers are virtually dislocation free. The mechanism of dislocation motion is attributed to electrically charged kinks which bridge secondary Peierls potentials. Theses kinks migrate in the direction of the dislocation line given adequate thermal energy and applied stress [20]. Haasen also developed expressions for kink migration based on activation energies necessary to overcome the Peierls potential [16]. Rybicki and Pirouz evaluated fracture and indentation plasticity in silicon using a microhardness tester and dislocation free float zone single crystal silicon [39]. P type silicon was boron doped in the melt to a concentration of 4.5×10^{17} atoms/cm^{-3}, and n type silicon was phosphorous doped to 6×10^{18} atoms/cm^{-3}. Rybicki and Pirouz found that doping affects hardness to a small extent, with n type doping giving the greatest dislocation mobility. The brittle to ductile transition temperature was reduced $15°C$ in the p material, and $35°C$ for the n type material. The DBTT for pure undoped silicon is $660°C$. Considering the high DBTT of silicon and the relatively small effect doping has on this temperature, room temperature dislocation motion should be negligibly small; therefore, dynamic fatigue of p-type silicon at room temperature is not expected. No doping related effects on fracture toughness were observed.

1.2.6 Conclusion

From review of the literature, fatigue in heavily boron doped micromechanical structures is an open question. Stress corrosion fatigue appears to be present in single crystal silicon, and may be enhanced by an electrical potential. Room temperature dynamic fatigue should not be a factor because of the high DBTT, and the relative insensitivity of this temperature to doping concentration.æ

Chapter 2

Fatigue Test Structure Design

2.1 Test Structure Layout (Generation 1)

Figures 2.1 and 2.2 are diagrams of the generation 1 micromechanical fatigue test structure. The dimensions shown correspond to the T38 series of fatigue devices which were the first operational devices fabricated providing preliminary test data. A cantilever beam $11.5\,\mu m \times 130\,\mu m \times 2.9\,\mu m$ thick is etched free from the surface of a silicon wafer. At the free end of the cantilever is a rectangular plate $120\,\mu m$ wide by $157\,\mu m$ long. This plate provides area for electrostatic forcing and sensing, and gold metallic plating which is used as a counterweight to lower and adjust the natural frequency. The volume etched beneath the beam and plate extends to a depth of approximately $70\,\mu m$. Gold bridge electrodes are used to electrostatically drive the cantilever and sense its motion.

$10\ \mu m$ from the base of the cantilever a pre-crack or crack initiation site is introduced into the surface of the beam. A small extension of a crack from this site reduces the total stiffness of the beam causing a significant change in resonant frequency. Therefore, driving the cantilever continuously at resonance while simultaneously measuring the frequency is a method by which the rate of propagation of a crack from the initiation site can be determined. The location of the crack initiation site, near the base of the cantilever, maximizes the frequency sensitivity to cracking and the bending moment applied to the crack.

The method of fatigue testing proposed in this experiment is analogous to

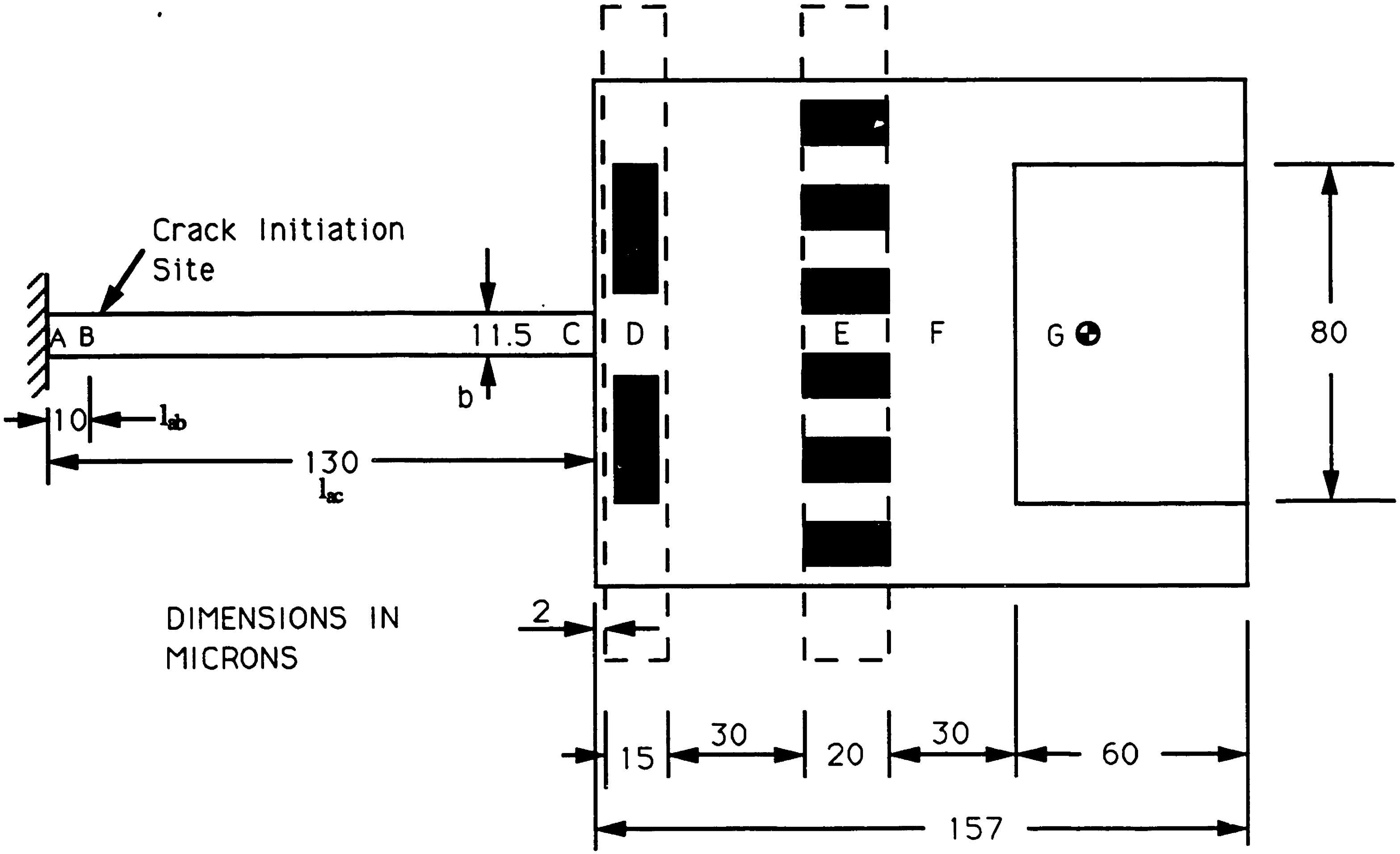


Figure 2.1: Micromechanical Test Structure (Plan View)

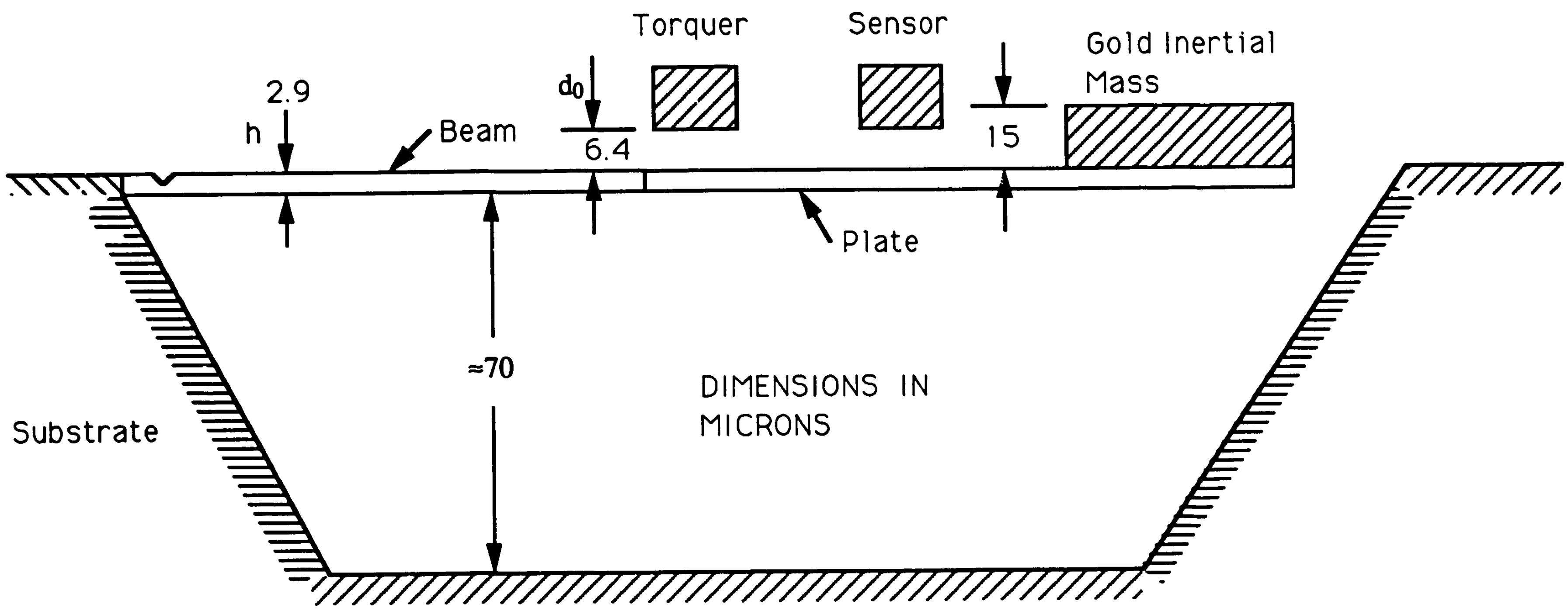


Figure 2.2: Micromechanical Test Structure (Side View)

compliance testing on a macro-scale where a constant load oı displacement is applied to a pre-cracked specimen and the corresponding change in load or displacement is measured as a function of time and correlated to crack growth. However, in this experiment, the frequency is measured which in turn is related to a stiffness change at the crack site.

The cantilever configuration has a number of advantages. First, and foremost, is simplicity. The geometry does not present unusual difficulties in analyzing the dynamics of the structure, or in determining the internal stresses. Furthermore, a tensile stress field will be generated in the vicinity of the crack initiation site so that mode I fracture and fatigue behavior should dominate. The device is free standing, meaning that it is connected at one point only to the base; therefore, the structure stress relieves itself from the lattice contraction caused by the boron diffusion[48]. If this contraction is not free to occur when the device is etched, as in a structure connected at two points, significant tensile stresses can be generated. A large pretension in the structure would likely cause uncontrolled breakage at the crack initiation site during etching. Lastly, the bridge electrodes allow a deep well to be etched beneath the beam and plate minimizing viscous air damping.

The dimensions corresponding to the plan view of the device, Figure 2.1, are fixed by the photomask. However, the thickness dimensions shown in the cross section, Figure 2.2, are process variables which can be varied by altering diffusion and deposition times.

2.2 Dynamic Modeling

An accurate dynamic model of the device is necessary to make predictions of the resonant frequency, and the anticipated frequency shift given a known crack geometry and extension. A number of alternatives are available for modeling. A three degree of freedom lumped system was chosen for three reasons. First, a discrete system simplifies the analysis versus solving the partial differential equation of a Bernoulli/Euler beam with a mass and inertia at the free end. Second, the effects of cracking and damping are easily included; plant modeling for control system design is simplified, and additional stiffnesses, such as that associated with flexing of the end plate, can be incorporated if necessary. Thirdly, for an uncracked uniform beam, two degrees of freedom appear to be the minimum necessary for accurate predictions of the first mode frequency because the large mass and inertia of the end plate produce translational and rotational inertial forces that tend to act independently in deflecting the beam. Single degree of freedom models do not fully account for this effect as shown later by the relatively high frequencies calculated.

In order to solve the equations of motion of the device many parameters must be computed: masses, inertias, stiffnesses, damping constants, and electrostatic forces. Furthermore, other important factors must be calculated including output voltages, control system parameters, stress intensities, surface tension forces, critical clearances in the electrode air gaps, electrode buckling calculations, and additional dynamics such as torsional modes and out of plane bending. To optimize the design of the fatigue structure many test cases must be analyzed until the best combination of dimensions and variable process parameters can be found that satisfies the requirements of the fatigue experiment and the fabrication process. Because of the large number of calculations involved in a single analysis, a design program was written in PASCAL to serve as the principal design, modeling, and analysis tool. This program is called the FTEST2.PAS design program. A copy of the code is included in Appendix G. Appendix G.1 contains the input file for the T38 design.

2.2.1 Derivation of Matrix Equations

In this section, Lagrange's equations are used to derive the matrix equations of motion of the system. A polynomial is chosen approximating the beam deflection and satisfying the geometric boundary conditions at the fixed end. Continuity must also be preserved on the interior of the beam to at least first order except at the crack site where a rotational spring that models the crack produces a discontinuity in slope. Equations 2.1 and 2.2 are used to as the approximation.

$$v_a(x,t) = v_1(t)\left(\frac{x}{\ell_{ac}}\right)^2 + v_2(t)\left(\frac{x}{\ell_{ac}}\right)^3 \qquad 0 \le x \le \ell_{ab} \qquad (2.1)$$

$$v_b(x,t) = v_1(t)\left(\frac{x}{\ell_{ac}}\right)^2 + v_2(t)\left(\frac{x}{\ell_{ac}}\right)^3 + \theta(x-\ell_{ab}) \qquad \ell_{ab} \le x \le \ell_{ac} \qquad (2.2)$$

The form of Equations 2.1 and 2.2 is that of a cantilever statically deflected by an external moment and force applied at the free end. By modeling this way, it is implicitly assumed the beam always responds statically to the inertial forces and torques of the end plate. Considering the large mass of the end plate ($M = 1.51 \times 10^{-9}$ kg), relative to the mass of the beam ($m_b = 9.97 \times 10^{-12}$ kg), the inertial forces of the beam will be almost negligible; therefore, the approximation should be close to the actual deflection. $v_1(t)$ and $v_2(t)$ are time varying independent degrees of freedom and $\theta(t)$ is the motion at the crack site. Equation 2.1 satisfies the geometric boundary conditions at the fixed end of the cantilever and preserves continuity to second order anywhere in the interior of the beam over the specified region; likewise for Equation 2.2. At $x = \ell_{ab}$, continuity in displacement must be maintained, however, continuity in slope is not because of the rotational spring, K_r. The boundary conditions therefore are:

$$v_a(0,t) = 0 \tag{2.3}$$

$$v_a(\ell_{ab},t) = v_b(\ell_{ab},t) \tag{2.4}$$

$$\left(\frac{\partial v_a(x,t)}{\partial x}\right)_{x=0} = 0 \tag{2.5}$$

$$\left(\frac{\partial v_b(x,t)}{\partial x}\right)_{x=\ell_{ab}} - \left(\frac{\partial v_a(x,t)}{\partial x}\right)_{x=\ell_{ab}} = \theta(t) \tag{2.6}$$

where

$$\frac{\partial v_a(x,t)}{\partial x} = \frac{2v_1}{\ell_{ac}^2}x + \frac{3v_2}{\ell_{ac}^3}x^2 \qquad 0 \le x \le \ell_{ab} \tag{2.7}$$

$$\frac{\partial v_b(x,t)}{\partial x} = \frac{2v_1}{\ell_{ac}^2}x + \frac{3v_2}{\ell_{ac}^3}x^2 + \theta \qquad \ell_{ab} \le x \le \ell_{ac} \tag{2.8}$$

Figure 2.3 defines the coordinate system and displacement vector of the deflected cantilever. Also shown are the nonconservative external forces of the driving electrode and damping.

Using the Bernoulli/Euler model, the strain energy stored in the cantilever can be written as follows.

$$V = \frac{EI}{2}\int_0^{\ell_{ab}} \left(\frac{\partial^2 v_a}{\partial x^2}\right)^2 dx + \frac{EI}{2}\int_{\ell_{ab}}^{\ell_{ac}} \left(\frac{\partial^2 v_b}{\partial x^2}\right)^2 dx + \frac{1}{2}K_r\theta^2 \tag{2.9}$$

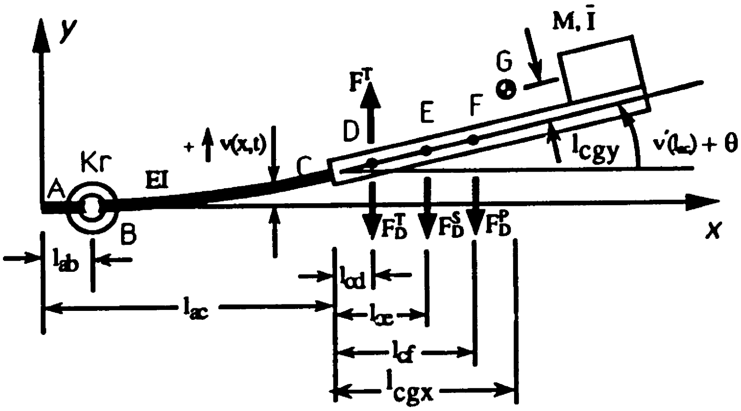


Figure 2.3: Micromechanical Cantilever

The strain energy stored in the cantilever reduces to an integration over the entire length of the beam, because the second derivative of v_a and v_b are identical. The strain energy can be simplified to the following equation.

$$V = \frac{EI}{2}\int_0^{\ell_{ac}}\left(\frac{\partial^2 v_a}{\partial x^2}\right)^2 dx + \frac{1}{2}K_r\theta^2 \tag{2.10}$$

The kinetic energy of the tip mass and beam are given by Equations 2.11 and 2.12 respectively.

$$\begin{aligned} T^*_M &= \frac{1}{2}M\left\{\left[\dot v_b(\ell_{ac}) + \left(\frac{\partial^2 v_b}{\partial x\partial t}\right)_{x=\ell_{ac}}\ell_{cgx}\right]^2 + \left[\left(\frac{\partial^2 v_b}{\partial x\partial t}\right)_{x=\ell_{ac}}\ell_{cgy}\right]^2\right\} \\ &\quad + \frac{1}{2}\bar I\left(\frac{\partial^2 v_b}{\partial x\partial t}\right)^2_{x=\ell_{ac}} \end{aligned} \tag{2.11}$$

$$T^*_B = \frac{1}{2}m\int_0^{\ell_{ab}} \dot v_a^2(x)\,dx + \frac{1}{2}m\int_{\ell_{ab}}^{\ell_{ac}} \dot v_b^2(x)\,dx \tag{2.12}$$

where,

$$m \;=\; \text{mass/length of beam: } \left(\frac{\text{kg}}{\text{m}}\right)$$

M = mass of end plate: (kg)

$\bar{I}$ = rotational inertia of end plate: (kg m^2)

ℓ_{cgx} = distance from point C to c.g. of end plate along x axis: (m)

ℓ_{cgy} = height of c.g. of end plate along y axis: (m)

Substituting Equations 2.1 and 2.2 into Equations 2.10, 2.11, and 2.12, and performing the indicated operations gives the following set of equations.

$$V = \frac{2EI}{\ell_{ac}^3}(v_1^2 + 3v_1v_2 + 3v_2^2) + \frac{1}{2}K_r\theta^2 \tag{2.13}$$

$$\begin{aligned} T_M^* &= \frac{1}{2}M\left\{((\zeta_1\dot{v}_1 + \zeta_2\dot{v}_2 + \zeta_3\dot{\theta}\ell_{bc})^2 + (\zeta_4\dot{v}_1 + \zeta_5\dot{v}_2 + \zeta_6\dot{\theta}\ell_{bc})^2\right\} \\ &\quad + \frac{1}{2}\bar{I}\left(\frac{2}{\ell_{ac}}\dot{v}_1 + \frac{3}{\ell_{ac}}\dot{v}_2 + \dot{\theta}\right)^2 \end{aligned} \tag{2.14}$$

where,

$$\begin{aligned} \zeta_1 &= \left(1 + 2\frac{\ell_{cgx}}{\ell_{ac}}\right) \\ \zeta_2 &= \left(1 + 3\frac{\ell_{cgx}}{\ell_{ac}}\right) \\ \zeta_3 &= \left(1 + \frac{\ell_{cgx}}{\ell_{bc}}\right) \\ \zeta_4 &= 2\frac{\ell_{cgy}}{\ell_{ac}} \\ \zeta_5 &= 3\frac{\ell_{cgy}}{\ell_{ac}} \\ \zeta_6 &= \frac{\ell_{cgy}}{\ell_{bc}} \end{aligned}$$

$$\begin{aligned} T_B^* &= \frac{m\ell_{ac}}{2}\left(\frac{\dot{v}_1^2}{5} + \frac{\dot{v}_1\dot{v}_2}{3} + \frac{\dot{v}_2^2}{7}\right) + m\frac{\dot{v}_1\dot{\theta}}{\ell_{ac}^2}\ell_{ab}^4\left(\frac{1}{12} - \frac{1}{3}\left(\frac{\ell_{ac}}{\ell_{ab}}\right)^3 + \frac{1}{4}\left(\frac{\ell_{ac}}{\ell_{ab}}\right)^4\right) \\ &\quad + m\frac{\dot{v}_2\dot{\theta}}{\ell_{ac}^3}\ell_{ab}^5\left(\frac{1}{20} - \frac{1}{4}\left(\frac{\ell_{ac}}{\ell_{ab}}\right)^4 + \frac{1}{5}\left(\frac{\ell_{ac}}{\ell_{ab}}\right)^5\right) \\ &\quad + \frac{m\dot{\theta}^2}{2}\left(\frac{\ell_{ac}^3}{3} + \ell_{ac}\ell_{ab}(\ell_{ab} - \ell_{ac}) - \frac{\ell_{ab}^3}{3}\right) \end{aligned} \tag{2.15}$$

For $\ell_{ac} \gg \ell_{ab}$ and $\ell_{ac} \approx \ell_{bc}$

$$\begin{aligned}
\left(\frac{1}{12} - \frac{1}{3}\left(\frac{\ell_{ac}}{\ell_{ab}}\right)^3 + \frac{1}{4}\left(\frac{\ell_{ac}}{\ell_{ab}}\right)^4\right) &\approx \frac{1}{4}\left(\frac{\ell_{ac}}{\ell_{ab}}\right)^4 \\
\left(\frac{1}{20} - \frac{1}{4}\left(\frac{\ell_{ac}}{\ell_{ab}}\right)^4 + \frac{1}{5}\left(\frac{\ell_{ac}}{\ell_{ab}}\right)^5\right) &\approx \frac{1}{5}\left(\frac{\ell_{ac}}{\ell_{ab}}\right)^5 \\
\left(\frac{\ell_{ac}^3}{3} + \ell_{ac}\ell_{ab}(\ell_{ab} - \ell_{ac}) - \frac{\ell_{ab}^3}{3}\right) &\approx \frac{\ell_{ac}^3}{3}
\end{aligned}$$

and with the following definitions

$$\begin{aligned}
v_3 &= \theta\ell_{bc} \\
K_B &= \frac{K_r}{\ell_{bc}^2}
\end{aligned}$$

the potential and kinetic energies can be rewritten as follows.

$$V = \frac{2EI}{\ell_{ac}^3}(v_1^2 + 3v_1v_2 + 3v_2^2) + \frac{1}{2}K_Bv_3^2 \tag{2.16}$$

$$\begin{aligned}
T_M^* &= \frac{1}{2}M\left\{((\zeta_1\dot{v}_1 + \zeta_2\dot{v}_2 + \zeta_3\dot{v}_3)^2 + (\zeta_4\dot{v}_1 + \zeta_5\dot{v}_2 + \zeta_6\dot{v}_3)^2\right\} \\
&+ \frac{1}{2}\bar{I}\left(\frac{2}{\ell_{ac}}\dot{v}_1 + \frac{3}{\ell_{ac}}\dot{v}_2 + \frac{\dot{v}_3}{\ell_{bc}}\right)^2
\end{aligned} \tag{2.17}$$

$$T_B^* = \frac{m\ell_{ac}}{2}\left(\frac{\dot{v}_1^2}{5} + \frac{\dot{v}_1\dot{v}_2}{3} + \frac{\dot{v}_2^2}{7}\right) + \frac{m\ell_{ac}}{4}\dot{v}_1\dot{v}_3 + \frac{m\ell_{ac}}{5}\dot{v}_2\dot{v}_3 + \frac{m\ell_{ac}}{6}\dot{v}_3^2 \tag{2.18}$$

Using Equations 2.16, 2.17, and 2.18, the Lagrangian, L, can be written in terms of v_1, v_2, and v_3.

$$L = T_M^* + T_B^* - V. \tag{2.19}$$

Lagrange's equations are written as follows.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{v}_1}\right) - \frac{\partial L}{\partial v_1} = \Xi_1 \tag{2.20}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{v}_2}\right) - \frac{\partial L}{\partial v_2} = \Xi_2 \tag{2.21}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{v}_3}\right) - \frac{\partial L}{\partial v_3} = \Xi_3 \tag{2.22}$$

Substituting Equations 2.16, 2.17, 2.18, and 2.19 into Equations 2.20 through 2.22 gives

$$\left(M(\zeta_1^2+\zeta_4^2)+\frac{4\,\bar{I}}{\ell_{ac}^2}+\frac{m\ell_{ac}}{5}\right)\ddot{v}_1+\left(M(\zeta_1\zeta_2+\zeta_4\zeta_5)+\frac{6\,\bar{I}}{\ell_{ac}^2}+\frac{m\ell_{ac}}{6}\right)\ddot{v}_2+$$
$$\left(M(\zeta_1\zeta_3+\zeta_4\zeta_6)+\frac{2\bar{I}}{\ell_{ac}\ell_{bc}}+\frac{m\ell_{ac}}{4}\right)\ddot{v}_3+\frac{4\,EI}{\ell_{ac}^3}v_1+\frac{6\,EI}{\ell_{ac}^3}v_2=\Xi_1 \qquad (2.23)$$

$$\left(M(\zeta_1\zeta_2+\zeta_4\zeta_5)+\frac{6\,\bar{I}}{\ell_{ac}^2}+\frac{m\ell_{ac}}{14}\right)\ddot{v}_1+\left(M(\zeta_2^2+\zeta_5^2)+\frac{9\,\bar{I}}{\ell_{ac}^2}+\frac{m\ell_{ac}}{7}\right)\ddot{v}_2+$$
$$\left(M(\zeta_2\zeta_3+\zeta_5\zeta_6)+\frac{3\bar{I}}{\ell_{ac}\ell_{bc}}+\frac{m\ell_{ac}}{5}\right)\ddot{v}_3+\frac{6\,EI}{\ell_{ac}^3}v_1+\frac{12\,EI}{\ell_{ac}^3}v_2=\Xi_2 \qquad (2.24)$$

$$\left(M(\zeta_1\zeta_3+\zeta_4\zeta_6)+\frac{2\bar{I}}{\ell_{ac}\ell_{bc}}+\frac{m\ell_{ac}}{4}\right)\ddot{v}_1+$$
$$\left(M(\zeta_2\zeta_3+\zeta_5\zeta_6)+\frac{3\bar{I}}{\ell_{ac}\ell_{bc}}+\frac{m\ell_{ac}}{5}\right)\ddot{v}_2+$$
$$\left(M(\zeta_3^2+\zeta_6^2)+\frac{\bar{I}}{\ell_{bc}^2}+\frac{m\ell_{ac}}{3}\right)\ddot{v}_3+K_Bv_3=\Xi_3 \qquad (2.25)$$

Expressions for the generalized forces, Ξ_1, Ξ_2, and Ξ_3 will be derived in a later section. First, the frequencies of the system will be computed and compared to that obtained for single degree of freedom models. Defining

$$m_{11} = M(\zeta_1^2+\zeta_4^2)+\frac{4\bar{I}}{\ell_{ac}^2}+\frac{m\ell_{ac}}{5} \qquad (2.26)$$

$$m_{12} = M(\zeta_1\zeta_2+\zeta_4\zeta_5)+\frac{6\bar{I}}{\ell_{ac}^2}+\frac{m\ell_{ac}}{6} \qquad (2.27)$$

$$m_{13} = M(\zeta_1\zeta_3+\zeta_4\zeta_6)+\frac{2\bar{I}}{\ell_{ac}\ell_{bc}}+\frac{m\ell_{ac}}{4} \qquad (2.28)$$

$$m_{21} = m_{12} \qquad (2.29)$$

$$m_{22} = M(\zeta_2^2+\zeta_5^2)+\frac{9\bar{I}}{\ell_{ac}^2}+\frac{m\ell_{ac}}{7} \qquad (2.30)$$

$$m_{23} = M(\zeta_2\zeta_3+\zeta_5\zeta_6)+\frac{3\bar{I}}{\ell_{ac}\ell_{bc}}+\frac{m\ell_{ac}}{5} \qquad (2.31)$$

$$
\begin{aligned}
m_{31} &= m_{13} && (2.32)\\
m_{32} &= m_{23} && (2.33)\\
m_{33} &= M(\zeta_3^2+\zeta_6^2)+\frac{\bar{I}}{\ell_{bc}^2}+\frac{m\ell_{ac}}{3} && (2.34)\\
k_{11} &= \frac{4\,EI}{\ell_{ac}^3} && (2.35)\\
k_{12} &= \frac{6\,EI}{\ell_{ac}^3} && (2.36)\\
k_{13} &= 0 && (2.37)\\
k_{21} &= k_{12} && (2.38)\\
k_{22} &= \frac{12\,EI}{\ell_{ac}^3} && (2.39)\\
k_{23} &= 0 && (2.40)\\
k_{31} &= k_{13} && (2.41)\\
k_{32} &= k_{23} && (2.42)\\
k_{33} &= K_B && (2.43)
\end{aligned}
$$

the matrix equation of motion can be written as

$$
\begin{bmatrix} m_{11} & m_{12} & m_{13}\\ m_{21} & m_{22} & m_{23}\\ m_{31} & m_{32} & m_{33}\end{bmatrix}
\begin{Bmatrix} \ddot{v}_1\\ \ddot{v}_2\\ \ddot{v}_3\end{Bmatrix}
+
\begin{bmatrix} k_{11} & k_{12} & k_{13}\\ k_{21} & k_{22} & k_{23}\\ k_{31} & k_{32} & k_{33}\end{bmatrix}
\begin{Bmatrix} v_1\\ v_2\\ v_3\end{Bmatrix}
=
\begin{Bmatrix} \Xi_1\\ \Xi_2\\ \Xi_3\end{Bmatrix}
\qquad (2.44)
$$

or,

$$\underline{\mathbf{M}}\,\underline{\ddot{\mathbf{U}}}+\underline{\mathbf{K}}\,\underline{\mathbf{U}}=\underline{\mathbf{R}} \qquad (2.45)$$

Defining

$$\begin{bmatrix} m_{11} & m_{12}\\ m_{21} & m_{22}\end{bmatrix} = \underline{\mathbf{M}}_A \qquad (2.46)$$

$$\begin{bmatrix} m_{13}\\ m_{23}\end{bmatrix} = \underline{\mathbf{M}}_C \qquad (2.47)$$

$$\begin{bmatrix} m_{13} & m_{23}\end{bmatrix} = \underline{\mathbf{M}}_C^T \qquad (2.48)$$

$$m_{33} = M_B \qquad (2.49)$$

$$\begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} = \underline{\mathbf{K}}_A \tag{2.50}$$

$$\begin{bmatrix} k_{13} \\ k_{23} \end{bmatrix} = \underline{\mathbf{K}}_C \tag{2.51}$$

$$\begin{bmatrix} k_{13} & k_{23} \end{bmatrix} = \underline{\mathbf{K}}_C^T \tag{2.52}$$

$$\left\{ \begin{matrix} \ddot{v}_1 \\ \ddot{v}_2 \\ \ddot{v}_3 \end{matrix} \right\} = \left\{ \begin{matrix} \underline{\mathbf{U}}_A \\ \ddot{v}_3 \end{matrix} \right\} \tag{2.53}$$

$$\left\{ \begin{matrix} \Xi_1 \\ \Xi_2 \\ \Xi_3 \end{matrix} \right\} = \left\{ \begin{matrix} \underline{\mathbf{R}}_A \\ \Xi_3 \end{matrix} \right\} \tag{2.54}$$

$$\begin{bmatrix} \underline{\mathbf{M}}_A & \underline{\mathbf{M}}_C \\ \underline{\mathbf{M}}_C^T & M_B \end{bmatrix} \left\{ \begin{matrix} \ddot{\underline{\mathbf{U}}}_A \\ \ddot{v}_3 \end{matrix} \right\} + \begin{bmatrix} \underline{\mathbf{K}}_A & \underline{\mathbf{K}}_C \\ \underline{\mathbf{K}}_C^T & K_B \end{bmatrix} \left\{ \begin{matrix} \underline{\mathbf{U}}_A \\ v_3 \end{matrix} \right\} = \left\{ \begin{matrix} \underline{\mathbf{R}}_A \\ \Xi_3 \end{matrix} \right\} \tag{2.55}$$

To analyze the dynamics of the cantilever system without cracking, the third row and column of the matrix equation of motion is eliminated giving the following equation.

$$\underline{\mathbf{M}}_A \, \ddot{\underline{\mathbf{U}}}_A + \underline{\mathbf{K}}_A \, \underline{\mathbf{U}}_A = \underline{\mathbf{R}}_A \tag{2.56}$$

The free vibration problem is solved by setting $\underline{\mathbf{R}}_A = 0$ and substituting a solution of the form

$$\underline{\mathbf{U}}_A = \underline{\phi}_A e^{j\omega t} \tag{2.57}$$

into Equation 2.56. For nontrivial solutions,

$$|\underline{\mathbf{K}}_A - \omega^2 \underline{\mathbf{M}}_A| = 0 \tag{2.58}$$

The determinant is expanded and solved for the natural frequencies:

$$\omega_{1_A} = \sqrt{\frac{A - (A^2 - 4B)^{1/2}}{2}} \tag{2.59}$$

$$\omega_{2_A} = \sqrt{\frac{A + (A^2 - 4B)^{1/2}}{2}} \tag{2.60}$$

where,

$$\begin{aligned} A &= \frac{k_{11}m_{22} + k_{22}m_{11} - 2m_{12}k_{12}}{m_{11}m_{22} - m_{12}^2} \\ B &= \frac{k_{11}k_{22} - k_{12}^2}{m_{11}m_{22} - m_{12}^2} \end{aligned}$$

Numerical values will be given for ω_{1_A} and ω_{2_A}, but first two single degree of freedom models will be evaluated.

2.2.2 Single Degree of Freedom Models

Two single degree of freedom models are now presented which illustrate the relative inaccuracy of these idealizations in computing the natural frequency. In the first model, rotational inertia effects are neglected by assuming the motion is predominately translational. A point load is applied at the center of gravity of the structure, point G, in Figure 2.3 and the deflection computed at point C, v_c. The spring constant associated with v_c is as follows.

$$K_c = \frac{3\ EI}{\ell_{ac}^3\left(1 + \frac{3\ell_{cgx}}{2\ell_{ac}}\right)} \tag{2.61}$$

Excluding damping, summing forces in the vertical direction at point C gives the following equation of motion.

$$M\ddot{v}_c + K_c v_c = F_T \tag{2.62}$$

The frequency for this model is given as follows.

$$\omega_c = \sqrt{\frac{K_c}{M}} \tag{2.63}$$

The second model includes rotational inertia. Summing moments about point A in Figure 2.3 gives the equation of motion.

$$\bar{I}_A\ddot{\theta} + K_\theta\theta = T_A \tag{2.64}$$

where,

$$T_A \;=\; F_T(\ell_{ac} + \ell_{cd})$$

DOF	Model Type	Frequency (Hz)
2	Lagrangian	$\omega_{1_A} = 2745$
		$\omega_{2_A} = 102153$
1	Translational	$\omega_c = 4554$
1	Rotational	$\omega_\theta = 3246$

Table 2.1: Frequency Comparison for T38 Series

$$\begin{aligned}
\bar{I}_A &= M(\ell_{cgx} + \ell_{ac})^2 + \bar{I} \\
\theta &= \frac{v_c}{\ell_{ac}} \\
K_\theta &= \frac{3\,EI}{\ell_{ac}}\left(\frac{1 + \frac{\ell_{cgx}}{\ell_{ac}}}{1 + \frac{3\ell_{cgx}}{2\ell_{ac}}}\right)
\end{aligned}$$

The natural frequency for this model is as follows.

$$\omega_\theta = \sqrt{\frac{K_\theta}{\bar{I}_A}} \tag{2.65}$$

2.2.3 Model Comparison

Table 2.1 compares the calculated frequencies for the different dynamic models. Details of these calculations are performed in Appendix A. Although it may appear that the two degree of freedom model is a rather complex and elaborate approach to a relatively simple problem, Table 2.1 shows substantial differences in the calculated frequencies for the single degree of freedom models versus the two degree of freedom model. Because the frequencies are higher, the single degree of freedom models represent stiffer structures; therefore, to obtain a particular K_I at the crack site a lower displacement would be calculated than for the two degree of freedom model. This fact has important implications on the overall design of the structure because the total travel of the end plate is limited by the air gap dimension under the bridge electrodes. This dimension is constrained by the fabrication process to at most about $12\mu m$. In essence, accurate dynamic modeling of the device in this investigation is the key to obtaining the correct stress intensity at the crack initiation site given the limited travel ot the end plate. It is the translational and rotational inertia forces of the end plate that must produce the necessary displacements and stresses to initiate and propagate a subcritical crack through the material. Fatigue in brittle solids occurs only over a narrow range of stress intensity factors; therefore, careful design and accurate predictions of frequency and displacement is necessary for success.

2.2.4 Finite Element Modal Analysis

As explained in the previous section, accurate dynamic modeling of the structure is critical for success. In the derivation of the equations of motion for the cantilever system, two key assumptions were made which could have significant bearing on the accuracy of the model. First, the end plate was assumed to act as a rigid body, and second, a 2 degree of freedom Bernoulli/Euler model was assumed to be an adequate representation of the displacement of the uncracked beam. In fact, the end plate flexes a small amount, some shear deformation occurs in the beam, and the polynomial representing the beam displacement is more complex than assumed in Equations 2.1 and 2.2. Therefore, to verify the validity of the equations of motion, and to check the calculations performed in the design program, FTEST2.PAS, a finite element modal analysis of the structure was performed using ABAQUS. The dynamics of the end plate were analyzed separately to determine if additional degrees of freedom should be included in the equations of motion to account for the flexibility of the end plate. The input data lists and pertinent output data for the cantilever system and end plate are included in Appendix B.

Modal Analysis of End Plate

Figure 2.4 is a plot of the end plate mesh. The dimensions are identical to those shown in Figure 2.1. Four node shell elements model the silicon plate, and eight node brick elements model the gold counterweight. A Cartesian rectangular coordinate system is located on the centerline of the plate on the fixed edge as shown in Figure 2.4. Figures 2.5 through 2.7 display the first three modes of the end plate. The frequency of the first bending mode of the end plate determines whether or not the end plate can be treated as a rigid body in the dynamic analysis. If this frequency is at least a factor of 4 larger than the first mode of the plate and cantilever structure, the dynamics of the end plate can be ignored. The finite element analysis computes this first mode frequency to be 25.6 KHz which is a factor of 9 greater than the

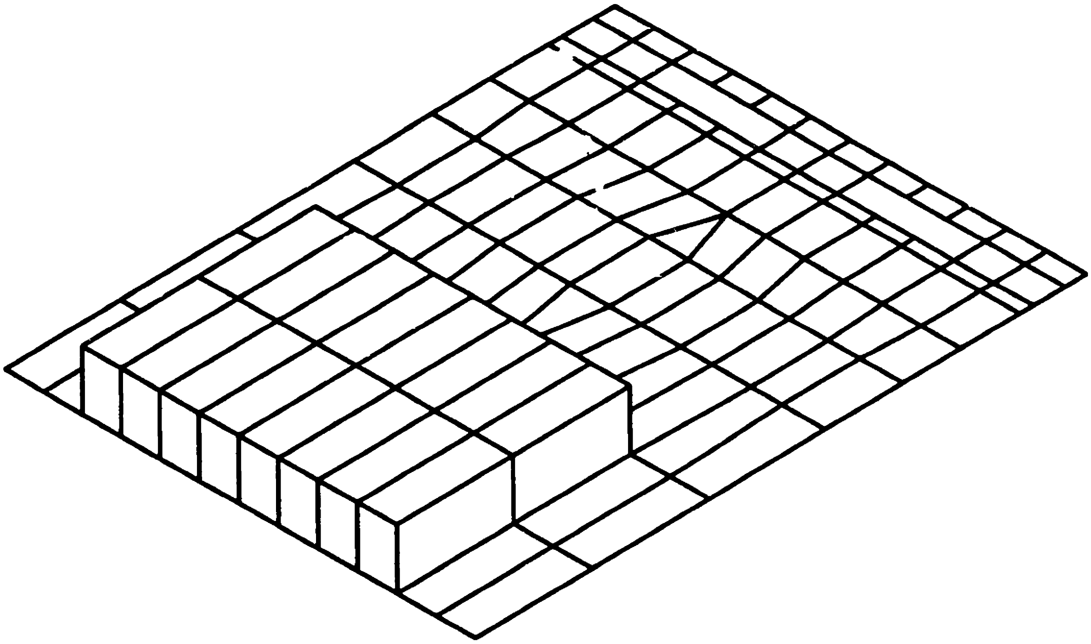


Figure 2.4: End Plate Mesh

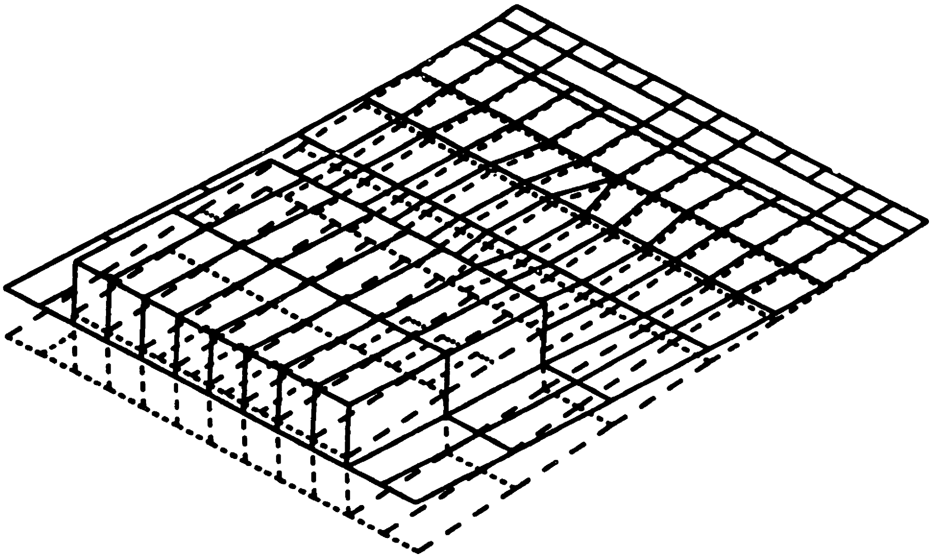


Figure 2.5: Mode 1 (Bending) $\omega_1 = 25.6$ KHz

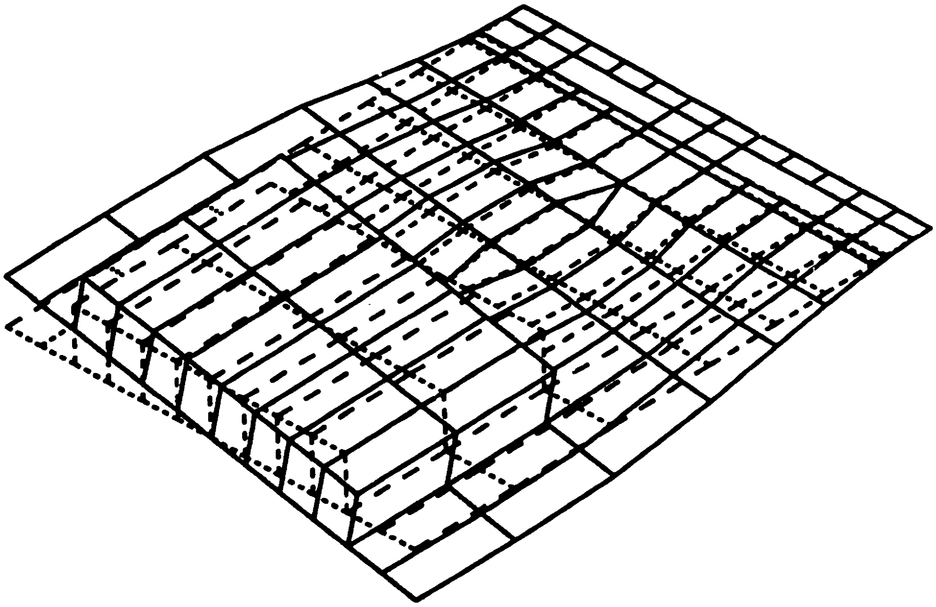


Figure 2.6: Mode 2 (Torsion) $\omega_2 = 148.34$ KHz

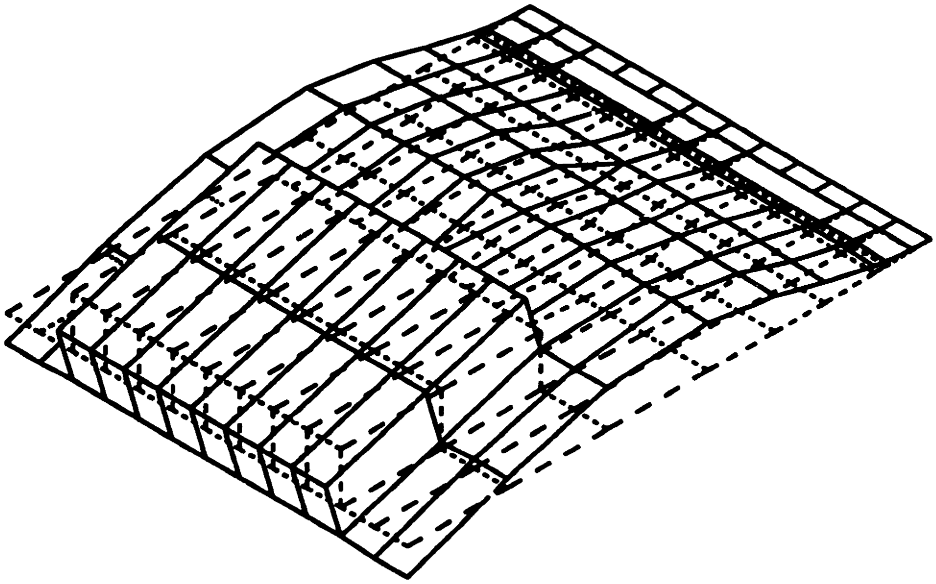


Figure 2.7: Mode 3 (Bending) $\omega_3 = 291.82$ KHz

first mode of the system, 2745 Hz. Given the large separation of the two frequencies in the frequency spectrum, the dynamics of the end plate will have negligible effect and can be ignored.

The FTEST2.PAS design program also computes the first mode of the end plate using beam theory and a single degree of freedom dynamic model. The model used in the program is similar to the one presented in Section 2.2.2 that includes rotational inertia. The design program computes a value of 28.5 KHz which is approximately 10% higher than the value computed from the finite element analysis. As expected, the single degree of freedom approach leads to a stiffer structure with a higher natural frequency; however, the level of accuracy obtained with the design program is sufficient to ensure that flexing of the end plate is not significant.

Modal Analysis Cantilever System

Figure 2.8 is a plot of the cantilever and plate system mesh. Thirteen two node beam elements model the cantilever. The model of the end plate remains unchanged. Figures 2.9 through 2.11 display the first three modes of the system. The first bending mode of the cantilever and end plate sys-

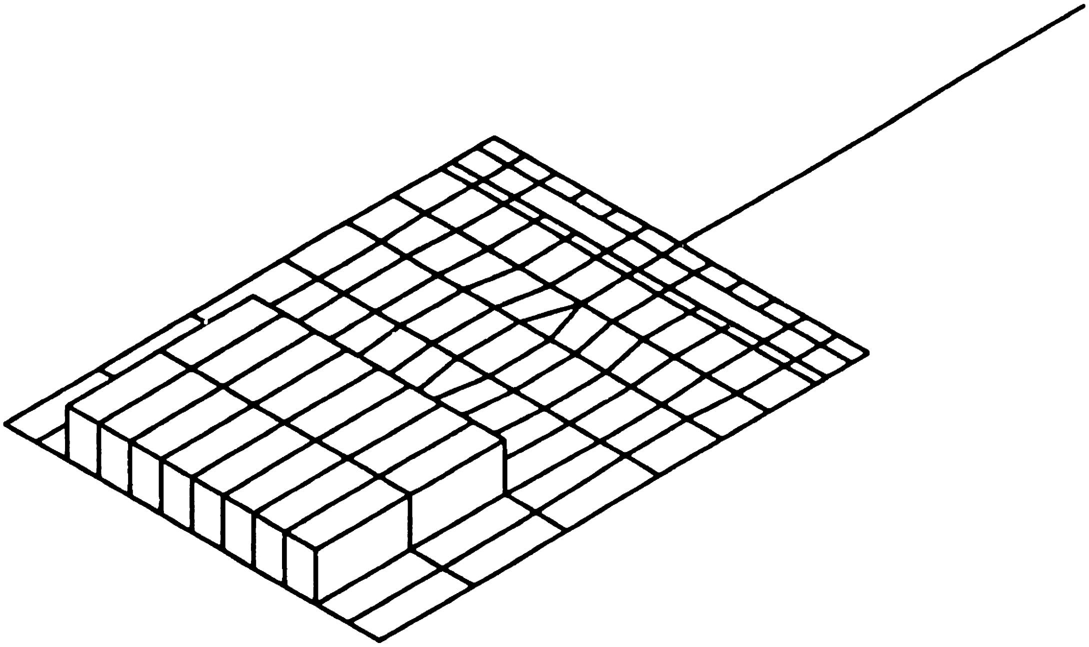


Figure 2.8: Mesh Plot of Cantilever and Plate System

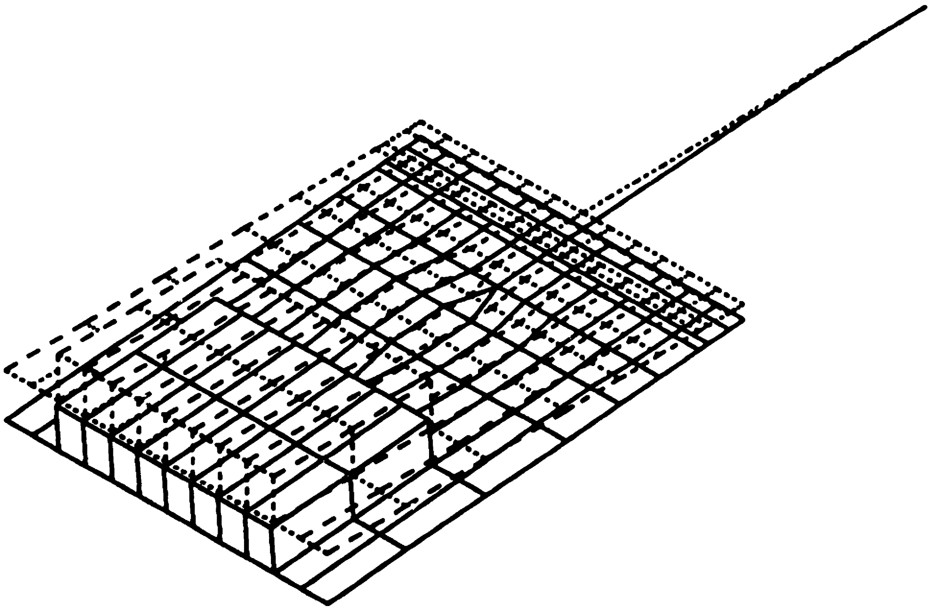


Figure 2.9: Mode 1 (Bending) $\omega_1 = 2720$ Hz

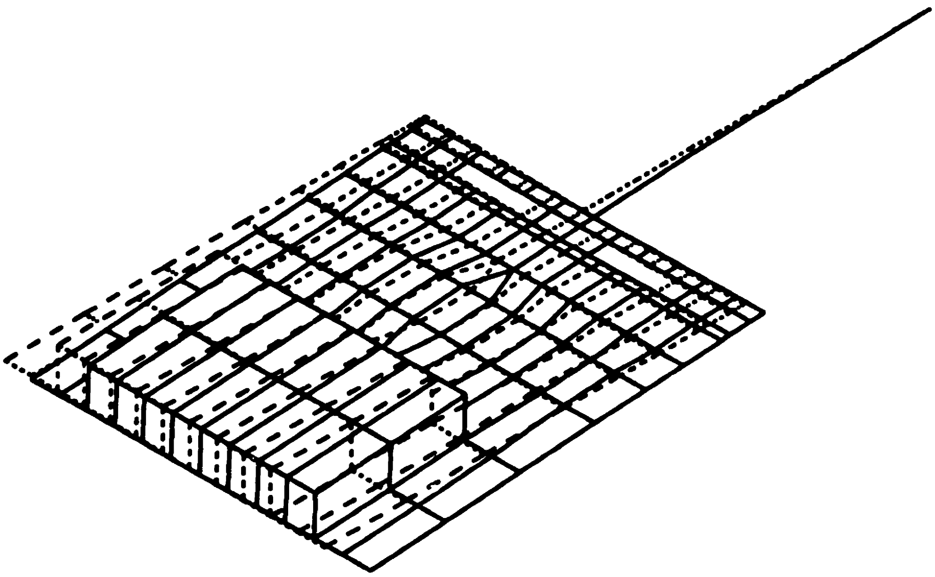


Figure 2.10: Mode 2 (Bending) $\omega_2 = 10.3$ KHz

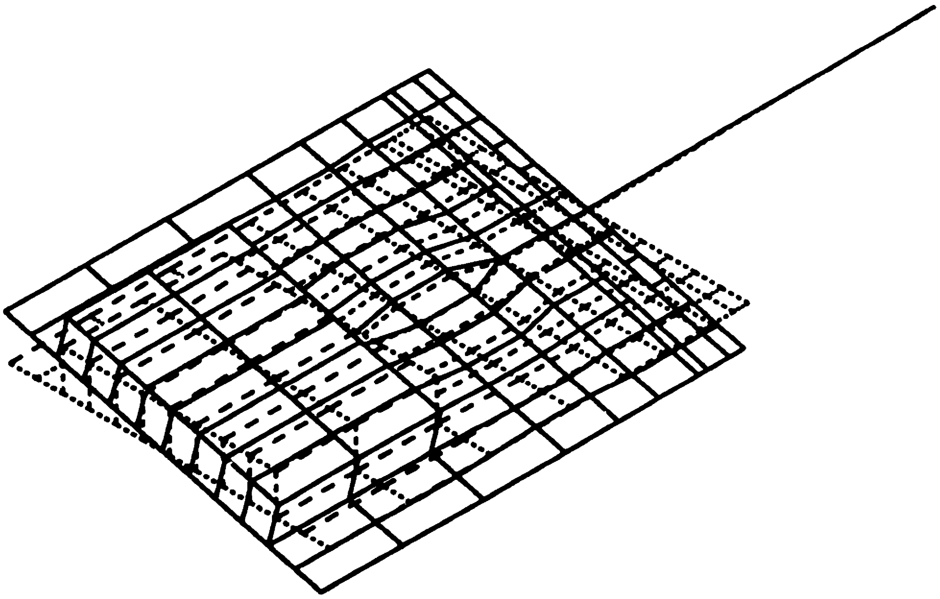


Figure 2.11: Mode 3 (Torsion) $\omega_3 = 21.0$ KHz

tem has a frequency of 2720 Hz which is within 1% of the value, 2745 Hz, computed by the design program FTEST2.PAS. The second mode is a combination of bending in the xy plane and torsion about the x axis. The torsional component is small and occurs by dynamic coupling through the product of inertia Ixz. The cross product, Ixz, occurs because the center of gravity of the structure is located ap; roximately $8.3\mu m$ above the x axis; therefore, motion in the xy plane couples into torsion about the x axis because the translational inertia forces acting at the center of mass acts through a lever arm with respect to the lengthwise axis of the beam. The bending frequency in the xy plane computed by the design program is 12.8 KHz which is 19% higher than the frequency computed by ABAQUS, 10.3 KHz. The higher frequency is accounted for by two factors. First, a single degree of freedom model for bending is employed which is inherently stiffer than a multi degree of freedom model; and second, the torsional coupling is neglected. Although, the calculated value is somewhat high, the accuracy achieved with the design program is still adequate enough to ensure this mode occurs at a frequency approximately a factor of 4 greater than the first bending mode. The third mode is again a combination of torsion about the x axis and bending in the xy plane. Contrary to the second mode, the torsional component is large and the bending component small. The design program calculates a torsional frequency of 22.5 KHz, which is 7% greater than the value, 21.0 KHz, computed by ABAQUS. Better accuracy is obtained in predicting this mode because the beam is prismatic (a constant cross section) so that torsion is accurately modeled by a single degree of freedom: the angular rotation of one cross section relative to the next. The difference in calculated frequency is again caused by neglecting the coupling into bending motion in the xy plane.

2.2.5 Nonlinear Dynamics

In Section 2.2.1, the equation of motion for the cantilever system, Equation 2.55, was derived including a rotational spring to account for the additional compliance of the pre-crack. However, this compliance behaves nonlinearly. As the beam travels upward through the zero displacement position, the crack closes giving zero compliance for the rotational spring; likewise, as the crack opens, the compliance of the spring increases to its previous value. To analyze this problem, the equation of motion for the uncracked beam can be solved, and at the zero displacement position, use this solution as an initial condition for the equation of motion of the cracked beam system. When the cantilever returns to the zero displacement position, use the solution of the cracked beam dynamics as an input to the uncracked system, and so on until a steady state condition is reached. The frequency at this condition will be close to an average of the first mode frequencies of the two systems. This approach is purely numerical, and therefore, lacks the advantages of working with linear equations. Furthermore, insight is lost by not having simple mathematical expressions to examine. Rather, a quasi-linearization of the system is desirable such that Equation 2.55 can be corrected accordingly, then analyzed using standard linear modal analysis techniques.

Dynamic systems with nonlinear elements can be analyzed using "describing functions". The method has been extensively developed for nonlinear control systems which operate with nonlinearities such as relays, backlash, static friction, and actuator saturation [43]. It is assumed that given a periodic input to the system, the effects of the nonlinearities are smoothed out by feedback and the filtering of the controller and plant. Therefore, the output remains essentially periodic despite the nonlinearities.

To determine the effect of the nonlinear compliance of the pre-crack, the moment at any point in the beam is first computed.

$$M_0(x,t) = EI\frac{\partial^2 v_a(x,t)}{\partial x^2} \quad (2.66)$$

$$= EI\frac{\partial^2 v_b(x,t)}{\partial x^2} \quad (2.67)$$

Either equation above will suffice in computing M_0, for the second derivative of v_a and v_b are equal. Setting $x = \ell_{ab}$ gives the moment at the crack site which can be written in vector notion as follows. Recall that $\underline{\mathbf{U}}(t)$ is the vector of generalized coordinates, $v_1(t)$, $v_2(t)$, and $v_3(t)$.

$$M_0(\ell_{ab},t) = \left[\begin{array}{ccc} \frac{2EI}{\ell_{ac}^2} & \frac{6EI}{\ell_{ac}^3} & 0 \end{array} \right] \underline{\mathbf{U}}(t) \quad (2.68)$$

Thus, the moment applied at the crack site is not a function of v_3, the degree of freedom associated with rotation at the pre-crack, rather, it is purely a function of the sinusoidal motion of the uncracked cantilever system. For the first mode, the moment applied at the pre-crack is written as follows.

$$M_0(\ell ab,t) = \left[\begin{array}{ccc} \frac{2EI}{\ell_{ac}^2} & \frac{6EI}{\ell_{ac}^3} & 0 \end{array} \right] \underline{\phi}_1 x_0 \sin(\omega t) \quad (2.69)$$

$$= M_0 \sin(\omega t) \quad (2.70)$$

where,

$$\underline{\mathrm{U}}(t) \approx \underline{\phi}_1 x_0 \sin(\omega t) \tag{2.71}$$

This sinusoidal moment is the input to the nonlinear crack system as shown in Figure 2.12. The equation for C_θ, the crack compliance, is given by Murakami and is discussed in the chapter dealing with fatigue modeling [34]. From our earlier derivation of the equations of motion, by definition:

$$C_\theta = \frac{1}{K_r} \tag{2.72}$$

and

$$k_{33} = \frac{K_r}{\ell_{bc}^2} \tag{2.73}$$

The output of the nonlinear crack system is a truncated sine wave which can be expanded into a Fourier series.

$$\theta(t) = \sum_{n=-\infty}^{+\infty} \theta_n \exp^{jn\omega t} \tag{2.74}$$

$$\theta(t) = C_\theta M_0 \left[-\frac{1}{\pi} + \frac{1}{2}\sin(\omega t) + \frac{2}{\pi}\left(\frac{\cos(2\omega t)}{3} + \frac{\cos(4\omega t)}{3 \cdot 5} + \frac{\cos(6\omega t)}{5 \cdot 7} + \cdots \right) \right] \tag{2.75}$$

The higher harmonics are assumed to be filtered by the inertia and damping of the system giving the following response.

$$\theta(t) = C_\theta M_0 \left(-\frac{1}{\pi} + \frac{1}{2}\sin(\omega t) \right) \tag{2.76}$$

Because the nonlinearity in the spring stiffness is unsymmetric, a static displacement, θ_s, is created which is a function of the applied load.

$$\theta_s = -\frac{C_\theta M_0}{\pi} \tag{2.77}$$

From Equation 2.76, the coefficient of $M_0 \sin(\omega t)$ is $C_\theta/2$ which represents the linear approximation of the nonlinear crack system; therefore, to linearize Equation 2.55, k_{33} is replaced with $2k_{33}$. Essentially, this linearization is nothing more than taking the average of C_θ and 0. To test the validity of the

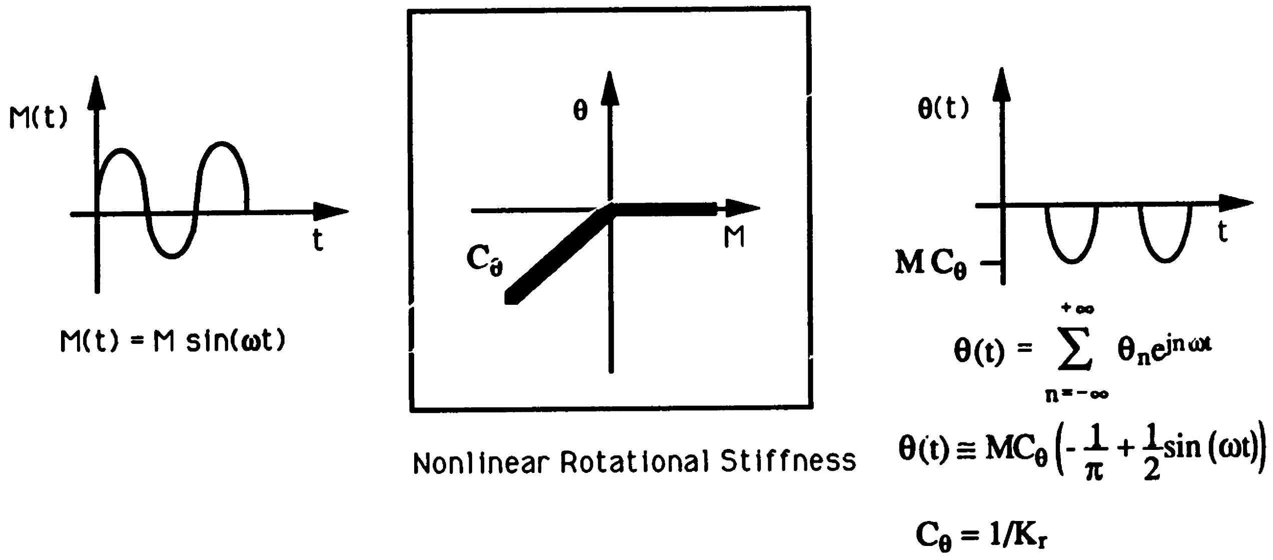

M(t)
t
$M(t) = M \sin(\omega t)$
θ
C_θ
M
Nonlinear Rotational Stiffness
$\theta(t)$
MC_θ
t
$\theta(t) = \sum_{n=-\infty}^{+\infty} \theta_n e^{jn\omega t}$
$\theta(t) \equiv MC_\theta \left(-\frac{1}{\pi} + \frac{1}{2}\sin(\omega t)\right)$
$C_\theta = 1/K_r$

Figure 2.12: Nonlinear Compliance of Precrack

describing function approach the FTEST2.PAS design program was used to extract the first mode solutions for both the cracked beam system, $C_\theta \neq 0$, and the uncracked beam system, $C_\theta = 0$. The first mode equation for the uncracked system is

$$\ddot{x} + c\dot{x} + k_1 x = f\sin(\omega t) \qquad \text{x} > 0 \tag{2.78}$$

and the cracked system equation is

$$\ddot{x} + c\dot{x} + k_2 x = f\sin(\omega t) \qquad \text{x} < 0 \tag{2.79}$$

where,

$$k_1 = 5.96 \times 10^9 \tag{2.80}$$

$$k_2 = 4.24 \times 10^9 \tag{2.81}$$

$$c = 18 \tag{2.82}$$

$$f = 0.001 \tag{2.83}$$

These equations were integrated numerically using the trapezoidal rule in a program written in PASCAL called INTEGRAT.PAS. This program is included in Appendix F. At the zero crossing, the program selects the appropriate equation to integrate. The simulation was run for four time constants (approximately 0.7 seconds) so that an essentially steady condition was reached. The maximum and minimum amplitudes were then recorded, the frequency incremented, and the simulation run again for four time constants. The simulation routine was repeated until the resonant peak of the system was captured. Figure 2.13 is a plot of the frequency response for f=0.001. The frequency range of the sinusoidal driving force is 300 Hz, and 100 integration points were selected. Figure 2.14 is a plot of the frequency response for f=0.01. Both the AC and DC amplitudes are plotted. In both figures, the DC amplitude is in reality negative, but is shown as positive. The procedure for linearizing these two equations with describing functions is virtually identical to the method outlined above. The solution of the linearized equations at resonance is given below.

$$x = \frac{f}{c\omega_n}\left(\cos(\omega_n t) - \frac{2(k_1 - k_2)}{\pi(k_1 + k_2)}\right) \tag{2.84}$$

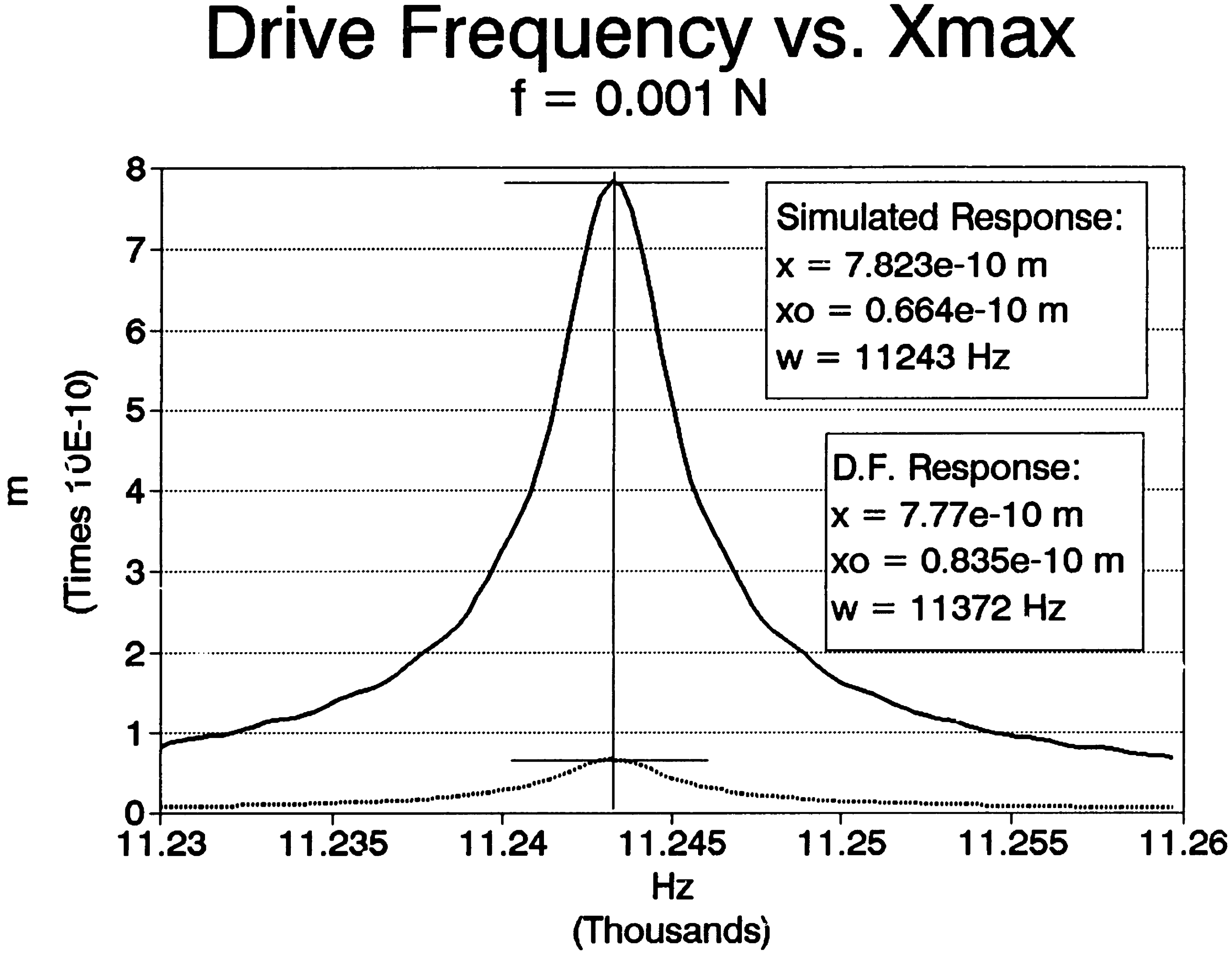


Figure 2.13: Nonlinear Simulation (f=0.001)

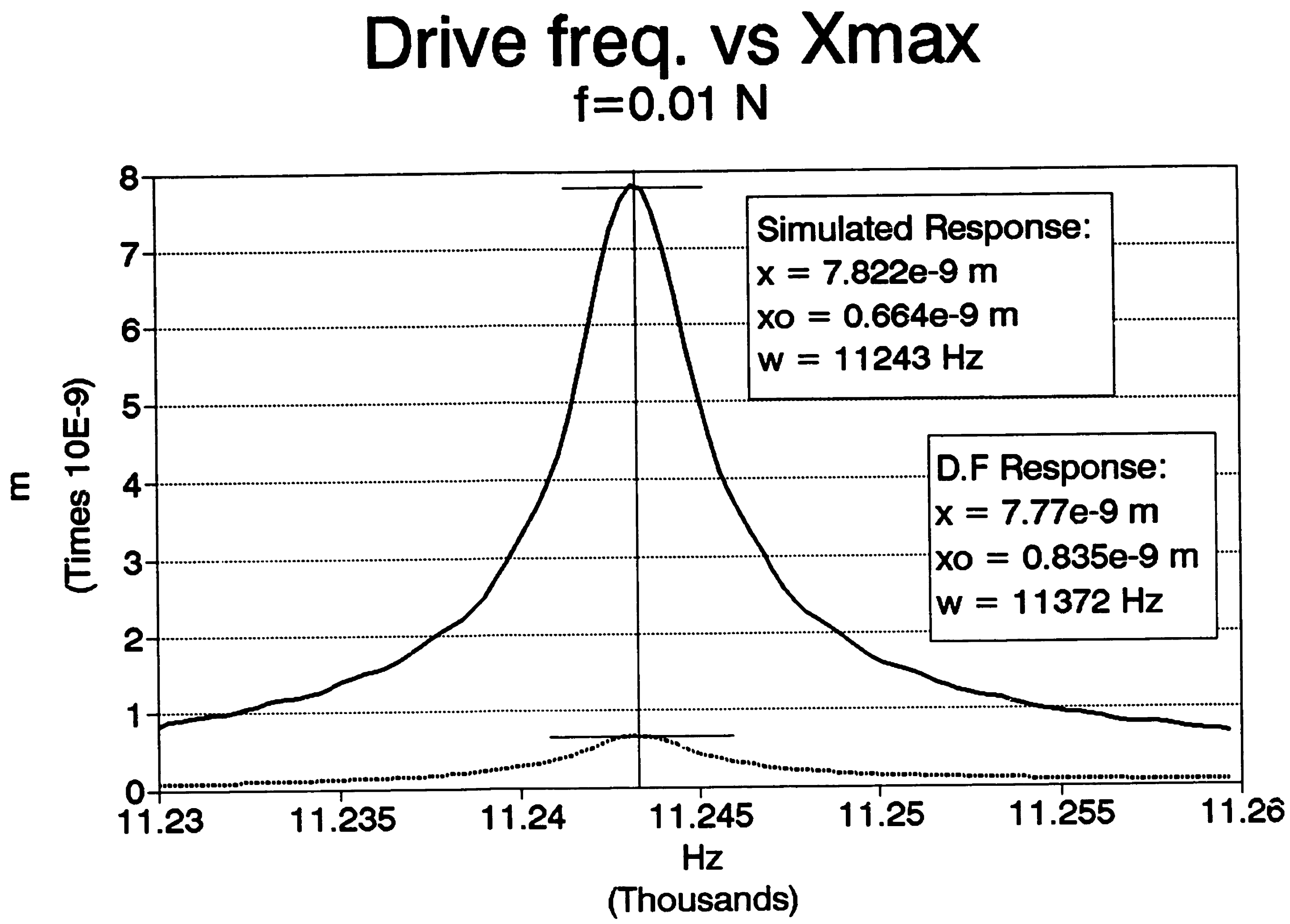


Figure 2.14: Nonlinear Simulation (f=0.01)

	SIDF	Simulation	% Difference
AC amplitude	7.7×10^{-10}	7.82×10^{-10}	1.5
DC amplitude	0.835×10^{-10}	0.662×10^{-10}	21
Drive Frequency	11372 Hz	11243 Hz	1.1

Table 2.2: SIDF - Direct Integration Comparison (f=0.001)

	SIDF	Simulation	% Difference
AC amplitude	7.7×10^{-9}	7.82×10^{-9}	1.5
DC amplitude	0.835×10^{-9}	0.662×10^{-9}	21
Drive Frequency	11372 Hz	11243 Hz	1.1

Table 2.3: SIDF - Direct Integration Comparison (f=0.01)

where,

$$\omega_n = \sqrt{\frac{k_1 + k_2}{2}} \tag{2.85}$$

Table 2.2 compares the results of the simulation to the linearized solution. Table 2.3 compares the results of the simulation to the linearized solution with the forcing function, f, increased an order of magnitude. The nonlinear crack system, as it is modeled, does not create a frequency sensitivity to amplitude. In Tables 2.2 and 2.3, the error in the AC amplitude is small; however, the DC amplitude shows a difference of 20%. The DC term is approximately a factor of ten less than the AC amplitude, and therefore, does not contribute significant error to the total displacement, AC plus DC. In conclusion, even for low values of damping, linearizing with describing functions very accurately models the true nonlinear system.

2.3 Fluid Damping and Inertial Effects

In this section, expressions are derived that model the viscous damping and inertial forces exerted by the fluid on the silicon plate and beam.

As envisioned, the fatigue device will operate in an environmental control chamber allowing for testing in various gaseous environments at atmospheric pressure, or at vacuum. However, at atmospheric pressure, the small size of the device gives Reynolds numbers in the range of viscous fluid flow. Minimizing the viscous damping effects as much as possible is important in this experiment primarily because sufficient sinusoidal displacements must generated that apply tensile stresses to the crack site at values of 50 to 80% of the critical stress intensity factor. Under normal operation, the electrostatic driving forces should create only compressive stesses at the crack site which would make stress corrosion fatigue unlikely.

Low damping is anticipated with the device shown in Figures 2.1 and 2.2 because of the deep well etched beneath the device. The deep well configuration minimizes the viscous and inertial pressure forces exerted by the gas on the plate. Also, holes in the silicon plate are positioned below the bridge electrodes which minimize damping in this region by allowing the gas to flow more freely in and out of the confined area. Fluid inertia alters the resonant frequency of the device to an extent, but will not result in significant frequency variations unless large temperature excursions ($> 10°C$) are encountered that significantly change the density of air. Young's modulus also varies with temperature and will cause a slight frequency shift (≈ 1 Hz/$°C$). To minimize errors cause by temperature fluctuations, the device is mounted on a copper block that is temperature controlled to within $0.5°C$.

In Equation 2.44, the terms Ξ_1, Ξ_2, and Ξ_3 are generalized forces associated with the nonconservative damping forces, the electrostatic driving force, and the inertial forces of the surrounding gas which appear as additional masses added to the total mass of the device. The virtual work of the generalized forces can be expressed as

$$\delta W = \Xi_1 \delta v_1 + \Xi_2 \delta v_2 + \Xi_3 \delta v_3 \tag{2.86}$$

where δv_1, δv_2, and δv_3 are virtual displacements. In Figure 2.3, the damping forces acting on the silicon end plate consist of point loads applied at the center of pressure of the forcing and sensing electrodes, and the bottom side of the end plate. The damping force acting on the beam is a distributed load, and will be dealt with separately. The work of the point loads in Equation 2.86 can be written in terms of the driving electrode force, F_T; the viscous damping force of the driving electrode, F_D^T; the sensing electrode damping force, F_D^S; the end plate damping force, F_D^P; the fluid inertia force at the driving electrode, F_I^T; the sensing electrode inertia force, F_I^S; and the end plate inertia force, F_I^P.

$$\begin{aligned}\delta W_p &= (F_T - F_D^T - F_I^T)\left(\delta v_b + \frac{\partial \delta v_b}{\partial x}\ell_{cd}\right)_{x=\ell_{ac}} \\ &\quad -(F_I^S + F_D^S)\left(\delta v_b + \frac{\partial \delta v_b}{\partial x}\ell ce\right)_{x=\ell_{ac}} \\ &\quad -(F_I^P + F_D^P)\left(\delta v_b + \frac{\partial \delta v_b}{\partial x}\ell cf\right)_{x=\ell_{ac}} \qquad (2.87)\end{aligned}$$

Substituting Equation 2.2 into Equation 2.87 gives the desired generalized forces of point loads.

$$\begin{aligned}\Xi_{1p} &= (F_T - F_D^T - F_I^T)\left(1 + 2\frac{\ell_{cd}}{\ell_{ac}}\right) - (F_I^S + F_D^S)\left(1 + 2\frac{\ell_{ce}}{\ell_{ac}}\right) \\ &\quad -(F_I^P + F_D^P)\left(1 + 2\frac{\ell_{cf}}{\ell_{ac}}\right) \qquad (2.88)\end{aligned}$$

$$\begin{aligned}\Xi_{2p} &= (F_T - F_D^T - F_I^T)\left(1 + 3\frac{\ell_{cd}}{\ell_{ac}}\right) - (F_I^S + F_D^S)\left(1 + 3\frac{\ell_{ce}}{\ell_{ac}}\right) \\ &\quad -(F_I^P + F_D^P)\left(1 + 3\frac{\ell_{cf}}{\ell_{ac}}\right) \qquad (2.89)\end{aligned}$$

$$\begin{aligned}\Xi_{3p} &= (F_T - F_D^T - F_I^T)\left(1 + \frac{\ell_{cd}}{\ell_{bc}}\right) - (F_I^S + F_D^S)\left(1 + \frac{\ell_{ce}}{\ell_{bc}}\right) \\ &\quad -(F_I^P + F_D^P)\left(1 + \frac{\ell_{cf}}{\ell_{bc}}\right) \qquad (2.90)\end{aligned}$$

The following constants are defined.

$$\eta_1 = \left(1 + 2\frac{\ell_{cd}}{\ell_{ac}}\right) \qquad (2.91)$$

$$\eta_2 = \left(1 + 3\frac{\ell_{cd}}{\ell_{ac}}\right) \tag{2.92}$$

$$\eta_3 = \left(1 + \frac{\ell_{cd}}{\ell_{bc}}\right) \tag{2.93}$$

$$\xi_1 = \left(1 + 2\frac{\ell_{ce}}{\ell_{ac}}\right) \tag{2.94}$$

$$\xi_2 = \left(1 + 3\frac{\ell_{ce}}{\ell_{ac}}\right) \tag{2.95}$$

$$\xi_3 = \left(1 + \frac{\ell_{ce}}{\ell_{bc}}\right) \tag{2.96}$$

$$\lambda_1 = \left(1 + 2\frac{\ell_{cf}}{\ell_{ac}}\right) \tag{2.97}$$

$$\lambda_2 = \left(1 + 3\frac{\ell_{cf}}{\ell_{ac}}\right) \tag{2.98}$$

$$\lambda_3 = \left(1 + \frac{\ell_{cf}}{\ell_{bc}}\right) \tag{2.99}$$

The ζ_i's, in equations 2.26 through 2.34, and the above definitions for the η_i's, ξ_i's, and λ_i's are factors that effectively increase the contribution of the conservative and nonconservative forces acting at various points on the structure. These factors are a result of the lever arms created by the point of application of these forces with respect to point C, the beam tip, in Figure 2.3. With the above definitions, the generalized forces of the point loads can be rewritten as follows.

$$\Xi_{1p} = (F_T - F_D^T - F_I^T)\eta_1 - (F_I^S + F_D^S)\xi_1 - (F_I^P + F_D^P)\lambda_1 \tag{2.100}$$

$$\Xi_{2p} = (F_T - F_D^T - F_I^T)\eta_2 - (F_I^S + F_D^S)\xi_2 - (F_I^P + F_D^P)\lambda_2 \tag{2.101}$$

$$\Xi_{3p} = (F_T - F_D^T - F_I^T)\eta_3 - (F_I^S + F_D^S)\xi_3 - (F_I^P + F_D^P)\lambda_3 \tag{2.102}$$

Equations for the damping and inertial forces of the fluid medium will now be developed for the case of one dimensional fully developed viscous incompressible flow between parallel plates. The assumption of incompressibility is verified by computing the Mach number assuming a resonant frequency of 12 KHz and a plate displacement of 10×10^{-6} μm. The Mach number is the ratio of the velocity of the air surrounding the device to sonic velocity. Mach

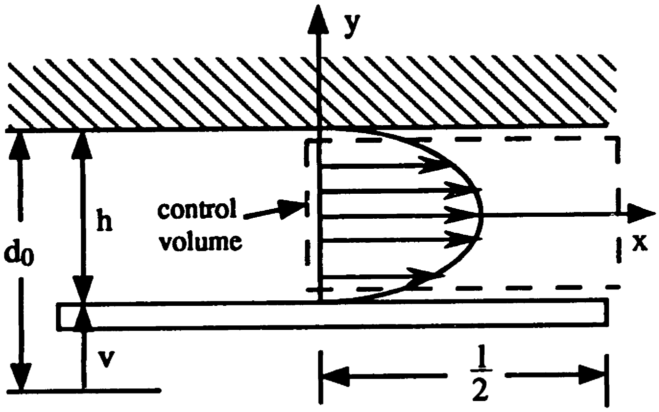


Figure 2.15: Viscous Flow Profile

numbers greater than 0.3 generally indicate that significant compressibility effects are present.

$$M = \frac{V}{V_s} \tag{2.103}$$

where,

$$\begin{aligned} M &= \text{Mach number} && (2.104)\\ V_s &= 347.2\ \text{m/s: sonic velocity at } 80°\ \text{F} && (2.105)\\ V &= \omega v_f\ \text{m/s: plate velocity} && (2.106)\\ \omega &= \ \text{rad/s: resonant frequency} && (2.107)\\ v_f &= \ \text{m/s: velocity of point f in Figure 2.3} && (2.108) \end{aligned}$$

Consequently,

$$M = 2.2 \times 10^{-3} \tag{2.109}$$

Given the low Mach number, compressibility of the air is negligible. Figure 2.15 defines the coordinate system and flow profile. The governing equation for incompressible viscous fluid flow, including the linear inertial component, is as follows.

$$\rho\frac{\partial u}{\partial t} = -\frac{\partial P}{\partial x} + \mu\frac{\partial^2 u}{\partial y^2} \tag{2.110}$$

where,

$$\begin{aligned} u(y,t) &= \text{fluid velocity: } \left(\frac{\mathrm{m}}{\mathrm{s}}\right) \\ \rho &= \text{fluid density: } \left(\frac{\mathrm{kg}}{\mathrm{m}^3}\right) \\ P(x) &= \text{pressure: } \left(\frac{\mathrm{N}}{\mathrm{m}^2}\right) \\ \mu &= \text{viscosity: } \left(\frac{\mathrm{N\ s}}{\mathrm{m}^2}\right) \end{aligned}$$

Taking the Laplace transform of Equation 2.110 with respect to time gives

$$\mu\frac{\partial^2 u}{\partial y^2} - \rho s u = \frac{\partial P}{\partial x} \tag{2.111}$$

Treating s as a constant, solutions for the homogeneous and particular solutions are superposed. The homogenous solution is as follows.

$$u_h(s,y) = c_1 e^{\lambda y} + c_2 e^{-\lambda y} \tag{2.112}$$

where,

$$\begin{aligned} \lambda &= \sqrt{\frac{s\rho}{\mu}} \\ c_1 &= \text{constant: (m)} \\ c_2 &= \text{constant: (m)} \end{aligned}$$

The particular solution is written as follows.

$$u_p(s,x) = \frac{-1}{\rho s}\frac{\partial P}{\partial x}. \tag{2.113}$$

Combining and applying the boundary conditions, $u(s,\frac{\pm h}{2}) = 0$, gives the following equation.

$$u(s,y) = \frac{1}{\rho s}\frac{\partial P}{\partial x}\left[\frac{\cosh \lambda y}{\cosh(\frac{\lambda h}{2})} - 1\right] \tag{2.114}$$

The flow rate is computed by integrating over the cross section of the channel.

$$Q(s) = b\int_{-\frac{h}{2}}^{\frac{h}{2}} u(s,y)\,dy \tag{2.115}$$

The solution for the flow rate is given below, where the plate width is b in meters.

$$Q(s) = \frac{-bh}{\rho s}\frac{\partial P}{\partial x}\left[1 - \frac{\tanh\kappa}{\kappa}\right] \tag{2.116}$$

where,

$$\kappa = \frac{\lambda h}{2} \tag{2.117}$$

Solving for the pressure gradient and integrating over the half length of the plate gives the pressure as a function of x.

$$P(x) = \frac{-s\rho\left(\frac{\ell}{2} - x\right)}{bh\left(1 - \frac{\tanh\kappa}{\kappa}\right)}\,Q(s). \tag{2.118}$$

Next, the pressure force acting on the plate is calculated by integrating over the entire length, ℓ, of the plate.

$$F = 2b\int_0^{\frac{\ell}{2}} P(x)dx. \tag{2.119}$$

Performing the integration gives a nonlinear expression for the force.

$$F = \frac{-s\rho\ell^2}{4h\left(1 - \frac{\tanh\kappa}{\kappa}\right)}\,Q(s) \tag{2.120}$$

The following expression in Equation 2.120 is expanded and approximated.

$$\frac{1}{\left(1 - \frac{\tanh\kappa}{\kappa}\right)} = \frac{\frac{3}{\kappa^2}}{1 - \left(\frac{2\kappa^2}{5} - \frac{17\kappa^4}{105} + \cdots\right)} \tag{2.121}$$

$$\begin{aligned} &= \frac{3}{\kappa^2}\left[1 + \left(\frac{2\kappa^2}{5} - \frac{17\kappa^4}{105} + \cdots\right) + \right. \\ &\qquad \left. \left(\frac{2\kappa^2}{5} - \frac{17\kappa^4}{105} + \cdots\right)^2 + \cdots\right] \qquad (2.122) \\ &\approx \frac{3}{\kappa^2}\left(1 + \frac{2\kappa^2}{5}\right) \qquad (2.123) \end{aligned}$$

Equation 2.123 is substituted into Equation 2.120.

$$F(s) = -\frac{3s\rho\ell^2}{4h\kappa^2}\left(1+\frac{2\kappa^2}{5}\right)Q(s) \tag{2.124}$$

where,

$$\kappa^2 = \frac{h^2 s\rho}{4\mu}$$

To relate the flow rate to the displacement of the plate, v, the conservation of mass is applied to the control volume shown in Figure 2.15.

$$\frac{d}{dt}\int_{c.v.}\rho dV + \int_{c.s.}\rho\vec{\mathbf{V}}\cdot\hat{n}dA = 0 \tag{2.125}$$

Performing the integration and applying the definition for the Laplace transform gives the flow rate as a function of s.

$$Q(s) = -\frac{\ell b}{2}(sh) \tag{2.126}$$

Substituting the above into Equation 2.124 relates the force to the displacement of the plate.

$$F(s) = \frac{3\mu b}{2}\left(\frac{\ell}{h}\right)^3(sh) + \frac{3\rho\ell^3 b}{20h}(s^2h) \tag{2.127}$$

Using the following definitions, Equation 2.127 is transformed back to the time domain.

$$\begin{aligned} sh &= \frac{dh}{dt} && (2.128)\\ s^2h &= \frac{d^2h}{dt^2} && (2.129)\\ h(t) &= d_0 - v(t) && (2.130) \end{aligned}$$

$$F(t) = -\left(\frac{3\mu b\ell^3}{2h^3}\right)\frac{dv}{dt} - \left(\frac{3\rho\ell^3 b}{20h}\right)\frac{d^2v}{dt^2} \tag{2.131}$$

Equation 2.131 has two components: the first is the force due to viscous damping, and the second is an inertial force created by the air mass. The coefficients of the two components can be used to estimate the total damping of the structure and the effective increase in structural mass. Simplified one dimensional flow profiles can be assumed over various sections of the structure and summed together to get the total effect. Although the actual flow profile is multidimensional and quite complex, it is reasonable to assume the dominant damping terms result from squeeze film effects under the driving and sensing electrodes, and the pumping action of the end plate on the air mass entrained in the volume etched beneath the device. The air slots under the sensor electrode are relatively narrow and closely spaced; therefore, it is likely that a nearly unidimensional flow profile exists on the surface of the silicon plate between these slots. Inevitably, some gas will exit in other directions depending on the exact nature of the pressure distribution; however, by constraining the flow to one direction, and by assuming a fully developed flow profile, a higher pressure drop will be calculated resulting in a more conservative damping estimate than would be calculated from the exact solution. Without precise knowledge of the flow distribution about the edges of the structure, or under the bridge electrodes and in the well, conservative damping estimates are preferable.

The damping forces F_D^T, F_D^S, and F_D^P are written as follows.

$$F_D^T = C_T \left(\dot{v}_b + \frac{\partial^2 v_b}{\partial x \partial t} \ell_{cd} \right)_{x=\ell_{ac}} \qquad (2.132)$$

$$F_D^S = C_S \left(\dot{v}_b + \frac{\partial^2 v_b}{\partial x \partial t} \ell_{ce} \right)_{x=\ell_{ac}} \qquad (2.133)$$

$$F_D^P = C_P \left(\dot{v}_b + \frac{\partial^2 v_b}{\partial x \partial t} \ell_{cf} \right)_{x=\ell_{ac}} \qquad (2.134)$$

The inertial forces are calculated as follows.

$$F_I^T = M_T \left(\ddot{v}_b + \frac{\partial^3 v_b}{\partial x \partial t^2} \ell_{cd} \right)_{x=\ell_{ac}} \qquad (2.135)$$

$$F_I^S = M_S \left(\ddot{v}_b + \frac{\partial^3 v_b}{\partial x \partial t^2} \ell_{ce} \right)_{x=\ell_{ac}} \qquad (2.136)$$

$$F_I^P = M_P \left(\ddot{v}_b + \frac{\partial^3 v_b}{\partial x \partial t^2} \ell_{cf} \right)_{x=\ell_{ac}} \qquad (2.137)$$

The damping constants: C_T, C_S, and C_P; and air masses: M_T, M_S, M_P are derived from superposition of one dimensional flow profiles assumed over sections of the end plate. The form is based on the derivation of the damping and mass constants in Equation 2.131. Substituting Equation 2.2 into Equations 2.132 through 2.137 gives the following set of equations.

$$F_D^T = C_T\left(1+2\frac{\ell_{cd}}{\ell_{ac}}\right)\dot{v}_1 + C_T\left(1+3\frac{\ell_{cd}}{\ell_{ac}}\right)\dot{v}_2 + C_T\left(1+\frac{\ell_{cd}}{\ell_{bc}}\right)\dot{v}_3 \quad (2.138)$$

$$F_D^S = C_S\left(1+2\frac{\ell_{ce}}{\ell_{ac}}\right)\dot{v}_1 + C_S\left(1+3\frac{\ell_{ce}}{\ell_{ac}}\right)\dot{v}_2 + C_S\left(1+\frac{\ell_{ce}}{\ell_{bc}}\right)\dot{v}_3 \quad (2.139)$$

$$F_D^P = C_P\left(1+2\frac{\ell_{cf}}{\ell_{ac}}\right)\dot{v}_1 + C_P\left(1+3\frac{\ell_{cf}}{\ell_{ac}}\right)\dot{v}_2 + C_P\left(1+\frac{\ell_{cf}}{\ell_{bc}}\right)\dot{v}_3 \quad (2.140)$$

$$F_I^T = M_T\left(1+2\frac{\ell_{cd}}{\ell_{ac}}\right)\ddot{v}_1 + M_T\left(1+3\frac{\ell_{cd}}{\ell_{ac}}\right)\ddot{v}_2 + M_T\left(1+\frac{\ell_{cd}}{\ell_{bc}}\right)\ddot{v}_3 \quad (2.141)$$

$$F_I^S = M_S\left(1+2\frac{\ell_{ce}}{\ell_{ac}}\right)\ddot{v}_1 + M_S\left(1+3\frac{\ell_{ce}}{\ell_{ac}}\right)\ddot{v}_2 + M_S\left(1+\frac{\ell_{ce}}{\ell_{bc}}\right)\ddot{v}_3 \quad (2.142)$$

$$F_I^P = M_P\left(1+2\frac{\ell_{cf}}{\ell_{ac}}\right)\ddot{v}_1 + M_P\left(1+3\frac{\ell_{cf}}{\ell_{ac}}\right)\ddot{v}_2 + M_P\left(1+\frac{\ell_{cf}}{\ell_{bc}}\right)\ddot{v}_3 \quad (2.143)$$

Using the definintions in Equations 2.91 through 2.99 the equations for the damping and inertial forces can be abbreviated as follows.

$$\begin{aligned}
F_D^T &= C_T\eta_1\dot{v}_1 + C_T\eta_2\dot{v}_2 + C_T\eta_3\dot{v}_3 && (2.144)\\
F_D^S &= C_S\xi_1\dot{v}_1 + C_S\xi_2\dot{v}_2 + C_S\xi_3\dot{v}_3 && (2.145)\\
F_D^P &= C_P\lambda_1\dot{v}_1 + C_P\lambda_2 v_2 + C_P\xi_3\dot{v}_3 && (2.146)\\
F_I^T &= M_T\eta_1\ddot{v}_1 + M_T\eta_2\ddot{v}_2 + M_T\eta_3\ddot{v}_3 && (2.147)\\
F_I^S &= M_S\xi_1\ddot{v}_1 + M_S\xi_2\ddot{v}_2 + M_S\xi_3\ddot{v}_3 && (2.148)\\
F_I^P &= M_P\lambda_1\ddot{v}_1 + M_P\lambda_2\ddot{v}_2 + M_P\lambda_3\ddot{v}_3 && (2.149)
\end{aligned}$$

Now that expressions for the damping and inertial forces of the fluid acting on the end plate have been derived, the distributed damping of the beam will be derived. The virtual work of the distribute damping is given as follows.

$$\delta W_d = C_B\left(\int_0^{\ell_{ab}} \frac{\dot{v}_a(x,t)\delta v_a(x,t)}{\ell_{ab}}\,dx + \int_{\ell_{ab}}^{\ell_{ac}} \frac{\dot{v}_b(x,t)\delta v_b(x,t)}{\ell_{bc}}\,dx\right) \quad (2.150)$$

where,

$$C_B = 8\mu b \tag{2.151}$$

The damping coefficient, C_B, is for viscous flow around a thin flat disk of diameter b. The disk is oriented such that the flat side is perpendicular to the flow [18]. Although the beam is a rectangle, the disk approximatation should be close. In any case, beam damping is expected to be small. For $\ell_{ac} \gg \ell_{ab}$, the first integral in Equation 2.150 is small and can be neglected. The integration can then be carried out from 0 to ℓ_{ac}. Using Equation 2.2, the following approximation for the virtual work of the distributed damping of the beam is obtained.

$$\begin{aligned}\delta W_d &= \frac{C_B}{\ell_{ac}} \int_0^{\ell_{ac}} \left(\dot{v}_1 \left(\frac{x}{\ell_{ac}}\right)^2 + \dot{v}_2 \left(\frac{x}{\ell_{ac}}\right)^3 + \dot{v}_3 \left(\frac{x}{\ell_{ac}}\right)\right) \\ &\quad \left(\delta v_1 \left(\frac{x}{\ell_{ac}}\right)^2 + \delta v_2 \left(\frac{x}{\ell_{ac}}\right)^3 + \delta v_3 \left(\frac{x}{\ell_{ac}}\right)\right) dx \end{aligned} \tag{2.152}$$

Performing the integration and simplifying gives the following set of generalized forces for the damping of the beam.

$$\Xi_{1d} = C_B \left(\frac{\dot{v}_1}{5} + \frac{\dot{v}_2}{6} + \frac{\dot{v}_3}{4}\right) \tag{2.153}$$

$$\Xi_{2d} = C_B \left(\frac{\dot{v}_1}{6} + \frac{\dot{v}_2}{7} + \frac{\dot{v}_3}{5}\right) \tag{2.154}$$

$$\Xi_{3d} = C_B \left(\frac{\dot{v}_1}{4} + \frac{\dot{v}_2}{5} + \frac{\dot{v}_3}{3}\right) \tag{2.155}$$

Equations 2.100 through 2.102 and Equations 2.153 through 2.155 are combined to give the total magnitude of the generalized forces. Using the expressions derived for the point forces, the damping matrix and an additional mass matrix are obtained.

$$\underline{\mathbf{R}} = \underline{\mathbf{P}} - \underline{\mathbf{C}}\,\dot{\underline{\mathbf{U}}} - \delta\underline{\mathbf{M}}\,\ddot{\underline{\mathbf{U}}} \tag{2.156}$$

where,

$$\underline{\mathbf{R}} = \left\{\begin{matrix} \Xi_1 \\ \Xi_2 \\ \Xi_3 \end{matrix}\right\} \tag{2.157}$$

$$\underline{\mathbf{P}} = \left\{\begin{matrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{matrix}\right\} F_T \tag{2.158}$$

$$
\begin{aligned}
\underline{\mathbf{C}} &= \begin{bmatrix} \eta_1^2 & \eta_1\eta_2 & \eta_1\eta_3 \\ \eta_1\eta_2 & \eta_2^2 & \eta_2\eta_3 \\ \eta_1\eta_3 & \eta_2\eta_3 & \eta_3^2 \end{bmatrix} C_T + \begin{bmatrix} \xi_1^2 & \xi_1\xi_2 & \xi_1\xi_3 \\ \xi_1\xi_2 & \xi_2^2 & \xi_2\xi_3 \\ \xi_1\xi_3 & \xi_2\xi_3 & \xi_3^2 \end{bmatrix} C_S \\
&+ \begin{bmatrix} \lambda_1^2 & \lambda_1\lambda_2 & \lambda_1\lambda_3 \\ \lambda_1\lambda_2 & \lambda_2^2 & \lambda_2\lambda_3 \\ \lambda_1\lambda_3 & \lambda_2\lambda_3 & \lambda_3^2 \end{bmatrix} C_P + \begin{bmatrix} \frac{1}{5} & \frac{1}{6} & \frac{1}{4} \\ \frac{1}{6} & \frac{1}{7} & \frac{1}{5} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{3} \end{bmatrix} C_B \qquad (2.159) \\
\delta\underline{\mathbf{M}} &= \begin{bmatrix} \eta_1^2 & \eta_1\eta_2 & \eta_1\eta_3 \\ \eta_1\eta_2 & \eta_2^2 & \eta_2\eta_3 \\ \eta_1\eta_3 & \eta_2\eta_3 & \eta_3^2 \end{bmatrix} M_T + \begin{bmatrix} \xi_1^2 & \xi_1\xi_2 & \xi_1\xi_3 \\ \xi_1\xi_2 & \xi_2^2 & \xi_2\xi_3 \\ \xi_1\xi_3 & \xi_2\xi_3 & \xi_3^2 \end{bmatrix} M_S \\
&+ \begin{bmatrix} \lambda_1^2 & \lambda_1\lambda_2 & \lambda_1\lambda_3 \\ \lambda_1\lambda_2 & \lambda_2^2 & \lambda_2\lambda_3 \\ \lambda_1\lambda_3 & \lambda_2\lambda_3 & \lambda_3^2 \end{bmatrix} M_P \qquad (2.160)
\end{aligned}
$$

The equation of motion can now be written as

$$\underline{\acute{\mathbf{M}}}\,\underline{\ddot{\mathbf{U}}} + \underline{\mathbf{C}}\,\underline{\dot{\mathbf{U}}} + \underline{\mathbf{K}}\,\underline{\mathbf{U}} = \underline{\mathbf{P}} \qquad (2.161)$$

where,

$$\underline{\acute{\mathbf{M}}} = \underline{\mathbf{M}} + \delta\underline{\mathbf{M}} \qquad (2.162)$$

In the next section, the electrostatic force will be linearized, and a corresponding stiffness matrix correction factor derived that accounts for the effects of a changing air gap on the electrostatic force.

2.4 Stiffness Matrix Correction

In the preceding sections, the equations of motion for the cantilever system were derived; however, an additional stiffness correction matrix arises that is subtracted from the stiffness matrix of the system. This matrix is a consequence of the inverse dependence of electrostatic driving force on the capacitance air gap, d_0. As the cantilever approaches the driving electrode on its upward travel, the magnitude of the electrostatic force increases displacing the cantilever further than it would travel otherwise (a corresponding decrease also occurs when the cantilever moves downward). The net effect is that of a negative spring. Equation 2.163 models the electrostatic force.

$$F_T = \frac{\epsilon_0 A_T}{2d(t)^2} V_T^2 \tag{2.163}$$

where,

$$\begin{aligned} \epsilon_0 &= \text{permittivity of a vacuum: } \left(\frac{\text{coul}^2}{\text{N m}^2}\right) \\ d(t) &= \text{variable air gap: (m)} \\ V_T &= \text{driving voltage: (volts)} \\ A_T &= \text{torquer electrode area: (m}^2\text{)} \end{aligned}$$

Linearizing with respect to the fixed air gap dimension, d_0, gives

$$F_T(V_T, d) = F_T(V_T, d_0) + \left(\frac{\partial F_T}{\partial d}\right)_{d=d_0} (d - d_0) \tag{2.164}$$

$$= \frac{\epsilon_0 A_T V_T^2}{2d_0^2} - \frac{\epsilon_0 A_T V_T^2}{d_0^3}(d - d_0) \tag{2.165}$$

By definition

$$d_0 - d(t) = v_b(\ell_{ac}) + \left(\frac{\partial v_b}{\partial x}\right)_{x=\ell_{ac}} \ell_{cd} \tag{2.166}$$

$$d_0 - d(t) = \eta_1 v_1 + \eta_2 v_2 + \eta_3 v_3 \tag{2.167}$$

Defining

$$V_T = V_{T0} \sin(\omega t) \tag{2.168}$$

$$
\begin{aligned}
F_0 &= \frac{\epsilon_0 A_T V_{T0}^2}{4d_0^2} && (2.169)\\
\delta k &= \frac{\epsilon_0 A_T V_{T0}^2}{2d_0^3} && (2.170)
\end{aligned}
$$

and substituting Equations 2.167, 2.168, 2.169, and 2.170 into Equation 2.165 gives

$$F_T = [F_0 + \delta k(\eta_1 v_1 + \eta_2 v_2 + \eta_3 v_3)](1 - \cos 2\omega t) \qquad (2.171)$$

Refering back to Equation 2.156, the forcing vector, $\underline{\mathbf{P}}$, can be expanded as follows.

$$\underline{\mathbf{P}} = \underline{\mathbf{P}}_0 + \delta\underline{\mathbf{K}}(t)\,\underline{\mathbf{U}} \qquad (2.172)$$

where,

$$
\begin{aligned}
\underline{\mathbf{P}}_0 &= \left\{ \begin{array}{c} \eta_1 \\ \eta_2 \\ \eta_3 \end{array} \right\} F_0(1 - \cos 2\omega t) && (2.173)\\
\delta\underline{\mathbf{K}}(t) &= \delta\underline{\mathbf{K}}(1 - \cos 2\omega t) && (2.174)
\end{aligned}
$$

The matrix correction factor is defined as follows.

$$\delta\underline{\mathbf{K}} = \begin{bmatrix} \eta_1^2 & \eta_1\eta_2 & \eta_1\eta_3 \\ \eta_1\eta_2 & \eta_2^2 & \eta_2\eta_3 \\ \eta_1\eta_3 & \eta_2\eta_3 & \eta_3^2 \end{bmatrix} \delta k \qquad (2.175)$$

$$\delta\underline{\mathbf{K}} = \underline{\eta}\,\delta k \qquad (2.176)$$

The matrix equations of motion can be rewritten as follows.

$$\acute{\underline{\mathbf{M}}}\,\ddot{\underline{\mathbf{U}}} + \underline{\mathbf{C}}\,\dot{\underline{\mathbf{U}}} + [\underline{\mathbf{K}} - \delta\underline{\mathbf{K}}(t)]\,\underline{\mathbf{U}} = \underline{\mathbf{P}}_0 \qquad (2.177)$$

Defining

$$\acute{\underline{\mathbf{K}}} = \underline{\mathbf{K}} - \delta\underline{\mathbf{K}} \qquad (2.178)$$

The equation of motion is again rewritten with the corrected stiffness matrix.

$$\acute{\underline{\mathbf{M}}}\,\ddot{\underline{\mathbf{U}}} + \underline{\mathbf{C}}\,\dot{\underline{\mathbf{U}}} + \acute{\underline{\mathbf{K}}}\,\underline{\mathbf{U}} = \underline{\mathbf{P}}_0 \qquad (2.179)$$

Without damping, Equation 2.177 has the form of the Mathieu equation where the stiffness matrix is time varying. Systems governed by Mathieu equations have regions of stability and instability that depend on the magnitude of the time varying term. The system will be stable, provided the eigenvalues of the free vibration problem are positive and real; therefore, a stability boundary can be determined based on the maximum value of $\delta\underline{\mathbf{K}}(t)$. The term $(1 - \cos 2\omega t)$ determines the magnitude of $\delta\underline{\mathbf{K}}(t)$ by sinusoidaly varying between 0 and 2 with an average value of 1. Using the maximum value, the stability condition is stated as follows: $|\underline{\mathbf{K}} - 2\delta\underline{\mathbf{K}}| > 0$, or, the system is guaranteed to be stable for all time if $\underline{\mathbf{K}} - 2\delta\underline{\mathbf{K}}$ is positive definite. For the uncracked beam, Equation 2.179 is 2×2, and the determinant is easily expanded algerbraically giving the stability criteria directly as follows.

$$\delta k < \frac{k_{11}k_{22} - k_{12}^2}{2(\eta_2^2 k_{11} + \eta_1^2 k_{22} - 2k_{12}\eta_1\eta_2)} \tag{2.180}$$

However, for the cracked beam Equation 2.179 is 3×3 and expanding the determinant is difficult. An alternative is to recast the criteria as an eigenvalue problem and solve for the stability condition numerically. The eigenvalue problem is stated and solved as follows.

$$\underline{\mathbf{K}}\,\underline{\phi} = 2\delta k \underline{\eta}\,\underline{\phi} \tag{2.181}$$

Defining,

$$\lambda = 2\delta k \tag{2.182}$$

and

$$\underline{\eta} = \underline{\mathbf{S}}\,\underline{\mathbf{S}}^T \tag{2.183}$$

the eigenvalue problem is rewritten as

$$\underline{\mathbf{K}}\,\underline{\phi} = \lambda\underline{\mathbf{S}}\,\underline{\mathbf{S}}^T\underline{\phi} \tag{2.184}$$

Defining

$$\underline{\psi} = \underline{\mathbf{S}}^T\underline{\phi} \tag{2.185}$$

$$\underline{\mathbf{S}}^{-1}\underline{\mathbf{K}}(\underline{\mathbf{S}}^T)^{-1}\underline{\psi} = \lambda\underline{\psi} \tag{2.186}$$

Defining

$$\tilde{\underline{\mathbf{K}}} = \underline{\mathbf{S}}^{-1}\underline{\mathbf{K}}(\underline{\mathbf{S}}^T)^{-1} \tag{2.187}$$

Thus, the eigenvalue problem is rewitten in standard form as follows.

$$\tilde{\underline{\mathbf{K}}}\,\underline{\psi} = \lambda\underline{\psi} \tag{2.188}$$

DOF	Model Type	Frequency (Hz)
3	Lagrangian	$\omega_1 = 2546$ $\omega_2 = 95.6 \times 10^3$ $\omega_3 = 1.9 \times 10^6$

Table 2.4: T38 Modal Frequencies: $2\mu m$ Crack Length

$$(\tilde{\underline{\mathbf{K}}} - \lambda\mathbf{I})\underline{\psi} = 0 \tag{2.189}$$

Taking the smallest eigenvalue, λ_1, stability is ensured if the following is satisfied.

$$\delta k \leq \frac{\lambda_1}{2} \tag{2.190}$$

In general, the elements of $\delta\underline{\mathbf{K}}$ are much smaller than the corresponding terms in $\underline{\mathbf{K}}$; therefore, stability is not expected to be an issue. The frequency, however, will change a small amount, but is accounted for by the stiffness matrix correction factor $\delta\underline{\mathbf{K}}$ in Equation 2.179.

Equation 2.179 will be decomposed into modal form and solved for 3 × 3 cracked beam case. The structure will be driven at its first mode resonance. Because of the relatively large spread in second and third mode frequencies, as shown in Table 2.4, these higher modes will have negligible contribution to the total displacement; therefore, the first mode solution will be used as a baseline model for design and analysis. The first mode frequency in Table 2.4 is less than that shown in Table 2.1 in Section 2.2.3 because the additional rotational degree of freedom associated with the pre-crack is included. The values given in Table 2.4 were obtained by assuming the pre-crack extended $2\mu m$ downward into the beam $10\mu m$ from the fixed end. Proceeding as before, the eigenvalue problem is solved giving the frequencies and mode shapes. The modal matrix diagonalizes the mass and stiffness matrix, but not the damping matrix. To obtain decoupled equations, the off-diagonal damping terms are neglected.

$$\hat{\underline{\mathbf{M}}}\,\ddot{\underline{\mathbf{X}}} + \hat{\underline{\mathbf{C}}}\,\dot{\underline{\mathbf{X}}} + \hat{\underline{\mathbf{K}}}\,\underline{\mathbf{X}} = \hat{\underline{\mathbf{P}}}_0 \tag{2.191}$$

Equation 2.191 is a decoupled set of modal equations where,

$$\hat{\underline{\mathbf{M}}} = \underline{\mathbf{\Phi}}^T\,\acute{\underline{\mathbf{M}}}\,\underline{\mathbf{\Phi}} \tag{2.192}$$

$$\hat{\underline{\mathbf{C}}} = \underline{\mathbf{\Phi}}^T\,\underline{\mathbf{C}}\,\underline{\mathbf{\Phi}} \quad (c_{ij} = 0 \text{ for } i \neq j) \tag{2.193}$$

$$\begin{aligned}
\hat{\underline{\mathbf{K}}} &= \underline{\mathbf{\Phi}}^T\,\acute{\underline{\mathbf{K}}}\,\underline{\mathbf{\Phi}} && (2.194)\\
\underline{\mathbf{U}} &= \underline{\mathbf{\Phi}}\,\underline{\mathbf{X}} && (2.195)\\
\hat{\underline{\mathbf{P}}}_0 &= \underline{\mathbf{\Phi}}^T\,\underline{\mathbf{P}}_0 && (2.196)\\
\underline{\mathbf{\Phi}} &= [\underline{\phi}_1\,\underline{\phi}_2\,\underline{\phi}_3] && (2.197)
\end{aligned}$$

The eigenvectors are defined as follows.

$$\underline{\phi}_1 = \begin{pmatrix} \alpha_1 \\ \beta_1 \\ \gamma_1 \end{pmatrix} \qquad (2.198)$$

$$\underline{\phi}_2 = \begin{pmatrix} \alpha_2 \\ \beta_2 \\ \gamma_2 \end{pmatrix} \qquad (2.199)$$

$$\underline{\phi}_3 = \begin{pmatrix} \alpha_3 \\ \beta_3 \\ \gamma_3 \end{pmatrix} \qquad (2.200)$$

The forced solution of Equation 2.191 can be transformed back to the original generalized coordinates expressed as a superposition of the modal coordinates.

$$\underline{\mathbf{U}} = \sum_{i=1}^{3} \underline{\phi}_i\, x_i(t) \approx \underline{\phi}_1\, x_1(t) \qquad (2.201)$$

The first mode equation is given by the first row and column of Equation 2.191 as follows.

$$\hat{m}_{11}\ddot{x} + \hat{c}_{11}\dot{x} + \hat{k}_{11} = p_1(t) \qquad (2.202)$$

where,

$$\begin{aligned}
p_1(t) &= \underline{\phi}_1^T\,\underline{\mathbf{P}}_0 && (2.203)\\
&= (\alpha_1\eta_1 + \beta_1\eta_2 + \gamma_1\eta_3)(1 - \cos 2\omega t)F_0 && (2.204)
\end{aligned}$$

Neglecting the static response, and dropping the modal notation, the forced solution is

$$x(t) = \frac{-p_0\cos(2\omega t - \phi)}{4m\left[\left(\left(\frac{\omega_n}{2}\right)^2 - \omega^2\right)^2 + \left(\frac{c\omega}{2m}\right)^2\right]^{\frac{1}{2}}} \qquad (2.205)$$

$$\phi = \arctan\left(\frac{\frac{c\omega}{2m}}{\left(\frac{\omega_n}{2}\right)^2 - \omega^2}\right) \tag{2.206}$$

where,

$$p_0 = (\alpha_1\eta_1 + \beta_1\eta_2 + \gamma_1\eta_3)F_0 \tag{2.207}$$

Driving the system at half the natural frequency, $\omega = \frac{\omega_n}{2}$, resonates the system giving the following response.

$$x(t) = \frac{-p_0 \sin(\omega_n t)}{c\omega_n} \tag{2.208}$$

Therefore, the system response is as follows.

$$\underline{\mathbf{U}} = \begin{pmatrix} \alpha_1 \\ \beta_1 \\ \gamma_1 \end{pmatrix} x(t) \tag{2.209}$$

or,

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \beta_1 \\ \gamma_1 \end{pmatrix} x(t) \tag{2.210}$$

Using Equations 2.2 and 2.1, the displacement at any point in the beam can be computed. For $0 \leq x \leq \ell_{ab}$

$$v_a(x,t) = \left[\alpha_1\left(\frac{x}{\ell_{ac}}\right)^2 + \beta_1\left(\frac{x}{\ell_{ac}}\right)^3\right] x(t) \tag{2.211}$$

and for $\ell_{ab} \leq x \leq \ell_{ac}$

$$v_b(x,t) = \left[\alpha_1\left(\frac{x}{\ell_{ac}}\right)^2 + \beta_1\left(\frac{x}{\ell_{ac}}\right)^3 + \gamma_1\left(\frac{x - \ell_{ab}}{\ell_{bc}}\right)\right] x(t) \tag{2.212}$$

Chapter 3

Control System and Electronics

3.1 Frequency Control System

In the previous sections, a dynamic model of the device was developed using a three degree of freedom approximation and Lagrange's equations. Because the modal frequencies of the model are well separated in the frequency spectrum, the first mode solution accurately approximates the motion of the beam at or near the first mode; therefore, a reduced order single degree of freedom model can be used for control system design and analysis.

The method of fatigue testing proposed in this experiment is based on continuous measurements of the natural frequency of the device as a crack propagates through the beam from an initiation site located near the base of the cantilever. An extension of the crack from the initiation site reduces the total stiffness of the structure and causes a downward shift in the natural frequency. To continuously maintain resonance, a control system is necessary that will automatically lock on the natural frequency and track it over the range of anticipated frequency shifts and crack tip velocities. The control system employed is a variation of a phase locked loop where a phase detector measures the difference in phase between the driving voltage applied to the forcing electrode and the output voltage of the sensor. The measured phase shift is compared to a reference value that corresponds to resonance and an appropriate command signal issued to a voltage controlled oscillator (VCO) whenever the measured phase shift deviates from this reference. The VCO is directly connected to the forcing electrode of the device, and issues a constant amplitude voltage at a frequency that is linearly proportional to an input DC command voltage. In Section 2.4, it was shown that the force applied to the end plate is proportional to the square of the driving voltage. Squaring the driving voltage doubles the frequency of the force, shifts the phase 90°, and introduces a minus sign in the magnitude. At resonance, another phase shift of 90° is introduced; therefore, the driving voltage is $V_{T0} \sin\left(\frac{\omega_n}{2} t\right)$, and the response is $-v_{s0} \sin(\omega_n t)$ where v_{s0} is the magnitude of the displacement of the end plate at the sensor electrode. For the moment the minus sign will be ignored. Assuming the response frequency is divided by two, the phase difference between input and output voltages of the device is 0°. This value of 0° is the reference phase the control system regulates.

At resonance, the displacement of the beam depends on the magnitude of the driving voltage, damping, and the natural frequency; therefore, changes in the natural frequency will cause the displacement to shift also. Ideally, the displacement should be held constant; however, in this experiment the frequency shift is anticipated to be small ($\approx$ 500 Hz) compared to the resonant frequency of the device ($\approx$ 5 KHz) so that displacement changes will be negligible; therefore, displacement control is not considered necessary. It should be noted that critical crack growth corresponding to brittle fracture occurs at the elastic wave velocity of the material. For pure silicon, this velocity is 8.6×10^3 m/s at room temperature. Under these circumstances, it is not possible for the control system to dynamically unload the beam quickly enough to arrest the crack. Therefore, the absence of dynamic fatigue or static fatigue in the test structure would be indicated by an instantaneous drop in frequency with no previous detectable shift.

The first step in the analysis is to determine the transfer function of the device that relates phase changes to perturbations in input frequency. In general, the relationship between phase and frequency for a single degree of freedom spring mass damper system in steady state harmonic motion is nonlinear involving an inverse tangent. Therefore, the system will be linearized about its natural frequency, and a linear approximation determined for small phase changes.

The transfer function for the system is as follows.

$$H(s) = \frac{1}{m(s-P)(s-P^*)} \tag{3.1}$$

$$= \frac{1}{m(s^2+2\zeta\omega_n s+\omega_n^2)} \tag{3.2}$$

with the poles at

$$P = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} \tag{3.3}$$

For $\zeta \ll 1$ the pole locations are approximated as follows.

$$P \cong -\zeta\omega_n \pm j\omega_n \tag{3.4}$$

Graphically, Figure 3.1 shows the pole locations in vector form. Vectorally, the phase angle is expressed as follows.

$$\angle H(s) = -\angle(\vec{s}-\vec{P}) - \angle(\vec{s}-\vec{P}^*) \tag{3.5}$$

where P^* is the complex conjugate of P. Assuming the device has small transient responses about the natural frequency, the vector $\vec{s}$ can be represented in the s-plane as a summation of a small transient perturbation, Δs, away from its steady state resonant frequency, ω_n.

$$\vec{s} = \Delta\vec{s} + \vec{j}\omega_n \tag{3.6}$$

Defining

$$\Phi(s) = \angle H(s) \tag{3.7}$$

$$\phi(s) = \angle(\vec{s}-\vec{P}) \tag{3.8}$$

$$\phi^*(s) = \angle(\vec{s}-\vec{P}^*) \tag{3.9}$$

The phase of the system is as follows.

$$\Phi(s) = -\phi(s) - \phi^*(s) \tag{3.10}$$

For $\zeta \ll 1$ and $s \approx j\omega_n$, the phase angle of $\phi^*(s)$ is approximately 90° and nearly constant.

$$\phi^*(s) \approx \frac{\pi}{2} \tag{3.11}$$

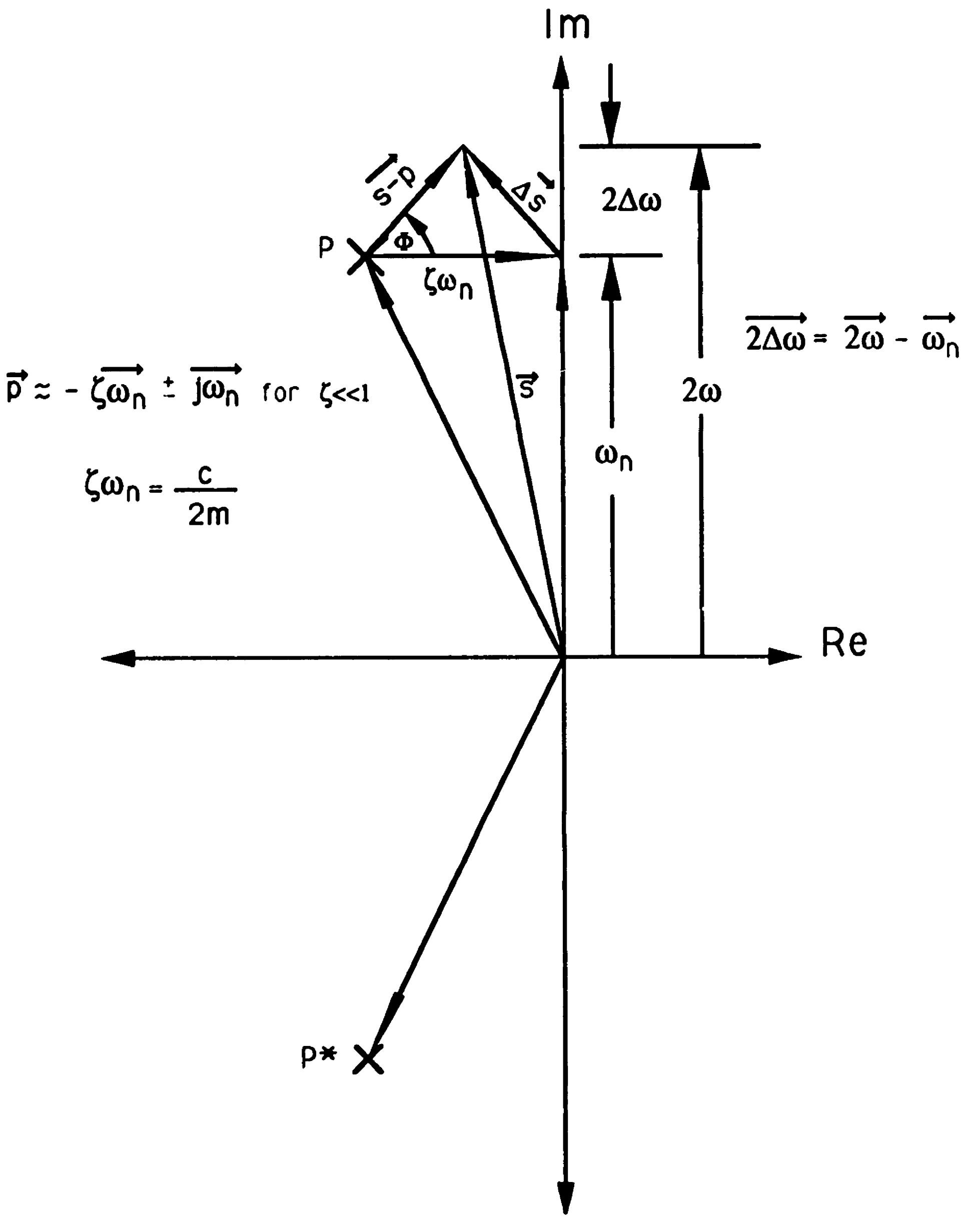

Im
Re
$\overrightarrow{s-p}$
$\overrightarrow{\Delta s}$
$2\Delta\omega$
p
Φ
$\zeta\omega_n$
$\overrightarrow{2\Delta\omega} = \overrightarrow{2\omega} - \overrightarrow{\omega_n}$
$\vec{p} \approx -\overrightarrow{\zeta\omega_n} \pm \overrightarrow{j\omega_n}$ for $\zeta \ll 1$
$\vec{s}$
2ω
ω_n
$\zeta\omega_n = \frac{c}{2m}$
p*

Figure 3.1: Vector Relationships in S-Plane

Dropping the vector notation, $\phi(s)$ is determined as follows.

$$\begin{aligned}
\sin\phi(s) &= \frac{\mathrm{Im}(s-P)}{s-P} && (3.12)\\
\mathrm{Im}(s-P) &= 2\Delta\omega && (3.13)\\
s-P &= \Delta s+\zeta\omega_n && (3.14)
\end{aligned}$$

For $\phi(s) \ll 1$, $\sin\phi(s) \approx \phi(s)$. Combining the above equations gives the following expression.

$$\phi(s)=\frac{2\Delta\omega}{\Delta s+\zeta\omega_n} \qquad (3.15)$$

Note that in Figure 3.1 the output frequency of the VCO, ω, is doubled because, as previously explained, the capacitive force is proportional to the VCO voltage squared which doubles the forcing frequency. The total linearized phase of the system is given below.

$$\Phi(s)=-\frac{2\Delta\omega}{\Delta s+\zeta\omega_n}-\frac{\pi}{2} \qquad (3.16)$$

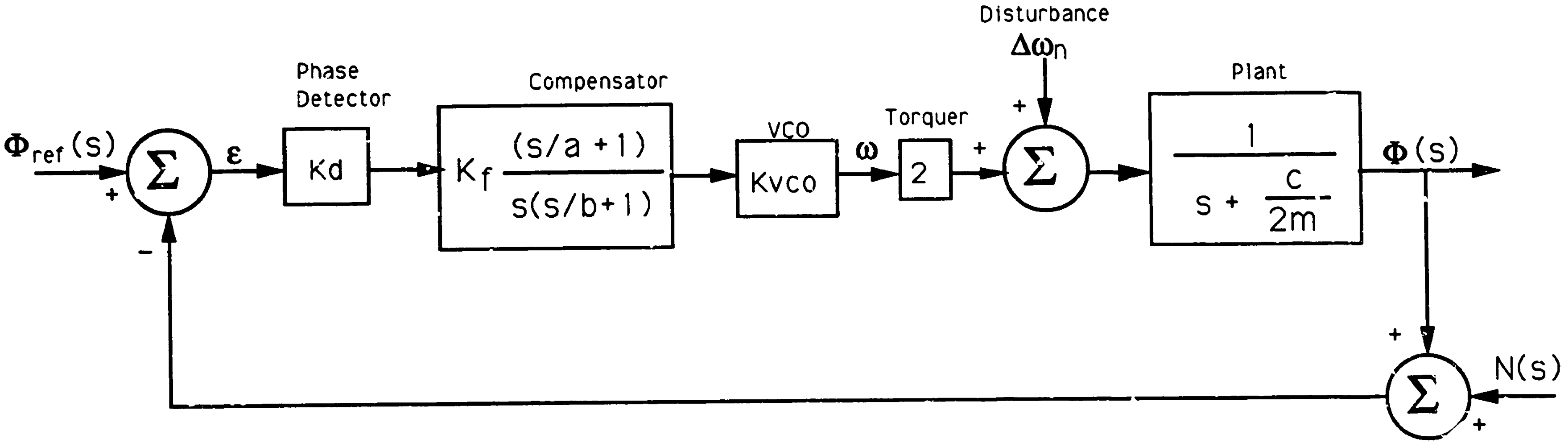
The phase deviation from resonance is defined as follows.

$$\Delta\Phi(s)=\Phi(s)+\frac{\pi}{2} \qquad (3.17)$$

It follows that Equation 3.18 is a linear first order transfer function that models phase deviations of the device from resonance. As the input frequency is increased, the phase decreases.

$$\Delta\Phi(s)=-\frac{2\Delta\omega}{\Delta s+\zeta\omega_n} \qquad (3.18)$$

Figure 3.2 shows the block diagram of the control system. For clarity, the Δ's on the phase angles have been dropped; thus, it is implied that the phase of the system to be controlled is a deviation from resonance. The minus sign on the plant transfer function has been transferred to the summing junction of the feedback loop and reference phase. Therefore, contrary to most servo controllers, this system is implemented with positive feedback because the sign inversion already occurs at the plant. The reference phase, Φ_{ref}, is always maintained at zero regulating the output to zero. The compensator



$K_d = \frac{5}{2\pi}$	Phase detector gain (volts/rad)
$K(s) = K_f \frac{(\frac{s}{a}+1)}{s(\frac{s}{b}+1)}$	Type I lead compensator
$K_{VCO} = 2000\pi$	VCO gain (rad/s/volt)
$G(s) = \frac{1}{s+\frac{c}{2m}}$	Plant transfer function (s)
$\Phi_{ref}(s)$	Reference phase (rad)
$\Phi(s)$	Measured phase (rad)
$N(s)$	Sensor noise
ω_n	Natural frequency (rad/s)
$\Delta\omega_n$	Change in natural frequency (rad/s)
m	Mass (kg)
c	Damping constant (n/rad/s)

Figure 3.2: Control System Block Diagram

is type I lead which minimizes steady state errors and response time. The constants a and b in the compensator are computed to give a nominal closed loop bandwidth of 15 Hz, which gives good tracking over a sufficiently large frequency range providing ample data points. A high bandwidth controller would tend to introduce jitter in the output frequency of the VCO sacrificing measurement accuracy. The disturbance, $\Delta\omega_n$, is a change in natural frequency induced by fatigue cracking in the cantilever. The anticipated rate of crack propagation is modeled by a power law relation related to the stress intensity factor at the crack site. Such a disturbance, can be approximated as a ramp for subcritical cracking. Near fracture, however, the crack velocity will begin to change rapidly, resulting in a significant phase error. Initially, such a phase error will cause a decrease in the amplitude of motion of the cantilever tending to arrest the crack. At very large phase errors ($> 30°$), enough nonlinearities are introduced in the system to cause the device to fall off resonance entirely. The system may reacquire lock, but failure would be imminent.

3.2 Sensing and Control Electronics

In section 3.1, a linearized plant transfer function was derived, and a frequency control system was presented which regulates phase perturbations about resonance to zero. Implementing the control system requires circuitry that will sense the motion of the cantilever, determine whether the device is on resonance, then correct the driving frequency accordingly. Figure 3.3, conceptualizes the sensing and torquing scheme. Sensing is accomplished by applying a known high frequency low amplitude carrier signal, V_c (5 volts at 300 KHz), to the p type beam. The carrier signal is biased a negative amount, V_b (-5 volts), so that the p-n junction is always reversed biased when the n type substrate is at ground potential. The reverse biasing prevents current from flowing from the beam into the substrate short circuiting the carrier signal. As the beam vibrates, the air gap of the capacitance bridge changes, thereby modulating the amplitude of the carrier signal. This signal is then amplified, demodulated, and filtered in the sensing circuit to give a voltage proportional to the amplitude of vibration of the beam at the frequency of the device. The control circuit compares the phase of the signal exiting the sensing circuit with that of the driving voltage. If the difference in phase between these signals deviates from the 0° reference, the control circuit issues an error signal, V_ϵ, to the VCO adjusting the driving frequency thereby maintaining the device on resonance. The sensing and torquing electrodes are plated atop a gold printed circuit pattern, deposited on the chip in a previous photomask and metalization process, providing current pathways to large area bonding pads. Wire bonds extend from these pads to electrically isolated leads on the package containing the chip. The leads are in turn "plugged in" to a PC board containing the pre-amp.

The variable capacitor of the sensor bridge, $C(t)$, is on the input side of a low noise, high speed FET input op-amp. A 1 pf capacitor, C_{fb}, is in the feedback loop giving an output voltage, V_0, proportional to the ratio of the bridge capacitance to feedback capacitance multiplied by the carrier voltage. To clarify the operation of the device, simplified input/output relations are now derived. The capacitance of the sensor is

$$C(t) = \frac{\epsilon_0 A_s}{d_0 - v_s(t)} \qquad (3.19)$$

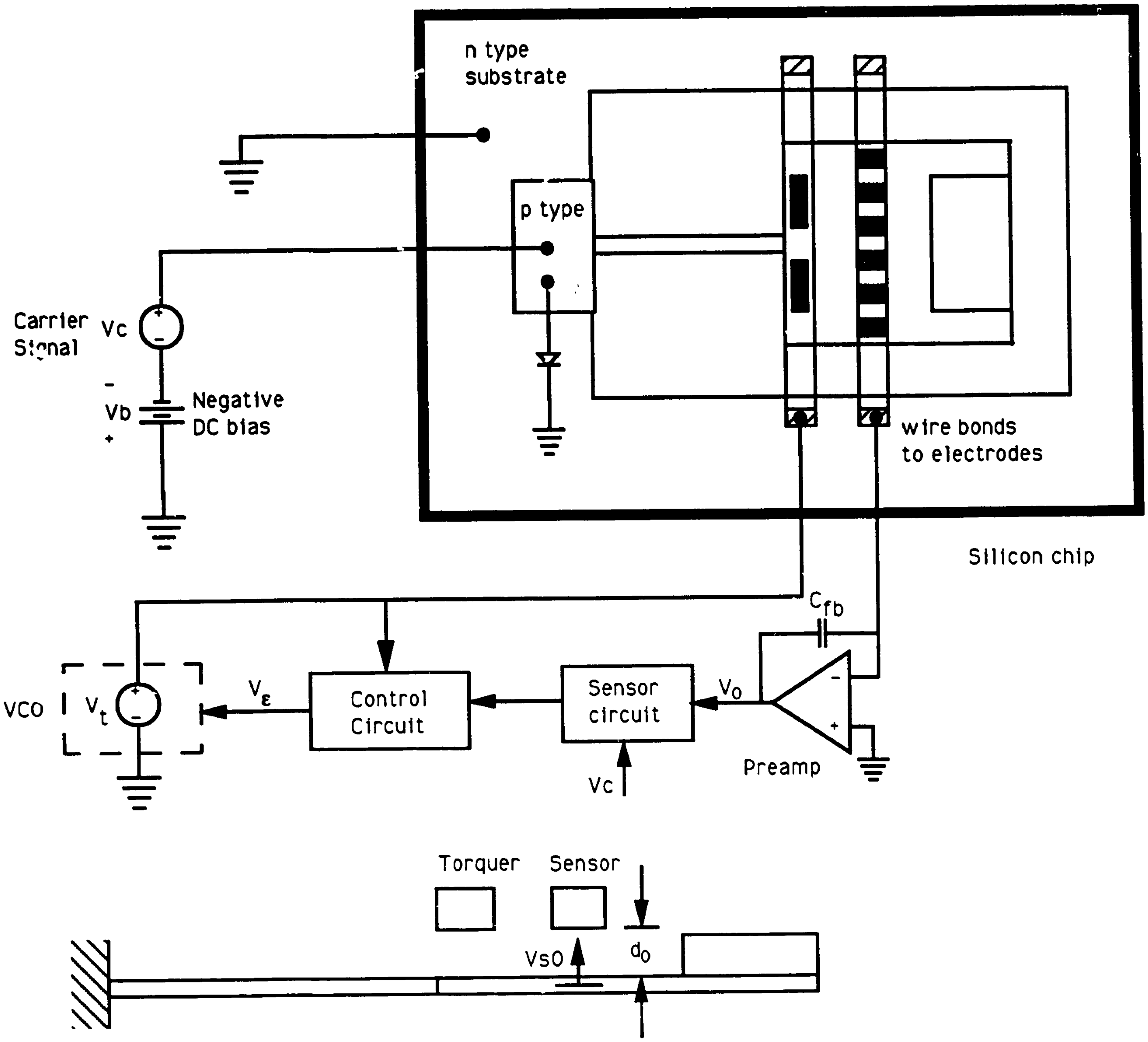


Figure 3.3: Sensing and Torquing Scheme

where,

$$\begin{aligned} v_s(t) &= \text{plate displacement under sensor: (m)} \\ &= -v_{s0}\sin(\omega t) \\ v_{s0} &= (\xi_1 v_1 + \xi_2 v_2 + \xi_3 v_3)\frac{p_0}{c\omega_n} \\ \epsilon_0 &= \text{permittivity of a vacuum: } \left(\frac{\text{coul}^2}{\text{N m}^2}\right) \\ d_0 &= \text{fixed air gap: (m)} \\ A_s &= \text{sensor electrode area: (m}^2\text{)} \end{aligned}$$

For large displacements, Equation 3.19 can be expanded into an infinite series.

$$C(t) = \frac{\epsilon_0 A_s}{d_0}\left[1 + \left(\frac{v_s(t)}{d_0}\right) + \left(\frac{v_s(t)}{d_0}\right)^2 + \left(\frac{v_s(t)}{d_0}\right)^3 + \cdots\right] \qquad (3.20)$$

The first term is a constant or nominal capacitance; the second is linear with displacement; subsequent terms are higher harmonics of decreasing magnitude. For small motions ($\frac{v_s(t)}{d_0} \approx 0.1$), the bridge capacitance is essentially linear with the beam displacement.

$$C(t) \approx \frac{\epsilon_0 A_s}{d_0}\left(1 + \frac{v_s(t)}{d_0}\right) \qquad (3.21)$$

If the frequency of the modulation envelope is approximately an order of magnitude less than the carrier frequency, the output voltage of the pre-amp is determined from the feedback rule for op-amps.

$$V_0 = -\frac{C(t)}{C_{fb}}V_c \qquad (3.22)$$

Equation 3.22 does not hold for high envelope frequencies because the basic equation for conservation of current at the inverting input of the op-amp is nonlinear and Laplace transforms do not apply. However, if the magnitude of the carrier signal does not change significantly over one cycle of the carrier frequency, the bridge capacitance can be treated as a constant and Equation 3.22 holds for steady harmonic motion.

Substituting Equation 3.21 into Equation 3.22 gives an expression for the output voltage of the pre-amp for small displacements.

$$V_0 = -\frac{\epsilon_0 A_s}{d_0 C_{fb}} \left(1 + \frac{v_s(t)}{d_0}\right) V_c \tag{3.23}$$

Next, the DC term in Equation 3.23 is blocked in a high pass filter and the resulting waveform boosted by a post amplifier. The carrier signal is then removed by a demodulator which multiplies the output signal of the post-amp by the original carrier signal. The result of this multiplication is a large separation of the device signal and carrier signal in the frequency spectrum. This separation allows the carrier to be attenuated by appropriate filters. The process is identical to AM radio reception. Eliminating the DC term in Equation 3.23 and multiplying by V_c and the post amp gain gives

$$V_0 = -k_a \frac{C_0 v_s(t)}{C_{fb} d_0} V_c^2 \tag{3.24}$$

where,

$$\begin{aligned} V_c &= V_{c0} \sin(\omega_c t) && (3.25) \\ C_0 &= \frac{\epsilon_0 A_s}{d_0} && (3.26) \\ \omega_c &= \text{carrier frequency: (rad/s)} \\ k_a &= \text{post amplifier gain: (volt/volt)} \end{aligned}$$

The demodulated output is as follows.

$$V_0 = \frac{k_a C_0 V_{c0}^2}{C_{fb} d_0} v_{s0} \sin(\omega_n t) \left(\frac{1}{2} - \frac{1}{2}\cos(2\omega_c t)\right) \tag{3.27}$$

The first term Equation 3.27 is directly proportional to $v_s(t)$. The second is composed of two sidebands at $2\omega_c - \omega$ and $2\omega_c + \omega$. These higher harmonics are attenuated by a low pass active filter and do not contribute significantly to the filtered output. The resulting output equation is as follows.

$$V_0 = -\frac{k_a C_0 V_{c0}^2}{2 C_{fb} d_0} v_s(t) \tag{3.28}$$

The previous operations show how the demodulation works in principle; however, Equation 3.28 is not the correct expression because the resulting units are volts squared which is inconsistent with the output. In practice, the physical demodulator circuit component operates with its own transfer function specified by the manufacturer. Therefore, to account for the gain of the demodulator, the nominal capacitance of the sensor bridge, the air gap, and other amplifiers and filters in the circuit the output equation is written in terms of a sensor constant: k_s with units of volts/meter.

$$V_0 = -k_s v_s(t) \tag{3.29}$$

$$k_s = k_d k_a k_f \frac{C_0 V_{c0}}{C_{fb} d_0} \tag{3.30}$$

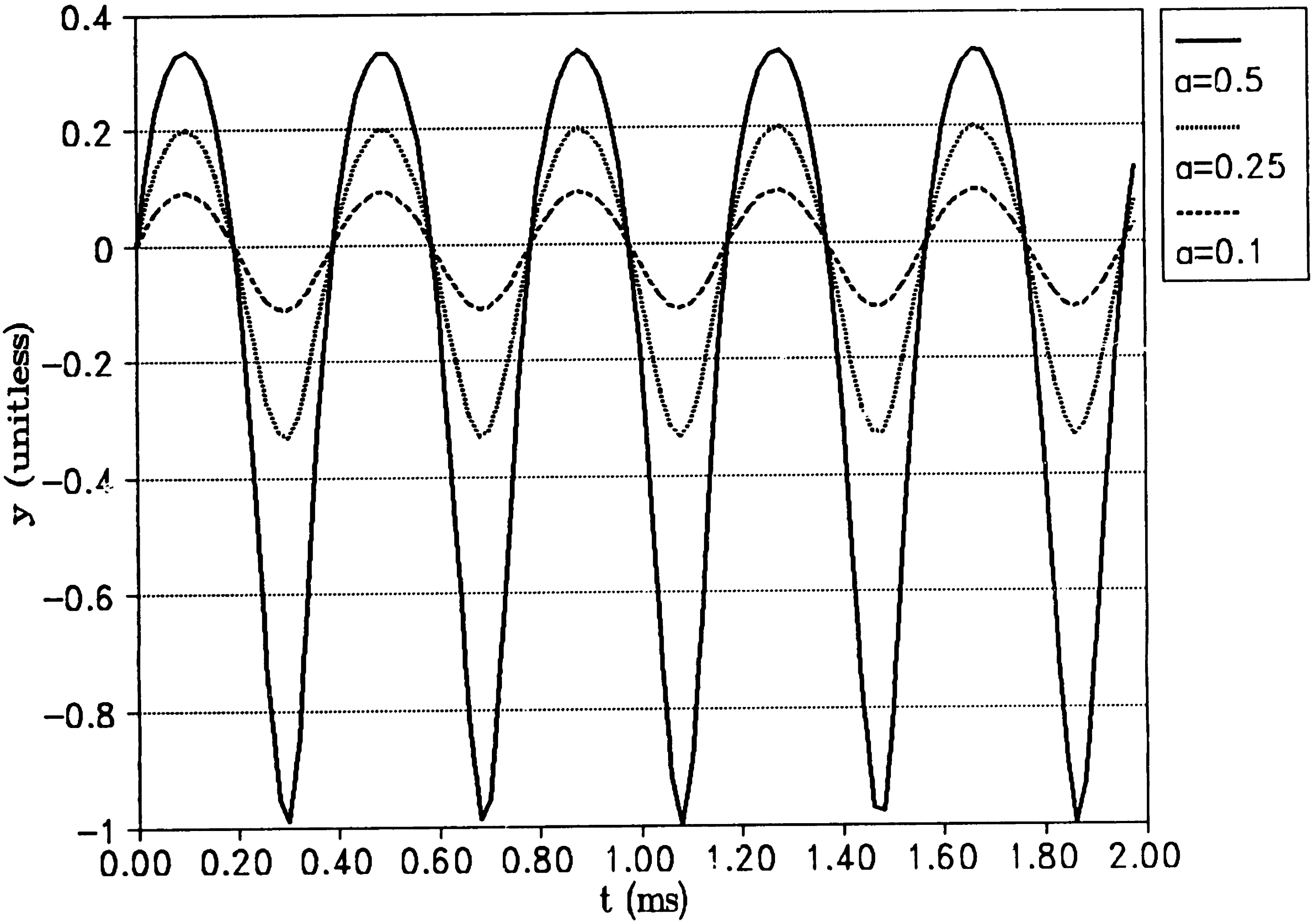
where,

$$\begin{aligned} k_d &= \text{demodulator gain: (volt/volt)} \\ k_f &= \text{filter gain: (volt/volt)} \end{aligned}$$

Equation 3.29 was derived for small motions; however, the same analysis procedure also applies for large displacements. In this case, Equation 3.20 is substituted into Equation 3.22 then demodulated and filtered. The result is a signal composed of multiple harmonics extending upward from the resonant frequency of the device .

$$V_0 = -k_s v_s(t) \left[1 + \left(\frac{v_s(t)}{d_0} \right) + \left(\frac{v_s(t)}{d_0} \right)^2 + \left(\frac{v_s(t)}{d_0} \right)^3 + \cdots \right] \tag{3.31}$$

Figure 3.4 shows the effect of increasing plate displacement. The plotted function, y, is identical in form to Equation 3.31 but unitless. Also, $v(t)$ has a minus sign which is incorporated into the function y. The amplitude ratio, a, is defined at the amplitude of motion divided by the air gap dimension. Note that in Figure 3.4 the zero crossing remains unchanged in spite of the distortion caused by large plate displacements. This fact means the frequency control system is basically insensitive to the inherent nonlinearity of the sensing scheme. However fortunate this result may be, the above analysis was somewhat simplified. The sensor electrode has finite width making it sensitive to capacitance changes caused by pure rotation of the end plate.



$$
\begin{aligned}
y &= \frac{a\sin(\omega_n t)}{1 + a\sin(\omega_n t)} \\
a &= \frac{v_{s0}}{d_0} \\
\omega_n &= 2550 \text{ Hz}
\end{aligned}
$$

Figure 3.4: Sensor Output Waveform

Therefore, both the angle and translational displacement of the end plate are required to fully account for the capacitance change. In section 2.2 it was shown that the translational and rotational degrees of freedom tend to act independently in deflecting the cantilever. A further complication arises because the sides of the sensor electrode radiate, or "fringe", to the end plate. Therefore, some phase error is introduced by large plate displacements; however, the effects generally remain small except at very large displacements where the end plate is in close proximity ($\approx 1\mu m$) to the outermost edge of the sensor electrode. In any event, such phase errors depend on the amplitude of motion which remains nearly constant for small frequency shifts. Therefore, these errors can be compensated for by the setpoint of the control electronics. In general, complex expressions are required to evaluate capacitance fringing, especially if rotation is included. Although the design program, FTEST2.PAS, does not incorporate fringing or rotational effects, reasonably accurate predictions of sensor output are obtained. The previous discussion clarifies how the sensing scheme works and aids in interpreting the performance and output of the device.

Figure 3.5 is a time trace of the output waveform of the T38 test device at resonance. A torquing voltage of 10 volts is applied giving an amplitude ratio of 0.23 which can be roughly compared to the ratio of 0.25 presented in Figure 3.4. The correspondence in shape and the relative height of the peaks is close. Figure 3.6 is a power spectrum of the waveform shown in Figure 3.5. The cursor is centered on the resonant frequency of the device. Harmonics extend from the resonant frequency upward in consecutive multiples (ω_n ,$2\omega_n$,$3\omega_n$,$\cdots$) as predicted by Equation 3.31. Figure 3.7 is a time trace of the device with a driving voltage of 5 volts applied giving an amplitude ratio of 0.06 which is well inside the linear range. Figure 3.7 shows the device approaching the limits of detectable motion as noise becomes evident at the peaks. The zero crossing, however, remains discernible so that the phase control system will continue operation without losing phase lock. A power spectrum of this waveform is shown in Figure 3.8. Only one additional harmonic is visible at twice the resonant frequency. The magnitude of this harmonic is approximately a factor of 10 less than that of the resonant peak, and therefore, contributes a negligible amount to the total output.

Figure 3.9 is a spectrum analysis of the T38 device with the cursor centered on the resonant frequency. The characteristics of the spectrum indicate the device behaves as a first order system with a phase shift of 90° at its first mode resonant peak. Multiple peaks are not present indicating that spurious torsional modes or modes associated with flexing of the end plate are not present. The driving voltage was limited to 5 volts ensuring a linear sensor output. The curve fitting routine of the spectrum analyzer computed the poles of the system to be -30.179 ± 2550 Hz. The calculated value of the real part of the pole is approximately -5 Hz, a factor of six less. The value given by the spectrum analyzer is too high, for the device could not produce measurable output with such high damping giving a torquing voltage of 5 volts. Furthermore, with such high damping the waveform shown in Figure 3.5 is not possible at 10 volts because sufficient displacement would not be generated to produce the nonlinearity. As shown in Figure 3.7, a significant level of noise is present when the device is driven with 5 volts. Noise has the effect of raising the magnitude of the flat portion of the spectrum thus flattening out the curve and consequently producing an overestimate of damping by the curve fitting routine.

Figures 3.10 through 3.12 are layouts of the actual circuits implemented. The pre-amp is housed inside the environmental control chamber as near as possible to the device to minimize parasitic capacitances that introduce noise. The preamp circuit components are protected from the humid atmosphere by conformal coating, and heat sinks to ground are provided to prevent overheating when operating in vacuum. The remainder of the circuit is outside the chamber mounted in standard aluminum chassis. The pre-amp, post-amp, and sensing circuit are fabricated on PC boards to minimize the sensitivity of the circuit to noise and electromagnetic interference.

The layout of the control circuit is shown in Figure 3.14. As discussed, the output frequency must be divided by two in order to compare phase. The divide by two flip-flop performs this division. The phase detector operates as an "exclusive or" gate by comparing the timing of the edges of the two input signals. The output is filtered and conditioned by the compensator and a DC signal proportional to the phase error is issued to the VCO. Figure 3.15 shows the device operating at resonance. The torquing voltage is the lower curve and is at half the frequency of the device shown directly above.

Data acquisition is accomplished by a Metrabyte CTM05 frequency counter board mounted in a Dell 316 computer. Software drivers for this board were purchased but the algorithm used to compute frequency was inefficient and required substantial modifications to improve acquisition time and accuracy. The counter operates by counting the zero crossings of an input signal when

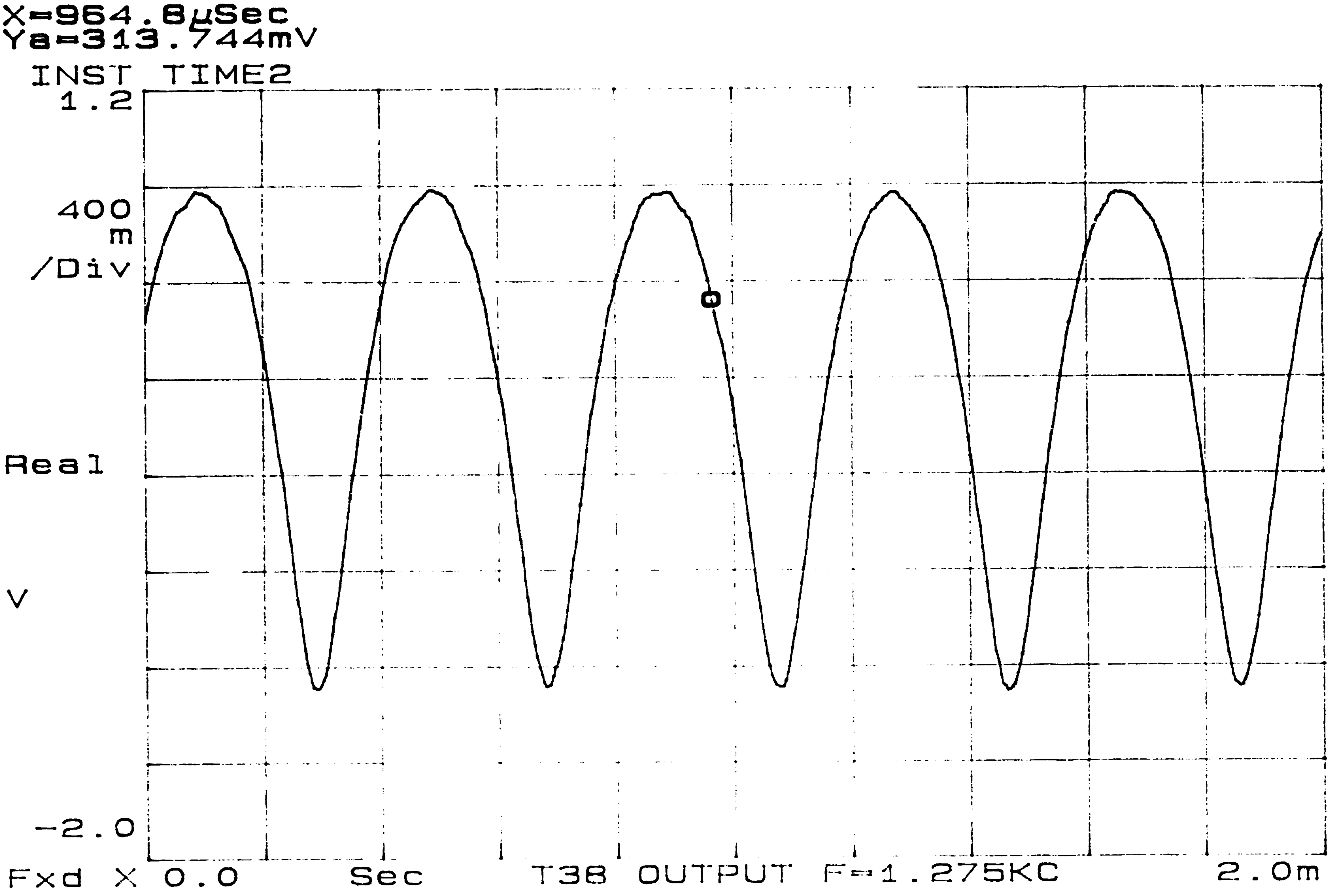


Figure 3.5: Sensor Waveform: Torquing Voltage = 10v

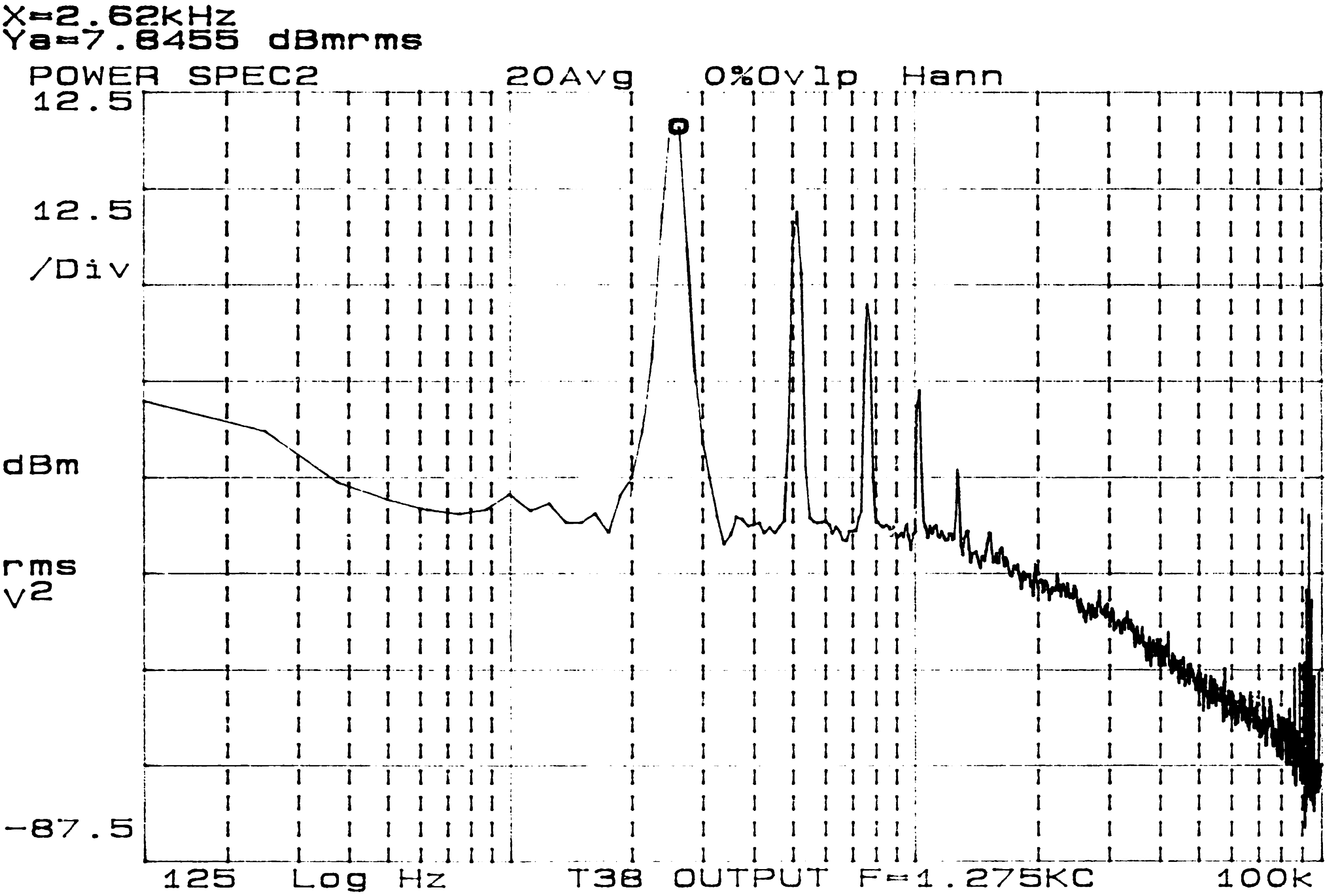


Figure 3.6: Power Spectrum: Torquing Voltage = 10 v

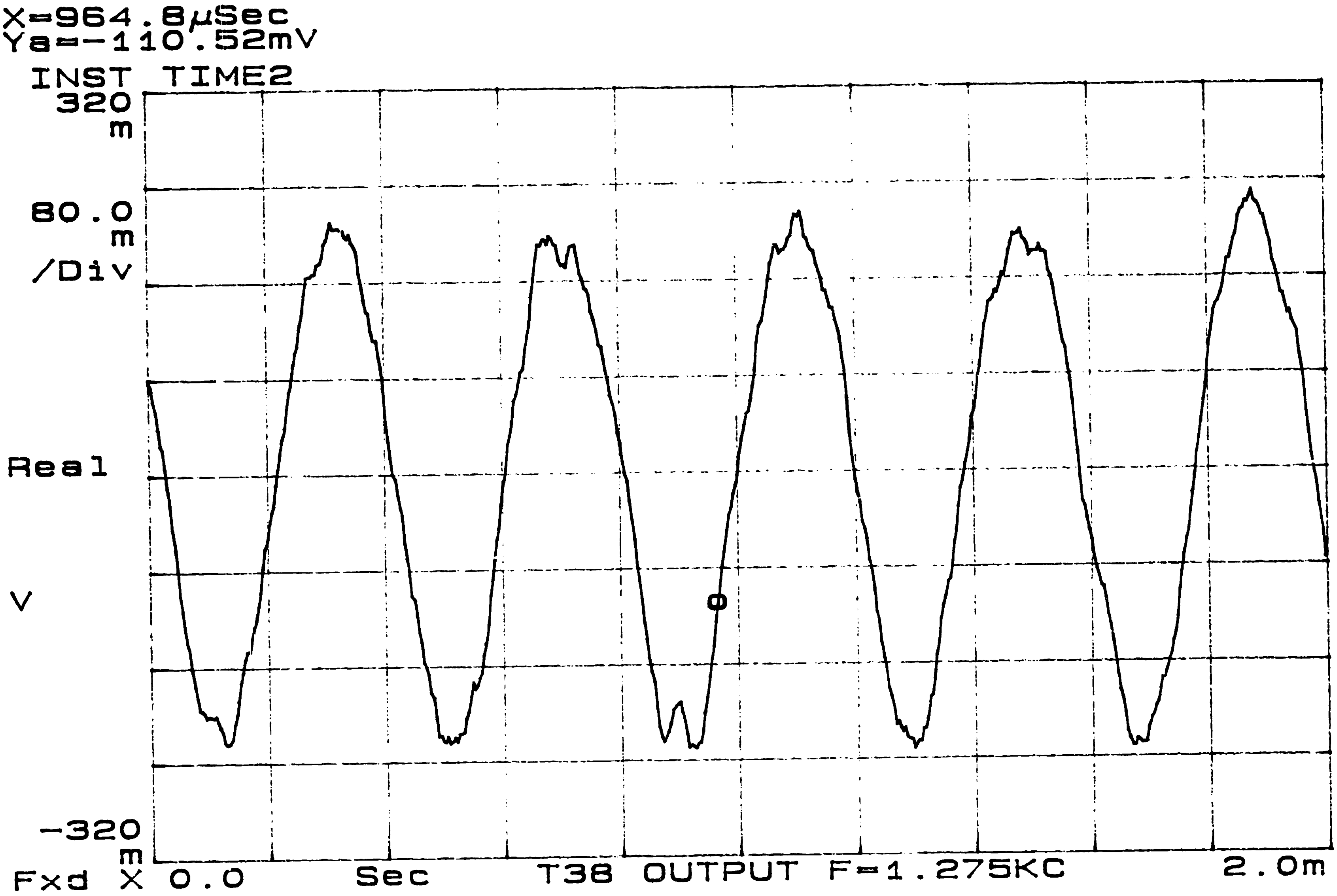


Figure 3.7: Sensor Waveform: Torquing Voltage = 5v

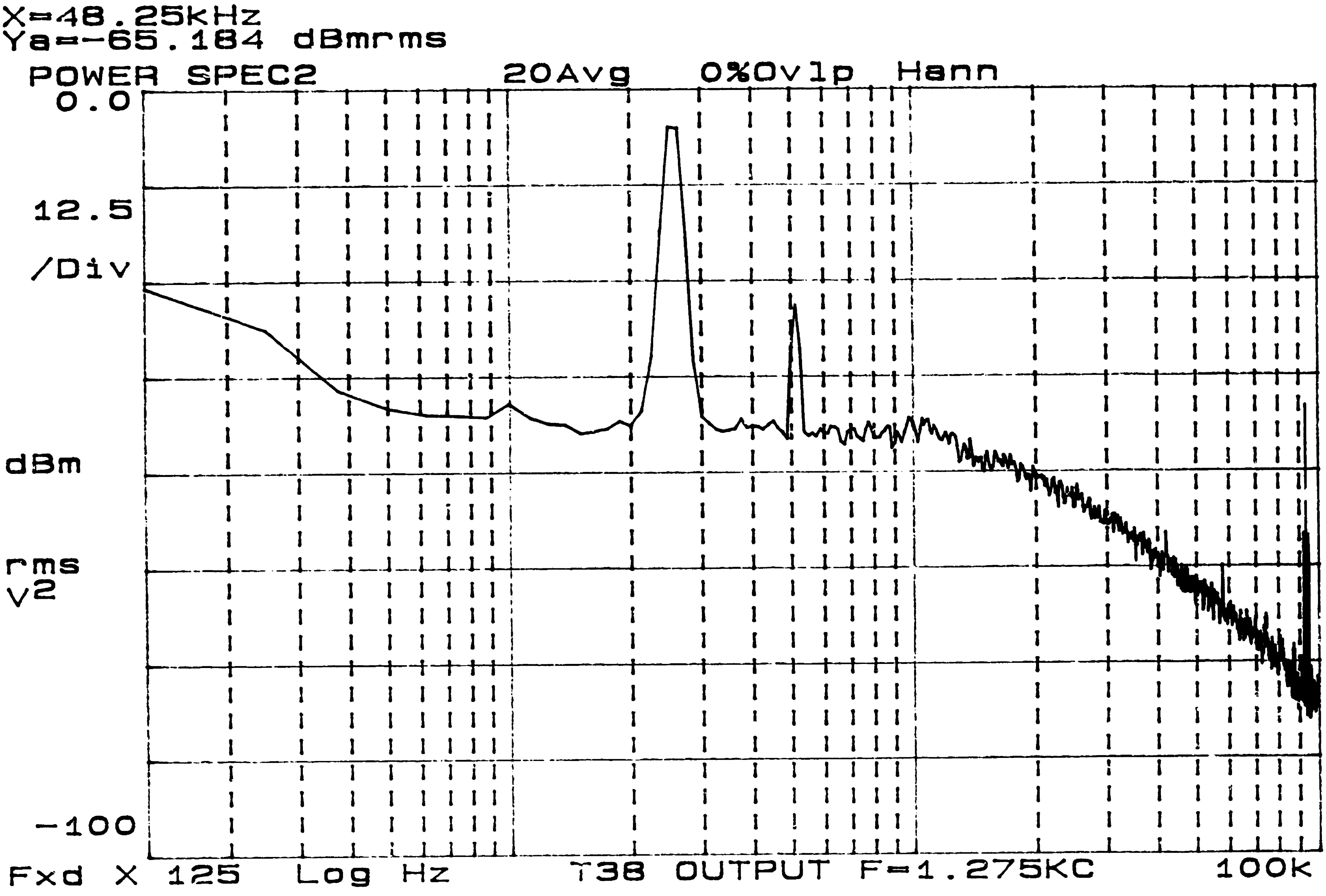


Figure 3.8: Power Spectrum: Torquing Voltage = 5v

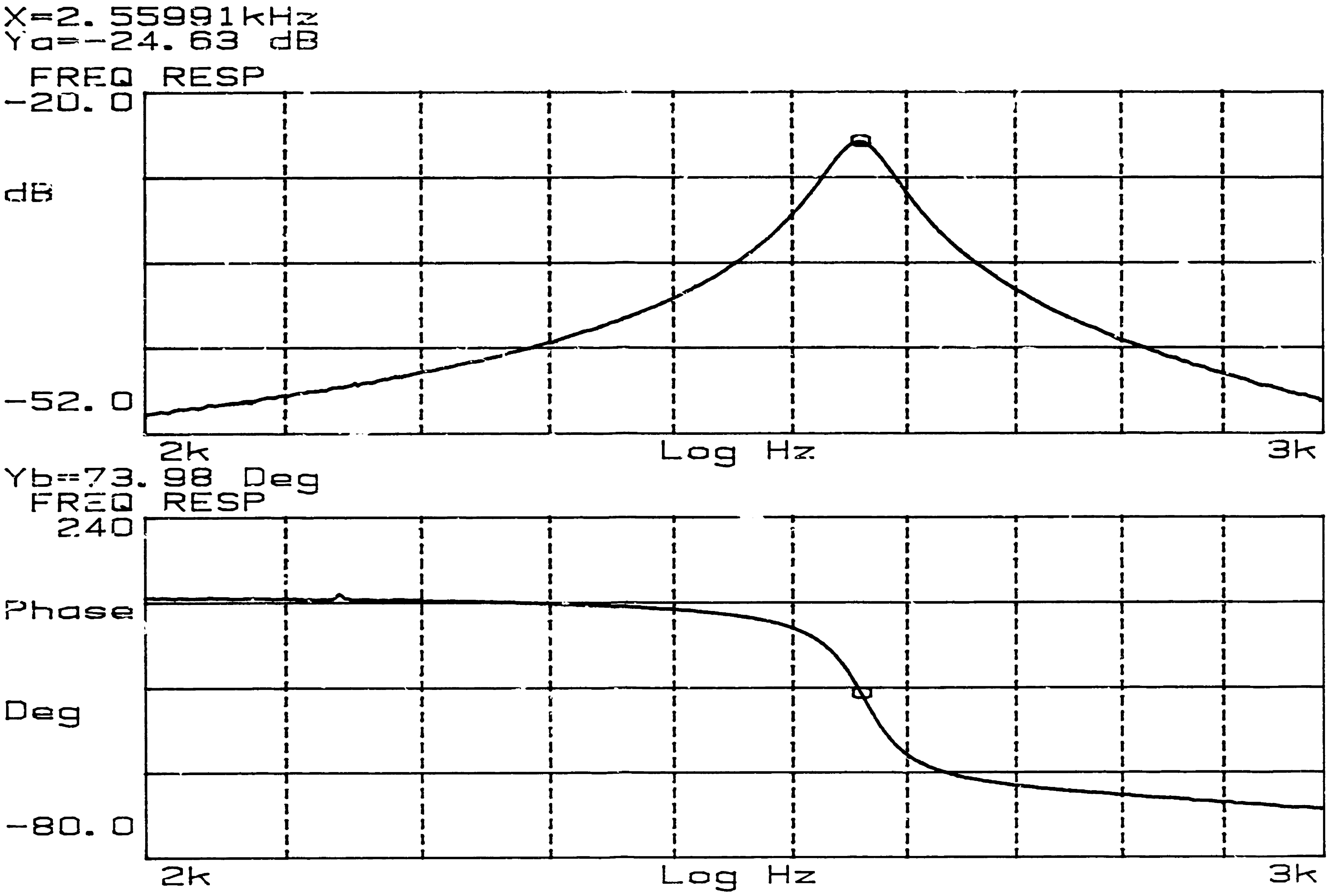


Figure 3.9: Spectrum Analysis: Torquing Voltage = 5v

the gate of the counter is high. In the original software, the gate time was set by the user and the device signal was routed to the input of the counter. Thirty to fortyfive seconds per measurement were required to accurately measure the resonant frequency of the device using this method. By dividing down the frequency of the device signal and routing it to the gate of the counter, the 1 MHz internal clock of the counter board could be used as the input signal. This method of frequency counting significantly improved acquisition time and accuracy. The modified code is included in Appendix E. The acquisition rate is approximately 5 samples per second with an accuracy of 0.5 Hz. Figure 3.13 shows the experimental setup.

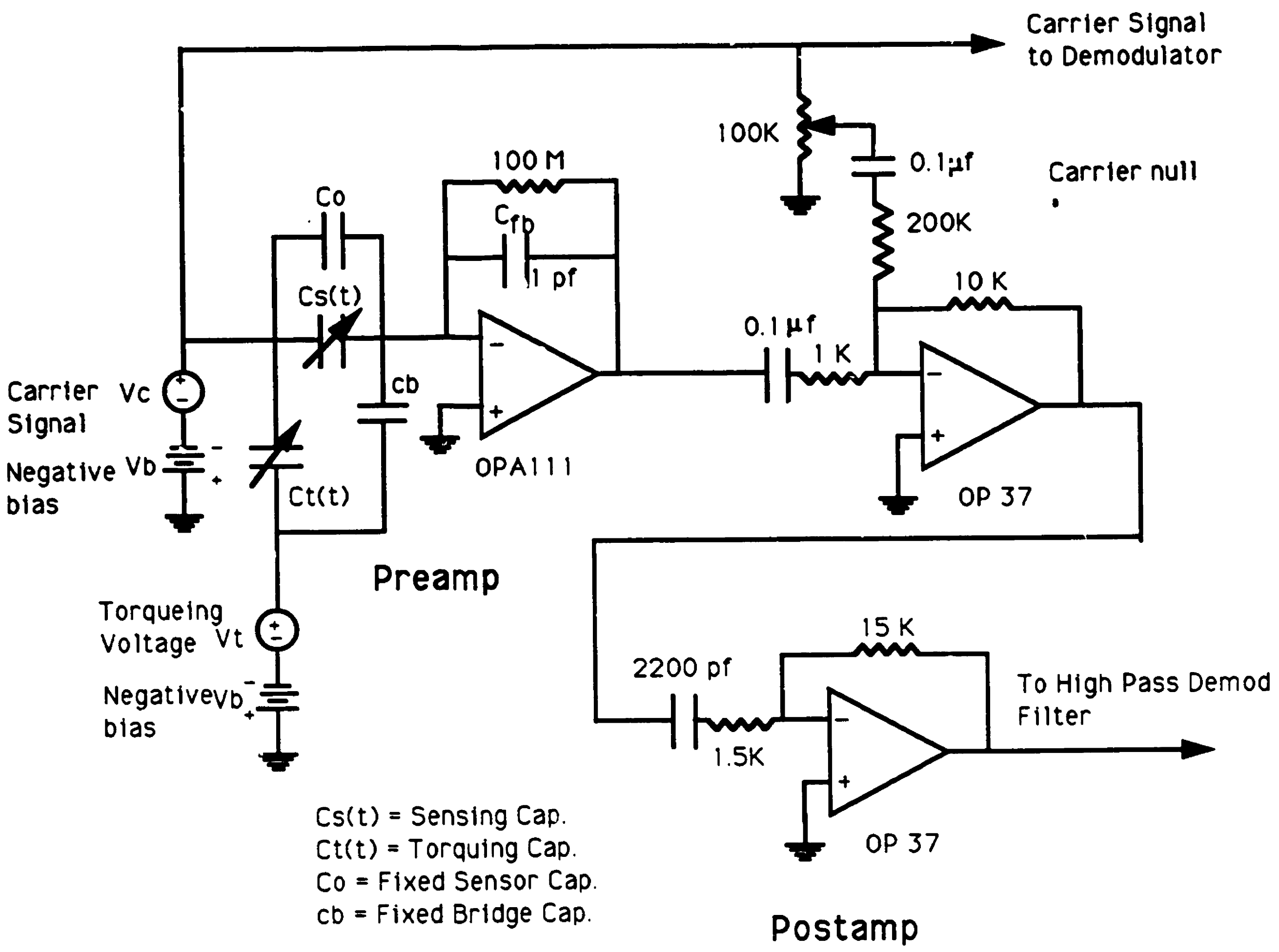


Figure 3.10: Pre-amp and Post-amp

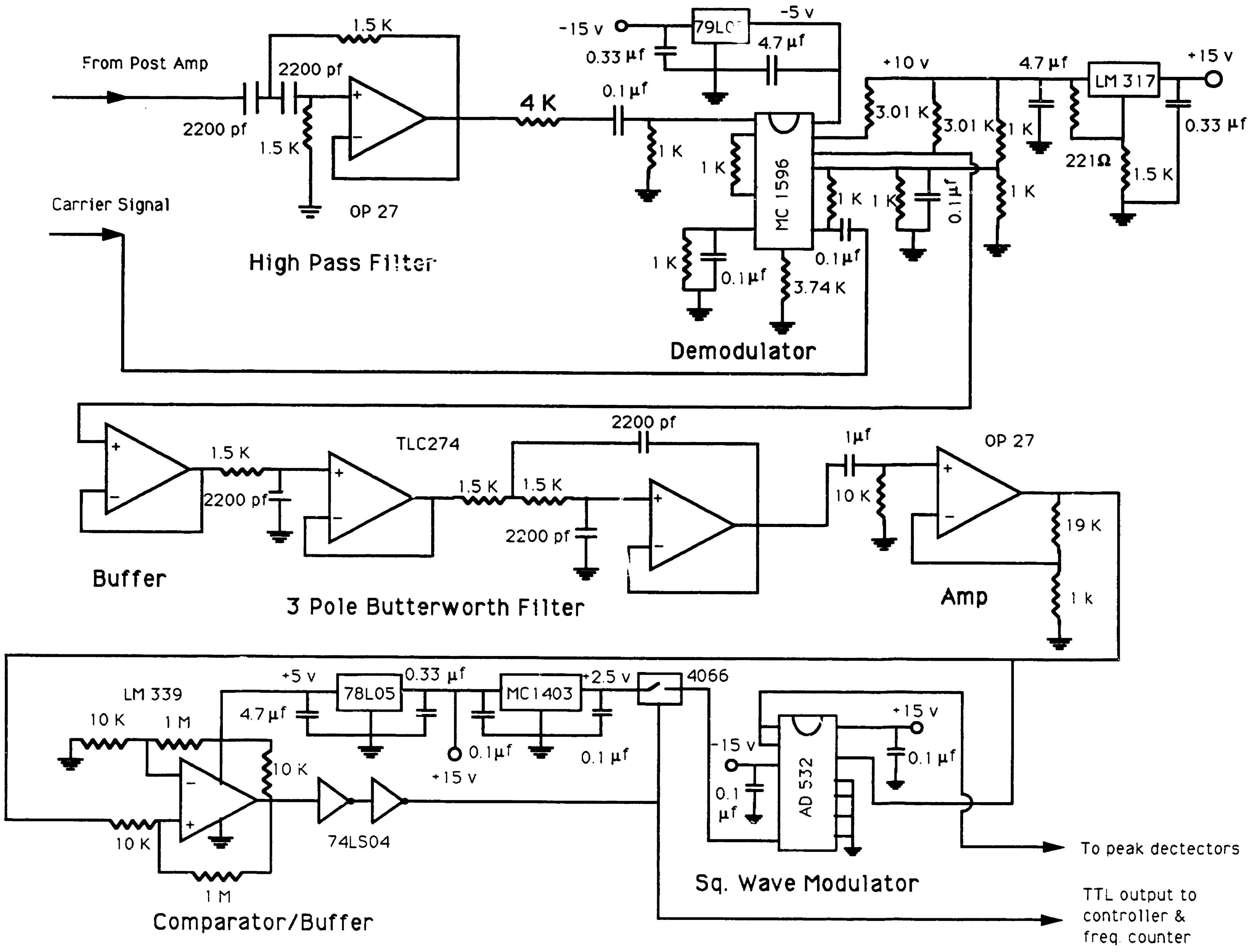


Figure 3.11: Sensor Circuit

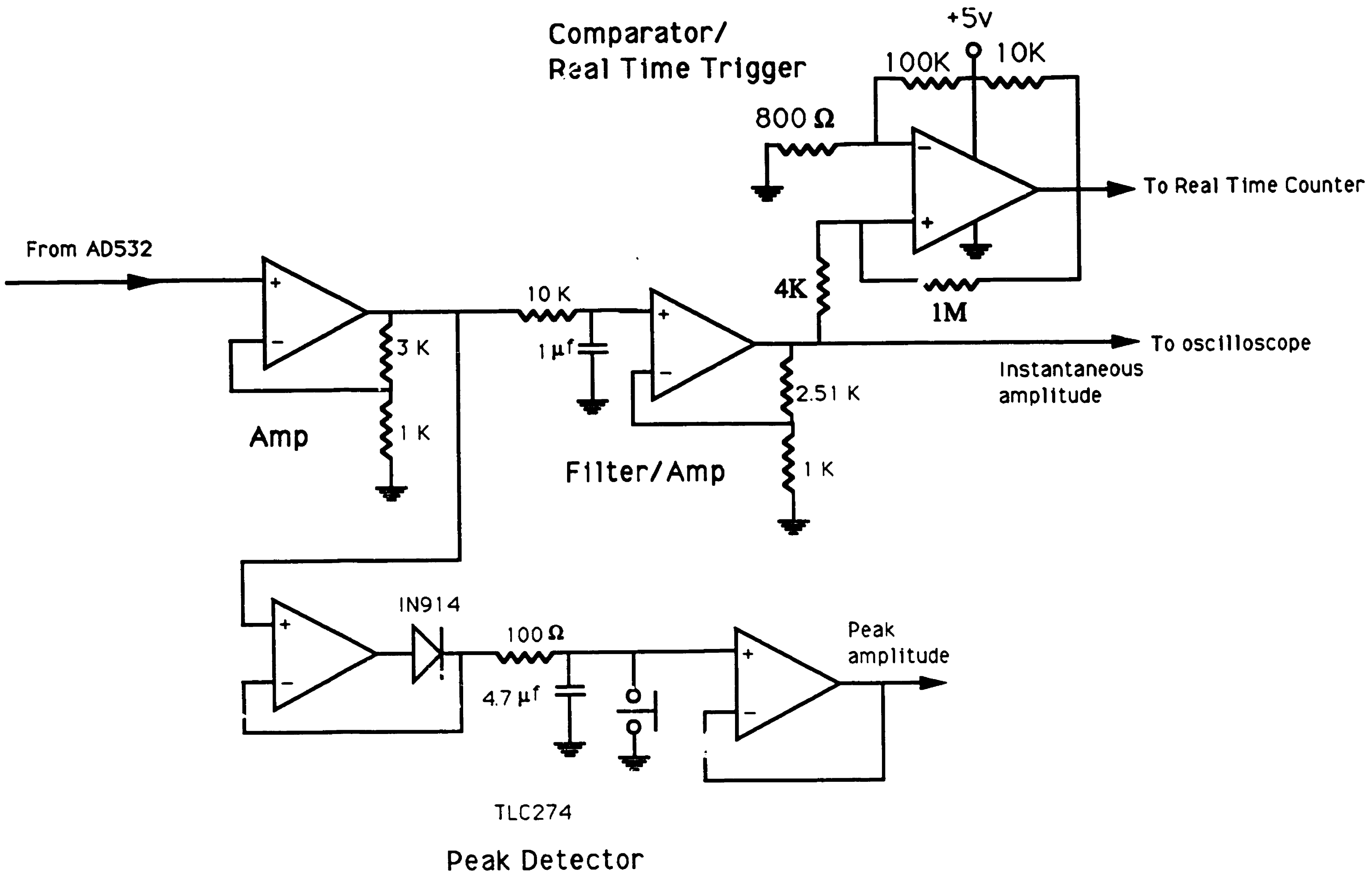


Figure 3.12: Instantaneous Amplitude and Peak Dectection Circuitry

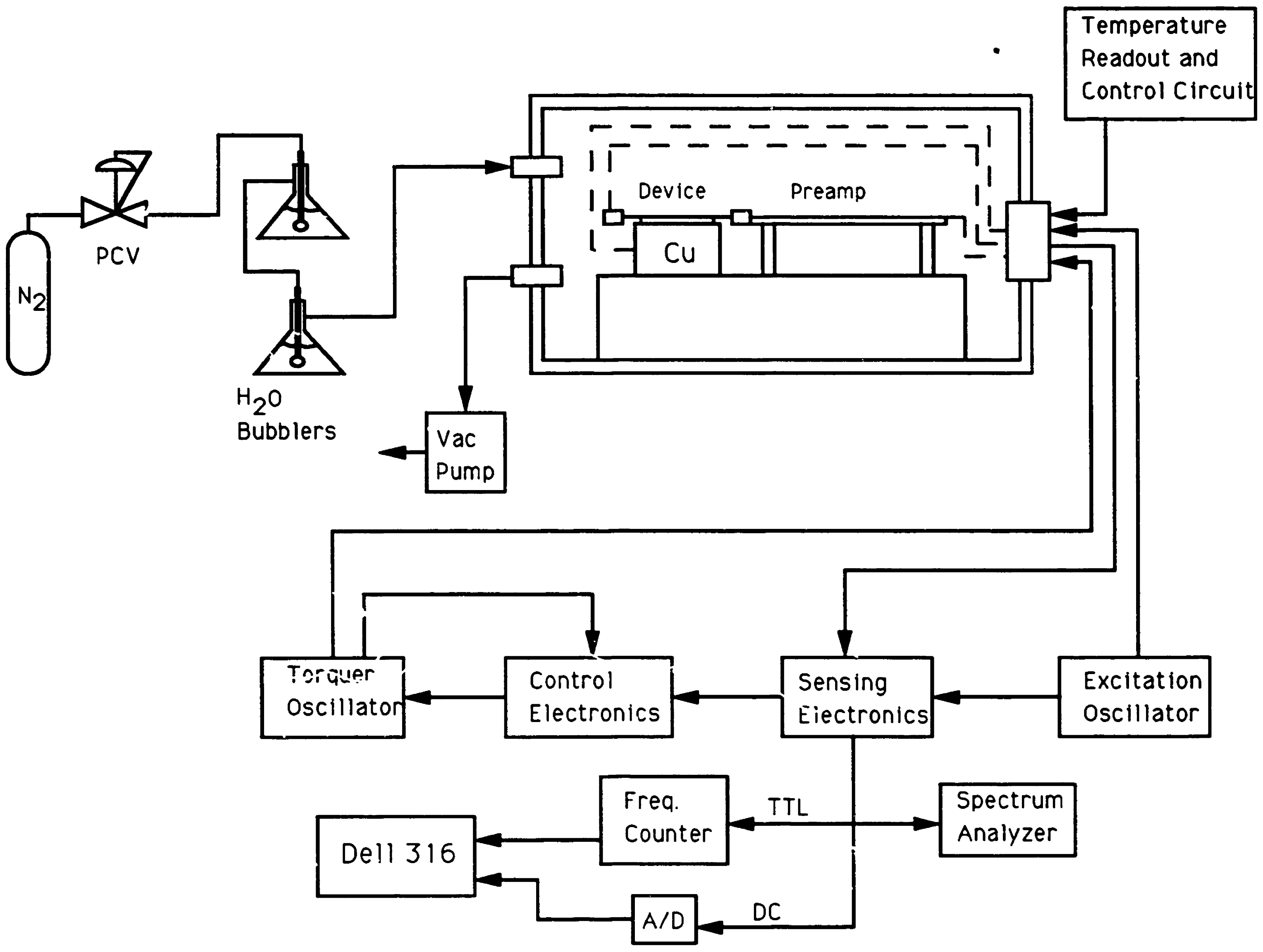


Figure 3.13: Experimental Setup

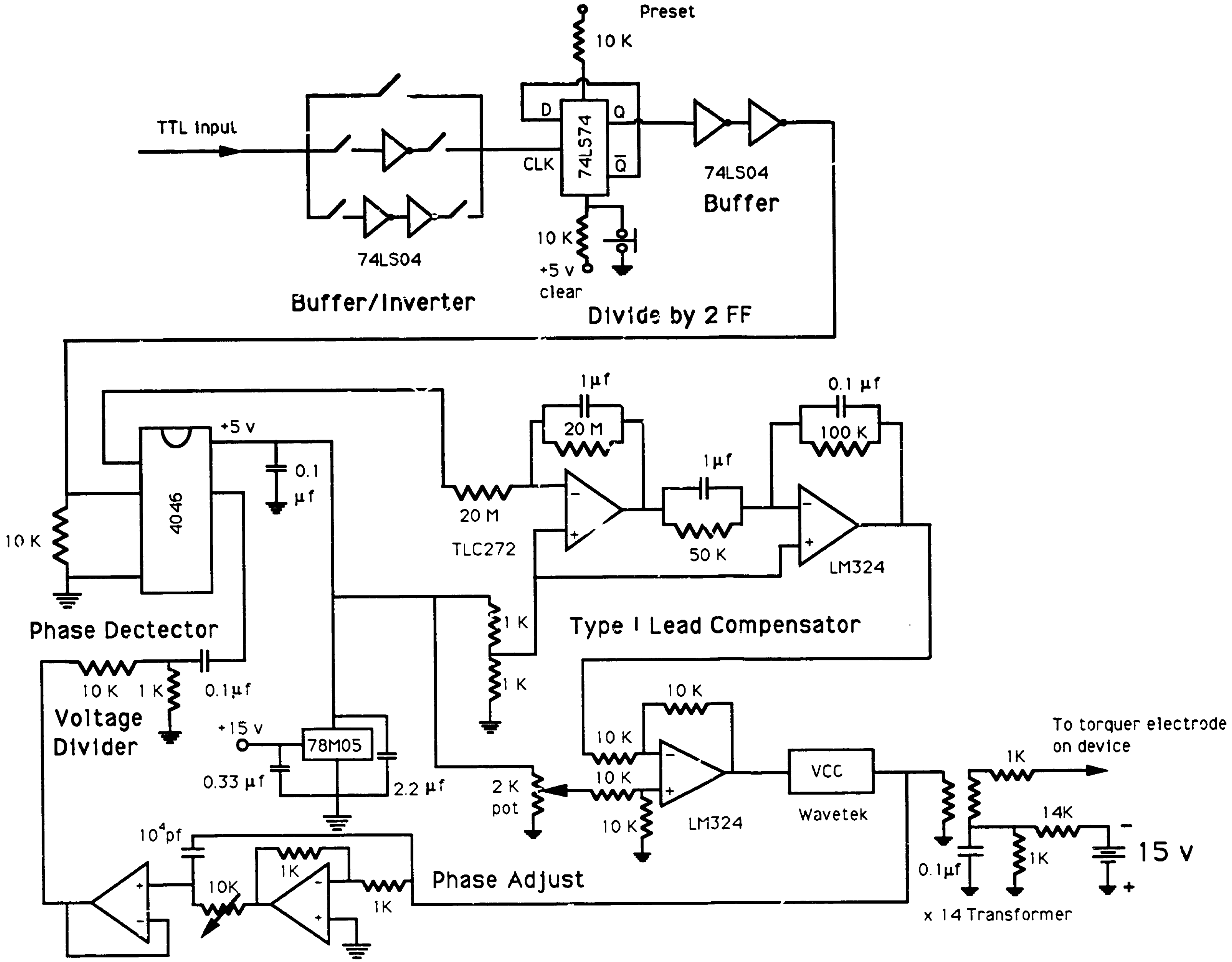


Figure 3.14: Control Circuit

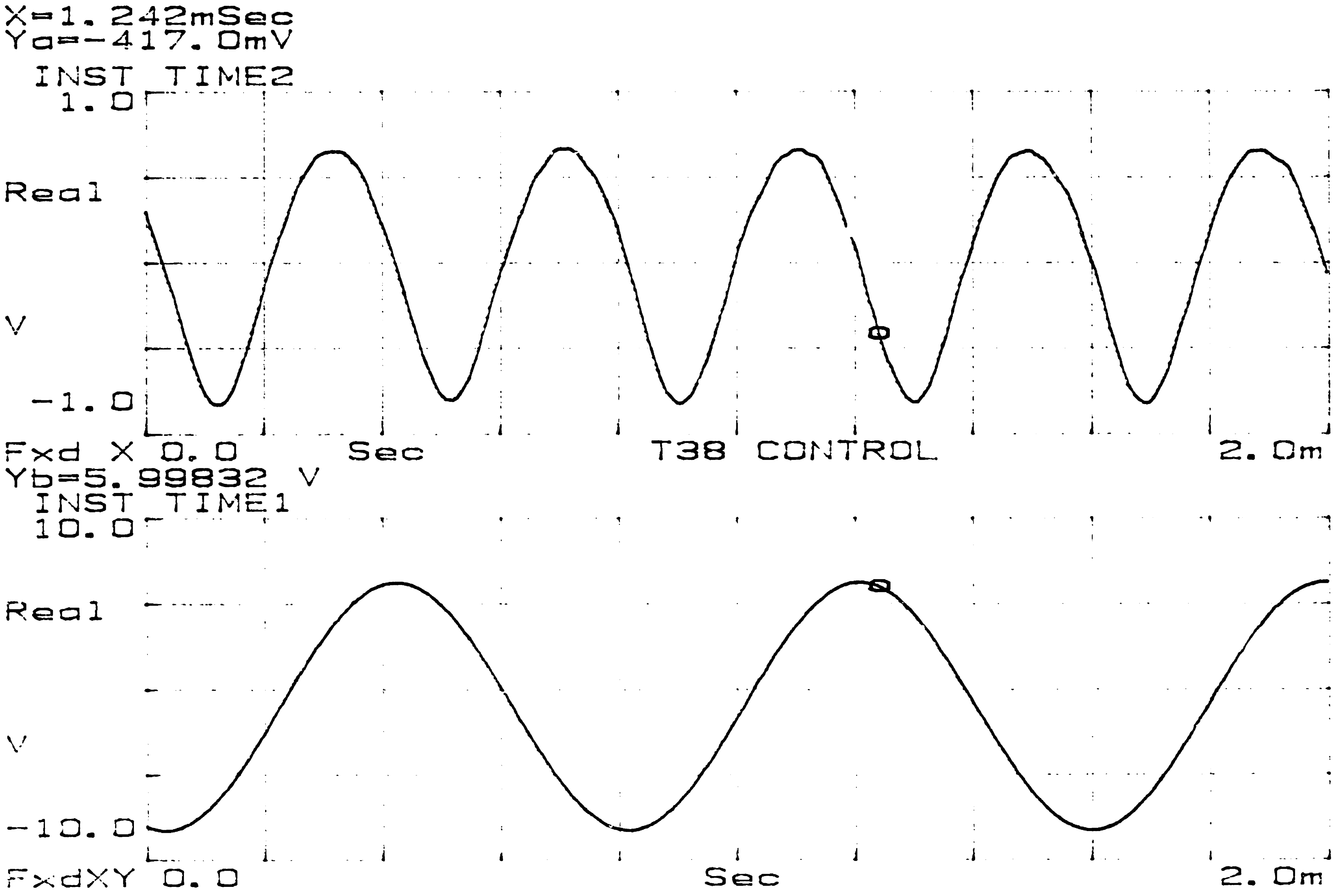


Figure 3.15: Controlled Response

Chapter 4

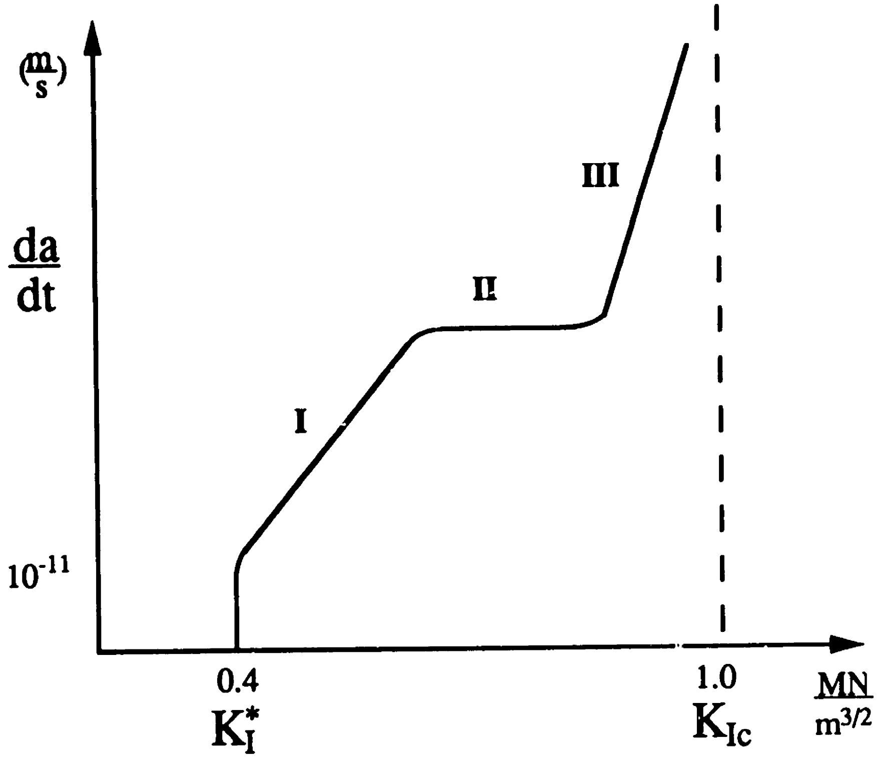
Fatigue Analysis

4.1 Theory

In the literature review, it was mentioned that static and dynamic stress corrosion fatigue measurements have been made on a number of ceramics including silicon. Figure 4.1 is a representation of static corrosion fatigue in silica glass as measured by Wiederhorn [45]. On the vertical axis is the crack tip velocity, $\frac{da}{dt}$, and on the horizontal axis is stress intensity, K_I. The curve is divided into three regions: I, II, and III. Region I is the slow growth region where the crack tip velocity is exponentially related to the applied stress, and also depends on the corrosion reaction rate at the crack tip. Region II is the stress independent region where the velocity depends only on the rate of transport of water molecules to the crack tip. Increasing the applied stress in this region does not enhance crack growth. Region III is the fast growth region where the corrosion reaction is independent of the water content. Region III behavior is attributed more to the structure and intrinsic properties of the material rather than environmental conditions. A curve similar to Figure 4.1 was expected in this experiment.

Figure 4.2 was extracted from Wiederhorn article on subcritical crack growth in ceramics [45]. This curve clearly shows the three regions and the effect of increasing humidity. As the humidity level is increased, the region I and II curves are translated up with constant slope. Translating the curves in this manner is equivalent to raising the slope intercept on the vertical axis which can be interpreted as an increase in the corrosion reaction

Expected Results



I. Slow Growth Region
(Depends on reaction rate at crack tip)

II. Stress Independent Region
(Depends of rate of transport of water to crack tip)

III. Fast Growth Region
(Independent of water content; depends on material properties alone)

Figure 4.1: Crack Velocity vs. Stress Intensity

rate at the crack tip. Note that in Figure 4.2 the lowest velocity measured was on the order of 1×10^{-8} m/s. In another experiment, Wiederhorn and Johnson measured crack velocities in PYREX as low as 1×10^{-12} m/s. With our apparatus, crack tip velocities can be measured on the order of 1×10^{-14} m/s, which is two orders of magnitude better resolution than Wiederhorn and Johnson's measurements. The size of the micromechanical fatigue structure is on the order of what is generally considered the minimum crack size necessary to initiate fatigue in macro-scale specimens. Therefore, given the accuracy of the testing technique, measurements of extremely small crack extensions are possible over extended periods of time. Such measurements have not previously been reported in the literature.

Figures 4.3 and 4.4 were extracted from Bhaduri and Wang's article on stress corrosion fatigue in silicon [2]. Bhaduri and Wang used pre-cracked double torsion single crystal silicon specimens in load relaxation experiments. Figure 4.3 does not show any region I or II behavior in air at 10% humidity, but these regions are seen in Figure 4.4 when an electric potential is applied to the specimen. The slope of the fatigue curve in region III was also decreased by the presence of an electric field. Bhaduri and Wang do not definitively account for these effects and were unable to show how the presence of an electric potential enhances fatigue. However, other researchers are referenced who reported hardness decreases in semiconductor crystals when an electric potential is applied.

Gilman *et al.* conducted a series of experiments on germanium wafers with a microhardness indenter when a voltage potential of a few volts was applied across the crystal [44]. Gilman measured hardness decreases of 30-60% when the potential was applied at room temperature or below. This electromechanical effect was only seen when the indenter tip was in the proximity of measurable current density and a potential difference existed between the indenter tip and crystal. The electromechanical effect disappeared when the indenter was moved away from the region of significant current density and when the indenter tip and surface were at the same potential. The electromechanical effect appeared near the surface, extending only about $2\mu m$ to $3\mu m$ into the crystal and was confined to the plastic region surrounding the indenter. A strong temperature dependence was also measured. The largest hardness decreases occurred at low temperatures, $-196°C$, with negligible decreases near 200°. High temperature indentation experiments conducted by Haasen and Hirsch showed that doped semiconductor materials have en-

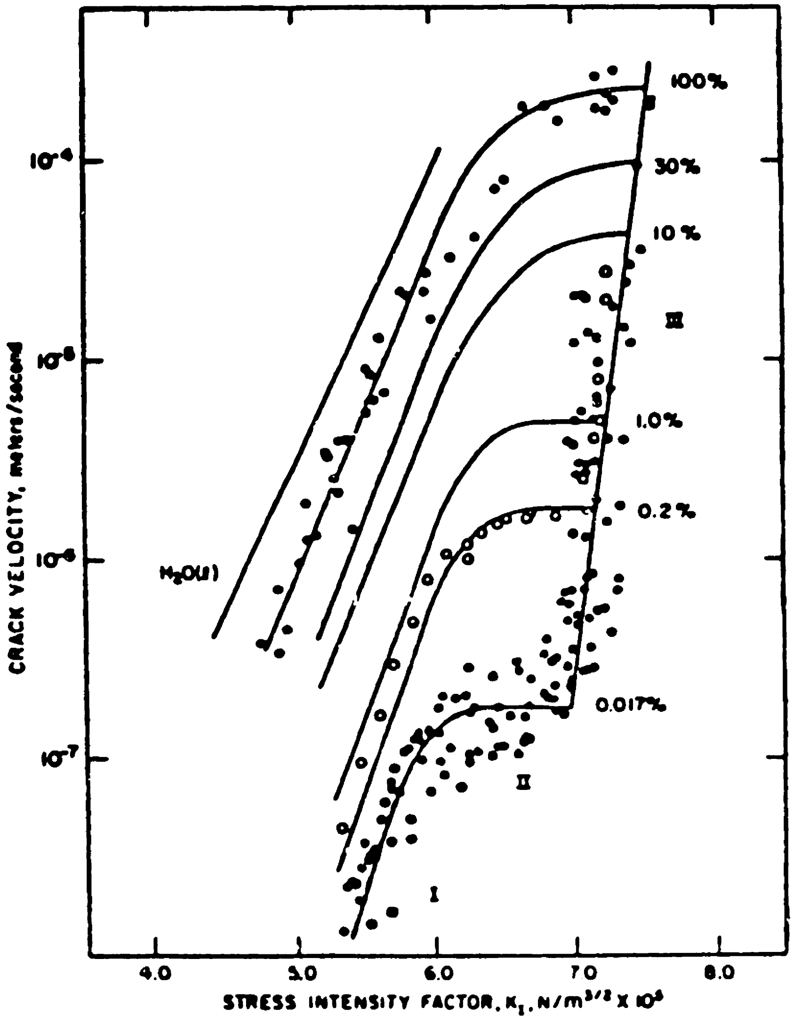


Figure 4.2: Typical Stress Corrosion Fatigue Curve (Wiederhorn)

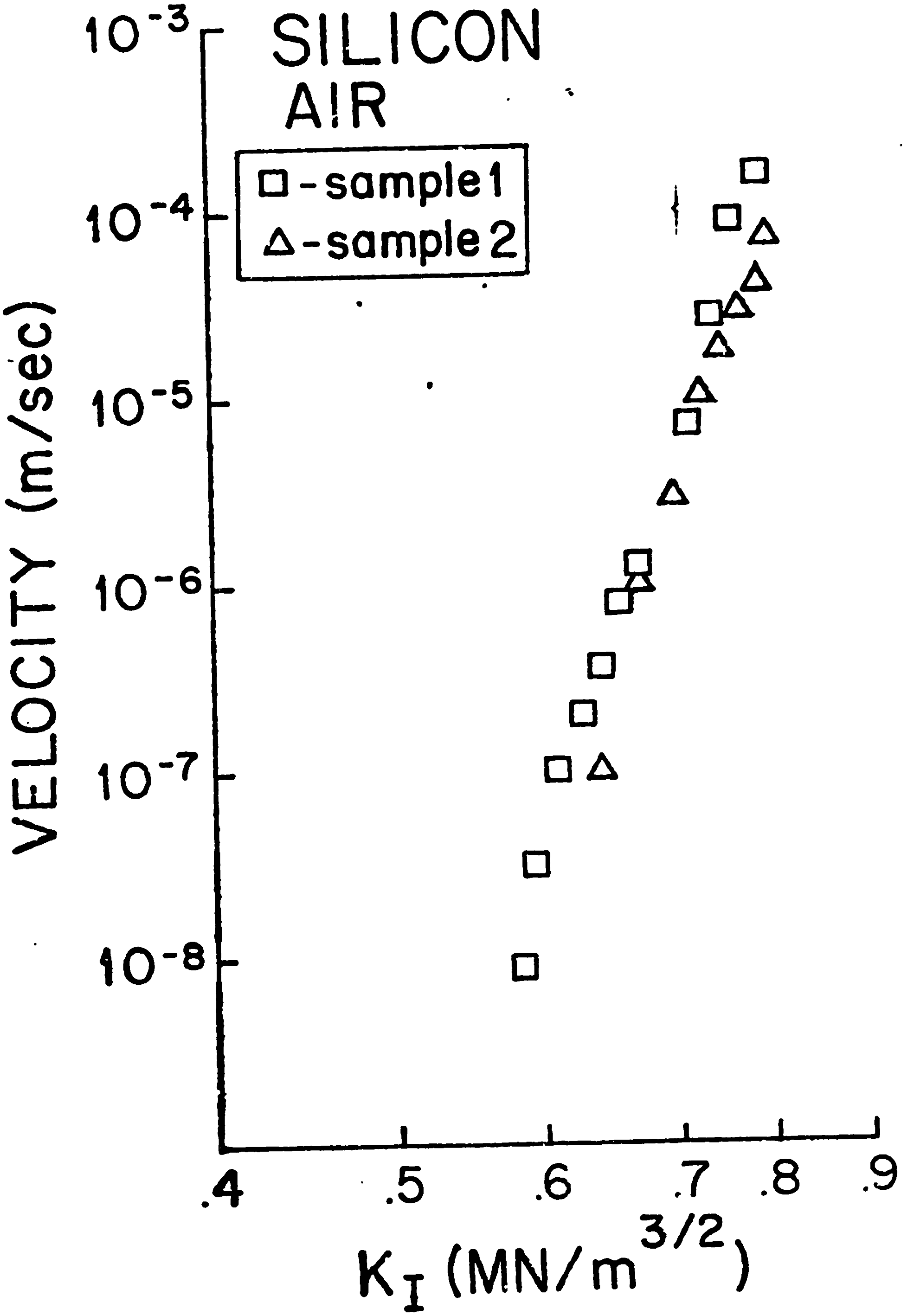


Figure 4.3: Fatigue of Silicon in Air at 10% Humidity

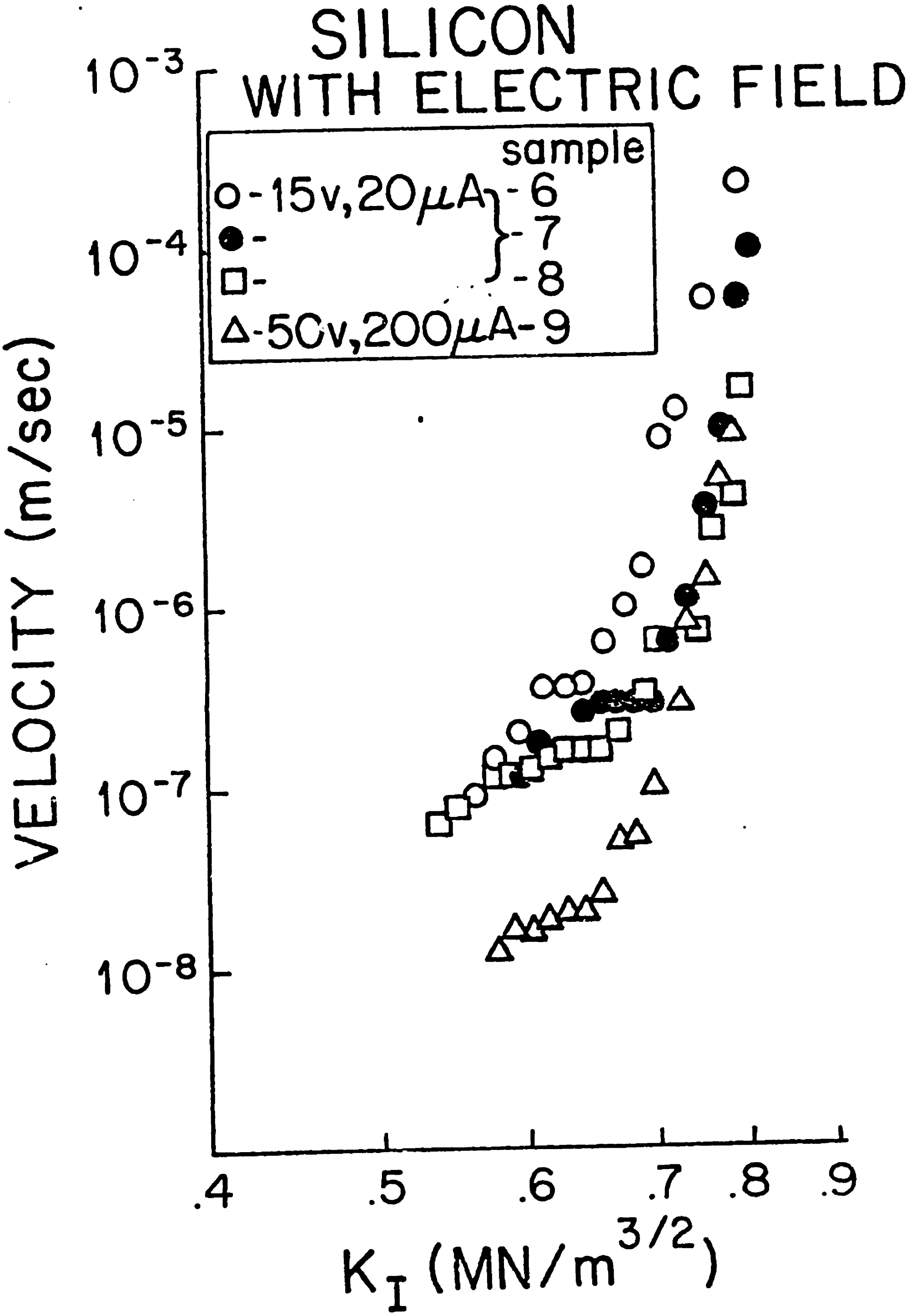


Figure 4.4: Fatigue of Silicon in Air with Electric Field

hanced dislocation motion, as opposed to undoped crystals, at temperatures near the ductile to brittle transition temperature (DBTT) [20] [16]. Hassen and Hirsch concluded that this enhanced dislocation mobility is caused by electrically charged "kinks" that bridge secondary Peierls potentials and migrate in the direction of the dislocation line given adequate thermal energy and applied stress. This theory has been explored by other researchers in the field and is generally accepted [37], [39], [25], [32]. In these high temperature indentation experiments, dislocation rosettes were revealed by etch pits extending to depths of several hundred microns; therefore, hardness reductions at these elevated temperatures are changes to the bulk properties of the crystal as opposed to Gilman's observations which are surface effects. Given a high state of compressive stress under the indenter, and a plastic zone containing electrically charged dislocations, Gilman's measurements are reasonable considering that it is likely electrostatic forces produced by the potential difference between the indenter tip and the surrounding plastic region would increase dislocation mobility. Gilman also measured crack lengths produced by the indentations with and without a 0.2 ma current flowing through the material. No change in the length of the cracks was measured indicating that fracture toughness is not affected by current. This fact perhaps has some connection to Pirouz's observation that fracture toughness is not affected by either n or p doping [39]. Gilman summarizes his conclusions as follows:

1. Electromechanical effects were observed in other semiconductor crystals as well as germanium: Si, InSb, SiC.

2. The effect is reversible over a period of 60 seconds: when current is applied, hardness decreases; when the current is stopped, hardness returns to its normal value.

3. Changing polarity and reversing the direction of current flow has no effect.

4. 60 Hz AC potentials produced identical hardness reductions as DC levels.

5. Voltage levels between 0.1-60 volts produced the same hardness reductions except at small current densities where hardness continuously increased to the zero current value .

6. Identical results were produced with soldered indium contacts and spring loaded copper contacts

7. Similar results were obtained with insulated indenter heads and conducting indenter heads.

In light of Gilman's experiments, Bhaduri and Wang may be seeing the electromechanical effect, however, this is unlikely for number of reasons. First, the plastic zone at the crack tip is small, much smaller than that associated with an indentation, and any potential across the plastic zone is a result of the resistivity of the material along the current path from one side of the crack to the other. This distance is very short giving a small potential difference. Second, the plastic zone is created by far field stresses acting at a singularity, not by a large external body of different potential acting locally in direct contact with the plastic zone. In Gilman's experiment, a small current flowed out of the indenter through the plastic zone and into the substrate. Third, when the potential was applied, region II appeared which is normally associated with stress corrosion cracking indicating the fatigue mechanism is an accelerated corrosion reaction rather than dynamic effects such as dislocation motion. However, Bhaduri and Wang also measured decreases in slope in region III indicating that perhaps dynamic fatigue effects were present; however, without an adequate potential across the plastic zone, the electromechanical effect does not occur. In that case, increased dislocation motion must be strongly related to thermal activation. Just as a crack introduces a singularity in the elastic field, it also produces a singularity in the electric field. Therefore, an accelerated current flow around the crack tip is present which would induce a localized i^2R heating. However, the temperatures generated by such a mechanism would have to be on the order of the DBTT for significant thermally activated dislocation motion to occur; therefore, enhanced dislocation motion caused by local crack tip heating is unlikely. Electrostatic forces across the crack tip opening can probably be ruled out for these forces are small and would act in compression tending to arrest the crack. A plausible explanation could be that an electrically induced surface charge on the crystal ionizes water in the environment releasing hydrogen and hydroxl ions. These ions readily oxidize the surface of the silicon to silicon dioxide thereby increasing the corrosion reaction rate. Ghandhi discusses a room temperature oxidation method for silicon called anodic oxidation which involves applying an electric potential to a silicon wafer in the presence of an electrolyte[15]. In this corrosion/oxidation theory, the surface of the silicon is oxidized to form a thin layer ($\approx 10\AA$) of silicon dioxide which is susceptible to stress corrosion fatigue. Given a pre-crack and corresponding stress intensity of sufficient magnitude, the silicon dioxide is continuously attacked by water vapor in the environment thereby propagating a crack through the layer exposing more silicon. The exposed silicon is in turn oxidized then attacked by moisture propagating the crack further. This process of oxidation and cracking continues until failure. Michalske has developed a molecular model for stress corrosion cracking of silica [31]. In this model, silicon-oxygen-silicon bonds carry the stress of the external loads. These bonds are weakened by hydrogen atoms in the water molecule bonding to the oxygen atoms forming two Si-O-H molecules held together by a weak hydrogen bond. The hydrogen to hydrogen bond then ruptures propagating a crack into the material. Extending this process to silicon, the cleaved hydroxl groups then release hydrogen and form a new monolayer of silica which is in turn attacked by both the released hydrogen and water vapor in the environment thus further weakening the silica structure[15]. Figure 4.5 illustrates the chemical reaction. The oxidation and corrosion mechanism just described still depends on the transport of water molecules to the crack tip; therefore, an enhanced corrosion and oxidation reaction produced by water and hydrogen should predominantly alter the fatigue curve in regions I and II. Region III, however, is insensitive to water content in the environment and reaction rates; therefore, to account for Bhaduri and Wang's measured decrease in slope of region III, another mechanism must be at work. Wiederhorn *et al.* discuses electrostriction which is a likely candidate for Bhaduri and Wang's observation[46]. This effect is observed for silica glass where small decreases in the slope of the da/dt vs. K_1 curve occur in Region III when the dielectric of the surrounding fluid is changed. Electrostrictive forces are generated when the silicon/oxygen bonds in SiO_2 are broken. Positive and negative charges build on the silicon and oxygen atoms, respectively, which produces an electrostatic closing force that in turn retards crack growth.

4.2 Modeling

In the previous section the mechanism of stress corrosion cracking was discussed. Modeling of fatigue involves mathematical expressions relating mea-

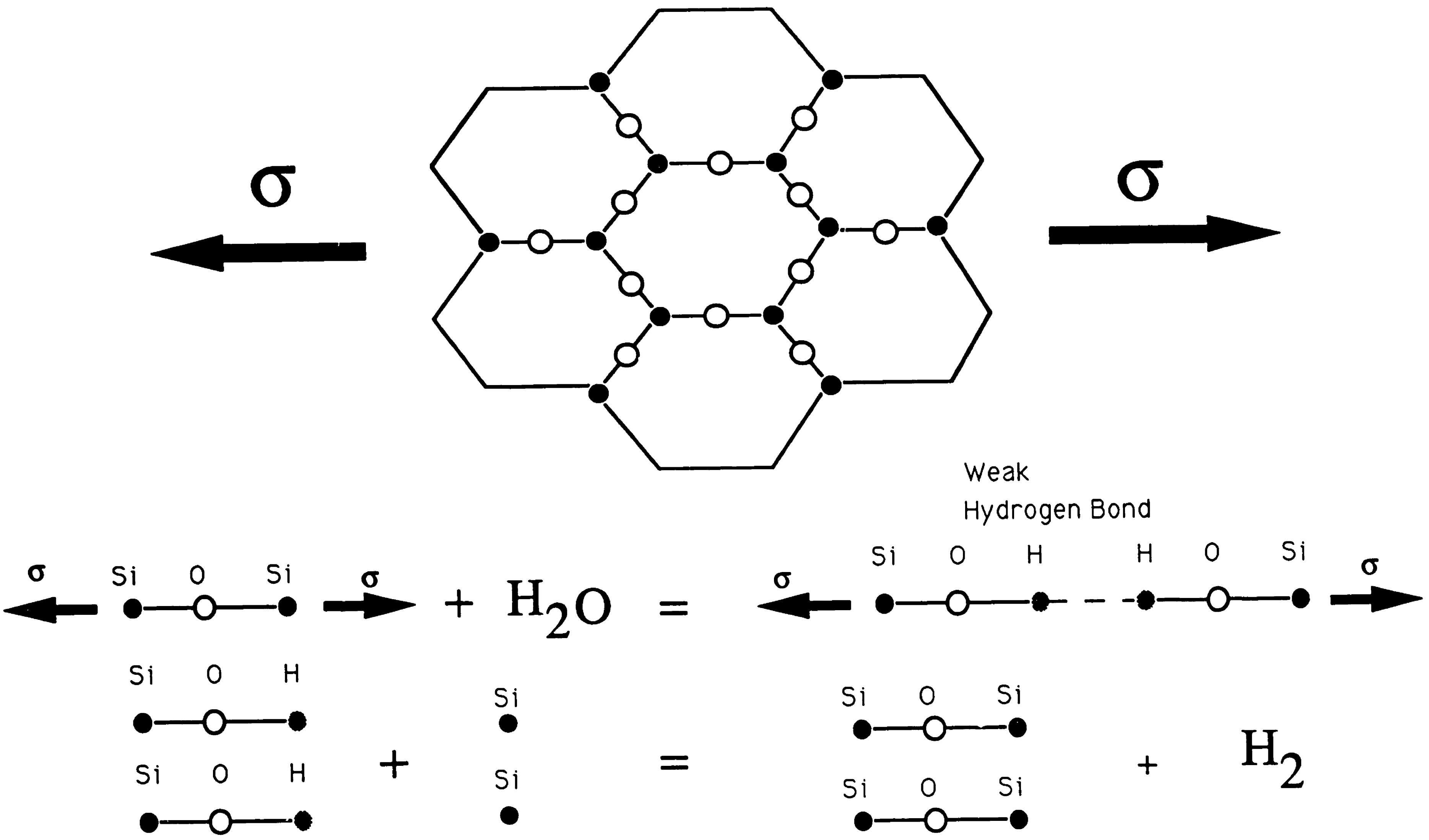


Figure 4.5: Stress Corrosion Mechanism of Si/SiO_2

sured crack tip velocities to a known stress intensity. Fatigue models generally fall into one of two categories: one, a power law model involving empirical constants that lump various effects together such as temperature, humidity, and reaction rate constants; two, a thermally activated model where the energy associated with extending a crack overcomes an initial activation energy. The power law model is written in one of two forms as follows.

$$v = AK_I^n \tag{4.1}$$

or normalized with respect to the fracture toughness

$$v = v_0 \left(\frac{K_I}{K_{Ic}}\right)^n \tag{4.2}$$

where,

$$\begin{aligned} v &= \text{crack tip velocity } \frac{\text{m}}{\text{s}} \\ v_0 &= \text{experimental constant } \frac{\text{m}}{\text{s}} \\ K_I &= \text{stress intensity factor } \frac{\text{MN}}{\text{m}^{3/2}} \\ K_{Ic} &= \text{fracture toughness } \frac{\text{MN}}{\text{m}^{3/2}} \\ n &= \text{experimental constant} \end{aligned}$$

The thermally activated model exponentially relates the crack velocity to a stress intensity, but also separates temperature, humidity, and the activation energy associated with the onset of the corrosion reaction [36].

$$v = v_0 \exp\left(\frac{-E^* + bK_I}{RT}\right) \tag{4.3}$$

where,

$$\begin{aligned} v_0 &= ax_0^f \left(\frac{\text{m}}{\text{s}}\right) \\ a &= \text{experimental constant} \\ x_0 &= \text{partial pressure of water } \left(\frac{\text{N}}{\text{m}^2}\right) \end{aligned}$$

$$
\begin{aligned}
f &= \text{order of the chemical reaction} \\
E^{*} &= \text{zero stress activation energy } \left(\frac{\text{KJ}}{\text{mol}}\right) \\
b &= \text{experimental constant } \left(\frac{\text{m}^{5/2}}{\text{mol}}\right) \\
R &= \text{universal gas constant } \left(\frac{\text{N m}}{\text{mol}^{\circ}\text{K}}\right) \\
T &= \text{temperature } (^{\circ}\text{K})
\end{aligned}
$$

For simplicity, the power law model will be used with the constants v_0, n, and K_{Ic} to be determined experimentally. In Section 2.2 a rotational spring, K_r, of variable stiffness is used to model the crack. An expression for this stiffness and K_I is given by Murakami as follows [34].

$$
\begin{aligned}
K_1 &= k(a)\, M && (4.4) \\
k(a) &= \frac{6}{bh^2}\sqrt{\pi a} f_1 && (4.5) \\
f_1 &= 1.22 - 1.4\alpha + 7.33\alpha^2 - 13.08\alpha^3 + 14\alpha^4 && (4.6) \\
\alpha &= \frac{a}{h} && (4.7)
\end{aligned}
$$

$$
\begin{aligned}
M &= K_r\theta && (4.8) \\
K_r &= \frac{1}{C_\theta(a)} && (4.9) \\
C_\theta(a) &= \frac{24}{Ebh^2}V_1 && (4.10)
\end{aligned}
$$

$$V_1 = 1.458 - 0.304\alpha - 0.924\alpha^2 + 48.34\alpha^3 - 123.5\alpha^4 + 120.5\alpha^5 \qquad (4.11)$$

$$\alpha = \frac{a}{h} \qquad (4.12)$$

where

$$
\begin{aligned}
C_\theta &= \text{compliance of pre-crack at point B in Figure 2.3: } \left(\frac{1}{\text{n m}}\right) \\
a &= \text{crack length: (m)} \\
E &= \text{Young's modulus: } \left(\frac{\text{N}}{\text{m}^2}\right)
\end{aligned}
$$

K_I = stress intensity: $\left(\frac{\mathrm{N}}{\mathrm{m}^{3/2}}\right)$

M = moment applied at the crack site: (N m)

h_b = $h - a$ ligament thickness: (m)

h = beam thickness: (m)

b = beam width: (m)

Murakami's equations for K_I and $C_\theta(a)$ are valid for for $\alpha \leq 0.6$ and $\alpha \leq 0.7$, respectively. Figure 4.6 shows the geometry of the pre-cracked bar with the applied moment. The thickness of the ligament is given by h_b. For deep pre cracks where $\alpha > 0.6$, the following equation for the stress intensity is accurate to within 1% [38].

$$K_I = \frac{4M}{bh_b^{3/2}} \tag{4.13}$$

The compliance of the deep cracked bar can be computed mathematically by using the relation shown below [24].

$$\frac{dC}{da} = \frac{2b}{E}\left(\frac{K_I}{M}\right)^2 \tag{4.14}$$

Substituting Equation 4.13 into Equation 4.14, and integrating from a_1 to a_2

$$\int_{a_1}^{a_2} dC = \frac{2b}{E}\int_{a_1}^{a_2}\left(\frac{K_I}{M}\right)^2 da \tag{4.15}$$

gives the following expression.

$$C(a_2) - C(a_1) = \frac{16}{Eb}\left[\frac{1}{(h-a_2)^2} - \frac{1}{(h-a_1)^2}\right] \tag{4.16}$$

By definition

$$C(a_1) = \frac{16}{Eb(h-a_1)^2} \tag{4.17}$$

and,

$$C(a_2) = \frac{16}{Eb(h-a_2)^2} \tag{4.18}$$

or more generally,

$$C(a) = \frac{16}{Eb(h_b)^2} \tag{4.19}$$

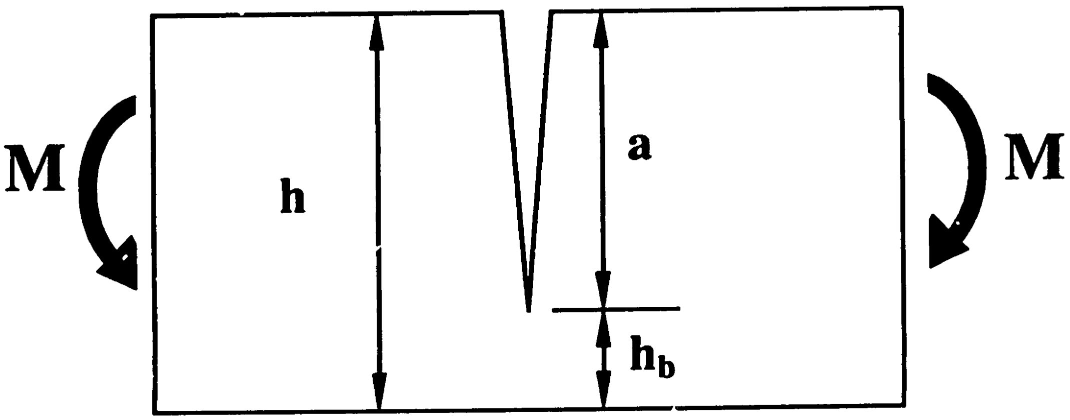


Figure 4.6: Pre Cracked Bar

Table 4.1 is a comparison of Murakami's equations for compliance and stress intensity to those of the deep cracked beam. α is given in the left column; the center column is the ratio of the stress intensity computed using the deep cracked bar equation to Murakami's; and the right column is the ratio of the compliances. As predicted, the stress intensities are nearly equal near $\alpha = 0.6$, and the compliances become close near $\alpha = 0.7$. In the FTEST2.PAS design program, these limits of accuracy are used to select either the deep cracked bar equations, or Murakami's equations.

The frequency can now be determined for any precrack of length, a, and any frequency shift given an incremental extension of the crack length, δa. The design program, FTEST2.PAS, incorporates this computation. Using the T38 series device with an initial crack length of $2\mu m$, and an incremental extension of $10nm$, the frequency shift is approximately 4 Hz. This shift translates to a frequency sensitivity of 2.5 nm/Hz. Because the con-

α	$\frac{K_{I\ Deep\ Crack}}{K_{I\ Murakami}}$	$\frac{C_{\theta\ Deep\ Crack}}{C_{\theta\ Murakami}}$
0.400	1.019	0.786
0.450	1.012	0.817
0.500	1.007	0.843
0.550	1.004	0.862
0.600	1.005	0.877
0.650	1.017	0.895
0.700	1.049	0.929
0.750	1.118	1.004
0.800	1.260	1.168

Table 4.1: Comparison of Deep Cracked Bar to Murakami

trol system and sensing electronics are capable of maintaining the device on resonance to within ±1 Hz, a 2.5 nanometer crack extension is detectable. This resolution is possible because of the small size of the device. In general, the stiffness of a cantilever scales in direct proportion to the length while the mass scales by the length cubed. Therefore, as a structure is shrunk in size, the inertia forces become relatively small in comparison to the elastic restoring forces of the material. Minute changes in the stiffness of the structure now significantly affect the frequency.

Next, the time necessary to extend the crack an amount δa will be derived. The stress intensity factor, K_I, shown in Equation 4.2 is composed of two parts: a constant static part, and a cyclic dynamic part. As shown in Section 2.2.5, the nonlinearity of the stiffness of the pre-crack results in a constant negative bias in displacement superposed with a cyclic component; therefore, the moment acting at the crack site is written as follows.

$$M_0(t) = M_0\left(\frac{1}{\pi} + \frac{1}{2}\sin(\omega t)\right) \tag{4.20}$$

Rewriting Equation 4.2 with the above expression for the moment gives the following.

$$v = v_0\left(\frac{K_{I0}}{K_{Ic}}\right)^n\left(\frac{1}{\pi} + \frac{1}{2}\sin(\omega t)\right)^n \tag{4.21}$$

Where,

$$K_{I0} = k(a)M_0 \tag{4.22}$$

Following the method outlined by Evans, an average crack tip velocity, v_{avg}, is defined for one cycle [9].

$$v_{avg} = \frac{1}{T}\int_0^T v(t)\,dt \tag{4.23}$$

Where,

$$T = \frac{2\pi}{\omega_n} = \text{period of one cycle: (s)}$$

Substituting Equation 4.21 into Equation 4.23 gives the following expression.

$$v_{avg} = v_0\left(\frac{K_{I0}}{K_{Ic}}\right)^n \left(\frac{1}{\pi}\right)^n \frac{1}{T}\int_0^T \left(1+\frac{\pi}{2}\sin(\omega_n t)\right)^n dt \tag{4.24}$$

Defining

$$g = \frac{1}{\pi^n T}\int_0^T \left(1+\frac{\pi}{2}\sin(\omega_n t)\right)^n dt \tag{4.25}$$

The solution gives the following expression [9].

$$g = \frac{1}{\pi^n}\sum_{\ell=0}^{\frac{n}{2}} \frac{n!}{(n-2\ell)!(\ell!)^2}\left(\frac{\pi}{4}\right)^{2\ell} \tag{4.26}$$

Therefore, Equation 4.24 is rewritten with the correction g to account for crack closure.

$$v_{avg} = v_0 g\left(\frac{K_{I0}}{K_{Ic}}\right)^n \tag{4.27}$$

Equation 4.27 is a quasi static fatigue model where the normalized velocity, v_0, is attenuated by a factor g determined by computing the average extension of the crack tip for one cycle. For the T38 devices, g is on the order of 0.01. With the average velocity of the crack tip determined, the time to failure can be computed by differentiating Equation 4.22 with respect to time and using the chain rule to substitute in Equation 4.27.

$$\frac{dK_{I0}}{dt} = \left(\frac{dK_{I0}}{da}\right)\frac{da}{dt} \tag{4.28}$$

$$\frac{dK_{I0}}{dt} = \left(\frac{dK_{I0}}{da}\right)v_{avg} \tag{4.29}$$

$$\frac{dK_{I0}}{dt} = \left(\frac{dK_{I0}}{da}\right) v_0 g \left(\frac{K_{I0}}{K_{Ic}}\right)^n \tag{4.30}$$

Separating variables and integrating gives the time required for the stress intensity factor to increase from an initial value, K_{I0i}, to a final value K_{I0f}.

$$\int_{K_{I0i}}^{K_{I0f}} \frac{dK_{I0}}{K_{I0}^n} = \left(\frac{dK_{I0}}{da}\right) \frac{v_0 g}{K_{Ic}^n} \int_0^\tau dt \tag{4.31}$$

In the above equation, $\frac{dK_{I0}}{da}$, is assumed constant over the limits of integration. The design program, FTEST2.PAS computes this number directly given a pre-crack of known depth. Integrating Equation 4.31gives the following equation.

$$\left.\frac{K_{I0}^{1-n}}{1-n}\right|_{K_{I0i}}^{K_{I0f}} = \left(\frac{dK_{I0}}{da}\right) \frac{v_0 g}{K_{Ic}^n} \tau \tag{4.32}$$

Defining

$$\gamma = \frac{K_{I0i}}{K_{I0f}} \tag{4.33}$$

and solving Equation 4.32 for τ gives the following equation.

$$\tau = \frac{K_{I0i}(1-\gamma^{n-1})}{(n-1)v_0 g \left(\frac{K_{I0}}{K_{Ic}}\right)^n \frac{dK_{I0}}{da}} \tag{4.34}$$

$$\tau = \frac{K_{I0i}(1-\gamma^{n-1})}{(n-1)v_{avg} \frac{dK_{I0}}{da}} \tag{4.35}$$

To compute the time to failure, K_{I0f} is replaced with a stress intensity near the fracture toughness of silicon: $K_{I0f} \approx 95\% K_{Ic}$ at fracture. The exact value of the fracture toughness is not used because as the stress intensity approaches the fracture toughness of the material the velocity of crack propagation approaches the elastic wave speed of the material and Equation 4.35 is no longer valid.

Now that equations have been derived which relate compliance changes and time to incremental extensions of crack length, the FTEST2.PAS design program can be used to simulate the anticipated frequency shifts with time assuming that crack growth obeys the power law model previously described.

Chapter 5

Crack Initiation Methods

In the previous sections, it has been assumed that a pre-crack resembling that shown in Figure 4.6 exists in the structure. Some means of introducing a geometrically similar flaw, or pre-crack, that is both repeatable and controlled is necessary. The controlled flaw must meet three basic criteria for success: first, the flaw must be small compared to the scale of the structure so that a K_I dominant field is generated; second, can be analytically characterized by a stress intensity; and third, must not cause uncontrolled breakage when the device is etched or wire bonded. Numerous alternatives are possible. Those with the greatest chance of success were attempted, and are explained in the following sections.

5.1 Micromachined V-notch

This method involves the fabrication process where KOH, a highly anisotropic etch, is used to etch downward along the $< 111 >$ crystallographic planes of the silicon until the two planes intersect forming a controlled V-notch in the top face of the beam. The radius of curvature at the root of the V must be small (less than $0.1\mu m$) so that a sizable stress concentration is developed which will exceed the cohesive strength of the crystal given a displacement that is within the allowable limits of travel of the end plate. This method was attempted with little success. The first attempt involved etching the V-notch as one of the last processing steps after the beam was diffused with boron. Although boron acts as an etch stop, anisotropic etching should still cut along the $< 111 >$ planes but at a reduced rate. However, the rate was too low, making this method unattractive. The next attempt involved cutting the notch first in undiffused silicon then diffusing with boron. Although a notch was made, subsequent processing steps involving oxidation leeched boron from the outermost layer of silicon so that during final etching the sharp notch was significantly rounded as shown in Figure 5.1. The last attempt involved cutting the notch first then depositing a silicon nitride layer over the notch to protect it fiom oxidation. However, processing difficulties were encountered with gold alloying into the silicon which destroyed the wafer. Given the limitations on available manpower in the CSDL microfabrication laboratory, and the processing difficulties encountered, the V-notch was abandoned. Conceptually, fabricating beams with a V-notch is possible; however, the process development time and trial runs required for success with this approach require a level of manpower that is not possible to provide. Furthermore, it is not a certainty that a stress concentration, even if it is large, will initiate subcritical cracking in silicon. Han and Tomozawa show that "blunt" cracks in silica glass do not exhibit slow crack growth, instead, crack initiation and propagation occur only when the applied stress is close to the failure stress of the material [17]. This failure stress is the stress necessary to fracture the material assuming the crack were "sharp". At this point, other methods for introducing a pre-crack were considered more viable than using processing techniques.

5.2 Scribing

Scribing was considered a viable alternative for introducing a pre-crack, or stress concentration, into the beam with the principle disadvantage being repeatability and crack characterizability. Tungsten carbide scribe probes were used with a probe station to generate a uniform scribe line across the diffused section of the beam. However, sufficient depth penetration of the scribe probe was not achieved. Scribed devices were etched and deflected on the probe station $20 - 30\mu m$ without breaking, indicating the lack of an adequate stress concentration. The smallest tip diameter available for scribe probes is approximately $2.5\mu m$ which is relatively large compared to the size of the structure so that the scribe lines must penetrate the surface of the silicon nearly to the depth of the beam. However, the probes break when forces necessary for this level of penetration are applied. Figure 5.2 is a SEM photograph of a scribe line made with a scribe probe.

Misra and Finnie [33] studied in detail the scribing and subsequent fracturing of silicon wafers using both spherical points and diamond points. Their results indicate that a sharp scribe, such as a diamond, produces both lateral and median cracks in the substrate while a spherical scribe results in partial cone cracks extending from the surface downwards along the sides of the scribe line. Therefore, a diamond scribe was fitted to the end of a modified probe manipulator. This method is difficult because the diamond must be large enough that it can be glued to the end of a metal shank. Since the diamond is cleaved along its $< 111 >$ planes, a rather large shank diameter is necessary. The large diameter of the shank completely obscures the silicon chip on the microscope table from view. Therefore, the scribe must be positioned blind within several microns. This level of positioning accuracy would be difficult to achieve. Test scribe lines showed that significant surface damage was introduced and the scribe line was at best crude. The scribe appeared to skip along the surface, rather than dig in and cut, which introduced significant chipping and extraneous surface damage. Controlling depth, load, and positioning would be difficult without precise load and displacement servo control. Otherwise, this method has little hope of working in a satisfactory and repeatable way.

5.3 Nanoindenting

The nanoindenter is a servo controlled indenting machine used to evaluate the hardness and modulus of thin films several microns thick. The nanoindenter can apply loads up to 12 grams with a force resolution of $2-3\mu N$ and a displacement accuracy of $2-3\,nm$. Nanoindenting is a reliable method for introducing repeatable controlled cracks into the beam. Surface damage is minimal, and uniform crack planes extend radially outward from the indentation site. Furthermore, the nanoindenter can be used to measure hardness, Young's modulus, and force displacement curves of cracked or uncracked cantilevers. The principle drawback of the nanoindenter is the complex crack geometry created by the angles formed by the radial cracks with respect to the sides of the beam; furthermore, multiple indentations will be required to fully extend a pre-crack across the width of the beam which can result

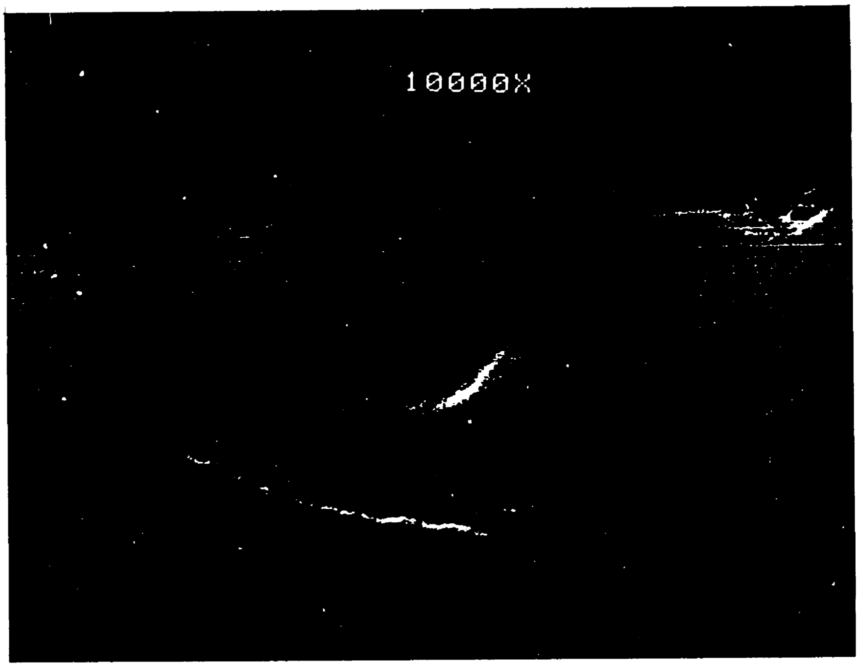


Figure 5.1: Micromachined V-notch

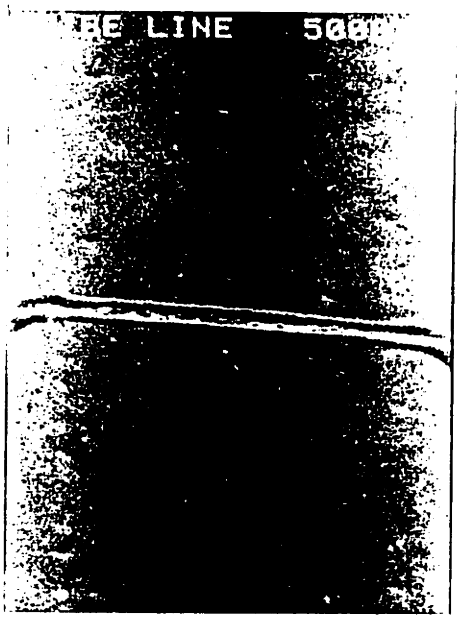


Figure 5.2: Scribe Line

in a nonplanar crack face depending on the orientation of the indenter and the indentation pattern. Although the crack geometry is potentially complex, accurate results may be obtained with the simple closed form equations for the stress intensity and strain energy release rate presented in the previous sections if the cantilever is fully pre-cracked across the width to a uniform depth. Similar geometric difficulties are encountered when considering anisotropic bodies. Kanninen states that the effects of nonplanar crack extension in anisotropic bodies are usually ignored in deriving expressions for the strain energy release rate because of the mathematical difficulties involved [24]. Therefore, it is not uncommon to use simplified equations based on the assumptions of material isotropy and planar crack extension to model complex fracture behavior. The principle criteria in this investigation is that an adequate stress intensity is developed which will either initiate fatigue or fracture the material given the limited travel under the bridge electrode, and that a significant frequency drop result from the pre-crack followed by a measurable frequency shift if fatigue occurs.

A depth penetration on the order of $400 nm$ in the boron diffused silicon is sufficient to generate radial cracks which extend outward from the vertices of an equilateral triangle formed by the three sided pyramidal diamond indenter tip. Figure 5.3 is a single indentation into the surface of an unetched T38 cantilever approximately $10 \mu m$ from the base. The indentation depth was approximately $460 nm$ and produced radial cracks approximately $2.7 \mu m$ long measured from the centroid of the triangle to the tip of the crack. The cantilever was subsequently etched then fractured at the indentation site by deflecting the tip of the end plate downward on the probe station. Figure 5.4 shows the cross section of the fracture surface and the nature of the pre-crack geometry as it extends into the beam. The three radial cracks form quarter ellipses joined along a common line that emanates from the centroid of the triangular indent downward at right angles to the surface. The length of this line is approximately 75% the length of the pre-cracks as measured along the surface of the cantilever as shown in Figure 5.3. The pre-cracked surface is clearly distinguishable from the remainder of the surface so that a precise measurement of the pre-crack depth can be made with a SEM obviating the need for any form of crack marking technique. The extent of the radial cracks is substantially increased by the additional tensile stress induced by the boron diffusion. This diffusion induced stress is removed when the device is etched and the structure is released from the substrate. The load versus displacement curve associated with this indentation is shown in Figure 5.5.

Figure 5.6 shows the indenting process in more detail. As the indenter penetrates the surface of the silicon, the load correspondingly increases to a maximum specified by the user, or to a predetermined depth, h_{total}, also specified by the user. The indentations performed in this experiment are load controlled so that the applied load, P, is specified. The total depth is the sum of reversible elastic and irreversible plastic deformation [28].

$$h_{total} = h_p + h_e \tag{5.1}$$

The plastic depth, h_p, is the depth the indenter head actually penetrates into the sample while the elastic depth, h_e, is the depth the surface is deflected downward. As the indenter tip is withdrawn, the material expands to partially fill the void produced by the indenter. The initial relaxation is elastic, and if it is assumed that the surface area of the indenter tip in contact with the sample remains constant during this relaxation interval, the projected area on the surface of the sample also remains constant. Sneddon has developed a stiffness relation for a rigid flat cylindrical punch depressing a flat elastic surface as follows [40].

$$P = \frac{2E}{1-\nu^2} A h_e \tag{5.2}$$

Where,

$$\begin{aligned} P &= \text{applied load: (N)} \\ \nu &= \text{Poisson's ratio} \\ A &= \pi a^2 \text{ cross sectional area of punch: (m)}^2 \\ a &= \text{radius of punch: (m)} \\ h_e &= \text{elastic depth: (m)} \end{aligned}$$

The nanoindenter can be modeled as a flat punch by equating the projected area of the indenter to A in Equation 5.3 [7]. Figure 5.7 shows the ideal geometry of the nanoindenter, and the projected area as a function of the plastic depth, h_p. Sneddon assumed the punch is rigid; however, the diamond tip of the indenter has a finite stiffness which is accounted for by summing the stiffness of the diamond in series with the sample. Rewriting Equation 5.2

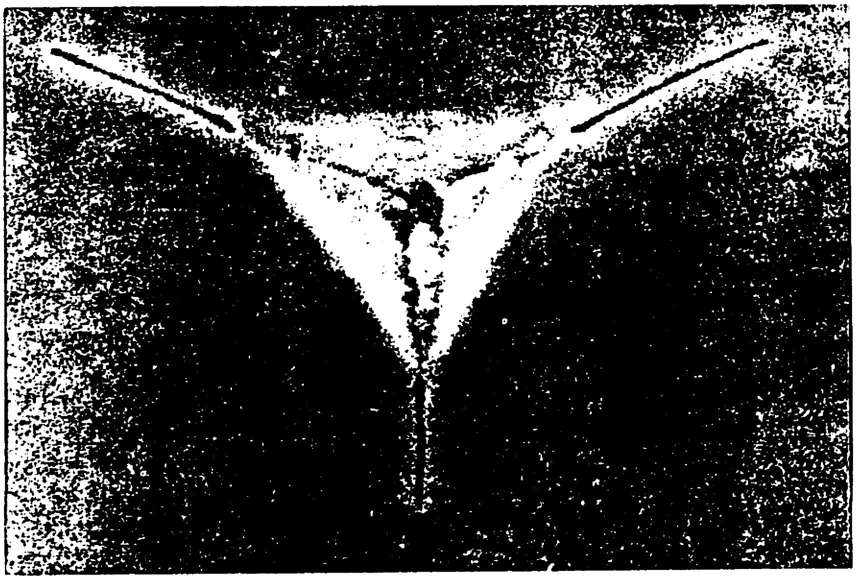


Figure 5.3: 500 nm Nanoindentation (Top View)

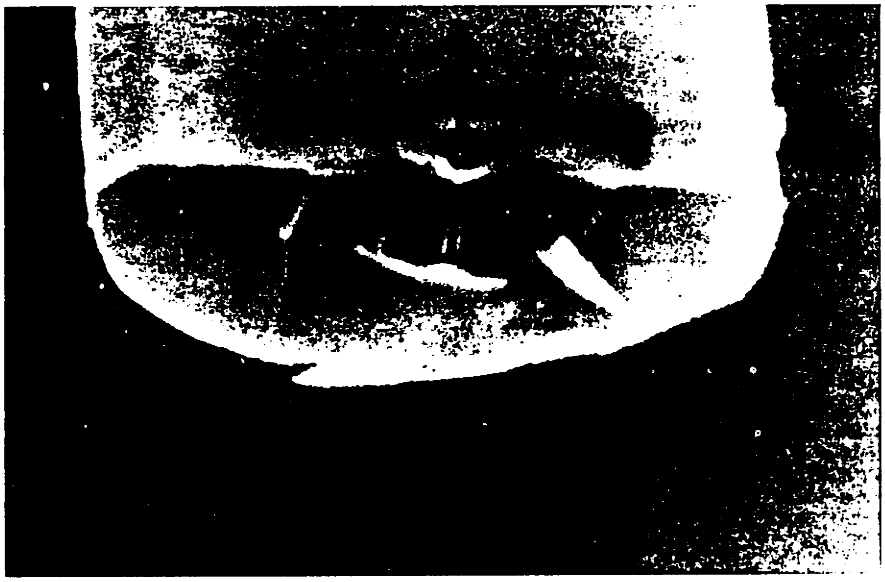


Figure 5.4: 500 nm Nanoindentation (Cross Sectional View)

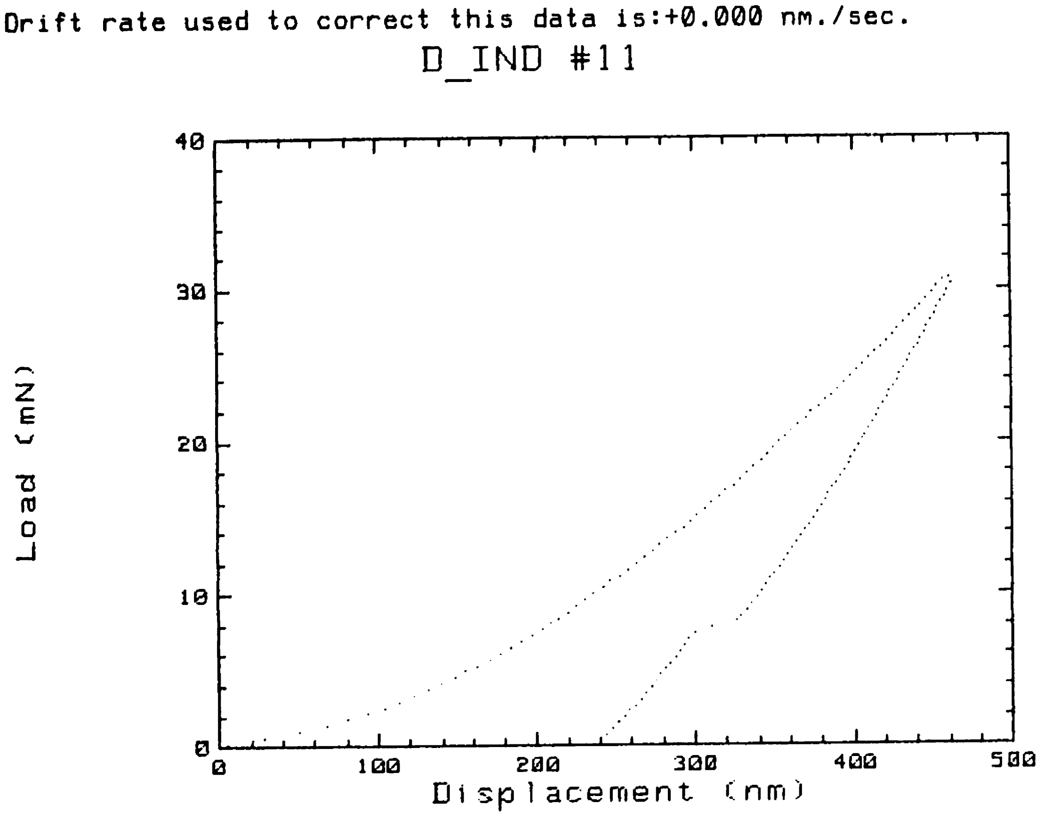


Figure 5.5: Nanoindentation Curve (Load vs. Displacement)

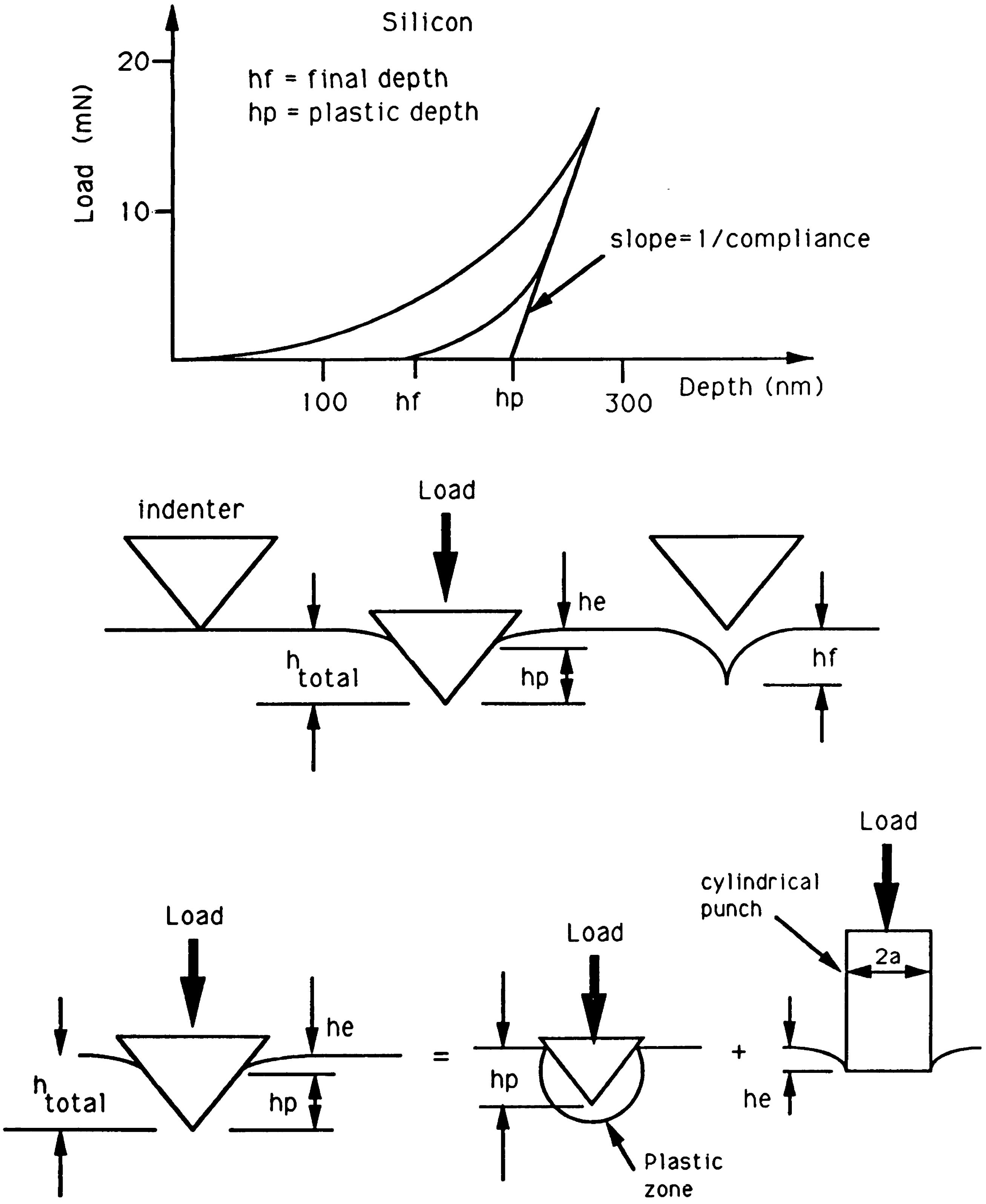


Figure 5.6: Typical Nanondentation Curve

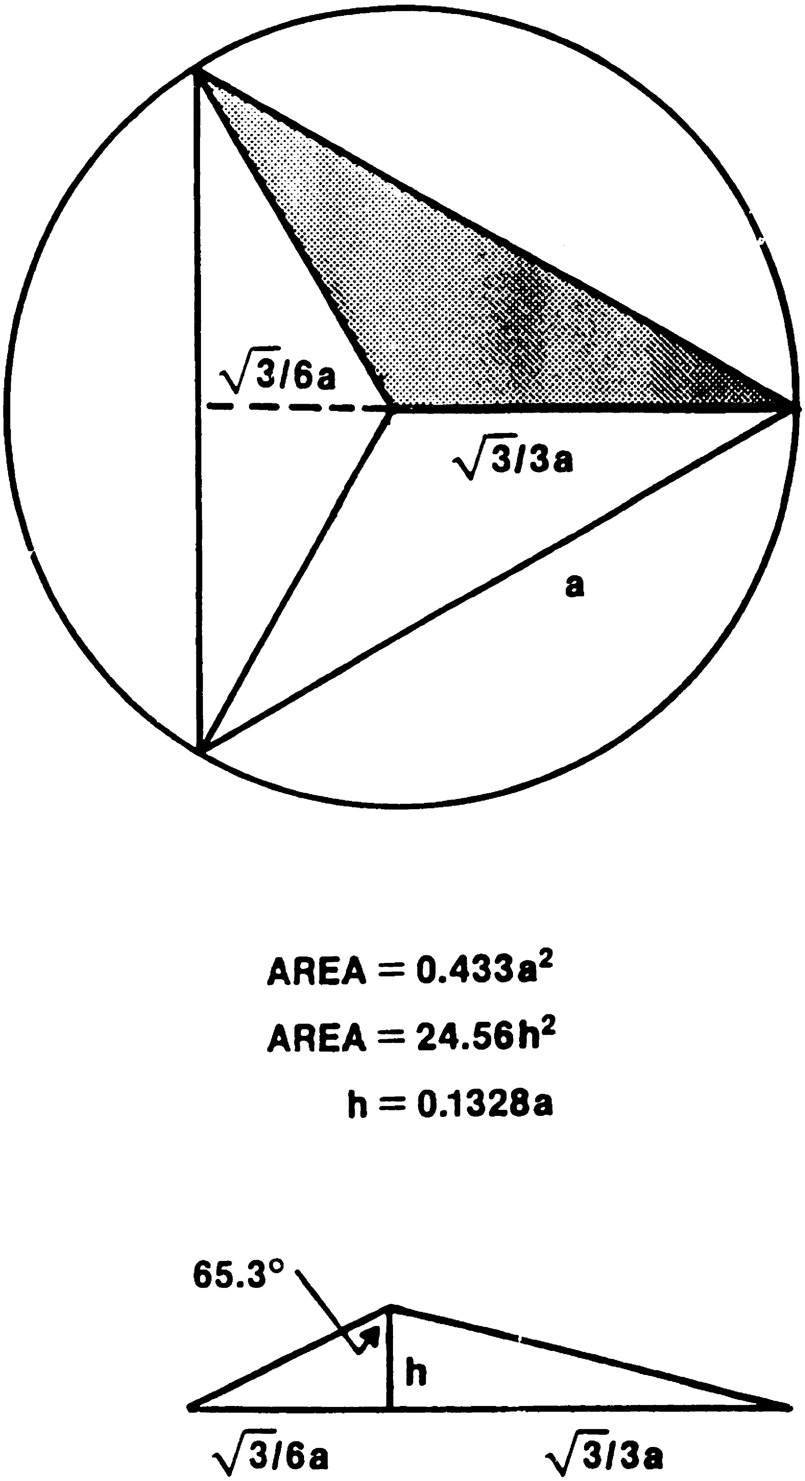


Figure 5.7: Ideal Nanoindenter Geometry

gives the following equation.

$$P = 2E_r h_p \left(\frac{24.56}{\pi}\right)^{-1} h_e \tag{5.3}$$

Where,

$$\begin{aligned} E_r &= \left(\frac{1-\nu^2}{E} + \frac{1-\nu_0^2}{E_0}\right)^{-1} \\ E &= \text{Young's modulus of the sample} \\ \nu &= \text{Poisson's ration of the sample} \\ E_0 &= \text{Young's modulus of the diamond indenter} \\ \nu_0 &= \text{Poisson's ratio of the diamond indenter} \end{aligned}$$

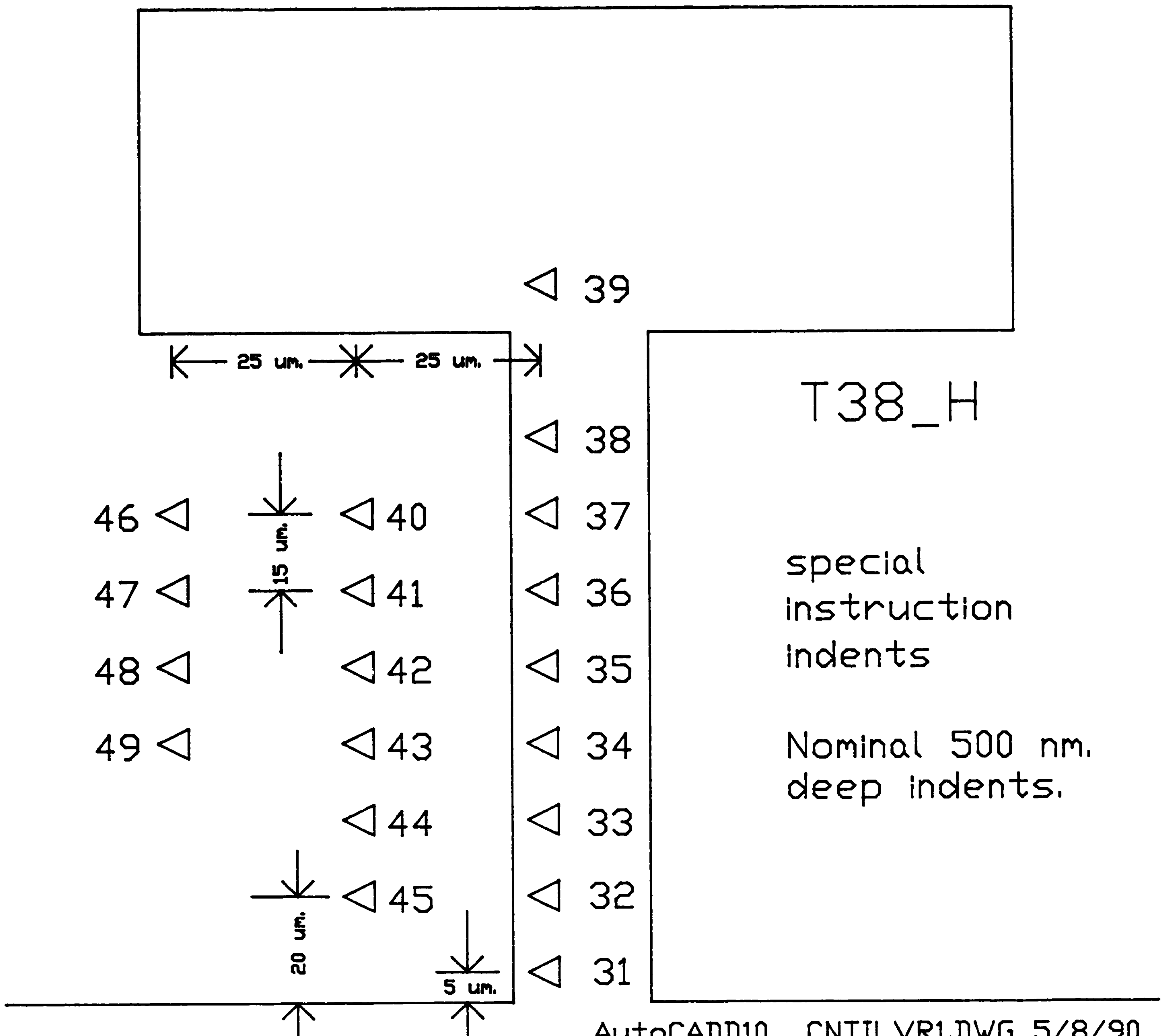
Differentiating Equation 5.3 with respect to h_e gives the slope of the tangent shown in Figure 5.6.

$$\frac{dP}{dh_e} = 2E_r h_p \left(\frac{24.56}{\pi}\right)^{1/2} \tag{5.4}$$

Equation 5.4 can be used to compute Young's modulus [7]. As mentioned, the nanoindenter is useful for several purposes: pre-cracking, obtaining hardness and the modulus of elasticity, estimating the fracture toughness, compliance testing, and break testing. In the following subsections nanoindenting results for hardness, Young's modulus, fracture toughness, and compliance of an etched device are presented.

5.3.1 Hardness and Young's Modulus for Silicon

Figure 5.8 is a diagram of showing two arrays of 500 nm deep indentations. Indentations 31 through 39 were performed in the boron diffused region of an unetched T38 cantilever, and indentations 45 through 49 were in the undiffused region. The purpose of this experiment was to determine if any differences exist in Young's modulus or hardness for the doped versus undoped regions. The raw data is contained in Appendix C. Table 5.1 compares the average values of hardness and Young's modulus for the doped and undoped regions. Young's modulus agrees to within 0.2% for both the diffused and undiffused regions; however, the hardness in the diffused region is 5% greater. The higher hardness in the p type diffused silicon versus the n type substrate


T38_H
special
instruction
indents
Nominal 500 nm.
deep indents.
39
38
37
36
35
34
33
32
31
40
41
42
43
44
45
46
47
48
49
25 um.
25 um.
15 um.
20 um.
5 um.
AutoCADD10 CNTILVR1.DWG 5/8/90

	Doped	Undoped
h (Pa)	1.62×10^{10}	1.54×10^{10}
E (Pa)	1.59×10^{11}	1.59×10^{11}

Table 5.1: Hardness and Young's Modulus for Si

is consistent with Pirouz's hardness measurements where dislocation motion was found to be greater in n type silicon than doped p type[39]. Pirouz's measured hardness values agree within 10% of the nanoindenter values; however, a direct comparison is not possible because the lowest temperature Pirouz measured hardness was $200°C$ whereas the nanoindentation was at room temperature.

5.3.2 Fracture Toughness of Silicon Using the Nanoindenter

An estimate of the fracture toughness can be made using Figures 5.3, 5.4, and 5.5. The loading and unloading cycle of the indentation was performed at a constant loading rate of $800\frac{\mu N}{s}$; therefore, crack initiation would indicated by a horizontal jog in the curve. Referring to Figure 5.5, the loading curve appears to be free of jogs while the unloading curve shows a significant step discontinuity in displacement indicating crack formation and significant extension. Two types of crack systems develop from an indentation: median or radial cracks which develop on loading and are extended on unloading, and lateral cracks which develop on unloading only. The plane of the median/radial crack system is perpendicular to the surface while the lateral system is parallel [30]. Figure 5.4 shows the system to be the median/radial system; therefore, crack initiation occurred during loading, but was small and was not measured by the nanoindenter, and a large measurable crack extension occurred on unloading. Lawn *et al.* has shown that the residual stresses on unloading provide the primary driving force for the median/radial system to fully develop into the half penny configuration [26]. The tensile stress resulting from the boron diffusion also contributes to the driving force for crack extension, and is accounted for by superposing the stress intensity of a half penny subject to a uniform far field stress with that of the residual stress. These stress intensities are summed and equated to the fracture toughness.

$$K_{Ic} = \chi_r \frac{P^*}{c^{3/2}} + \sigma_0 \sqrt{\pi c}\psi \tag{5.5}$$

Where,

$$\begin{aligned}
\chi_r &= 0.069 \\
P^* &= 31\text{mN} = \text{maximum load applied by the indenter} \\
c &= 2.7\mu m = \text{equilibrium crack length} \\
\sigma_0 &= 1.957 \times 10^8 \frac{\text{N}}{\text{m}^2} = \text{ diffusion stress} \\
\psi &= 0.66
\end{aligned}$$

The constant, χ_r, was experimentally determined by Lawn and Evans through a series of indentation experiments on silicon [26]. The quantity, ψ, is a geometric factor dependent on the half width of the elliptical crack, depth of the crack, and beam thickness [38]. The stress due to the boron diffusion is calculated as follows [27].

$$\sigma_0 = \frac{\beta E C_s}{(1-\nu)C_{Si}} \tag{5.6}$$

$$\beta = \left[1 - \left(\frac{r_B}{R_{Si}}\right)^3\right] \tag{5.7}$$

Where,

$$\begin{aligned}
E &= 1.59 \times 10^{11} \frac{\text{N}}{\text{m}^2} = \text{ Young's modulus} \\
C_s &= 1 \times 10^{20}/\text{cm}^3 = \text{ surface concentration of boron} \\
C_{Si} &= 5 \times 10^{22}/\text{cm}^3 = \text{ concentration of silicon atoms} \\
\beta &= \text{Lattice contraction coefficient} \\
r_B &= 0.98\AA = \text{ Atomic radius of boron atom} \\
r_{Si} &= 1.17\AA = \text{ Atomic radius of silicon atom} \\
\nu &= 0.33 = \text{ Poisson's ratio of silicon}
\end{aligned}$$

With the diffusion stress known, K_{Ic} is computed as $8.58 \times 10^5 \frac{\text{N}}{\text{m}^{3/2}}$ which is close to the value of $8.2 \times 10^5 \frac{\text{N}}{\text{m}^{3/2}}$ reported by Chen and Leipold [5]. Equation 5.5 is used in the design program FTEST2.PAS to estimate the depth of the pre-crack generated by the nanoindenter.

5.3.3 Compliance Test

The nanoindenter is a load and displacement sensing instrument; therefore, it is possible to use the nanoindenter to obtain a measurement of thc stiffness of a released cantilever. A finite element analysis can then be used to compare measured versus calculated values of stiffness. The stiffness of the cantilever computed in the FTEST2.PAS design program is based on the conservative assumption that the cross section of the beam is an ellipse. In fact, the true cross section is more similar to a half ellipse with rounded corners. Therefore, to account for this cross section accurately, the cantilever was modeled with ABAQUS using 8 node brick elements distributed along a parabola defined by the total width and maximum thickness. The value for Young's modulus used in ABAQUS was 1.59×10^{11}N/m^2 which corresponds to the average value computed by the nanoindenter in Table 5.1. Poisson's ratio was assumed to be 0.33 [47].

A T38 series device was etched, then statically deflected by the nanoindenter $25\mu m$ from the fixed end. The sample was subsequently potted in Lucite, then polished to obtain the true cross section. Figure 5.9 is a top view of the cantilever that was deflected by the nanoindenter, and Figure 5.10 is the cross section. Figure 5.11 is the cross section of the finite element model of the beam, and Figure 5.12 is the mesh plot of the cantilever. Figure 5.13 is a side view of the deflected cantilever mesh. The total width of the beam and thickness of the beam, as measured in Figure 5.10, is $10\mu m$ and $3.0\mu m$, respectively. Figure 5.14 is the load versus displacement plot generated by the nanoindenter as the beam was deflected. The slope of this curve is the stiffness which is calculated by the nanoindenter to be 452.68 N/m^2. The calculated stiffness was determined from ABAQUS by applying a load in the z direction at the center end node of the cantilever. A load of 0.5 mN produced a tip displacement of 1.214 μm; therefore, the stiffness determined from ABAQUS is 411.86 N/m^2. The two numbers agree within 10%. In the finite element analysis, it was assumed that silicon is isotropic and homogeneous. However, single crystal silicon is anisotropic and a matrix of elastic constants are used to relate the components of stress to strain. Throughout the dynamic and fatigue modeling of the device, it has been assumed the silicon can be treated as an isotropic homogeneous material. This assumption of homogeneity is valid to the extent that significant dynamic coupling, through the stiffness matrix, was not observed with the spectrum analysis of the generation 1 device. The spectrum analysis indicates that the device acts as a single degree of freedom spring, mass, damper system at the first mode resonance. Spurious torsional modes, or out of plane bending modes, were not apparent. Furthermore, given the geometry of a particular device, the frequency and amplitude predicted by the dynamic model correlates closely to the measured values. Therefore, it is likely the principle error in the compliance measurement is the ability of the nanoindenter to accurately position the indenter head precisely $25\mu m$ from the base. At best, this positioning accuracy is $\pm 1\mu m$. This test demonstrates the feasibility of break testing pre-cracked devices with the nanoindenter.

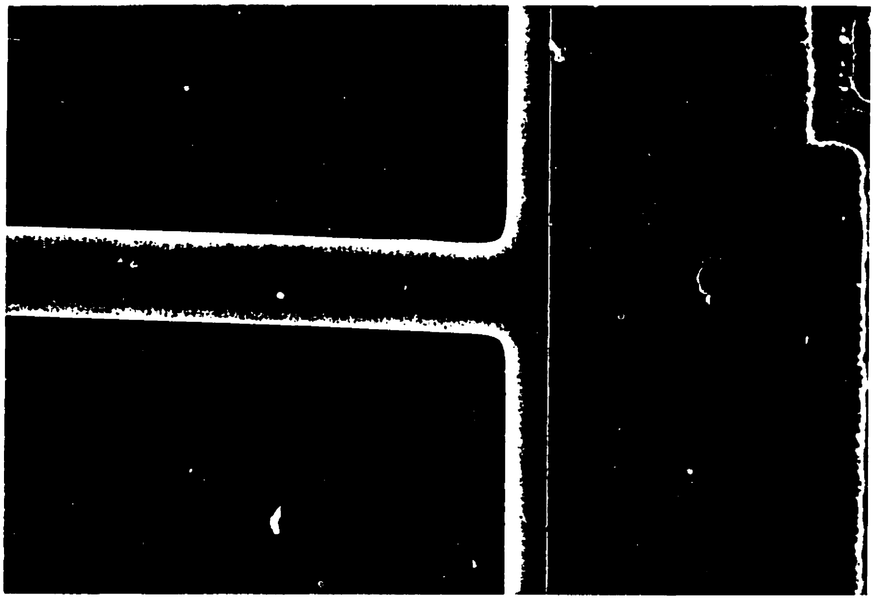


Figure 5.9: Top View of T38 Cantilever

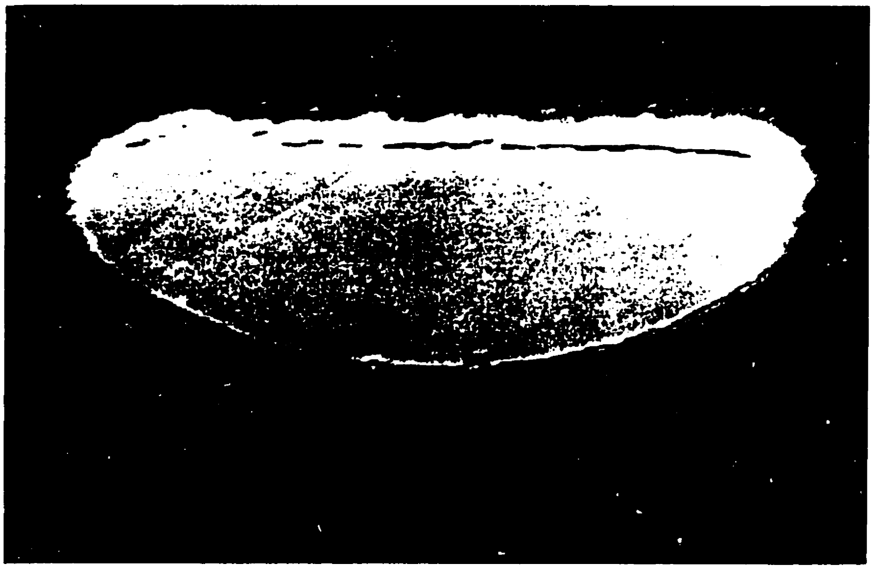


Figure 5.10: Cross Section of T38 Cantilever

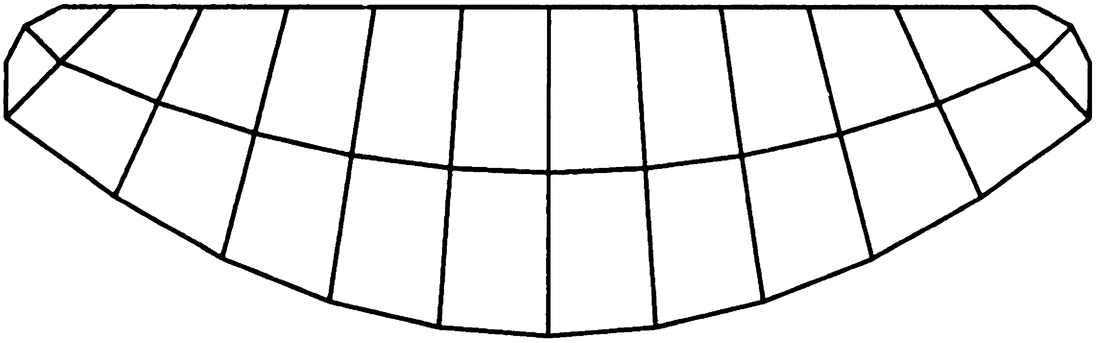


Figure 5.11: Mesh Plot of T38 Cross Section

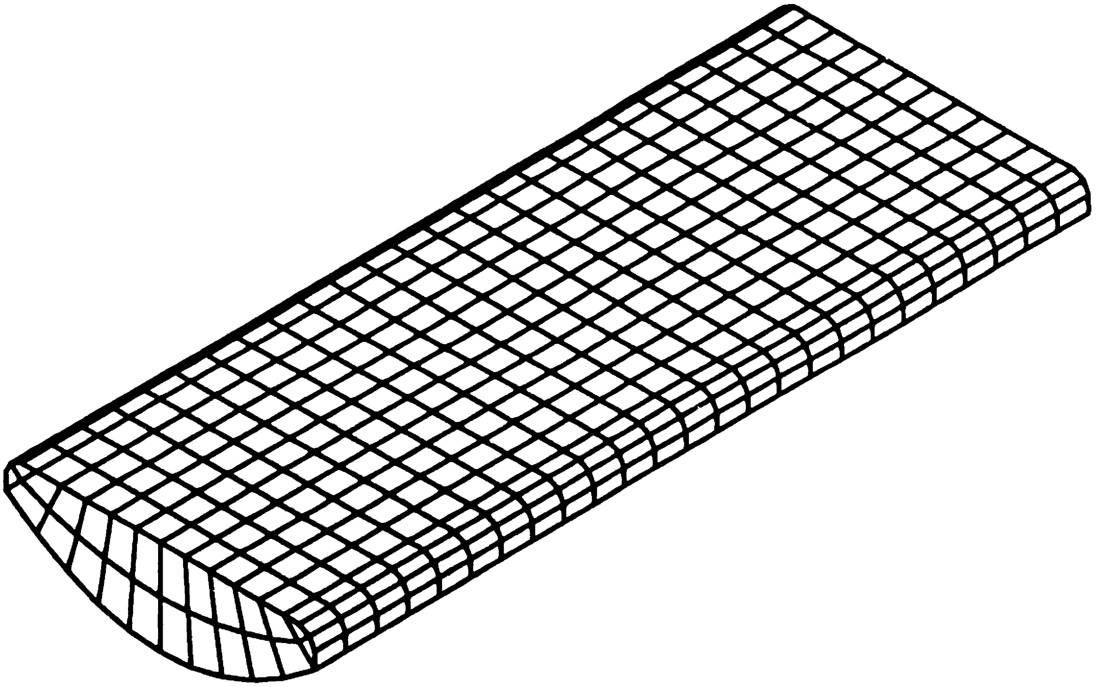


Figure 5.12: Mesh Plot of T38 Cantilever

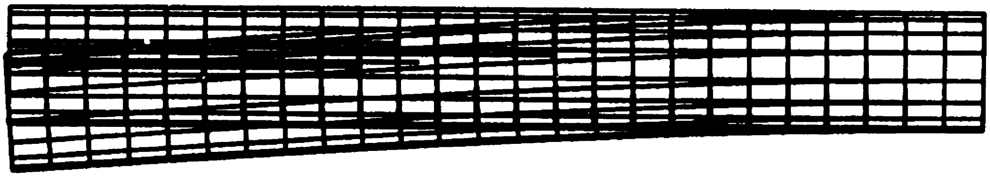


Figure 5.13: Displaced Mesh of T38 Cantilever

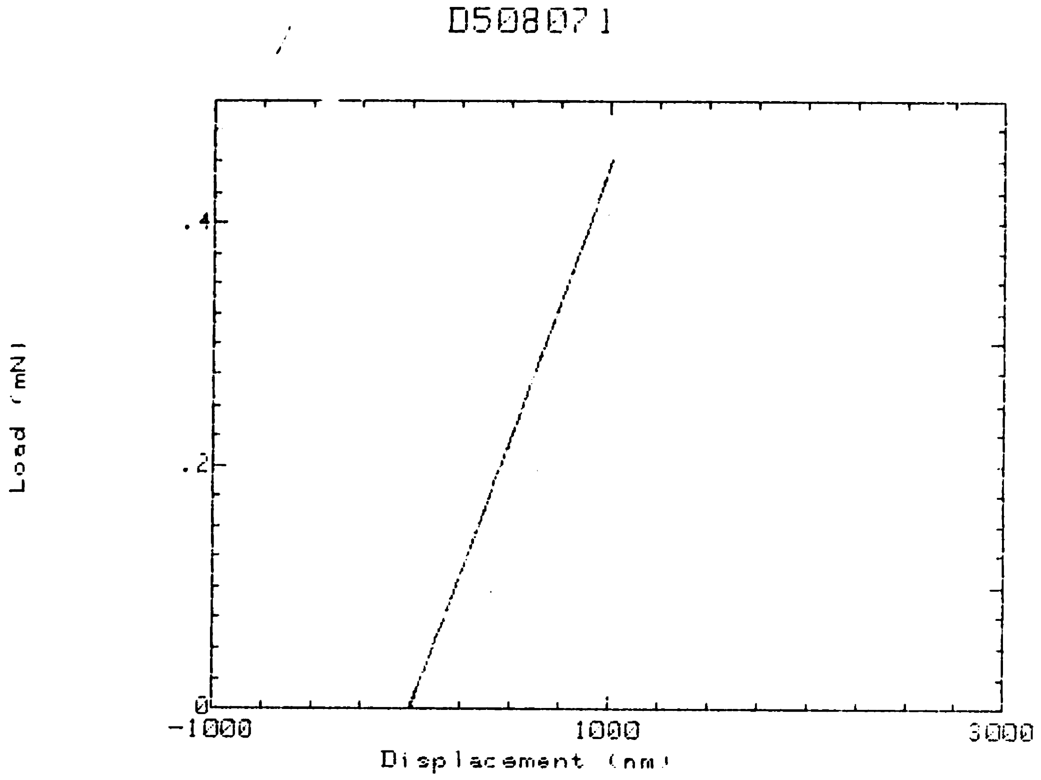


Figure 5.14: Load vs. Displacement of T38 Cantilever

Chapter 6

Generation 1 Fatigue Testing (T38 Series)

On August 6 and 7, 1990 several devices were nanoindented at AT&T. One of these devices, designated as the 091890 device, was indented with two side by side indentations 400*nm* deep with a 5μm separation. The location and orientation of the indentations is shown in Figure 6.1. The two indentations produced an elliptical crack 2μm deep and 8μm long as shown in Figure 6.2. The device was cycled at maximum amplitude for approximately 1 week. No fatigue was observed. The device was remounted for transport to AT&T where the nanoindenter was used to statically deflect the device until fracture recording both load and displacement. The fractured surface was then photographed with the SEM then sectioned and polished to reveal the true cross section. The computed fracture toughness agrees closely with the measurements; however, the compliance does not. A detailed account of the fatigue and break test follows.

6.1 091890 Fatigue Test

Fatigue testing of the 400 nm double indented device, 091890, commenced on August 23, 1990. The starting natural frequency, recorded in air, was 2070 Hz with a driving voltage of 5 volts. Normally, the uncracked device has a frequency of approximately 2500 Hz; therefore, the device does in fact experience a significant frequency drop caused by the pre-crack indicating

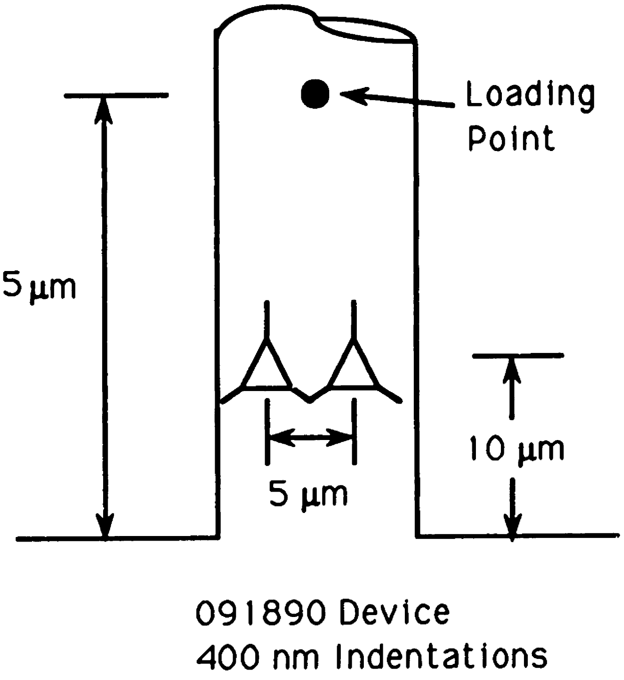


Figure 6.1: Indentation Pattern for 091890 Device

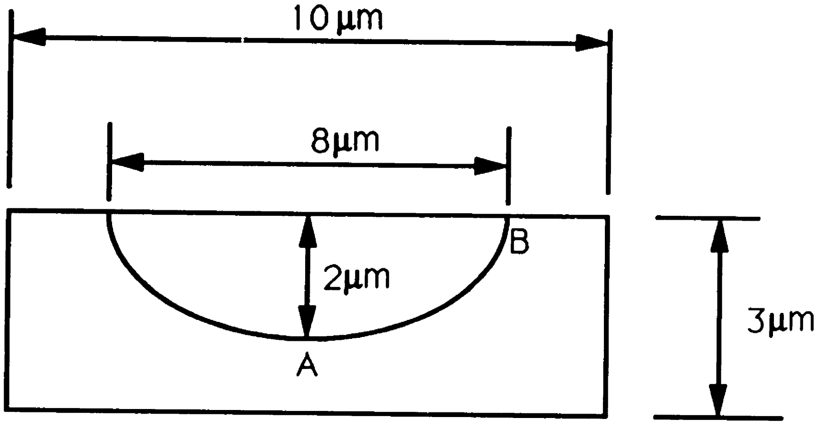


Figure 6.2: Elliptical Crack Geometry

the validity of the experimental technique. Unfortunately, the bridge electrodes of the uncracked devices fell off because of bad metal adhesion so that a comparison of cracked to uncracked device frequencies could not be made. Devices from the same chip are virtually identical and such a frequency comparison would be useful to determine the precise extent of the pre-crack. However, the frequency resolution can be conservatively estimated by assuming a pre-crack of 2 microns and a frequency drop of 500 Hz giving a resolution of 250 Hz/micron. Because the control system can maintain the device on resonance with an accuracy of 1 Hz, a 4 nm crack extension should be detectable with this device. This estimated resolution is close to the theoretical value of approximately 2.5 nm/Hz.

The frequency of the device in air is sensitive to the amplitude of motion. With a driving voltage of 5 volts, the frequency is 2070 Hz; at 11 volts, the frequency is 2040 Hz. Although the absolute level shifts from one test to the next, once a level is established, the relative changes remain small, approximately 0.5 Hz for stable quasi-linear operation. Therefore, overall measurement accuracy is not lost. A number of factors contribute to the sensitivity of frequency to amplitude: nonlinear damping, air mass loading, electrostatic attraction, nonlinearities in the sensor readout, and the variable stiffness of the beam. At low amplitudes the effects are small but still contribute some error to the frequency measurement, particularly from mass loading and nonlinear damping. These combined effects make it difficult to obtain the same starting frequency in air; therefore, baseline frequency measurements are taken in vacuum with a Hewlett-Packard frequency counter at constant test conditions. The test conditions are as follows.

1. $T_d = 30°C$ (device temperature)

2. P = 200 mTorr (chamber pressure)

3. $E_d = 1$ volt (down sensor voltage)

The recorded variables, shown for a measurement on 9/11/90, are:

1. $\omega_v = 2076$ Hz (device frequency)

2. $T_a = 25°C$ (ambient air temperature)

3. $E_u = 1$ v (up sensor voltage)

4. $V_0 = 1$ v (driving voltage)

Initially, the device was cycled at low amplitude with a torquing voltage of 5 volts, however, low amplitude motion produced no measurable frequency shift. Therefore, the voltage was incrementally increased until the maximum was reached. The maximum limit, approximately 15 volts, is where the device cannot be driven at higher amplitudes without hitting the bridge electrodes.

After driving the device at high amplitudes for approximately 1 week no frequency shifts were recorded. The device temperature was maintained at 30°C at 75% relative humidity throughout this testing period. The absence of fatigue is perhaps to be expected, considering the stress intensity at the crack site is approximately 3% of the K_{IC} (this value was confirmed by the break test in the next section). Values of 60% to 70% are more typical of what is required to initiate fatigue. At this point, the operating temperature was raised to 100°C in order to lower the activation energy of any temperature dependencies of the corrosion mechanism. After 4 days no shift was recorded, and on 9/18/90 the temperature was raised to 150°C which is near the limit the of the temperature control system.

On 9/20/90 it was noted the device amplitude decreases significantly as the temperature is raised. The decrease is attributed to increased damping caused by an increase in the viscosity of the air. At first, the decrease was thought to be caused by thermal breakdown of the pn junction of the device which would result in leakage currents shorting the carrier signal to ground. However, this possibility was eliminated by applying the carrier signal to the beam with the driving voltage off. The output of the pre-amp/post-amp was monitored continuously as the device temperature rose from 30 to 150°C. No change in carrier amplitude was detected. The pn junction was later tested with a curve tracer on the probe station and showed the usual current versus voltage characteristic of a diode. Short circuits across the electrodes were also checked by applying DC voltages to the torquer and sensor electrodes with the curve tracer. Unfortunately, during this testing the device "snapped" to the electrode and excessive current was applied to the structure melting the center portion of the beam. In light of this mishap, a new procedure was adopted for checking continuity with the curve tracer involving a large series resistance. The proximity of the pre-crack to the substrate apparently created enough of a heat sink such that the device did not melt at this point. The device was reinstalled in the test chamber and continued to function but at a higher baseline frequency of 2257 Hz. Apparently, the cross section of the beam became more square increasing the stiffness accordingly.

After cycling at high amplitude for 24 hrs at 150°C, the frequency was recorded at 2257 Hz on 9/21/90; virtually unchanged from the previous day. Testing resumed, and after 6 consecutive days (9/27/1990 - 10/2/1990) at the maximum amplitude at 150°C, no change in frequency was recorded. At this point it was decided to stop testing.

Given the absence of any frequency shift with this sample, it is fair to say that subcritical crack growth in micromechanical devices does not occur at stress intensity values less than 3% of K_{IC}.

6.2 091890 Break test

After the fatigue test, the device was removed from the package, remounted for nanoindentation, and transported to AT&T. The nanoindenter was successfully used to statically deflect the cantilever until failure. The load versus displacement curve for the break test is shown in Figure 6.3. The fracture surface was photographed with the SEM at Draper Laboratory as shown in Figure 6.4. The chip was then potted in Lucite and polished such that the true cross section of the cantilever was revealed as shown in Figure 6.5. Sectioning the device is necessary because the SEM exaggerates the dimensions of the cantilever and crack surface when viewed at an angle. Therefore, obtaining a true cross section of the cantilever allows the photos of the fracture surface to be approximately scaled to the correct dimensions.

The software package NASCRAC was employed to compute the stress intensity of the elliptical crack geometry in order to include the effects of finite width. Equation 6.1 was also used to compute the stress intensity for an elliptical crack in an infinite slab as a comparison to the NASCRAC values [38].

$$K_I = \frac{2.4M\sqrt{\pi c}}{h^2 b} \tag{6.1}$$

where,

$$\begin{aligned} M &= 2.25 \times 10^{-10}\, Nm \\ h &= 3 \times 10^{-6}\, m \\ b &= 10 \times 10^{-6}\, m \end{aligned}$$

	Location	K_I	$\%K_{IC}$
NASCRAC	A	20795	2
	B	31718	3
Eq. 6.1	A	15054	1.5

Table 6.1: K_I for Elliptical Crack

$$c = 2 \times 10^{-6}\, m$$

Given the dimensions of the 091890 device, the true cross section of the cantilever, the location, depth, and geometry of the indentation, Table 6.1 gives the computed values for K_I at points A and B in Figure 6.2.

NASCRAC gives the maximum stress intensity applied at point B in Figure 6.2 at 3% of the fracture toughness of silicon, $K_{IC} = 1\ MN/m^{3/2}$. This value is far below the level necessary to initiate fatigue. Values on the order of 60 to 70% of K_{IC} are typical of static fatigue experiments with ceramics. The moment necessary to fracture the device is found by dividing M by 3%.

$$\begin{aligned} M_f &= \frac{M}{0.03} \\ &= 7.5 \times 10^{-9}\, Nm \end{aligned} \tag{6.2}$$

In reference to the break test, the load necessary to statically fracture the cantilever can be found by dividing the failure moment, M_f, by the lever arm of the applied load. The nanoindenter deflected the cantilever $15\mu m$ from the indentation site, and the indentation site was located $10\mu m$ from the fixed end of the cantilever. See Figure 6.1. This translates into a failure load of 0.50 mN. Figure 6.3 shows the device failing at 0.38 mN (0.53 - 0.15 mN). The calculated value is 19% greater than measured. Some error is attributed to the procedure by which the nanoindenter is programed to "find" the surface, and finite thickness effects. Finite width effects were shown to be significant; therefore, finite thickness effects are likely to be significant also.

When an indentation site is specified, the nanoindenter first moves the sample horizontally under the indenter head using precision x-y stepping motors to a position $50\mu m$ from the indentation site. The nanoindenter then makes a coarse surface find which positions the entire indentation head assembly close to the surface of the sample using a stepping motor in the z-direction. Within the indentation head assembly is the indenter head which is independently controlled in the z-direction via a precision voice coil. The indenter head is a diamond point mounted on a cylindrical metal shaft that is suspended by leaf springs to the head assembly. It is the precise control of the current flowing through this coil that gives the nanoindenter its force/displacement accuracy. After the coarse surface find, the z-motor is shut off and the nanoindenter picks up the indenter head using the voice coil and moves $25\mu m$ closer to the target then rests on the surface allowing the stepping motors to cool, and the machine components to come to thermal equilibrium. Next, the nanoindenter moves above the indention site and begins lowering from a height of $20\mu m$ until the indenter head is $1\mu m$ from the surface. At this point, the nanoindenter enters the search mode and begins lowering the indenter head at a rate of 10 nm/s while continuously measuring its own spring stiffness. These values of stiffness are referred to as the approach data. When the indenter head contacts the surface the stiffness of the sample is combined in parallel with the leaf springs of the nanoindenter and the total stiffness increases until it is a factor of four greater than the stiffness of the nanoindenter. The nanoindenter has now found the surface. The load and displacement settings are zeroed, the approach data purged except for the last 17 points, and the indentation cycle begins. The difficulty arises because the pre-cracked cantilever has a stiffness of 266 N/m which is only a factor of 3.3 greater than the nanoindenter stiffness, 80 N/m. Therefore, the cantilever is deflected some amount before the nanoindenter realizes it is on the surface so that the curve shown in Figure 6.3 under estimates the actual load applied to the device. Figures 6.7 and 6.6 are regression curves of the actual data saved from the nanoindenter. The approach data has only 17 points but shows no significant transition in spring stiffness from 80 N/m to 266 N/m; therefore, the "knee" of the curve was not captured and the actual load applied to the device is not known with certainty. After consultation with the inventor of the nanoindenter, the software driving the machine was modified to save several hundred data points of the approach data which was adequate to capture the true surface contact point.

The break test result is still encouraging because the error is only 19%, not an order of magnitude which would indicate a drastic error. Also, the stiffness of the cracked cantilever was close to the prescribed factor of 4; therefore, the actual load applied to the device must be close to the measured value. The fracture was purely brittle with no rounding of the curve near the failure load indicating the absence of plastic deformation. Lastly, the method of break testing can be made to work.

Although the load required for failure correlates within 19% of the NASCRAC analysis, the slope of the curve shown in Figure 6.3 does not correlate with the calculated change in stiffness as predicted by fracture mechanics theory. Substituting Equation 6.1 into Equation 4.14 and integrating from 0 to a gives the rotational compliance of the crack.

$$C(a) = \frac{(2.4a)^2\pi}{h^4bE} \tag{6.3}$$

The crack compliance is then related to the displacement of the cantilever at the point of application of the load by defining an equivalent spring stiffness, K_c.

$$K_c = \frac{1}{C(a)L_a^2} \tag{6.4}$$

where L_a is the lever arm of the applied load with respect to the indentation.

$$\begin{aligned} C(a) &= 5.26 \times 10^5 \ (N\ m)^{-1} \\ L_a &= 15 \times 10^{-6}\ m \\ K_c &= 8.5 \times 10^3\ N/m^{3/2} \end{aligned}$$

K_c is then added in series with the stiffness of the uncracked beam to give the total stiffness of the cracked cantilever, K_T. The stiffness of the uncracked beam was measured directly with the nanoindenter by deflecting an uncracked beam at the same location as the cracked beam. See Section 5.3.3. The measured stiffness was $K_{uc} = 452\ N/m$; therefore, the nanoindenter found the surface accurately, and Figure 5.14 accurately represents the actual loads and displacements applied to the cantilever. The total stiffness of the cracked cantilever is as follows.

$$\begin{aligned} K_T &= \frac{K_cK_{uc}}{K_c + K_{uc}} \\ &= 429\ N/m \end{aligned} \tag{6.5}$$

The stiffness of the pre-cracked cantilever, however, is 266 N/m which indicates that K_c should be on the order of 646 N/m which is a factor of 13 less than the calculated value of 8.5×10^3. Note however, that Equation 6.1 does not account for finite width or thickness effects; therefore, the value for K_c would be conservative.

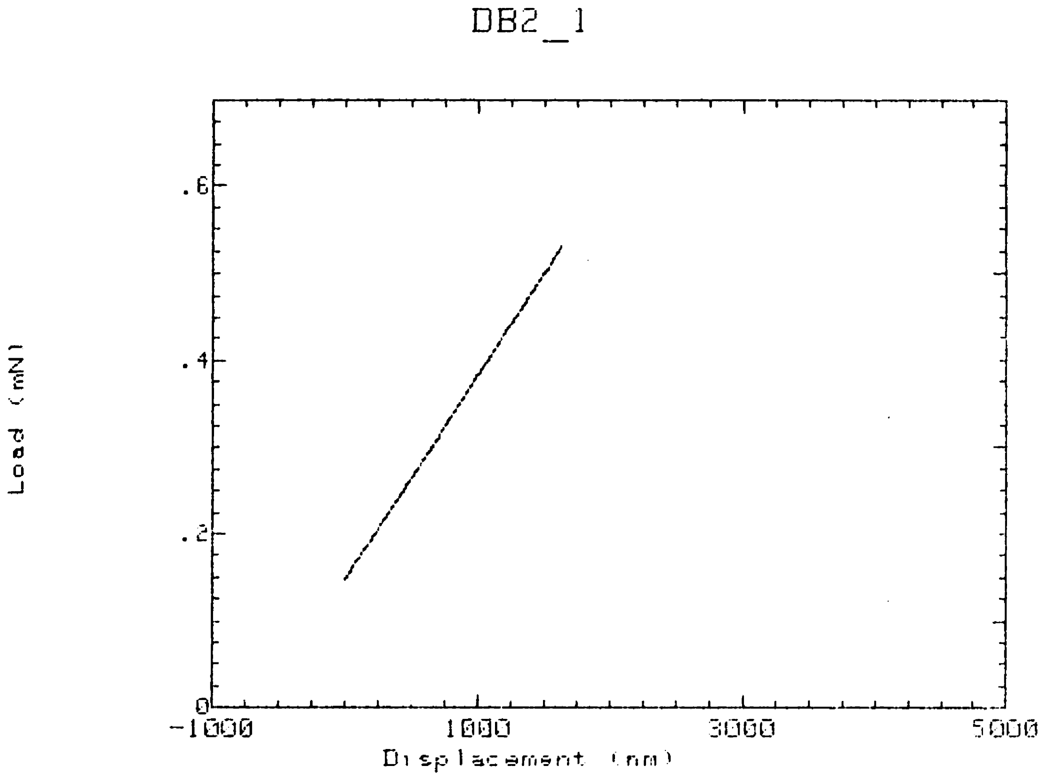


Figure 6.3: Load vs. Displacement to Failure

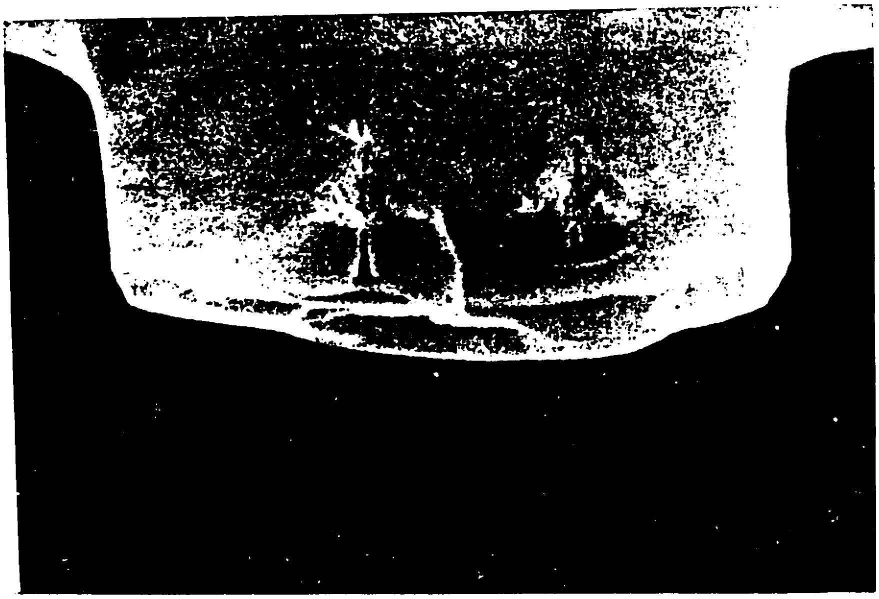


Figure 6.4: Fracture Surface

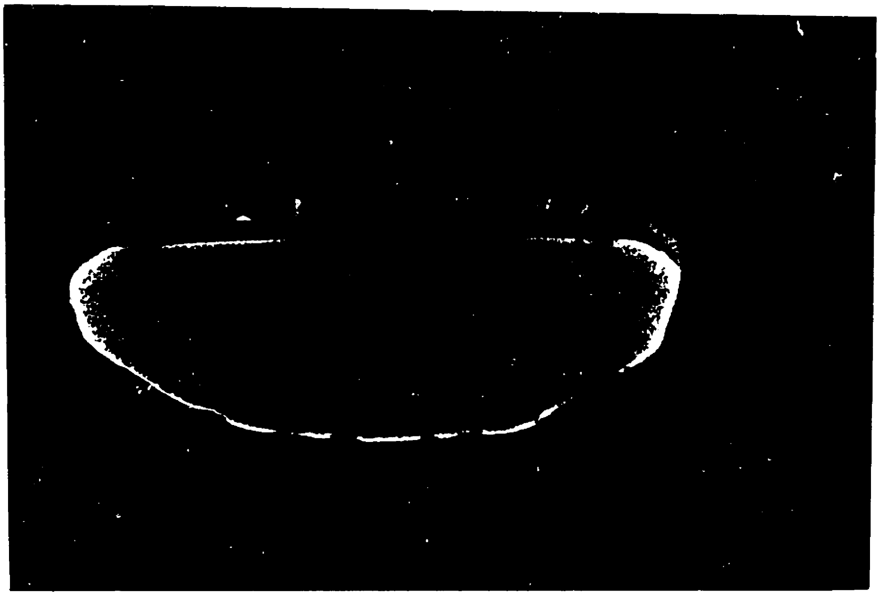


Figure 6.5: Cross Section of Cantilever

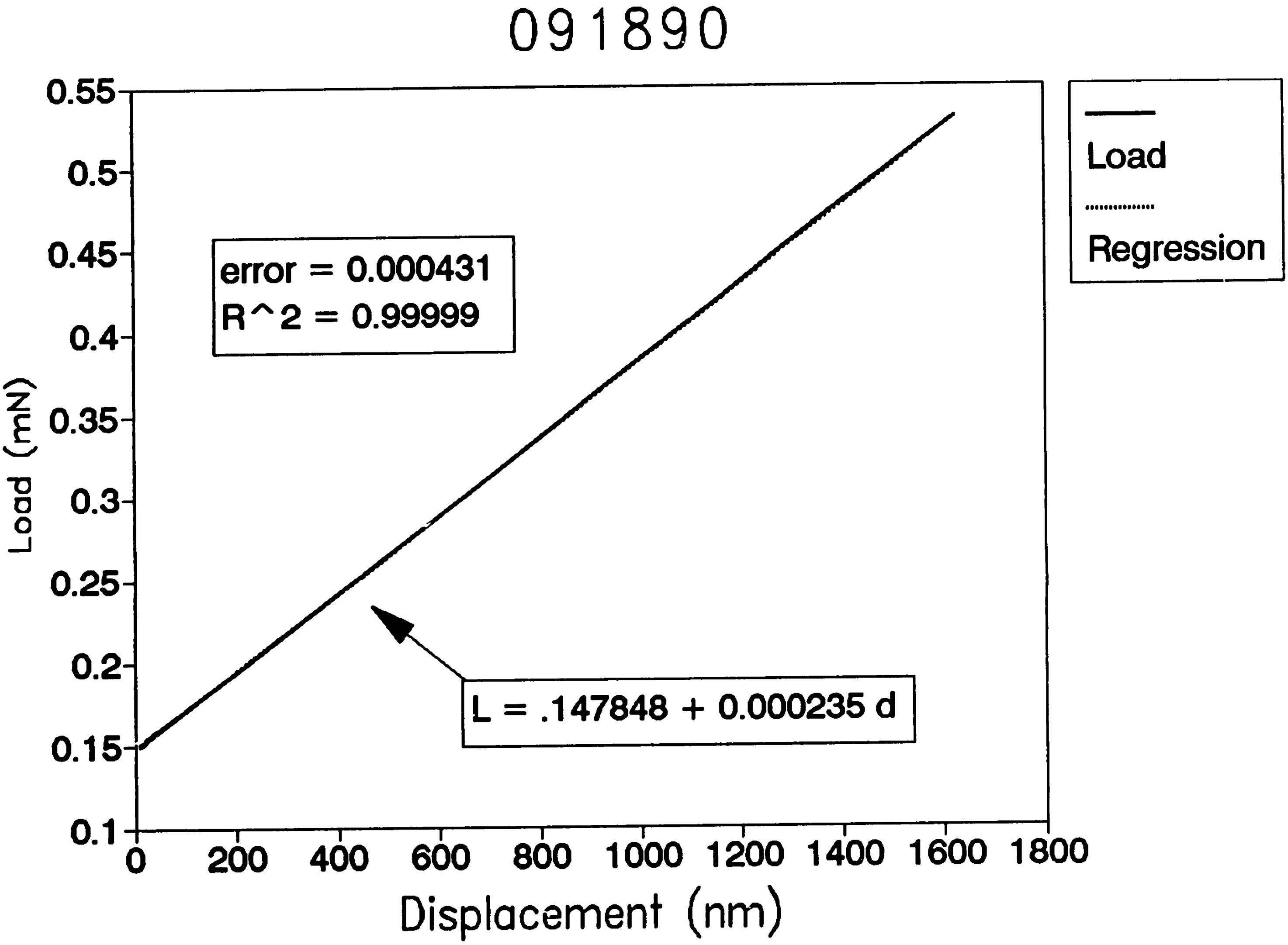


Figure 6.6: Load Data

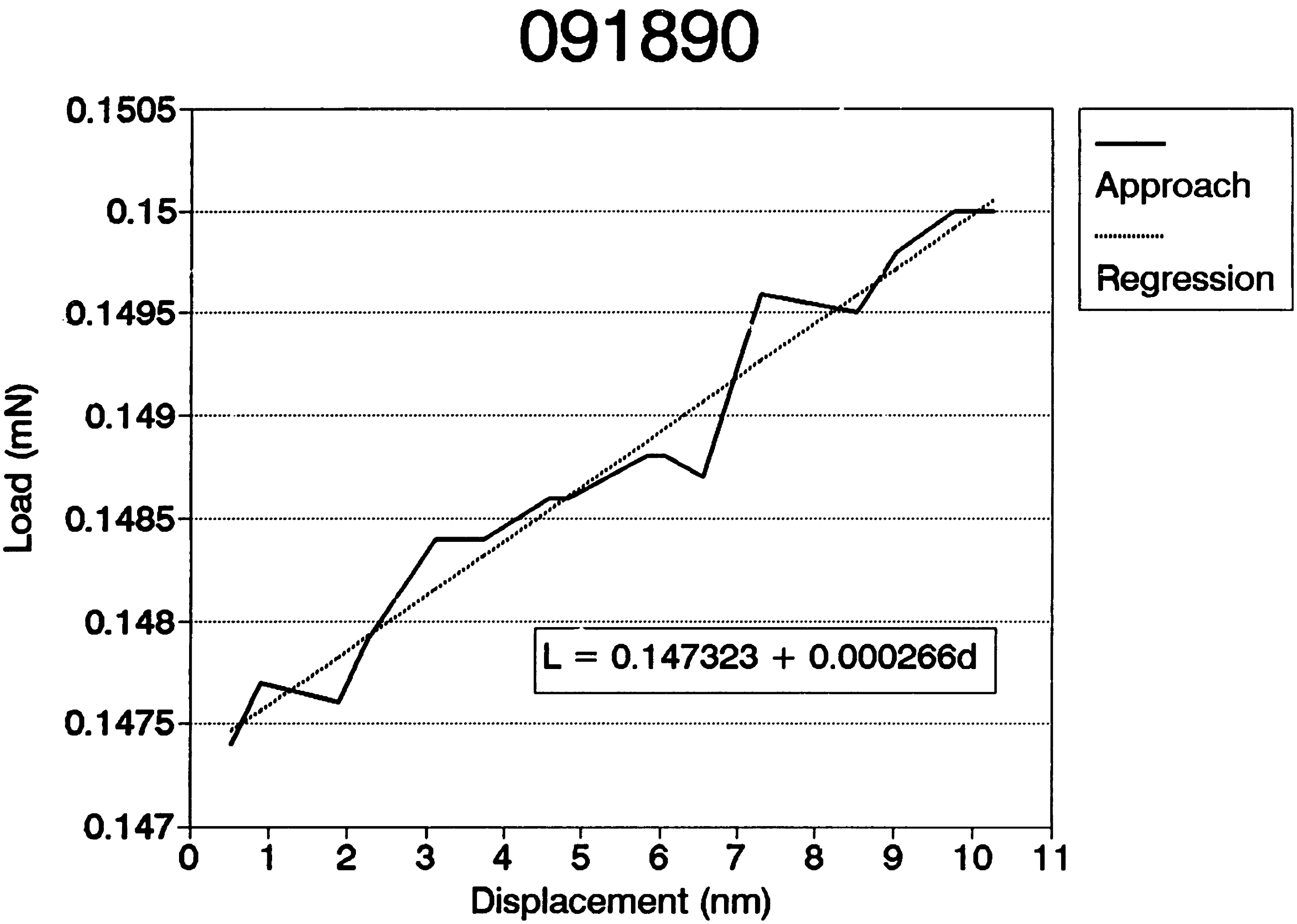


Figure 6.7: Approach Data

6.3 Conclusions of Generation 1 Fatigue Testing

Subcritical crack growth was not measured with the T38, or generation 1, devices. The absence of measurable crack growth is attributed to the design of the device which limited the maximum stress intensity at the crack site to approximately 3% of K_{IC} which is insufficient to initiate fatigue. The generation 1 device design has the following limitations.

1. In the preliminary design stage, the dynamic model of the device incorporated two degrees of freedom for the cracked beam structure leading to a design that was too compliant. A three degree of freedom model was then employed to give a better prediction of the first mode frequency. ABAQUS was used to verify the model. These results were presented in Section 2.2.

2. The beam cannot be fully cracked across the width without failing during etching and rinsing. Calculations show that the surface tension of water trapped in the void below the end plate of the device will cause failure during etching and rinsing.

3. The footings for the sensor electrode, torquer electrode, and counterweight are too closely spaced to achieve uniformity in the photoresist spacer layer. This layer produces the air gap between the end plate and bridge electrodes. Therefore, the air gap dimension cannot be maximized with respect to the required displacement of the beam. This limitation includes devices produced with thicker beams with the generation 1 design.

Because of the limitations described above, (particularly the air gap dimension), the fatigue device must be redesigned and a new mask set issued. The next chapter describes the new generation 2 device design.

Chapter 7

Generation 2 Fatigue Testing (N141 Series)

7.1 New Device Features

As explained in the last section, the design of the generation 1 devices (T38 series) led to a structure that was too compliant. With these devices, it was not possible to apply a stress intensity at the crack site high enough to initiate slow crack growth. Therefore, the test structure was redesigned greatly increasing the stiffness so that high stresses could be applied at the crack site. As mentioned in Section 2.1, the cantilever configuration has many advantages. Therefore, rather than abandon this concept, the design was optimized. The goal of the redesign effort was to produce a device that could be repeatably pre-cracked with the nanoindenter across the full width of the cantilever, etched, rinsed, packaged, and ultrasonically wire bonded without failure. Following wire bonding, the device must be capable of being broken with the drive electronics given a maximum clearance of $12\mu m$ beneath the electrodes. This redesign effort was entirely successful. The principal design changes are enumerated as follows.

1. The separation of the counterweight and sensor electrode was increased from 30 to 100 microns, and the separation between the sensor and torquer electrodes was increased from 30 to 40 microns. The electrode footings were also spaced further apart, from 200 to 400 microns. These changes allowed a thick photoresist spacer layer, approximately 12 mi-

crons, to be spun onto the wafer thereby maximizing the air gap. The electrode footings and counterweight produce step changes in the height of the photoresist which act as boundary conditions on the flow properties of the resist and round somewhat during the postbake destroying uniformity in the air gap dimensions. The postbake is a processing step that cures the photoresist by removing trapped solvents through baking in an oven. Therefore, adequate separation is necessary for uniform coverage at the required thickness.

2. The increased separation of the electrodes and counterweight correspondingly increased the size of the silicon paddle and raised the damping. The higher damping was offset by carefully setting the spacing of the electrode footings which determine the size of the etch window. The 400 micron spacing created a deep well depth beneath the device sufficient to cut damping about a factor of 3 from the generation 1 design. This increase in footing separation was limited by buckling of the electrodes during the low temperature 360°C adhesion bake. This bake is a standard processing step that improves metal to metal adhesion, and metal to oxide adhesion.

3. In addition to lower damping, the deep well design reduces liquid surface tension forces which were identified as the primary cause of failure during the drying step after etching and rinsing. The liquid evaporates from the well last producing negative pressure forces under the paddle which suck the paddle downward causing premature failure. Although the larger paddle area increased this pressure force, the depth of the well had a proportionally greater effect in reducing it from the previous design.

4. The beam was shortened a factor of 2, from 130 microns to 75, and the beam thickness increased from 3 to 8 microns. These design modifications created a much stiffer structure so that high stress intensities could be developed at the crack site given the limited travel under the bridge electrodes. The thick beam allows the cantilever to be fully pre-cracked across the width while leaving a large ligament, about 4 microns thick, which is necessary to prevent failure during etching and rinsing. The length of the beam was limited to 75 microns to retain

adequate clearance for the nanoindenter to indent the base of the cantilever without striking the torquer bridge electrode. The thickness of the beam was also limited to approximately 8 microns by the diffusion time, and to ensure the nanoindenter can indent the beam to the required depth without exceeding the maximum load capability.

5. The height of the gold counterweight was maintained at 15 microns which allowed for easy fabrication. Plating heights greater than 15 microns would not provide adequate clearance for the nanoindenter to indent the cantilever given a 12 micron air gap and a 75 micron beam length. This plating height is also great enough to prevent buckling of the bridge electrodes during the $360°C$ sinter. A large plating height, however, is desirable to the extent that less voltage is required to drive the structure to a given amplitude at resonance. Furthermore, a lower device frequency means that a lower carrier frequency can be used reducing phase shift at the pre and post amplifiers.

6. Given a stiffer structure, the resonant frequency increased from 3 KHz to 12 KHz which was high enough that the carrier frequency had to be moved from 100 KHz to 300 KHz. Raising the carrier frequency caused too much phase shift at the single stage post-amp creating excessive signal attenuation after demodulation. Therefore, the front end amplifiers were redesigned with a low noise pre-amp and a two stage, high speed post-amp. The carrier signal level was raised from 1 volt to 5 volts improving the signal to noise ratio which was reduced by the large air gap. The stiff structure required torquing voltages of approximately 100 volts so that a step-up transformer was added. However, with the transformer operating at 12 KHz excessive phase shift was added in the torquing circuit, therefore, a phase adjust pot was added to the control circuit allowing the device to be set exactly on resonance. The torquing voltage is limited to less than 300 volts to prevent dielectric breakdown of the field oxide which will cause the torquer electrode to short to the substrate.

7. The bond pads were relocated further from the device to minimize vibration transmission to the device from ultrasonic wire bonding. The electrode footing size was also increased to maximize bridge metal adhesion. Poor adhesion was a constant problem with the generation 1

devices. Because of the large electrode footings and long metal lines to the bond pads, capacitance to the substrate was sufficient that a carrier null circuit at the 2nd stage post-amp was added. See Figure 3.10. The null circuit canceled the static carrier signal prior to demodulation reducing the filtering requirements in the sensor circuit.

Figure 7.1 and Figure 7.2 are schematic diagrams of the redesigned fatigue structure. Figure 7.3 is a SEM photo of an actual generation 2 device.

7.2 102491 Fatigue Test

From 9/30/91 to 10/5/91 several generation 2 devices were nanoindented at Oak Ridge National Laboratory. Of these devices, several were successfully etched, rinsed, packaged and wire bonded for testing. One of these devices, the 102491 device, was nanoindented with 5 indents at 50 mN with a $4.7\mu m$ spacing between indents. The orientation of these indents is shown in Figure 7.4. Testing began on 10/24/91. The relative humidity at the device was approximately 75%. This device, and all other subsequent devices, exhibited similar crack growth behavior. Initially, the frequency drops rapidly with crack velocities on the order of 1.0×10^{-9}m/s, then the velocity exponentially decreases to levels on the order of 1.0×10^{-13}m/s. Step increases in stress intensity produce the same characteristic fast transient followed by the frequency leveling off. Eight test runs were recorded for the 102491 device, failure occurred on the ninth run. At the start of each run, the voltage was increased producing a step change in stress intensity, followed by the corresponding frequency drop and leveling off. Frequency measurements were continued until further changes were small. Figure 7.5 is the first test run of the 102491 device. Figure 7.6 is the last 2000 seconds of the first run with a linear regression curve fit to the data. From the regression curve, the crack velocity in the slow region is approximately 8.4×10^{-13} m/s. Figure 7.7 is the second test run after a step increase in stress intensity. Figure 7.8 is the last 5000 seconds of the second run. Appendix D contains the frequency data for all seven recorded runs of the 102491 device. These runs are condensed and tabulated in Table 7.1 and plotted in Figure 7.9. The lowest crack velocity measured was 8.1×10^{-15} m/s; however, considerable error exists in the regression analysis. The frequency may have been dropping, but at a rate that was not detectable given the accuracy of the measuring technique.

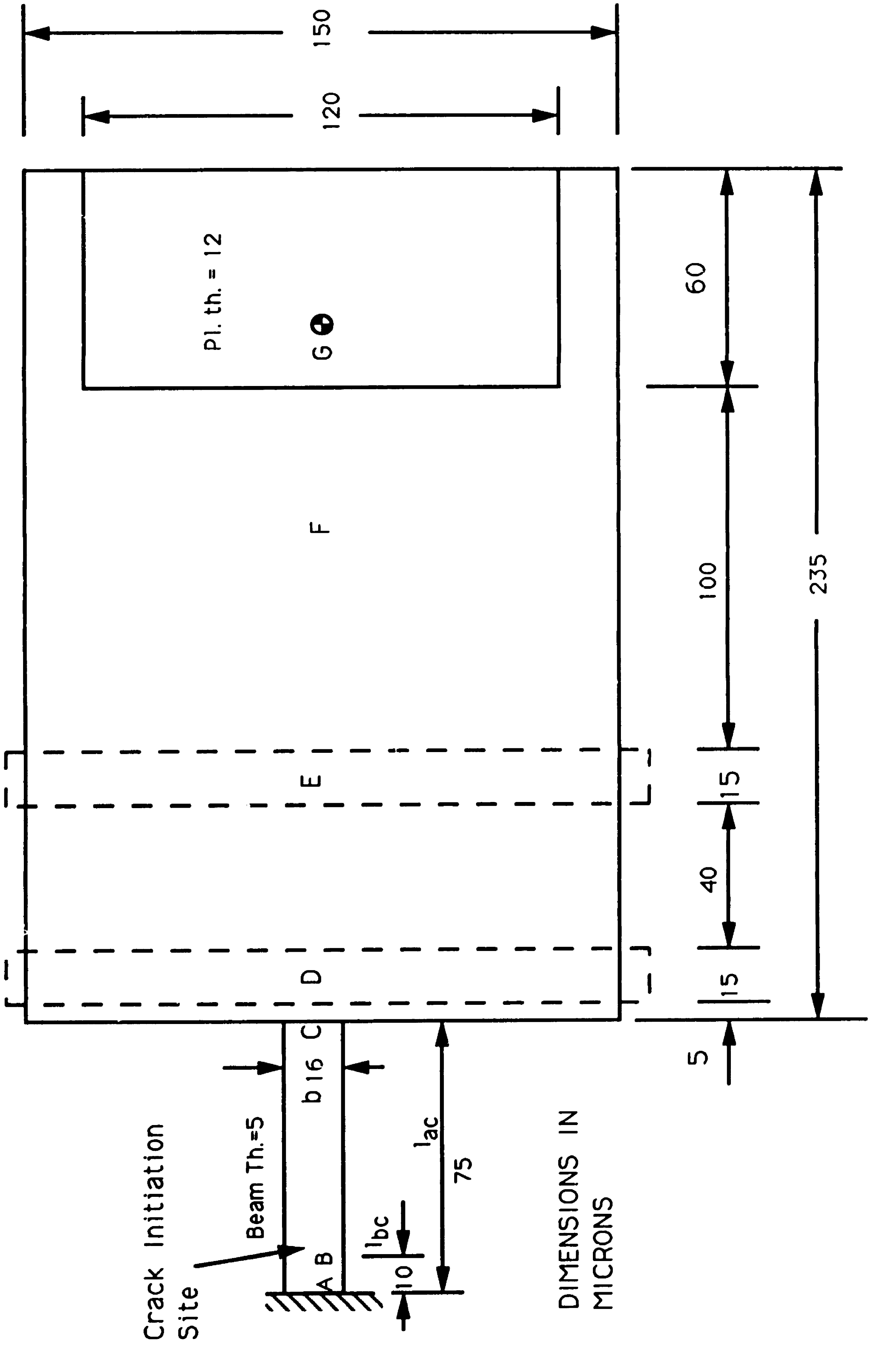


Figure 7.1: Generation 2 Fatigue Device (Top View)

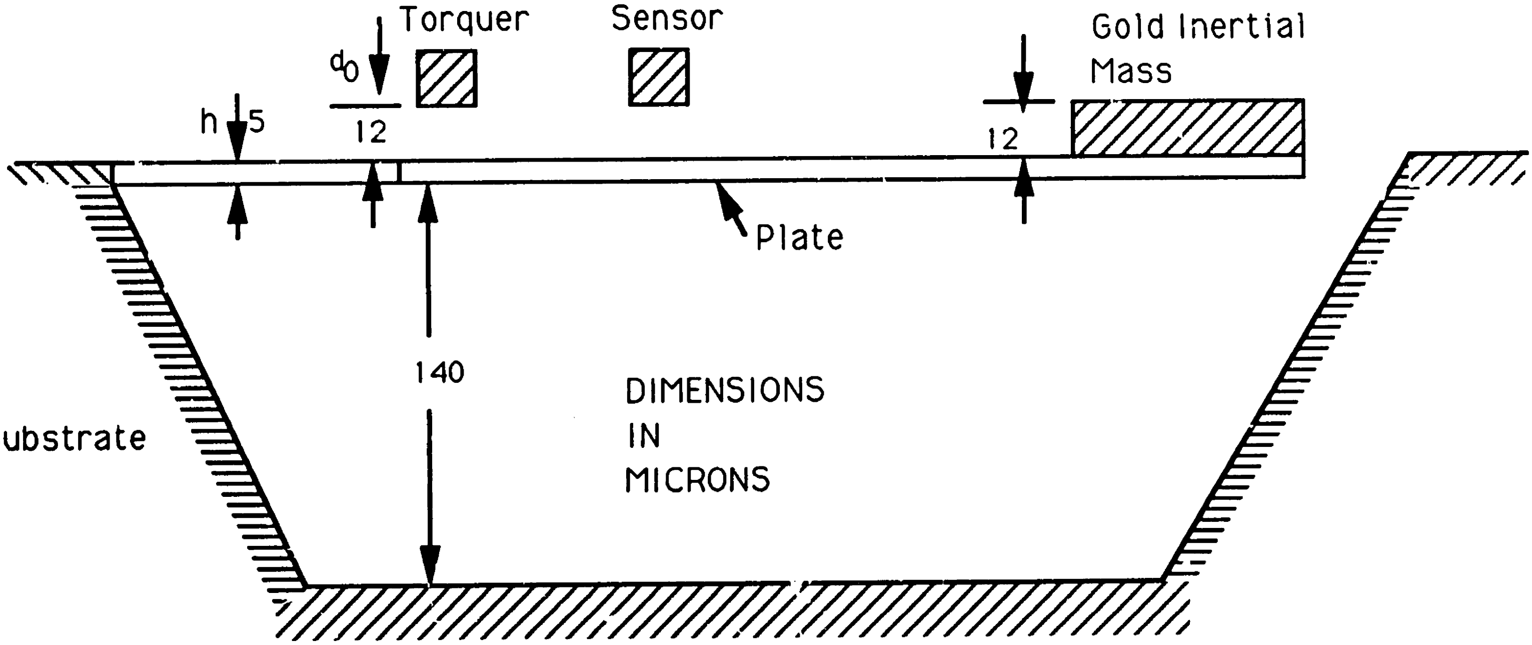


Figure 7.2: Generation 2 Fatigue Device (Side View)

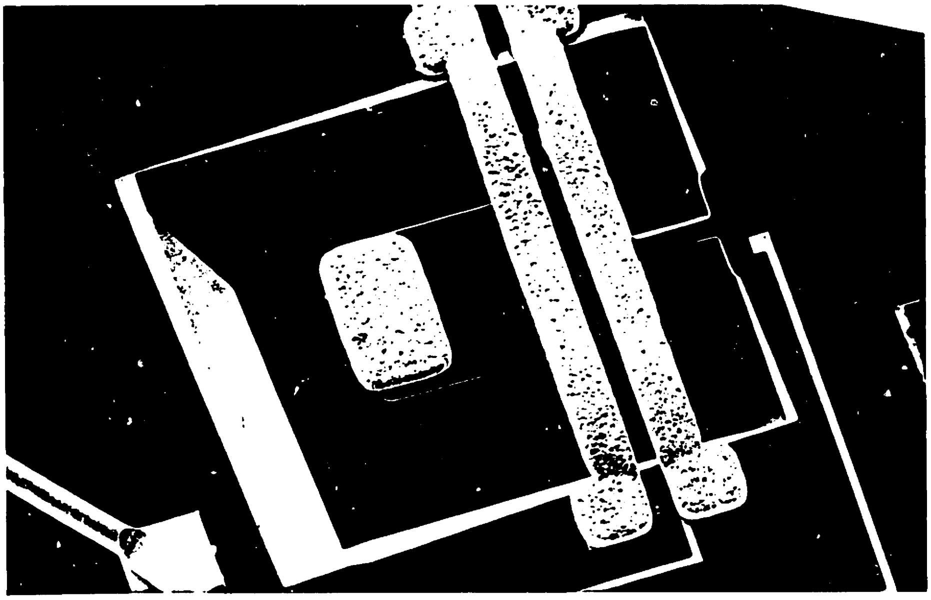


Figure 7.3: Generation 2 Fatigue Device (SEM Photo)

Run	ω_i	ω_f	$\Delta\omega$	V_0	K_I	V	da
	Hz	Hz	Hz	V	$MN/m^{3/2}$	m/s	nm
1	11101	11082	-19	65	0.31	8.4×10^{-13}	70
2	11080	11062	-18	78	0.44	1.82×10^{-13}	63
3	11062	11056	-6	84.5	0.52	8.1×10^{-15}	21
4	11057	11055	-2	91	0.60	1.89×10^{-12}	7
5	11057	11051	-6	97.5	0.69	5.25×10^{-14}	21
6	11053	11050	-3	104	0.78	4.55×10^{-13}	10.5
7	11054	11049	-5	110.5	0.87	5.9×10^{-13}	17.5
8	11050	11049	-1	117	0.97	1.7×10^{-12}	3.5

Table 7.1: 102491 Data: Runs 1 - 8

Improvements in the sensing electronics, particularly the thermal stability of the VCO, should allow for precise measurements of velocities on the order of 10^{-15}m/s or better.

The characteristic of the fatigue curve is contrary to what was expected given the work of Bhaduri & Wang and Wiederhorn [2][45]. Figure 7.10 is a frequency versus time simulation of the 102491 assuming that crack growth obeys the power law model previously described. The constants used in this model were extracted from Bhaduri & Wang and the FTEST2.PAS design program was used for the simulation.

The initial fast growth region is thought to be a combination of the crack front equilibrating and the intrinsic nature of the observed rate limiting mechanism. The initial series of indentations tends to produce an irregular crack front which then smooths out to a uniform depth across the width of the cantilever. Crack front jumping, or stable fracturing, has been measured in the fast growth region indicating the presence of irregularities in the crack front where locally high stress intensities exceed the fracture toughness. A jump appears as an instantaneous step decrease in frequency in an otherwise smooth monotonically decreasing frequency versus time curve. Following the fast growth region, the velocities in the slow growth region all appear to be relatively on the same order of magnitude regardless of the stress intensity. This behavior suggests that crack growth is independent of stress intensity after the initial transient has occurred.

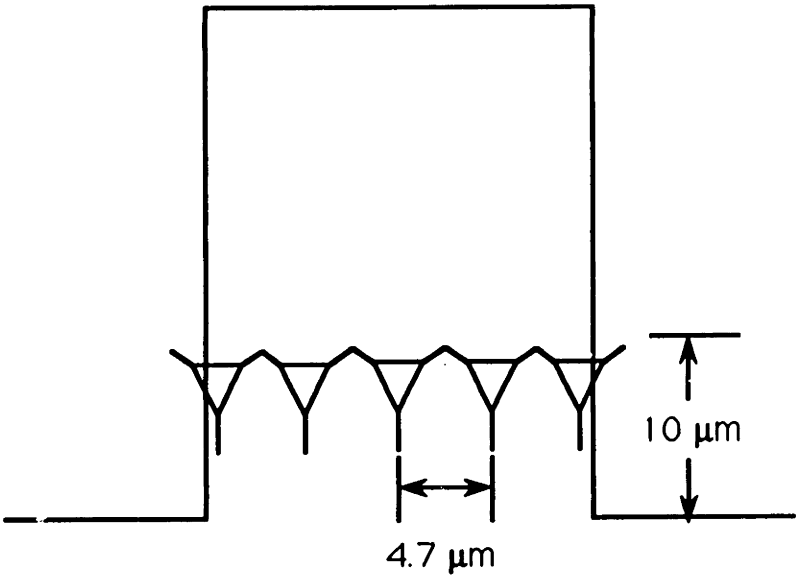


Figure 7.4: 102491 Nanoindentation Pattern

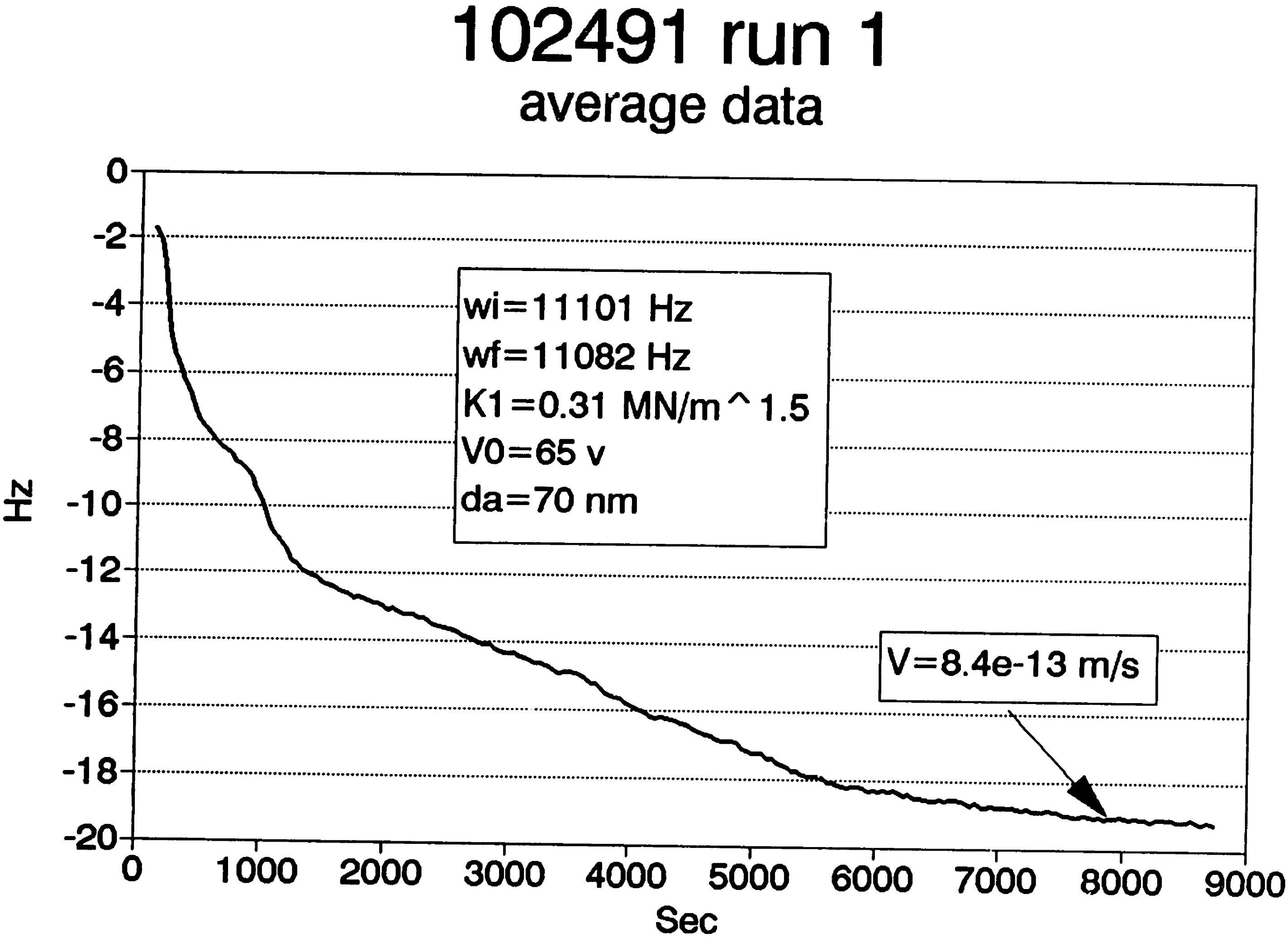


Figure 7.5: 102491 Run 1

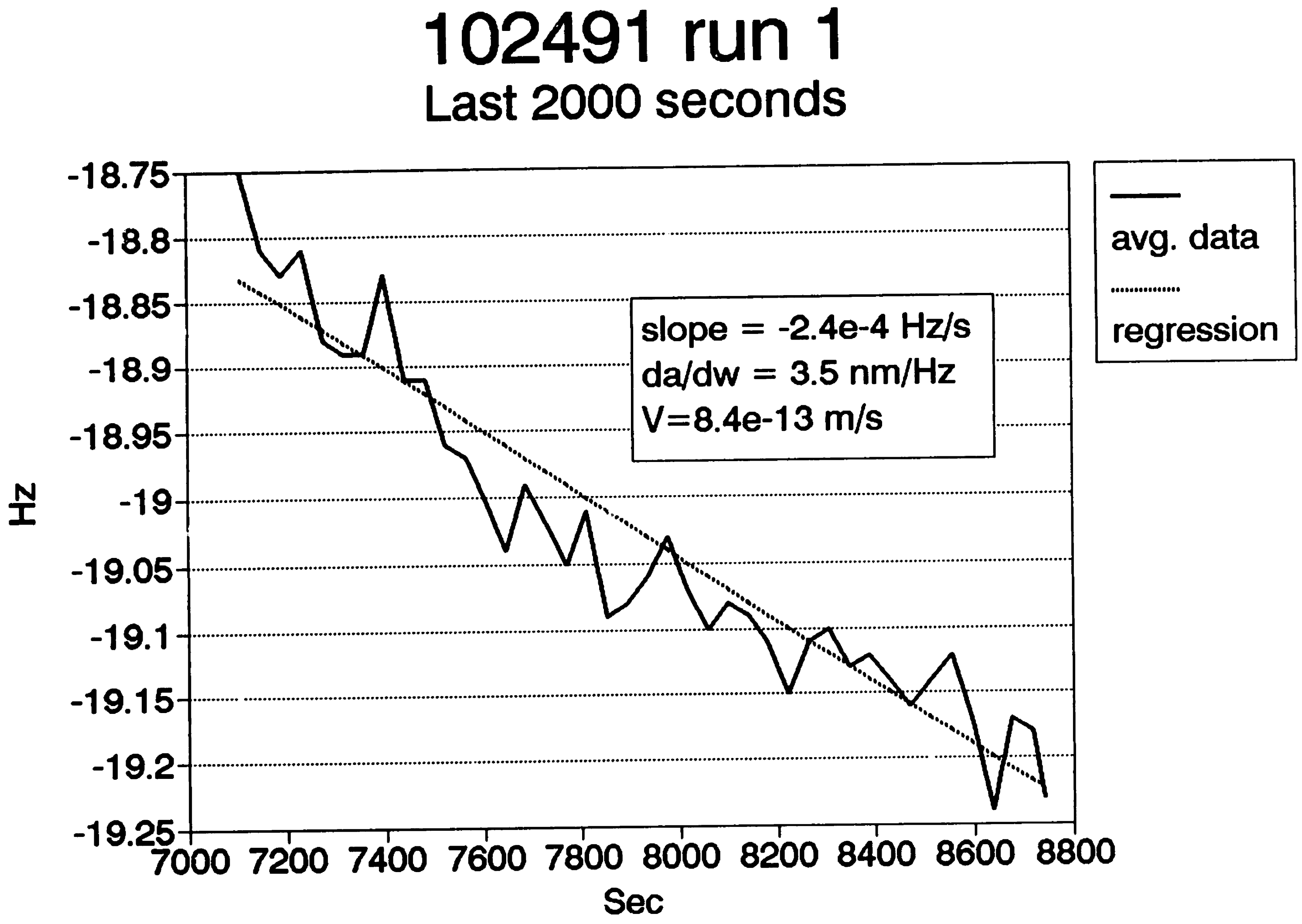


Figure 7.6: 102491 Run 1 (Last 2000 seconds)

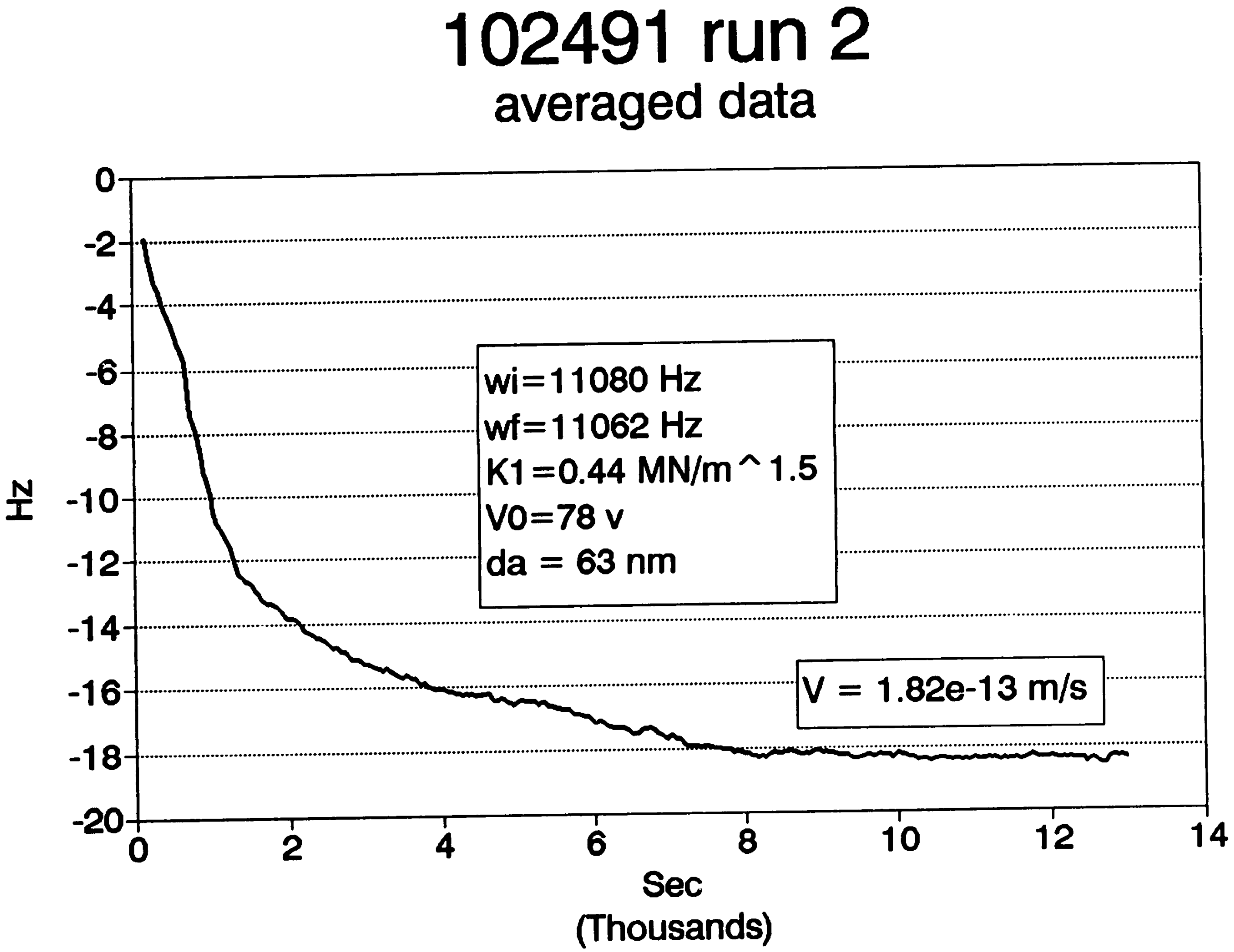


Figure 7.7: 102491 Run 2

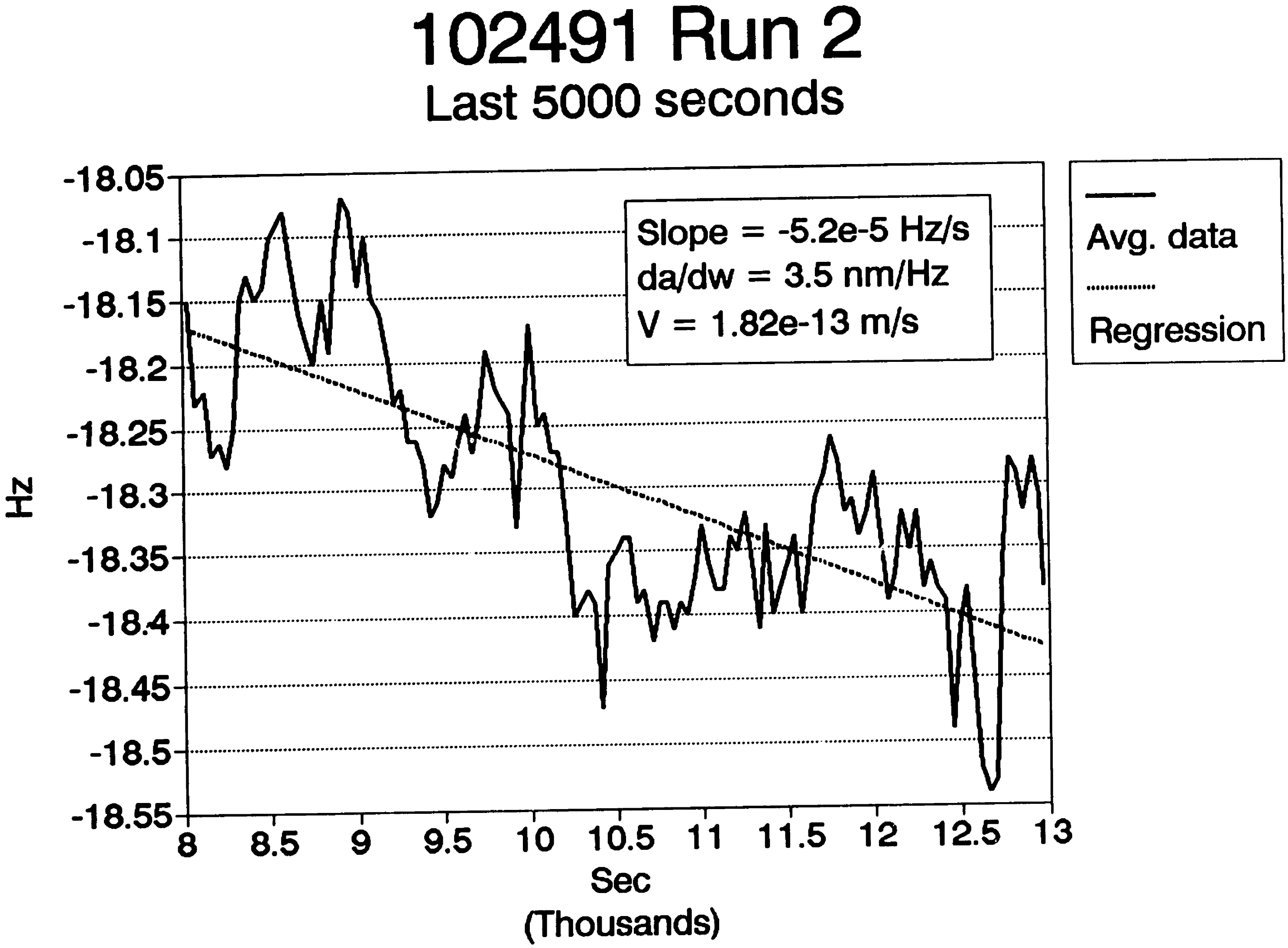


Figure 7.8: 102491 Run 2 (Last 5000 seconds)

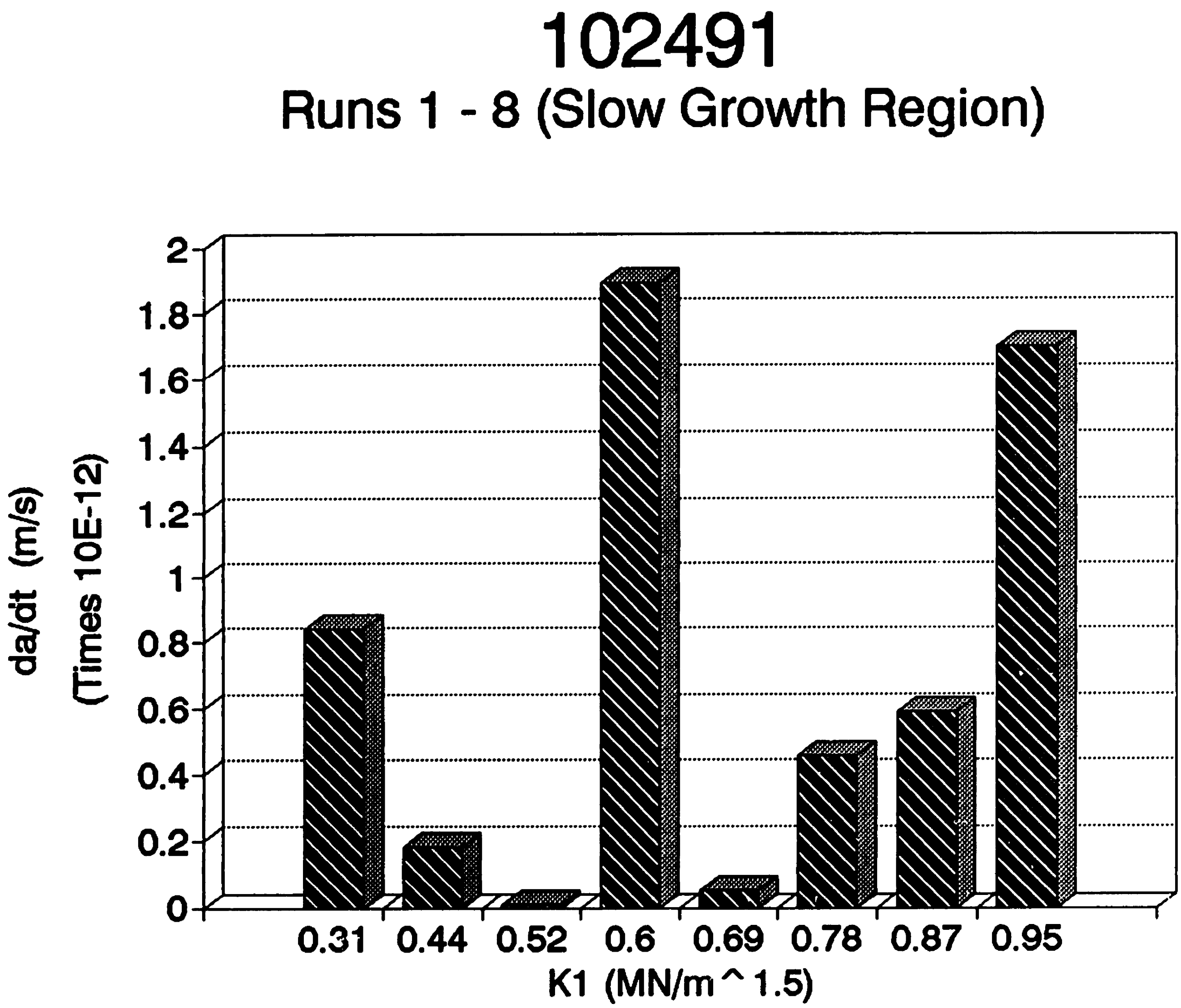


Figure 7.9: 102491 K_I versus $\frac{da}{dt}$

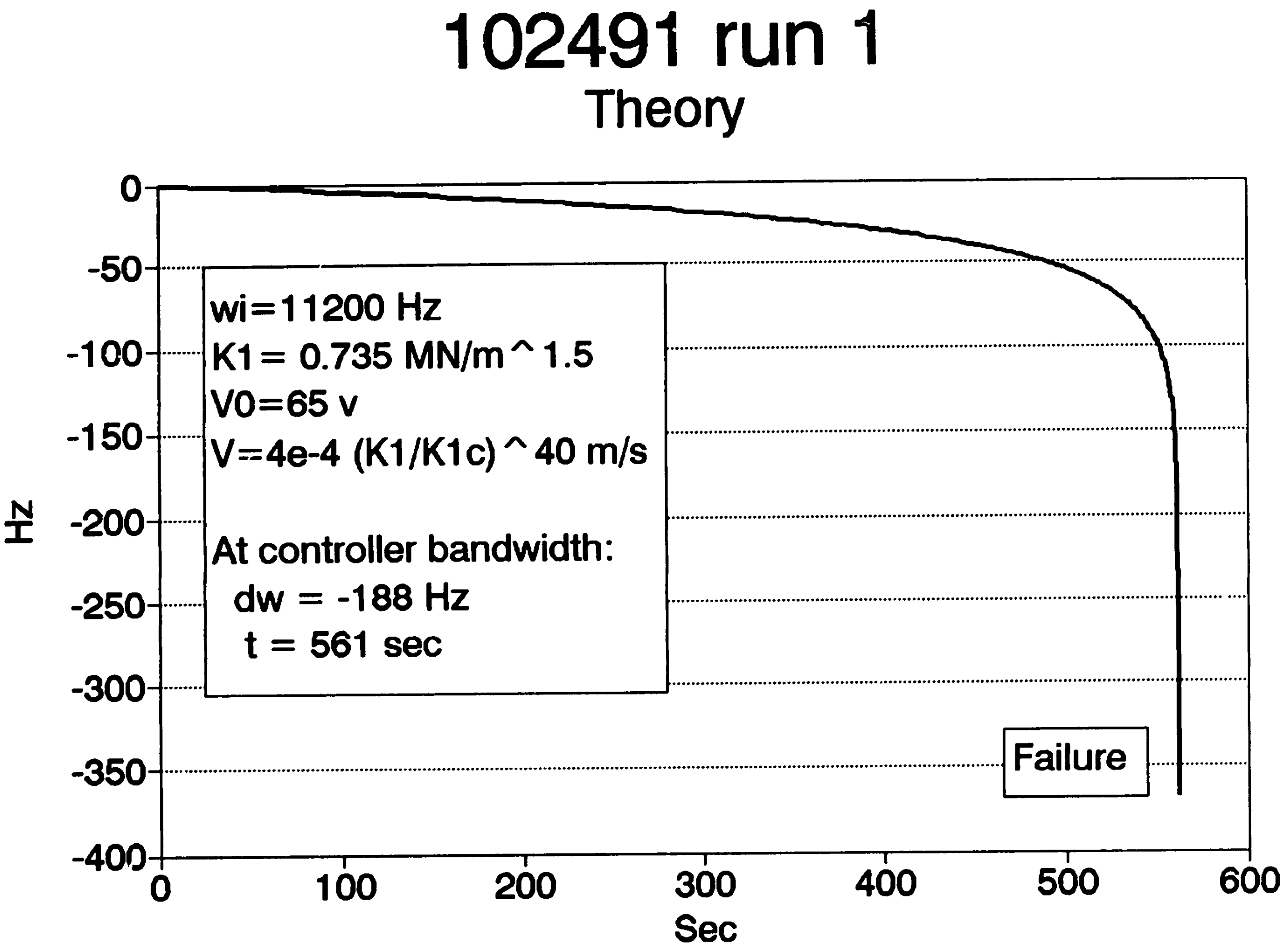


Figure 7.10: 102491 Theoretical Frequency versus Time

7.3 110291 Fatigue Test

The 110291 device was indented with the 102491 device at Oak Ridge National Laboratory. This device had 5 indentations at 60 mN with a 5.5 micron spacing between indentations. This device exhibited the usual fast growth characteristic and the frequency leveling; however, unlike previous devices, the 110291 device was cycled to failure. The calculated starting stress intensity was near the fracture toughness so that a small increase in crack length could cause failure. The frequency versus time curve is shown in Figure 7.11. Figure 7.12 is the fracture surface viewed from an angle of approximately 50° from the vertical. The fracture surface appears as the lower half-ellipse which is lighter in color than the adjacent material. By decreasing the viewing angle to 35.3°, the plane will coincide with the $< 111 >$ plane. Figure 7.13 shows the same fracture surface viewed from an angle of 30° from the vertical. The fracture line, at the time of failure, is clearly visible with uniform depth across the width. The indentation load and baseline starting frequency approximately correlates to the depth of the fracture line. The 130 Hz frequency shift translates to a crack extension of approximately $0.27\mu m$ which is equivalent to 1/16" in Figure 7.12. A relatively uniform crack front appears to have been generated by the successive nanoindentations which suggests that the initial fast growth transient recorded with this device could be an accurate representation of the rate limiting mechanism without substantial effects from the crack front equilibrating. However, some irregularities must be present in this initial crack front, for it is very unlikely a perfectly uniform crack front can be generated with successive nanoindentations. The extent to which these minor irregularities affect the fast growth transient is not known.

7.4 011092 Fatigue Test

In this experiment the effect of a dry versus humid environment was examined. The 011092 device was nanoindented at Oak Ridge between 9/30/91 and 10/5/91 with 6 indentations at 50 mN with a $3.7\mu m$ spacing between indents. The device was dried to the extent possible by baking for 48 hours at $155°C$ (the limit of the temperature control system) in the test chamber which was continuously purged with dry N_2 ($< 3ppm$ H_20). Following the

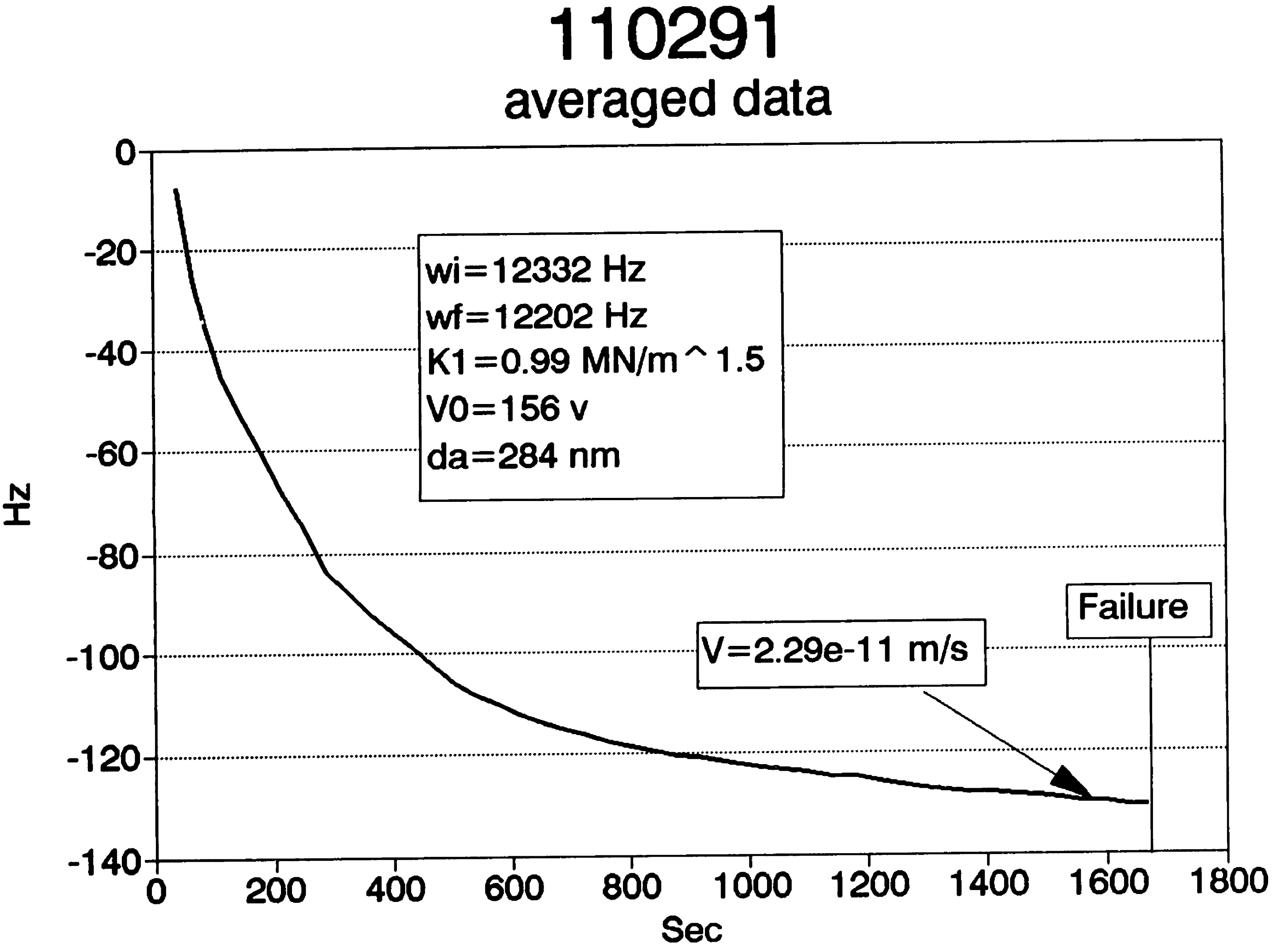


Figure 7.11: 110291 Frequency Versus Time

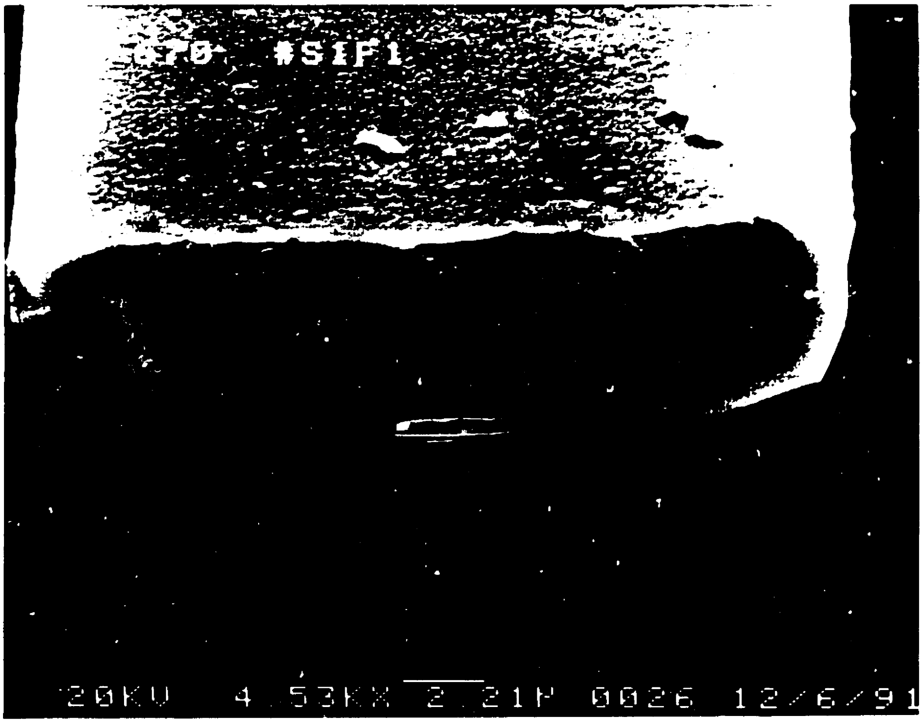


Figure 7.12: 110291 Fracture Surface at 50°

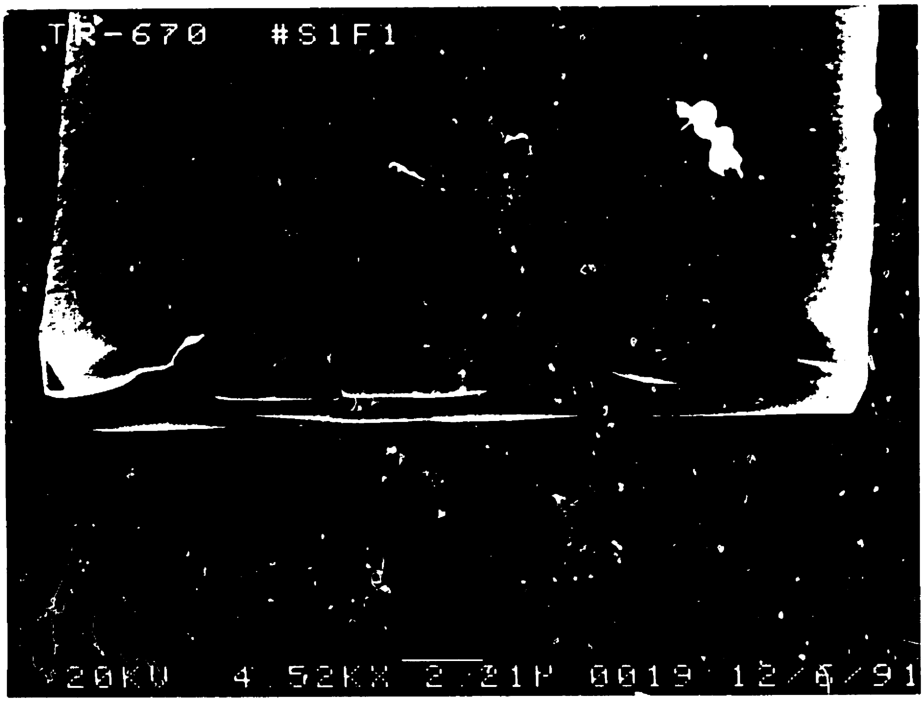


Figure 7.13: 110291 Fracture Surface at 30°

drying, the temperature was reduced to 30°C, and the test begun. Again, the initial rapid transient occurred followed by the frequency leveling. Figure 7.14 is the frequency versus time data for the first run. Note that a low value of stress intensity, ≈ 10% K_{1c} ($K_{1c} = 1\ MN/m^{1.5}$), produced a significant change in frequency. Following the frequency leveling in the first run, the torquing voltage was increased, thereby, raising the stress intensity to approximately 40% of the fracture toughness. This increase was necessary to obtain a relatively noise free output voltage assuring accurate frequency measurements. Again, a rapid initial frequency drop occurred followed by the typical leveling. The device was allowed to resonate until the frequency remained constant for a period of approximately 48 hrs, then moisture was introduced into the environment. Figure 7.15 shows the effect of adding moisture. The frequency immediately dropped in a manner similar to an increase in stress intensity then leveled off to what appears to be a constant downward slope corresponding to a velocity of 2.75×10^{-13} m/s. Over the next 5 days the frequency decreased a total of 42 Hz and the velocity decreased to approximately 4×10^{-14} m/s which is near the limit of resolution of the experiment. Accurate measurements below a threshold of approximately 1×10^{-14} are not possible for extended periods of time because of thermal drift in the electronics.

7.5 Generation 2 Static Break Test

From 9/30/91 to 10/5/91 several generation 2 devices, without bridge electrodes, were nanoindented at Oak Ridge National Laboratory. These devices were etched then remounted and transported back to Oak Ridge for break testing. The nanoindenter was used to statically deflect the cantilevers until failure recording both load and displacement. To ensure the contact point of the nanoindenter and beam was captured, the software controlling the surface find procedure of the nanoindenter was modified to save the last 200 points of the approach data. The purpose of this series of tests was to determine if the load to failure and compliance are being accurately computed by the design program, FTEST2.PAS. The DS3B2 device was fully pre-cracked across the width of the cantilever with an array of six indentations at an indentation load of 45 mN with a $4\mu m$ spacing between indents The 45 mN indentation load corresponds to an indentation depth of approximately

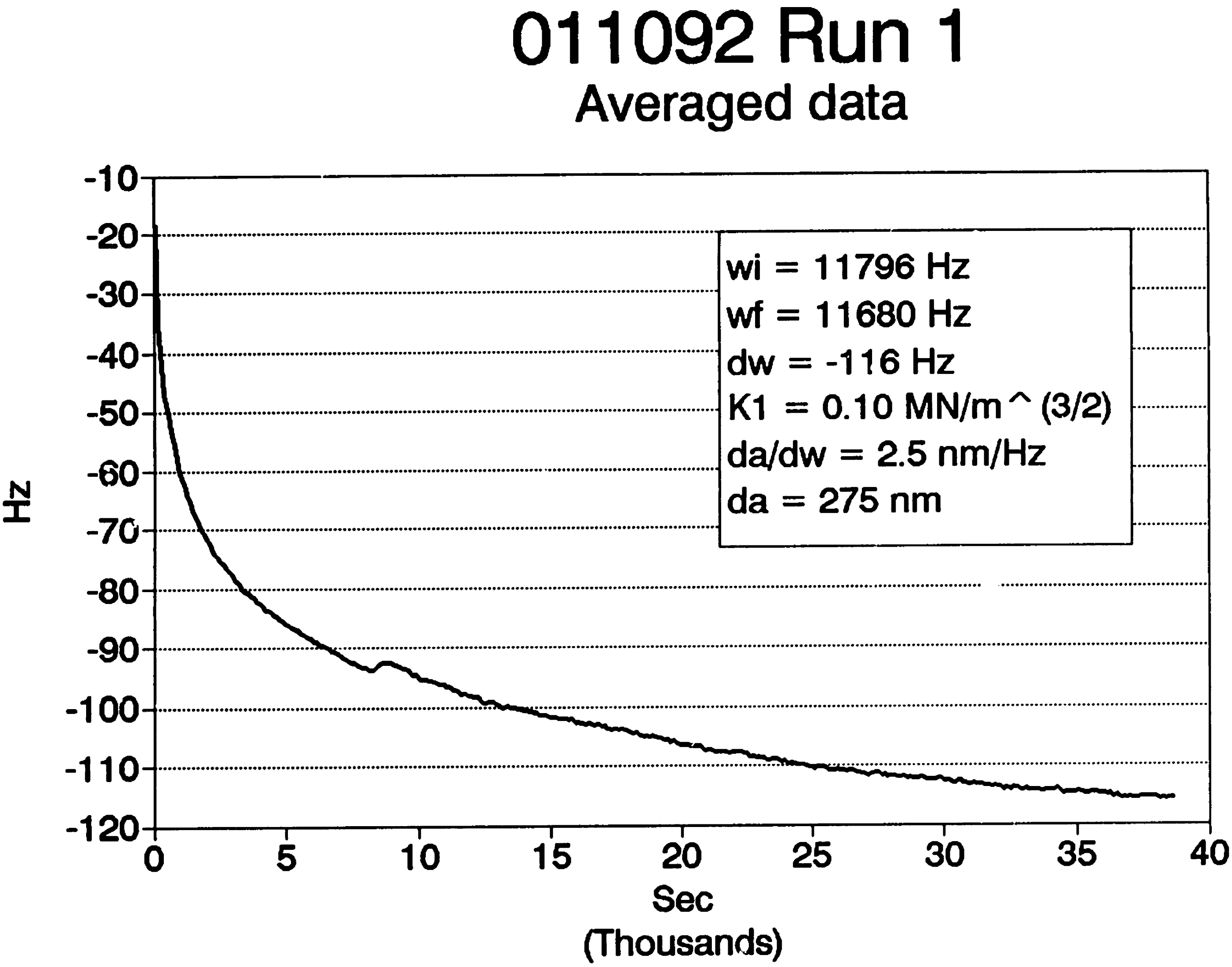


Figure 7.14: 011092 Run 1 (Dry Environment)

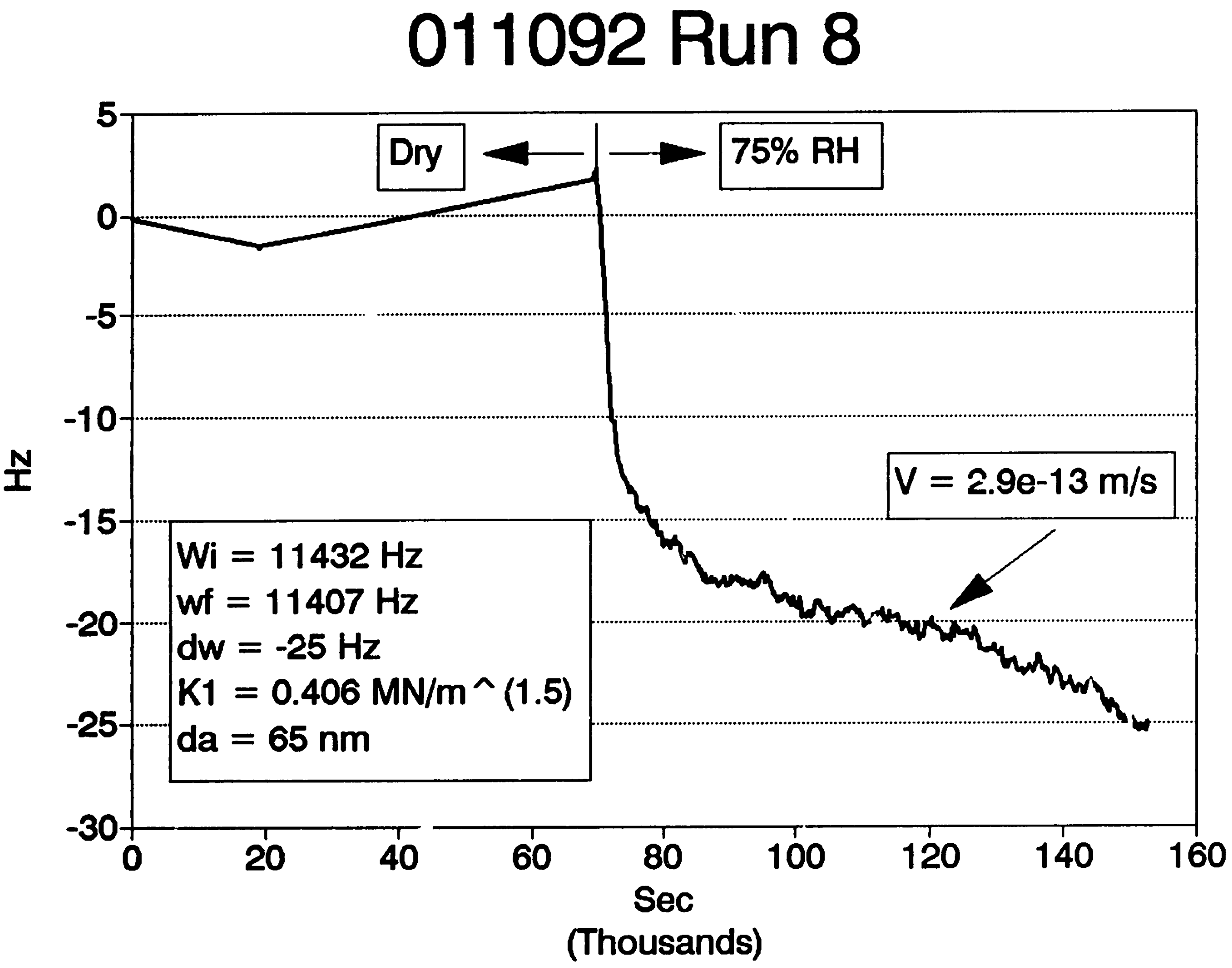


Figure 7.15: 011092 Run 8 (Moist Environment)

$3.3\mu m$. Figure 7.16 shows the fracture surface of the cantilever viewed from an angle of 70° from the vertical. Figure 7.17 shows the fracture surface viewed from a shallow angle of 20° from the vertical. The fracture line is clearly visible in this photo, and is essentially straight except for minor irregularities from the indentations. A similar uncracked device designated as DS3A1 was also deflected so that the uncracked compliance could be used in conjunction with the cracked beam compliance to determine the compliance of the pre-crack alone. Figure 7.18 shows the layout of the two devices and the corresponding cross section. Figure 7.19 is the load versus displacement curve of the uncracked device and Figure 7.20 is the load versus displacement curve of the cracked device. Note that in each case the "knee" of the curve was captured; therefore, the true load applied to the device is known with accuracy. The experimental values of crack compliance and fracture toughness will now be computed and compared to those obtained from Murakami's equations [34]. To compute the uncracked compliance, the compliance of the DS3A1 device is used and corrected for the difference in length.

$$K_{uc} = 6082\left(\frac{24.9}{25.7}\right)^3 \tag{7.1}$$

$$K_{uc} = 5530 \text{ N/M} \tag{7.2}$$

Next, the rotational stiffness at the crack site is computed using the value for the uncracked stiffness and the cracked beam stiffness.

$$K_r = L_a^2\left(\frac{K_c K_{uc}}{K_{uc} - K_c}\right) \tag{7.3}$$

$$K_r = 3.534 \times 10^{-6} \text{ Nm} \tag{7.4}$$

The stiffness computed using Murakami's equations gives: $K_r = 3.48 \times 10^{-6}$ Nm which is 2% less than measured. The fracture toughness is computed as follows.

$$K_{1C} = k(a)M_f \tag{7.5}$$

$$M_f = P_f L_a \tag{7.6}$$

P_f is the failure load recorded by the nanoindenter: $P_f = 2.129$ mN from Figure 7.20. This failure load gives $K_{1C} = 0.71$ MN/m$^{3/2}$ which is approximately 30% less than the reported value of 1 MN/m$^{3/2}$ given in the literature. It is likely the lower value is attributed to the fact the cross section

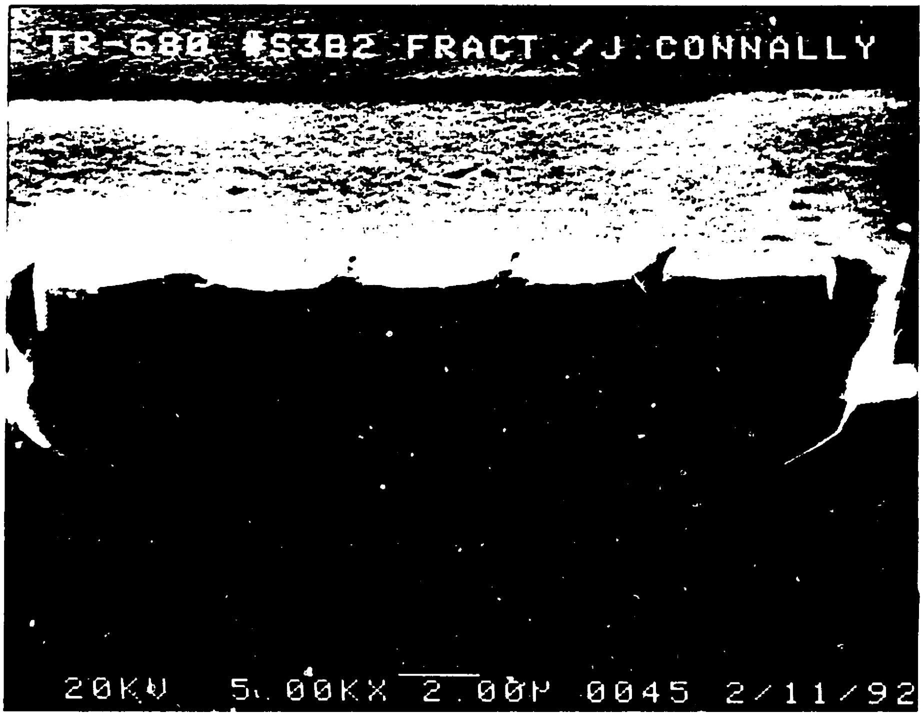

TR-680 #S3B2 FRACT./J.CONNALLY
20KV 5.00KX 2.00P 0045 2/11/92

Figure 7.16: DS3B2 Fracture Surface (70°)

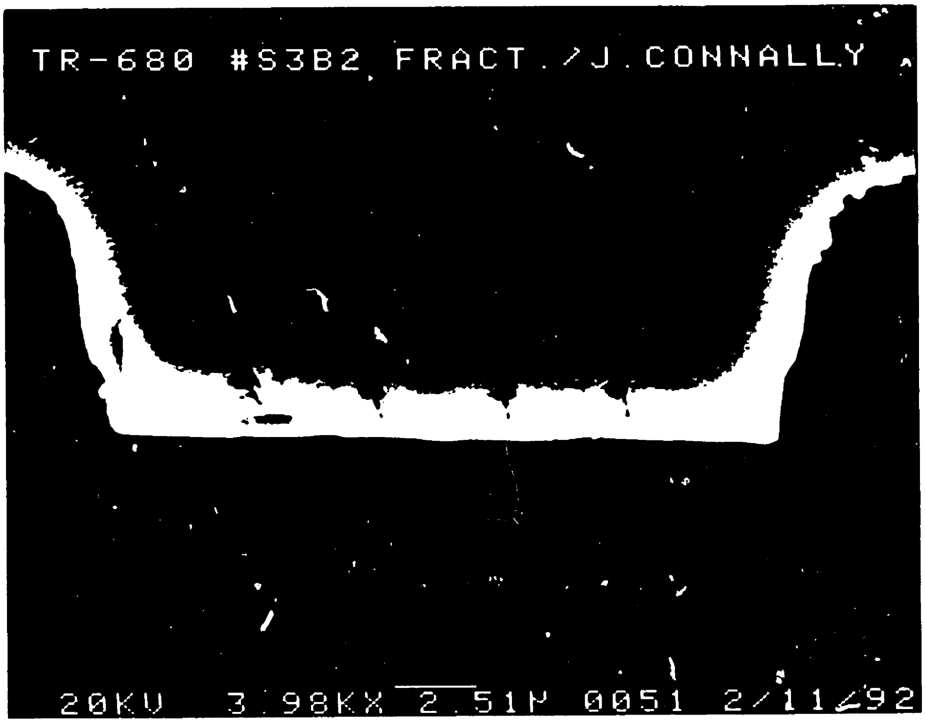

TR-680 #S3B2 FRACT./J.CONNALLY
20KV 3.98KX 2.51μ 0051 2/11/92

Figure 7.17: DS3B2 Fracture Surface (20°)

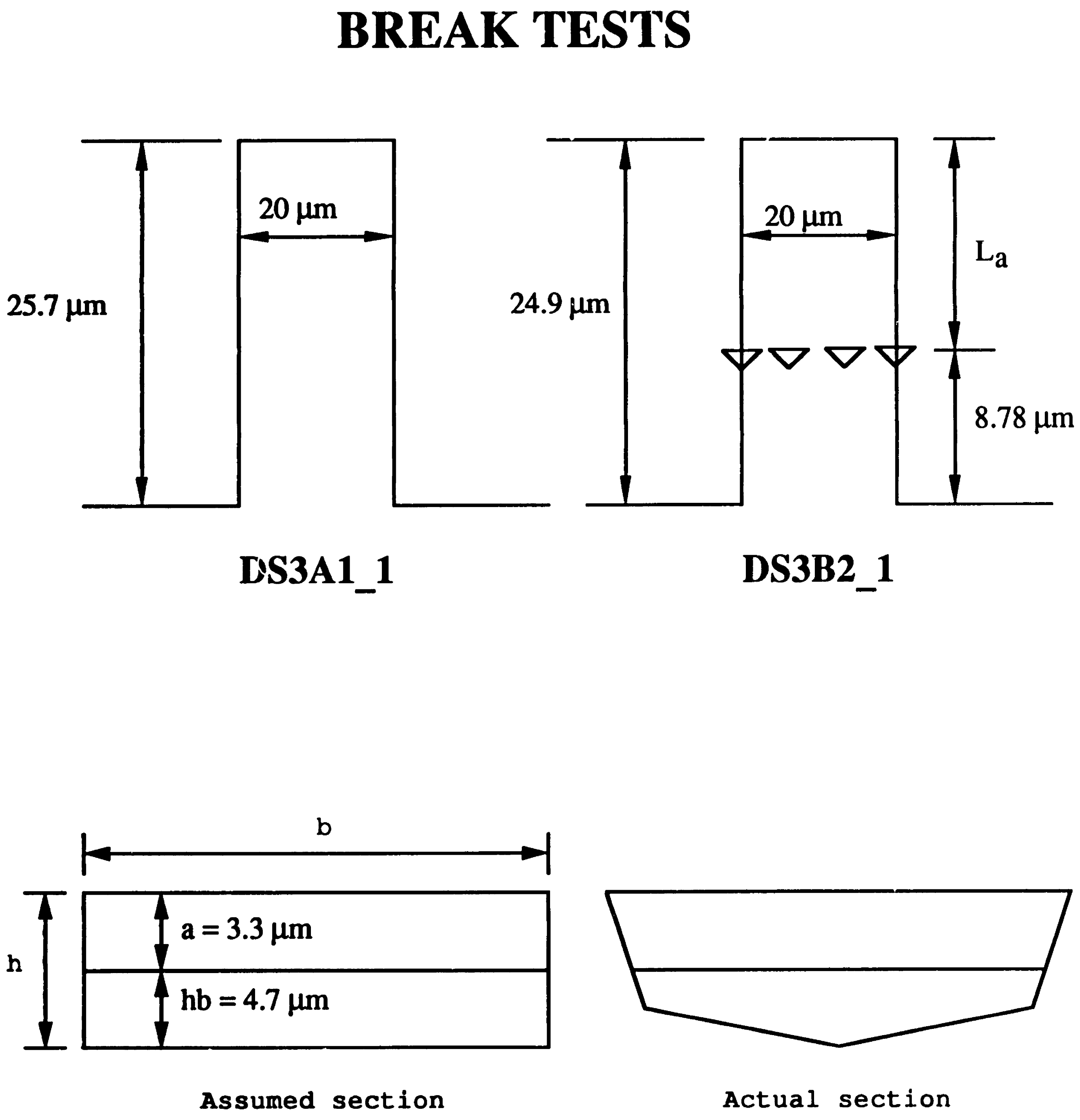


Figure 7.18: Plan View and Cross Section of DS3A1 and DS3B2

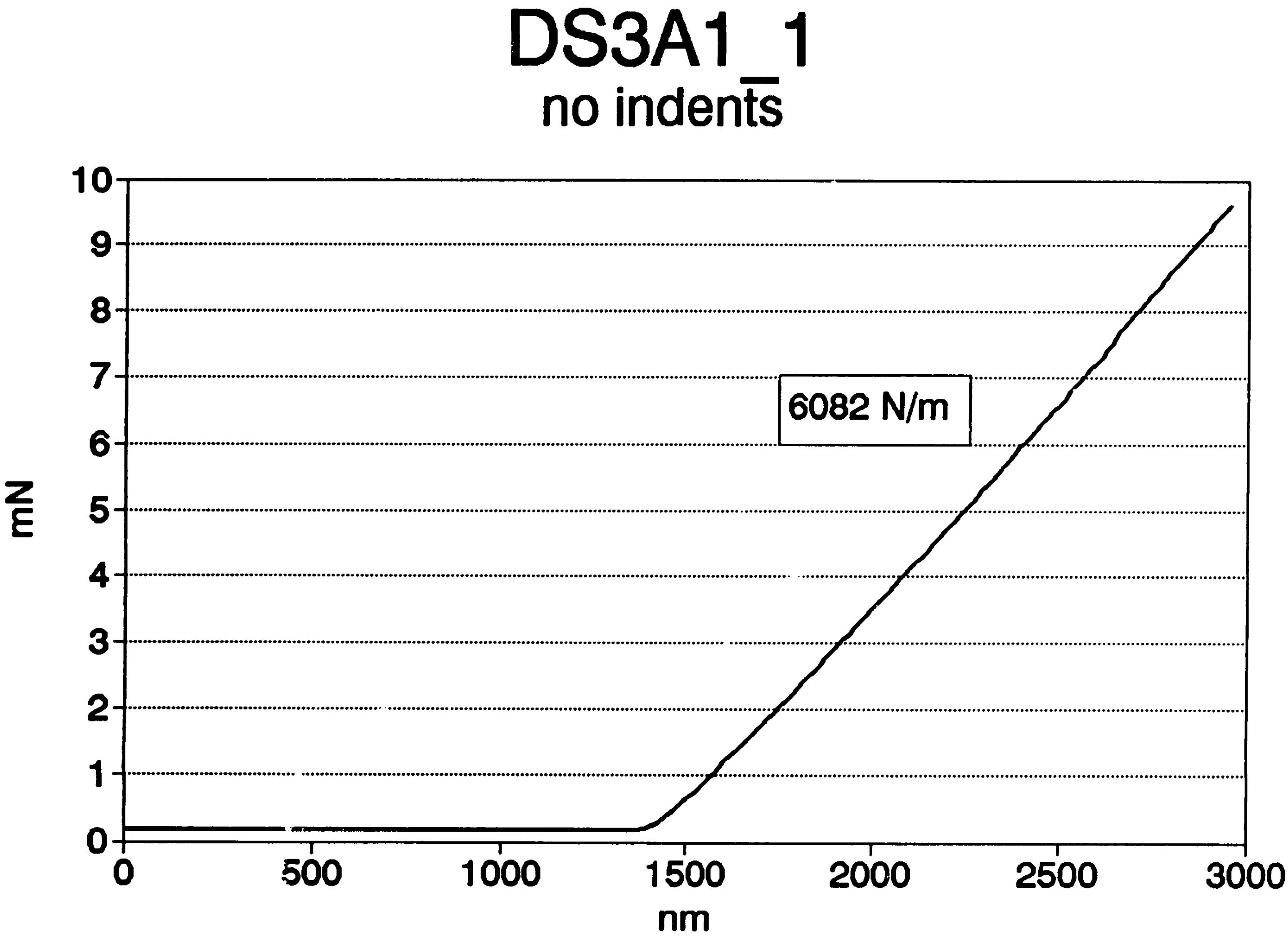


Figure 7.19: DS3A1 Load versus Displacement

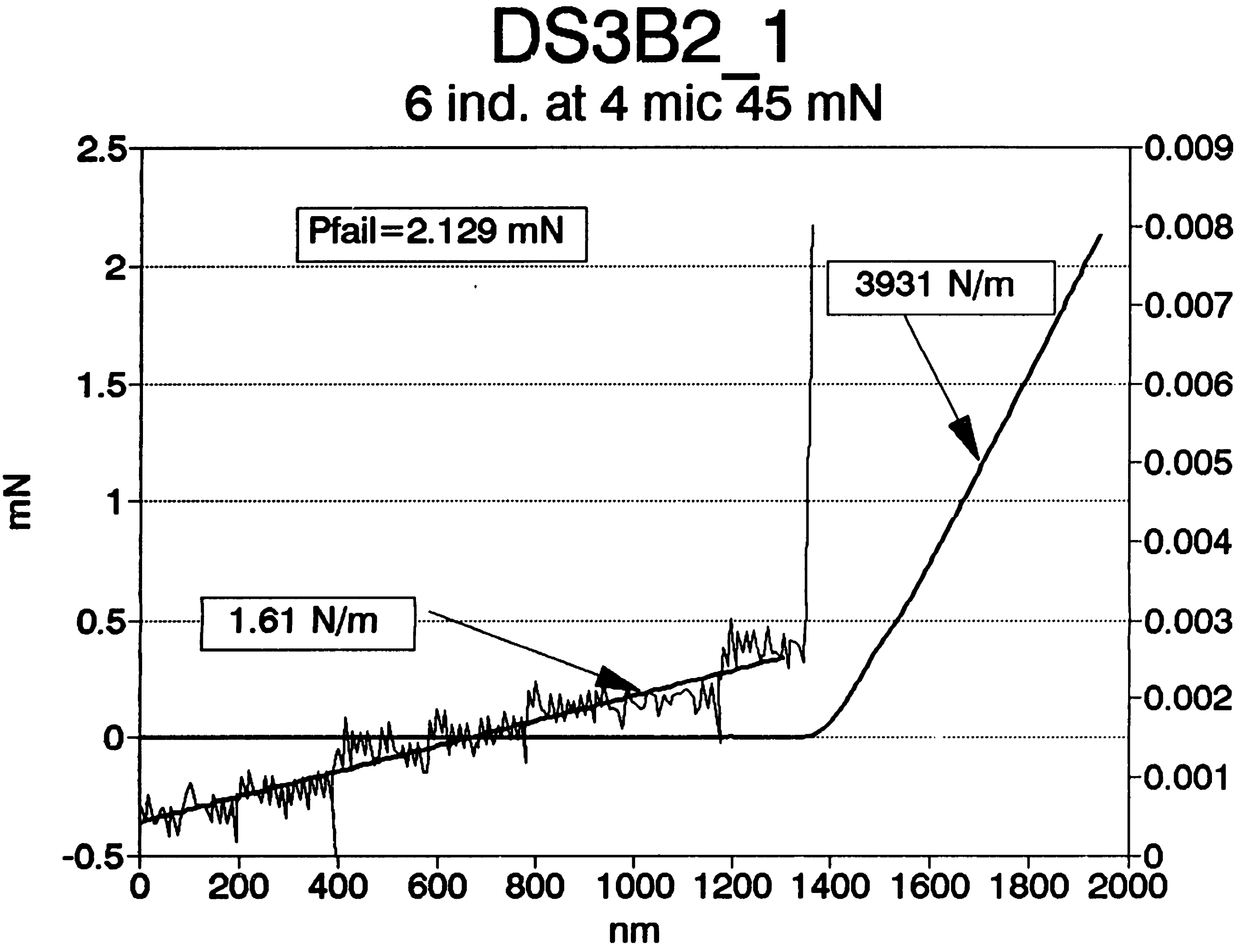


Figure 7.20: DS3B2 Load versus Displacement

of the cantilever is angled and rounded somewhat as shown in Figure 7.18. Murakami's equations were derived for a bar of rectangular cross section; therefore, a higher stress intensity would be computed because the ligament thickness of the angled section is small near the edges.

The results of the break test confirm the validity of the equations being used in the FTEST2.PAS design program to the extent that gross errors are not apparent. Fracture is K_1 controlled. Compliance is accurately modeled, therefore, frequency shifts are accurately correlated to crack extensions. The compliance of the pre-crack is linear. This fact is important because nonlinear spring effects, such as hardening, will create an undesirable frequency sensitivity to amplitude. Such nonlinearties are not present; however, the spring stiffness cannot be measured in the reverse direction. In some devices, particularly those that may not be fully pre-cracked, a frequency sensitivity to amplitude has been measured. A 25 Hz per volt sensor output was measured for the 102591 device. It is possible some hardening effects are present on the upswing as the crack closes and compresses the surrounding silicon. If the damage zone created by the nanoindentations has a nonlinear stiffness characteristic that depends on amplitude, a frequency sensitivity to amplitude will be introduced. In general, for fully pre-cracked beams, a strong frequency sensitivity to amplitude has not been measured.

7.6 020392 Fatigue Test

The 020392 device was nanoindented at Oak Ridge between 9/30/91 and 10/5/91. This device had 6 indentations at 55 mN with a $4\mu m$ spacing between indents. To verify that fracture can be initiated with a K_1 controlled load following extended fatigue, the 020392 device was cycled for 3 days then fractured with the drive electronics recording both displacement and applied voltage. In the first run, the device exhibited the usual fast growth fatigue characteristic followed by the frequency leveling. The stress intensity was approximately 0.45 $MN/m^{3/2}$, and the velocity in the slow growth region was approximately 2.14×10^{-14}m/s. In the second run, the torquing voltage was increased raising the stress intensity to 0.61 $MN/m^{3/2}$. Again the usual fast transient was observed with the corresponding leveling. Although the resolution was poor, a velocity in the slow growth region was measured to be approximately 2.4×10^{-15} m/s. Figures 7.21 and 7.22 are the frequency versus time plots of the data. In the last run, the voltage was increased until failure recording both sensor output and torquing voltage. Using the FTEST2.PAS design program and careful measurements of the device geometry with a microscope and filar, the failure displacement corresponds to a fracture toughness of 0.67 MN/m$^{3/2}$. This number is close to the value of 0.71 MN/m$^{3/2}$ recorded during the static break test with the nanoindenter. The significance of this result indicates that extended cycling does not reduce the stress intensity dramatically, and therefore, rules out two possible explanations for the rate limiting process. The first explanation is crack tip blunting whereby the crack tip rounds to the extent that a K_I dominant stress field is no longer obtained, and consequently, causes the corrosion mechanism to slow or stop. The second explanation is that the crack extends and changes its geometry in a manner that causes the stress intensity to decrease, and again, causes the corrosion mechanism to slow or stop. These explanations are unlikely for two reasons: first, the total frequency shift for runs 1 and 2 corresponds to a crack extension of 19 nm which would not appreciably affect the pre-crack geometry; second, much larger displacements than measured would have been required to fracture the devices if either of these mechanisms were operating. Therefore, the rate limiting process observed in these experiments should not be interpreted as a reduction in stress intensity.

7.7 Observations of Generation 2 Fatigue Testing

The principal observations of these experiments are as follows.

1. Nano scale crack growth in heavily boron doped single crystal silicon micromechanical devices has been measured. Crack extensions on the order of 200 to 300 nm have been measured with a resolution of approximately 1 part in 100. These extensions are on the order of 20 to 30 times the 10 to 20 $\AA$ oxide film that forms when freshly cleaved silicon is exposed to a moist room temperature environment.

2. The fatigue mechanism is a fundamentally rate limiting process. Starter pre-cracks do not exponentially grow with stress intensity ultimately causing failure. However, failure can be caused if the starting stress

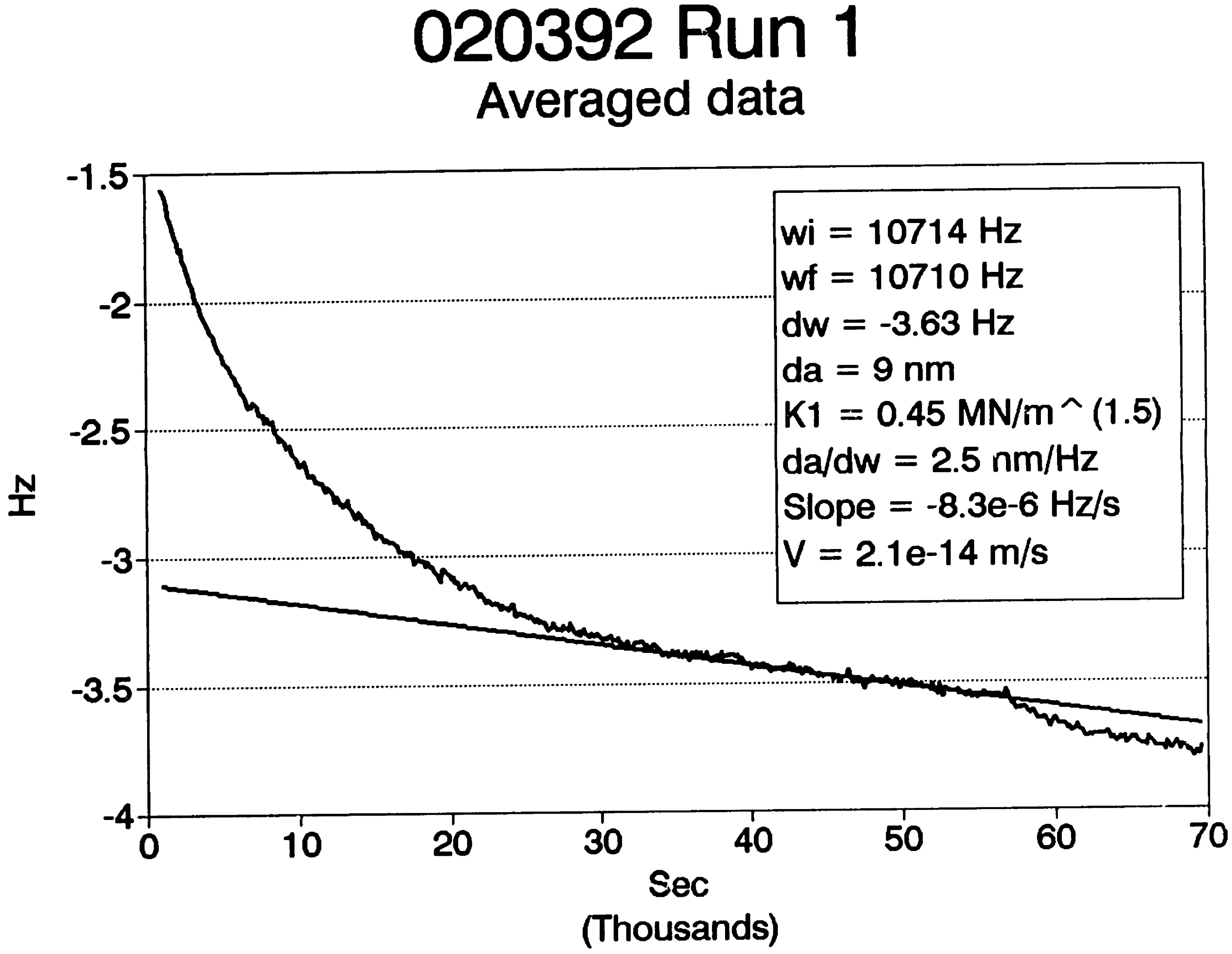


Figure 7.21: 020392 Run 1

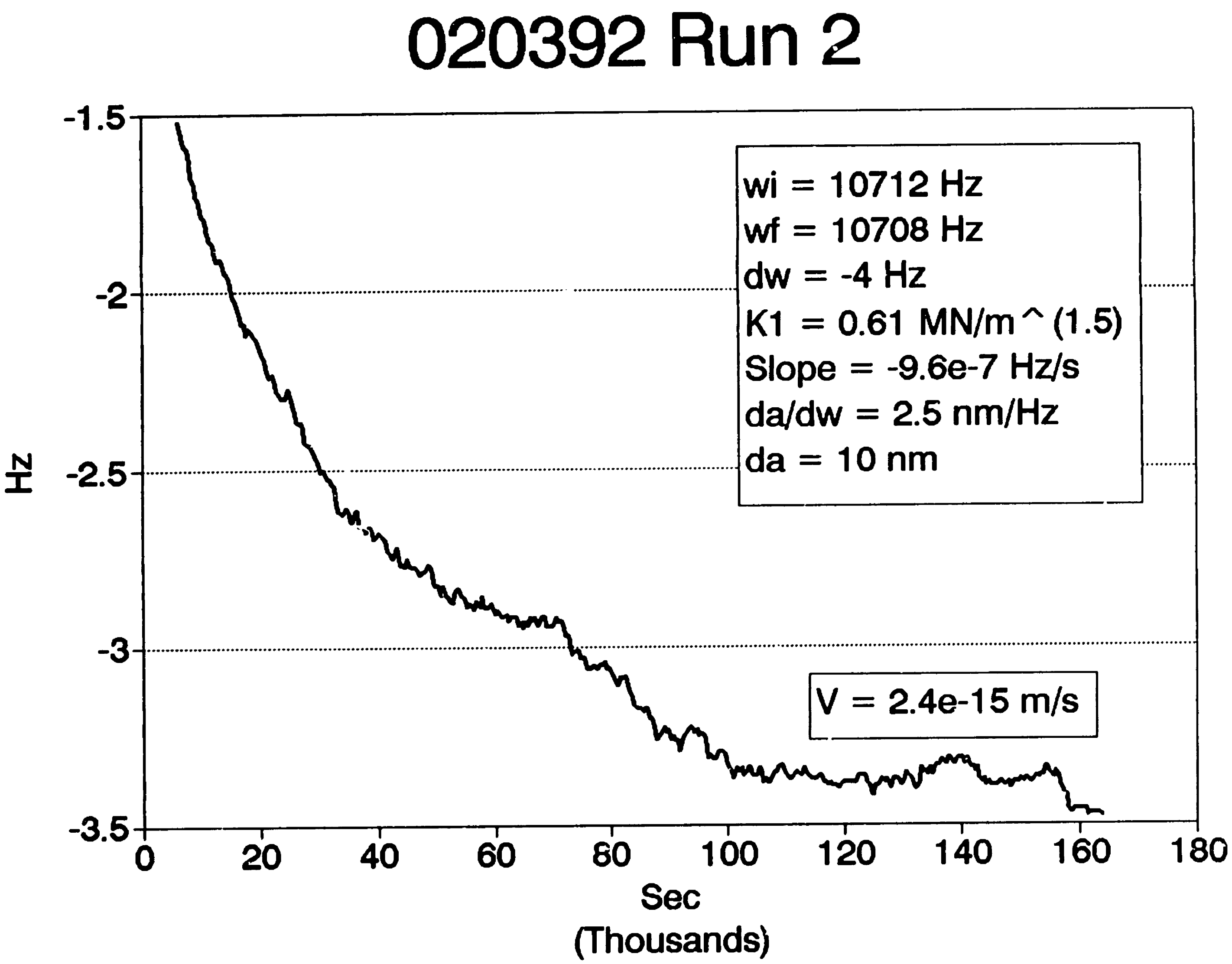


Figure 7.22: 020392 Run 2

intensity is very near the fracture toughness. Initially, crack growth occurs rapidly with velocities on the order of 1.0×10^{-9} m/s, but exponentially decays to velocities that cannot be accurately measured. The lower bound on accurate measurements is approximately 1.0×10^{-14} m/s.

3. Step increases in stress intensity cause similar fast growth transients followed by the exponential leveling off. Step changes in the moisture content also produce fast growth transients, similar in character to those obtained by increases in stress intensity alone, followed by a decaying velocity.

4. The crack growth characteristics are repeatable. Similar fatigue characteristics have been observed in separate devices from the same wafer, and with separate devices from a different wafer processed independently.

5. Similar rate limiting crack growth behavior has been observed in devices with resonant frequencies of 7 KHz and 12 KHz respectively, and stress intensities varying from $30\% K_{1C}$ to $97\% K_{1C}$. High stress intensities do not alter the rate limiting process, or introduce an exponentially increasing crack velocity. Failure, when it occurs, is instantaneous.

6. The observed fatigue crack growth appears to be induced by corrosive species in the environment, especially water. A step increase in humidity, from $\approx 0\%$ to $\approx 75\%$ RH, produced a distinctive fast crack growth transient followed by a long decay in crack velocity. The crack extensions are on the order of 50 times the thickness of the 20 to 40 angstrom silica layer initially present indicating that a corrosion/oxidation mechanism, similar to that previously described, does operate for a finite period of time.

7. The exponentially decreasing crack velocity in the fast region may be intrinsic to the rate limiting mechanism rather than just the crack front equilibrating. The crack extension obtained in the 110291 test is too small to substantially change the uniformity of the starting rack front. However, the precise extent to which minor irregularities in the crack front affect fast crack growth is unknown. Crack front jumping,

which are clearly distinguishable step changes in frequency, have been measured in the fast region. These step changes are probably the crack front equilibrating.

8. Crack growth in the slow growth region appears to be independent of stress intensity as observed in the experiments with the 102491 device. Table 7.1 shows crack velocities in the slow growth region are on the order of 1.0×10^{-13} m/s even though the stress intensities differ significantly.

9. Fracture mechanics applies on a micro-scale. The static break tests of the generation 2 devices gives reasonable correlation with the predicted failure load, and good correlation with compliance. No rounding in the load versus displacement curve was measured suggesting the absence of measurable plastic deformation.

10. The rate limiting process is not caused by a reduction in stress intensity. Following extended cycling, the dynamic break test of the 020392 device correlates with the predicted failure displacement assuming K_1 controlled fracture. Crack tip blunting, or other K_1 reducing mechanisms, do not appear to occur to a significant extent .

Chapter 8

Summary

This experiment was conceived and executed for two purposes: first, to introduce and demonstrate a novel fatigue testing technique particularly suited for micromechanical testing; and second, to utilize the technique in an experimental investigation of nano scale fracture and fatigue of silicon based micro electro-mechanical devices. The technique has been proven: micromechanical designers and material scientists now have a vehicle through which micron size test structures processed in various ways can be fractured and fatigued in a range of environments, temperatures, and pressures. New information and insight into the failure mechanisms of single crystal silicon micromechanical devices has been obtained and presented.

The initial objective of the experiment was to design and construct a micromechanical fatigue test structure, and the associated systems, such that the required experimental data could be obtained. The cantilever structure was chosen for simplicity of fabrication and the relatively straightforward dynamic modeling. An accurate model of the device, and the pre-crack, is important as an analytical aid in designing the device for optimal performance, and in the analysis of predicted and measured frequency shifts during operation. Many tradeoffs were necessary both in the electronics and in the mechanics of the device to achieve a successful design. The critical tradeoffs are enumerated as follows.

1. Noise in the sensing electronics was a major consideration in designing the interface electronics and in setting the electrode air gap dimensions. A large signal to noise is desirable so that the zero crossings of the sensor

output signal can be precisely determined. The signal to noise ratio is defined as the sensor output voltage, assuming zero noise, divided by the RMS sensor output assuming the device is not moving and only noise is present. The smaller the air gap, the higher the signal to noise ratio; however, small air gaps limit the travel of the end plate and the stress at the crack site. Large air gaps, $> 12\mu m$, are limited by the fabrication process.

2. Low resonant frequencies, $\approx 3KHz$, are desirable because the frequency of the carrier signal can be kept near 100 KHz which minimizes phase shift in the post amplifiers. This phase shift translates into signal attenuation following demodulation, and therefore, reduces the signal to noise ratio. However, a low resonant frequency implies that the structure is compliant; consequently, large displacements are required to generate adequate stress at the crack site. Again, the air gap limits the total travel of the end plate.

3. The thickness of the cantilever is limited by the boron diffusion time. A thick beam is desirable because a large ligament thickness is possible at the crack site making the structure resistant to unintentional failure caused by surface tension forces during etching and rinsing, or vibration transmission from ultrasonic wirebonding. However, a very long diffusion time severely weakens the field oxide mask by destroying the dielectric properties of the oxide and by creating pin holes. An inevitable consequence of a thick beam is a stiff structure which requires a large driving voltage to obtain adequate stress at the crack site and an acceptable signal to noise ratio. Voltages that are too high break down the weakened oxide, particularly at the pin holes, resulting in the electrodes shorting to the substrate. A 13 hour diffusion results in a beam thickness of approximately $8\mu m$. Based on experience, a $1\mu m$ field oxide subjected to this boron diffusion can withstand 100 volts indefinitely about 50% of the time.

4. To generate adequate stress at the crack site, given the displacement constraint at the bridge electrodes, a stiff structure is necessary. Aside from increasing the thickness, the length of the beam can be shortened. However, clearance for the nanoindenter must be maintained at the bridge electrodes. Given a maximum plating height of $15\mu m$ and a

maximum electrode air gap of $12\mu m$, the length is limited to no less than $75\mu m$. The same cautions about excessive voltage and signal to noise ratio apply.

5. The plating height of the counterweight and bridge electrodes is limited by the fabrication process to approximately $15\mu m$. Larger plating heights are possible, but require more complex processing. Maximizing the plating height is desirable because the resonant frequency is reduced, and the voltage necessary to drive the device to a given amplitude at resonance is reduced. A further advantage is an increase in the stiffness of the bridge electrodes which eliminates buckling during the low temperature $360°C$ sinter. This sinter is a standard processing step that improves metal adhesion.

Other tradeoffs and modeling tasks were important such as the dynamics associated with flexing of the end plate, torsional modes, out of plane bending modes, bridge electrode vibration, and especially damping. Low viscous damping at atmospheric pressure is necessary for two reasons. First, a predominately AC displacement must be generated so that large tensile stresses are applied at the crack site. At resonance, the AC amplitude depends inversely on damping. Second, low driving voltages, < 100 volts, prevents accumulative damage to the oxide and minimizes the DC force that pulls the plate toward the electrode (recall that the electrostatic force depends on the voltage squared). Therefore, modeling and minimizing damping is important. To accomplish these tasks, and perform the tradeoffs, numerous repetitive calculations are involved; therefore, a design program was written in PASCAL incorporating all aspects of modeling and analysis. This program is designated as FTEST2.PAS.

Finite element modeling of the structure and numerical simulations of the crack nonlinearity proved the accuracy of the equations being used in the design program. Break tests and compliance measurements with the nanoindenter also verified that the equations being used to compute the rotational stiffness of the crack and the stress intensity are not introducing gross errors. Therefore, a high degree of confidence exists in correlating the frequency data to crack extension, and in determining the accuracy of the measurement technique. Uncracked devices have been continuously run for periods up to 1 week with frequency shifts less than 1 Hz which indicates the stability of the sensing system.

Two important parameters are determined from the FTEST2.PAS design program in determining the fatigue history of a device: crack extension, δa, and stress intensity, K_I. The total length of the pre-crack can be determined in several ways: the baseline frequency of the cracked device, the indentation load of the nanoindenter, and measurements from SEM photos of the fracture surface once the experiment is complete. However, the crack extensions measured are too small for adequate resolution with the SEM, and therefore, must determined from the frequency data using the dynamic model. K_I is also determined from the dynamic model and the measured amplitude of motion. The method of computing δa, K_I, and the measurement error is now summarized.

To analyze a particular device, the geometry is first determined by measuring all dimensions of the device. Planar dimensions are those specified on the photomask and are accurately measured with the microscope and filar. The actual planar dimensions are approximately 3 to 5 μm greater than the photomask dimensions because during processing the boron diffusion occurs outward, as well as downward into the silicon. Air gap dimensions and plating heights are determined by depth of field measurements with the microscope. The thickness of the device is determined from a test wafer diffused simultaneously with the device wafer. After diffusion, the test wafer is stained to reveal the extent of the boron diffusion then etched. An accurate measurement of the etch stop depth is made with the microscope and filar. The exact thickness of the device under test cannot be measured until the experiment is complete and the device cross sectioned for SEM analysis; however, etch stop depths generally do not vary by more than 5% on a given wafer.

The measured dimensions of the device are input to the design program including the nanoindentation load which can be related to the pre-crack depth through Equation 5.5 or post SEM analysis of the fracture surface. The natural frequency and amplitude of motion of the pre-cracked device is then measured and correlated to the dynamic model.

The stress intensity is computed by first determining the moment applied at the crack site. The moment is computed from the polynomial which approximates the beam deflection and the first mode solution of the dynamic model.

$$v_b(x,t) = v_1(t)\left(\frac{x}{\ell_{ac}}\right)^2 + v_2(t)\left(\frac{x}{\ell_{ac}}\right)^3 + \theta(x-\ell_{ab}) \qquad \ell_{ab} \le x \le \ell_{ac} \qquad (8.1)$$

$$M_0(x,t) = EI\frac{\partial^2 v_b(x,t)}{\partial x^2} \tag{8.2}$$

$$M_0(\ell_{ab},t) = \left[\begin{array}{ccc} \frac{2\,EI}{\ell_{ac}^2} & \frac{6\,EI}{\ell_{ac}^3} & 0 \end{array} \right] \underline{\mathbf{U}}(t) \tag{8.3}$$

$\underline{\mathbf{U}}(t)$ is the vector of generalized coordinates, $v_1(t)$, $v_2(t)$, and $v_3(t)$. These coordinates are determined from the first mode solution of the equations of motion.

$$\underline{\acute{\mathbf{M}}}\,\underline{\ddot{\mathbf{U}}} + \underline{\mathbf{C}}\,\underline{\dot{\mathbf{U}}} + \underline{\acute{\mathbf{K}}}\,\underline{\mathbf{U}} = \underline{\mathbf{P}}_0 \tag{8.4}$$

The displacement vector is a superposition of the three modes of the system model. Because the structure is driven at the first mode resonance, the first mode solution accurately approximates the true motion of the structure. The higher modes in the 2D Lagrangian model are spurious. Recall that a 3D finite element analysis shows that the next two modes are out of plane bending and torsion which was not included in the dynamic analysis.

$$\underline{\mathbf{U}} = \sum_{i=1}^{3} \underline{\phi}_i\, x_i(t) \approx \underline{\phi}_1\, x_1(t) \tag{8.5}$$

Dropping the subscripts, the solution of the system at the first mode resonance in steady harmonic motion is given as follows.

$$x(t) = \frac{-p_0 \sin(\omega_n t)}{c\omega_n} \tag{8.6}$$

Therefore, the moment is computed as follows.

$$M_0(\ell_{ab},t) = \left[\begin{array}{ccc} \frac{2\,EI}{\ell_{ac}^2} & \frac{6\,EI}{\ell_{ac}^3} & 0 \end{array} \right] \underline{\phi}_1 \frac{-p_0 \sin(\omega_n t)}{c\omega_n} \tag{8.7}$$

Using either Murakami's equation or the deep cracked beam equation the stress intensity is computed as follows.

$$K_1 = k(a)\, M \tag{8.8}$$

The accuracy of the equation for M_0 is contingent on the natural frequency, ω_n; the damping, c; and the forcing function, p_0. The natural frequency is measured directly with the frequency counter and accurately correlates with the dynamic model. The accuracy of the model was verified by linear finite element analysis, nonlinear numerical simulations, and the compliance data from the break tests with the nanoindenter. The equation for the forcing function, p_0, is given below.

$$p_0 = \underline{\phi}_1^T \underline{\eta} \frac{\epsilon_0 A_T V_{T0}^2}{4 d_0^2} \tag{8.9}$$

where,

ϵ_0 = permittivity of a vacuum: $\left(\frac{\text{coul}^2}{\text{N m}^2}\right)$

d_0 = air gap: (m)

V_{T0} = magnitude of driving voltage: (volts)

A_T = torquer electrode area: (m^2)

$\underline{\eta}$ = vector of length ratios defined in Section 2.3

The driving voltage, V_{T0}, is measured directly with a digital volt meter. The electrode area, A_T, is specified by the mask dimension and can be measured with the microscope and filar. Therefore, the accuracy of p_0 depends primarily on the air gap dimension, d_0. The air gap is approximately $12\mu m$ and can be measured to $\pm 1\mu m$ using depth of field.

Because the pre-crack introduces additional damping that is difficult to characterize, the principal uncertainty in determining M_0 is the damping constant, c. Therefore, this parameter is adjusted in the model until the measured sensor output agrees with the calculated value. The expression for the sensor output is given by Equation 3.31 and is restated below.

$$V_0 = -k_s \frac{v_s(t)}{1 - \frac{v_s(t)}{d_0}} \tag{8.10}$$

where,

V_0 = sensor output: (volts)

k_s = sensor constant: (volts/m)

$v_s(t)$ = $-v_{s0} \sin(\omega t)$ plate displacement under sensor: (m)

v_{s0} = $\underline{\xi}^T \underline{\phi}_1 \frac{p_0}{c\omega_n}$

$\underline{\xi}$ = a vector of length ratios defined in Section 2.3

The sensor constant can be determined experimentally by measuring the output of the post amplifiers for the zero displacement position of the device, and by inputting known test signals into the sensing electronics and measuring the output. Therefore, the sensor output is a measure of the same parameters that determine the moment. The accuracy also depends on the air gap, d_0. The sensor output voltage error is determined by differentiating V_0 with respect to d_0 and normalizing. Performing this differentiation and simplifying gives the following expression for the error in the displacement measurement.

$$\frac{d\,V_0}{V_0} = \frac{3\delta d_0}{d_0 - v_{s0}} \tag{8.11}$$

δd_0 is the uncertainty in the air gap dimension. For $\delta d_0 = 1\mu m$, $d_0 = 12\mu m$, and $v_s = \pm 4\mu m$, the error in the displacement measurement is 37% as the end plate nears the electrode, and 19% on the downswing. The downswing measurements were used in calibrating the damping at the crack site. The error analysis indicates that M_0 and K_I are known with an accuracy of approximately 20%. However, the amplitude at failure and the driving voltage at failure are two additional pieces of information that can be used with the dynamic model to calibrate the range of values of K_I applied to the device during the test runs. Based on values reported in the literature, fracture in silicon occurs when $K_{Ic} = 1$ MN/m$^{3/2}$. The data from the static break tests indicate that fracture occurs when $K_I = 0.71$ MN/m$^{3/2}$ as computed using Murakami's equation. However, this equation was developed for a rectangular beam with a constant ligament thickness, and therefore, probably underestimates the true stress intensity near the ends where the ligament thickness is smaller. Because all of the devices have similar crack geometries, beam cross sections, and indentation depths, Murakami's equation can be used with some degree of confidence to calibrate how close a particular device is to fracture. The results of the dynamic break test of the 020392 device demonstrate the correlation of predicting fracture using the dynamic model and Murakami's equation to the value obtained from the break test. The two numbers for K_I agreed within 7%. Therefore, given the geometry of the device to measurement accuracy, the resonant frequency, the driving voltage at failure, the amplitude at failure, and the break test data as a reference point for determining when K_{Ic} is exceeded, the dynamic model using Murakami's equation provides an estimate of the true stress intensity of approximately 10% to not more than 20%.

The crack extension, δa is computed by incrementing the crack length in the equation for the compliance of the pre-crack, $C_\theta(a)$, then solving for the frequency directly from the modal analysis in the design program. Damping, air gap dimensions, voltages, amplifiers, demodulator and filter gains are not involved in this computation. The accuracy of calculating the exact amount of crack extension is determined by how accurately Murakami's equation, or the deep cracked beam equation, models compliance; and the accuracy of the dynamic model. The break test data indicates that Murakami's equation for compliance is modeled with an accuracy of 2%. Finite element analysis and numerical simulations also verified the equations of motion used to model the structure are accurate. Therefore, the calculated values of crack extension are believed to be a close approximation of the true extensions. To correlate crack extension to frequency with greater precision requires extensive modeling of the true crack geometry using finite elements or other suitable modeling software. It should be noted that the actual crack advance is not measured. Before and after crack length measurements are not possible with the cantilever geometry, and post failure SEM analysis does not have the resolution necessary to unmistakably determine the geometry of the crack advance; therefore, an assumed crack advance would also be involved in using finite element analysis.

Chapter 9

Conclusions and Recommendations

Slow crack growth in heavily boron doped single crystal silicon has been unmistakably measured. The fatigue characteristic is a fundamentally rate limiting process that is affected by the presence of a corrosive species in the environment. Water which is both aggressive and ubiquitous in most operating environments has been shown to dramatically increase crack growth in single crystal silicon micromechanical devices. Failure can be caused by crack extension, and therefore, is an isssue in assesing the reliability of a silicon based micromechanical device. Fracture mechanics theories apply on a microscale; however, theories governing crack growth rates and mechanisms may need to be rethought. Fatigue crack growth in silicon and silica macro scale double torsion specimens exhibit classical static fatigue crack growth where crack velocity exponentially depends on K_I; however, this does not appear to be the case for silicon on a micro-scale.

The exact nature and cause of the rate limiting process remains unresolved. The issue is complex and requires extensive study and alternative avenues of experimentation to determine the controlling factor. Three major catagories of study are enumerated and discussed as follows.

1. Surface chemistry effects:

   (a) Kinetics of crack growth in silica.

   (b) Kinetics of oxide formation.

2. Crack closure effects.

3. Size effects.

Surface chemistry effects are present and do play a role enhancing static stress corrosion fatigue of silicon. The experimental data of the 011092 device exhibited a strong sensitivity to moisture in the environment which, we postulate, is caused by a corrosion/oxidation mechanism whereby a crack propagates through the surface silica layer exposing fresh silicon that is then instantly oxidized and subjected to chemical attack propagating the crack further. A rate limiting process is observed; however, by assuming the chemical kinetics of crack growth in silica and oxide formation are mutually exclusive processes that cannot occur simultaneously, one is faced with the ambiguity of sustained macro-scale static fatigue of silicon [2]. A size effect is implicated, however, it is not certain at this point that this, or some other rate limiting mechanism, is the controlling factor. The two possibilities, crack closure and size effects, should be rigorously examined.

A size effect is present if the laws governing static fatigue crack growth are fundamentally different on a micro-scale than on a macro-scale. To eliminate the complexity of the chemical kinetics associated with oxide formation, a micromechanical fatigue device can be constructed from pure $Si0_2$. The fabrication processes for silica micromechanical devices exist, and a candidate design is discussed in the next section. Fatigue of silica glasses has been extensively studied and a large body of literature is available concerning the kinetics of crack growth in macro-scale silica specimens. Therefore, a solid framework for comparison of macro versus micro static fatigue is established. To unambiguously determine if a size effect is present, the device should be designed such that only tensile stresses are applied to the crack site so that no crack closure occurs. If a rate limiting process occurs in the silica device, a size effect has been established and this becomes the focus of the research. If, on the other hand, micro-scale static fatigue replicates macro-scale fatigue, then a dynamic fatigue experiment should be the next step where the two crack faces come into contact. If a rate limiting process occurs, then one has established that crack closure causes the static fatigue process to cease. If no rate limiting process is measured, then crack closure is not an issue.

If no rate limiting mechanisms are found with the pure silica devices, we proceed as before with identical silicon structures. The first experiments involve large static loads where the two crack faces do not come into contact.

If a rate limiting process is observed, the issue is one of understanding the chemical kinetics at the crack tip on a micro-scale. Experiments utilizing highly controlled environments, and if feasible, *in situ* experiments utilizing surface chemistry analysis such as Auger can be considered. Because macro-scale fatigue of silicon has been demonstrated, a size effect that is unique to silicon may be established. However, additional macro-scale static fatigue experiments are also probably in order to verify that silicon does in fact exhibit macro-scale fatigue. If no rate limiting process is observed in the micromechanical structure, dynamic fatigue experiments where the crack faces come into contact should attempted next. By elimination, a rate limiting process should be observed. The differences and ambiguities between the silica and silicon devices would need further study.

The resolution of this experimental technique has allowed for the measurement of crack extensions on the order of 2 to 3 nanometers and crack growth velocities as low as 1.0×10^{-14} m/s. No other experimental techniques or measurements reported in the known body of literature has matched the resolution and accuracy obtained. Prior to this thesis, virtually nothing was known about fracture or fatigue on a micro-scale, nor had any experimental techniques been developed to address the reliability of micromechanical devices from a unified standpoint with the level of sophistication necessary to produce meaningful results. This research has accomplished both these goals, and represents an important advance in obtaining new insight into the fatigue and fracture characteristics of silicon based micromechanical devices. The experimental technique has been proven to be an accurate and viable approach that can be extended to polysilicon, silica, and metal structures.

Although the cantilever was a successful concept, a number of limitations have been identified and are listed below.

1. Crack length and extension cannot be measured directly with a microscope, rather, these variables are inferred from the indentation load, initial frequency, subsequent frequency shifts, and post failure SEM examination.

2. The capacitance air gap cannot be measured directly and is constrained by the fabrication process to about $12\mu m$; therefore, the maximum amplitude of the device is limited by a process constraint rather than a design specification and some uncertainty exists in determining the actual amplitude of the device during operation.

3. The pre-crack is generated by a series of indentations which does not produce a uniform crack front and introduces damage at the surface of the cantilever. The damage zone may be responsible for a frequency sensitivity to amplitude, and the nonuniform crack front makes it difficult to attribute the fast growth transient exclusively to the crack front equilibrating, or to an intrinsic property of the rate limiting mechanism.

4. The beam is not rectangular in cross section; therefore, given a constant indentation depth, the ligament thickness varies across the width of the cantilever producing a variable stress intensity. Modeling this variation is not simple.

5. DC loads cannot be applied to the structure that will put the pre-crack in a constant state of tension. A fatigue experiment utilizing resonant frequency measurements where the two crack faces do not come into contact is not possible.

Figure 9.1 is the plan view of a new fatigue/fracture test structure which avoids the problems described above and is considered an advance over the current cantilever design. The structure is a Y shaped resonant device with two tines extending from a central beam cantilevered over the substrate, or, in the case of boron doped single crystal silicon a well. The thickness is on the order of $1\mu m$ or less and the excitation is in plane. Counterweights are positioned at the ends of the tines which lower the natural frequency and provide surface area for electrostatic forcing. The advantages are described as follows.

1. The pre-crack length and crack extension can be directly measured with a microscope and filar. *in situ* experiments with Auger or SEM may be possible because a plan view of the crack is possible.

2. A single nanoindentation can generate a crack of uniform depth through the entire thickness of the structure without introducing residual surface damage, or additional cracks, following etching. Time on the nanoindenter is also greatly reduced. The crack can be accurately characterized with handbook equations.

3. The thickness of the structure is small enough that alternative materials requiring a deposition process can be used for fabrication: polysilicon, nickel, tungsten, silica.

4. Finite element modeling of the structure, particularly the crack using singular elements, is greatly simplified through the use of 2D plane stress elements.

5. The total allowable deflection of the tines is determined by the photo mask dimensions that are controlled by the designer, not a process limitation.

6. The air gap spacing for the sensor and torquer can be measured directly with a microscope and filar, rather than inferred from optical depth of field and electrical measurements.

7. The thickness of the structure is small enough that precise thickness measurements can be made with the Nanospec.

8. DC electrostatic forces can be applied to the structure such that fatigue experiments that do not cause the crack faces to come into contact are possible.

Dynamically, the Y shaped structure is more of a challenge to model than the simple cantilever and probably will not have the same frequency sensitivity to crack growth. Modal analysis and optimization is again the key to good performance. Despite a likely reduction in frequency sensitivity, this design can be fully characterized with simple and highly accurate measurement techniques introducing very little uncertainty in interpreting frequency shifts as crack growth. Finally, because the mode of operation is in-plane, the baseline frequency of this structure will likely be high, $\approx$ 25 KHz; therefore, phase shift in the post amplifiers may cause excessive signal attenuation. Further refinements in the sensing electronics will probably be necessary. Long term thermal stability of the system could be improved by designing and building a stable VCO. Alternatively, self resonant AC control schemes that do not require a VCO and DC signals could be explored. Fatigue experiments that do not put the crack faces into contact should be possible. A high DC voltage, approximately 300 volts, could be applied to the electrodes summed with an AC tickle voltage that resonates the structure enough to pick off the zero crossings of the response. High speed measurements of the resonant frequency, approximately 100 Hz acquisition rate, are possible with the more elaborate HP frequency counters with the GPIB interface. This

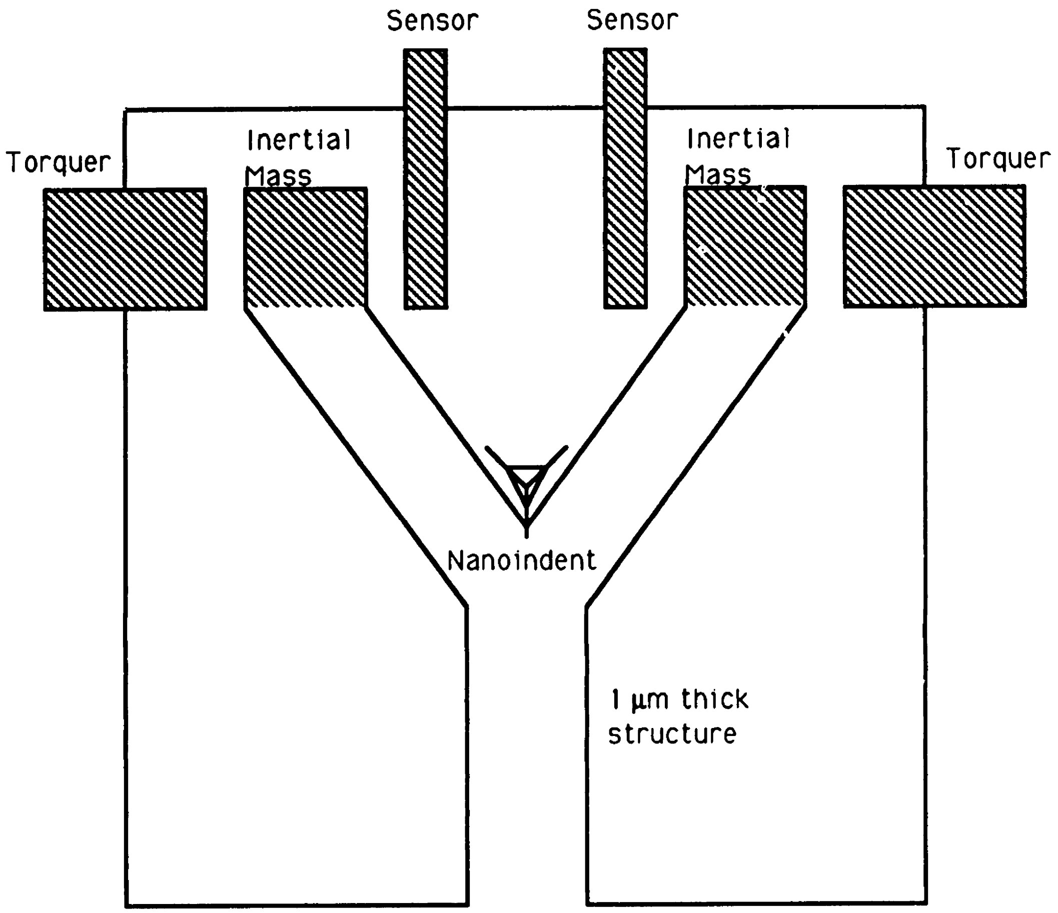


Figure 9.1: Generation 3 Design

technique utilizes the advantages of measuring frequency, eliminates the dynamic nonlinearity of the pre-crack, and allows the effects of crack closure to be examined with regard to the rate limiting process.

Appendix A

Frequency Calculation

In this section, frequencies are calculated for the 2 degree of freedom dynamic model and single degree of freedom models. The computations were performed with the aid of the design program, FTEST2.PAS, using the T38 structure as the baseline model. The design program is the primary modeling and analysis tool for this experiment. A copy of the code is included in Appendix G.7.

A.1 Two Degree of Freedom Model

$$\begin{aligned}
\omega_1 &= \sqrt{\frac{A-(A^2-4B)^{1/2}}{2}} \\
&= 2745 \text{ Hz} \\
\omega_2 &= \sqrt{\frac{A+(A^2-4B)^{1/2}}{2}} \\
&= 102153 \text{ Hz}
\end{aligned}$$

where,

$$\begin{aligned}
A &= \frac{k_{11}m_{22}+k_{22}m_{11}-2m_{12}k_{12}}{m_{11}m_{22}-m_{12}^2} \\
&= 4.123 \times 10^{11} \\
B &= \frac{k_{11}k_{22}-k_{12}^2}{m_{11}m_{22}-m_{12}^2}
\end{aligned}$$

$$= 1.226 \times 10^{20}$$

$$\begin{aligned} k_{11} &= 3.986 \\ k_{12} &= 5.978 \\ k_{21} &= 5.978 \\ k_{22} &= 11.96 \end{aligned}$$

$$\begin{aligned} m_{11} &= 1.290 \times 10^{-8} \\ m_{12} &= 1.717 \times 10^{-8} \\ m_{21} &= 1.717 \times 10^{-8} \\ m_{22} &= 2.286 \times 10^{-8} \end{aligned}$$

A.2 Single Degree of Freedom (Translational)

$$\begin{aligned} K_c &= \frac{3\,EI}{\ell_{ac}^3\left(1 + \frac{3\ell_{cg}}{2\ell_{ac}}\right)} \\ &= 1.233\frac{\mathrm{N}}{\mathrm{m}} \\ \omega_c &= \sqrt{\frac{K_c}{M}} \\ &= 4554 \text{ Hz} \end{aligned}$$

where,

$$\begin{aligned} E &= 1.59 \times 10^{11}\frac{\mathrm{N}}{\mathrm{m}^2} \\ I &= 1.377 \times 10^{-23}\ \mathrm{m}^4 \\ \ell_{ac} &= 1.3 \times 10^{-4}\ \mathrm{m} \\ \ell_{cgx} &= 1.235 \times 10^{-4}\ \mathrm{m} \\ M &= 1.506 \times 10^{-9}\ \mathrm{kg} \end{aligned}$$

A.3 Single Degree of Freedom (Rotational)

$$
\begin{aligned}
\bar{I}_A &= M(\ell_{cgx} + \ell_{ac})^2 + \bar{I} \\
&= 1.102 \times 10^{-16} \text{ kg m}^2 \\
K_\theta &= \frac{3\,EI}{\ell_{ac}} \left(\frac{1 + \frac{\ell_{cgx}}{\ell_{ac}}}{1 + \frac{3\ell_{cgx}}{2\ell_{ac}}} \right) \\
&= 4.062 \times 10^{-8} \\
\omega_\theta &= \sqrt{\frac{K_\theta}{\bar{I}_A}} \\
&= 3246 \text{ Hz}
\end{aligned}
$$

where,

$$\bar{I} = 9.111 \times 10^{-19} \text{ kg m}^2$$

Appendix B

Finite Element Modal Analysis

The following sections contain the ABAQUS input files for modal analysis of the generation 1 and 2 devices.

B.1 Generation 1 Analysis

```
*HEADING
ENDPLATE
*NODE
1,0,60,0
*NODE
13,0,-60,0
*NGEN,NSET=A
1,13,1
*NCOPY,CHANGE NUMBER=13,OLDSET=A,SHIFT,NEW SET=B
5.5,0,0
0,0,0,0,0,1.0,0
*NCOPY,CHANGE NUMBER=13,OLDSET=B,SHIFT,NEW SET=C
8,0,0
0,0,0,0,0,1.0,0
*NCOPY,CHANGE NUMBER=13,OLDSET=C,SHIFT,NEW SET=D
3.5,0,0
0,0,0,0,0,1.0,0
*NCOPY,CHANGE NUMBER=13,OLDSET=D,SHIFT,NEW SET=E
15,0,0
0,0,0,0,0,1.0,0
*NODE
79,47,-60,0
*NODE
78,47,-49,0
*NODE
77,47,-41,0
*NODE
76,47,-31,0
*NODE
75,47,-23,0
*NODE
74,47,-13,0
*NODE
73,47,-5,0
*NODE
72,47,5,0
*NODE
71,47,13,0
*NODE
70,47,23,0
*NODE
69,47,31,0
*NODE
68,47,41,0
*NODE
67,47,49,0
*NODE
66,47,60,0
*NSET,NSET=F,GENERATE
66,79,1
*NCOPY,CHANGE NUMBER=14,OLDSET=F,SHIFT,NEW SET=G
20,0,0
0,0,0,0,0,1.0,0
*NCOPY,CHANGE NUMBER=41,OLDSET=E,SHIFT,NEW SET=H
65,0,0
0,0,0,0,0,1.0,0
*NCOPY,CHANGE NUMBER=13,OLDSET=H,SHIFT,NEW SET=I
30,0,0
0,0,0,0,0,1.0,0
*NCOPY,CHANGE NUMBER=13,OLDSET=I,SHIFT,NEW SET=J
30,0,0
0,0,0,0,0,1.0,0
*ELEMENT,TYPE=S4R
1,2,15,14,1
*ELGEN,ELSET=COL1
1,1,13,12,12,1,1,1,1,1
```

```
*ELEMENT,TYPE=S4R
13,15,28,27,14
*ELEMENT,TYPE=S4R
14,16,29,28,15
*ELSET,ELSET=TOR1
13,14
*ELCOPY,ELEMENT SHIFT=2,OLD SET=TOR1,NEW SET=TOR2,SHIFT NODES=5
*ELCOPY,ELEMENT SHIFT=2,OLD SET=TOR2,NEW SET=TOR3,SHIFT NODES=5
*ELSET,ELSET=COL2
TOR1,TOR2,TOR3
*ELEMENT,TYPE=S4R
19,28,41,40,27
*ELGEN,ELSET=COL3_4
19,2,13,12,12,1,1,1,1,1
*ELEMENT,TYPE=S4R
43,54,67,66,53
*ELGEN,ELSET=TOPCOL5
43,1,13,13,6,1,1,1,1,1
*ELEMENT,TYPE=STRI3
49,59,73,72
*ELEMENT,TYPE=S4R
50,60,74,73,59
*ELGEN,ELSET=BOTCOL5
50,1,13,14,6,1,1,1,1,1
*ELSET,ELSET=COL5
TOPCOL5,49,BOTCOL5
*ELEMENT,TYPE=S4R
56,67,81,80,66
*ELGEN,ELSET=COL6
56,1,7,14,7,2,1,1,1,1,1
*ELEMENT,TYPE=S4R
63,81,95,94,80
*ELGEN,ELSET=TOPCOL7
63,1,13,13,6,1,1,1,1,1
*ELEMENT,TYPE=STRI3
69,87,100,86
*ELEMENT,TYPE=S4R
70,88,101,100,87
*ELGEN,ELSET=BOTCOL7
70,1,13,13,6,1,1,1,1,1
*ELSET,ELSET=COL7
TOPCOL7,69,BOTCOL7
*ELEMENT,TYPE=S4R
76,95,108,107,94
*ELGEN,ELSET=COL8_9
76,2,13,12,12,1,1,1,1,1
*ELSET,ELSET=SIPL
COL1,COL2,COL3_4,COL5,COL6,COL7,COL8_9
*MATERIAL,NAME=SILICON
*DENSITY
2.3E-15
*ELASTIC,TYPE=ISO
0.159,0.33
*SHELL SECTION,ELSET=SIPL,MATERIAL=SILICON
2.9
*BOUNDARY
1000,1,6
*NSET,NSET=HG,GENERATE
96,104
*NSET,NEST=IG,GENERATE
109,117
*NSET,NSET=JG,GENERATE
122,130
*NSET,NSET=BOTGOLD
HG,IG,JG
*NCOPY,CHANGE NUMBER=100,OLDSET=BOTGOLD,NEW SET=TOPGOLD,SHIFT
```

```
0,0,15
0,0,0,0,0,1,0
*ELEMENT,TYPE=C3D8
178,97,110,109,96,197,210,209,196
*ELGEN,ELSET=GOLDM
178,2,13,12,8,1,1,1,1,1
*MATERIAL,NAME=GOLD
*DENSITY
1.93E-14
*ELASTIC,TYPE=ISO
8E-2,0.3
*SOLID SECTION,ELSET=GOLDM,MATERIAL=GOLD
*NODE
1000,-130,0,0
*NODE
1012,-10,0,0
*NGEN,NSET=BEAMNODE
1000,1012,1
*ELEMENT,TYPE=B31
200,1000,1001
*ELGEN,ELSET=BEAMEL
200,12,1,1,1,1,1,1,1,1
*ELEMENT,TYPE=B31
213,1012,7
*BEAM GENERAL SECTION,SECTION=GENERAL
26.19,0,0,13.77,0,216.5,51.78
0,1,0
0.159,0.051,1.2E-6,0.0
*MPC
12,7,1012,20
*PLOT
*VIEWPOINT
1,1,1,0,0,1
*DRAW,HIDE
*NSET,NSET=MODEOUT
1000,1005,1010,7,20,33,46,59,100,113,126
*STEP,LINEAR=NEW
*FREQUENCY
3,100000
*FILE FORMAT,ASCII
*NODE PRINT,GLOBAL=YES,LASTMODE=3,NSET=MODEOUT
U
*PLOT
*VIEWPOINT
1,1,1,0,0,1
*DISPLACED,HIDE
U
*PLOT
*VIEWPOINT
0,1,0,0,0,1
*DISPLACED,HIDE
U
*VIEWPOINT
0,0,1,0,0,1
*DISPLACED
U
*END STEP
```

```
*HEADING
MICROCANTILEVER
*NODE
1,-4.5,0,0
*NODE
2,-4,0,0
*NODE
14,-4.83,-0.17,0
*NODE
27,-5,-0.5,0
*NODE
12,4,0,0
*NODE
13,4.5,0,0
*NODE
26,4.83,-0.17,0
*NODE
39,5,-0.5,0
*NODE
38,5,-1,0
*NODE
28,-5,-1,0
*NODE
33,0,-3,0
*NGEN,NSET=TOPA
2,12,1
*NGEN,LINE=P,NSET=BOTA
28,38,1,33
*NFILL,NSET=BACKA
TOPA,BOTA,2,13
*NSET,NSET=BACK
BACKA,1,14,27,13,26,39
*NCOPY,CHANGE NUMBER=975,OLDSET=BACK,SHIFT,NEW SET=FRONT
0,0,25.0
0,0,0,0,0,25.0,0
*NFILL
BACK,FRONT,25,39
*ELEMENT,TYPE=C3D8
1,14,15,2,1,53,54,41,40
*ELGEN,ELSET=SIBEAM
1,12,1,1,2,13,12,25,39,24
*MATERIAL,NAME=SILICON
*ELASTIC,TYPE=ISO
0.159,0.33
*SOLID SECTION,ELSET=SIBEAM,MATERIAL=SILICON
*BOUNDARY
BACK,1,3
*PLOT
*VIEWPOINT
1,1,1,0,1,0
*DRAW,HIDE
*VIEWPOINT
0,0,1,0,1,0
*DRAW,HIDE
*STEP,LINEAR
*STATIC
*CLOAD
982,2,-5E-4
*PLOT,COLORS=11
*VIEWPOINT
1,0,0,0,1,0
*DISPLACED,HIDE
U,1
*END STEP
```

```
*HEADING
ENDPLATE
*NODE
1,0,60,0
*NODE
13,0,-60,0
*NGEN,NSET=A
1,13,1
*NCOPY,CHANGE NUMBER=13,OLDSET=A,SHIFT,NEW SET=B
5.5,0,0
0,0,0,0,0,1.0,0
*NCOPY,CHANGE NUMBER=13,OLDSET=B,SHIFT,NEW SET=C
8,0,0
0,0,0,0,0,1.0,0
*NCOPY,CHANGE NUMBER=13,OLDSET=C,SHIFT,NEW SET=D
3.5,0,0
0,0,0,0,0,1.0,0
*NCOPY,CHANGE NUMBER=13,OLDSET=D,SHIFT,NEW SET=E
15,0,0
0,0,0,0,0,1.0,0
*NODE
79,47,-60,0
*NODE
78,47,-49,0
*NODE
77,47,-41,0
*NODE
76,47,-31,0
*NODE
75,47,-23,0
*NODE
74,47,-13,0
*NODE
73,47,-5,0
*NODE
72,47,5,0
*NODE
71,47,13,0
*NODE
70,47,23,0
*NODE
69,47,31,0
*NODE
68,47,41,0
*NODE
67,47,49,0
*NODE
66,47,60,0
*NSET,NSET=F,GENERATE
66,79,1
*NCOPY,CHANGE NUMBER=14,OLDSET=F,SHIFT,NEW SET=G
20,0,0
0,0,0,0,0,1.0,0
*NCOPY,CHANGE NUMBER=41,OLDSET=E,SHIFT,NEW SET=H
65,0,0
0,0,0,0,0,1.0,0
*NCOPY,CHANGE NUMBER=13,OLDSET=H,SHIFT,NEW SET=I
30,0,0
0,0,0,0,0,1.0,0
*NCOPY,CHANGE NUMBER=13,OLDSET=I,SHIFT,NEW SET=J
30,0,0
0,0,0,0,0,1.0,0
*ELEMENT,TYPE=S4R
1,2,15,14,1
*ELGEN,ELSET=COL1
1,1,13,12,12,1,1,1,1,1
```

```
*ELEMENT,TYPE=S4R
13,15,28,27,14
*ELEMENT,TYPE=S4R
14,16,29,28,15
*ELSET,ELSET=TOR1
13,14
*ELCOPY,ELEMENT SHIFT=2,OLD SET=TOR1,NEW SET=TOR2,SHIFT NODES=5
*ELCOPY,ELEMENT SHIFT=2,OLD SET=TOR2,NEW SET=TOR3,SHIFT NODES=5
*ELSET,ELSET=COL2
TOR1,TOR2,TOR3
*ELEMENT,TYPE=S4R
19,28,41,40,27
*ELGEN,ELSET=COL3_4
19,2,13,12,12,1,1,1,1,1
*ELEMENT,TYPE=S4R
43,54,67,66,53
*ELGEN,ELSET=TOPCOL5
43,1,13,13,6,1,1,1,1,1
*ELEMENT,TYPE=STRI3
49,59,73,72
*ELEMENT,TYPE=S4R
50,60,74,73,59
*ELGEN,ELSET=BOTCOL5
50,1,13,14,6,1,1,1,1,1
*ELSET,ELSET=COL5
TOPCOL5,49,BOTCOL5
*ELEMENT,TYPE=S4R
56,67,81,80,66
*ELGEN,ELSET=COL6
56,1,7,14,7,2,1,1,1,1,1
*ELEMENT,TYPE=S4R
63,81,95,94,80
*ELGEN,ELSET=TOPCOL7
63,1,13,13,6,1,1,1,1,1
*ELEMENT,TYPE=STRI3
69,87,100,86
*ELEMENT,TYPE=S4R
70,88,101,100,87
*ELGEN,ELSET=BOTCOL7
70,1,13,13,6,1,1,1,1,1
*ELSET,ELSET=COL7
TOPCOL7,69,BOTCOL7
*ELEMENT,TYPE=S4R
76,95,108,107,94
*ELGEN,ELSET=COL8_9
76,2,13,12,12,1,1,1,1,1
*ELSET,ELSET=SIPL
COL1,COL2,COL3_4,COL5,COL6,COL7,COL8_9
*MATERIAL,NAME=SILICON
*DENSITY
2.3E-15
*ELASTIC,TYPE=ISO
0.159,0.33
*SHELL SECTION,ELSET=SIPL,MATERIAL=SILICON
2.9
*BOUNDARY
A,1,6
*NSET,NSET=HG,GENERATE
96,104
*NSET,NEST=IG,GENERATE
109,117
*NSET,NSET=JG,GENERATE
122,130
*NSET,NSET=BOTGOLD
HG,IG,JG
*NCOPY,CHANGE NUMBER=100,OLDSET=BOTGOLD,NEW SET=TOPGOLD,SHIFT
```

```
,0,15
,0,0,0,0,1,0
ELEMENT,TYPE=C3D8
78,97,110,109,96,197,210,209,196
ELGEN,ELSET=GOLDM
78,2,13,12,8,1,1,1,1,1
MATERIAL,NAME=GOLD
DENSITY
.93E-14
ELASTIC,TYPE=ISO
E-2,0.3
SOLID SECTION,ELSET=GOLDM,MATERIAL=GOLD
PLOT
VIEWPOINT
,1,1,0,0,1
DRAW,HIDE
NSET,NSET=MODEOUT
,20,33,46,59,100,113,126
STEP,LINEAR=NEW
FREQUENCY
,100000
FILE FORMAT,ASCII
NODE PRINT,GLOBAL=YES,LASTMODE=3,NSET=MODEOUT

PLOT
VIEWPOINT
,1,1,0,0,1
DISPLACED,HIDE

PLOT
VIEWPOINT
,1,0,0,0,1
DISPLACED,HIDE

END STEP
```

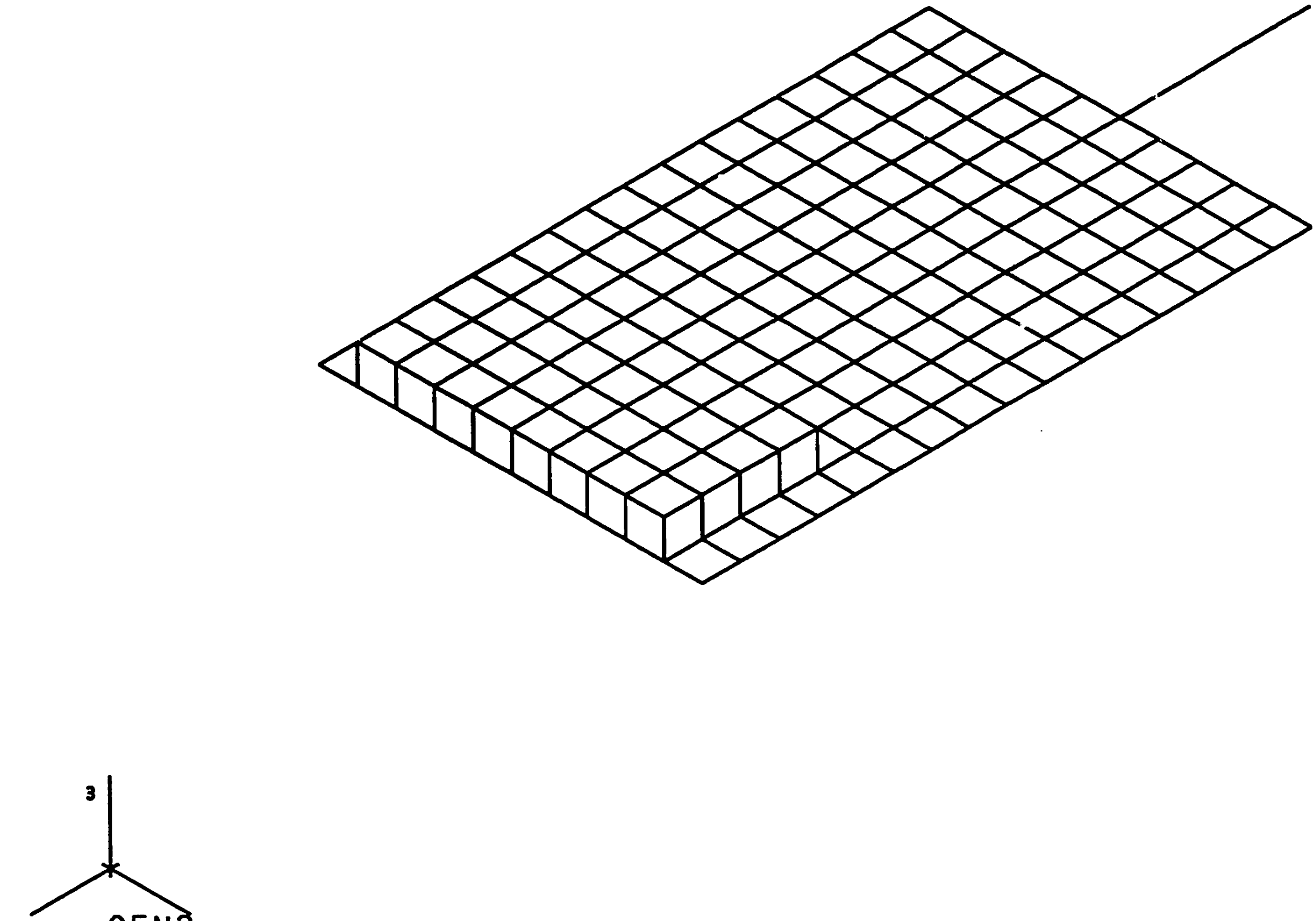
B.2 Generation 2 Analysis

FEA OF NEW DEVICE

MODAL FREQUENCY COMPARISON

MODE	FEA	FTEST2	% DIFF.
1	13.073	13.438	2.7
2	31.569	42.955	26.5
3	74.468	85.736	13.1

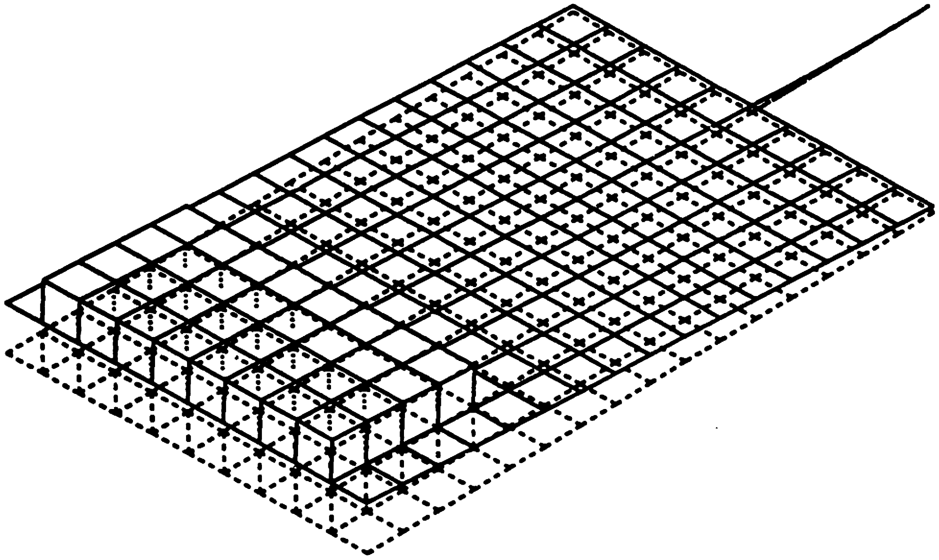
1. In plane bending.
2. Out of plane bending (unmodeled in Lagrangian)
3. Torsion (unmodeled in Lagrangian)

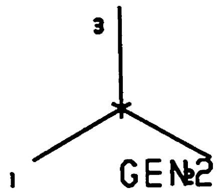


GEN2

ABAQUS VERSION 4-7-25 DATE: 26-MAR-92 TIME: 13:19:43

U
MAG. FACTOR = +1.8E+01
SOLID LINES - DISPLACED MESH
DASHED LINES - ORIGINAL MESH

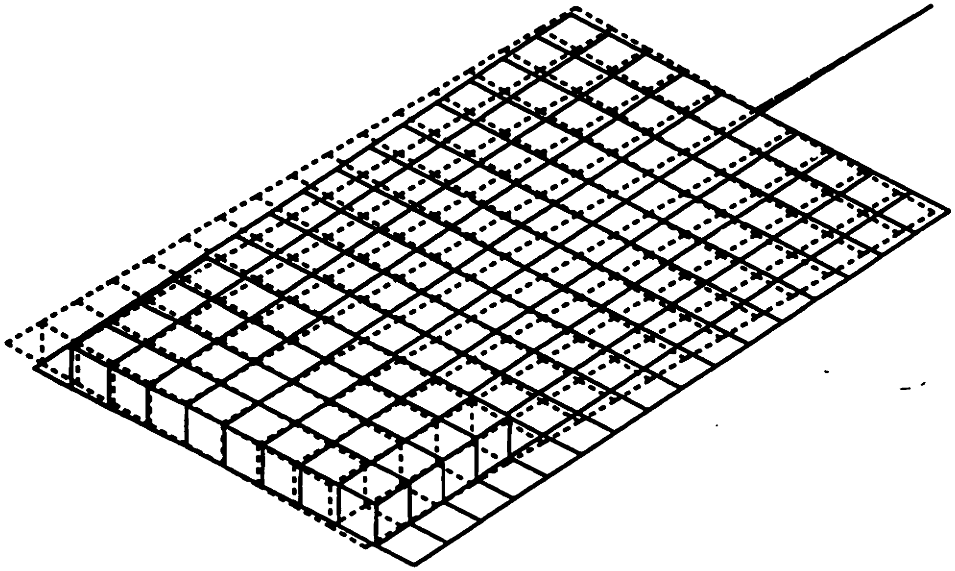


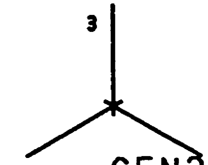


GEN№2

EIGENVALUE = +6.747E+03
ABAQUS VERSION 4-7-25 DATE: 26-MAR-92 TIME: 16:17:56

U
MAG. FACTOR = +1.6E+01
SOLID LINES - DISPLACED MESH
DASHED LINES - ORIGINAL MESH



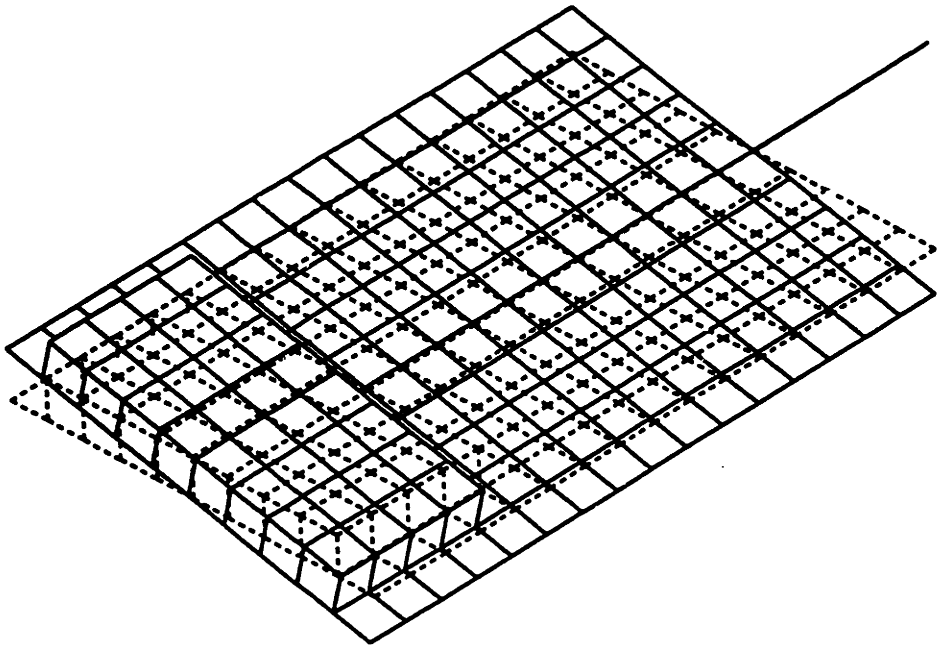


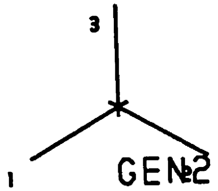
GEN2

EIGENVALUE = +3.934E+04

ABAQUS VERSION 4-7-25 DATE: 26-MAR-92 TIME: 16:17:56

U
MAG. FACTOR = +1.7E+01
SOLID LINES - DISPLACED MESH
DASHED LINES - ORIGINAL MESH





GEN№2

EIGENVALUE = +2.189E+05

ABAOUS VERSION U-7-25 DATE: 26-MAR-92 TIME: 18:17:56

```
*HEADING
GEN2
*NODE
1,75,75,0
*NODE
11,75,-75,0
*NODE
177,315,75,0
*NODE
187,315,-75,0
*NGEN,NSET=A1
1,11,1
*NGEN,NSET=B1
177,187,1
*NFILL,NSET=C1
A1,B1,16,11
*NCOPY,CHANGE NUMBER=200,OLD SET=C1,SHIFT,NEW SET=D1
0,0,15
75,75,0,75,75,15,0
*ELEMENT,TYPE=S4R
1,1,2,13,12
*ELGEN,ELSET=PLATE
1,10,1,1,16,11,10,1,1,1
*ELEMENT,TYPE=C3D8
161,134,135,146,145,334,335,346,345
*ELGEN,ELSET=CW
161,8,1,1,4,11,8,1,1,1
*NODE
401,0,0,0
*NODE
411,75,0,0
*NGEN,NSET=E
401,411,1
*ELEMENT,TYPE=B32
200,401,402,403
*ELGEN,ELSET=CANT1
200,4,2,1,1,1,1,1,1,1
*ELEMENT,TYPE=B32
204,409,410,6
*ELSET,ELSET=CANT
CANT1,204
*BOUNDARY
401,1,6
*MPC
1,6,410,17
*BEAM GENERAL SECTION,SECTION=GENERAL,DENSITY=2.3E-15,ELSET=CANT
502.7,0,0,3141.6,0,502.7,1733.3
0,0,1
0.159,0.051,1.2E-6,0
*SHELL SECTION,ELSET=PLATE,MATERIAL=SI
8
*SOLID SECTION,ELSET=CW,MATERIAL=AU
*MATERIAL,NAME=SI
*DENSITY
2.3E-15
*ELASTIC,TYPE=ISO
0.159,0.33
*MATERIAL,NAME=AU
*DENSITY
1.93E-14
*ELASTIC,TYPE=ISO
8E-2,0.3
*PLOT
*VIEWPOINT
0,0,1,0,1,0
*DRAW
```

```
*VIEWPOINT
1,1,1,0,0,1
*DRAW,HIDE
*STEP,LINEAR=NEW
*FREQUENCY
3,100000
*PLOT
*VIEWPOINT
1,1,1,0,0,1
*DISPLACED,HIDE
U
*END STEP
```

Appendix C

Nanoindenter Hardness/Modulus Data

File name: T38_H

Pt. \#	Depth Total (nm.)	Depth Plastic (nm.)	Load (mN)	Area (nm^2)	E (kg/mm^2)	H (kg/mm^2)
31	438.49	287.78	40.41	2360853.53	16222.262	1711.576
32	437.61	286.32	40.82	2338594.28	16429.895	1745.473
33	439.95	293.64	40.16	2451122.56	16310.898	1638.452
34	440.48	290.02	40.13	2395186.54	15990.131	1675.344
35	438.77	290.34	40.07	2400088.80	16197.713	1669.637
36	439.98	294.15	40.11	2459071.70	16319.424	1631.134
37	438.42	291.58	39.70	2419222.74	16148.067	1640.887
38	439.93	295.02	40.04	2472560.40	16354.547	1619.491
39	440.01	296.35	39.33	2493348.62	16103.454	1577.596
	439 <AVE.					

```
AVERAGES OF (E) & (H) =        16230.710      1656.621
   Standard deviation =          137.556        50.356
   Error =                          .85%         3.04%
```

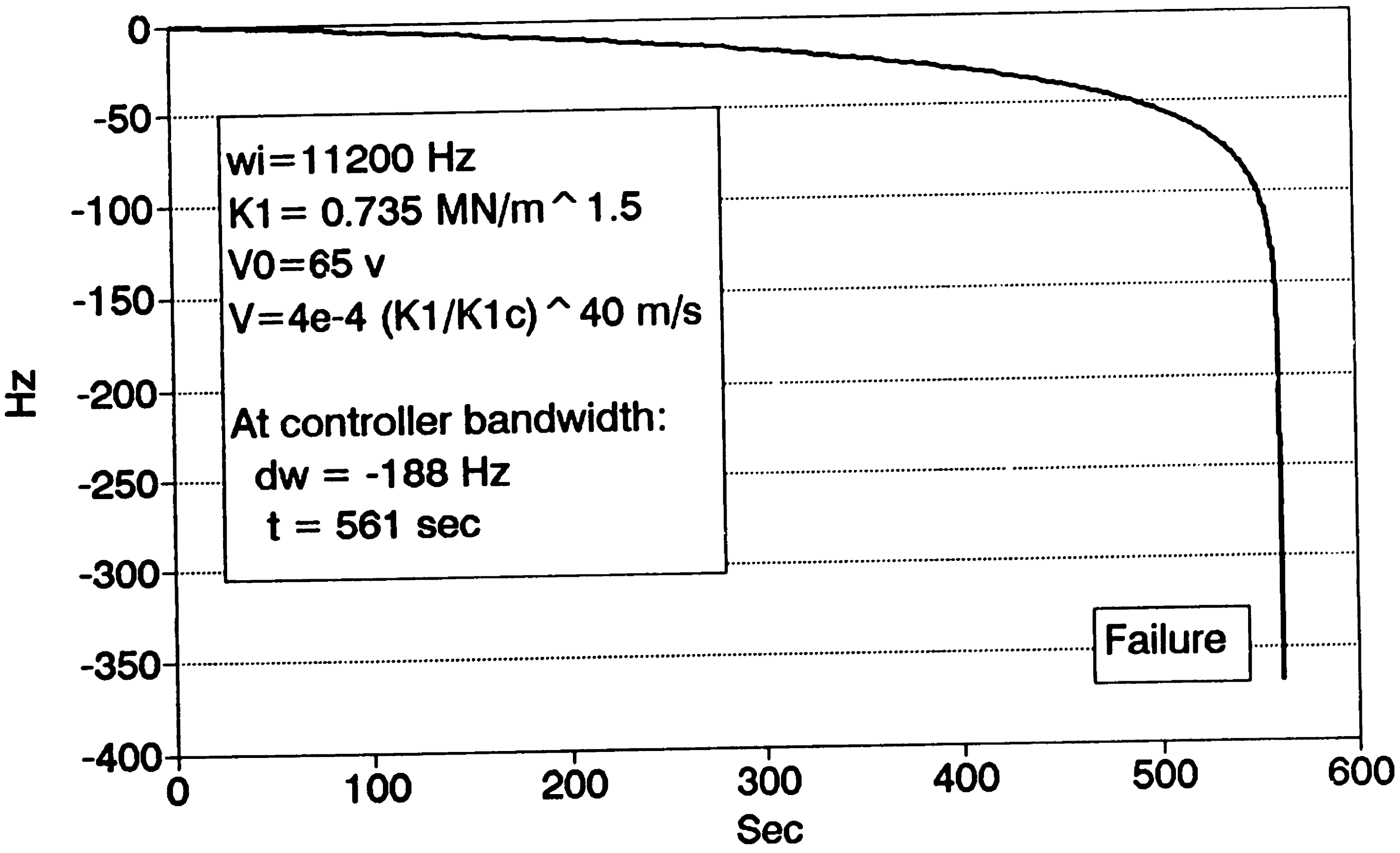
```
Diamond # 3    A(F)= 18.5 * D^2 + 205.3 * D^3/2 +(-603 * D) + 0 * D^0.5 + 0
                                 File name: T38_H

          Depth      Depth
Pt.       Total      Plastic           Load        Area              E              H
 #        (nm.)      (nm.)             (mN)       (nm^2)        (kg/mm^2)     (kg/mm^2)

40       438.43      290.55           39.69     2403347.46      16076.673      1651.550
41       439.13      295.04           39.68     2472895.25      16289.002      1604.689
42       440.34      295.25           39.63     2476114.68      16121.433      1600.326
43       439.43      293.00           39.53     2441237.87      16040.580      1619.417
44       441.03      297.70           39.69     2514612.59      16233.918      1578.248
45       439.73      295.13           39.64     2474268.13      16198.765      1602.158
46       441.58      299.21           39.51     2538422.02      16191.266      1556.674
47       440.79      303.11           39.38     2600249.59      16531.802      1514.534
48       441.77      302.47           38.97     2590144.47      16151.778      1504.647
49       440.02      307.40           38.94     2669305.19      16785.332      1458.874
         440 <AVE.

                               AVERAGES OF (E) & (H) =          16262.055      1569.112
                                  Standard deviation =            229.121        59.799
                                  Error =                           1.41%         3.81%
```

Appendix D

102491 Frequency Data

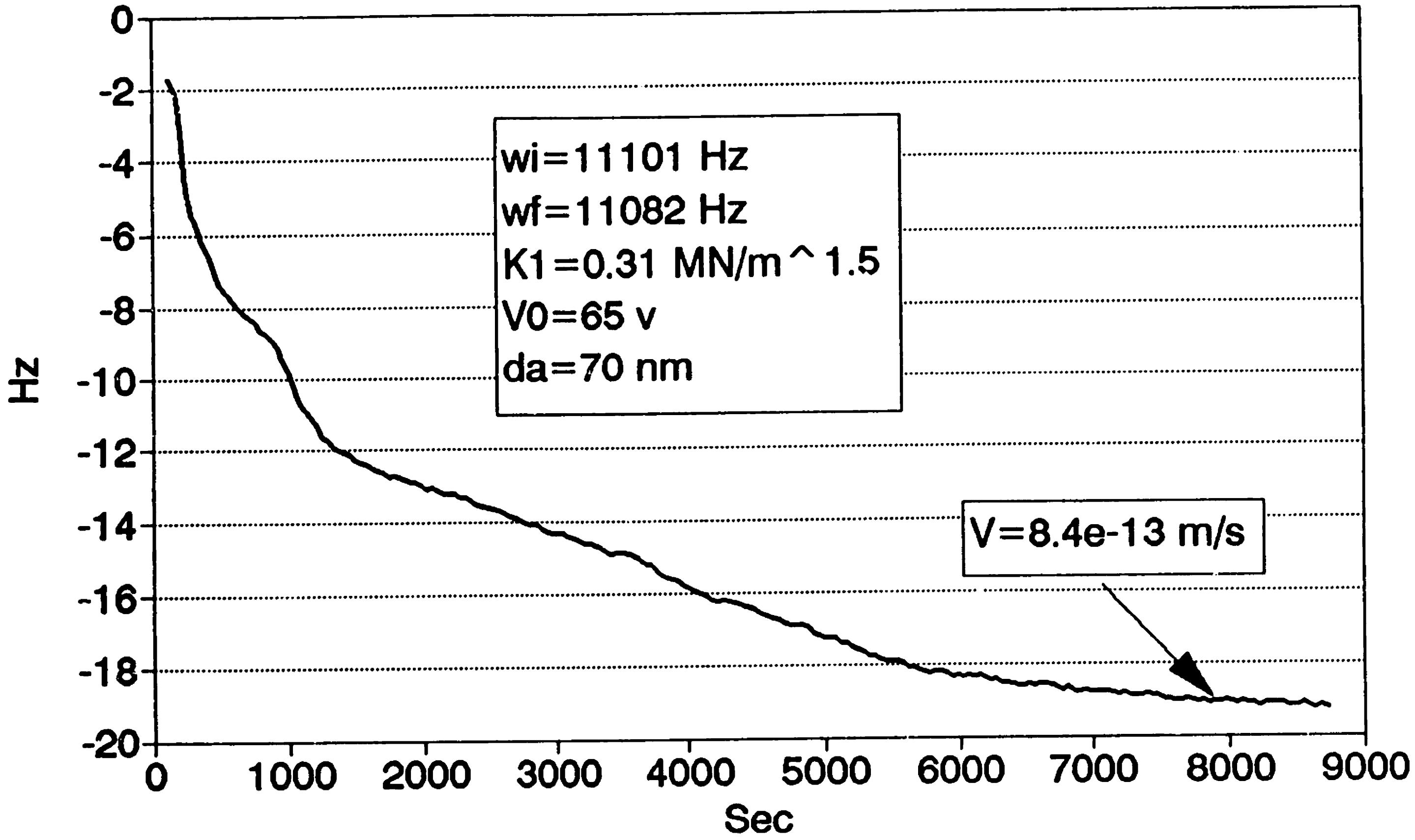
102491 run 1

Theory



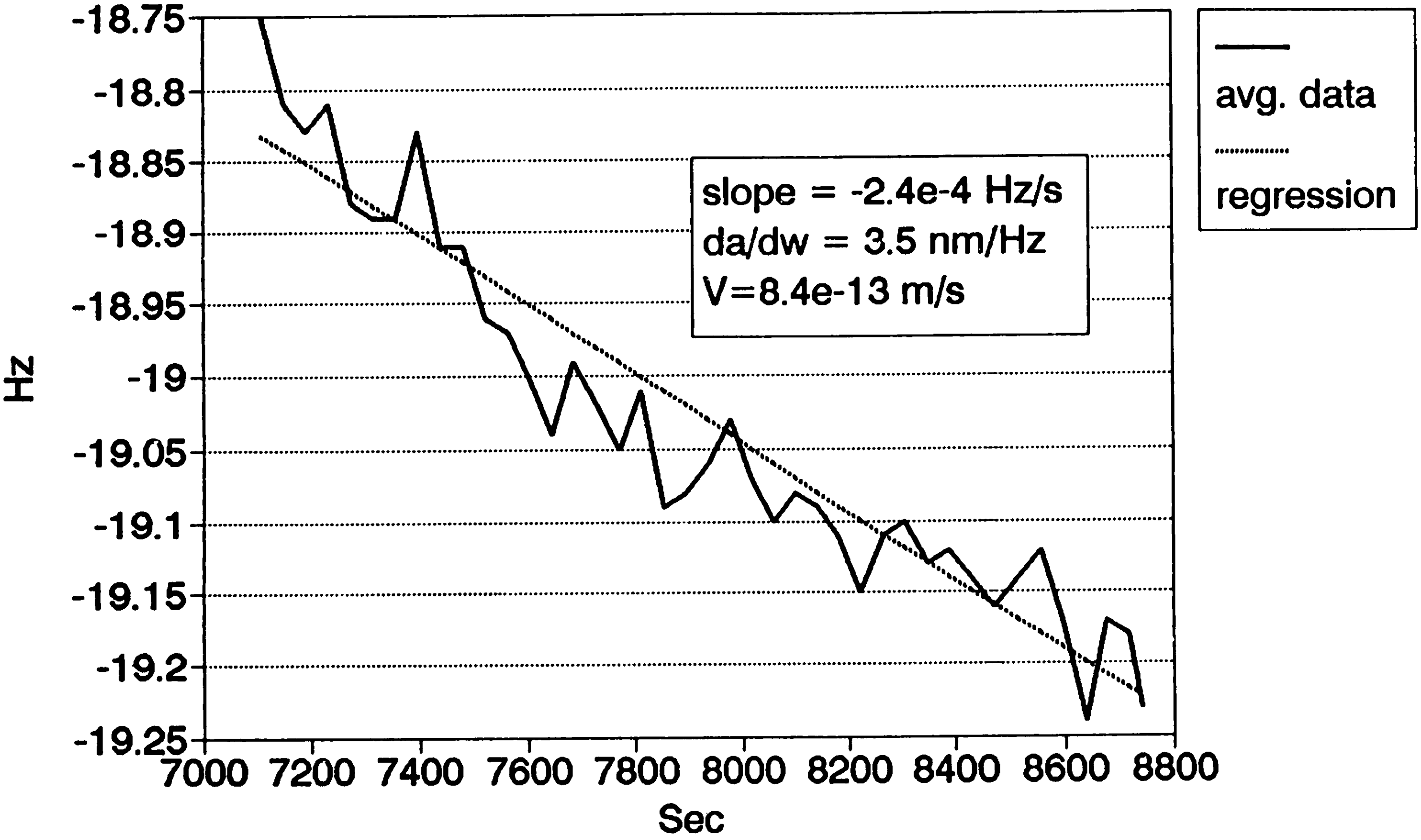
102491 run 1

average data



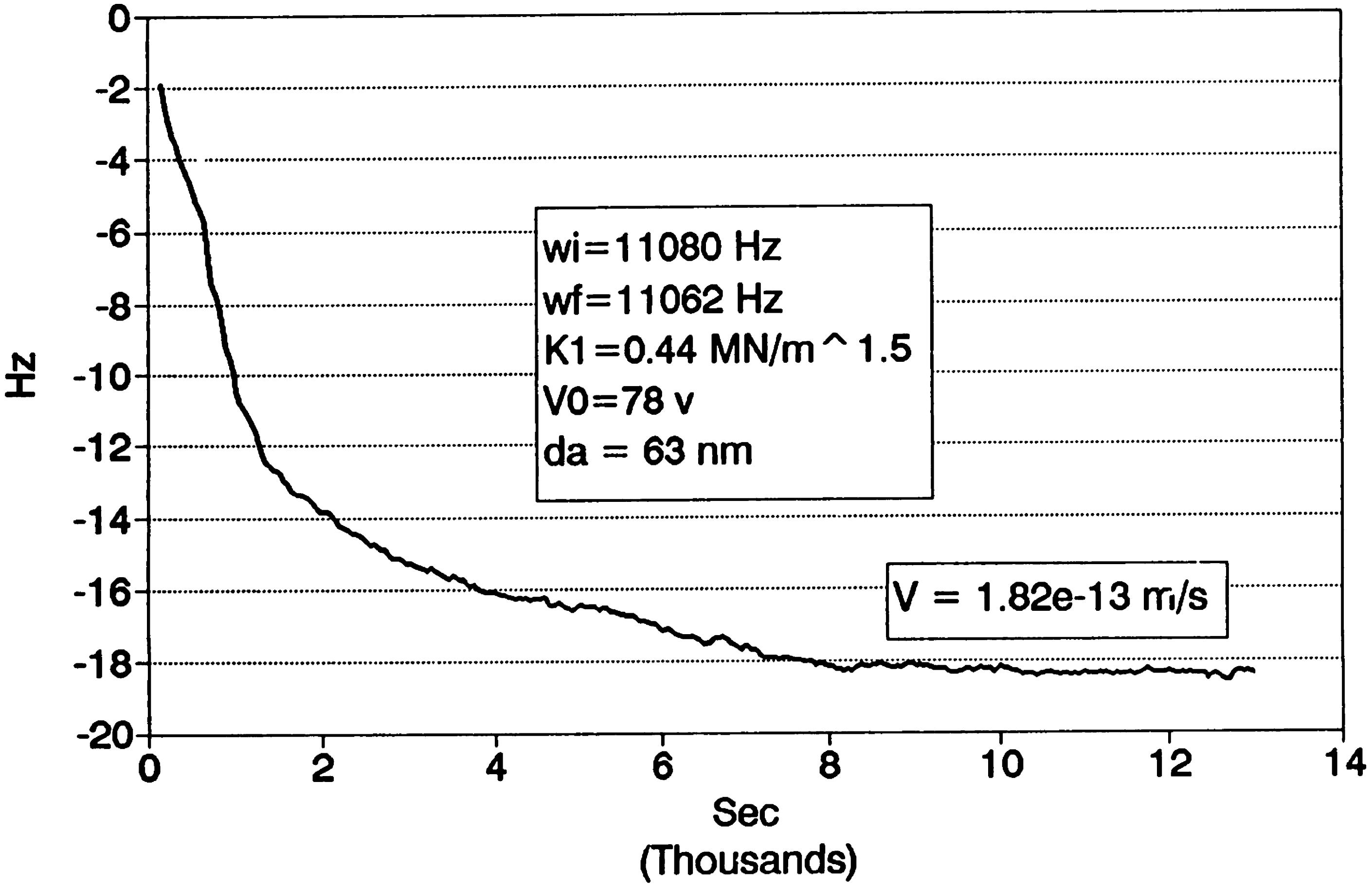
102491 run 1

Last 2000 seconds



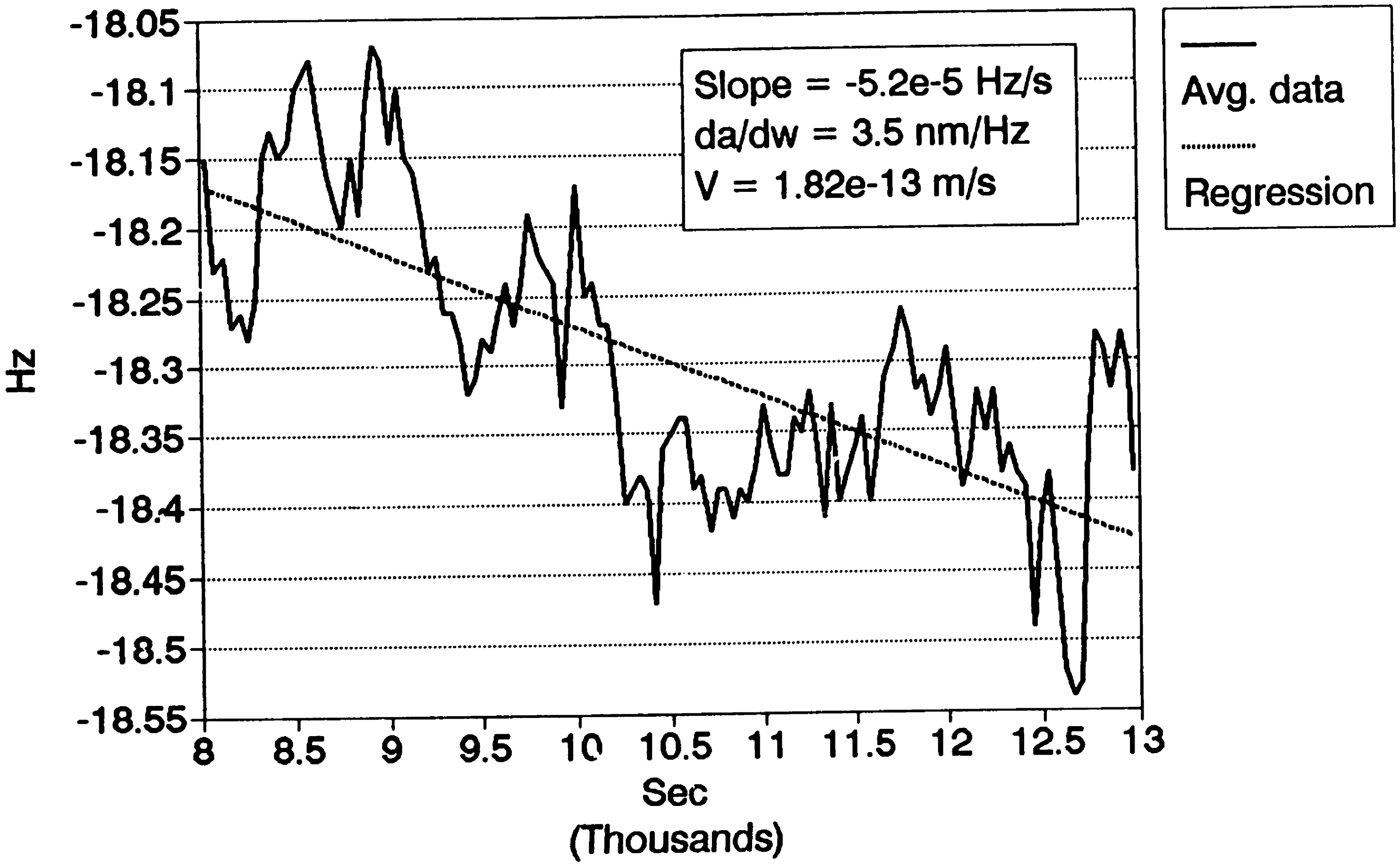
102491 run 2

averaged data



102491 Run 2

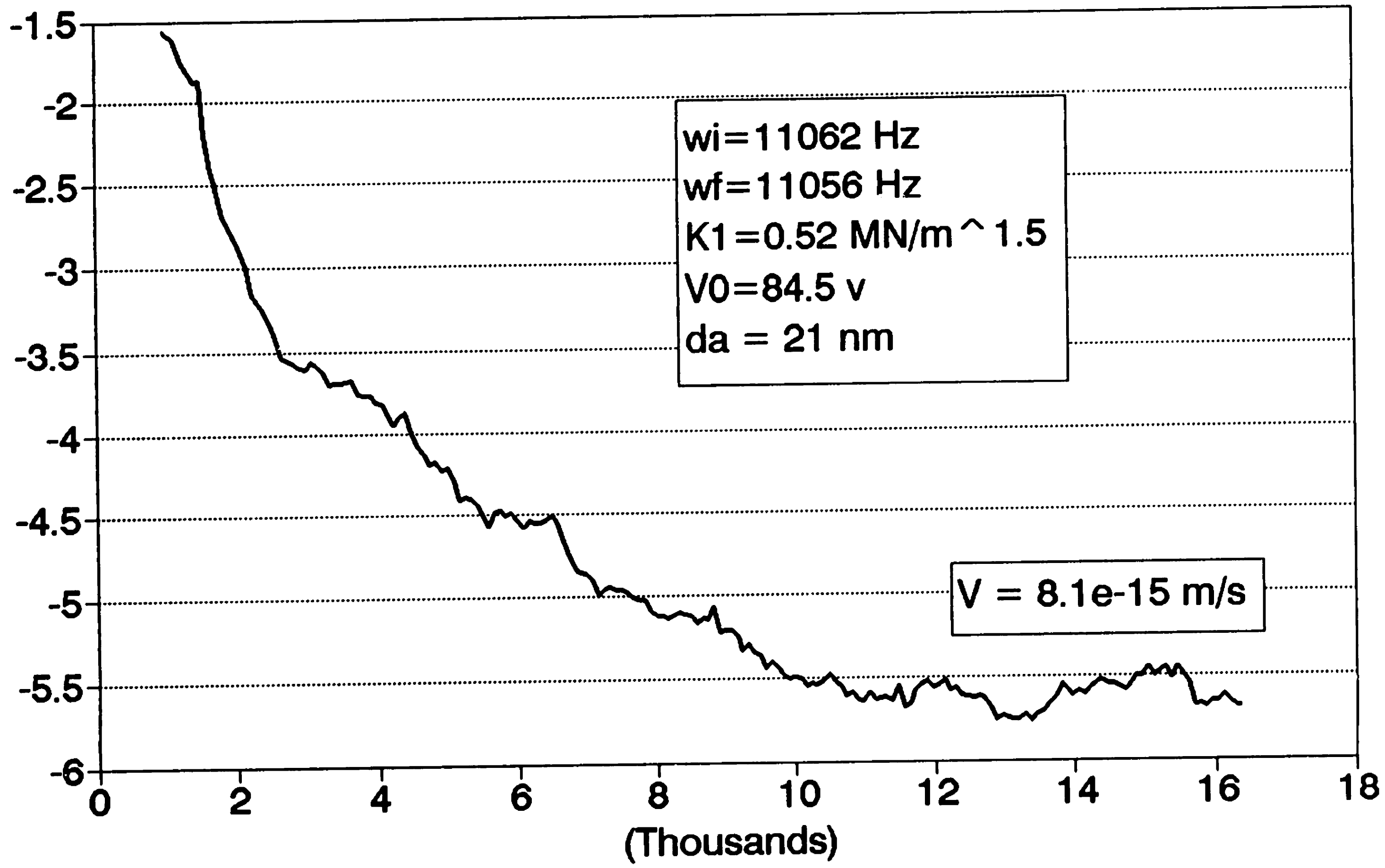
Last 5000 seconds



Slope = -5.2e-5 Hz/s
da/dw = 3.5 nm/Hz
V = 1.82e-13 m/s

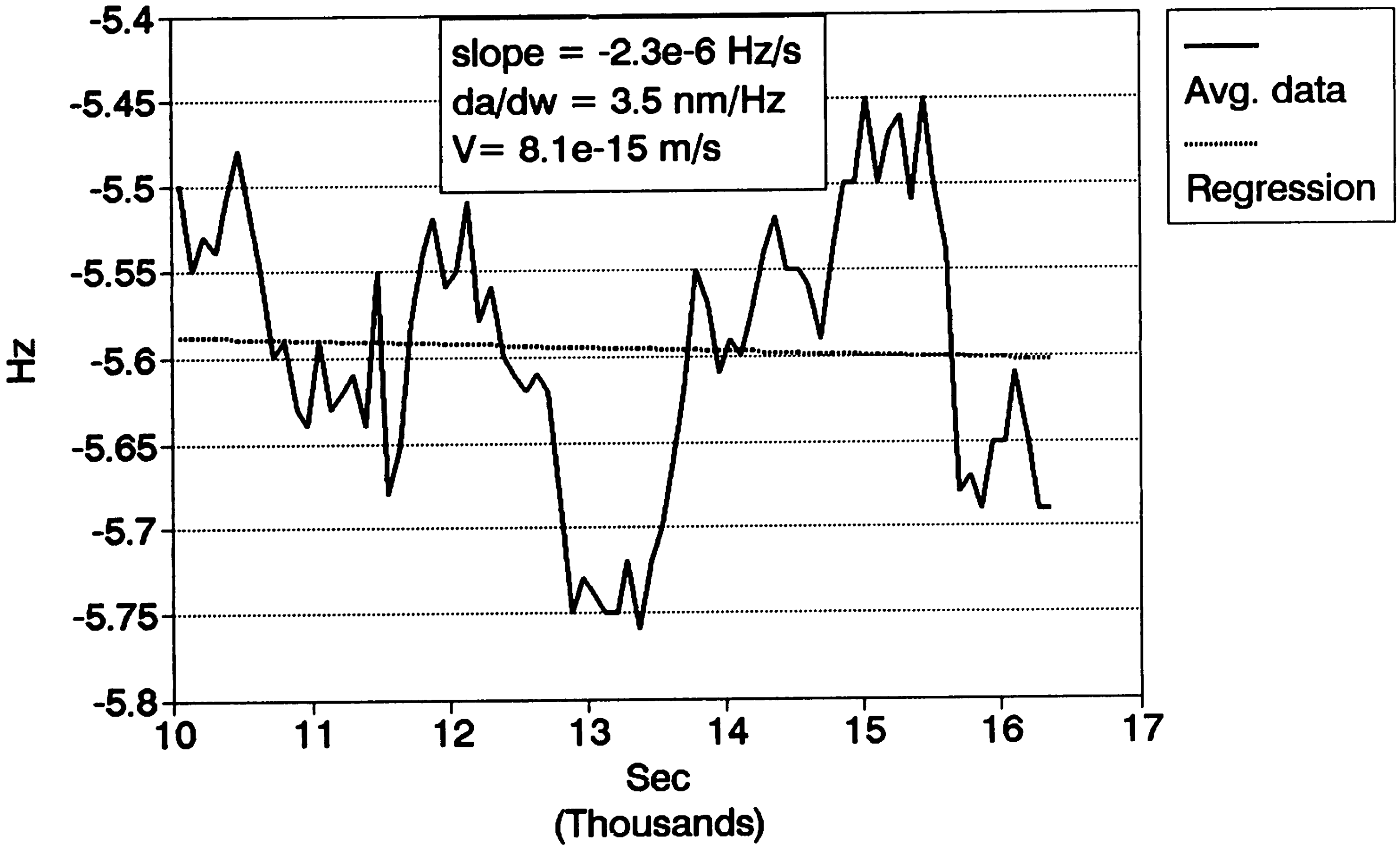
102491 run 3

averaged data



102491 Run 3

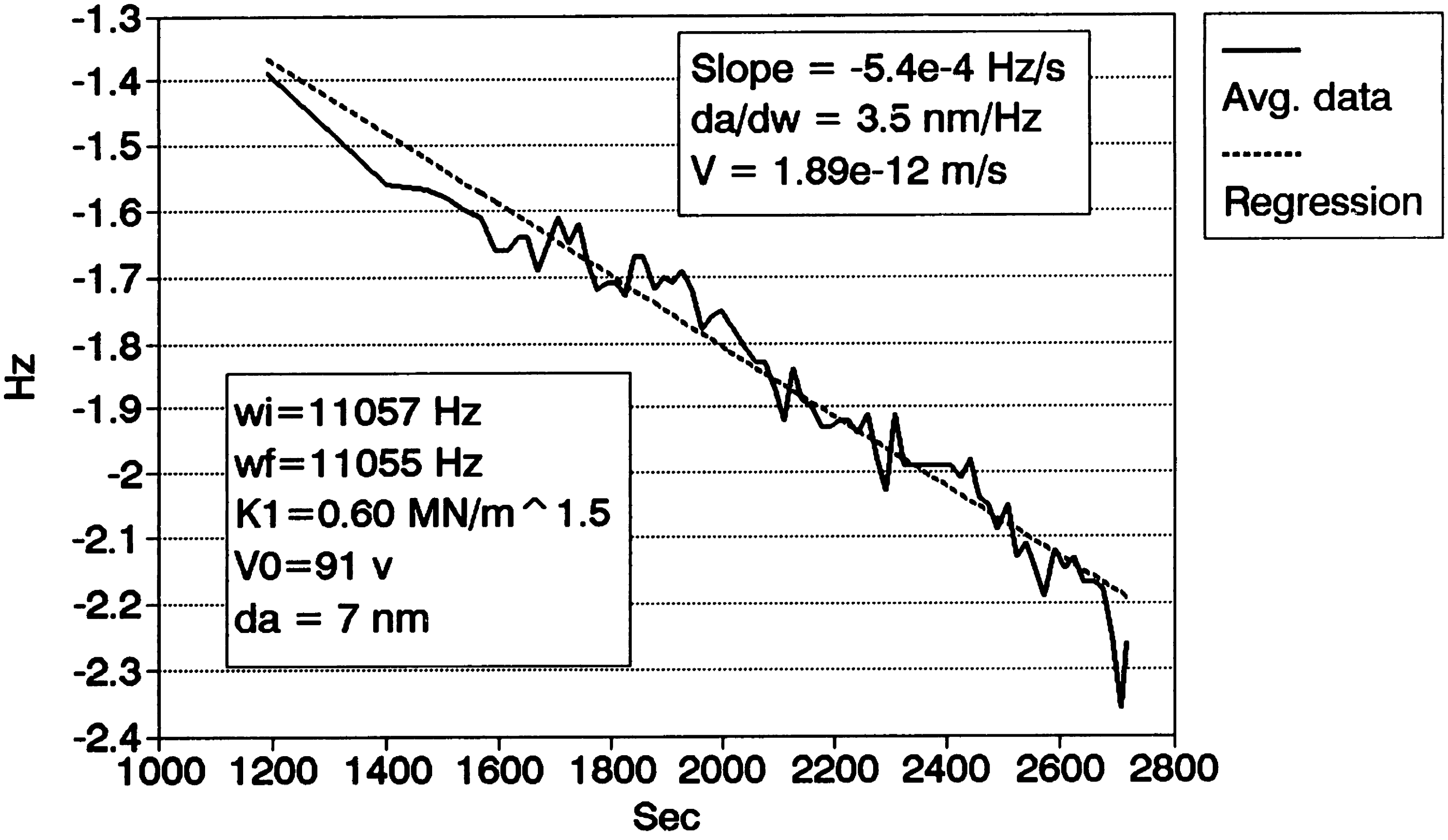
Last 6000 seconds



slope = -2.3e-6 Hz/s
da/dw = 3.5 nm/Hz
V= 8.1e-15 m/s

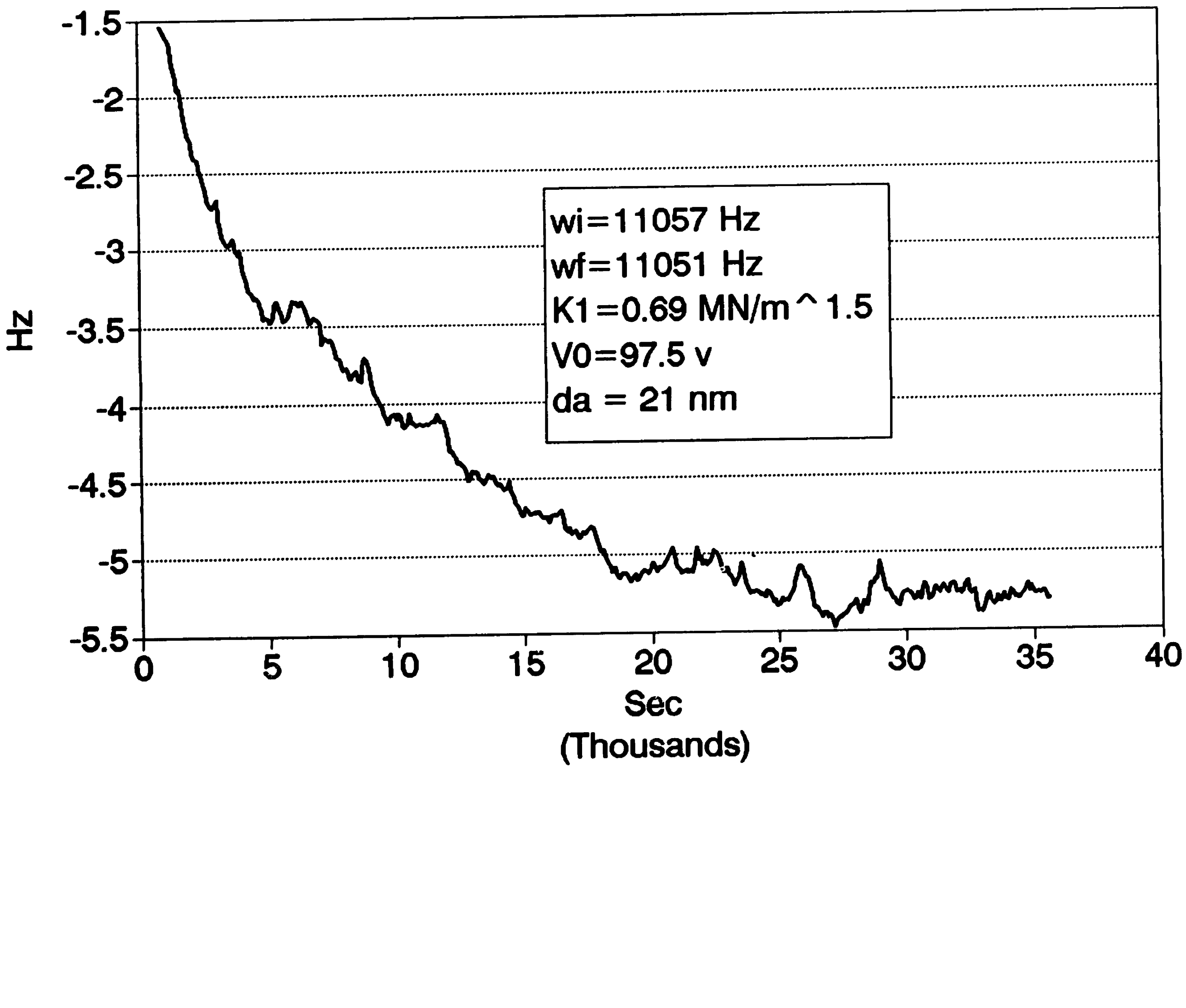
102491 run 4

averaged data



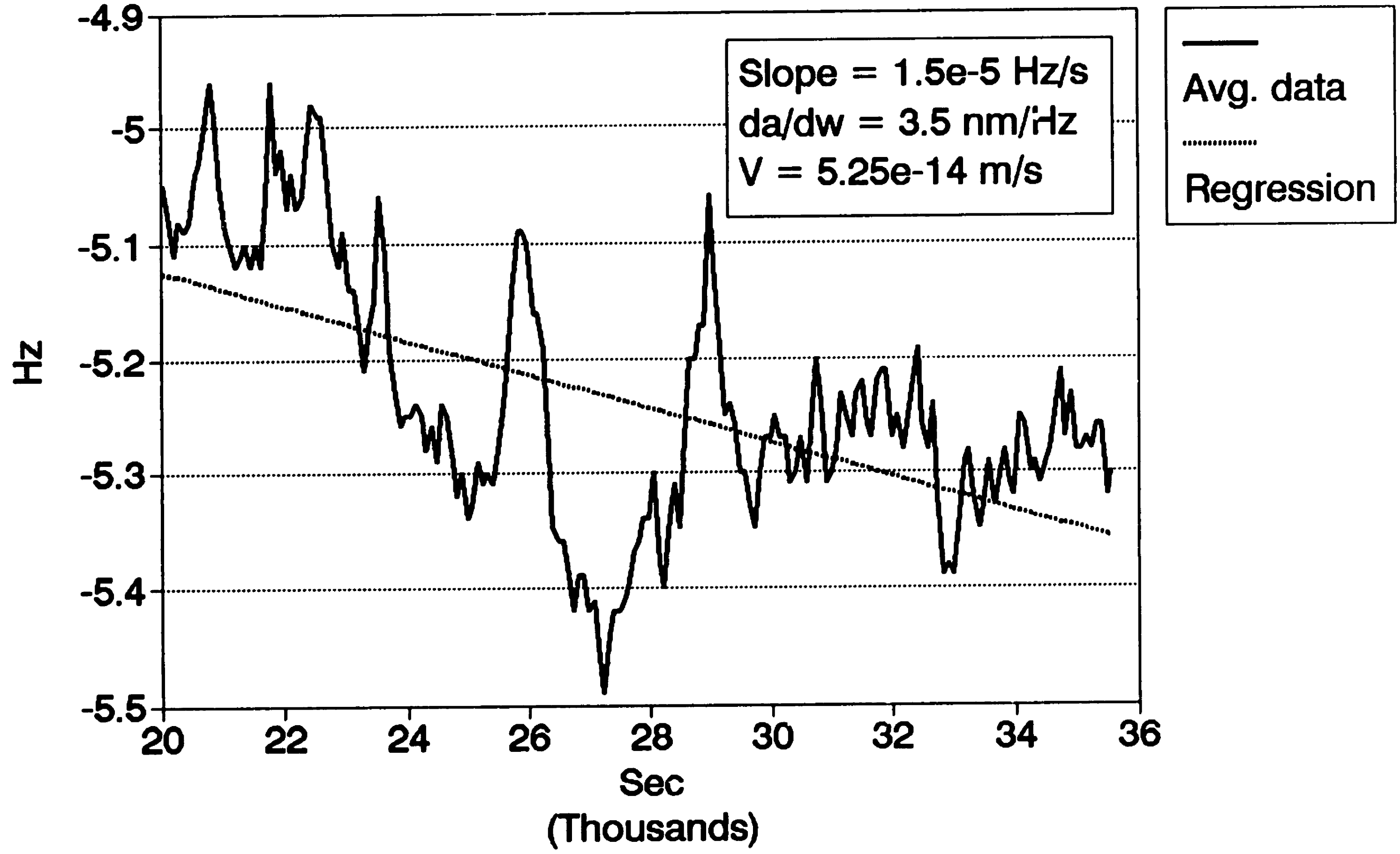
102491 run 5

averaged data



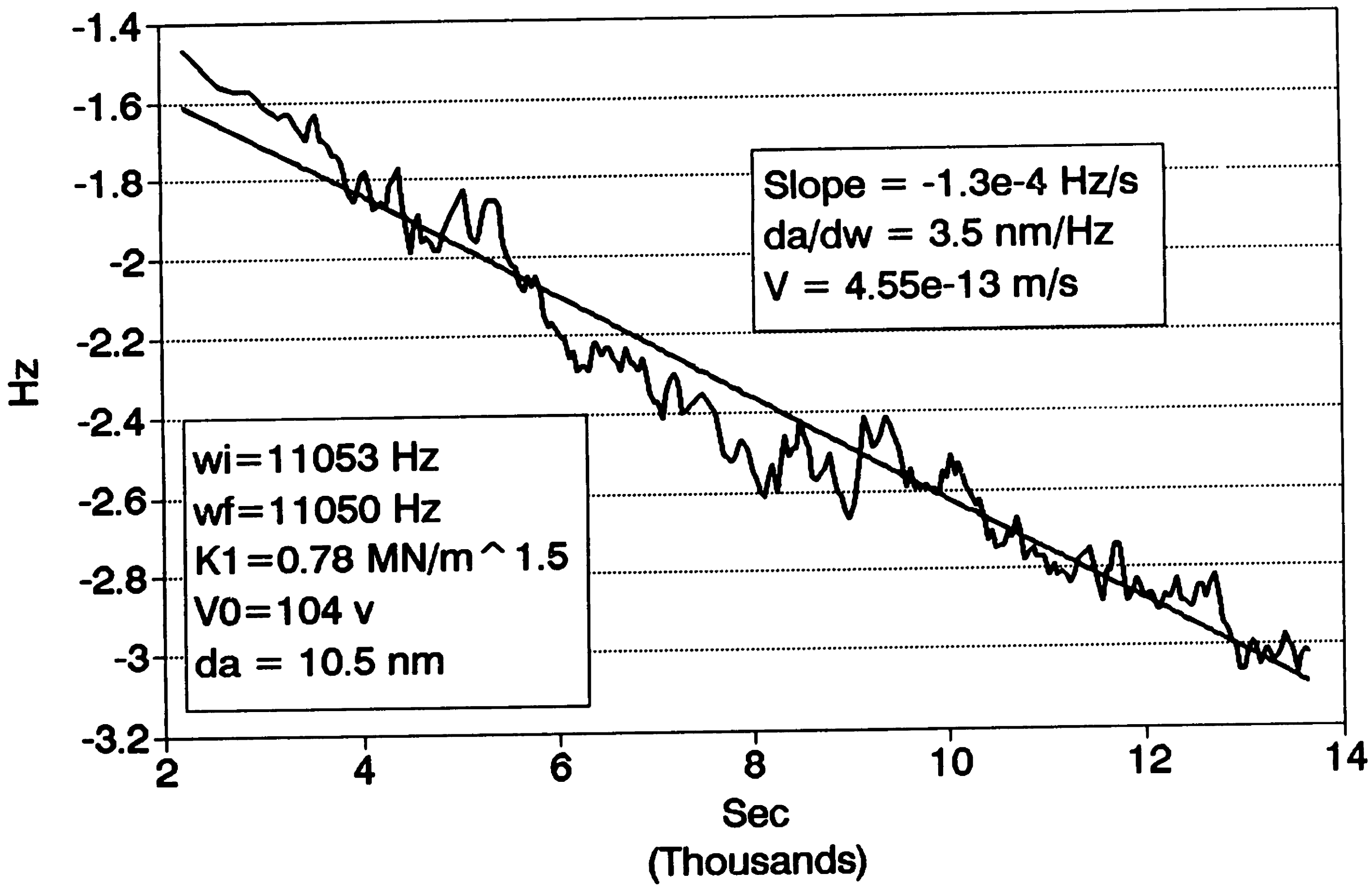
102491 Run 5

Last 15,000 seconds



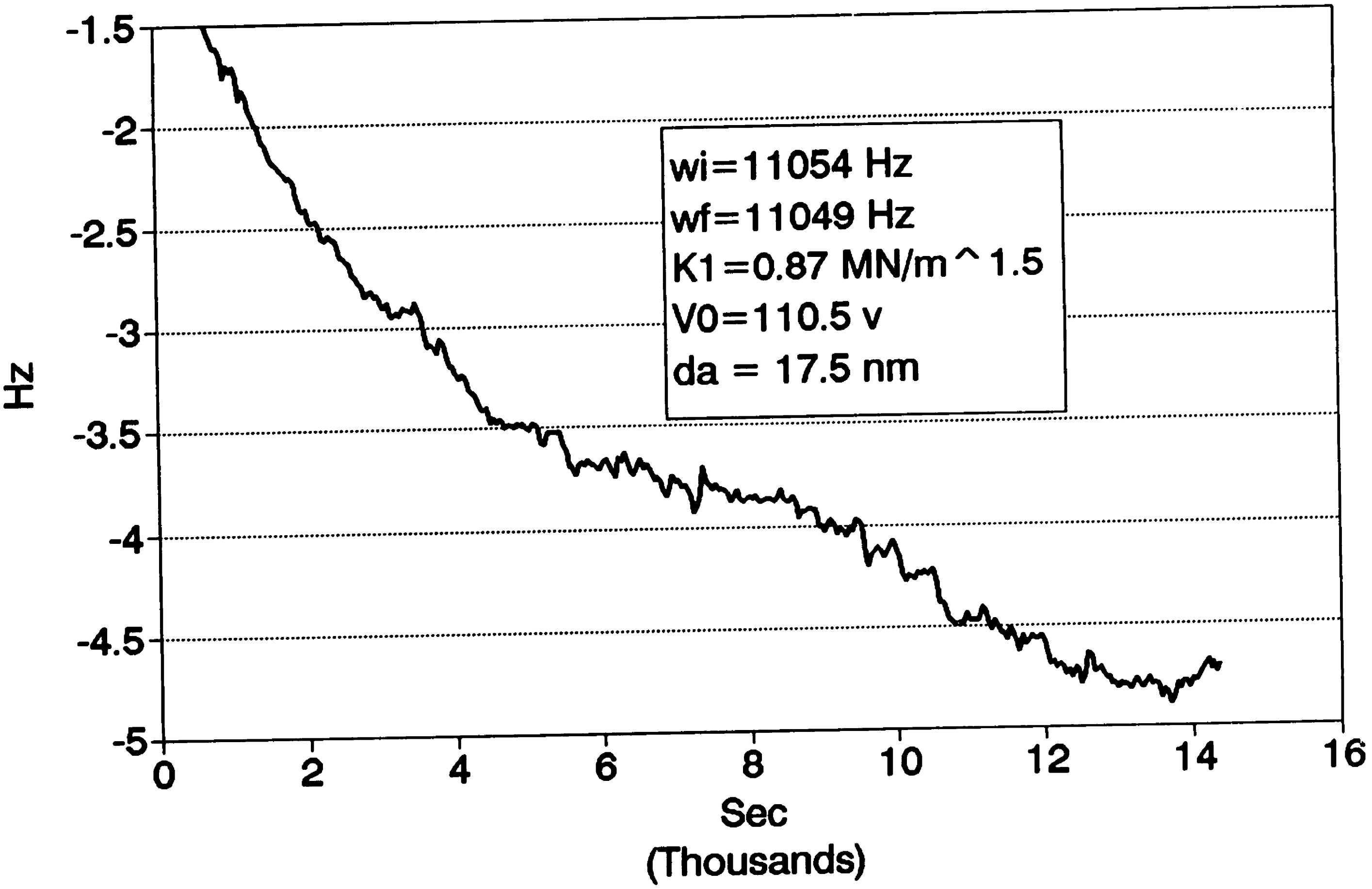
102491 run 6

averaged data

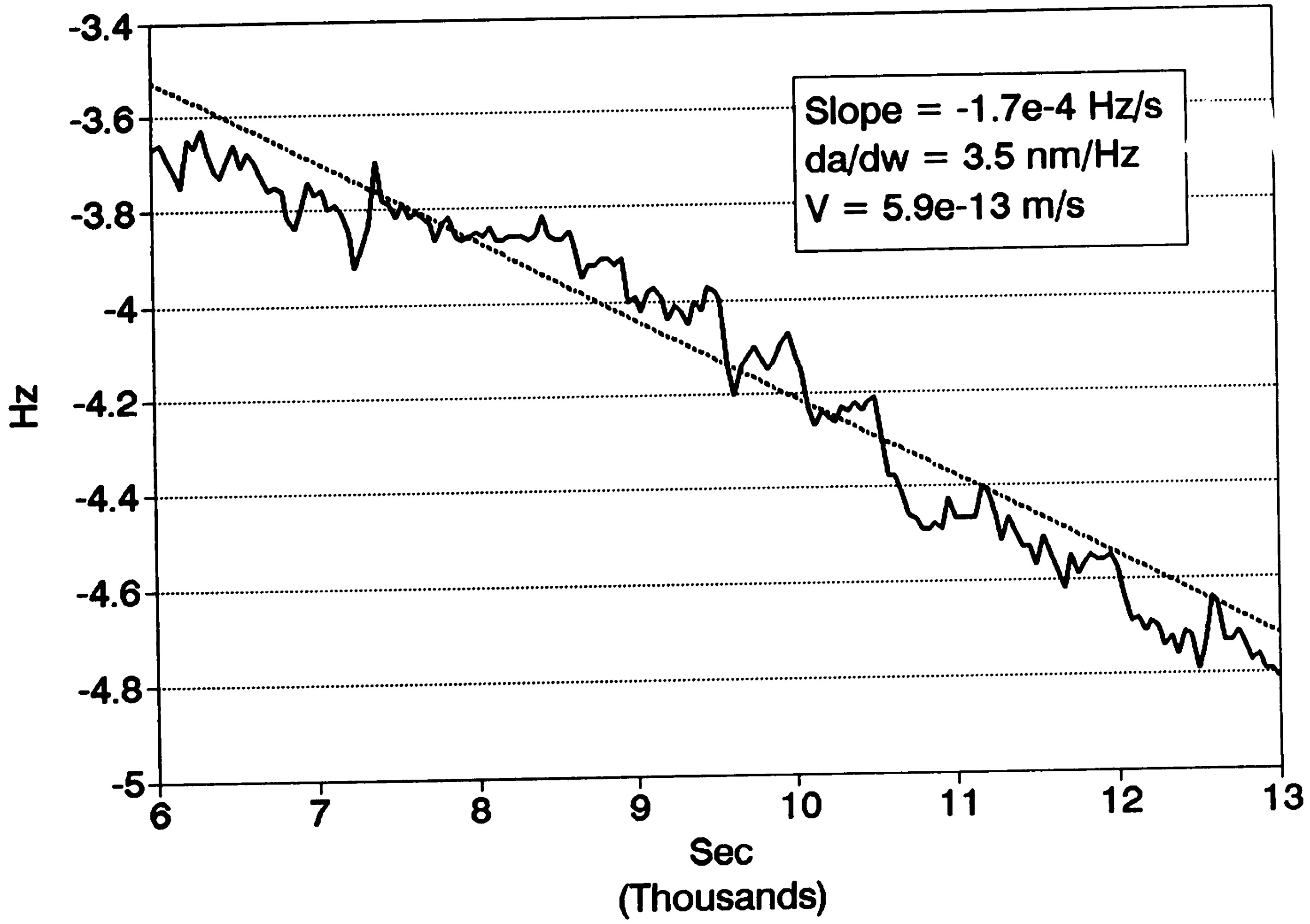


102491 run 7

averaged data



102491 Run 7


Slope = -1.7e-4 Hz/s
da/dw = 3.5 nm/Hz
V = 5.9e-13 m/s
-3.4
-3.6
-3.8
-4
-4.2
-4.4
-4.6
-4.8
-5
Hz
6
7
8
9
10
11
12
13
Sec
(Thousands)

Appendix E

FMEAS2.PAS

The FMEAS2.PAS program is the data acquisition program that commands the CTM05 frequency counter board to measure frequency and elapsed time. The system operates in three modes. In the first mode, the frequency is measured and averaged over twenty samples. The average frequency is then compared to a baseline average frequency recorded at the start of the test run. If the measured average frequency differs from the baseline average frequency by less than 1.5 Hz no data is stored on disk. When the frequency shift exceeds 1.5 Hz, the system then operates in mode two where the average frequency is continuously written to the disk. If the frequency shift is greater than 1.5 Hz and the rate of change greater than 1 Hz per average sample, the system operates in mode three where each instantaneous frequency measurement is written to the disk.

```
program fmeas2;
uses
  dos,
  crt,
  fproc2,
  fdriv2;

label 99;

begin{main program}
  initialize;
  if not quit then
    repeat
    begin{continuous frequency measurements}
      n:=n+1;
      j:=j+1;
      i:=i+1;
      rel_time:=rel_time+sample_rate;
      if ((j=jmax) and not quit) then calibrate(sample_rate);

      ctm05_measure_freq(freq[i],quit);
      if quit then goto 99;

      if (not averaging) and (not stream_data) then
        writeln('not writing',freq[i]:15:2);
      if averaging then
        writeln('averaging data',freq[i]:15:2);
      if stream_data then
      begin
        writeln('streaming data',freq[i]:15:2);
        dw:=freq[i]-freq0;
        writeln(outfile,n:13,'     ',rel_time:13,freq[i]:15:2,dw:15:2);
        nline:=nline+1;
      end;

      if (keypressed and (readkey='r')) then
      begin
        writeln('----------------FREQUENCY RESET---------------');
        reset:=true;
      end;

      {check frequncy shift and rate of change}
      favg:=favg+freq[i];
      if i = imax then
      begin
        favg:=favg/imax;
        df:=favg-favg0;
        dw:=favg-freq0;
        if reset then
        begin
          dw:=0;
          df:-0;
          favg0:-favg;
          freq0:-favg;
          reset:-false;
          writeln(outfile,
          '----------------------FREQUENCY RESET--------------------------');
          writeln(outfile,n:13,'     ',rel_time:13,favg:15:2,dw:15:2);
          nline:-nline+2;
        end;
        if n - imax then
        begin
          df:-0;
          freq0:-favg;
          writeln(outfile,n:13,'     ',rel_time:13,favg:15:2,dw:15:2);
          nline:-nline+1;
        end;

        if (abs(dw) > {1.5}1) and (abs(df) < 1.0{0.2}) then
        begin
          averaging:-true;
          if not stream_data then
          begin
            writeln(outfile,n:13,'     ',rel_time:13,favg:15:2,dw:15:2);
            nline:-nline+1;
          end;
        end
        else averaging:-false;

        if (abs(df) >- 1.0{C.2}) and (abs(dw) >- 1.5{1}) then
stream_data:-true
        else stream_data:-false;

        i:-0;
        favg0:-favg;
        favg:-0;
      end; {if i - imax}

      if nline >- 3000 then
      begin
        nline:-0;
        close(outfile);
        filenum:-filenum+1;
        open_data_file(filename,filenum);
      end;
    end;{continuous measurement}
99: until (keypressed and (readkey-stop)) or quit;

    quit:-true;
    {get total elapsed time from counters 1 and 2}
    read_time_line(quit,t);
    {add footer to last data file}
    writeln(outfile,
    '-----------------------------------------------------------');
    write(outfile,'Test Stopped: ');
    print_timeanddate;
    write('V0 = ');
    readln(V0);
    write('Td = ');
    readln(Td);
    write('Ta = ');
    readln(Ta);
    write('Ed = ');
    readln(Ed);
    write('Eu = ');
    readln(Eu);
    writeln(outfile,'Test Conditions:');
    writeln(outfile,'V0 = ',V0,'  v');
    writeln(outfile,'Td = ',Td,'  degC');
    writeln(outfile,'Ta = ',Ta,'  degC');
    writeln(outfile,'Ed = ',Ed,'  v');
    writeln(outfile,'Eu = ',Eu,'  v');
    writeln(outfile,'Total Samples = ',n:13);
    writeln(outfile,'Elapsed Time = ',t:13);
    writeln(outfile,'Sample Rate = ',1/sample_rate:10:3);
    writeln(outfile,'Data Files = ',filename,'0',' - ',filename,filenum);
    close(outfile);
    chdir('..');
end.{main program}
```

```
unit fproc2;
interface
uses
  fdriv2,
  dos,
  crt;

const
  fsmax = 8.5;                    {max sample rate [samples/s]}
  smax  = 1/fsmax;                {starting sample time [sec/sample]}
  imax = 10;                      {number of samples to average}
  jmax = 1000;                    {number of samples until time line checked}
  stop='q';

var                               {GLOBAL VARIABLES}
  reset,
  quit,
  averaging,
  stream_data:boolean;
  n:longint;                      {number of samples}
  filenum,                        {number of files output to disk}
  nline,                          {# of lines written to outfile}
  i,                              {array index}
  j,                              {time line index}
  error:integer;                  {error code number}
  accum_counts:word;              {no. of counts of ctounter# 5 during
 gate_interval}
  freq:array[1..imax] of real;    {measured frequency}
  df,                             {rate of avg. frequency shift, Hz/s}
  favg0,                          {initial average frequency, Hz}
  favg,                           {average frequency, Hz}
  freq0,                          {initial frequency, Hz}
  dw,                             {average frequency shift, Hz}
  rel_time,                       {incremental time from sample rate, s}
  t,                              {real time from counters 1 & 2, s}
  t0,                             {time of sample n-1, s}
  dt,                             {time between samples, s}
  sample_rate:real;               {actual sample rate s/sample}
  outfile:text;
  V0,                             {drive voltage, volts}
  Td,                             {device temp, degC}
  Ta,                             {amb temp, degC}
  Ed,                             {down sens. voltage, volts}
  Eu:string;                      {up sens. voltage, volts}
  filename:string[12];

 procedure calibrate(var sample_rate:real);
procedure print_timeanddate;

procedure initialize;

procedure open_data_file(filename:string; filenum:integer);

implementation
procedure print_timeanddate;
const
  s10 = '          ';          {skip 10 spaces}

var
  hour,min,sec,sec100,dayofweek,day,month,year:word;
  dayname:string[9];

begin
  gettime(hour,min,sec,sec100);
  getdate(year,month,day,dayofweek);
  case dayofweek of
    1: dayname:='Monday';
    2: dayname:='Tuesday';
    3: dayname:='Wednesday';
    4: dayname:='Thursday';
    5: dayname:='Friday';
    6: dayname:='Saturday';
    0: dayname:='Sunday';
  end;
  write(outfile,'Date = ',dayname,' ',month,'/',day,'/',year);
  writeln(outfile,s10,'Time = ',hour,':',min,':',sec);
end; {timeanddate}

procedure open_data_file(filename:string; filenum:integer);
const
  appen='.dat';
var
  filenumc:string;
begin
  str(filenum,filenumc);
  filename:=filename+filenumc+appen;
  assign(outfile,filename);
  rewrite(outfile);
end;

procedure initialize;
var
  rnum,                            {run number}
  dnum:string;                     {device number}
  ioerr:word;
```

```
begin {initialize}
  {initiaize global variables and indices}
  sample_rate:=smax;
  rel_time:=0;
  dw:=0;
  filenum:=0;
  nline:=0;
  n:=0;
  favg:=0;
  i:=0;
  j:=0;
  reset:=false;
  stream_data:=false;
  averaging:=false;
  quit:=false;
  error:=0;
  filename:='TEST';

  {initialize ctm05_board and check for errors}
  ctm05_init(error);
  if error > 0 then quit:=true
  else
  begin {check for frequency measurement errors}
    ctm05_freq_error(error);
    if error > 0 then quit:=true
    else
    begin {initialize output file}
      write('Device # = ');
      readln(dnum);
      write('Run # = ');
      readln(rnum);
      mkdir(dnum+'_'+rnum);
      chdir(dnum+'_'+rnum);
      writeln('Return to start.');
      readln;
      open_data_file(filename,filenum);
      writeln(outfile,'Device number = ',dnum);
      writeln(outfile,'Run number = ',rnum);
      writeln(outfile,
      '----------------------------------------------------------------');
      {set time and date of run}
      write(outfile,'Test started: ');
      print_timeanddate;
      writeln(outfile,
      '----------------------------------------------------------------');
      writeln(outfile,
      '  sample number      time [s]          freq [hz]         dw [hz]');
      writeln(outfile,
      '----------------------------------------------------------------');

      {start time line on counters 1 and 2}
      start_time_line(t);

      {set baseline frequency}
      ctm05_measure_freq(freq0,quit);
      if not quit then
      begin
        writeln(outfile,n:13,'     ',rel_time:13,freq0:15:2,dw:15:2);
        nline:=nline+1;
      end;
    end;{initialize output file}
  end;{check for frequency measurement error}
end; {initialize}

procedure calibrate(var sample_rate:real);
begin {calibrate}
  if n=jmax then t0:=0;
  read_time_line(quit,t);
  dt:=t-t0;
  t0:=t;
  if dt > 0 then sample_rate:=dt/jmax;
  j:=0;
end;

begin {initialization}
end.
```

```
unit fdriv2;
interface
Uses
  crt,
  dos;


procedure ctm05_init(var err_code: integer);

procedure ctm05_freq_error(var err_code:integer);

procedure ctm05_measure_freq(var freq:real; var quit:boolean);

procedure start_time_line(var seconds:real);

procedure read_time_line(quit:boolean;var seconds:real);

implementation
const
  board_num_err = 2;
  base_adr_err = 3;
  interrupt_level_err = 4;
  counter_num_err = 5;
  fout_div_err = 6;
  fout_source_err = 7;
  comp2_err = 8;
  comp1_err = 9;
  tod_err = 10;
  gate_cntrl_err = 11;
  count_edge_err = 12;
  count_source_err = 13;
  special_gate_err = 14;
  reload_err = 15;
  count_repeat_err = 16;
  count_type_err = 17;
  count_dir_err = 18;
  output_cntrl_err = 19;
  counter_command_err = 20;
  select_counter_err = 21;
  counter_data_err = 22;
  digout_data_err = 23;
  count_err = 24;
  start_ipo_err = 25;
  gate_interval_err = 26;
  signal_source_err = 27;

 type
  ctm05_gv_type = record
```

```
                    data_reg,
                    command_reg,
                    digin_reg,
                    digout_reg,
                    iflg: integer;
                  end;

var
  board_num,                    {board number}
  base_adr,                     {base address setting of dip switches on
ctm05}
  fout_div,                     {frequency divider ratio 16}
  fout_source,                  {freq. source: 1MHz, 100KHz, 1KHz, 100Hz}
  comp2,                        {compare mode counter 2, 0=disable, 1=enable}
  comp1,                        {compare mode counter 1, 0=disable, 1=enable}
  tod,                          {time of day for conters 1 & 2}
  gate_interval,                {gate interval for counter 5 in milliseconds}
  signal_source:integer;        {input signal source (1-15) gate1=6 (pin 37)}
  ctm05_gv: array[0..3] of ctm05_gv_type;

function check_range(test_val, low_val, high_val: integer): boolean;
var
  temp: boolean;
begin
  if (test_val >= low_val) and (test_val <= high_val) then temp := false
  else temp := true;
  check_range := temp;
end;

procedure cmt05_at_delay;
begin
end;

procedure ctm05_print_error(err_code: integer);
var
  es: string[80];

begin{ctm05_print_error}
  case err_code of
    0: es := 'No error';
    1: es := 'CTM05 driver not initialized - use ctm05_init';
    2: es := 'Board number out of range (0-5)';
    3: es := 'Base address of CTM05 out of range ($100 - $3fc)';
    4: es := 'Interrupt level out of range (2-7)';
    5: es := 'Counter number out of range (1-5)';
    6: es := 'Fout divider ratio error in ctm05_init  (0-15)';
    7: es := 'Fout source error in ctm05_init (0-15)';
    8: es := 'Compare 2 disable/enable error in ctm05_init (0-1)';
    9: es := 'Compare 1 disable/enable error in ctm05_init (0-1)';
    10: es := 'Time of data mode control error in ctm05_init (0-3)';
    11: es := 'Gate control error in ctm05_set_counter_mode_reg (0-7)';
    12: es := 'Count edge error in ctm05_set_counter_mode_reg (0-1)';
    13: es := 'Count source error in ctm05_set_counter_mode_reg (0-15)';
    14: es := 'Disable special gate error in ctm05_set_counter_mode_reg (0-
1)';
    15: es := 'Reload error in ctm05_set_counter_mode_reg (0-1)';
    16: es := 'Count once/count repetitively error in
ctm05_set_counter_mode_reg  (0-1)';
    17: es := 'Binary count/ BCD count error in ctm05_set_counter_mode_reg (0-
1)';
    18: es := 'Count down/up error in ctm05_set_counter_mode_reg (0-1)';
    19: es := 'Output control error in ctm05_set_counter_mode_reg (0-5 but not
3)';
    20: es := 'Command error in ctm05_multiple_counter_control (1-6)';
    21: es := 'Select counter error in ctm05_multiple_counter_control (0-1)';
    22: es := 'Output data error in ctm05_cset (-32768 to 32767)';
    23: es := 'Output data error in ctm05_bins (0-255)';
    24: es := 'Count error in ctm05_store_on_interrupt (1-32767)';
    25: es := 'Start ipo error in ctm05_store_on_interrupt (0-1)';
    26: es := 'Gate interval error in ctm05_store_on_interrupt (1-32767)';
    27: es := 'Signal source error in ctm05_store_on_interrupt (1-15)';
  end;{end case err_code}
  writeln(es);
end;{ctm05_print_error}

procedure ctm05_init(var err_code: integer);

var
  temp: word;

begin{ctm05_init}
  if check_range(board_num,0,3) then err_code := board_num_err
  else
    if check_range(base_adr,$100,$3fc) then err_code := base_adr_err
    else
      if check_range(fout_div,0,15) then err_code := fout_div_err
      else
        if check_range(fout_source,0,15) then err_code := fout_source_err
        else
          if check_range(comp2,0,1) then err_code := comp2_err
          else
            if check_range(comp1,0,1) then err_code := comp1_err
            else
              if check_range(tod,0,3) then err_code := tod_err
              else
                with ctm05_gv[board_num] do
```

```
               begin{with ctm05_gv}

                 {set values of i/o ports}
                 data_reg := base_adr;
                 command_reg := base_adr + 1;
                 digin_reg := base_adr + 2;
                 digout_reg := base_adr + 3;

                 {set to master mode register to 1100101011110000 base 2;
                  51952 base 10; p.9 ctm05 manual
                  mm15 = 1: bcd division
                  mm14 = 1: disable data pointer control
                  mm13 = 0: 0 8 bit data bus
                  mm12 = 0: fout on
                  mm11 - mm8: 1010; fout_div shl 8; fout_div = 10
                  mm7 - mm4: 1111; fout_source shl4;
                             fout_source = 15 = f5 =100 hz
                  mm3 = 0: comparator 2 diabled
                  mm2 = 0: comparator 1 diabled
                  mm1 - mm0 = 00: time of day diabled}

                 temp := $c000 or (fout_div shl 8) or (fout_source shl 4) or
                         (comp2 shl 3) or (comp1 shl 2) or tod;

                 port[command_reg] := $FF;      {master reset}
                 cmt05_at_delay;
                 port[command_reg] := $17;      {load master mode register}
                 cmt05_at_delay;
                 port[data_reg] := lo(temp);    {write first 8 bytes of temp}
                 cmt05_at_delay;
                 port[data_reg] := hi(temp);    {write second 8 bytes of
temp}
                 err_code := 0;
                 iflg := 0;
                end;{with ctm05_gv}
  if err_code > 0 then
  begin
    writeln('INITIALIZATION ERROR:');
    ctm05_print_error(err_code);
  end;
end;{ctm05_init}

procedure ctm05_freq_error(var err_code:integer);
begin
  if check_range(board_num,0,3) then err_code := board_num_err
  else
    if check_range(gate_interval,1,32767) then err_code := gate_interval_err
    else
```

```
      if check_range(signal_source,1,15) then err_code := signal_source_err;
    if err_code > 0 then
    begin
      writeln('FREQUENCY MEASUREMENT ERROR:');
      ctm05_print_error(err_code);
    end;
end;

procedure ctm05_measure_freq(var freq:real; var quit:boolean);
label 99;
const
  stop='q';
var
  accum_counts,
  temp: word;

 begin{ctm05_measure_freq}
       with ctm05_gv[board_num] do
       begin{with ctm05_gv}
         {counter 4}
         {write to mode register of counter 4: p.12 ctm05 manual}
         port[command_reg] := $4;
         cmt05_at_delay;

         {write gating control, count source selection, count control, and
  output
          control to counter 4}
         {write lo byte first: $22=34=00100010
          - special gate diabled
          - reload from load
          - count repetatively
          - binary count
          - count down
          - tc toggled}
         port[data_reg] := $22;
         cmt05_at_delay;

         {write hi byte:$4 = 4 = 00000100
          - no gating  000
          - count on rising edge of source 4 (pin 15 on
            D connector: this will be the measured frequency input 00100}
         port[data_reg] := $4;
         cmt05_at_delay;

         {write gate_interval (100 ms) to counter 4 load register}
         port[command_reg] := $c;
         cmt05_at_delay;
         port[data_reg] := lo(gate_interval);
```

```
cmt05_at_delay;
port[data_reg] := hi(gate_interval);
cmt05_at_delay;

{Load and arm counter 4}
port[command_reg] := $68;
cmt05_at_delay;

{set counter 5 mode register}
port[command_reg] := $5;
temp := (signal_source shl 8) or $8029;   {signal_source = 11 = 1mhz}
{counter 5 mode register set to:1000101100101001
 - counter counts on high level of gate 5;
   output of counter 4 (pin 32) hard wired to counter 5 gate (pin 12)
 - counts on rising edge of 1 mhz source
 - special gate disabled
 - reload from load
 - count repetatively
 - binary count
 - count up
 - output active at high tc pulse}
port[data_reg] := lo(temp);
cmt05_at_delay;
port[data_reg] := hi(temp);
cmt05_at_delay;

{counter 5 load register}
port[command_reg] := $d;

{initializing counter 5 load register to zero}
port[data_reg] := 0;
cmt05_at_delay;
port[data_reg] := 0;
cmt05_at_delay;

{waiting for counter 4 tc toggled output to go hi}
repeat
until ((port[command_reg] and $10) = $10) or
      (keypressed and (readkey = stop));
if port[command_reg] and $10 <> $10 then
begin
  quit:=true;
  goto 99;
end;

{waiting for counter 4 tc toggled output to go lo}
repeat
until ((port[command_reg] and $10) <> $10) or
```

```
              (keypressed and (readkey = stop));
        if port[command_reg] and $10 = $10 then
        begin
          quit:=true;
          goto 99;
        end;

        {load and arm counter 5}
        port[command_reg] := $70;
        cmt05_at_delay;

        {waiting for counter 4 output to go hi}
        repeat
        until ((port[command_reg] and $10) = $10) or
              (keypressed and (readkey = stop));
        if port[command_reg] and $10 <> $10 then
        begin
          quit:=true;
          goto 99;
        end;

        {waiting for counter 4 output to go lo}
        repeat
        until ((port[command_reg] and $10) <> $10) or
              (keypressed and (readkey = stop));
        if port[command_reg] and $10 = $10 then
        begin
          quit:=true;
          goto 99;
        end;

        {save counter 5 count in hold register}
        port[command_reg] := $b0;
        cmt05_at_delay;

        {data pointer at counter 5 hold register}
        port[command_reg] := $15;
        cmt05_at_delay;

        {reading counter 5 hold register: lo byte first}
        accum_counts := port[data_reg];
        accum_counts := accum_counts or (port[data_reg] shl 8);
        freq:=gate_interval*1e6/(accum_counts);
      end;{with ctm05_gv}
99: end;{ctm05_measure_freq}

procedure start_time_line(var seconds:real);
begin
```

```
with ctm05_gv[board_num] do
begin
  seconds:=0;

  {write to counter 1 mode register}
  port[command_reg]:=$1;

  {counter 1 mode p.12 ctm05 manual:
  - high level gate 5         100
  - count on rising edge      0
  - 100 hz freq. source (f5)  1111
  - special gate disabled     0
  - reload from load          0
  - count repetatively        1
  - binary count              0
  - count down                0
  - output active at TC       000 }

  cmt05_at_delay;
  port[data_reg]:=lo($8F20);
  cmt05_at_delay;
  port[data_reg]:=hi($8F20);

  {set load register of counter 1 to 6000}
  cmt05_at_delay;
  port[command_reg]:=$9;
  cmt05_at_delay;
  port[data_req]:=lo(6000);
  cmt05_at_delay;
  port[data_reg]:=hi(6000);

  {write to counter 2 mode register}
  cmt05_at_delay;
  port[command_reg]:=$2;

  {set counter 2 mode:
  - no gating               000
  - count on rising edge    0
  - output of counter 1     0000
  - special gate diabled    0
  - reload from load        0
  - count repetatively      1
  - binary count            0
  - count up                1
  - no output               000}

  cmt05_at_delay;
  port[data_reg]:=lo($28);
```

```
    cmt05_at_delay;
    port[data_reg]:=hi($28);
    cmt05_at_delay;

    {load counter 2 with zero}
    port[command_reg]:=$A;
    cmt05_at_delay;
    port[data_reg]:=0;
    cmt05_at_delay;
    port[data_reg]:=0;

    {load and arm counters 1 and 2}
    cmt05_at_delay;
    port[command_reg]:=$63;
  end;
end; {start_time_line}

procedure read_time_line(quit:boolean;var seconds:real);
var
  minutes:real;
  min,
  sec:word;

begin
  with ctm05_gv[board_num] do
  begin
    sec:=0;
    min:=0;
    minutes:=0;
    seconds:=0;

    {disarm and save counters 1 and 2 in hold register}
    if quit then port[command_reg]:=$83
    else port[command_reg]:=$A3;

    {read seconds from counter 1}
    cmt05_at_delay;
    port[command_reg]:=$11;
    cmt05_at_delay;
    sec:=port[data_reg];
    cmt05_at_delay;
    sec:=sec or (port[data_reg] shl 8);
    seconds:=sec;

    {read minutes from counter 2}
    port[command_reg]:=$12;
    cmt05_at_delay;
    min:=port[data_reg];
    cmt05_at_delay;
    min:=min or (port[data_reg] shl 8);
    minutes:=min;
    seconds:=(6000-seconds)/100+60*minutes;
  end;
end; {stop_time_line}

begin {initialization}
  clrscr;
  board_num := 0;
  base_adr := $310;              {784 decimal}
  fout_div := 10;
  fout_source :=15;              {100 Hz}
  comp1 := 0;
  comp2 := 0;
  tod := 0;
  gate_interval := 600;          {gate_interval=10 for 100 ms gate interval for
ctr 5 (set by ctr 4)}
                                 {use a gate interval of 200 for 8 micron thick
devices (run 061891)}
                                 {use a gate interval of 100 for 7 micron
devices run 040991}
  signal_source := 11;           {1 mhz signal source to ctr 5}
end.
```

```
{$I-}
{$N+}
program avg;
uses
  dos,
  crt;

const
  nmax=1000;

var
  option:char;
  filecheck:pathstr;
  dirname,
  file_dir,
  orig_dir,
  file_no,
  in_filename,
  out_filename:string;
  infile,
  outfile:text;
  i,
  j,
  k,
  n,
  fileno:integer;
  nsample:longint;
  k1,
  dadt,
  f,
  t,
  dw:array[1..nmax] of double;
  navg,
  k1avg,
  dadtavg,
  favg,
  tavg,
  dwavg:double;

begin
  clrscr;
  writeln('Choose option.');
  writeln('1 = Average a series of raw data files (TEST0, TEST1,...)');
  writeln('2 = Average a K1 vs da/dt date file');
  writeln('3 = Quit program');
  repeat
    option:=readkey;
  until option in ['1','2','3'];
  case option of
  '1':begin
        clrscr;
```

```
write('Directory of files to average = ');
readln(dirname);
getdir(0,orig_dir);
if (orig_dir = 'C:\') or (orig_dir = 'D:\') then
file_dir:=orig_dir+dirname
else
file_dir:=orig_dir+'\'+dirname;
chdir(file_dir);
write('Base file name of input = ');
readln(in_filename);
write('Starting file number = ');
readln(j);
write('Ending file number = ');
readln(k);
write('Output data file name = ');
readln(out_filename);
assign(outfile,out_filename);
rewrite(outfile);
write('Number of points to average = ');
readln(n);
if n > nmax then
begin
  writeln('Max number of points to average is ',nmax:10);
  writeln('n = ',nmax);
  n:=nmax;
end;
for fileno:=j to k do
begin
  str(fileno,file_no);
  filecheck:=fsearch(in_filename+file_no+'.dat',file_dir);
  if filecheck = '' then writeln('File ',in_filename+file_no:10,
                               ' does not exist in this directory');
  if filecheck <> '' then
  begin
    assign(infile,in_filename+file_no+'.dat');
    reset(infile);
    clrscr;
    writeln('Working on ',in_filename+file_no+'.dat');
    repeat
    begin
      i:=0;
      favg:=0;
      tavg:=0;
      dwavg:=0;
      navg:=0;
      repeat
      begin
        i:=i+1;
        readln(infile,nsample,t[i],f[i],dw[i]);
        if ioresult = 106 then
        begin
```

```
                i:=i-1;
                readln(infile);
              end
              else
              begin
                favg:=favg+f[i];
                tavg:=tavg+t[i];
                dwavg:=dwavg+dw[i];
                navg:=navg+nsample;
              end;
            end;
            until (i=n) or eof(infile);
            if i > 0 then
            begin
              favg:=favg/i;
              tavg:=tavg/i;
              dwavg:=dwavg/i;
              navg:=navg/i;
              writeln(outfile,navg:15,'  ',tavg:15,{,favg:15}dwavg:15:2);
            end;
          end;
          until eof(infile);
          close(infile);
        end;
      end;{for fileno=j to k}
      close(outfile);
      chdir(orig_dir);
      writeln;
      writeln('Finished');
    end;{case option of 1}
'2':begin
      clrscr;
      write('File name of input = ');
      readln(in_filename);
      getdir(0,file_dir);
      filecheck:=fsearch(in_filename,file_dir);
      if filecheck = '' then
      begin
        writeln('File ',in_filename:12,
        ' does not exist in directory',file_dir);
        halt;
      end;
      assign(infile,in_filename);
      reset(infile);
      write('Output data file name = ');
      readln(out_filename);
      assign(outfile,out_filename);
      rewrite(outfile);
      write('Number of points to average = ');
      readln(n);
      if n > nmax then
```

```
        begin
          writeln('Max number of points to average is ',nmax:10);
          writeln('n = ',nmax);
          n:=nmax;
        end;
        repeat
        begin
          i:=0;
          k1avg:=0;
          dadtavg:=0;
          repeat
          begin
            i:=i+1;
            readln(infile,k1[i],dadt[i]);
            if ioresult = 106 then
            begin
              writeln('Incorrect numeric format on input.');
              close(infile);
              close(outfile);
              erase(outfile);
              halt;
            end
            else
            begin
              k1avg:=k1avg+k1[i];
              dadtavg:=dadtavg+dadt[i];
            end;
          end;
          until (i=n) or eof(infile);
          if i > 0 then
          begin
            k1avg:=k1avg/i;
            dadtavg:=dadtavg/i;
            writeln(outfile,k1avg:15,dadtavg:15);
          end;
        end;{repeat}
        until eof(infile);
        close(infile);
        close(outfile);
        writeln;
        writeln('Finished');
      end; {case option 2}
  '3':begin
      end;
  end; {case}
end.
```

Appendix F

INTEGRAT.PAS

The program INTEGRAT.PAS numerically integrates a second order differential equation using the trapezoidal rule. This program was used to compute the frequency response of the cracked and uncracked first mode equations of motion of the cantilever beam system. The results of this simulation were compared to the solutions obtained from linearizing the equations of motion using describing functions.

```
{$N+}{8087 mode}
program integrat;
uses
  dos,
  crt;

const
  nptsmax=200;

var
  k1,
  k2,
  c,
  m,
  f,
  xmin,
  xmax,
  wdf,
  dw,
  w1,
  w2,
  w,
  wi,
  wf:extended;
  tin,
  tfin,
  npts,
  i:integer;
  omega,
  fdriv,
  umax,
  umin:array[1..nptsmax] of double;
  outfile:text;

  procedure newmark;
  var
    tf,
    t,
    n:longint;
    dt,
    alpha,
    delta,
    a0,
    a1,
    a2,
    a3,
    a4,
    a5,
    a6,
    a7,
    ft,
  fhat,
  k,
  khat,
  x,
  xd,
  xdd,
  tx,
  txd,
  txdd:double;
  stopi,
  stopf:boolean;

begin
  dt:=pi/(200*w);
  npts:=100;
  dw:=0.3*2*pi;
  w:=w-npts/2*dw;

  assign(outfile,'intg.sim');
  rewrite(outfile);
  writeln(outfile,'fdriv','         ','omega','          ' ,'umax',
  '          ','umin');

  {itegration constants}
  delta:=0.5;
  alpha:=0.25;
  a0:=1/(alpha*sqr(dt));
  a1:=delta/(alpha*dt);
  a2:=1/(alpha*dt);
  a3:=1/(2*alpha)-1;
  a4:=delta/alpha-1;
  a5:=dt/2*(delta/alpha-2);
  a6:=dt*(1-delta);
  a7:=delta*dt;

  {initial conditions}
  tx:=0;
  txd:=0;
  txdd:=0;
  k:=k1;

  {newmark integration}
  tf:=trunc((8*m/c)/dt);
  for i:=1 to npts do
  begin
    clrscr;
    writeln(npts:2,' ',npts-i:2);
    xmax:=0;
    xmin:=0;
    stopi:=false;
    stopf:=false;
    for t:=0 to tf do
    begin
      khat:=k+a0*m+a1*c;
      ft:=f*sin(w*t*dt);
      fhat:=ft+m*(a0*tx+a2*txd+a3*txdd)+
            c*(a1*tx+a4*txd+a5*txdd);
      x:=fhat/khat;
      xdd:=a0*(x-tx)-a2*txd-a3*txdd;
      xd:=txd+a6*txdd+a7*xdd;
      if x >= 0 then k:=k1 else k:=k2;
      if t >= tf-3*(trunc(2*pi/(w*dt))) then
      begin
        if x > xmax then xmax:=x;
        if x < xmin then xmin:=x;
        if (not stopi) and (tx < 0) and (x > 0) then
        begin
          tin:=t;
          stopi:=true;
        end;
        if (not stopf) and (tx > 0) and (x < 0) then
        begin
          tfin:=t;
          stopf:=true;
        end;
      end;
      tx:=x;
      txdd:=xdd;
      txd:=xd;
    end;
    fdriv[i]:=w/(2*pi);
    omega[i]:=1/(2*abs(tfin-tin)*dt);
    umax[i]:=xmax;
    umin[i]:=xmin;
    w:=w+dw;
    writeln(outfile,fdriv[i]:15,' ',omega[i]:15,
            ' ',umax[i]:15,' ',umin[i]:15);
  end;
  close(outfile);
end;

procedure sim_non;
var
  x10,
  x20,
  dx1,
  x1,
  x2,
  p1,
  p2,
  pd1,
  pd2,
  t,
  phi1,
  phi2,
  w1d,
  w2d,
  a1,
  a2,
  b1,
  b2,
  alpha1,
  beta1,
  alpha2,
  beta2,
  delta1,
  delta2,
  gamma1,
  gamma2:extended;
  n:longint;

begin
  phi1:=arctan((c/m*w)/(sqr(w1)-sqr(w)));
  if phi1 < 0 then phi1:=pi+phi1;
  phi2:=arctan((c/m*w)/(sqr(w2)-sqr(w)));
  if phi2 < 0 then phi2:=pi+phi2;
  w1d:=sqrt(sqr(w1)-sqr(c/(2*m)));
  w2d:=sqrt(sqr(w2)-sqr(c/(2*m)));

  {IC at t=0}
  a1:=f/m*(c/(2*m*w1d)*sin(phi1)-w/w1d*cos(phi1))/
      sqrt(sqr(sqr(w1)-sqr(w))+sqr(c/m*w));;
  b1:=f/m*sin(phi1)/sqrt(sqr(sqr(w1)-sqr(w))+sqr(c/m*w));
  a2:=0;
  b2:=0;
  n:=0;
  x10:=0;

  {IC at t=n*pi/w}
  repeat
  begin
    n:=n+1;
    t:=n*pi/w;

    {p1:=f/m*sin(w*t-phi1)/sqrt(sqr(sqr(w1)-sqr(w))+sqr(c/m*w)); }
    {p2:=f/m*sin(w*t-phi2)/sqrt(sqr(sqr(w2)-sqr(w))+sqr(c/m*w)); }
    if odd(n) then
    begin
      p1:=f/m*sin(phi1)/sqrt(sqr(sqr(w1)-sqr(w))+sqr(c/m*w));
      p2:=f/m*sin(phi2)/sqrt(sqr(sqr(w2)-sqr(w))+sqr(c/m*w));
      pd1:=-f/m*w*cos(phi1)/sqrt(sqr(sqr(w1)-sqr(w))+sqr(c/m*w));
      pd2:=-f/m*w*cos(phi2)/sqrt(sqr(sqr(w2)-sqr(w))+sqr(c/m*w));
    end
```

```
      else
      begin
        p1:=-f/m*sin(phi1)/sqrt(sqr(sqr(w1)-sqr(w))+sqr(c/m*w));
        p2:=-f/m*sin(phi2)/sqrt(sqr(sqr(w2)-sqr(w))+sqr(c/m*w));
        pd1:=f/m*w*cos(phi1)/sqrt(sqr(sqr(w1)-sqr(w))+sqr(c/m*w));
        pd2:=f/m*w*cos(phi2)/sqrt(sqr(sqr(w2)-sqr(w))+sqr(c/m*w));
      end;

      {pd1:=f/m*w*cos(w*t-phi1)/sqrt(sqr(sqr(w1)-sqr(w))+sqr(c/m*w)); }
      {pd2:=f/m*w*cos(w*t-phi2)/sqrt(sqr(sqr(w2)-sqr(w))+sqr(c/m*w)); }

      alpha1:=exp(-c/(2*m)*t)*sin(w1d*t);
      beta1:=exp(-c/(2*m)*t)*cos(w1d*t);
      delta1:=exp(-c/(2*m)*t)*w1d*cos(w1d*t)-c/(2*m)*exp(-
c/(2*m)*t)*sin(w1d*t);
      gamma1:=-c/(2*m)*exp(-c/(2*m)*t)*cos(w1d*t)-w1d*exp(-
c/(2*m)*t)*sin(w1d*t);

      alpha2:=exp(-c/(2*m)*t)*sin(w2d*t);
      beta2:=exp(-c/(2*m)*t)*cos(w2d*t);
      delta2:=exp(-c/(2*m)*t)*w2d*cos(w2d*t)-c/(2*m)*exp(-
c/(2*m)*t)*sin(w2d*t);
      gamma2:=-c/(2*m)*exp(-c/(2*m)*t)*cos(w2d*t)-w2d*exp(-
c/(2*m)*t)*sin(w2d*t);

      if odd(n) then
      begin
        {b2:=(-delta2*(p1-p2)+alpha2*(pd1-pd2)-
            delta2*alpha1*a1-
beta1*delta2*b1+alpha2*delta1*a1+alpha2*gamma1*b1)/
            (alpha2*gamma2-delta2*beta2);
        a2:=(gamma2*(p1-p2)-beta2*(pd1-pd2));
        a2:=a2+(alpha1*gamma2*a1+beta1*gamma2*b1-
            delta1*beta2*a1-beta2*gamma1*b1);
        a2:=a2/(alpha2*gamma2-delta2*beta2); }
        a2:=(-p2*gamma2-beta2*(pd1-pd2+a1*delta1+b1*gamma1))/(alpha2*gamma2-
delta2*beta2);
        b2:=(p2*delta2+alpha2*(pd1-pd2+a1*delta1+b1*gamma1))/(alpha2*gamma2-
delta2*beta2);
      end
      else
      begin
        {a1:=(gamma1*(alpha2*a2+beta2*b2+p2-p1)-
            beta1*(delta2*a2+gamma2*b2+pd2-pd1))/(alpha1*gamma1-beta1*delta1);
        b1:=(-delta1*(alpha2*a2+beta2*b2+p2-p1)+
            alpha1*(delta2*a2+gamma2*b2+pd2-pd1))/(alpha1*gamma1-
beta1*delta1);}
        a1:=-(p1*gamma1+beta1*(pd2-pd1+a2*delta2+b2*gamma2))/(alpha1*gamma1-
delta1*beta1);
        b1:=(p1*delta1+gamma1*(pd2-pd1+a2*delta2+b2*gamma2))/(alpha1*gamma1-
delta1*beta1);
```

```
      end;
    end;
    x1:={p1+}a1*alpha1+b1*beta1;
    x2:={p2+}a2*alpha2+b2*beta2;
    if n=trunc(w1*8*m/(pi*c)) then
    begin
      x10:=x1;
    end;
    until n = trunc(w1*8*m/(pi*c));
    x10:=x1;
  end;

begin
  f:=1e-3;
  m:=1;
  c:=18;
  k1:=5.96e9;
  k2:=4.24e9;
  wdf:=sqrt((k1+k2)/(2*m));
  w1:=sqrt(k1/m);
  w2:=sqrt(k2/m);
  w:=11245*2*pi;

  newmark;
  {sim_non;}
end. {end mainprogram}
```

Appendix G

FTEST2.PAS Design Program

The FTEST2.PAS design program is an integral part of this thesis. The program computes the dynamic, electrical, and fatigue properties of the device and serves as the principle modeling tool of the experiment for data reduction and analysis. Subsections of this part of the appendix include the input files for the generation 1 design, and the 091890, 102491, 110291, 011092, 020392 devices.

G.1 Generation 1: T38.inp

```
    10.000     1      lab length of segment ab [mic]
   120.000     2      lbc length of segment bc [mic]
     2.000     3      x1 edge clearances [mic]
    30.000     4      x2 distance between torquer and sensor plates [mic]
    30.000     5      x3 distance between sensor plate and plate 2 [mic]
   120.000     6      lpy width of plate [mic]
     8.000     7      lh1x length of hole at torquer [mic]
    30.000     8      lh1y width of at torquer hole [mic]
     8.000     9      lh3y length of hole at sensor [mic]
    45.000    10     lbh3y loc of cg of hole 3 [mic]
    27.000    11     lbh4y loc of cg of hole 4 [mic]
     9.000    12     lbh5y loc of cg of hole 5 [mic]
    60.000    13     lp2x length of tip mass [mic]
    80.000    14     lp2y width of tip mass [mic]
    15.000    15     lp2z thickness of tip mass [mic]
     0.000    16     lp1z thickness of plate mass [mic]
    11.500    17     b0 width of beam at point a (fixed end) [mic]
    11.500    18     bb width at V-notch [mic]
     2.900    19     h thickness of beam [mic]
     1.000    20     hb V-notch depth [mic]
    15.000    21     ltx length of torquer [mic]
    20.000    22     lsx length of sensor  [mic]
     6.400    23     d0s distance from sensor elec. to plating [mic]
     6.400    24     d0t distance from torquer elec. to plateing [m]
  5.5E-0013   25     cfb feedback capacitance [f]
     1.000    26     vexc excitation voltage on sensor [v]
     0.0100   27     da crack length [mic]
     0.0500   28     rtip tip radius of V-notch [mic]
    15.0000   29     wservo servo bandwidth [hz]
    10.0000   30     v0 applied torquer voltage [v]
    70.0000   31     h0p plate airgap [mic]
     1.0000   32     vbias sensor bias voltage [v]
          2   33     ndof # DOF of analysis (2 or 3)
          0   34     failure simulation (1 = yes, 0 = no)
          0   35     fatigue simulation (1 = yes, 0 = no)
     0.0004   36     vsiO2 crack velocity [m/s]
    32.0000   37     nsiO2 fatigue constant

New file created on:
Date of run: Saturday 2/15/1992          Time of run: 20:35:16
-------------------------------------------------------------
Test case for comparison with ABAQUS FEA.
```

G.2 091890.inp

```
   10.000     1      lab length of segment ab [mic]
  120.000     2      lbc length of segment bc [mic]
    2.000     3      x1 edge clearances [mic]
   30.000     4      x2 distance between torquer and sensor plates [mic]
   30.000     5      x3 distance between sensor plate and plate 2 [mic]
  124.000     6      lpy width of plate [mic]
    8.000     7      lh1x length of hole at torquer [mic]
   30.000     8      lh1y width of at torquer hole [mic]
    8.000     9      lh3y length of hole at sensor [mic]
   45.000    10    lbh3y loc of cg of hole 3 [mic]
   27.000    11    lbh4y loc of cg of hole 4 [mic]
    9.000    12    lbh5y loc of cg of hole 5 [mic]
   60.000    13    lp2x length of tip mass [mic]
   80.000    14    lp2y width of tip mass [mic]
   17.000    15    lp2z thickness of tip mass [mic]
    0.000    16    lp1z thickness of plate mass [mic]
   10.000    17    b0 width of beam at point a (fixed end) [mic]
   10.000    18    bb width at V-notch [mic]
    3.000    19    h thickness of beam [mic]
    2.000    20    hb V-notch depth [mic]
   15.000    21    ltx length of torquer [mic]
   20.000    22    lsx length of sensor  [mic]
    6.400    23    d0s distance from sensor elec. to plating [mic]
    6.400    24    d0t distance from torquer elec. to plateing [m]
 5.5E-0013   25    cfb feedback capacitance [f]
    1.000    26    vexc excitation voltage on sensor [v]
   0.0100    27    da crack length [mic]
   0.0500    28    rtip tip radius of V-notch [mic]
  15.0000    29    wservo servo bandwidth [hz]
  10.0000    30    v0 applied torquer voltage [v]
  70.0000    31    h0p plate airgap [mic]
   1.0000    32    vbias sensor bias voltage [v]
        3    33    ndof # DOF of analysis (2 or 3)
        0    34    failure simulation (1 = yes, 0 = no)
        0    35    fatigue simulation (1 = yes, 0 = no)
   0.0004    36    vsi02 crack velocity [m/s]
  12.0000    37    nsi02 fatigue constant

New file created on:
Date of run: Saturday 2/15/1992          Time of run: 20:49:16
------------------------------------------------------------------
```

G.3 102491.inp

```
    13.000    1    lab length of segment ab [mic]
    60.000    2    lbc length of segment bc [mic]
    15.000    3    x1 edge clearances [mic]
    40.000    4    x2 distance between torquer and sensor plates [mic]
   100.000    5    x3 distance between sensor plate and plate 2 [mic]
   162.000    6    lpy width of plate [mic]
     0.000    7    lh1x length of hole at torquer [mic]
    30.000    8    lh1y width of at torquer hole [mic]
     0.000    9    lh3y length of hole at sensor [mic]
    45.000   10   lbh3y loc of cg of hole 3 [mic]
    27.000   11   lbh4y loc of cg of hole 4 [mic]
     9.000   12   lbh5y loc of cg of hole 5 [mic]
    70.000   13   lp2x length of tip mass [mic]
   130.000   14   lp2y width of tip mass [mic]
    17.000   15   lp2z thickness of tip mass [mic]
     0.000   16   lp1z thickness of plate mass [mic]
    20.000   17   b0 width of beam at point a (fixed end) [mic]
    17.000   18   bb width at V-notch [mic]
     8.300   19   h thickness of beam [mic]
     4.700   20   hb V-notch depth [mic]
    15.000   21   ltx length of torquer [mic]
    15.000   22   lsx length of sensor  [mic]
    12.000   23   d0s distance from sensor elec. to plating [mic]
    12.000   24   d0t distance from torquer elec. to plateing [m]
 1.0E-C012   25   cfb feedback capacitance [f]
     5.000   26   vexc excitation voltage on sensor [v]
    0.0100   27   da crack length [mic]
    0.0500   28   rtip tip radius of V-notch [mic]
   15.0000   29   wservo servo bandwidth [hz]
   78.0000   30   v0 applied torquer voltage [v]
   78.0000   31   h0p plate airgap [mic]
    1.0000   32   vbias sensor bias voltage [v]
         3   33   ndof # DOF of analysis (2 or 3)
         0   34   failure simulation (1 = yes, 0 = no)
         0   35   fatigue simulation (1 = yes, 0 = no)
    0.0004   36   vsi02 crack velocity [m/s]
   40.0000   37   nsi02 fatigue constant

New file created on:
Date of run: Friday 2/14/1992          Time of run: 2:7:15
------------------------------------------------------------------
```

G.4 110291.inp

```
    13.000      1     lab length of segment ab [mic]
    60.000      2     lbc length of segment bc [mic]
    15.000      3     x1 edge clearances [mic]
    40.000      4     x2 distance between torquer and sensor plates [mic]
   100.000      5     x3 distance between sensor plate and plate 2 [mic]
   160.000      6     lpy width of plate [mic]
     0.000      7     lh1x length of hole at torquer [mic]
    30.000      8     lh1y width of at torquer hole [mic]
     0.000      9     lh3y length of hole at sensor [mic]
    45.000     10    lbh3y loc of cg of hole 3 [mic]
    27.000     11    lbh4y loc of cg of hole 4 [mic]
     9.000     12    lbh5y loc of cg of hole 5 [mic]
    70.000     13    lp2x length of tip mass [mic]
   130.000     14    lp2y width of tip mass [mic]
    13.000     15    lp2z thickness of tip mass [mic]
     0.000     16    lp1z thickness of plate mass [mic]
    20.000     17    b0 width of beam at point a (fixed end) [mic]
    17.000     18    bb width at V-notch [mic]
     8.300     19    h thickness of beam [mic]
     4.200     20    hb V-notch depth [mic]
    15.000     21    ltx length of torquer [mic]
    15.000     22    lsx length of sensor  [mic]
    15.000     23    d0s distance from sensor elec. to plating [mic]
    17.000     24    d0t distance from torquer elec. to plateing [m]
 1.0E-0012     25    cfb feedback capacitance [f]
     5.000     26    vexc excitation voltage on sensor [v]
    0.0100     27    da crack length [mic]
    0.0500     28    rtip tip radius of V-notch [mic]
   15.0000     29    wservo servo bandwidth [hz]
  156.0000     30    v0 applied torquer voltage [v]
   82.0000     31    h0p plate airgap [mic]
    1.0000     32    vbias sensor bias voltage [v]
         3     33    ndof # DOF of analysis (2 or 3)
         0     34    failure simulation (1 = yes, 0 = no)
         0     35    fatigue simulation (1 = yes, 0 = no)
    0.0004     36    vsi02 crack velocity [m/s]
   40.0000     37    nsi02 fatigue constant

New file created on:
Date of run: Friday 2/14/1992         Time of run: 2:18:52
-----------------------------------------------------------------
```

G.5 011092.inp

```
    13.000     1      lab length of segment ab [mic]
    60.000     2      lbc length of segment bc [mic]
    15.000     3      x1 edge clearances [mic]
    40.000     4      x2 distance between torquer and sensor plates [mic]
   100.000     5      x3 distance between sensor plate and plate 2 [mic]
   162.000     6      lpy width of plate [mic]
     0.000     7      lh1x length of hole at torquer [mic]
    30.000     8      lh1y width of at torquer hole [mic]
     0.000     9      lh3y length of hole at sensor [mic]
    45.000    10     lbh3y loc of cg of hole 3 [mic]
    27.000    11     lbh4y loc of cg of hole 4 [mic]
     9.000    12     lbh5y loc of cg of hole 5 [mic]
    70.000    13     lp2x length of tip mass [mic]
   130.000    14     lp2y width of tip mass [mic]
    16.000    15     lp2z thickness of tip mass [mic]
     0.000    16     lp1z thickness of plate mass [mic]
    20.000    17     b0 width of beam at point a (fixed end) [mic]
    17.000    18     bb width at V-notch [mic]
     8.300    19     h thickness of beam [mic]
     4.700    20     hb V-notch depth [mic]
    15.000    21     ltx length of torquer [mic]
    15.000    22     lsx length of sensor  [mic]
    12.500    23     d0s distance from sensor elec. to plating [mic]
    12.500    24     d0t distance from torquer elec. to plateing [m]
 1.0E-0012    25     cfb feedback capacitance [f]
     5.000    26     vexc excitation voltage on sensor [v]
    0.0100    27     da crack length [mic]
    0.0500    28     rtip tip radius of V-notch [mic]
   15.0000    29     wservo servo bandwidth [hz]
   87.0000    30     v0 applied torquer voltage [v]
   75.0000    31     h0p plate airgap [mic]
    1.0000    32     vbias sensor bias voltage [v]
         3    33     ndof # DOF of analysis (2 or 3)
         0    34     failure simulation (1 = yes, 0 = no)
         0    35     fatigue simulation (1 = yes, 0 = no)
    0.0004    36     vsi02 crack velocity [m/s]
   40.0000    37     nsi02 fatigue constant

New file created on:
Date of run: Friday 2/14/1992         Time of run: 2:43:25
-----------------------------------------------------------------
```

G.6 020392.inp

```
     3.000     1     lab length of segment ab [mic]
    62.000     2     lbc length of segment bc [mic]
    15.000     3     x1 edge clearances [mic]
    40.000     4     x2 distance between torquer and sensor plates [mic]
   100.000     5     x3 distance between sensor plate and plate 2 [mic]
   162.000     6     lpy width of plate [mic]
     0.000     7     lh1x length of hole at torquer [mic]
    30.000     8     lh1y width of at torquer hole [mic]
     0.000     9     lh3y length of hole at sensor [mic]
    45.000    10    lbh3y loc of cg of hole 3 [mic]
    27.000    11    lbh4y loc of cg of hole 4 [mic]
     9.000    12    lbh5y loc of cg of hole 5 [mic]
    70.000    13    lp2x length of tip mass [mic]
   130.000    14    lp2y width of tip mass [mic]
    18.000    15    lp2z thickness of tip mass [mic]
     0.000    16    lp1z thickness of plate mass [mic]
    19.000    17    b0 width of beam at point a (fixed end) [mic]
    17.000    18    bb width at V-notch [mic]
     8.100    19    h thickness of beam [mic]
     4.300    20    hb V-notch depth [mic]
    15.000    21    ltx length of torquer [mic]
    15.000    22    lsx length of sensor  [mic]
    13.000    23    d0s distance from sensor elec. to plating [mic]
    12.500    24    d0t distance from torquer elec. to plateing [m]
 1.0E-0012    25    cfb feedback capacitance [f]
     5.000    26    vexc excitation voltage on sensor [v]
    0.0100    27    da crack length [mic]
    0.0500    28    rtip tip radius of V-notch [mic]
   15.0000    29    wservo servo bandwidth [hz]
  100.0000    30    v0 applied torquer voltage [v]
   88.0000    31    h0p plate airgap [mic]
    1.0000    32    vbias sensor bias voltage [v]
         3    33    ndof # DOF of analysis (2 or 3)
         0    34    failure simulation (1 = yes, 0 = no)
         0    35    fatigue simulation (1 = yes, 0 = no)
    0.0004    36    vsi02 crack velocity [m/s]
   40.0000    37    nsi02 fatigue constant

New file created on:
Date of run: Thursday 2/13/1992         Time of run: 20:5:42
-----------------------------------------------------------------
```

G.7 FTEST2.PAS Code

```
{$M 65520,0,655360}
{$N+} {compiler directive for 8087 mode}
program ftest2;
uses
  crt,
  dos,
  init2,
  matrix2,
  dyn2,
  frac2,
  elec2,
  menu2,
  out2;

var
  hits_torquer,
  hits_sensor:boolean;

label start;

begin {Ftest.pas}
  while rerun do
  begin {while rerun do}
    start: dynamics(vc,thetac);
           hits_sensor:=false;
           dimension_check(vc,thetac,v3static,hits_sensor,hits_torquer);
           if hits_sensor then
           begin
             clrscr;
             writeln('WARNING: PLATE HITS SENSOR');
             writeln('d0smin = ',d0smin/clgth:10:3);
           end;
           if hits_torquer then
           begin
             clrscr;
             writeln('WARNING: PLATE HITS TORQUER');
             writeln('d0tmin = ',d0tmin/clgth:10:3);
           end;
           if hits_torquer or hits_sensor then
           begin
             readln;
             clrscr;
             reinitialize;
             goto start;
           end;
           sensoroutput(vc,thetac,eoutu,eoutd);
           control(ndof,repole);
           fracture_analysis;
           run_menu;
  end; {while rerun do}
end. {Ftest.pas}
```

```
{$N+}{compiler directive for 8087 double precision mode}
unit init2;
interface
uses
  crt,
  dos;

const
  rhoair=1.2;                       {density of air [kg/m^3]}
  rhosi=2.3e3;                      {density of silicon [kg/m^3]}
  rhoau=1.93e4;                     {density of gold [kg/m^3]}
  rhoni=8.602e3;                    {density of nickel [kg/m^3]}
  eni=2.2063e11;                    {Modulus of elasticity of ni [n/m^2]}
  esi=1.59e11;                      {Modulus of elasticity of si [n/m^2]}
                                    {determined by UTRC nanoindenting}
  cs=1e20;                          {surface concen. of boron [atoms/cm^3]}
  csi=5e22;                         {concen. of Si atom sites [atoms/cm^3]}
  rb=0.98;                          {atomic radius of boron [angs]}
  rsi=1.17;                         {atomic radius of silicon [angs]}
  nusi=0.33;                        {Poisson's ratio of silicon}
  eau=8.0e10;                       {Modulus of elasticity of Au [n/m^2]}
  dEdt=1.1e8;                       {temp sens of Youngs mod. n/(m^2*C)}
  dGdt=1.5e7;                       {temp sens of torsional mod. n/(m^2*C)}
  alphasi=2.5e-6;                   {thermal coeff of exp. of silicon /deg C}
  alphaau=14.2e-6;                  {thermal coeff of exp. of gold /deg c}
  gsi=5.1e10;                       {Torsional modulus of si [n/m^2]}
  clgth=1.0e-6;                     {conversion factor [m/mic]}
  e0=8.85e-12;                      {permitivity of a vacuum [coul^2/n*m^2]}
  ka=1.0;                           {dielectric of air}
  k1c111=1e6;                       {crit. stress factor <111> [n/m^(3/2)]}
  k1c110=9.0e5;                     {crit. stress factor <110> [n/m^(3/2)]}
  k1c100=9.5e5;                     {crit. stress factor <100> [n/m^(3/2)]}
  pctk1cf=1.0;                      {% final stress intensity factor}
  sigmaf=esi*0.0206;                {fracture stress of si whiskers reported
                                     by Eisner to be 2.06% of E}
  gamma_h20=73e-3;                  {surface tension H20 [n/m]}
  gamma_meth=22.1e-3;               {surface tension methanol [n/m]}
  muair=1.81e-5;                    {viscosity of air [n*sec/m^2]}
  agrav=9.8;                        {acceleration due to gravity [m/s^2]}
  nimp=37;                          {number of inputs on data file}
  nsys=3;                           {size of dynamic system matices}
  nsub=2;                           {size of substructure matrices}
  qvac=1e4;                         {Q in vacuum; determine from spectrum anal}
  Pamb=1.013565e5;                  {atmospheric pressure [n/m^2]}
  Tsin=360;                         {process sinter temp [deg C]}
  Twb=300;                          {TC wire bond temp [deg C]}
  vbdsi02=6e8;                      {break down voltage of si02 [v/m]}
  tsi02=0.5e-6;                     {si02 thick. under bridge electrodes [m]}
```

```
type
  piece=(p,p1,p2,h1s,h2s,h3s,h4s,h5s,h6s,h7s,h8s,
                h1a,h2a,h3a,h4a,h5a,h6a,h7a,h8a); {plate,plating1,plating2,
                                                   holes 1 through 5}
                                    {dimensions of above and loc. wrt pt.c}
  dim=(x,y,z);
  sysmat=array[1..nsys,1..nsys] of double;
  sysvec=array[1..nsys] of double;

var                                 {nanoindenter clearance [m]}
  Icl,                              {nanoindenter indentation load [n]}
  Pind,                             {edge clearance [m]}
  x1,                               {spacing bet. torquer and sensor [m]}
  x2,                               {spacing bet. sensor and bar mass [m]}
  x3,                               {length of segment ab [m]}
  lab,                              {length of segment bc [m]}
  lbc,                              {length of segment bg [m]}
  lbg,                              {length from point a to point c [m]}
  lac,                              {length from point b to point d [m]}
  lbd,                              {length from point a to point g (center of
  lag,                              {length of plate [m]}
  lpx,                              {width of plate [m]}
  lpy,                              {length of torquer hole [m]}
  lh1x,                             {width of torquer hole [m]}
  lh1y,                             {width of sensor hole 3 [m]}
  lh3y,                             {c.g. of hole 3 in y direction [m]}
  lbh3y,                            { "   "  "   4  " "    "        " }
  lbh4y,                            { "   "  "   5  " "    "        " }
  lbh5y,                            {thickness of plate mass [m]}
  lp1z,                             {length of tip mass [m]}
  lp2x,                             {width of tip mass [m]}
  lp2y,                             {thickness of tip mass [m]}
  lp2z,                             {VCO gain [rad/(s*v)]}
  kvco,                             {type 1 filter gain [s]}
  kf1,                              {plant gain at d.c. [s]}
  kp,                               {control system gain [1/s]}
  gp,                               {real pole location [rad/s]}
  repole,                           {imaginary pole location [rad/s]}
  impole,                           {real pole location in a vacuum [rad/s]}
  repolevac,                        {imag pole lacation in a vacuum [rad/s]}
  impolevac,                        {type II phase detector gain [volts/rad]}
  kd,                               {torquer constant [n/volt]}
  kt,                               {servo bandwidth [rad/s]}
  wservo,                           {velocity error [rad]}
  verror,                           {max controllable velocity error [rad]}
  verrormax,                        {velocity error at failure [rad]}
  verrorf,                          {type1 filter zero, rad/s}
  a,
 b,                          {typel filter pole, rad/s}
 d0t,                        {dist. from torquer elec. to si plate [m]}
 d0s,                        {dist. from sensor elec. to si plate [m]}
 d0smin,                     {minimum sensor airgap [m]}
 d0tmin,                     {minimum torquer airgap [m]}
 dcurls,                     {curl displacement at sensor [m]}
 dcurlt,                     {curl displacement at torquer [m]}
 ht,                         {torquer airgap corrected for v3static, [m]}
 hs,                         {sensor airgap corrected for v3static, [m]}
 h0t,                        {dist. between plating and torquer elec. [m]}
 h0s,                        {dist. between plating and sensor elec. [m]}
 h0p,                        {dist. between plate and bottom of well [m]}
 b0,                         {width of beam @ pt. a (fixed end) [m]}
 bb,                         {thickness of beam at v-notch [m]}
 h,                          {thickness of beam [m]}
 hb,                         {th. of beam @ pt.b (str. concen.) [m]}
 wv,                         {width of V-notch [m]}
 ltx,                        {length of torquer [m]}
 lty,                        {width of torquer  [m]}
 lbtx,                       {length to torquer c.g. [m]}
 lsx,                        {length of sensor  [m]}
 lsy,                        {width of sensor  [m]}
 lbsx,                       {length to sensor c.g. [m]}
 Iab,                        {area moment of inertia of beam at v-notch
                              [m^4]}
 sr,                         {stiffness ratio of plate to beam}
 c_kies,                     {crack compliance Kies 1/(n*m)}
 c_mura,                     {crack compliance Murakami 1/(n*m)}
 c_paris,                    {crack compliance Paris 1/(n*m)}
 c_ellip,                    {elliptical crack compliance 1/(n*m)}
 c_dc,                       {deep crack compliance 1/(n*m)}
 krbl,                       {rotational stiffness of v-notch w.r.t.
                              point c [n*m/rad]}
 da,                         {incremental crack length [m]}
 ktor,                       {torsional stiffness of beam, [n*m/rad]}
 krp,                        {composite rot. stiff. of plate [n*m/rad]}
 dklda,                      {sensitivity of stress intensity factor to
                              crack length [n-m^5/2]}
 dkldaf,                     {rate of change of kl at failure [n/m^5/2]}
 dkldactl,                   {rate of chg of kl at bandwidth [n/m^5/2]}
 rtip,                       {tip radius of V-notch [m]}
 kadc,
 kakies,
 kaellip,
 kamur,
 klf,                        {final stress intensity 95% klclll [n/m^3/2]}
 klcc,                       {klclll corrected for crack dynamcis
[n/m^3/2]}
```

```
  kli,                          {initial stress intensity factor [n/m^(3/2)]}
  kl,                           {stress intensity at da [n/m^(3/2)]}
  pctklc,                       {% instantaneous stress intensity factor}
  pctklci,                      {% initial stress intensity factor}
  pctklctl,                     {% klclll at controller bandwidth}
  vsi02,                        {nominal crack velocity in si02 [m/s]}
  nsi02,                        {power law relation for si02}
                                {constants extracted from p.29 Table 1
                                 Structural Reliability of Brittle
                                 Materials, Cook and Lawn}
  dwctl,                        {freq. shift at bandwidth [rad]}
  dactl,                        {crack length at bandwidth [m]}
  vctl,                         {displ at c at bandwidth [m]}
  thetactl,                     {angle at c at bandwidth [rad]}
  dwdtctl,                      {rate of change of freq at bandwidth
[rad/s^2]}
  dwdtmax,                      {max rate of change of freq based on ver-
rormax [rad/s^2]}
  dwf,                          {freq. shift to klf}
  daf,                          {increase in crack length to failure}
  tf,                           {dynamic time to failure based on si02 [hr]}
  ta,                           {incremental time for crack to propagate
                                 amount da [hr]}
  tctl,                         {time until bandwidth exceeded}
  vcf,                          {displacement at c at failure [m]}
  thetacf,                      {angle at c at failure [rad]}
  dwdtf,                        {rate of change of freq at failure [rad/s^2]}
  pctsf,                        {% fracture stress at V-notch}
  mt,                           {total mass of end plate [kg]}
  mb,                           {mass/unit length of beam [kg/m]}
  dw,                           {freq  shift due to crack da [rad/s]}
  dwdt,                         {rate of nat freq due to crack da [rad/s^2]}
  wxx,                          {torsional frequency of beam about x [rad/s]}
  wzz,                          {1st mode bending frequency about z axis
 [rad/s]}
  wp,                           {freq. of mode 1 of end plate [rad/s]}
  kbr,                          {brigde stiffness [n/m]}
  wbr,                          {freq. of mode 1 of torquer electrode
 [rad/s]}
  vbr,                          {static displ. of bridge electrode, [m]}
  bucr,                         {buckling ration of bridge electrode}
  Tmax,                         {max temp w/o bridge buckling [degC]}
  thetacstatic,                 {static rot. at pt c due to v0 [rad]}
  vcstatic,                     {static displ. at pt. c due to v0 [m]}
  v3static,                     {static displ. at pt. c due to nonlin.
 spring, [m]}
  v3,
  vc,                           {displacement at point c [m]}
  thetac,                            {angle at point c [m]}
  vcmax,                             {max deflection at pt c [m]}
  thetacmax,                         {max deflection at pt c [m]}
  sigmab,                            {far field stress at point b [n/m^2]}
  momb,                              {moment applied at point b [n*m]}
  mombf,                             {failure momonet at b [n*m]}
  mombst,                            {moment at point b due to surf. tension
[n*m]}
  f0,                                {magnitude of torquer load [n]}
  v0,                                {applied torquer voltage [v]}
  vbd,                               {si02 break-down voltage [v]}
  v0c,                               {fracture voltage [v]}
  v0vac,                             {applied torquer voltage in vacuum [v]}
  v0f,                               {failure voltage [v]}
  vbias,                             {sensor bias voltage [v]}
  vcfail,                            {failure displacement at point c [m]}
  thetacfail,                        {failure angle at point c [rad]}
  vsnap,                             {snap voltage, [v]}
  delcu,                             {upward delta capacitance [f]}
  delcd,                             {downward delta capacitance [f]}
  c0,                                {nominal capacitance zero displacement [f]}
  vexc,                              {excitation voltage on sensor [v]}
  eoutu,                             {sensor output upward travel [v]}
  eoutd,                             {sensor output downward travel [v]}
  cfb:double;                        {feedback capacitor [f]}
  lb,                                {dim. array of pieces wrt pt.c [m]}
  l:array[piece,dim] of double;      {dim. array of pieces [m]}
  Iac,                               {area moment of beam about x,y,x axis [m^4]}
  krc,                               {rot. stiff. of beam at c about x,y,z [N*m]}
  Ip:array[dim,dim] of double;       {inertia matrix of end plate [kg*m^2]}
  E,                                 {Young's modulus [n/m^2]}
  rho,                               {density array [kg/m^3]}
  m:array[piece] of double;          {mass array of end plate [kg]}
  klc,                               {linear stiffness of beam in x,y,z at c
[N/m]}
  lcg:array[dim] of double;          {center of gravity of end plate [m]}
  zeta,                              {lever arm matrix of cg}
  eta,                               {  "    "     "    " torquer}
  xi,                                {lever arm matrix of sensor}
  lambda,                            {lever arm matrix of plate}
  ma,                                {mass matrix}
  mac,                               {corrected mass matrix}
  k,                                 {stiffness matrix}
  kc,                                {corrected stiffness matrix}
  phi,                               {modal matrix of kc and mac}
  phinorm,                           {normalized modal matrix of kc and mac}
  phivac,                            {modal matrix of ma and k}
  phinormvac,                        {normalized vacuum modal matrix}
  c:sysmat;                       {damping matrix}
  mhat,                           {modal mass matrix of mac}
  khat,                           {modal stiff matrix of kc}
  chat,                           {modal damping matrix}
  phat,                           {modal load vector of p0}
  omega,                          {frequency vector of mac and kc [rad/s]}
  omegavac,                       {frequency vector of ma and k [rad/s]}
  etavec,                         {vector of lever arms for torquer}
  xivec,                          {vector of lever arms for sensor}
  P0:sysvec;                      {loading vector}
  filevar:array[1..nimp] of double; {input variable array}
  fati_sim,                       {flag to run fatigue simulation}
  fail_sim,                       {flag to run a failure simulation}
  rerun:boolean;                  {flag to rerun program}
  option:char;                    {output options}
  frac,                           {input for a failure simulation}
  fati,                           {input for a fatigue simulation}
  ndof:integer;                   {number of degrees of freedom}

  procedure reinitialize;
  procedure initialize;
  procedure indata;
  function thetacurl(lgth,gap:double):double;
  function deld0(lgth,gap:double):double;
  procedure dimension_check(v,theta,v3static:double;
                          var hits_selec,hits_telec:boolean);

implementation
  function tan(deg:double):double;
  begin
    deg:=deg*pi/180;
    tan:=sin(deg)/cos(deg);
  end;

  function thetacurl(lgth,gap:double):double;
  begin
    thetacurl:=2*(3.0e-4)*lgth/gap;             {rad}
  end;

  function deld0(lgth,gap:double):double;
  begin
    deld0:=3.0e-4*sqr(lgth)/gap;         {9.597e-5 is theorectical value}
  end;

  procedure dimension_check(v,theta,v3static:double;
                          var hits_selec,hits_telec:boolean);
  var
    dcurlt,
  dcurls:double;
begin
  {compute min allowable airgap based on max. deflection}
  dcurlt:=deld0(lac+lbtx+ltx/2,h);
  dcurls:=deld0(lac+lbsx+lsx/2,h);
  d0tmin:=v+vcstatic+(theta+thetacstatic)*(x1+ltx)+dcurlt;
  d0smin:=v+vcstatic+(theta+thetacstatic)*(x1+x2+ltx+lsx)+dcurls;
  if d0smin>=d0s then hits_selec:=true
  else hits_selec:=false;
  if d0tmin>=d0t then hits_telec:=true
  else hits_telec:=false;
end;

procedure indata;
const
  current_dir='c:\jcon\progms\fatigue\ftest_2';
var
  filecheck:pathstr;
  filename:string;
  infile:text;
  n:integer;

begin {indata}
  clrscr;
  writeln('Specify input filename.');
  repeat
    readln(filename);
  until filename <> '';
  filecheck:=fsearch(filename,current_dir);
  if filecheck='' then
  begin
    Writeln(filename,' not found in this directory.');
    halt;
  end
  else
  begin
    assign(infile,filename);
    reset(infile);
    n:=0;
    repeat
      begin
        n:=n+1;
        readln(infile,filevar[n]);
      end;
    until n=nimp;
    close(infile);
  end;
end; {indata}
```

```
procedure initialize;
begin {initialize}
  {preliminary initialization}
  lab:=filevar[1]*clgth;
  lbc:=filevar[2]*clgth;
  x1:=filevar[3]*clgth;
  x2:=filevar[4]*clgth;
  x3:=filevar[5]*clgth;
  lpy:=filevar[6]*clgth;
  lh1x:=filevar[7]*clgth;
  lh1y:=filevar[8]*clgth;
  lh3y:=filevar[9]*clgth;
  lbh3y:=filevar[10]*clgth;
  lbh4y:=filevar[11]*clgth;
  lbh5y:=filevar[12]*clgth;
  lp2x:=filevar[13]*clgth;
  lp2y:=filevar[14]*clgth;
  lp2z:=filevar[15]*clgth;
  lp1z:=filevar[16]*clgth;
  b0:=filevar[17]*clgth;
  bb:=filevar[18]*clgth;
  h:=filevar[19]*clgth;
  hb:=filevar[20]*clgth;
  ltx:=filevar[21]*clgth;
  lsx:=filevar[22]*clgth;
  d0s:=filevar[23]*clgth;
  d0t:=filevar[24]*clgth;
  cfb:=filevar[25];
  vexc:=filevar[26];
  da:=filevar[27]*clgth;
  rtip:=filevar[28]*clgth;
  wservo:=filevar[29]*2*pi;
  v0:=filevar[30];
  h0p:=filevar[31]*clgth;
  vbias:=filevar[32];
  ndof:=trunc(filevar[33]);
  frac:=trunc(filevar[34]);
  fati:=trunc(filevar[35]);
  vsi02:=filevar[36];
  nsi02:=filevar[37];

  if (ndof <> 2) and (ndof <> 3) then
  begin
    writeln('ERROR IN ASSIGNMENT OF NDOF: NDOF RESET TO 3');
    ndof:=trunc(3.0);
    filevar[31]:=ndof;
    readln;
```

```
end;

if (frac <> 0) and (frac <> 1) then
begin
  writeln('ERROR IN ASSIGNMENT OF FRAC: FRAC RESET TO 0');
  frac:=0;
  filevar[32]:=0;
end;

if (ndof = 3) and (frac = 1) then fail_sim:=true
else
begin
  fail_sim:=false;
  frac:=0;
end;

if (ndof = 3) and (fati = 1) then fati_sim:=true
else
begin
  fati_sim:=false;
  fati:=0;
end;

if ndof = 3 then
begin
  {check precrack dimensions}
  if hb <= da then
  begin
    writeln('WARNING: hb <= da, da reset to da:=0.1*hb');
    da:=0.1*hb;
    readln;
  end;
  if da<=0 then
  begin
    writeln('WARNING: da <= 0, da reset to da:=0.1*hb');
    da:=0.1*hb;
    readln;
  end;
end;

{compute dielectric breakdown voltage}
vbd:=vbdsi02*tsi02;
if v0 >= vbd then
begin
  writeln('WARNING: DIELECTRIC BREAKDOWN AT BRIDGE ELECTRODE');
  readln;
  clrscr;
end;
```

```
{compute nanoindenter clearance}
Icl:=(lbc+x1-lp2z*tan(35{30}))*tan(90-65.3)-(d0t+lp2z);
if Icl <= 0 then
begin
  writeln('WARNING: NANOINDENTER DOES NOT CLEAR BRIDGE ELECTRODE');
  readln;
  clrscr;
end;

{stress intensity initialization [n/m^(3/2)]}
k1cc:=k1c111/(1/pi+1/2);
k1f:=pctk1cf*k1cc;

{torquer initialization}
lty:=lpy;
lbtx:=ltx/2+x1;

{sensor initialization}
lsy:=lpy;
lbsx:=x1+x2+ltx+lsx/2;

{beam dimensions}
lbd:=lbc+lbtx;
lac:=lab+lbc;

{airgap correction due to curling}
h0t:=d0t-deld0(lac+lbtx,h);
h0s:=d0s-deld0(lac+lbsx,h);
ht:=h0t;
hs:=h0s;

{silicon plate initialization}
l[p,x]:=x1+x2+x3+ltx+lsx+lp2x;
l[p,y]:=lpy;
l[p,z]:=h;
lb[p,x]:=l[p,x]/2;
lb[p,y]:=0.0;
lb[p,z]:=0.0;

{plate 1 initialization}
l[p1,x]:=l[p,x];
l[p1,y]:=l[p,y];
l[p1,z]:=lp1z;
lb[p1,x]:=l[p,x]/2;
lb[p1,y]:=0.0;
lb[p1,z]:=(l[p1,z]+l[p,z])/2;
```

```
{plate 2 initialization}
l[p2,x]:=lp2x;
l[p2,y]:=lp2y;
l[p2,z]:=lp2z;
lb[p2,x]:=x1+x2+x3+ltx+lsx+l[p2,x]/2;
lb[p2,y]:=0.0;
lb[p2,z]:=(l[p2,z]+l[p,z])/2;

{hole 1 initialization}
l[h1s,x]:=lh1x;
l[h1s,y]:=lh1y;
l[h1s,z]:=l[p,z];
l[h1a,x]:=lh1x;
l[h1a,y]:=lh1y;
l[h1a,z]:=l[p1,z];
lb[h1a,x]:=ltx/2+x1;
lb[h1a,y]:=(l[p,y]-2*l[h1a,y])/6+l[h1a,y]/2;
lb[h1a,z]:=(l[h1s,z]+l[h1a,z])/2;
lb[h1s,x]:=ltx/2+x1;
lb[h1s,y]:=(l[p,y]-2*l[h1s,y])/6+l[h1s,y]/2;
lb[h1s,z]:=0;

{hole 2 initialization}
l[h2s,x]:=lh1x;
l[h2s,y]:=lh1y;
l[h2s,z]:=l[p,z];
l[h2a,x]:=lh1x;
l[h2a,y]:=lh1y;
l[h2a,z]:=l[p1,z];
lb[h2a,x]:=lb[h1a,x];
lb[h2a,y]:=lb[h1a,y];
lb[h2a,z]:=(l[h2s,z]+l[h2a,z])/2;
lb[h2s,x]:=lb[h1s,x];
lb[h2s,y]:=-lb[h1s,y];
lb[h2s,z]:=0;

{hole 3 initialization}
l[h3s,x]:=lsx;
l[h3s,y]:=lh3y;
l[h3s,z]:=l[p,z];
l[h3a,x]:=lsx;
l[h3a,y]:=lh3y;
l[h3a,z]:=l[p1,z];
lb[h3s,x]:=lbsx;
lb[h3s,y]:=lbh3y;
lb[h3s,z]:=0;
lb[h3a,x]:=lbsx;
lb[h3a,y]:=lbh3y;
```

```
lb[h3a,z]:=(l[h3s,z]+l[h3a,z])/2;

{hole 4 initialization}
l[h4s,x]:=l[h3s,x];
l[h4s,y]:=l[h3s,y];
l[h4s,z]:=l[h3s,z];
l[h4a,x]:=l[h3a,x];
l[h4a,y]:=l[h3a,y];
l[h4a,z]:=l[h3a,z];
lb[h4s,x]:=lb[h3s,x];
lb[h4s,y]:=lbh4y;
lb[h4s,z]:=0;
lb[h4a,x]:=lb[h3a,x];
lb[h4a,y]:=lbh4y;
lb[h4a,z]:=(l[h4a,z]+l[h4s,z])/2;

{hole 5 initialization}
l[h5s,x]:=l[h3s,x];
l[h5s,y]:=l[h3s,y];
l[h5s,z]:=l[h3s,z];
l[h5a,x]:=l[h3a,x];
l[h5a,y]:=l[h3a,y];
l[h5a,z]:=l[h3a,z];
lb[h5s,x]:=lb[h3s,x];
lb[h5s,y]:=lbh5y;
lb[h5s,z]:=0;
lb[h5a,x]:=lb[h3a,x];
lb[h5a,y]:=lbh5y;
lb[h5a,z]:=(l[h5s,z]+l[h5a,z])/2;

{hole 6 initialization}
l[h6s,x]:=l[h3s,x];
l[h6s,y]:=l[h3s,y];
l[h6s,z]:=l[h3s,z];
l[h6a,x]:=l[h3a,x];
l[h6a,y]:=l[h3a,y];
l[h6a,z]:=l[h3a,z];
lb[h6s,x]:=lb[h3s,x];
lb[h6s,y]:=-lb[h5s,y];
lb[h6s,z]:=0;
lb[h6a,x]:=lb[h3a,x];
lb[h6a,y]:=-lb[h5a,y];
lb[h6a,z]:=(l[h6a,z]+l[h6s,z])/2;

{hole 7 initialization}
l[h7s,x]:=l[h3s,x];
l[h7s,y]:=l[h3s,y];
l[h7s,z]:=l[h3s,z];
```

```
l[h7a,x]:=l[h3a,x];
l[h7a,y]:=l[h3a,y];
l[h7a,z]:=l[h3a,z];
lb[h7s,x]:=lb[h3s,x];
lb[h7s,y]:=-lb[h4s,y];
lb[h7s,z]:=0;
lb[h7a,x]:=lb[h3a,x];
lb[h7a,y]:=-lb[h4a,y];
lb[h7a,z]:=(l[h7s,z]+l[h7a,z])/2;

{hole 8 initialization}
l[h8s,x]:=l[h3s,x];
l[h8s,y]:=l[h3s,y];
l[h8s,z]:=l[h3s,z];
l[h8a,x]:=l[h3a,x];
l[h8a,y]:=l[h3a,y];
l[h8a,z]:=l[h3a,z];
lb[h8s,x]:=lb[h3s,x];
lb[h8s,y]:=-lb[h3s,y];
lb[h8s,z]:=0;
lb[h8a,x]:=lb[h3a,x];
lb[h8a,y]:=-lb[h3a,y];
lb[h8a,z]:=(l[h8a,z]+l[h8s,z])/2;

{density initialization}
rho[p]:=rhosi;
rho[p1]:=rhoau;
rho[p2]:=rhoau;
rho[h1s]:=rho[p];
rho[h1a]:=rho[p1];
rho[h2s]:=rho[p];
rho[h2a]:=rho[p1];
rho[h3s]:=rho[p];
rho[h3a]:=rho[p1];
rho[h4s]:=rho[p];
rho[h4a]:=rho[p1];
rho[h5s]:=rho[p];
rho[h5a]:=rho[p1];
rho[h6s]:=rho[p];
rho[h6a]:=rho[p1];
rho[h7s]:=rho[p];
rho[h7a]:=rho[p1];
rho[h8s]:=rho[p];
rho[h8a]:=rho[p1];

{Young's Modulus initialization}
E[p]:=esi;
E[p1]:=eau;
```

```
    E[p2]:=eau;
  end; {initialize}

  procedure printvar1;
  begin {printvar1}
    writeln(filevar[1]:10:3,'      1      lab length of segment ab [mic]');
    writeln(filevar[2]:10:3,'      2      lbc length of segment bc [mic]');
    writeln(filevar[3]:10:3,'      3      x1 edge clearances [mic]');
    writeln(filevar[4]:10:3,'      4      x2 distance between torquer and sensor
plates [mic]');
    writeln(filevar[5]:10:3,'      5      x3 distance between sensor plate and
plate 2 [mic]');
    writeln(filevar[6]:10:3,'      6      lpy width of plate [mic]');
    writeln(filevar[7]:10:3,'      7      lh1x length of hole at torquer
[mic]');
    writeln(filevar[8]:10:3,'      8      lh1y width of at torquer hole [mic]');
    writeln(filevar[9]:10:3,'      9      lh3y length of hole at sensor [mic]');
    writeln(filevar[10]:10:3,'     10     lbh3y loc of cg of hole 3 [mic]');
    writeln(filevar[11]:10:3,'     11     lbh4y loc of cg of hole 4 [mic]');
    writeln(filevar[12]:10:3,'     12     lbh5y loc of cg of hole 5 [mic]');
    writeln(filevar[13]:10:3,'     13     lp2x length of tip mass [mic]');
    writeln(filevar[14]:10:3,'     14     lp2y width of tip mass [mic]');
    writeln(filevar[15]:10:3,'     15     lp2z thickness of tip mass [mic]');
    writeln(filevar[16]:10:3,'     16     lp1z thickness of plate mass [mic]');
    writeln(filevar[17]:10:3,'     17     b0 width of beam at point a (fixed
end) [mic]');
    writeln(filevar[18]:10:3,'     18     bb width at V-notch [mic]');
  end;{printvar1}

  procedure printvar2;
  begin
    writeln(filevar[19]:10:3,'     19     h thickness of beam [mic]');
    writeln(filevar[20]:10:3,'     20     hb V-notch depth [mic]');
    writeln(filevar[21]:10:3,'     21     ltx length of torquer [mic]');
    writeln(filevar[22]:10:3,'     22     lsx length of sensor  [mic]');
    writeln(filevar[23]:10:3,'     23     d0s distance from sensor elec. to
plating [mic]');
    writeln(filevar[24]:10:3,'     24     d0t distance from torquer elec. to
plating [mic]');
    writeln(filevar[25]:10,  '     25     cfb feedback capacitance [f]');
    writeln(filevar[26]:10:3,'     26     vexc excitation voltage on sensor
[v]');
    writeln(filevar[27]:10:4,'     27     da crack length [mic]');
    writeln(filevar[28]:10:4,'     28     rtip tip radius of V-notch [mic]');
    writeln(filevar[29]:10:4,'     29     wservo servo bandwidth [hz]');
    writeln(filevar[30]:10:4,'     30     v0 applied torquer voltage [v]');
    writeln(filevar[31]:10:4,'     31     h0p plate airgap [mic]');
    writeln(filevar[32]:10:4,'     32     vbias sensor bias voltage [v]');
    writeln(trunc(filevar[33]):10,'    33    ndof # DOF of analysis (2 or
3)');
    writeln(trunc(filevar[34]):10,'    34    failure simulation (1 = yes, 0 =
no)');
    writeln(trunc(filevar[35]):10,'    35    fatigue simulation (1 = yes, 0 =
no)');
    writeln(filevar[36]:10,'    36    vsi02 crack velocity [m/s]');
    writeln(filevar[37]:10:2,'    37    nsi02 fatigue constant');
  end; {printvar2}

  procedure reinitialize;
  var
    varno:string[2];
    varsel:0..nimp;
    code:integer;
    quitinitial:boolean;

    procedure printhead;
    begin
      writeln('  Value      Var No.        Description');
      writeln('--------------------------------------------------------------
');
    end;

  begin {reinitialize}
    quitinitial:=false;
    clrscr;
    repeat {until quitinitial=true}
      repeat {until varno=0}
        printhead;
        printvar1;
        writeln('Select Var No. or 0 to quit');
        repeat
          readln(varno);
          val(varno,varsel,code);
        until (varsel in [0..nimp]) and (code=0);
        if varsel in [1..nimp] then
        begin
          writeln('Enter new variable value.');
          readln(filevar[varsel]);
        end;
        clrscr;
      until varno='0';
      repeat
        printvar2;
        writeln('Selact Var No. or 0 to quit');
        repeat
          readln(varno);
        val(varno,varsel,code);
      until (varsel in [0..nimp]) and (code=0);
      if varsel in [1..nimp] then
      begin
        writeln('Enter new variable value.');
        readln(filevar[varsel]);
      end
      else quitinitial:=true;
      clrscr;
    until varno='0';
  until quitinitial;
initialize;
end; {reinitialize}

begin
  rerun:=true;
  indata;
  initialize;
end.
```

```
{$N+}{8087 compiler mode directive}
unit mains2;
interface
uses
  crt,
  dos,
  init2;

procedure massandinertia;
procedure stiffness;
function varstiff(da:double):double;
function power(num,n:double):double;

implementation
  function power(num,n:double):double;
  begin
    power:=exp(n*ln(num));
  end;

procedure massandinertia;
  procedure masscalc;
  var
    n:piece;
    cx,
    cy,
    cz:double;

    function mass(rho,lgth,width,thick:double):double;
    begin
      mass:=rho*lgth*width*thick;
    end;

  begin{masscalc}
    cx:=0;
    cy:=0;
    cz:=0;
    mt:=0;
    for n:=p to h8a do
    begin
      m[n]:=mass(rho[n],l[n,x],l[n,y],l[n,z]);
      if n in [h1s..h8a] then m[n]:=-1*m[n];
      mt:=mt+m[n];
      cx:=cx+m[n]*lb[n,x];
      cy:=cy+m[n]*lb[n,y];
      cz:=cz+m[n]*lb[n,z];
    end;
    lcg[x]:=cx/mt;
    lcg[y]:=cy/mt;
  lcg[z]:=cz/mt;
  lag:=lac+lcg[x];
  lbg:=lbc+lcg[x];
  mb:=mass(rhosi,lac,b0,h)/lac;
end;{masscalc}

procedure inertiacalc;
type
  inertia=array[piece,dim,dim] of double;
var
  Ic:inertia;
  alpha,beta:dim;
  n:piece;

  procedure Icomp(n:piece; mi,lx,ly,lz,a,b,c:double; var Ic:inertia);
  begin
    Ic[n,x,x]:=1/12*mi*(sqr(ly)+sqr(lz))+mi*(sqr(b)+sqr(c));
    Ic[n,x,y]:=-mi*a*b;
    Ic[n,x,z]:=-mi*a*c;
    Ic[n,y,x]:=Ic[n,x,y];
    Ic[n,y,y]:=1/12*mi*(sqr(lx)+sqr(lz))+mi*(sqr(a)+sqr(c));
    Ic[n,y,z]:=-mi*b*c;
    Ic[n,z,x]:=Ic[n,x,z];
    Ic[n,z,y]:=Ic[n,y,z];
    Ic[n,z,z]:=1/12*mi*(sqr(lx)+sqr(ly))+mi*(sqr(a)+sqr(b));
  end;

begin{inertiacalc}
  {initialize end plate inertia}
  for alpha:=x to z do
  for beta:=x to z do Ip[alpha,beta]:=0;

  {compute inertias of end plate components}
  for n:=p to h8a do
  begin
    Icomp(n,m[n],l[n,x],l[n,y],l[n,z],lcg[x]-lb[n,x],lcg[y]-lb[n,y],
                     lcg[z]-lb[n,z],Ic);
    for alpha:=x to z do
    for beta:=x to z do
    Ip[alpha,beta]:=Ip[alpha,beta]+Ic[n,alpha,beta];
  end;

  {correct masses and inertias}
  for n:=h1s to h8s do
  begin
    for alpha:=x to z do
    for beta:=x to z do
    Ic[p,alpha,beta]:=Ic[p,alpha,beta]+Ic[n,alpha,beta];
```

```
      m[p]:=m[p]+m[n];
    end;

    for n:=h1a to h8a do
    begin
      for alpha:=x to z do
      for beta:=x to z do
      Ic[p1,alpha,beta]:=Ic[p1,alpha,beta]+Ic[n,alpha,beta];
      m[p1]:=m[p1]+m[n];
    end;
  end; {inertiacalc}

begin{massandinertia}
  masscalc;
  inertiacalc;
end;{massandinertia}

procedure stiffness;
var
  Aac,
  ktor1,
  ktor2,
  Icorr:double;

  function areainertia(h:double):double;
  {areainertia computes area moment of inertia per unit width
   of a prismatic beam}
  begin
    areainertia:=(1/12*h*h*h);
  end;

  function tan(thetadeg:double):double;
  begin
    tan:=sin(thetadeg*pi/180)/cos(thetadeg*pi/180);
  end;

  function EIc(h1,h2,E1,E2:double):double;
  var
    I1,
    I2,
    c,
    c1,
    c2,
    zeta,
    con1,
    con2:double;
  begin
    I1:=areainertia(h1);
  I2:=areainertia(h2);
    c:=(sqr(h1)+2*E2/E1*h1*h2+E2/E1*sqr(h2))/
       (2*(h1+E2/E1*h2));
    c1:=c/h1;
    c2:=c/h2;
    zeta:=h1/h2;
    con1:=4-12*c1+12*sqr(c1);
    con2:=4+12*sqr(zeta)+12*zeta-
        24*c*h1/sqr(h2)-12*c/h2+12*sqr(c/h2);
    EIc:=E1*I1*con1+E2*I2*con2;
end;

procedure platestiff;
var
  l1,                          {length of spring 1}
  l2,                          {  "     "      "   2}
  l3,                          {  "     "      "   3}
  l4,                          {  "     "      "   4}
  l5,                          {  "     "      "   5}
  le1,                         {lever arm spring 1}
  le2,                         {  "     "      "   2}
  le3,                         {  "     "      "   3}
  le4,                         {  "     "      "   4}
  le5,                         {  "     "      "   5}
  b1,                          {width of spring 1}
  b2,                          {  "    "     "    2}
  b3,                          {  "    "     "    3}
  b4,                          {  "    "     "    4}
  b5,                          {  "    "     "    5}
  k1,                          {stiffness of spring 1 [n/m]}
  k2,                          {     "      "    "    2   "  }
  k3,                          {     "      "    "    3   "  }
  k4,                          {     "      "    "    4   "  }
  k5,                          {     "      "    "    5   "  }
  ktot,                        {total plate stiffness [n/m]}
  EI:double;                   {composite plate stiffness per width [n*m^2/m]}

 function krot(li,bi,lei:double):double;
 begin
   if (li<=0) or (lei<=0) then krot:=1e20
   else
   krot:=EI*bi/(sqr(lei)*li);
 end;

 function klin(li,bi,lei:double):double;
 begin
   if li<=0 then klin:=1e20
   else
  klin:=EI*bi/(li*li*li*(lei/li+sqr(lei/li)+1/3));
end;

begin {platestiff}
  {compute stiffness per unit width}
  if l[p1,z]=0 then EI:=E[p]*areainertia(l[p,z])
  else EI:=EIc(l[p,z],l[p1,z],E[p],E[p1]);

  {compute spring const 1}
  l1:=lbtx-l[h1s,x]/2;
  b1:=l[p,y];
  le1:=lcg[x]-(lbtx-l[h1s,x]/2);
  k1:=krot(l1,b1,le1);

  {compute spring const 2}
  l2:=l[h1s,x];
  b2:=l[p,y]-l[h1s,y]-l[h2s,y];
  le2:=lcg[x]-(lbtx+l[h1s,x]/2);
  k2:=krot(l2,b2,le2);

  {compute spring const 3}
  l3:=x2+(ltx-l[h1s,x])/2;
  b3:=l[p,y];
  le3:=lcg[x]-(lbsx-lsx/2);
  k3:=klin(l3,b3,le3);

  {compute spring const 4}
  l4:=lsx;
  b4:=l[p,y]-l[h3s,y]-l[h4s,y]-l[h5s,y]-l[h6s,y]-l[h7s,y]-l[h8s,y];
  le4:=lcg[x]-(lbsx+lsx/2);
  k4:=klin(l4,b4,le4);

  {compute spring const 5}
  if x3=0 then k5:=1.0e20
  else
  begin
    if lcg[x]<(lb[p2,x]-l[p2,x]/2) then
    begin
      l5:=lcg[x]-(lbsx+lsx/2);
      le5:=0;
    end;
    if lcg[x]>(lb[p2,x]-l[p2,x]/2) then
    begin
      l5:=x3;
      le5:=lcg[x]-(lb[p2,x]-l[p2,x]/2);
    end;
    if lcg[x]=(lb[p2,x]-l[p2,x]/2) then
    begin
```

```
      l5:=x3;
      le5:=0;
    end;
    b5:=l[p,y];
    k5:=klin(l5,b5,le5);
  end;

  {compute total linear stiffness at c.g. point g}
  ktot:=1/(1/k1+1/k2+1/k3+1/k4+1/k5);

  {compute stiffness ratio}
  sr:=ktot/klc[z];

  {compute composite rotational stiffness of end plate wrt. pt. c}
  krp:=ktot*sqr(lcg[x]);
end; {platestiff}

procedure dof2;
var
  kc,
  kth,
  Ith,
  wc,
  wth,
  k11,
  k12,
  k22,
  m11,
  m12,
  m22,
  z1,
  z2,
  z3,
  z4,
  a,
  b,
  w1a,
  w2a:double;

 begin
   z1:=1+2*lcg[x]/lac;
   z2:=1+3*lcg[x]/lac;
   z3:=2*lcg[z]/lac;
   z4:=3*lcg[z]/lac;
   k11:=4*E[p]*Iac[y,y]/(lac*lac*lac);
   k12:=6*E[p]*Iac[y,y]/(lac*lac*lac);
   k22:=12*E[p]*Iac[y,y]/(lac*lac*lac);
   m11:=mt*(sqr(z1)+sqr(z3))+4*Ip[y,y]/sqr(lac)+mb*lac/5;
  m12:=mt*(z1*z2+z3*z4)+6*Ip[y,y]/sqr(lac)+mb*lac/6;
  m22:=mt*(sqr(z2)+sqr(z4))+9*Ip[y,y]/sqr(lac)+mb*lac/7;
  a:=(k11*m22+k22*m11-2*m12*k12)/(m11*m22-sqr(m12));
  b:=(k11*k22-sqr(k12))/(m11*m22-sqr(m12));
  kc:=klc[z]/(1+3*lcg[x]/(2*lac));
  kth:=krc[y,y]*(1+lcg[x]/lac)/(1+3*lcg[x]/(2*lac));
  Ith:=mt*sqr(lcg[x]+lac)+Ip[y,y];

  w1a:=sqrt((a-sqrt(sqr(a)-4*b))/2)/(2*pi);
  w2a:=sqrt((a+sqrt(sqr(a)-4*b))/2)/(2*pi);
  wc:=sqrt(kc/mt)/(2*pi);
  wth:=sqrt(kth/Ith)/(2*pi);
end; {2dof}

procedure beam_inertia(var I:double; btemp,htemp:double);
var
  I1,
  a1,
  y1,
  I2,
  a2,
  y2,
  ybar:double;

  procedure Ie(var I1,a1,y1:double);
  var
    a,
    b,
    I:double;

  begin {Ie}
    a:=btemp/2;
    b:=htemp;
    a1:=pi*a*b/2-2*a*b*(sqrt(3)/8+pi/12);
    y1:=2*a*sqr(b)/3*power(0.75,1.5)/a1;
    I:=pi*a*power(b,3)/8-2*a*power(b,3)*(sqrt(3)/32-power(3,1.5)/64+pi/48);
    I1:=I-a1*sqr(y1);
  end;

  procedure It(var I2,a2,y2:double);
  var
    h,
    a,
    b:double;

  begin {It}
    h:=htemp/2;
    b:=btemp;
```

```
        a:=b-2*h*tan(13.5);
        a:=b*sqrt(0.75);
        a2:=h*(a+b)/2;
        y2:=h*(2*a+b)/(3*(a+b));
        I2:=power(h,3)*(sqr(a)+4*a*b+sqr(b))/(36*(a+b));
      end;

    begin {beam_inertia}
      Ie(I1,a1,y1);
      It(I2,a2,y2);
      ybar:=(y1*a1+y2*a2)/(a1+a2);
      I:=(I1+a1*sqr(ybar-y1))+(I2+a2*sqr(ybar-y2));
    end;

  begin {stiffness}
    {compute area moments of inertia of rectangular beam and V-notch [m^4]}
    Icorr:=12*(9*sqr(pi)-64)/(144*pi);              {Isemicircle*12}
    Iac[y,y]:=areainertia(h)*b0;                    {rectangle}
    Iab:=areainertia(hb)*bb;                        {retangle}

    {compute linear spring constant of beam between pts. a & c [n/m]}
    klc[z]:=3*E[p]*Iac[y,y]/(lac*lac*lac);                  {rectangle}
    klc[z]:=3*E[p]*Iac[y,y]*Icorr/(lac*lac*lac);            {half elllipse}

    {area moment of inertias for full ellipse [m^4]}
    Aac:=pi*b0*h;                                   {area of ellipse [m^2]}
    Iac[y,y]:=pi/64*h*h*h*b0;                       {inertia for ellipse about y
  axis}
    Iac[z,z]:=pi/64*b0*b0*b0*h;                     {inertia for ellipse about z
  axis}
    Iac[x,x]:=pi/64*b0*h*(sqr(b0)+sqr(h));          {inertia for ellipse about x
  axis}
    Iab:=pi/64*h*h*h*bb;                            {inertia for ellipse about y at
  pt B}

    beam_inertia(Iac[y,y],b0,h);                    {inertia for 1/4 ellipse and
  trapezoid about y axis}

    {linear spring stiffness of beam with
    ellipticall x-sec in x,y,z [N/m]}
    klc[z]:=3*E[p]*Iac[y,y]/(lac*lac*lac);
    klc[y]:=3*E[p]*Iac[z,z]/(lac*lac*lac);
    klc[x]:=E[p]*Aac/lac;

    {rotational spring stiffness of beam with
    elliptical x-sec about x,y,z [N*m]}
    krc[z,z]:=klc[y]*sqr(lac);
    krc[y,y]:=klc[z]*sqr(lac);
  krc[x,x]:=1/3*gsi*b0*h*h*h/lac;                        {thin rec.
strip}
  krc[x,x]:=pi*gsi/(16*lac)*power(b0*h,3)/(sqr(b0)+sqr(h));     {ellipse}

  {compute rotational spring constant of v-notch at point b [n*m]}
  {krbl:=E[p]*Iab/hb*tan(54.74)*sqr(h)/(sqr(h)-sqr(hb));}    {V-notch}
  {krbl:=E[p]*Iab/(2*lab);}                                  {necked beam}
  krbl:=E[p]*Iab/h*1e6;                                      {solid beam}

  {compute plate stiffness}
  platestiff;

  {algerbraic computation of frequencies for 2dof system}
  dof2;
end; {stiffness}

function varstiff(da:double):double;
  function krb(da:double):double;
  {rot stiffness as a function of crack length da}
  var
    v1,
    sa,
    alpha,                       {dimless parameter in Kies eqn}
    delcomp:dcuble;              {change in compliance of V-notch}

  begin
    {compliance based on Kies equation}
    {factor of -3/2 dropped; see Rice 3/15 p.6 of frac. mech. notes}
    alpha:=(hb-da)/h;
    c_kies:=sqr(4.12)/(E[p]*bb*sqr(h))*
             (1/sqr(alpha)+power(alpha,4)/2-3/2);
    c_kies:=sqr(4.12)/(E[p]*bb*sqr(h))*
             (1/sqr(alpha)+power(alpha,4)/2);

    {compliance based on asymtotic approx of deep crack in beam}
    {factor of -1 dropped; see Rice 3/15 p.6 of frac. mech. notes}
    c_dc:=16/(E[p]*bb*sqr(h))*(1/sqr(alpha)-1);
    c_dc:=16/(E[p]*bb*sqr(h))*(1/sqr(alpha));

    {compliance based on elliptical crack}
    c_ellip:=2.25*sqr(6*0.6*(h-hb+da))*pi/(bb*power(h,4)*E[p]);
    {factor of 2.25 comes from comparison between NASCRAC and elliptical
     crack in an infinite slab for run 061891 geometry}

    {compliance from Y. Murakami p.11 vol.1, Stress Intensity Factors
    Handbook, Pergamon Press 1987}
    {valid for alpha <= 0.7}
    alpha:=(h-hb+da)/h;
    if alpha <= 0.7 then
    begin
      v1:=1.458-0.304*alpha-0.924*sqr(alpha)+48.34*power(alpha,3)-
        123.5*power(alpha,4)+120.5*power(alpha,5);
      c_mura:=24/(E[p]*b0*sqr(h))*v1;
      delcomp:=c_mura;
    end
    else delcomp:=c_dc;

    {compliance based on empirical formula's from Paris
     1% accuracy for any alpha}
    alpha:=(h-hb+da)/h;
    sa:=sqr(alpha/(1-alpha))*(5.93-19.69*alpha+37.14*sqr(alpha)
        -35.84*power(alpha,3)+13.12*power(alpha,4));
    c_paris:=24/(b0*sqr(h)*E[p])*sa;

    krb:=2/(delcomp + 1/krb1);   {change 2 to 0.5 for conserv. est.}
  end;

begin {varstiff}
  varstiff:=krb(da)/sqr(lbc);
end; {varstiff}

begin
end.
```

```
{$N+}{compiler directive for 8087 double precision mode}
unit matrix2;
interface
uses
  init2,
  crt,
  dos;

  procedure initialize_matrices;
  procedure stiffnessmatrix(da:double; var k:sysmat);
  procedure correct_massmatrix(ma:sysmat; var mac:sysmat);
  procedure correct_stiffnessmatrix(v0,vexc,vbias:double;
                                    k:sysmat; var kc:sysmat);


implementation
uses
  dyn2,
  elec2,
  mains2;

var
  l1,                           {length of area 1}
  l2,                           {   "    "    " 2}
  l3,                           {   "    "    " 3}
  l4,                           {   "    "    " 4}
  l5,                           {   "    "    " 5}
  l6,                           {   "    "    " 6}
  l7,                           {   "    "    " 7}
  lcd,
  lce,
  lcf,
  zeta1,
  zeta2,
  zeta3,
  zeta4,
  zeta5,
  zeta6,
  eta1,
  eta2,
  eta3,
  xi1,
  xi2,
  xi3,
  lambda1,
  lambda2,
  lambda3:double;

 procedure corrmatrix;
```

```
begin{corrmatrix}
  zeta[1,1]:=sqr(zeta1);
  zeta[1,2]:=zeta1*zeta2;
  zeta[1,3]:=zeta1*zeta3;
  zeta[2,1]:=zeta[1,2];
  zeta[2,2]:=sqr(zeta2);
  zeta[2,3]:=zeta2*zeta3;
  zeta[3,1]:=zeta1*zeta3;
  zeta[3,2]:=zeta[2,3];
  zeta[3,3]:=sqr(zeta3);

  eta[1,1]:=sqr(eta1);
  eta[1,2]:=eta1*eta2;
  eta[1,3]:=eta1*eta3;
  eta[2,1]:=eta[1,2];
  eta[2,2]:=sqr(eta2);
  eta[2,3]:=eta2*eta3;
  eta[3,1]:=eta[1,3];
  eta[3,2]:=eta[2,3];
  eta[3,3]:=sqr(eta3);

  xi[1,1]:=sqr(xi1);
  xi[1,2]:=xi1*xi2;
  xi[1,3]:=xi1*xi3;
  xi[2,1]:=xi[1,2];
  xi[2,2]:=sqr(xi2);
  xi[2,3]:=xi2*xi3;
  xi[3,1]:=xi[1,3];
  xi[3,2]:=xi[2,3];
  xi[3,3]:=sqr(xi3);

  lambda[1,1]:=sqr(lambda1);
  lambda[1,2]:=lambda1*lambda2;
  lambda[1,3]:=lambda1*lambda3;
  lambda[2,1]:=lambda[1,2];
  lambda[2,2]:=sqr(lambda2);
  lambda[2,3]:=lambda2*lambda3;
  lambda[3,1]:=lambda[1,3];
  lambda[3,2]:=lambda[2,3];
  lambda[3,3]:=sqr(lambda3);

  etavec[1]:=eta1;
  etavec[2]:=eta2;
  etavec[3]:=eta3;

  xivec[1]:=xi1;
  xivec[2]:=xi2;
  xivec[3]:=xi3;
```

```
end; {corrmatrix}

procedure massmatrix(var ma:sysmat);
begin{massmatrix}
  ma[1,1]:=mt*(sqr(zeta1)+sqr(zeta4))+4*Ip[y,y]/sqr(lac)+mb*lac/5;
  ma[1,2]:=mt*(zeta1*zeta2+zeta4*zeta5)+6*Ip[y,y]/sqr(lac)+mb*lac/6;
  ma[1,3]:=mt*(zeta1*zeta3+zeta4*zeta6)+2*Ip[y,y]/(lac*lbc)+mb*lac/4;
  ma[2,2]:=mt*(sqr(zeta2)+sqr(zeta5))+9*Ip[y,y]/sqr(lac)+mb*lac/7;
  ma[2,3]:=mt*(zeta2*zeta3+zeta5*zeta6)+3*Ip[y,y]/(lac*lbc)+mb*lac/5;
  ma[3,3]:=mt*(sqr(zeta3)+sqr(zeta6))+Ip[y,y]/sqr(lbc)+mb*lac/3;
  ma[2,1]:=ma[1,2];
  ma[3,1]:=ma[1,3];
  ma[3,2]:=ma[2,3];
end;{massmatrix}

procedure correct_massmatrix(ma:sysmat; var mac:sysmat);
var
  dm:sysmat;
  i,j:integer;
  dmcg,
  dmt,
  dms,
  dmp:double;

  procedure massconst;
  const
    mcg=2;
  var
    vol:double;

    function airmass1(width,lgth,gap:double):double;
    begin
      airmass1:=3*rhoair*lgth*lgth*lgth*width/(20*gap);
    end;

    function airmass2(width,lgth,gap:double):double;
    begin
      airmass2:=3*rhoair*lgth*lgth*lgth*width/(5*gap);
    end;

  begin {massconst}
    vol:=l[p,x]*l[p,y]*l[p,z]+l[p2,x]*l[p2,y]*l[p2,z];
    dmcg:=mcg*rhoair*vol;

    dmt:=2*airmass1(l[h1s,y],(ltx-l[h1s,x])/2,h0t)+
         2*airmass1(l[h2s,y],(ltx-l[h2s,x])/2,h0t)+
         airmass1((l[p,y]-l[h1s,y]-l[h2s,y]),ltx,h0t);
    dms:=airmass1(lsx,l1,h0s)+airmass1(lsx,l2,h0s)+airmass1(lsx,l3,h0s)+
         airmass1(lsx,l4,h0s)+airmass1(lsx,l5,h0s)+airmass1(lsx,l6,h0s)+
         airmass1(lsx,l7,h0s);

    dmp:=airmass2(l[p,y],l[p,x],h0p);
  end; {massconst}

begin {correct_massmatrix}
  massconst;
  for i:=1 to nsys do
  for j:=1 to nsys do
  begin
    dm[i,j]:=eta[i,j]*dmt+xi[i,j]*dms+lambda[i,j]*dmp+zeta[i,j]*dmcg;
    mac[i,j]:=ma[i,j]+dm[i,j];
  end;
end;

procedure stiffnessmatrix(da:double; var k:sysmat);
begin
  k[1,1]:=4*E[p]*Iac[y,y]/(lac*lac*lac);
  k[1,2]:=6*E[p]*Iac[y,y]/(lac*lac*lac);
  k[1,3]:=0;
  k[2,1]:=k[1,2];
  k[2,2]:=12*E[p]*Iac[y,y]/(lac*lac*lac);
  k[2,3]:=0;
  k[3,1]:=k[1,3];
  k[3,2]:=k[2,3];
  k[3,3]:=varstiff(da);           {see unit mainstiff.pas}
end;

procedure correct_stiffnessmatrix(v0,vexc,vbias:double;
                                   k:sysmat; var kc:sysmat);

var
  i,j:integer;
  dkt,
  dks:double;

begin
  deltak(v0,vexc,vbias,dkt,dks);
  for i:=1 to nsys do
  for j:=1 to nsys do
  kc[i,j]:=k[i,j]-eta[i,j]*dkt-xi[i,j]*dks;
end;

procedure loadvector(v0:double);
var
  f0s:double;                   {sensor force [n]}
  i:integer;
```

```
begin
  torqueforce(v0,f0);
  f0s:=0;
  for i:=1 to nsys do
  p0[i]:=etavec[i]*f0+xivec[i]*f0s;
end;

procedure dampmatrix;
var
  cb,
  ct,
  cs,
  cp,
  cc:double;
  cbeam:array[1..nsys,1..nsys] of double;
  i,j:integer;

  procedure dampconst;
    function dampc(b,l,a,v:double):double;
    begin
      dampc:=3*b*l*sqr(a)*muair/(v*v*v);
    end;

    function damp0(width:double):double;
    begin
      damp0:=8*muair*width;
    end;

    function damp1(width,lgth,gap:double):double;
    begin
      damp1:=1.5*muair*width*lgth*lgth*lgth/(gap*gap*gap);
    end;

    function damp2(width,lgth,gap:double):double;
    begin
      damp2:=6*muair*width*lgth*lgth*lgth/(gap*gap*gap);
    end;

  begin {dampconst}
    cb:=damp0(b0);

    if l[h1s,x] < 0.25*h0t then ct:=damp1(l[p,y],ltx,h0t)
    else
    ct:=2*damp1(l[h1s,y],(ltx-l[h1s,x])/2,h0t)+
         2*damp1(l[h2s,y],(ltx-l[h2s,x])/2,h0t)+
         damp1((l[p,y]-l[h1s,y]-l[h2s,y]),ltx,h0t);

    if l[h3s,y] < 0.25*h0s then cs:=damp1(l[p,y],lsx,h0s)
    else
    cs:=damp1(lsx,l1,h0s)+damp1(lsx,l2,h0s)+damp1(lsx,l3,h0s)+
        damp1(lsx,l4,h0s)+damp1(lsx,l5,h0s)+damp1(lsx,l6,h0s)+
        damp1(lsx,l7,h0s);

    cp:=damp2(l[p,y],l[p,x],h0p);

    cc:=0;{dampc(bb,lbc,h-hb,abs(v3));}
  end; {dampconst}

begin{dampmatrix}
  dampconst;

  {beam damping matrix}
  cbeam[1,1]:=1/5;
  cbeam[1,2]:=1/6;
  cbeam[1,3]:=1/4;
  cbeam[2,1]:=1/6;
  cbeam[2,2]:=1/7;
  cbeam[2,3]:=1/5;
  cbeam[3,1]:=1/4;
  cbeam[3,2]:=1/5;
  cbeam[3,3]:=1/3;

  for i:=1 to nsys do
  for j:=1 to nsys do
  c[i,j]:=eta[i,j]*ct+xi[i,j]*cs+lambda[i,j]*cp+cbeam[i,j]*cb;
  c[3,3]:=c[3,3]+cc;
end;{dampmatrix}

procedure initialize_matrices;
begin{matrices}
  {compute mass inertia and stiffness from unit mainstif.pas}
  massandinertia;
  stiffness;

  {compute lengths of lever arms}
  lcd:=lbtx;
  lce:=lbsx;
  lcf:=lb[p,x];

  {compute lever arm ratios}
  zeta1:=1+2*lcg[x]/lac;
  zeta2:=1+3*lcg[x]/lac;
  zeta3:=1+lcg[x]/lbc;
  zeta4:=2*lcg[z]/lac;
  zeta5:=3*lcg[z]/lac;
  zeta6:=lcg[z]/lbc;
  eta1:=1+2*lcd/lac;
  eta2:=1+3*lcd/lac;
  eta3:=1+lcd/lbc;

  xi1:=1+2*lce/lac;
  xi2:=1+3*lce/lac;
  xi3:=1+lce/lbc;

  lambda1:=1+2*lcf/lac;
  lambda2:=1+3*lcf/lac;
  lambda3:=1+lcf/lbc;

  {compute lengths of areas 1-7 on sensor}
  l1:=l[p,y]/2-(lb[h3s,y]+l[h3s,y]/2);
  l2:=(lb[h3s,y]-l[h3s,y]/2)-(lb[h4s,y]+l[h4s,y]/2);
  l3:=(lb[h4s,y]-l[h4s,y]/2)-(lb[h5s,y]+l[h5s,y]/2);
  l4:=(lb[h5s,y]-l[h5s,y]/2)+(-lb[h6s,y]-l[h6s,y]/2);
  l5:=(-lb[h7s,y]-l[h7s,y]/2)-(-lb[h6s,y]+l[h6s,y]/2);
  l6:=(-lb[h8s,y]-l[h8s,y]/2)-(-lb[h7s,y]+l[h7s,y]/2);
  l7:=l[p,y]/2-(-lb[h8s,y]+l[h8s,y]/2);

  corrmatrix;
  massmatrix(ma);
  stiffnessmatrix(da,k);
  loadvector(v0);
  dampmatrix;
end; {initialize_matrices}

begin
end.
```

```
{$N+}{compiler directive for 8087 double precision mode}
unit modal2;
interface
uses
  crt,
  dos,
  init2;

procedure EVP(mass,stiff,damp:sysmat; load:sysvec; np:integer;
                var evec:sysmat; var eval, chat, phat:sysvec);

procedure statics(k:sysmat; p:sysvec; var ustatic:sysvec; np:integer);
procedure normalize(phi:sysmat; np:integer; var phinorm:sysmat);
procedure modal_matrix(m,k,c,phinorm:sysmat; p:sysvec; np:integer;
                        var mhat,khat,chat,phat:sysvec);


implementation
const
  npmax=5;

type
  g]npnp = ARRAY [1..npmax,1..npmax] OF double;
  glnp = ARRAY [1..npmax] OF double;

PROCEDURE jacobi(VAR a: glnpnp; n: integer;VAR d: glnp;
       VAR v: glnpnp; VAR nrot: integer);

{note: This procedure solves the standard EVP: (A-d*I)V=0.
       a is n by n symetric real matrix;
       d is vector containing eigenvalues;
       V is n by n modal matrix.}

LABEL 99;
CONST
   nmax=100;
VAR
   j,iq,ip,i: integer;
   tresh,theta,tau,t,sm,s,h,g,c: double;
   b,z: ARRAY [1..nmax] OF double;
BEGIN
   FOR ip := 1 to n DO BEGIN
      FOR iq := 1 to n DO BEGIN
         v[ip,iq] := 0.0
      END;
      v[ip,ip] := 1.0
   END;
   FOR ip := 1 to n DO BEGIN
      b[ip] := a[ip,ip];
   d[ip] := b[ip];
   z[ip] := 0.0
END;
nrot := 0;
FOR i := 1 to 50 DO BEGIN
   sm := 0.0;
   FOR ip := 1 to n-1 DO BEGIN
      FOR iq := ip+1 to n DO BEGIN
         sm := sm+abs(a[ip,iq])
      END
   END;
   IF (sm = 0.0) THEN GOTO 99;
   IF (i < 4) THEN tresh := 0.2*sm/sqr(n)
   ELSE tresh := 0.0;
   FOR ip := 1 to n-1 DO BEGIN
      FOR iq := ip+1 to n DO BEGIN
         g := 100.0*abs(a[ip,iq]);
         IF ((i > 4) AND ((abs(d[ip])+g) = abs(d[ip]))
            AND ((abs(d[iq])+g) = abs(d[iq]))) THEN
            a[ip,iq] := 0.0
         ELSE IF (abs(a[ip,iq]) > tresh) THEN BEGIN
            h := d[iq]-d[ip];
            IF ((abs(h)+g) = abs(h)) THEN BEGIN
               t := a[ip,iq]/h
            END ELSE BEGIN
               theta := 0.5*h/a[ip,iq];
               t := 1.0/(abs(theta)+sqrt(1.0+sqr(theta)));
               IF (theta < 0.0) THEN t := -t
            END;
            c := 1.0/sqrt(1+sqr(t));
            s := t*c;
            tau := s/(1.0+c);
            h := t*a[ip,iq];
            z[ip] := z[ip]-h;
            z[iq] := z[iq]+h;
            d[ip] := d[ip]-h;
            d[iq] := d[iq]+h;
            a[ip,iq] := 0.0;
            FOR j := 1 to ip-1 DO BEGIN
               g := a[j,ip];
               h := a[j,iq];
               a[j,ip] := g-s*(h+g*tau);
               a[j,iq] := h+s*(g-h*tau)
            END;
            FOR j := ip+1 to iq-1 DO BEGIN
               g := a[ip,j];
               h := a[j,iq];
               a[ip,j] := g-s*(h+g*tau);
```

```
                  a[j,iq] := h+s*(g-h*tau)
               END;
               FOR j := iq+1 to n DO BEGIN
                  g := a[ip,j];
                  h := a[iq,j];
                  a[ip,j] := g-s*(h+g*tau);
                  a[iq,j] := h+s*(g-h*tau)
               END;
               FOR j := 1 to n DO BEGIN
                  g := v[j,ip];
                  h := v[j,iq];
                  v[j,ip] := g-s*(h+g*tau);
                  v[j,iq] := h+s*(g-h*tau)
               END;
               nrot := nrot+1
            END
         END
      END
   END;
   FOR ip := 1 to n DO BEGIN
      b[ip] := b[ip]+z[ip];
      d[ip] := b[ip];
      z[ip] := 0.0
   END;
   writeln('pause in routine JACOBI');
   writeln('50 iterations should not happen'); readln;
99:
END;

procedure matrixmultiply(var c:glnpnp;a,b:glnpnp; np:integer);
{This procedure multiplies two n by n matrices: a*b=c}
var
  l,m,i,j:integer;                 {array indices}
begin
  for m:=1 to np do
  begin
    for l:=1 to np do
    begin
      c[l,m]:=0;
      for j:=1 to np do c[l,m]:=c[l,m]+a[l,j]*b[j,m];
    end;
  end;
end;

procedure transpose(var a:glnpnp; b:glnpnp; np:integer);
var
  i,j:integer;
begin
```

```
  {a=b^T}
  for j:=1 to np do
  begin
    for i:=1 to np do a[i,j]:=b[j,i];
  end;
end;

procedure matrixdefine(a:sysmat; var b:glnpnp; np:integer);
var
  i,j:integer;
begin
  for i:=1 to npmax do
  for j:=1 to npmax do
  if (i > np) or (j > np) then b[i,j]:=0
  else b[i,j]:=a[i,j];
end;

procedure vectordefine(a:sysvec; var b:glnpnp; np:integer);
var
  i,j:integer;
begin
  for i:=1 to npmax do
  for j:=1 to npmax do
  if (i > np) or (j > 1) then b[i,j]:=0
  else b[i,1]:=a[i];
end;

procedure nullglnpnp(var null:glnpnp);
var
  i,j:integer;
begin
  for i:=1 to npmax do
  for j:=1 to npmax do
  null[i,j]:=0;
end;

procedure nullglnp(var null:glnp);
var
  i:integer;
begin
  for i:=1 to npmax do
  null[i]:=0;
end;

procedure nullsys(var null:sysmat);
var
  i,j:integer;
begin
```

```
  for i:=1 to nsys do
  for j:=1 to nsys do
  null[i,j]:=0;
end;

procedure nullsysvec(var null:sysvec);
var
  i:integer;
begin
  for i:=1 to nsys do
  null[i]:=0;
end;

procedure EVP(mass,stiff,damp:sysmat; load:sysvec; np:integer;
              var evec:sysmat; var eval, chat, phat:sysvec);

      {This procedure solves the generalized EVP (K-d*M)phi=0.
       phi^T*M*phi=I, modal matrix normalized wrt. mass matrix;
       phi^T*K*phi=D, D is diagonal containing E-values}

var
  null,                         {zero matrix}
  m,                            {mass matrix}
  k,                            {stiffness matrix}
  c,                            {damping matrix}
  p,                            {loading matrix}
  dum,                          {dummy matrix: k*(s^T)^(-1)}
  vm,                           {eigenvectors of m}
  s,                            {definition of s: s=vm*em^1/2}
  sinv,                         {inverse of s: s^(-1)}
  strans,                       {transpose of s: s^T}
  stransinv,                    {inverse of transpose of s: (s^T)^(-1)}
  kcurl,                        {standard form of k: s^(-1)*k*(s^T)^(-1)}
  vcurl,                        {eigenvectors of kcurl}
  ccurl,                        {modal damping matrix npmax by npmax}
  pcurl,                        {modal loading matrix}
  phitrans,                     {transpose of eigenvectors}
  phi:glnpnp;                   {matrix of eigenvectors}
  em,                           {eigenvalues of m}
  ecurl:glnp;                   {eigenvalues of kcurl}
  nrotcurl,                     {# rotations on kcurl}
  nrotm,                        {# rotations on m}
  i,j:integer;                  {array indices}

  PROCEDURE eigsrt(VAR d: glnp; VAR v: glnpnp; n: integer);

  {note: This procedure sorts eigenvalues and corresponding
```

```
        eigenvectors in decending order via}

  VAR
     k,j,i: integer;
     p: double;
  BEGIN
     FOR i := 1 to n-1 DO BEGIN
        k := i;
        p := d[i];
        FOR j := i+1 to n DO BEGIN
           IF (d[j] >= p) THEN BEGIN
              k := j;
              p := d[j]
           END
        END;
        IF (k <> i) THEN BEGIN
           d[k] := d[i];
           d[i] := p;
           FOR j := 1 to n DO BEGIN
              p := v[j,i];
              v[j,i] := v[j,k];
              v[j,k] := p
           END
        END
     END
  END;

begin{EVP}
  matrixdefine(mass,m,np);
  matrixdefine(stiff,k,np);
  matrixdefine(damp,c,np);
  vectordefine(load,p,np);
  nullglnpnp(null);

  {decompose mass matrix and find square root: m=vm*em*vm^T=s*s^T}
  jacobi(m,np,em,vm,nrotm);
  s:=vm;
  stransinv:=vm;
  for j:=1 to np do
  begin
    for i:=1 to np do
    begin
      s[i,j]:=vm[i,j]*sqrt(em[j]);
      stransinv[i,j]:=vm[i,j]/sqrt(em[j]);
    end;
  end;

  {compute s^(-1) and s^T}
  transpose(strans,s,np);
  transpose(sinv,stransinv,np);

  {compute kcurl}
  matrixmultiply(dum,k,stransinv,np);
  matrixmultiply(kcurl,sinv,dum,np);

  {decompose kcurl into modal form: kcurl=vcurl*ecurl*vcurl^T}
  jacobi(kcurl,np,ecurl,vcurl,nrotcurl);

  {compute modal matrix: phi=(s^T)^(-1)*vcurl}
  matrixmultiply(phi,stransinv,vcurl,np);

  {zero superfulous elements}
  for i:=1 to npmax do
  begin
    for j:= 1 to npmax do
    begin
      if (i > np) or (j > np) then phi[i,j]:=0;
    end;
    if i > np then ecurl[i]:=0;
  end;

  {sort E-values and E-vectors}
  eigsrt(ecurl,phi,np);

  for i:=1 to nsys do
  begin
    for j:=1 to nsys do evec[i,j]:=phi[i,j];
    eval[i]:=ecurl[i];
  end;

  {compute modal damping matrix neglecting off diagonal terms}
  phitrans:=null;
  dum:=null;
  ccurl:=null;
  pcurl:=null;
  transpose(phitrans,phi,np);
  matrixmultiply(dum,c,phi,np);
  matrixmultiply(ccurl,phitrans,dum,np);
  for i:=1 to nsys do chat[i]:=ccurl[i,i];

  {compute modal load vector}
  matrixmultiply(pcurl,phitrans,p,np);
  for i:=1 to nsys do phat[i]:=pcurl[i,1];
end;{EVP}

procedure statics(k:sysmat; p:sysvec; var ustatic:sysvec; np:integer);
```

```
var
  i,j,
  nrot:integer;
  evalinv,
  eval:glnp;
  utemp,
  ptemp,
  null,
  dum,
  transevec,
  evec,
  ktempinv,
  ktemp:glnpnp;
  kinv:sysmat;

begin
  nullglnp(eval);
  nullglnp(evalinv);
  nullglnpnp(null);
  evec:=null;
  utemp:=null;
  ktempinv:=null;
  transevec:=null;

  matrixdefine(k,ktemp,np);
  vectordefine(p,ptemp,np);
  jacobi(ktemp,np,eval,evec,nrot);
  transpose(transevec,evec,np);

  for i:=1 to np do
  for j:=1 to np do
  dum[i,j]:=transevec[i,j]/eval[i];

  matrixmultiply(ktempinv,evec,dum,np);
  matrixmultiply(utemp,ktempinv,ptemp,np);
  for i:=1 to np do
  begin
    for j:=1 to np do kinv[i,j]:=ktempinv[i,j];
    ustatic[i]:=utemp[i,1];
  end;
end;

procedure normalize(phi:sysmat; np:integer; var phinorm:sysmat);
var
  i,j:integer;
  mag:double;

begin{normalize}
```

```
  nullsys(phinorm);
  for j:=1 to np do
  begin
    mag:=0;
    for i:=1 to np do
    mag:=mag + sqr(phi[i,j]);
    mag:=sqrt(mag);
    for i:=1 to np do
    phinorm[i,j]:=phi[i,j]/mag;
  end;
end;{normalize}

procedure modal_matrix(m,k,c,phinorm:sysmat; p:sysvec; np:integer;
                       var mhat,khat,chat,phat:sysvec);

var
  i,j:integer;
  dum,
  mtemp,
  ktemp,
  ctemp,
  ptemp,
  phitemp,
  phinormtemp,
  phinormtemptrans,
  mhattemp,
  khattemp,
  chattemp,
  phattemp:glnpnp;

begin{modal_matrix}
  nullsysvec(mhat);
  nullsysvec(khat);
  nullsysvec(chat);
  nullsysvec(phat);
  nullglnpnp(dum);

  matrixdefine(m,mtemp,np);
  matrixdefine(k,ktemp,np);
  matrixdefine(c,ctemp,np);
  matrixdefine(phinorm,phinormtemp,np);
  vectordefine(p,ptemp,np);
  transpose(phinormtemptrans,phinormtemp,np);

  matrixmultiply(dum,mtemp,phinormtemp,np);
  matrixmultiply(mhattemp,phinormtemptrans,dum,np);

  matrixmultiply(dum,ktemp,phinormtemp,np);
  matrixmultiply(khattemp,phinormtemptrans,dum,np);
  matrixmultiply(dum,ctemp,phinormtemp,np);
  matrixmultiply(chattemp,phinormtemptrans,dum,np);

  matrixmultiply(phattemp,phinormtemptrans,ptemp,np);
  for i:=1 to np do
  begin
    mhat[i]:=mhattemp[i,i];
    khat[i]:=khattemp[i,i];
    chat[i]:=chattemp[i,i];
    phat[i]:=phattemp[i,1];
  end;
end;{modal_matrix}

begin
end.
```

```
{$N+}{compiler directive for 8087 double precision mode}
unit dyn2;
interface
uses
  crt,
  init2,
  dos;

procedure dynamics(var vc,thetac:double);

procedure structure(m,k,c:sysmat; p:sysvec; n:integer; verror:double;
                    var phi:sysmat; var omega:sysvec;
                    var vx1,thetax1,v3static1,momx2,sigmax2,
                    repole,impole:double);

implementation
uses
  elec2,
  modal2,
  mains2,
  matrix2;

var
  svec,
  mvec,
  vvec,
  thetavec:sysvec;

procedure bridge_analysis;
const
  cx=125e-6;
  {cx=40e-6; design 1}

var
  EA,
  pt,
  pcr,
  mbr,
  lx,
  EI:double;

begin
  {bridge electrode frequency}
  if l[p2,z]<>0 then
  begin
    lx:=l[p,y]+2*cx;
    EI:=1/12*ltx*power(l[p2,z],3)*E[p2];
    mbr:=rho[p2]*ltx*l[p2,z];
    wbr:=22.4/sqr(lx)*sqrt(EI/mbr);
    vbr:=f0/l[p,y]*power(lx,4)/(384*EI);
    kbr:=48*EI/power(lx,3);

    {bridge electrode buckling}
    pcr:=4*sqr(pi)*EI/sqr(lx);
    EA:=l[p2,z]*ltx*E[p2];
    pt:=(alphaau-alphasi)*Tsin*EA;
    bucr:=pcr/pt;
    Tmax:=pcr/((alphaau-alphasi)*EA);
  end
  else
  begin
    wbr:=0;
    vbr:=0;
    bucr:=0;
    Tmax:=0;
  end;
end; {bridge_analysis}

procedure other_freq;
var
  kzz,
  Izz,
  r1,
  r2:double;

begin {other_freq}
  {end plate frequency}
  r1:=sqrt(sqr(lcg[x])+sqr(lcg[z]));
  r2:=sqrt(sqr(lcg[y])+sqr(lcg[z]));
  wp:=sqrt(krp/(mt*sqr(r1)+Ip[y,y]));              {rad/s}

  {torsional frequency about x axis}
  wxx:=sqrt(krc[x,x]/(mt*sqr(r2)+Ip[x,x]));       {rad/s}

  {rotational frequency about z axis}
  kzz:=krc[z,z]*(1+lcg[x]/lac)/(1+3*lcg[x]/(2*lac));
  Izz:=mt*sqr(lcg[x]+lac)+Ip[z,z];
  wzz:=sqrt(kzz/Izz);                              {rad/s}
end; {other_freq}

procedure indentation;
const
  Pac=60e-3;
  phi=0.66;
  ksi=0.069;
```

```
var
  c,
  s0,
  beta,
  delta:double;

begin
  c:=h-hb;
  beta:=1-power(rb/rsi,3);
  s0:=beta*esi*cs/((1-nusi)*csi);
  Pind:=(k1c111-s0*sqrt(pi*c)*phi)/ksi*power(c,1.5);
  {c:=0;                                                       }
  {repeat                                                      }
  {  c:=c+h/1000;                                              }
  {  delta:=k1c111-ksi*Pac/power(c,1.5)-s0*sqrt(pi*c)*phi;     }
  {until abs(delta) < 0.01*k1c111;                             }
  {c:=0;                                                       }
end;

procedure vectors(x1,x2:double);
begin
  {displacement vectors at x1}
  vvec[1]:=sqr(x1/lac);
  vvec[2]:=power(x1/lac,3);
  vvec[3]:=(x1-lab)/lbc;
  thetavec[1]:=2*x1/sqr(lac);
  thetavec[2]:=3*sqr(x1)/power(lac,3);
  thetavec[3]:=1/lbc;

  {moment and stress vectors at x2}
  mvec[1]:=2*e[p]*Iac[y,y]/sqr(lac);
  mvec[2]:=6*e[p]*Iac[y,y]*x2/power(lac,3);
  mvec[3]:=0;
  svec[1]:=mvec[1]*h/(2*Iac[y,y]);
  svec[2]:=mvec[2]*h/(2*Iac[y,y]);
  svec[3]:=mvec[3]*h/(2*Iac[y,y]);
end;

procedure structure(m,k,c:sysmat; p:sysvec; n:integer; verror:double;
                    var phi:sysmat; var omega:sysvec;
                    var vx1,thetax1,v3static1,momx2,sigmax2,
                    repole,impole:double);
var
  i,j:integer;
  chat,
  phat,
  omegasqr:sysvec;
  x:double;
```

```
begin {structure}
  EVP(m,k,c,p,n,phi,omegasqr,chat,phat);
  if chat[n]=0 then
  begin
    x:=phat[n]*qvac/omegasqr[n];
    chat[n]:=sqrt(omegasqr[n])/qvac;
  end
  else
  x:=phat[n]/(sqrt(omegasqr[n])*chat[n])/sqrt(1+sqr(verror/2));
  vx1:=0;
  thetax1:=0;
  momx2:=0;
  sigmax2:=0;
  for i:=1 to n do
  begin
    vx1:=vx1+vvec[i]*phi[i,n];
    thetax1:=thetax1+thetavec[i]*phi[i,n];
    momx2:=momx2+mvec[i]*phi[i,n];
    sigmax2:=sigmax2+svec[i]*phi[i,n];
  end;
  vx1:=vx1*x;
  thetax1:=thetax1*x;
  momx2:=momx2*x;
  sigmax2:=sigmax2*x;
  if n = 3 then
  begin
    v3:=phi[3,n]*x;
    v3static1:=-2/(pi*k[3,3]*lbc)*momx2;
  end
  else
    v3static:=0;
  {v3static:=0.0001e-6;}
  omega:=omegasqr;
  for i:=1 to n do
  begin
    if omegasqr[i] > 0 then omega[i]:=sqrt(omegasqr[i])
    else omega[i]:=0;
  end;
  repole:=-chat[n]/2;
  impole:=sqrt(omegasqr[n]-sqr(chat[n]/2));
end;  {structure}

procedure vacuum(m,k:sysmat; n:integer; var phivac:sysmat;
          var omegavac:sysvec; var v0vac,repolevac,impolevac:double);
var
  i,j:integer;
  kctemp,
  c:sysmat;
  vac_const,
  vcvac,
  thetacvac,
  v3staticvac,
  mombvac,
  sigmabvac:double;

begin
  for i:=1 to n do
  for j:=1 to n do c[i,j]:=0;
  correct_stiffnessmatrix(0,vexc,vbias,k,kctemp);
  structure(m,kctemp,c,p0,n,0,phivac,omegavac,vcvac,thetacvac,v3staticvac,
            mombvac,sigmabvac,repolevac,impolevac);
  vac_const:=vc/vcvac;
  vcvac:=vc;
  thetacvac:=thetacvac*vac_const;
  sigmabvac:=sigmabvac*vac_const;
  mombvac:=mombvac*vac_const;
  v3staticvac:=v3staticvac*vac_const;
  v0vac:=v0*sqrt(vac_const);
end;

procedure staticanalysis(kc:sysmat; p0:sysvec; n:integer);
var
  i,j:integer;
  sigmabstatic,
  ustatic:sysvec;

begin
  statics(kc,p0,ustatic,n);      {see modal.pas}
  vcstatic:=0;
  thetacstatic:=0;
  for i:=1 to n do
  begin
    vcstatic:=vcstatic + ustatic[i]*vvec[i];
    thetacstatic:=thetacstatic + ustatic[i]*thetavec[i];
  end;
    vcstatic:=vcstatic+v3static;
    thetacstatic:=thetacstatic+v3static/lbc;
end;

procedure stability_check(n:integer);
var
  dktor,
  dksens:double;
  kctemp,
  evectemp:sysmat;
  chattemp,
  phattemp,
  evaltemp:sysvec;

begin
  correct_stiffnessmatrix(0,vexc,vbias,k,kctemp);
  EVP(k,eta,c,p0,n,evectemp,evaltemp,chattemp,phattemp);
  deltak(1.0,0,0,dktor,dksens);
  vsnap:=sqrt(1/(dktor*2*evaltemp[1]));
end;

procedure dynamics(var vc,thetac:double);
var
  error,
  dt,
  ds,
  dti,
  dsi:double;
begin
  if ndof = 2 then
  begin
    initialize_matrices;
    vectors(lac,lab);
    correct_massmatrix(ma,mac);
    correct_stiffnessmatrix(v0,vexc,vbias,k,kc);
    structure(mac,kc,c,p0,ndof,0,phi,omega,vc,thetac,v3static,momb,sigmab,
            repole,impole);
  end
  else
  begin
    dti:=0;
    dsi:=0;
    v3:=0;
    repeat
      initialize_matrices;
      vectors(lac,lab);
      correct_massmatrix(ma,mac);
      correct_stiffnessmatrix(v0,vexc,vbias,k,kc);
      structure(mac,kc,c,p0,ndof,0,phi,omega,vc,thetac,v3static,momb,sigmab,
               repole,impole);
      dt:=(lbtx/lbc+1)*v3static;
      ds:=(lbsx/lbc+1)*v3static;
      error:=abs((dt-dti)/dt);
      ht:=h0t-dt;
      hs:=h0s-ds;
      dti:=dt;
      dsi:=ds;
    until error < 1e-3;
  end;
  normalize(phi,ndof,phinorm);
  modal_matrix(mac,kc,c,phinorm,p0,ndof,mhat,khat,chat,phat);
  vacuum(ma,k,ndof,phivac,omegavac,v0vac,repolevac,impolevac);
  normalize(phivac,ndof,phinormvac);
  staticanalysis(kc,p0,ndof);
  stability_check(ndof);
  other_freq;
  bridge_analysis;
  indentation;
end; {dynamics}

begin
  verror:=0;
end.
```

```
{$N+}{compiler directive for 8087 double precision mode}
unit frac2;
interface
uses
  crt,
  init2,
  dos;

procedure fracture_analysis;

implementation
uses
  elec2,
  modal2,
  mains2,
  dyn2,
  matrix2;

var
  mhat0,
  khat0,
  chat0,
  phat0,
  omega0:sysvec;
  phi0norm,
  phi0,
  k0,
  k0c:sysmat;
  sigmab0,
  repole0,
  impole0,
  thetac0,
  vc0,
  v3static0,
  momb0:double;

 function fac(num:integer):double;
 var
   i:integer;
   p:double;
 begin
   if (num=0) or (num=1) then fac:=1
   else
   begin
     p:=num;
     for i:=num downto 2 do
     begin
       p:=p*(i-1);
```

```
    end;
    fac:=p;
  end;
end;

function ka(da:double):double;            {Kies equation without moment}
var
  f1,
  alpha:double;

begin
  {stress intensity based on Kies Eqn}
  alpha:=(hb-da)/h;
  kakies:=4.12/(bb*sqrt(h*h*h))*
          sqrt(1/(alpha*alpha*alpha)-alpha*alpha*alpha);

  {for alpha <= 0.6 stress intensity from Murakami p.11 vol 1
   for alpha > 0.6 asymptotic expansion for deep cracked beam}
  alpha:=(h-hb+da)/h;
  if alpha <=0.6 then
  begin
    f1:=1.122-1.4*alpha+7.33*sqr(alpha)-13.08*power(alpha,3)+
      14.0*power(alpha,4);
    ka:= 6/(bb*sqr(h))*sqrt(pi*(h-hb+da))*f1;
    kamur:= 6/(bb*sqr(h))*sqrt(pi*(h-hb+da))*f1;
  end
  else
  begin
    ka:=4/(bb*power(hb-da,3/2));
    kadc:=4/(bb*power(hb-da,3/2));
  end;

  {stress intensity based on elliptical crack in infinite slab;
   factor of 1.5 from NASCRAC comparison}
  kaellip:=1.5*0.4*6*sqrt(pi*(h-hb+da))/(bb*sqr(h));
end;

procedure stressintensity(da,moment:double; var k1:double);
begin
  k1:=1.5*ka(da)*moment;
  {factor of 1.5 accounts for break test data at Oak Ridge: 11/17/91-11/23/91}
end;

procedure crackvelocity(var vavg:double; nexp:double; k1i:double);
var
  gamma,
  vavgs,                        {average static velocity [m/s]}
  g:double;                     {dynamic velocity scale factor}
```

```
procedure velocity_corr1;
var
  m,
  n:longint;

begin {velocity_corr1}
  n:=round(nexp);
  if odd(n) then
  begin {n is odd}
    m:=1;
    g:=1;
    for m:=1 to trunc((n-1)/2) do g:=g*2*m/(2*m+1);
    g:=g/pi;
  end {n is odd}
  else
  begin {n is even}
    m:=1;
    g:=1;
    for m:=1 to trunc(n/2) do g:=g*(2*m-1)/(2*m);
    g:=g/2;
  end; {n is even}
end;{velocity_corr1}

procedure velocity_corr2;
var
  l,m,n:integer;                {indices}
  c1,c2,c3,c4:double;           {calculation constants}

begin {velocity_corr2}
  g:=0;
  m:=round(nexp/2);
  n:=round(nexp);
  c1:=fac(n);
  for l:=0 to m do
  begin
    c2:=power(pi/4,2*l);
    c3:=fac(n-2*l);
    c4:=sqr(fac(l));
    g:=g+c2/(c3*c4);
  end;
  g:=g*c1/power(pi,n);
end; {velocity_corr2}

begin {crackvel}
  gamma:=k1i/k1c111;
  velocity_corr2;
  vavg:=vsi02*power(gamma,nexp)*g;
  vavgs:=vsi02*power(gamma,nexp);
end; {crackvel}

procedure cracktime(k1i,k1f,dk1da,nexp:double;var tau:double);
{note: tau given in hrs}
var
  vi,
  gamma:double;
begin
  crackvelocity(vi,nexp,k1i);
  gamma:=k1i/k1f;
  tau:=k1i/(vi*dk1da*(nexp-1))*(1-power(gamma,nexp-1));
  tau:=tau/(3600);
end; {cracktime}

procedure frac_param(n:integer; nexp:double);
var
  hits_torquer,
  hits_sensor:boolean;

  procedure zero_frac_param;
  begin
  end;

begin {frac_param}
  if n = 2 then
  begin
    ta:=0;
    tf:=0;
    dw:=0;
    dwdt:=0;
    dk1da:=0;
    v0c:=0;
    verror:=0;
    k1:=0;
    k1i:=0;
    pctk1ci:=0;
    pctk1c:=0;
    momb0:=momb;
  end
  else
  begin {fracture analysis}
    stiffnessmatrix(0,k0);
    correct_stiffnessmatrix(v0,vexc,vbias,k0,k0c);
    structure(mac,k0c,c,p0,n,0,phi0,omega0,vc0,
             thetac0,v3staticO,momb0,sigmab0,repole0,impole0);
    dw:=impole-impole0;
    stressintensity(da,momb,k1);
```

```
   stressintensity(0,momb0,k1i);
   pctk1c:=k1/k1cc;
   pctk1ci:=k1i/k1cc;
   dk1da:=(k1-k1i)/da+(momb-momb0)/da;
   dk1da:=(k1-k1i)/da;
   v0c:=sqrt(1/pctk1ci)*v0;
   mombf:=momb*k1c111/k1;
   if (pctk1c > 1) or (pctk1ci > 1 ) then
   begin
     ta:=0;
     tf:=0;
     dwdt:=dwdtmax;
     verror:=verrormax;
     clrscr;
     writeln('DEVICE FAILED INSTANTEOUSLY');
     readln;
   end
   else
   begin
     hits_torquer:=false;
     hits_sensor:=false;
     dimension_check(vc/pctk1ci,thetac/pctk1ci,v3static/pctk1ci,
                    hits_sensor,hits_torquer);
     if hits_sensor then
     begin
       clrscr;
       writeln('CANNOT BREAK DEVICE WITH DRIVE ELECTRONICS: PLATE HITS
SENSOR.');
       readln;
     end;
     if hits_torquer then
     begin
       clrscr;
       writeln('CANNOT BREAK DEVICE WITH DRIVE ELECTRONICS: PLATE HITS
TORQUER.');
       readln;
     end;
     cracktime(k1i,k1,dk1da,nexp,ta);
     cracktime(k1i,k1f,dk1da,nsi02,tf);
     dwdt:=dw/(ta*3600);
     velocity_error(dwdt,verror);
    end;
  end;{failure analysis}
end; {frac_param}

procedure fatigue_sim(m,c:sysmat;p:sysvec;n:integer);
const
  inc=1.0e-8;
  tol=1e-4;
  narray=1000;

var
  filename:string;
  in_filename,
  out_filename:text;
  t,
  f:array[0..narray] of double;
  i,
  imax:integer;
  ki,
  kic,
  phii:sysmat;
  chati,
  phati,
  omegai:sysvec;
  vci,
  thetaci,
  v3statici,
  mombi,
  repolei,
  dt,
  df,
  w1,
  w2,
  dadt,
  da,
  da0,
  k1i,
  sigmabi:double;

begin
  clrscr;
  write('Input file name = ');
  readln(filename);
  assign(in_filename,filename);
  reset(in_filename);
  i:=0;
  while not eof(in_filename) or (i = narray) do
  begin
    i:=i+1;
    readln(in_filename,t[i],f[i]);
    f[i]:=f[i]*2*pi;
  end;
  close(in_filename);
  write('Output file name = ');
  readln(filename);
  assign(out_filename,filename);
  rewrite(out_filename);
  imax:=i;
  t[0]:=0;
  f[0]:=0;
  da0:=0;
  da:=inc;
  for i:=1 to imax do
  begin
    dt:=t[i]-t[i-1];
    w2:=impole0+f[i];
    df:=f[i];
    repeat
    begin
      da:=da-inc*(f[i]-df)/sqrt(sqr(f[i])+sqr(dw));
      stiffnessmatrix(da,ki);
      correct_stiffnessmatrix(v0,vexc,vbias,ki,kic);
      structure(mac,kic,c,p0,ndof,0,phii,omegai,vci,
                thetaci,v3statici,mombi,sigmabi,repolei,w1);
      df:=w1-w2;
    end;
    until abs((f[i]-df)/f[i]) < tol;
    stressintensity(da,mombi,k1i);
    dadt:=(da-da0)/dt;
    {writeln(out_filename,k1i:15,'     ',dadt:15);}
    writeln(out_filename,t[i]:15,'    ',da:15);
    {writeln(out_filename,k1i:15,'     ',da:15);  }
    da0:=da;
    clrscr;
    writeln(imax-i:5);
  end; {for i:=1 to imax}
  close(out_filename);
end;{fatigue_sim}

procedure fail_param(np:integer; nexp:double);
var
  quit_sim:char;
  hits_torquer,
  hits_sensor,
  exceeds_bandwidth:boolean;
  n:integer;
  nmax,
  vi0,
  vi,
  k1,
  k10,
  t,
  wi,
  wi0,
  di,
  di0,
  inc,
  dwdti,
  dk1di,
  vci,
  thetaci,
  v3statici,
  sigmabi,
  mombi,
  repolei,
  impolei,
  eoutu0,
  eoutui,
  eoutd0,
  eoutdi:double;
  ki,kic,phii:sysmat;
  omegai:sysvec;
  outfile:text;
  filename:string[12];

begin
  clrscr;
  writeln('Fracture simulation: q to quit, return to start.');
  quit_sim:=readkey;
  if quit_sim='q' then exit;
  clrscr;
  writeln('Specify fracture simulation output file or return to screen');
  readln(filename);
  if filename='' then filename:='CON';
  assign(outfile,filename);
  rewrite(outfile);
  clrscr;
  writeln(outfile,
  '    time [sec]        freq [hz]         dw [hz]          eoutd [volts]');
  nmax:=100;
  nmax:=tf*3600*5;  { # samples = failure time * sample rate}
  inc:=hb/nmax;
  n:=0;
  wi0:=0;
  k10:=k1i;
  di0:=0;
  tf:=0;
  verror:=0;
  hits_sensor:=false;
  exceeds_bandwidth:=false;
  crackvelocity(vi0,nexp,k10);
```

```
sensoroutput(vc0,thetac0,eoutu0,eoutd0);
writeln(outfile,tf:15:5,impole0/(2*pi):15:5,wi0/(2*pi):15:5,eoutd0:15:5);
repeat
  n:=n+1;
  di:=di0+n*inc;
  stiffnessmatrix(di,ki);
  correct_stiffnessmatrix(v0,vexc,vbias,ki,kic);
  structure(mac,kic,c,p0,np,verror,phii,omegai,vci,
            thetaci,v3statici,mombi,sigmabi,repolei,impolei);
  dimension_check(vci,thetaci,v3statici,hits_sensor,hits_torquer);
  wi:=impolei-impole0;
  stressintensity(di,mombi,k1);
  dk1di:=(k1-k10)/inc;
  cracktime(k10,k1,dk1di,nexp,t);
  t:=t*3600;
  tf:=tf+t;
  dwdti:=(wi-wi0)/t;
  velocity_error(dwdti,verrorf);
  if (not hits_sensor) and (not hits_torquer) then
    sensoroutput(vci,thetaci,eoutui,eoutdi)
  else
  begin
    eoutui:=0;
    eoutdi:=0;
  end;
  writeln(outfile,tf:15:5,impolei/(2*pi):15:5,wi/(2*pi):15:5,eoutdi:15:5);

  {The following is a direct integration scheme}
  {crackvelocity(vi,nexp,k1);          }
  {viavg:=(vi+vi0)/2;                  }
  {t:=inc/viavg;                       }
  {dvi:=(vi-vi0)/vi0;                  }
  {if dvi > 0.05 then inc:=inc*0.5;    }

  if (abs(verrorf) > verrormax) and (exceeds_bandwidth = false) then
  begin
    exceeds_bandwidth:=true;
    vctl:=vci;
    thetactl:=thetaci;
    tctl:=tf/3600;
    dwctl:=wi;
    dactl:=di;
    pctk1ctl:=k1/k1cc;
    dk1dactl:=dk1di;
    dwdtctl:=dwdti;
    writeln(outfile,
    '--------------------CONTROLLER BANDWIDTH EXCEEDED-----------------');
  end;
  wi0:=wi;
  k10:=k1;
  vi0:=vi;
until (k1/k1f > 1.0) or (n=nmax) or (keypressed and ('q'=readkey)) or
      (hits_sensor and hits_torquer and (exceeds_bandwidth=false));
close(outfile);
if filename='CON' then readln;
tf:=tf/3600;
vcf:=vci;
thetacf:=thetaci;
dk1daf:=dk1di;
dwdtf:=dwdti;
dwf:=wi;
daf:=di;

if ((k1/k1f > 1) or (n=nmax)) and (hits_sensor = false) and
(exceeds_bandwidth = false) and (hits_torquer = false) then
begin
  tctl:=tf;
  vctl:=vcf;
  thetactl:=thetacf;
  dk1dactl:=dk1daf;
  dwctl:=dwf;
  dactl:=daf;
  pctk1ctl:=k1f/k1cc;
  dwdtctl:=dwdtf;
end;

if (exceeds_bandwidth = false) and (hits_sensor or hits_torquer) then
begin
  writeln;
  if hits_sensor then
  begin
    writeln('WARNING: PLATE HAS STRUCK SENSOR ELECTRODE WITHOUT');
    writeln('EXCEEDING CONTROLLER BANDWIDTH');
  end;
  if hits_torquer then
  begin
    writeln('WARNING: PLATE HAS STRUCK TORQUER ELECTRODE WITHOUT');
    writeln('EXCEEDING CONTROLLER BANDWIDTH');
  end;
  readln;
  tctl:=tf;
  vctl:=vcf;
  thetactl:=thetacf;
  dk1dactl:=dk1daf;
  dwctl:=dwf;
  dactl:=daf;
```

```
    pctk1ctl:=k1/k1cc;
    dwdtctl:=dwdtf;
  end;
end;{fail_param}

procedure stress_concentration;
var
  pstatic,
  momf,
  smax,
  kf:double;

begin
  Kf:=1+0.68*sqrt((h-hb)/rtip)-0.31*power((h-hb)/rtip,0.25);
  smax:=Kf*momb0*hb/(2*Iab);
  pctsf:=smax/sigmaf*100;
  momf:=sigmaf/Kf*(2*Iab)/hb;
  vcfail:=vc0*momf/momb0;
  thetacfail:=thetac0*momf/momb0;
  v0f:=v0*sqrt(momf/momb0);
end;

procedure etch_fail;
var
  fst,
  delp,
  ap:double;

begin
  ap:=l[p,x]*l[p,y];
  delp:=2*gamma_h20/h0p;
  fst:=delp*ap;
  mombst:=fst*(lbc+lb[p,x]);
  if mombst > mombf then
  begin
    writeln('FAILURE IN ETCH');
    readln;
    clrscr;
  end;
end;

procedure fracture_analysis;
begin
  frac_param(ndof,nsi02);
  if ndof=3 then etch_fail;
  stress_concentration;
  if fail_sim then
  begin
```

```
    clrscr;
    fail_param(ndof,nsi02);
  end;
  if fati_sim then
  begin
    clrscr;
    fatigue_sim(mac,c,p0,ndof);
  end;
end;

begin
end.
```

```
{$N+}{8087 double precision mode}
unit elec2;
interface
uses
  crt,
  dos,
  init2;

procedure torqueforce(v0:double; var f0:double);
procedure deltak(v0,vexc,vbias:double; var dkt,dks:double);
procedure sensoroutput(v,theta:double; var eoutu,eoutd:double);
procedure control(np:integer; pole:double);
procedure velocity_error(dwdt:double; var verror:double);

implementation
uses
  mains2;
var
  verror_const:double;

procedure torqueforce(v0:double; var f0:double);
var
  a0t,                          {total torquer area [m^2]}
  a1,                           {area of torquer hole 1 [m^2]}
  a2,                           {area of torquer hole 2 [m^2]}
  ff0,                          {force based on a0t [N]}
  ff1,                          {force lost due to a1 [N]}
  ff2,                          {force lost due to a2 2 [n]}
  zeta1,                        {(width of hole 1)/(torquer gap)}
  zeta2:double;                 {(width of hole 2)/(torquer gap)}
  areator:double;

  function ff(v,h,a,zeta:double):double;
  begin
    ff:=e0*a*sqr(v)/(2*pi*sqr(h))*arctan(zeta/2);
  end;

 begin
  a0t:=ltx*lty;
  a1:=l[h1s,x]*l[h1s,y];
  a2:=l[h2s,x]*l[h2s,y];

  zeta1:=l[h1s,x]/ht;
  zeta2:=l[h2s,x]/ht;

  ff0:=e0*a0t*sqr(v0)/(4*sqr(ht));
  ff1:=ff(v0,ht,a1,zeta1);
  ff2:=ff(v0,ht,a2,zeta2);
  f0:=ff0-ff1-ff2;

  {torquer force without fringe effects}
{  areator:=ltx*lty-l[h1s,x]*l[h1s,y]-l[h2s,x]*l[h2s,y];  }
{  f0:=e0*areator*sqr(v0)/(4*sqr(ht));                   }
{  dk:=2*f0/d0                                            }
end;

procedure deltak(v0,vexc,vbias:double; var dkt,dks:double);
  var
    a0t,                          {total torquer area [m^2]}
    a0s,                          {total sensor area [m^2]}
    a1,                           {area of hole 1 [m^2]}
    a2,                           {area of hole 2 [m^2]}
    a3,                           {area of hole 3 [m^2]}
    a4,                           {area of hole 4 [m^2]}
    a5,                           {area of hole 5 [m^2]}
    a6,                           {area of hole 6 [m^2]}
    a7,                           {area of hole 7 [m^2]}
    a8,                           {area of hole 8 [m^2]}
    zeta1,                        { = l[h1s,x]/ht}
    zeta2,                        { = l[h2s,x]/ht}
    zeta3,                        { = l[h3s,y]/h0s}
    zeta4,                        { = l[h4s,y]/h0s}
    zeta5,                        { = l[h5s,y]/h0s}
    zeta6,                        { = l[h6s,y]/h0s}
    zeta7,                        { = l[h7s,y]/h0s}
    zeta8,                        { = l[h8s,y]/h0s}
    dk0t,                         {stiffness correction based on a0t [n/m]}
    dk1,                          {stiffness correction based on a1 [n/m]}
    dk2,                          {stiffness correction based on a2 [n/m]}
    dk0s,                         {stiffness correction based on a0s [n/m]}
    dk3,                          {stiffness correction based on a3 [n/m]}
    dk4,                          {stiffness correction based on a4 [n/m]}
    dk5,                          {stiffness correction based on a5 [n/m]}
    dk6,                          {stiffness correction based on a6 [n/m]}
    dk7,                          {stiffness correction based on a7 [n/m]}
    dk8:double;                   {stiffness correction based on a8 [n/m]}

   function ff(v,h,a,zeta:double):double;
   begin
     ff:=e0*a*sqr(v)/(2*pi*sqr(h))*arctan(zeta/2);
   end;

   function dk(v,h,a,zeta:double):double;
   begin
     dk:=2*ff(v,h,a,zeta)/h+
          e0*sqr(v)*a/(4*pi*power(h,3))*zeta/(1+sqr(zeta/2));
    end;

  begin {deltak}
    a0t:=ltx*lty;
    a0s:=lsx*lsy;
    a1:=l[h1s,x]*l[h1s,y];
    a2:=l[h2s,x]*l[h2s,y];
    a3:=l[h3s,x]*l[h3s,y];
    a4:=l[h4s,x]*l[h4s,y];
    a5:=l[h5s,x]*l[h5s,y];
    a6:=l[h6s,x]*l[h6s,y];
    a7:=l[h7s,x]*l[h7s,y];
    a8:=l[h8s,x]*l[h8s,y];

    zeta1:=l[h1s,x]/ht;
    zeta2:=l[h2s,x]/ht;
    zeta3:=l[h3s,y]/hs;
    zeta4:=l[h4s,y]/hs;
    zeta5:=l[h5s,y]/hs;
    zeta6:=l[h6s,y]/hs;
    zeta7:=l[h7s,y]/hs;
    zeta8:=l[h8s,y]/hs;

    dk0s:=e0*a0s/(2*sqr(hs)*hs)*(sqr(vexc)+2*sqr(vbias));
    dk3:=dk(vexc,hs,a3,zeta3)+2*dk(vbias,hs,a3,zeta3);
    dk4:=dk(vexc,hs,a4,zeta4)+2*dk(vbias,hs,a4,zeta4);
    dk5:=dk(vexc,hs,a5,zeta5)+2*dk(vbias,hs,a5,zeta5);
    dk6:=dk(vexc,hs,a6,zeta6)+2*dk(vbias,hs,a6,zeta6);
    dk7:=dk(vexc,hs,a7,zeta7)+2*dk(vbias,hs,a7,zeta7);
    dk8:=dk(vexc,hs,a8,zeta8)+2*dk(vbias,hs,a8,zeta8);
    dks:=dk0s-dk3-dk4-dk5-dk6-dk7-dk8;

    dk0t:=e0*a0t/(2*sqr(ht)*ht)*(sqr(v0)+sqr(vexc));
    dk1:=dk(v0,ht,a1,zeta1)+dk(vexc,ht,a1,zeta1);
    dk2:=dk(v0,ht,a2,zeta2)+dk(vexc,ht,a2,zeta2);
    dkt:=dk0t-dk1-dk2;
   end; {deltak}

   procedure torqueconstant;
   begin
     kt:=2*f0/v0;
   end;

   procedure sensoroutput(v,theta:double; var eoutu,eoutd:double);

     function cap0(area,gap:double):double;
     begin
  cap0:=e0*area/gap;
end;

procedure sensor_model_1(var delcu,delcd,c0:double);
var
  cu,
  cd,
  gup,                         {airgap for pos. v [m]}
  gdn,                         {airgap for neg. v [m]}
  areasens,                    {sensor plate area [m^2]}
  width,                       {width of sensing area [m]}
  x11,                         {dist. to inner edge of sensor [m]}
  x22:double;                  {dist. to outer edge of sensor [m]}

  function capup(wdth,theta,gap,x1,x2:double):double;
  begin
    capup:=e0*wdth/theta*ln((gap-theta*x1)/(gap-theta*x2));
  end;

  function capdn(wdth,theta,gap,x1,x2:double):double;
  begin
    capdn:=e0*wdth/theta*ln((gap+theta*x2)/(gap+theta*x1));
  end;

begin{model_1}
  gup:=hs-v;
  gdn:=hs+v;

  {compute location of sensor plate wrt pt. c}
  x11:=lbsx-lsx/2;
  x22:=lbsx+lsx/2;

  width:=lsy-l[h3s,y]-l[h4s,y]-l[h5s,y]-l[h6s,y]-l[h7s,y]-l[h8s,y];
  areasens:=lsx*width;

  cu:=capup(width,theta,gup,x11,x22);
  cd:=capdn(width,theta,gdn,x11,x22);
  c0:=cap0(areasens,hs);

  delcu:=cu-c0;
  delcd:=cd-c0;
end;{model_1}

procedure sensor_model_2(var delcu,delcd,c0:double);
var
  areasens,
  gup,
  gdn,
  cu,
  cd,
  c3u,                       {edge capac. pos. v [f] for hole 3}
  c3d,                       {edge capac. pos. v [f] for hole 3}
  c4u,                       {edge capac. pos. v [f] for hole 4}
  c4d,                       {edge capac. neg. v [f] for hole 4}
  c5u,                       {edge capac. pos. v [f] for hole 5}
  c5d,                       {edge capac. pos. v [f] for hole 5}
  c6u,                       {edge capac. pos. v [f] for hole 6}
  c6d,                       {edge capac. pos. v [f] for hole 6}
  c7u,                       {edge capac. pos. v [f] for hole 7}
  c7d,                       {edge capac. neg. v [f] for hole 7}
  c8u,                       {edge capac. pos. v [f] for hole 8}
  c8d,                       {edge capac. pos. v [f] for hole 8}
  c30,                       {nominal capac. v=theta=0 [f] for hole 3}
  c40,                       {nominal capac. v=theta=0 [f] for hole 4}
  c50,                       {nominal capac. v=theta=0 [f] for hole 5}
  c60,                       {nominal capac. v=theta=0 [f] for hole 6}
  c70,                       {nominal capac. v=theta=0 [f] for hole 7}
  c80:double;                {nominal capac. v=theta=0 [f] for hole 8}

  function edjcap(lgth,s,gap:double):double;
  begin
    edjcap:=2*e0/pi*lgth*(s/gap*arctan(s/(2*gap))-
            ln(1+0.25*sqr(s/gap)));
  end;

begin{model_2}
  gup:=hs-(v+theta*lbsx);
  gdn:=hs+(v+theta*lbsx);
  areasens:=lsx*lsy;

  {compute total upward sensor capacitance}
  cu:=cap0(areasens,gup);

  {correct sensor capacitance for decrease due to holes 3,4,5,6,7,8}
  c3u:=edjcap(lsx,l[h3s,y],gup);
  c4u:=edjcap(lsx,l[h4s,y],gup);
  c5u:=edjcap(lsx,l[h5s,y],gup);
  c6u:=edjcap(lsx,l[h6s,y],gup);
  c7u:=edjcap(lsx,l[h7s,y],gup);
  c8u:=edjcap(lsx,l[h8s,y],gup);
  cu:=cu-(c3u+c4u+c5u+c6u+c7u+c8u);

  {compute total downward sensor capacitance}
  cd:=cap0(areasens,gdn);

  {correct sensor capacitance for decrease due to holes 3,4,5,6,7,8}
  c3d:=edjcap(lsx,l[h3s,y],gdn);
  c4d:=edjcap(lsx,l[h4s,y],gdn);
  c5d:=edjcap(lsx,l[h5s,y],gdn);
  c6d:=edjcap(lsx,l[h6s,y],gdn);
  c7d:=edjcap(lsx,l[h7s,y],gdn);
  c8d:=edjcap(lsx,l[h8s,y],gdn);
  cd:=cd-(c3d+c4d+c5d+c6d+c7d+c8d);

  {compute total nominal capacitance}
  c0:=cap0(areasens,hs);

  {compute sensor capacitance decrease due to holes 3,4,5,6,7,8}
  c30:=edjcap(lsx,l[h3s,y],hs);
  c40:=edjcap(lsx,l[h4s,y],hs);
  c50:=edjcap(lsx,l[h5s,y],hs);
  c60:=edjcap(lsx,l[h6s,y],hs);
  c70:=edjcap(lsx,l[h7s,y],hs);
  c80:=edjcap(lsx,l[h8s,y],hs);
  c0:=c0-(c30+c40+c50+c60+c70+c80);

  {compute delta cap.}
  delcu:=cu-c0;
  delcd:=cd-c0;
end; {model_2}

procedure counterweight_cap;
var
  cp0,                        {nomincal cap. to counter weight [f]}
  delcpu,                     {change in cap. "    "      "    up [f]}
  delcpd,                     { "     "  "  "  "       "    dn  " }
  a1u,                        {calculation constants}
  a2u,                        {   "              "   }
  a1d,                        {   "              "   }
  a2d,                        {   "              "   }
  a3,                         {   "              "   }
  a4:double;                  {   "              "   }

  function tan(x:double):double;
  begin
    tan:=sin(x)/cos(x);
  end;

begin{counterweigth}
  if l[p2,z] > hs then
  begin
    a1u:=x3-(hs-vc)*tan(pi/6);
    a2u:=x3-(2*l[p2,z]-hs+vc)*tan(pi/6);
    a1d:=x3-(hs+vc)*tan(pi/6);
```

```
      a2d:=x3-(2*l[p2,z]-hs-vc)*tan(pi/6);
      a3:=x3-(2*l[p2,z]-hs)*tan(pi/6);
      a4:=x3-hs*tan(pi/6);
      if (a1u > 0) and (a2u > 0) and (a1d > 0) and (a2d > 0) and (a3 > 0)
      and (a4 > 0) then
      begin
        delcpu:=e0*l[p1,y]/(2*tan(pi/6))*ln(a1u/a2u*a3/a4);
        delcpd:=e0*l[p1,y]/(2*tan(pi/6))*ln(a1d/a2d*a3/a4);
        cp0:=e0*l[p1,y]/(2*tan(pi/6))*ln(a4/a3);
      end
      else
      begin
        delcpu:=0;
        delcpd:=0;
        cp0:=0;
      end;
    end
    else
    begin
      delcpu:=0;
      delcpd:=0;
      cp0:=0;
    end;
  end; {counterweigth}

  procedure sensconst(var delcu,delcd:double);
  const
    rem=26;              {transistor dynamic ammiter resistance at 25 degC}
  var
    RL,                  {load resistance to ground, ohms}
    Re,                  {gain adjust, ohms}
    kdmod,
    kamp,
    kfs:double;

  begin
    RL:=3000;                   {3K nominal value choosen}
    Re:=1000;                   {1K   "        "      "    }
    kdmod:=0.637*RL/(Re+2*rem);
    kfs:=20;
    kamp:=100 {old post amp: kamp=17};
    eoutu:=-1*(delcu)/cfb*vexc*kdmod*kamp*kfs;
    eoutd:=-1*(delcd)/cfb*vexc*kdmod*kamp*kfs;
  end; {sensconst}

begin {sensor output}
  {sensor_model_1(delcu,delcd,c0);}
  sensor_model_2(delcu,delcd,c0);
  sensconst(delcu,delcd);
end; {sensor output}

procedure velocity_error(dwdt:double; var verror:double);
begin
  verror:=dwdt/verror_const;
end;

procedure control(np:integer; pole:double);
const
  zeta=0.6;                 {damping ratio of closed loop system}
var
  phaseplant,               {phase of plant at wc, rad}
  phasemargin,              {required c.l. phase margin, rad}
  phasemax,                 {required phase lead of compensator, rad}
  magcomp,                  {magnitude of compensator pole and zero at wc}
  magsys,                   {magnitude of system at wc}
  eta,                      {ratio of b/a}
  c1,                       {calculation const}
  wn,                       {natural frequency of c.l. system, rad/s}
  wc:double;                {cross over frequency, rad/s}

begin {control}
  {compute system and plant gains}
  pole:=abs(pole);
  kvco:=2*pi*1.0e3;                   {vco gain: wavetek, rad/(s*volt)}
  kd:=5/(2*pi);                       {gain of phase dectector, volts/rad}
  gp:=2*kvco*kd;                      {system gain}
  kp:=2/pole;                         {dc plant gain}
                                      {compute crossover freq. and phase angles}
  c1:=sqrt(2-4*sqr(zeta)+4*power(zeta,4));
  wn:=wservo/sqrt(1-2*sqr(zeta)+c1);
  wc:=wn*sqrt(1-2*sqr(zeta));
  phaseplant:=-pi/2-arctan(wc/pole);
  phasemargin:=100*zeta*pi/180;
  phasemax:=phasemargin-(pi+phaseplant);

  {compute type 1 compensator pole and zero}
  eta:=(1+sin(phasemax))/(1-sin(phasemax));
  a:=round(wc/sqrt(eta));
  b:=round(sqr(wc)/a);

  {compute compensator gain}
  magcomp:=sqrt((sqr(wc/a)+1)/(sqr(wc/b)+1))/wc;
  magsys:=gp/sqrt(sqr(wc)+sqr(pole));
  kf1:=1/(magcomp*magsys);

  {determine system performance}
  verror_const:=kf1*gp;
  verrormax:=pi/6;
  dwdtmax:=verrormax*verror_const;
end;{control}

begin
end.
```

```
{$N+}{8087 mode compiler directive}
unit menu2;
interface
uses
  crt,
  dos,
  out2,
  init2;

  procedure run_menu;
  procedure change_output;

implementation
const
  sk10 = '          ';
var
  quitmenu:boolean;
  faildat,
  outfile:text;
  filename:string[12];
  prn_dim,
  prn_dyn,
  prn_ctl,
  prn_elec,
  prn_frac,
  print_short,
  print_long:boolean;

procedure change_output;
var
  quit_change,
  stop:boolean;
  num,
  format:char;

begin{change output}
  quit_change:=false;
  repeat
    clrscr;
    writeln('Set Output Format');
    write('S - short table format',sk10);
    if print_short then writeln('yes') else writeln('no');
    write('L - long table format ',sk10);
    if print_long then writeln('yes') else writeln('no');
    writeln('C - change long table');
    writeln('Q - quit menu');
    begin
      repeat
    format:=readkey;
   until (format in ['s','l','c','q'])
  end;
  clrscr;
  case format of
  's':begin
     print_short:=not print_short;
    end;
  'l':begin
     print_long:=not print_long;
    end;
  'c':begin
     stop:=false;
     repeat
     begin
      writeln('Select output of long table');
      write('1 - Dimensions           ',sk10);
      if prn_dim then writeln('yes') else writeln('no');
      write('2 - Dynamics             ',sk10);
      if prn_dyn then writeln('yes') else writeln('no');
      write('3 - Electrical           ',sk10);
      if prn_elec then writeln('yes') else writeln('no');
      write('4 - Control              ',sk10);
      if prn_ctl then writeln('yes') else writeln('no');
      write('5 - Fracture paramenters ',sk10);
      if prn_frac then writeln('yes') else writeln('no');
      writeln('6 - Quit selection');
      repeat
       num:=readkey;
      until num in ['1'..'6'];
      case num of
       '1':prn_dim:=not prn_dim;
       '2':prn_dyn:=not prn_dyn;
       '3':prn_elec:=not prn_elec;
       '4':prn_ctl:=not prn_ctl;
       '5':if ndof=3 then prn_frac:=not prn_frac;
       '6':stop:=true;
      end;{case num}
      clrscr;
     end;
     until stop;
    end; {case of 'c'}
  'q':begin
     quit_change:=true;
    end;
  end; {case table of}
 until quit_change;
end; {change output}
```

```
procedure print_output(var outfile:text);
begin
  clrscr;
  print_heading(outfile);
  if print_short then print_short_table(outfile);
  if print_long then
  begin
    if prn_dim then print_dimensions(outfile);
    if prn_dyn then print_dynamics(outfile);
    if prn_elec then print_electrical(outfile);
    if prn_ctl then print_control(outfile);
    if prn_frac then print_fracture(outfile);
    if fail_sim then print_simulation(outfile);
  end;
  if option = 'd' then print_comments(outfile);
end;

procedure run_menu;
begin {run_menu}
  quitmenu:=false;
  if ndof = 2 then prn_frac:=false;
  while not quitmenu do
  begin
    clrscr;
    writeln('Choose output option.');
    writeln('S = screen');
    writeln('P = printer');
    writeln('D = disk');
    writeln('Q = quit program');
    writeln('R = rerun with new input values');
    writeln('N = generate new input file');
    writeln('F = rerun with new input file');
    writeln('C = change format');
    begin
      repeat
        option:=readkey;
      until option in ['s','p','d','q','r','n','c','f'];
    end;
    case option of
      's':begin
            assign(outfile,'CON');
            rewrite(outfile);
            print_output(outfile);
            close(outfile);
          end;
      'p':begin
            assign(outfile,'PRN');
            rewrite(outfile);
            print_output(outfile);
            close(outfile);
          end;
      'd':begin
            writeln('Specify output filename.');
            repeat
              readln(filename);
            until filename <> '';
            assign(outfile,filename);
            rewrite(outfile);
            print_output(outfile);
            close(outfile);
          end;
      'q':begin
            quitmenu:=true;
            rerun:=false;
          end;
      'r':begin
            quitmenu:=true;
            reinitialize;
          end;
      'n':begin
            writeln('Specify new input filename.');
            repeat
              readln(filename);
            until filename <> '';
            assign(outfile,filename);
            rewrite(outfile);
            print_var(outfile);
            writeln(outfile);
            writeln(outfile,'New file created on:');
            print_timeanddate(outfile);
            print_comments(outfile);
            close(outfile);
          end;
      'c':begin
            change_output;
          end;
      'f':begin
            quitmenu:=true;
            indata;
            initialize;
          end;
    end;{case option of}
  end; {while not quitmenu do}
end; {run_menu}
```

```
begin
  print_short:=true;
  print_long:=false;
  prn_dim:= true;
  prn_dyn:=true;
  prn_elec:=true;
  prn_ctl:=true;
  if ndof = 3 then prn_frac:=true;
  if ndof = 2 then prn_frac:=false;
end.
```

```
{$N+}{8087 mode compiler directive}
unit out2;
interface
uses
  crt,
  dos,
  init2,
  dyn2,
  elec2;

  procedure print_short_table(var outfile:text);
  procedure print_dimensions(var outfile:text);
  procedure print_dynamics(var outfile:text);
  procedure print_electrical(var outfile:text);
  procedure print_control(var outfile:text);
  procedure print_fracture(var outfile:text);
  procedure print_heading(var outfile:text);
  procedure print_comments(var outfile:text);
  procedure print_var(var outfile:text);
  procedure print_timeanddate(var outfile:text);
  procedure print_simulation(var outfile:text);

implementation
const
  s10='          ';
  s5='     ';

var
  i,j:integer;

procedure stopscroll;
var
  wait:char;
begin
  if option='s' then
  begin
    repeat
      wait:=readkey;
    until wait<>'';
    clrscr;
  end;
end; {stopscroll}

procedure skipline(var outfile:text; n:integer);
var
  i:integer;
begin
  if option='p' then
  begin
    for i:=1 to n do
    writeln(outfile);
  end;
end;

procedure print_dimensions(var outfile:text);
begin
  writeln(outfile,
  '                              DIMENSIONS');
  writeln(outfile,
  '---------------------------------------------------------------------');
  writeln(outfile,
  '                               BEAM');
  writeln(outfile,s5,'Beam Length  ',s5,'lac  ',s5,'[mic]',s5,lac/clgth:15:2);
  writeln(outfile,s5,'Beam Width   ',s5,'b0   ',s5,'[mic]',s5,b0/clgth:15:2);
  writeln(outfile,s5,'Beam Thick.  ',s5,'h    ',s5,'[mic]',s5,h/clgth:15:2);
  writeln(outfile);
  writeln(outfile,
  '                               V-NOTCH');
  writeln(outfile,s5,'V-notch loc. ',s5,'lab  ',s5,'[mic]',s5,lab/clgth:15:2);
  writeln(outfile,s5,'V-notch th.  ',s5,'hb   ',s5,'[mic]',s5,hb/clgth:15:2);
  writeln(outfile,s5,'V-notch wdth ',s5,'bb   ',s5,'[mic]',s5,bb/clgth:15:2);
  writeln(outfile,s5,'Crack lgth   ',s5,'da   ',s5,'[mic]',s5,da/clgth:15:4);
  writeln(outfile,s5,'tip radius   ',s5,'rtip
',s5,'[mic]',s5,rtip/clgth:15:4);
  writeln(outfile);
  writeln(outfile,
  '                              SILICON PLATE');
  writeln(outfile,s5,'Plate Length ',s5,'l[p,x]',s5,'[mic]
',s5,l[p,x]/clgth:15:2);
  writeln(outfile,s5,'Plate width  ',s5,'l[p,y]',s5,'[mic]
',s5,l[p,y]/clgth:15:2);
  writeln(outfile,s5,'Plate thick  ',s5,'l[p,z]',s5,'[mic]
',s5,l[p,z]/clgth:15:2);
  writeln(outfile,s5,'CG x
',s5,'lb[p,x]',s5,'[mic]',s5,lb[p,x]/clgth:15:2);
  writeln(outfile,s5,'CG y
',s5,'lb[p,y]',s5,'[mic]',s5,lb[p,y]/clgth:15:2);
  writeln(outfile,s5,'CG z
',s5,'lb[p,z]',s5,'[mic]',s5,lb[p,z]/clgth:15:2);
  stopscroll;
  writeln(outfile,
  '                              Plate 1');
  writeln(outfile,s5,'length        ',s5,'l[p1,x]',s5,'[mic]
',s5,l[p1,x]/clgth:15:2);
  writeln(outfile,s5,'width         ',s5,'l[p1,y]',s5,'[mic]
',s5,l[p1,y]/clgth:15:2);
```

```
  writeln(outfile,s5,'thick        ',s5,'l[p1,z]',s5,'[mic]
',s5,l[p1,z]/clgth:15:2);
  writeln(outfile,s5,'CG x
',s5,'lb[p1,x]',s5,'[mic]',s5,lb[p1,x]/clgth:15:2);
  writeln(outfile,s5,'CG y
',s5,'lb[p1,y]',s5,'[mic]',s5,lb[p1,y]/clgth:15:2);
  writeln(outfile,s5,'CG z
',s5,'lb[p1,z]',s5,'[mic]',s5,lb[p1,z]/clgth:15:2);
  writeln(outfile);
  writeln(outfile,
  '                            Plate 2');
  writeln(outfile,s5,'length       ',s5,'l[p2,x]',s5,'[mic]
',s5,l[p2,x]/clgth:15:2);
  writeln(outfile,s5,'width        ',s5,'l[p2,y]',s5,'[mic]
',s5,l[p2,y]/clgth:15:2);
  writeln(outfile,s5,'thick        ',s5,'l[p2,z]',s5,'[mic]
',s5,l[p2,z]/clgth:15:2);
  writeln(outfile,s5,'CG x
',s5,'lb[p2,x]',s5,'[mic]',s5,lb[p2,x]/clgth:15:2);
  writeln(outfile,s5,'CG y
',s5,'lb[p2,y]',s5,'[mic]',s5,lb[p2,y]/clgth:15:2);
  writeln(outfile,s5,'CG z
',s5,'lb[p2,z]',s5,'[mic]',s5,lb[p2,z]/clgth:15:2);
  writeln(outfile);
  stopscroll;
  writeln(outfile,
  '                            Airgap');
  writeln(outfile,s5,'Sensor gap   ',s5,'d0s
',s5,'[mic]',s5,d0s/clgth:15:2);
  writeln(outfile,s5,'Torquer gap  ',s5,'d0t
',s5,'[mic]',s5,d0t/clgth:15:2);
  writeln(outfile,s5,'Min sense gap',s5,'d0smin
',s5,'[mic]',s5,d0smin/clgth:15:2);
  writeln(outfile,s5,'Min tor. gap ',s5,'d0tmin
',s5,'[mic]',s5,d0tmin/clgth:15:2);
  writeln(outfile,s5,'Nom. Tor gap ',s5,'h0t
',s5,'[mic]',s5,h0t/clgth:15:2);
  writeln(outfile,s5,'Nom. Sen gap ',s5,'h0s
',s5,'[mic]',s5,h0s/clgth:15:2);
  writeln(outfile,s5,'Act. Tor gap ',s5,'ht    ',s5,'[mic]',s5,ht/clgth:15:2);
  writeln(outfile,s5,'Act. Sen gap ',s5,'hs    ',s5,'[mic]',s5,hs/clgth:15:2);
  writeln(outfile,s5,'Plate gap    ',s5,'h0p
',s5,'[mic]',s5,h0p/clgth:15:2);
  writeln(outfile);
  writeln(outfile,
  '                             TORQUER');
  writeln(outfile,s5,'length       ',s5,'ltx ',s5,'[mic]',s5,ltx/clgth:15:2);
  writeln(outfile,s5,'width        ',s5,'lty ',s5,'[mic]',s5,lty/clgth:15:2);
  writeln(outfile,s5,'             ',s5,'lbtx',s5,'[mic]',s5,lbtx/clgth:15:2);
  writeln(outfile);
  writeln(outfile,
  '                                SENSOR');
  writeln(outfile,s5,'length       ',s5,'lsx ',s5,'[mic]',s5,lsx/clgth:15:2);
  writeln(outfile,s5,'width        ',s5,'lsy ',s5,'[mic]',s5,lsy/clgth:15:2);
  writeln(outfile,s5,'             ',s5,'lbsx',s5,'[mic]',s5,lbsx/clgth:15:2);
  writeln(outfile);
  stopscroll;
  writeln(outfile,
  '                              HOLE DIMENSIONS');
  writeln(outfile,s5,'Hole 1');
  writeln(outfile,s5,'  Lgth       ',s5,'l[h1s,x]
',s5,'[mic]',s5,l[h1s,x]/clgth:15:2);
  writeln(outfile,s5,'  wdth       ',s5,'l[h1s,y]
',s5,'[mic]',s5,l[h1s,y]/clgth:15:2);
  writeln(outfile,s5,'
',s5,'lb[h1s,x]',s5,'[mic]',s5,lb[h1s,x]/clgth:15:2);
  writeln(outfile,s5,'
',s5,'lb[h1s,y]',s5,'[mic]',s5,lb[h1s,y]/clgth:15:2);
  writeln(outfile,s5,'Hole 2');
  writeln(outfile,s5,'  Lgth       ',s5,'l[h2s,x]
',s5,'[mic]',s5,l[h2s,x]/clgth:15:2);
  writeln(outfile,s5,'  wdth       ',s5,'l[h2s,y]
',s5,'[mic]',s5,l[h2s,y]/clgth:15:2);
  writeln(outfile,s5,'
',s5,'lb[h2s,x]',s5,'[mic]',s5,lb[h2s,x]/clgth:15:2);
  writeln(outfile,s5,'
',s5,'lb[h2s,y]',s5,'[mic]',s5,lb[h2s,y]/clgth:15:2);
  writeln(outfile,s5,'Hole 3');
  writeln(outfile,s5,'  Lgth       ',s5,'l[h3s,x]
',s5,'[mic]',s5,l[h3s,x]/clgth:15:2);
  writeln(outfile,s5,'  wdth       ',s5,'l[h3s,y]
',s5,'[mic]',s5,l[h3s,y]/clgth:15:2);
  writeln(outfile,s5,'
',s5,'lb[h3s,x]',s5,'[mic]',s5,lb[h3s,x]/clgth:15:2);
  writeln(outfile,s5,'
',s5,'lb[h3s,y]',s5,'[mic]',s5,lb[h3s,y]/clgth:15:2);
  writeln(outfile,s5,'Hole 4');
  writeln(outfile,s5,'  Lgth       ',s5,'l[h4s,x]
',s5,'[mic]',s5,l[h4s,x]/clgth:15:2);
  writeln(outfile,s5,'  wdth       ',s5,'l[h4s,y]
',s5,'[mic]',s5,l[h4s,y]/clgth:15:2);
  writeln(outfile,s5,'
',s5,'lb[h4s,x]',s5,'[mic]',s5,lb[h4s,x]/clgth:15:2);
  writeln(outfile,s5,'
',s5,'lb[h4s,y]',s5,'[mic]',s5,lb[h4s,y]/clgth:15:2);
  stopscroll;
  writeln(outfile,s5,'Hole 5');
  writeln(outfile,s5,'  Lgth        ',s5,'l[h5s,x]
',s5,'[mic]',s5,l[h5s,x]/clgth:15:2);
  writeln(outfile,s5,'  wdth        ',s5,'l[h5s,y]
',s5,'[mic]',s5,l[h5s,y]/clgth:15:2);
  writeln(outfile,s5,'
',s5,'lb[h5s,x]',s5,'[mic]',s5,lb[h5s,x]/clgth:15:2);
  writeln(outfile,s5,'
',s5,'lb[h5s,y]',s5,'[mic]',s5,lb[h5s,y]/clgth:15:2);
  writeln(outfile);
  skipline(outfile,5);
  writeln(outfile,s5,'Hole 6');
  writeln(outfile,s5,'  Lgth        ',s5,'l[h6s,x]
',s5,'[mic]',s5,l[h6s,x]/clgth:15:2);
  writeln(outfile,s5,'  wdth        ',s5,'l[h6s,y]
',s5,'[mic]',s5,l[h6s,y]/clgth:15:2);
  writeln(outfile,s5,'
',s5,'lb[h6s,x]',s5,'[mic]',s5,lb[h6s,x]/clgth:15:2);
  writeln(outfile,s5,'
',s5,'lb[h6s,y]',s5,'[mic]',s5,lb[h6s,y]/clgth:15:2);
  writeln(outfile);
  writeln(outfile,s5,'Hole 7');
  writeln(outfile,s5,'  Lgth        ',s5,'l[h7s,x]
',s5,'[mic]',s5,l[h7s,x]/clgth:15:2);
  writeln(outfile,s5,'  wdth        ',s5,'l[h7s,y]
',s5,'[mic]',s5,l[h7s,y]/clgth:15:2);
  writeln(outfile,s5,'
',s5,'lb[h7s,x]',s5,'[mic]',s5,lb[h7s,x]/clgth:15:2);
  writeln(outfile,s5,'
',s5,'lb[h7s,y]',s5,'[mic]',s5,lb[h7s,y]/clgth:15:2);
  writeln(outfile);
  writeln(outfile,s5,'Hole 8');
  writeln(outfile,s5,'  Lgth        ',s5,'l[h8s,x]
',s5,'[mic]',s5,l[h8s,x]/clgth:15:2);
  writeln(outfile,s5,'  wdth        ',s5,'l[h8s,y]
',s5,'[mic]',s5,l[h8s,y]/clgth:15:2);
  writeln(outfile,s5,'
',s5,'lb[h8s,x]',s5,'[mic]',s5,lb[h8s,x]/clgth:15:2);
  writeln(outfile,s5,'
',s5,'lb[h8s,y]',s5,'[mic]',s5,lb[h8s,y]/clgth:15:2);
  writeln(outfile);
  stopscroll;
end; {print_dimensions}

procedure prn_mass_inertia(var outfile:text);
begin
  writeln(outfile,'Mass');
  writeln(outfile,
    s5,'Si plate mass  ',s5,'m[p]   ',s5,'[kg]           ',s5,m[p]:15);
    writeln(outfile,
    s5,'Au plate mass  ',s5,'m[p1]  ',s5,'[kg]           ',s5,m[p1]:15);
    writeln(outfile,
    s5,'Au bar mass    ',s5,'m[p2]  ',s5,'[kg]           ',s5,m[p2]:15);
    writeln(outfile,
    s5,'beam mass      ',s5,'mb     ',s5,'[kg]           ',s5,mb*lac:15);
    writeln(outfile,
    s5,'plate mass     ',s5,'mt     ',s5,'[kg]           ',s5,mt:15);
    writeln(outfile,
    s5,'End Plate CG   ',s5,'lcg[x]   ',s5,'[mic]          ',lcg[x]/clgth:15:2);
    writeln(outfile,
    s5,'               ',s5,'lcg[y]   ',s5,'[mic]          ',lcg[y]/clgth:15:2);
    writeln(outfile,
    s5,'               ',s5,'lcg[z]   ',s5,'[mic]          ',lcg[z]/clgth:15:2);
    writeln(outfile,'Inertia');
    writeln(outfile,
    s5,'x-axis         ',s5,'Ixx    ',s5,'[kg*m^2]     ',s5,Ip[x,x]:13);
    writeln(outfile,
    s5,'y-axis         ',s5,'Iyy    ',s5,'[kg*m^2]     ',s5,Ip[y,y]:15);
    writeln(outfile,
    s5,'z-axis         ',s5,'Izz    ',s5,'[kg*m^2]     ',s5,Ip[z,z]:15);
    writeln(outfile);
  end; {prn_mass_inertia}

  procedure print_dynamics(var outfile:text);
  begin {prndynamics}
    writeln(outfile,
    '
                                   DYNAMICS');
    writeln(outfile,
    '------------------------------------------------------------------------');
    writeln(outfile,
    'Mode shapes and frequencies: ndof = ',ndof:1,s5,'Frequency [hz]':20);
    writeln(outfile,'Air');
    for j:=1 to nsys do
    begin
      write(outfile,s5,'mode ',4-j:1);
      for i:=1 to nsys do
      write(outfile,phinorm[i,j]:10:4);
      writeln(outfile,s10,omega[j]/(2*pi):10:2);
    end;
    writeln(outfile);
    writeln(outfile,'Vacuum');
    for j:=1 to nsys do
    begin
      write(outfile,s5,'mode ',4-j:1);
      for i:=1 to nsys do
      write(outfile,phinormvac[i,j]:10:4);
```

```
   writeln(outfile,s10,omegavac[j]/(2*pi):10:2);
 end;
 writeln(outfile);
 writeln(outfile,
 s5,'real pole      ',s5,'repole      ',s5,'[hz]     ',s5,repole/(2*pi):15:3);
 writeln(outfile,
 s5,'imag pole      ',s5,'impole      ',s5,'[hz]     ',s5,impole/(2*pi):15:3);
 writeln(outfile,
 s5,'Tip Displ at c ',s5,'vc          ',s5,'[mic]    ',s5,vc/clgth:15:3);
 writeln(outfile,
 s5,'Tip Rot. at c  ',s5,'thetac      ',s5,'[deg]    ',s5,thetac*180/pi:15:3);
 writeln(outfile,
 s5,'Static Tip Disp',s5,'vcstatic    ',s5,'[mic]
',s5,vcstatic/clgth:15:3);
 writeln(outfile,
 s5,'Static Tip Rot ',s5,'thetacstatic',s5,'[deg]
',s5,thetacstatic*180/pi:15:3);
 writeln(outfile,
 s5,'tors. freq     ',s5,'wxx         ',s5,'[hz]     ',s5,wxx/(2*pi):15:2);
 stopscroll;
 writeln(outfile,
 s5,'mode 1 z-axis  ',s5,'wzz         ',s5,'[hz]     ',s5,wzz/(2*pi):15:2);
 writeln(outfile,
 s5,'plate freq     ',s5,'wp          ',s5,'[hz]     ',s5,wp/(2*pi):15:2);
 writeln(outfile,
 s5,'bridge freq    ',s5,'wbr         ',s5,'[hz]     ',s5,wbr/(2*pi):15:2);
 writeln(outfile);
 writeln(outfile,'Vacuum');
 writeln(outfile,
 s5,'real pole      ',s5,'repolevac   ',s5,'[hz]
',s5,repolevac/(2*pi):15:3);
 writeln(outfile,
 s5,'imag pole      ',s5,'impolevac   ',s5,'[hz]
',s5,impolevac/(2*pi):15:3);
 writeln(outfile);
 writeln(outfile,
 s5,'Static beam stiffness x',s5,'klc[x]':10,'[n/m]':10,s5,klc[x]:15);
 writeln(outfile,
 s5,'Static beam stiffness y',s5,'klc[y]':10,'[n/m]':10,s5,klc[y]:15);
 writeln(outfile,
 s5,'Static beam stiffness z',s5,'klc[z]':10,'[n/m]':10,s5,klc[z]:15);
 writeln(outfile,
 s5,'Bridge stiffness z     ',s5,'kbr':10,'[n/m]':10,s5,kbr:15);
 writeln(outfile,
 s5,'Plate/beam stiff. ratio',s5,'sr':10,'[ ]':10,s5,sr:15:2);
 writeln(outfile,
 s5,'Static bridge disp     ',s5,'vbr':10,'[mic]':10,s5,vbr/1e-6:15:4);
 stopscroll;
```

```
  prn_mass_inertia(outfile);
  stopscroll;
end; {print_dynamics}

procedure print_simulation(var outfile:text);
begin
  writeln(outfile,'Fracture parameters at controller bandwidth');
  writeln(outfile,
  s5,'displ at c     ',s5,'vctl      ',s5,'[mic]       ',s5,vctl/clgth:15:3);
  writeln(outfile,
  s5,'rot at c       ',s5,'thetactl  ',s5,'[deg]
',s5,thetactl*180/pi:15:3);
  writeln(outfile,
  s5,'crack lgth     ',s5,'dactl     ',s5,'[mic]       ',s5,dactl/clgth:15:3);
  writeln(outfile,
  s5,'freq shift     ',s5,'dwctl     ',s5,'[hz]
',s5,dwctl/(2*pi):15:3);
  writeln(outfile,
  s5,'freq rate      ',s5,'dwdtctl   ',s5,'[hz/s]
',s5,dwdtctl/(2*pi):15:3);
  writeln(outfile,
  s5,'time to dactl  ',s5,'tctl      ',s5,'[hrs]       ',s5,tctl:15:3);
  writeln(outfile,
  s5,'% klclll       ',s5,'pctklctl  ',s5,'[ ]         ',s5,pctklctl:15:3);
  writeln(outfile,
  s5,'crack stability',s5,'dkldactl  ',s5,'[n/m^(5/2)] ',s5,dkldactl:10:4);
  writeln(outfile);
  writeln(outfile,'Fracture Paramaters at failure');
  writeln(outfile,
  s5,'displ at c     ',s5,'vcf       ',s5,'[mic]       ',s5,vcf/clgth:15:3);
  writeln(outfile,
  s5,'rot at c       ',s5,'thetacf   ',s5,'[deg]
',s5,thetacf*180/pi:15:3);
  writeln(outfile,
  s5,'crack lgth     ',s5,'daf       ',s5,'[mic]       ',s5,daf/clgth:15:3);
  writeln(outfile,
  s5,'freq shift     ',s5,'dwf       ',s5,'[hz]        ',s5,dwf/(2*pi):15:3);
  writeln(outfile,
  s5,'freq rate      ',s5,'dwdtf     ',s5,'[hz/s]
',s5,dwdtf/(2*pi):15:3);
  writeln(outfile,
  s5,'velocity error ',s5,'verrorf   ',s5,'[deg]
',s5,verrorf*180/pi:15:3);
  writeln(outfile,
  s5,'time to daf    ',s5,'tf        ',s5,'[hrs]       ',s5,tf:15:3);
  writeln(outfile,
  s5,'% klclll       ',s5,'pctklcf   ',s5,'[ ]         ',s5,pctklcf:15:3);
  writeln(outfile,
  s5,'crack stability',s5,'dk1daf    ',s5,'[n/m^(5/2)] ',s5,dk1daf:10:4);
  stopscroll;
end;

procedure print_fracture(var outfile:text);
begin
  writeln(outfile,
  '                        CRACK COMPLIANCE PARAMETERS');
  writeln(outfile,
  '----------------------------------------------------------------------');
  writeln(outfile,
  s5,'Kies           ',s5,'c_kies':10,'[n/m]':10,s5,c_kies:15);
  writeln(outfile,
  s5,'Murakami       ',s5,'c_mura':10,'[n/m]':10,s5,c_mura:15);
  writeln(outfile,
  s5,'Paris          ',s5,'c_paris':10,'[n/m]':10,s5,c_paris:15);
  writeln(outfile,
  s5,'Deep crack     ',s5,'c_dc':10,'[n/m]':10,s5,c_dc:15);
  writeln(outfile,
  s5,'Elliptical     ',s5,'c_ellip':10,'[n/m]':10,s5,c_ellip:15);
  writeln(outfile);
  writeln(outfile,
  '                           FRACTURE PARAMETERS');
  writeln(outfile,
  '----------------------------------------------------------------------');
  writeln(outfile,'Stress Concentration Parameters');
  writeln(outfile,
  s5,'V-notch ratio  ',s5,'hb/h      ',s5,'[ ]         ',s5,hb/h:15:3);
  writeln(outfile,
  s5,'Notch radius   ',s5,'rtip      ',s5,'[mic]       ',s5,rtip/clgth:15:4);
  writeln(outfile,
  s5,'Failure Displ. ',s5,'vcfail    ',s5,'[mic]
',s5,vcfail/clgth:15:3);
  writeln(outfile,
  s5,'Failure Rot    ',s5,'thetacfail',s5,'[def]
',s5,thetacfail*180/pi:15:3);
  writeln(outfile,
  s5,'% frac stress  ',s5,'pctsf     ',s5,'[ ]         ',s5,pctsf:15:3);
  writeln(outfile,
  s5,'Fail voltage   ',s5,'v0f       ',s5,'[v]         ',s5,v0f:15:3);
  writeln(outfile);
  stopscroll;
  writeln(outfile,'Stress Intensity Parameters');
  writeln(outfile,
  s5,'K1c            ',s5,'k1c111    ',s5,'[n/m^(3/2)] ',s5,k1c111:15);
  writeln(outfile,
  s5,'initial k1     ',s5,'k1i       ',s5,'[n/m^(3/2)] ',s5,k1i:15);
  writeln(outfile,
  s5,'final kl       ',s5,'klf       ',s5,'[n/m^(3/2)] ',s5,klf:15);
  writeln(outfile,
  s5,'inst. kl       ',s5,'kl        ',s5,'[n/m^(3/2)] ',s5,kl:15);
  writeln(outfile,
  s5,'% Klci         ',s5,'pctklci   ',s5,'[ ]         ',s5,100*pctklci:15:2);
  writeln(outfile,
  s5,'% Klc          ',s5,'pctklc    ',s5,'[ ]         ',s5,100*pctklc:15:2);
  writeln(outfile,
  s5,'% Klcf         ',s5,'pctklcf   ',s5,'[ ]         ',s5,100*pctklcf:15:2);
  writeln(outfile,
  s5,'kaellip        ',s5,'kaellip   ',s5,'[1/m^(5/2) ]',s5,kaellip:15);
  writeln(outfile,
  s5,'Crack lgth     ',s5,'da        ',s5,'[mic]       ',s5,da/clgth:15:4);
  writeln(outfile,
  s5,'Crack Stability',s5,'dkl/da    ',s5,'[n/m^(5/2)] ',s5,dklda:15);
  writeln(outfile,
  s5,'freq shift     ',s5,'dw        ',s5,'[hz]        ',s5,dw/(2*pi):15:4);
  writeln(outfile,
  s5,'freq rate      ',s5,'dwdt      ',s5,'[hz/s^2]    ',s5,dwdt/(2*pi):15);
  writeln(outfile,
  s5,'Time to failure',s5,'tf        ',s5,'[hrs]       ',s5,tf:15:2);
  writeln(outfile,
  s5,'Time to da     ',s5,'ta        ',s5,'[hrs]       ',s5,ta:15:2);
  writeln(outfile,
  s5,'Crack velocity ',s5,'vsi02     ',s5,'[m/s]       ',s5,vsi02:15);
  writeln(outfile,
  s5,'Fatigue const  ',s5,'nsi02     ',s5,'[ ]         ',s5,nsi02:15:2);
  writeln(outfile,
  s5,'Frac Voltage   ',s5,'v0c       ',s5,'[v]         ',s5,v0c:15:3);
  writeln(outfile,
  s5,'nano clear     ',s5,'Icl       ',s5,'[mic]       ',s5,Icl/clgth:15:3);
  writeln(outfile,
  s5,'nano load      ',s5,'Pind      ',s5,'[mN]        ',s5,Pind*1e3:15:3);
  writeln(outfile,
  s5,'moment at b    ',s5,'momb      ',s5,'[n*m]       ',s5,momb:15);
  writeln(outfile,
  s5,'fail mom at b  ',s5,'mombf     ',s5,'[n*m]       ',s5,mombf:15);
  writeln(outfile,
  s5,'st mom at b    ',s5,'mombst    ',s5,'[n*m]       ',s5,mombst:15);
  writeln(outfile,
  s5,'etch fail ratio',s5,'mst/mf    ',s5,'[ ]
',s5,mombst/mombf*100:15:2);
  writeln(outfile);
  stopscroll;
end; {print_fracture}

procedure print_electrical(var outfile:text);
begin
```

```
  writeln(outfile,'                         ELECTRICAL');
  writeln(outfile,
  '----------------------------------------------------------------------');
  writeln(outfile,
  s5,'Sensor Volt dn ',s5,'eoutd  ',s5,'[v]            ',s5,eoutd:15:3);
  writeln(outfile,
  s5,'Sensor Volt up ',s5,'eoutu  ',s5,'[v]            ',s5,eoutu:15:3);
  writeln(outfile,
  s5,'Delta Cap up   ',s5,'delcu  ',s5,'[f]            ',s5,delcu:15);
  writeln(outfile,
  s5,'Delta Cap dn   ',s5,'delcd  ',s5,'[f]            ',s5,delcd:15);
  writeln(outfile,
  s5,'Nominal Cap    ',s5,'c0     ',s5,'[f]            ',s5,c0:15);
  writeln(outfile,
  s5,'Feedback Cap   ',s5,'cfb    ',s5,'[f]            ',s5,cfb:15);
  writeln(outfile,
  s5,'Torquer Voltage',s5,'v0     ',s5,'[v]            ',s5,v0:15:3);
  writeln(outfile,
  s5,'Snap  Voltage ',s5,'vsnap  ',s5,'[v]            ',s5,vsnap:15:3);
  writeln(outfile,
  s5,'Brk dwn Voltage',s5,'vbd    ',s5,'[v]            ',s5,vbd:15:3);
  writeln(outfile,
  s5,'max bridge temp',s5,'Tmax   ',s5,'[deg C]        ',s5,Tmax:15:3);
  writeln(outfile,
  s5,'sinter temp    ',s5,'Tsin   ',s5,'[deg C]        ',s5,Tsin:15);
  writeln(outfile,
  s5,'TC bond temp   ',s5,'Twb    ',s5,'[deg C]        ',s5,Twb:15);
  writeln(outfile);
  stopscroll;
end; {print_electrical}

procedure print_control(var outfile:text);
begin
  writeln(outfile,'                         CONTROL');
  writeln(outfile,
  '---------------------------------------------------------------------');
  writeln(outfile,
  s5,'vco gain       ',s5,'kvco    ',s5,'[rad/(s*v)]',s5,kvco:15:3);
  writeln(outfile,
  s5,'phase det. gain',s5,'kd      ',s5,'[volts/rad]',s5,kd:15:3);
  writeln(outfile,
  s5,'d.c. plant gain',s5,'kp      ',s5,'[s]        ',s5,kp:15:3);
  writeln(outfile,
  s5,'system gain    ',s5,'gp      ',s5,'[1/s]      ',s5,gp:15:3);
  writeln(outfile,
  s5,'plant pole     ',s5,'repole  ',s5,'[1/s]      ',s5,repole:15:3);
  writeln(outfile,
  s5,'servo bandwidth',s5,'wservo  ',s5,'[hz]       ',s5,wservo/(2*pi):15:3);
  writeln(outfile);
  writeln(outfile,s5,'Type 1 lead compensator');
  writeln(outfile,
  s5,'filter gain    ',s5,'kfl      ',s5,'[1/s]      ',s5,kfl:15:3);
  writeln(outfile,
  s5,'filter zero    ',s5,'a        ',s5,'[rad/s]    ',s5,a:15:2);
  writeln(outfile,
  s5,'filter pole    ',s5,'b        ',s5,'[rad/s]    ',s5,b:15:2);
  writeln(outfile,
  s5,'velocity error ',s5,'verror   ',s5,'[rad]      ',s5,verror:15:4);
  writeln(outfile,
  s5,'max vel. error ',s5,'verrormax',s5,'[rad]      ',s5,verrormax:15:4);
  stopscroll;
end; {print_control}

procedure print_short_table(var outfile:text);
begin {short table}
  writeln(outfile,
  s5,'Tip Displ.     ',s5,'vc      ',s5,'[mic]       ',s5,vc/clgth:15:2);
  writeln(outfile,
  s5,'Tip Rot.       ',s5,'thetac  ',s5,'[deg]       ',s5,thetac*180/pi:15:3);
  writeln(outfile,
  s5,'min airgap     ',s5,'d0smin  ',s5,'[mic]       ',s5,d0smin/clgth:15:3);
  writeln(outfile,
  s5,'airgap         ',s5,'d0s     ',s5,'[mic]       ',s5,d0s/clgth:15:3);
  writeln(outfile,
  s5,'Torq Volt      ',s5,'v0      ',s5,'[v]         ',s5,v0:15:3);
  writeln(outfile,
  s5,'Frac Voltage   ',s5,'v0c     ',s5,'[v]         ',s5,v0c:15:3);
  writeln(outfile,
  s5,'Sensor Volt up ',s5,'eoutu   ',s5,'[mv]        ',s5,eoutu*1e3:15:2);
  writeln(outfile,
  s5,'Sensor Volt dn ',s5,'eoutd   ',s5,'[mv]        ',s5,eoutd*1e3:15:2);
  writeln(outfile,
  s5,'mode 1         ',s5,'w1      ',s5,'[hz]
',s5,omega[ndof]/(2*pi):15:3);
  writeln(outfile,
  s5,'freq shift     ',s5,'dw      ',s5,'[hz]        ',s5,dw/(2*pi):15:3);
  writeln(outfile,
  s5,'Crack lgth     ',s5,'da      ',s5,'[mic]       ',s5,da/clgth:15:4);
  writeln(outfile,
  s5,'Time to da     ',s5,'ta      ',s5,'[hrs]       ',s5,ta:15:2);
  writeln(outfile,
  s5,'Time to failure',s5,'tf      ',s5,'[hrs]       ',s5,tf:15:2);
  writeln(outfile,
  s5,'% initial k1   ',s5,'pctk1ci ',s5,'[ ]         ',s5,pctk1ci:15:3);
  writeln(outfile,
  s5,'Crack Stability',s5,'dk1/da  ',s5,'[n/m^(5/2)]',s5,dk1da:15);
  writeln(outfile,
  s5,'% frac stress  ',s5,'pctsf    ',s5,'[ ]       ',s5,pctsf:15:2);
  writeln(outfile,
  s5,'Crack velocity ',s5,'vsi02    ',s5,'[m/s]     ',s5,vsi02:15);
  writeln(outfile,
  s5,'Fatigue const  ',s5,'nsi02    ',s5,'[ ]       ',s5,nsi02:15:2);
  writeln(outfile,
  s5,'nano clear     ',s5,'Icl      ',s5,'[mic]     ',s5,Icl/clgth:15:3);
  writeln(outfile,
  s5,'nano load      ',s5,'Pind     ',s5,'[mN]      ',s5,Pind*1e3:15:3);
  writeln(outfile,
  s5,'max temp       ',s5,'Tmax     ',s5,'[deg C]   ',s5,Tmax:15:3);
  stopscroll;
end; {short table}

procedure print_var(var outfile:text);
begin
  writeln(outfile,filevar[1]:10:3,'     1     lab length of segment ab
[mic]');
  writeln(outfile,filevar[2]:10:3,'     2     lbc length of segment bc
[mic]');
  writeln(outfile,filevar[3]:10:3,'     3     x1 edge clearances [mic]');
  writeln(outfile,filevar[4]:10:3,'     4     x2 distance between torquer and
sensor plates [mic]');
  writeln(outfile,filevar[5]:10:3,'     5     x3 distance between sensor plate
and plate 2 [mic]');
  writeln(outfile,filevar[6]:10:3,'     6     lpy width of plate [mic]');
  writeln(outfile,filevar[7]:10:3,'     7     lh1x length of hole at torquer
[mic]');
  writeln(outfile,filevar[8]:10:3,'     8     lh1y width of at torquer hole
[mic]');
  writeln(outfile,filevar[9]:10:3,'     9     lh3y length of hole at sensor
[mic]');
  writeln(outfile,filevar[10]:10:3,'    10    lbh3y lcc of cg of hole 3
[mic]');
  writeln(outfile,filevar[11]:10:3,'    11    lbh4y loc of cg of hole 4
[mic]');
  writeln(outfile,filevar[12]:10:3,'    12    lbh5y loc of cg of hole 5
[mic]');
  writeln(outfile,filevar[13]:10:3,'    13    lp2x length of tip mass [mic]');
  writeln(outfile,filevar[14]:10:3,'    14    lp2y width of tip mass [mic]');
  writeln(outfile,filevar[15]:10:3,'    15    lp2z thickness of tip mass
[mic]');
  writeln(outfile,filevar[16]:10:3,'    16    lp1z thickness of plate mass
[mic]');
  writeln(outfile,filevar[17]:10:3,'    17    b0 width of beam at point a
(fixed end) [mic]');
  writeln(outfile,filevar[18]:10:3,'    18    bb width at V-notch [mic]');
  writeln(outfile,filevar[19]:10:3,'     19     h thickness of beam [mic]');
  writeln(outfile,filevar[20]:10:3,'     20     hb V-notch depth [mic]');
  writeln(outfile,filevar[21]:10:3,'     21     ltx length of torquer [mic]');
  writeln(outfile,filevar[22]:10:3,'     22     lsx length of sensor  [mic]');
  writeln(outfile,filevar[23]:10:3,'     23     d0s distance from sensor elec.
to plating [mic]');
  writeln(outfile,filevar[24]:10:3,'     24     d0t distance from torquer elec.
to plateing [m]');
  writeln(outfile,filevar[25]:10,  '     25     cfb feedback capacitance [f]');
  writeln(outfile,filevar[26]:10:3,'     26     vexc excitation voltage on
sensor [v]');
  writeln(outfile,filevar[27]:10:4,'     27     da crack length [mic]');
  writeln(outfile,filevar[28]:10:4,'     28     rtip tip radius of V-notch
[mic]');
  writeln(outfile,filevar[29]:10:4,'     29     wservo servo bandwidth [hz]');
  writeln(outfile,filevar[30]:10:4,'     30     v0 applied torquer voltage
[v]');
  writeln(outfile,filevar[31]:10:4,'     31     h0p plate airgap [mic]');
  writeln(outfile,filevar[32]:10:4,'     32     vbias sensor bias voltage [v]');
  writeln(outfile,trunc(filevar[33]):10,'    33    ndof # DOF of analysis (2
or 3)');
  writeln(outfile,trunc(filevar[34]):10,'    34    failure simulation (1 =
yes, 0 = no)');
  writeln(outfile,trunc(filevar[35]):10,'    35    fatigue simulation (1 =
yes, 0 = no)');
  writeln(outfile,filevar[36]:10:4,'     36     vsi02 crack velocity [m/s]');
  writeln(outfile,filevar[37]:10:4,'     37     nsi02 fatigue constant');
end; {printvar}

procedure print_timeanddate(var outfile:text);
var
  hour,min,sec,sec100,dayofweek,day,month,year:word;
  dayname:string[9];
begin
  gettime(hour,min,sec,sec100);
  getdate(year,month,day,dayofweek);
  case dayofweek of
    1: dayname:='Monday';
    2: dayname:='Tuesday';
    3: dayname:='Wednesday';
    4: dayname:='Thursday';
    5: dayname:='Friday';
    6: dayname:='Saturday';
    0: dayname:='Sunday';
  end;
  write(outfile,'Date of run: ',dayname,' ',month,'/',day,'/',year);
  writeln(outfile,s10,'Time of run: ',hour,':',min,':',sec);
end; {timeanddate}
```

```
procedure print_heading(var outfile:text);
begin
  print_timeanddate(outfile);
  writeln(outfile,
  '                            OUTPUT SUMMARY');
  writeln(outfile,
  '---------------------------------------------------------------------');
{  writeln(outfile,s5,'Description',s10,'Var',s10,'units',s10,s5,'Value');  }
{  writeln(outfile,                                                          }
{   '---------------------------------------------------------------------');}
end;{heading}

procedure print_comments(var outfile:text);
var
  line:string;

begin
  if (option='p') or (option='d') or (option='n') then
  begin
    writeln('Comments:');
    writeln(outfile,
    '------------------------------------------------------------------');
      while not eoln do
      begin
        readln(line);
        writeln(outfile,line);
      end;
  end,
end;

begin
end.
```

æ

Bibliography

[1] M.S. Atlova. Electromechanical effect in ge, si, and insb. *Soviet Physics - Solid State*, 6(10):pp. 2520–2521, April 1965.

[2] S.B. Bhaduri and F.F.Y. Wang. Slow crack growth studies in silicon. In *Fracture Mechanics of Ceramics*, chapter 5, pages pp. 327–336, Plenum Press, 1983.

[3] C.P. Chen and S.Y. Hsu. Acoustic emission monitoring crack propagation in single crystal silicon. In D.O. Thompson and D.E. Chimenti, editors, *Review of Progress in Quantitative Nondestructive Evaluation*, chapter 4B, pages pp. 689–699, Plenum Press, 1985.

[4] C.P. Chen and M.H. Leipold. *Crack Growth in Single Crystal Silicon.* NASA Tech Brief Vol. 10, No. 3, Item no. 106, National Aeronautics and Space Administration, NASA Technology Transfer Div., P.O. Box 8757, BWI Airport, MD. 21240, May/June 1986.

[5] C.P. Chen and M.H. Leipold. Fracture toughness if silicon. *American Ceramic Society Bulletin*, 59(4):pp. 469–472, 1980.

[6] T.J. Chen and W.J. Knapp. The fracture of single crystal silicon wafers under several liquid environments. *Journal of the American Ceramic Society - Discussion and Notes*, March-April 1980.

[7] M.F. Doerner and W.D. Nix. A method for interpreting the data from depth sensing indentation instruments. *Journal of Materials Science*, pp. 601–609, July/August 1986.

[8] R.L. Eisner. Tensile tests on silicon whiskers. *ACTA Metallurgica*, 3:pp. 414–415, 1955.

[9] A.G. Evans. Fatigue in ceramics. *International Journal of Fracture*, 16(6):pp. 485–498, December 1980.

[10] A.G. Evans. Fracture mechanics determination. In *Fracture Mechanics of Ceramics*, chapter 1, pages pp. 17–48, Plenum Press, New York-London, 1974. Concepts, Flaws, and Fractography.

[11] A.G. Evans. Slow crack growth in brittle materials under dynamic loading conditions. *International Journal of Fracture*, 10:pp. 251–259, 1974.

[12] Long-Sheng Fan, Roger T. Howe, and Richard S. Muller. Microstructures for fracture toughness characterization of brittle thin films. In *IEEE Micro Electro Mechanical Systems Workshop*, pages pp. 40–41, February 20-22, 1989. Salt Lake City, Utah Conference.

[13] L.S. Fan, Y.C. Tai, and R.S. Muller. Pin joints, gears, springs, cranks, and other novel micromechanical structures. In *International Conference on Solid State Sensors and Actuators*, pages pp. 849–852, Tokyo, Japan, 1987.

[14] K.J. Gabriel, B.Behi, R. Mahadevan, J.A. Walker, and M. Mehregany. Silicon based micromechanical structures and devices. In *Proceedings of the SPIE - International Society for Optical Engineers*, pages pp. 137–150, Precision Engineering and Optomechanics, 1989. San Diego, CA. Conference.

[15] Sorab K. Ghandhi. *VLSI Fabrication Principles*, chapter 7. John Wiley and Sons, 1983.

[16] P. Haasen. Kink formation and migration as dependent on the fermi level. *Journal de Physique*, 40:pp. C6–111, 1979.

[17] Won-Taek Han and Minoru Tomozawa. Crack initiation and mechanical fatigue of silica glass. *Journal of Non-Crystalline Solids*, 122(1):pp. 90–100, June 1990.

[18] J. Happel. *Low Reynolds Number Hydrodynamics.* p.

[19] Robert E. Hetrick. Acoustically driven cantilever for gas sensing. *Sensors and Actuators*, 18:pp. 131–147, 1989. Research Staff, Ford Motor Corporation.

[20] P.B. Hirsch, P. Pirouz, S.G. Roberts, and P.D. Warren. Effects of doping on mechanical properties of semiconductors. In *Science of Hard Materials*, pages pp. 83–105, 1986.

[21] S. Hong, T.P. Weihs, J.C. Bravman, and W.D. Nix. The determination of mechanical parameters and residual stresses for thin films using microcantilever beams. In *Materials Research Symposium*, pages pp. 93–98, Materials Research Society, 1989. volume 130.

[22] S. Hong, T.P. Weihs, J.C. Bravman, and W.D. Nix. Measuring the strength and stiffness of thin film materials by mechanically deflecting cantilever micro-beams. In *Materials Research Symposium*, pages pp. 87–92, Materials Research Society, 1989. volume 130.

[23] Stefan Johansson and Jan-Ake Schweitz. Fracture testing of silicon microelements in situ in a scanning electron microscope. *Journal of Applied Physics*, 10(63):pp. 4799–4803, May 15, 1988.

[24] Melvin F. Kanninen and Carl H. Popelar. *Advanced Fracture Mechanics*, chapter 6, page p. 158 and 398. Oxford University Press, 1985.

[25] S.B. Kulkarni and W.S. Williams. Dislocation velocities and electronic doping in silicon. *Journal of Applied Physics*, 47:pp. 4318–4324, 1976.

[26] B.R. Lawn, A.G. Evans, and D.B. Marshall. Elastic/plastic indentation damage in ceramics: the median/radial crack system. *Journal of the American Ceramic Society*, 63:pp. 574–581, 1980.

[27] J.E. Lawrence. Diffusion induced stress and lattice disorders in silicon. *Journal of the Electrochemical Society*, 113:pp. 819–824, 1966.

[28] Jean L. Loubet, Jean M. Georges, and Gerard Meille. Vickers indentation curves of elastoplastic materials. In P.J. Blau and B.R. Lawn, editors, *Microindentation Techniques in Materials Science and Engineering, ASTM/STP 889*, pages pp. 72–89, American Society for Testing and Materials, Philadelphia, 1986.

[29] Matthew B. Magida, Katherine A. Forrest, and Thomas M. Hesli. Dynamic and static fatigue of a machinable glass ceramic. In S.W. Freiman

and C.M. Hudson, editors, *Methods for Assessing the Structural Reliability of Brittle Materials*, pages pp. 81–94, ASTM, Philadelphia, 1984.

[30] David B. Marshall and Brian R. Lawn. Indentation of brittle materials. In P.J. Blau and B.R. Lawn, editors, *Microindentation Techniques in Materials Science and Engineering, ASTM/STP 889*, pages pp. 26–45, American Society for Testing and Materials, Philadelphia, 1986.

[31] T.A. Michalske and S.W. Freeman. A molecular mechanism for stress corrosion in vireous silica. *Journal of the American Ceramic Society*, 66(4):pp. 284–288, 1983.

[32] L.S. Milevskil and Yu. N. Chuvilin. Properties of the motion of dislocation loop segments formed in heavily doped silicon crystals. *Sov. Phys. Solid State*, 22(9):pp. 1536–1540, September 1980.

[33] A. Misra and I.Finnie. On the scribing and subsequent fracturing of silicon semiconductor wafers. *Journal of Material Science*, 14(11):pp. 2567–2574, Nov. 1979.

[34] Murakami. *Stress Intensity Factors Handbook*, page p.11. Volume 1, Pergamon Press, 1987.

[35] G.L. Peterson, W.T. Read, and W.L. Feldmann. Deformation and fracture of small silicon crystals. *ACTA Metallurgica*, 5:pp. 181–191, 1957.

[36] John E. Ritter Jr., Karl Jakus, and Robert C. Babinski. Effect of temperature and humidity on delayed failure of optical glass fibers. In S.W. Freiman and C.M. Hudson, editors, *Methods for Assessing the Structural Reliability of Brittle Materials*, pages pp. 131–141, ASTM, Philadelphia, 1984.

[37] S.G. Roberts, P. Pirouz, and P.B. Hirsch. A simple technique for measu ing doping effects on dislocation motion in silicon. *Journal De Physique*, 44(Colloque C4), 1983.

[38] D.P. Rooke and D.J. Cartwright. *Compendium of Stress Intensity Factors*, page pages 86 and 298. Her Majesty's Stationary Office, London, 1976.

[39] George C. Ryhbicki and P. Pirouz. *Indentation Plasticity and Fracture in Silicon*. Technical Report 2863, NASA, 1988.

[40] I.N. Sneddon. The relation between load and penetration in the axisymmetric boussinesq problem for a punch of arbitrary profile. *International Journal of Engineering Science*, 3:pp. 47–57, 1965.

[41] Yu-Chong Tai and Richard S. Muller. Measurement of young's modulus on microfabricated structures using a surface profiler. In *IEEE Proceedings Micro Electro Mechanical Systems*, pages pp. 147–152, IEEE Robotics and Automation Society, February 11-14, 1990. Napa Valley, CA. Conference.

[42] W.S.N. Trimmer and K.J. Gabriel. Design considerations for a practical electrostatic micro-motor. *Sensors and Actuators*, 11(11):pp. 189–206, 1987.

[43] Wallace E. Vander Velde and Arthur Gelb. *Multiple-input Describing Functions and Nonlinear System Design*. McGraw Hill, 1968.

[44] J.H. Westbrook and J.J. Gilman. An electromechanical effect in semiconductors. *Journal of Applied Physics*, 33(7):pp. 2360–2369, July 1962.

[45] S.M. Wiederhorn. Subcritical crack growth in ceramics. In *Fracture Mechanics of Ceramics*, chapter 2, pages pp. 613–646, Plenum Press, New York-London, 1974. Microstructure, Materials and Applications.

[46] S.M. Wiederhorn, E.R. Fuller, and R. Thompson. Micromechanisms of crack growth in ceramics and glasses in corrosive environments. *Metal Science*, 14:pp. 450–458, 1980.

[47] Helmut F. Wolf. *Semiconductors*, chapter 1-9, page p. 128. Peragamon Press, 1969.

[48] Helmut F. Wolf. *Silicon Semiconductor Data*. Volume 9, Peragamon Press, 1969. p. 199.

[49] B. Wong and R.J. Holbrook. Microindentation for fracture and stress corrosion cracking studies in single crystal silicon. *Journal of the Electrochemical Society*, September 1987.