THERMAL-MECHANICAL FATIGUE BEHAVIOR OF NICKEL-BASE SUPERALLOYS

VOL 1

by

NORMAN J. MARCHAND

B. Ing., Ecole Polytechnique de Montreal, 1980
M.Sc.A., Ecole Polytechnique de Montreal, 1982

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF

DOCTOR OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
January 1986

Signature of Author . . . .
Department of Materials Science & Engineering
January 10, 1986

Certified by . . . .
R. M. Pelloux
Thesis Supervisor

Accepted by . . . .
Bernhardt J. Wuensch
Chairman, Departmental Graduate Committee

THERMAL-MECHANICAL FATIGUE BEHAVIORS OF NICKEL-BASE SUPERALLOYS

by

NORMAN J. MARCHAND

Submitted to the Department of Materials Science and Engineering on January10,1986 in partial fulfillment of the requirements for the degree of Doctor of Science.

ABSTRACT

The thermal-mechanical fatigue (TMF) behavior of three nickel-based superalloys, which cover a wide range of ductility/strength ratio, was studied in the temperature range from 300 to 925°C. The materials investigated are Inconel X-750, Hastelloy-X and B-1900+Hf. In-phase (T_{max} at σ_{max}, ε_{max}), out-of-phase (T_{min} at σ_{max}, ε_{max}) and isothermal tests at T_{min} and T_{max} were performed under fully reversed cyclic conditions.

A microprocessor based system has been developed to control temperature and mechanical strain during the tests. The system has the ability to simultaneously control mechanical load/strain in addition to temperature. A DC electrical potential method was used to measure crack length. The electrical potential response obtained for each cycle of a given wave form and R value yields information on crack closure and crack extension per cycle.

The detailed cyclic hardening behavior of B-1900+Hf under TMF conditions was studied. The cyclic response was investigated for constant amplitude, fully reversed, strain cycling of uniaxially loaded specimens in the temperature range from 400 to 925°C. Cycling over a ductility minimum was observed to cause more cyclic hardening than isothermal fatigue experiments. In terms of mean stress and plastic strain range, out-of-phase cycling was more deleterious than in-phase cycling or isothermal cycling. On the basis of stable stress range, however, there was little difference between in-phase and out-of-phase cycling. TEM observations have shown that coarsening of γ' phase is primarily responsible for the observed behavior in TMF testing.

The thermal-mechanical fatigue crack growth (TMFCG) tests of Inconel X-750 was studied under controlled load amplitude in the temperature range from 300 to 650°C. In-phase, out-of-phase and isothermal tests at 650°C were performed on single-edge notch bars

under fully reversed cyclic conditions. A limited number of TMFCG in tension-tension (R=0.05) were also performed. The macroscopic crack growth rates were reported as a function of the stress intensity factor ΔK. Faster crack growth rates were measured under out-of-phase than under in-phase cycling at R= -1 or R=0.05. This behavior was rationalized by introducing the concept of an effective closure stress which was defined as the applied stress at the crossover point of the $V(\sigma)$ potential curve. This closure stress was shown to represent the effective contribution to cracking of the damage taking place during the compressive part of the cycle. Correction to the applied stress were introduced in order to take into account this damage. Correlation between TMFCG rates and ΔK_{eff} were shown to be valid provided elastic conditions prevailed in the bulk.

The TMFCG behavior of Hastelloy-X and B-1900+Hf was studied under fully reversed strain-controlled conditions in the temperature range from 400 to 925°C. The crack growth rates were first reported as a function of the strain intensity factor (ΔK_{ε}) derived from a crack compliance analysis. Out-of-phase cycling showed faster crack growth rates than isothermal or in-phase cycling at $\Delta \varepsilon = 0.25\%$. Under fully plastic cycling the opposite results were observed, i.e. crack growth rates under isothermal cycling are faster than under TMF cycling. On a ΔK_{ε} basis, a strain range effect was observed in both alloys.

The results were rationalized using a corrected stress-intensity factor computed from the actual load, the closing bending moment caused by the increase of compliance with crack length, and with the effective closure stress measured with the electrical potential signal. The closure stress was shown to be controlled by the testing condition (in-phase, out-of-phase or isothermal) rather than by the strain range or by the resulting stress range. Each mode of fracture was found to be characterized by a unique crack growth rate vs ΔK_{eff} curve.

Thesis Supervisor: Dr. R. M. Pelloux

Title: Professor of Materials Engineering

Table of Contents

List of Figures

List of Tables

Acknowledgements

The author is grateful to a number of people who assisted and encouraged him throughout this work and during the course of his stay at the Massachusetts Institute of Technology.

To the National Aeronautics and Space Administration who sponsored this work (Grant #NAG3-280). To this regard, the supervision of Drs. Marvin Hirschberg and Bob Bill is gratefully acknowledged.

To Professor R.M. Pelloux, his thesis advisor, for his invaluable technical and personal guidance.

To Professors F.A. McClintock and D.M. Parks for their review of the manuscript, but most of all, for their enlightening comments and for the many discussions on fracture mechanics and thermal-fatigue.

To Professor R.G. Ballinger for his review of the manuscript and for his help in designing the experimental set-up.

To Pete Stahl (MTS Corporation) for his assistance and guidance in repairing and improving the servo-hydraulic machine.

To Dr. Gilles L'Esperance (Ecole Polytechnique de Montreal) for his assistance in carrying-on the TEM study.

To Jeff Hill (Pratt & Whitney Aircraft) and Eric Jordan (Univ. of Conn.) for providing all the materials and specimens used in this investigation, and for the many discussions on fracture mechanics.

To Heather Ballinger who went through the pain of typing the manuscript, I am grateful.

To my close friends in the fatigue group; Ioannis, Bertrand, Jung, Glenn, Jon and Allison, for all their support and encouragement throughout the completion of this work. Without them a considerable fraction of this thesis would not have been completed. I am eternally indebted to them.

To all my other friends; Ibrahim, Janine, John, Bob, Edmund, Arthur, Charley and Magnus, I am grateful. They have made my stay at MIT a very fruitful and pleasant one. Their friendship will always remain a very pleasant memory.

1.0 Introduction

Many fatigue problems in high temperature machinery involve thermal as well as mechanical loadings. By thermal loading it is meant that the material is subject to cyclic temperature simultaneously with cyclic stress or strain. Analysis of these conditions and consideration of the attendant fatigue damage becomes very complex and often gross simplifications are introduced.

Low cycle fatigue and thermal fatigue are recognized failure modes for structures such as gas turbine components which operate with combined temperature and strain cycling. A limited listing of materials, operating temperatures and potential life-limiting failure modes for aircraft engine components are given in Table 1.1.

The thermal fatigue cracking encountered in power plant components and gas turbine engines is due to internal constraints [1-2], arising when surfaces are heated rapidly and are prevented from expanding by the bulk section whose temperature rises more slowly. Under such circumstances, compressive plastic yielding may occur during heating and later, when the bulk material reaches the suface temperature, tensile residual stresses will be set up which, if large enough, will cause reverse plastic strains. The process may reverse on shut-down and thus a low cycle fatigue situation can arise during each start-up, shut-down operational cycle. Figure 1.1 from Moreno et al. [13] shows the kind of strain temperature cycle believed to be generated during non-uniform heating and cooling of high temperature components. The important point to note is that this fatigue cycle does not arise under isothermal conditions.

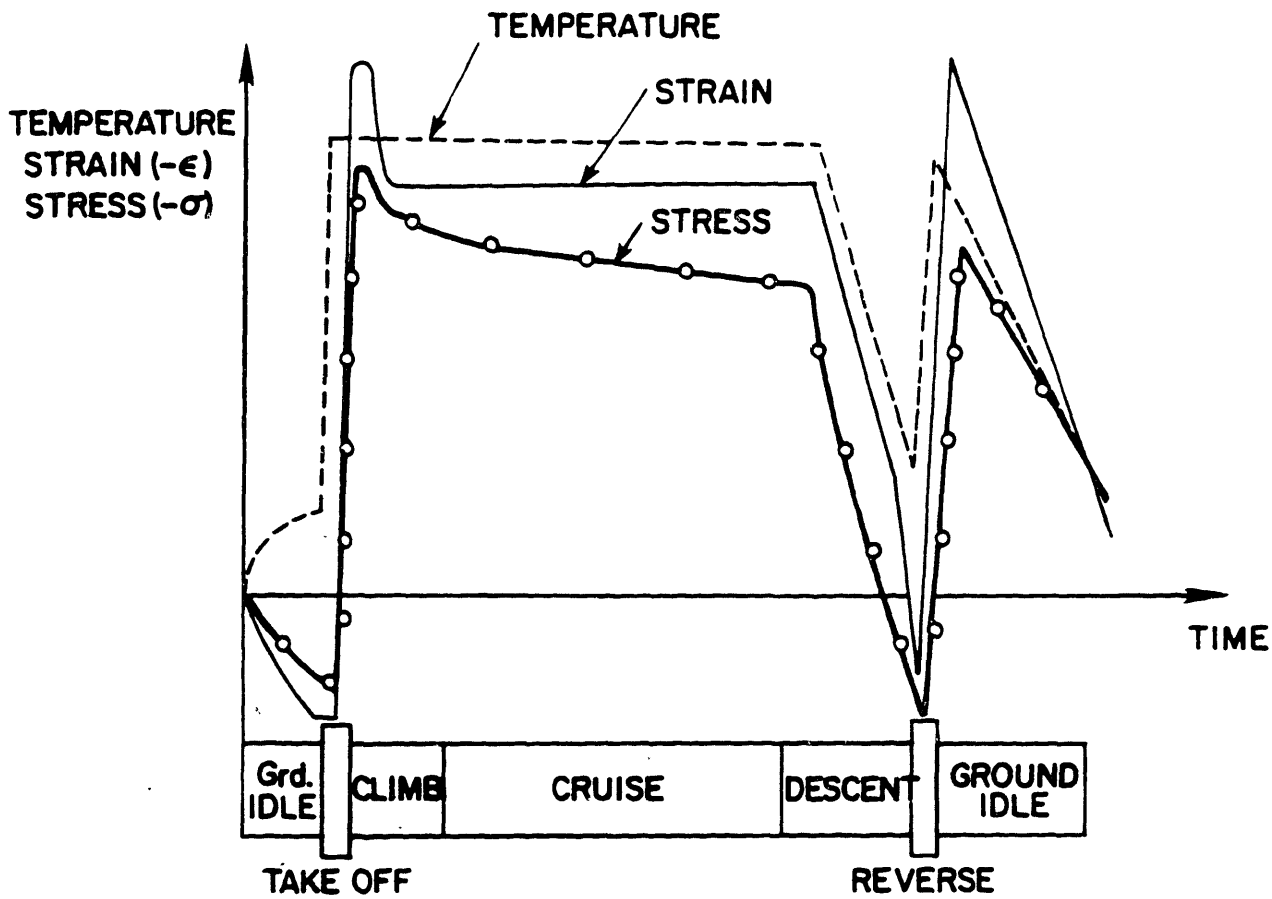


Figure 1.1 Stress-strain-temperature cycle experience at the lower tip of a combuster liner[13].

Table 1.1

POTENTIAL PROBLEMS IN GAS TURBINE MATERIALS

Component	Material Type	Temperature(°C)	Concern
compressor disk	Ni-based	< 400	burst, low cycle fatigue
compressor casing	age-hardened stainless steel or titanium alloy	< 400	high cycle fatigue
shaft	Fe-Ni-Cr	< 225	high cycle fatigue, high-strain fatigue
combustor liner	Co-based and/or Ni-based	≤1000	thermal fatigue, thermal mechanical fatigue
combustor casing	Ni-based	< 550	thermal fatigue
turbine blades	Ni-based	≤1000	thermal mechanical fatigue, high cycle fatigue
turbine disks	Ni-based	370-650	burst (over speed), low cycle fatigue high cycle fatigue creep fatigue

The influence of temperature on low cycle fatigue life is well documented [3-6] but the mechanisms by which temperature influences the fatigue process are not well understood. Low cycle fatigue life is generally acknowledged to be directly related to material ductility; however, as ductility increases with temperature, low cycle fatigue life normally decreases. Creep and environmental effects also are known to influence LCF behavior, but the relative contribution of these two factors is not well defined [2,4-6].

A major objective of most high-temperature fatigue research programs is to develop and verify models for thermal fatigue by comparing experimental data with analytical and/or computer-derived predictions of thermal fatigue life.

Prediction of crack propagation rates in structural components from specimen data generated in the laboratory is only possible if a parameter can be found which characterizes the severity of the stress and strain cycles near the crack tip. Such a parameter is needed to match a particular loading and crack length in a component with the correct equivalent specimen loading and crack length. In cases of cyclic loading mainly involving linear elastic deformation (small scale yielding), the stress intensity factor (ΔK) is a widely used and successful parameter. However, the stress intensity factor may not be applicable for some hot section components, since cracks may grow through regions of substantial plastic deformation.

The prediction of propagation life in engine components requires the consideration of thermal-mechanical fatigue (TMF) cycles. The problem of thermo-mechanically driven crack growth in the presence of significant inelastic strain is a challenging problem. In order for a single parameter to be useful for predicting thermal-mechanical crack growth in components, it should satisfy the following conditions:

(1) It should predict crack growth rate from a single crack growth rate versus parameter curve. In this way, cracks of different lengths loaded in such a way as to yield the same value of the parameter, experience the same crack growth rate (similitude principle).

(2) It should correctly predict fatigue crack growth rates independent of part geometry.

(3) It should be calculable for complex real part geometries.

Obviously, parameters not satisfying the above requirements would be of limited value, since component or simulated component testing would always be required to obtain crack growth rate information.

In this research program, it was proposed to assess the suitability of various parameters for correlating high temperature and thermal-mechanical crack growth rates. A parameter was sought which can correlate data for the full range of conditions from elastic stress and strain cycling to substantially plastic strain cycling. The ultimate goal of establishing such a parameter is the prediction of the propagation life of real engine components, since it is known that cracks initiate early in fatigue life.

A second goal was to combine the crack growth data generated with optical, fractographic, and TEM observations to assess a fundamental mechanistic understanding of the thermal-mechanical fatigue behavior of typical nickel-based superalloys.

To achieve these goals, the following procedure was decided upon. First, TMF was studied under linear elastic loading conditions to assess the applicability of fracture mechanics to correlate TMFCG. Then, displacement controlled (i.e. strain controlled) TMF cycling under elastic and fully plastic conditions was studied. The TMF cyclic stress-strain behavior was combined with the crack growth data to select the parameter that best correlates the crack growth behavior.

2.0 Introduction to Thermal-Mechanical Fatigue

2.1 Definition

The subject of thermal fatigue which involves combined temperature and strain cycling, is less well understood than isothermal elevated temperature fatigue. Taira has described [7] thermal fatigue as a special case of low cycle fatigue but a more specific definition has been given by Spera [1], i.e., "thermal fatigue is the gradual deterioration and eventual cracking of a material by alternate heating and cooling during which free thermal expansion is partially or completely restrained". Spera has also proposed the following terminology by which he subdivides the general field of high temperature fatigue into thermal and isothermal categories, depending on whether temperature is cyclic or constant. As an example, thermal fatigue might result from starting and stopping of a of high temperature component, while isothermal fatigue might be the consequence of vibration during steady-state operation. The thermal fatigue category may then be subdivided into thermal-mechanical fatigue and thermal-stress fatigue, depending on whether constraints are external or internal (Figure 2.0).

As pointed out before, thermal fatigue testing can be divided into two categories, the first involving internal constraint, the second external constraint. The classic example of the former type is the use of tapered disc specimens which are alternately immersed in high and low temperature fluidized beds [8-11]. Such tests can provide a reasonable simulation of operational conditions encountered by such components as turbine blades and can readily assess the relative thermal fatigue

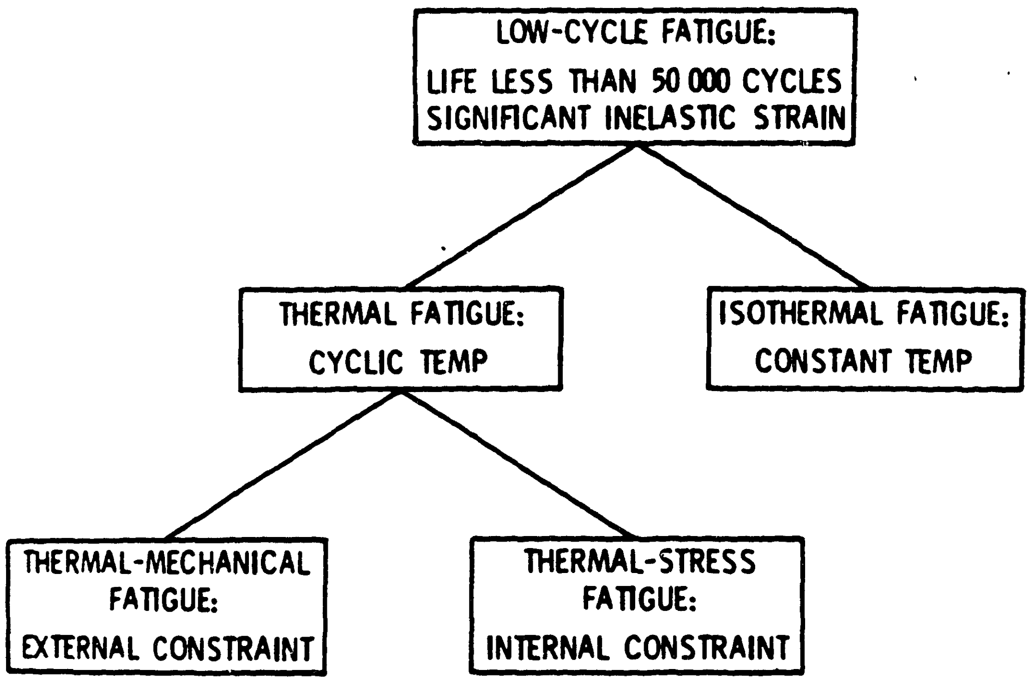


Figure 2.0 Proposed terminology for high temperature fatigue[1].

resistance of different materials. Unfortunately, stress and strain cannot be measured directly in this kind of test so that quantitative analyses are usually precluded, although use of finite element techniques has improved the situation [12-14].

External constraints are those provided by boundary forces applied to the surfaces of the body which is being heated and cooled. This type of constraint is more typical of specimen testing than it is of components in-service because designers of high-temperature equipment usually take considerable care to provide for overall thermal expansion and contraction through the use of clearances, sliding supports, bellows, etc. Thus, thermal fatigue with external constraint is primarily a laboratory testing practice in which external forces on a test specimen are used to simulate internal thermal stresses in an actual component [1].

External constraint was first used by Coffin who applied temperature cycles to hourglass type fatigue specimens held in a constraining jig [15]. The fact that most tests since then have involved isothermal strain cycling is perhaps unfortunate, since the operational fatigue situation usually involves cyclic strain resulting from and accompanying cyclic temperature. This situation has arisen partly because isothermal tests are relatively simple to perform, but also because it has often been felt that such tests carried out at the maximum service temperature should give worst-case results. However, several studies have compared fatigue resistance under thermal cycling conditions with that in isothermal tests and have shown that in many cases, the latter, rather than giving a worst-case situation, can seriously overestimate the fatigue life under more realistic conditions [16-35].

In recent years, there has been some revision to the Coffin type approach involving cyclic temperature and external constraint, but the simple restraining jig has been replaced by servo-controlled fatigue machines in which temperature and strain cycles can be applied independently using suitable programming devices. This, in Spera's terminology, is the thermal-mechanical fatigue test [1]. In its most sophisticated form, thermal-mechanical testing involves direct strain control using high temperature extensometry with provision for automatic thermal strain compensation. Hopkins [36], Sobolev and Egorov [37], and Ellison [38] have described the state-of-the-art of this type of test.

2.2 Parameters Affecting TMF

The material properties other than ductility which are important in thermal fatigue are hot tensile strength, elastic modulus, thermal conductivity, and thermal expansion. Oxidation resistance also plays an important role in thermal fatigue. The interactions between material properties, imposed thermal cycle, and component geometry define the ability of a structure to resist thermal fatigue. However, the synergistic effects between these variables are quite complex and prediction of thermal fatigue behavior from basic properties is difficult. To clarify the situation, it is thought appropriate to review separately the mechanical response of alloys under diverse applied loading conditions and the effect of oxidation on the high temperature fatigue.

2.2.1 Cyclic Stress-Strain Responses

Largely because of the large body of information that has been accumulated, it is common to look upon the triangular wave shape test under fully reversed straining as a standard test from which one can predict other wave shape effects. This is referred to as the symmetrical hysteresis stress-strain loop with zero mean stress (Fig. 2.1a). Other wave shapes are obtained by changing various parameters (stress, temperature, strain rate, etc.); some of which are shown in Figure 2.1. The study of the material responses to changes in these parameters have allowed us to identify the main mechanical variables controlling fatigue lives and crack growth under TMF conditions.

<u>Mean Stress</u>

One family of shape involves strain hold period (in tension, compression or both). Figure 2.1b illustrates the tension hold. Note that the hysteresis loop will show a mean stress which is a function of the difference in the time spent in the tensile versus compressive part of the cycle. This mean stress effect can be important particularly when the plastic strain is small relative to the total strain range. In that case elastic effects dominate and a lowering of the relaxed stress with decreasing hold time is very effective in shifting the loop. On the other hand, if the plastic strain range is large relative to the total strain range, the magnitude of the relaxed stress in inconsequential, the maximum stress and relaxed stress become approximately equal and their magnitudes are controlled by plastic flow considerations.

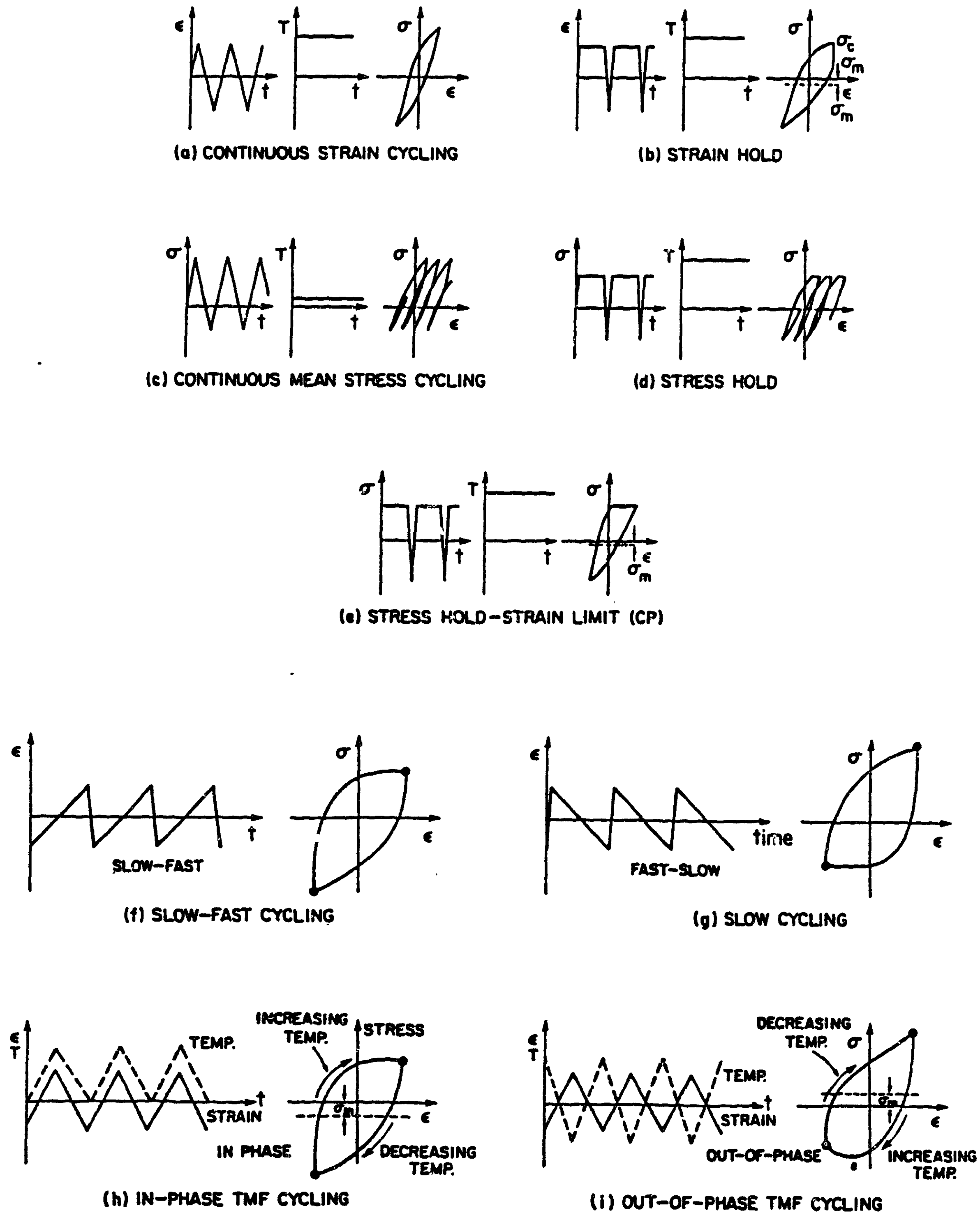


Figure 2.1 Typical stress-strain responses obtained at high temperature.

The consequences of a mean stress on fatigue life can be accounted for if one considers most of the life as spent in crack propagation, assuming that microcracks have nucleated early in the life. Here, the magnitude of the mean stress and its duration is important. Although the plastic strain range of the specimen is the same for tensile and compressive strain hold tests, the mean stress influences the local tensile plastic strain at the crack tip. A high compressive mean stress is likely to retard the amount of opening of the crack and lower the growth rate while a tensile mean stress causes a greater crack opening and enhances crack growth. Although documentation of the effect of mean stress on crack growth at high strain levels considered is lacking, there is ample evidence to support the observed effects at lower growth rates.

Ratchetting

Figure 2.1c shows a wave shape due to continuous stress cycling. Here a mean stress exists and ratchetting (incremental increase in strain) takes place, depending on the transition fatigue life. This form of ratchetting can be time independent and will occur at all temperatures, depending only on the magnitude of the mean stress and the plastic strain range. Also, time-dependent ratchetting can be expected at elevated temperature due to creep effects.

Another type of wave shape (Figure 2.1d) involves stress control. When the stress is fully reversed but hold time periods are introduced, ratchetting may lead to an increase of the strain during each cycle. The ratchetting here is the result of creep during the hold period and is associated with the difference in the hold time period for tensile versus compressive stress.

Creep Strain

A wave shape that has received much attention is that shown in Figure 2.1e. Wave shapes in this category are identified as cp or pc, depending on whether the hold period is in tension or compression. It develops that, for the same inelastic strain and at high temperature (creep effects are sufficiently rapid to operate), the cp tests give shorter lives than the pc. Thus in Figure 2.1e, most of all the tension- going deformation is by a time-dependent mechanism, while all compression-going deformation is presumably by a time-independent process. If the microstructural features of these two distinct modes differ, as, for example, if all time-dependent deformation is by grain boundary sliding and cavities formation, and the time-independent deformation is by deformation within the grain, internal damage processes could be expected to be greater than for the equivalent deformation modes in both directions [23-24]. For example, in crack propagation, crack advance by GBS followed by plastic crack closure can be pictured as contributing to greater crack growth than crack advance by grain boundary sliding following by crack closure by the same mechanism.

Strain Rate Effects

Figures 2.1f and 2.1g show hysteresis loops of stress and plastic strains corresponding to unequal plastic strain ramp rates. One could see that unequal ramp rates generate unbalanced hysteresis loops. Where the ramp rate is slow, the corresponding stress developed just prior to strain reversal is low; when the ramp rate is fast, the stress produced is high which is in agreement with normal stress-strain rate dependence. Thus, a slow-fast loop will be lower in tension and higher in compression, while a fast-slow is just the opposite. Correspondingly, the unequal loops will be reversed such that a slow-fast loop will have a compressive mean stress and a fast-slow loop a tensile mean stress.

Temperature

Wave shape effects are very important for thermal cycling where the effects can be accentuated by changing the temperature such that high temperature occurs during the tension-going deformation and low temperature during fast reversal. This type of temperature-strain cycling is known as in-phase cycling and is shown in Figure 2.1h. When compressive deformation occurs at high temperature and tensile deformation at low temperature, also known as out-of-phase cycling, the shape of the hysteresis loop differs from in-phase cycling. Figures 2.1g, 2.1h, and 2.1i show that rather unusual loops are produced under these conditions. In-phase cycling generates loops having a net compressive stress, while out-of-phase cycling leads to a mean tensile stress. As discussed earlier, one would conclude that out-of-phase cycling is more damaging than in-phase; this is what has been observed for nickel-base alloys [19, 27-29], tantalum alloys [20] and some low carbon steels [23-26]. However, there are numerous cases, especially with 304 and 316 SS, where the opposite effect have been observed which clearly indicate that the presence of a mean stress is not solely responsible for the shortening of fatigue life. Nevertheless, the existence of a mean stress in TMF (in-phase and out-of-phase) points out the need for its quantitative determination under complex straining and thermal histories. This points out the importance of proper constitutive equations and models for handling these equations if problems of practical importance are to be solved.

2.2.2 Effects of Environment

During the last few years it has become a much debated question as to what extent the environment affects cyclic life at elevated temperatures. It is usually observed after testing in air that fatigue cracks are oxide-filled, and it is tempting to ascribe the well-known frequency-and time-dependence of high temperature fatigue to oxidation, since oxidation is a time-dependent process. On the other hand, creep processes also become important at high temperatures and creep is also a time-dependent process. It is a difficult problem to differentiate the effects of creep from the effects of environment because the susceptibility to environmental damage increases with temperature. This can be seen in Figure 2.2, where the fatigue lives of IN-706 are plotted versus temperature at constant frequency and strain range [39].

To clarify the situation, a number of literature surveys dealing with the effects of environment on high temperature fatigue have been written [39, 43-46]. Especially worth mentioning are the reviews by Cook and Shelton [43] and by Marshall [46], which thoroughly discuss environmental effects.

It is evident from these reviews that environment has a pronounced effect on fatigue at elevated temperatures. The most

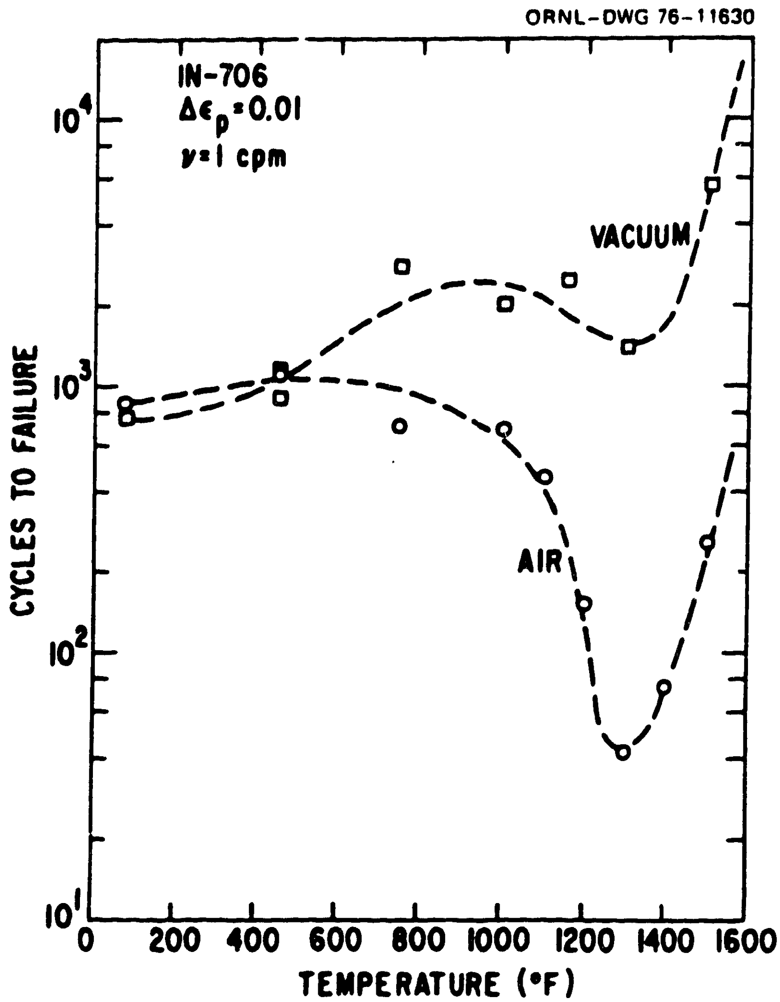


Figure 2.2 Comparative low-cycle fatigue resistance of wrought IN-706 in air and vacuum[39].

important results highlighted by these reviews may be summarized in Table 2.1 and 2.2, where the types of metal/environment interactions are listed in Table 2.1 and the principal effects on the fatigue and creep resistance listed in Table 2.2.

One can see in Table 2.2 that environmental effects have been studied mostly in terms of how an environment is changing the mechanical properties at or near the highly strained regions i.e., the surface or the crack tip of a material. Segregation of elements including O_2 and N_2 to and from grain boundaries is likely to be accentuated by plastic straining. Howes [48] concluded that the two dominant mechanisms in the formation of a thermal stress fatigue crack are local plastic strain and oxidation. After an incubation period, cracks almost invariably occur in areas of highest plastic strain [48]. Oxidation is often preferential at grain boundaries [5-6, 42-46, 48-54], forming wedges which grow in size until a crack is initiated and begins to propagate. Oxidation increases the amount of local plastic strain by reducing an alloy's strength through depletion of strengthening elements and phases. Also, plastic straining is likely to favor oxygen embrittlement at grain boundaries by a dislocation pumping mechanism [55].

Indications that the rate at which an environment affects the fatigue behavior depends on the strain amplitude have been reported by Wareing and Vaughan [3, 56], Skelton and Bucklow [57], and by Strangman et al [51, 58]. Wareing and Vaughan [3, 56] in discussing crack growth in 316 stainless steel at 625°C find a linear relation between predicted (according to Tomkins' model) and actual crack growth rate, i.e.,

Table 2.1

POSSIBLE TYPES OF METAL/ENVIRONMENTAL INTERACTIONS

Direction of mass transport	Types of metal/environment interactions
from atmosphere to metal	1. formation of a thin adsorbed layer on the specimen
	2. formation of an adherent surface layer of second phase oxidation
	3. formation of a second phase at the surface
	4. formation of second phase particles at/ or grain boundary films
from metal to atmosphere	1. evaporative loss or chemical dissolution of element from specimen to the environment

Table 2.2

SUMMARY OF ENVIRONMENT EFFECTS ON FATIGUE

Type of metal/ environment interactions	Enhancing fatigue resistance	Reducing fatigue resistance
	$N_f\uparrow$ or $\frac{da}{dN}\downarrow$	$N_f\downarrow$ or $\frac{da}{dN}\uparrow$
Adsorbed layer		-reduced surface energy of crack face ($da/dN\uparrow$) -reduce cohesive energy of grain boundary ($N_i\downarrow$)
Formation of an adherent surface layer (oxidation)	-disperse slip ($N_i\uparrow$) -crack blunting $da/dN\downarrow$ -crack closure $da/dN\downarrow$ -prevent crack sharpening in compression (da/dN)	-prevent slip reversal ($N_i\downarrow$) -prevent crack rewelding ($da/dN\uparrow$)
Formation of second phase particles or grain boundary films	-dispersing slip ($N_i\uparrow$)	-easy crack propagation path ($da/dN\uparrow$) -cavitation
Formation of second phase at the surface	-precipitation hardening ($N_i\uparrow$)	-loss of solid strengthening element ($N_i\downarrow$) -cavitation
Evaporative loss or chemical dissolution of element from specimen		-loss of solid strengthening element ($N_i\downarrow$) -Kirkendall vacancies and voids ($da/dN\uparrow$)

$$\frac{da}{dN}_{air} = F_{ox} \frac{da}{dN}_{vacuum}$$

They concluded that if this is due to oxidation enhanced by plasticity, any crack growth additional oxidation contributed growth must be governed by the parameters governing plastic-controlled growth.

In another work, Skelton and Bucklow [57] show that, for the 1/2% Cr-MoV and 1% Cr-MoV steels, the oxide growth rate is acceleratd by ongoing plastic straining. They also found that a fatigue crack surface displays a higher oxide growth rate than a polished and annealed surface.

Strangman et al [51, 58] have studied thermal-mechanical fatigue in nickel-base superalloys B-1900, IN-100, NX-188 and MAR-M200 with NiCoCrAlY coating. The testing was done out-of-phase and in-phase. They found an acceleration of the fatigue crack growth due to oxidation both in the substrate and in the coating for total strains above a certain limit. The previous results show that the environmental effects cannot be explained satisfactorily in terms of metal/temperature/frequency alone, and that a more global approach involving the applied strain ranges has to be considered.

3.0 Review of Experimental Data

3.1 Introduction

Two approaches have been successfully applied to practical fatigue problems: (1) The accumulation of cyclic strains (low-cycle fatigue or total endurance concept) and (2) fracture by crack growth (fracture mechanics concept). In general, these two approaches have been applied independently of each other; materials with no flaw at the initial stage of fatigue are handled by approach 1, and the materials with a flaw or flaws handled by approach 2. To combine the two approaches, the high strain fatigue (HSF) crack growth concept has been proposed. Most of the HSF crack growth data at room and elevated temperatures published between 1965 and 1982 have been reviewed by Shelton [2, 59]. However, this concept is generally overlooked and, as mentioned before, design approaches which use LEFM crack growth or total endurance data are more familiar. As a matter of fact, the thermal-mechanical fatigue problem has also been addressed separately by the two approaches.

3.2 Endurance Data

Steels

It has been pointed out by Udoguchi and Wada [17], Sheffler [20-21], Taira et al {7, 23-24], and by others [18, 25-26, 30-32, 60-62] in studies of thermal fatigue of high temperature alloys, that thermal fatigue lives are generally shorter than isothermal fatigue lives at the mean [7], or the maximum temperature of the cycle [17-18]. Some of the results giving N_f as a function of the plastic strain range for an A-286 [21] steel are reported in Figure 3.1. These results [18, 23-26] show that in-phase cycling (i.e., maximum temperature at maximum tensile stress) is more damaging than out-of-phase (maximum temperature at compressive stress), which in turn caused fracture in fewer cycles than isothermal cycling (whether the latter was based on the mean or maximum temperature). Very similar results were found for 304 stainless steel [18, 25-26, 31]. However, when expressed in terms of total strain range ($\Delta\varepsilon_{tot}$), Kuwabara and Nitta [25] found that the isothermal and out-of-phase changed rank and that this ranking was consistent with the fraction of intergranular cracks that was measured [26].

Superalloys

For the superalloy IN-738, HSF tests between 750 and 950°C in-phase or out-of-phase cycling all gave the same endurances at a given $\Delta\varepsilon_{tot}$ level [48]. Other works on a coated [19] and uncoated nickel-base alloy [146] have shown that out-of-phase cycling is more damaging than in-phase and isothermal cycling, whereas the opposite behavior was observed by Bill et al. [130]. More recently, Nitta et al [66] have recently shown that the endurance of IN-738 depends on the applied strain $\Delta\varepsilon_t$ being the lowest for the isothermal test (T_{max}) at low $\Delta\varepsilon_t$, but the lowest for out-of-phase cycling at high $\Delta\varepsilon_t$ cycled between 400 and 925°C. Their results have also shown that the thermal fatigue life of nine different superalloys was strongly dependent upon tensile and creep fracture ductilities as well as the strength of the alloys.

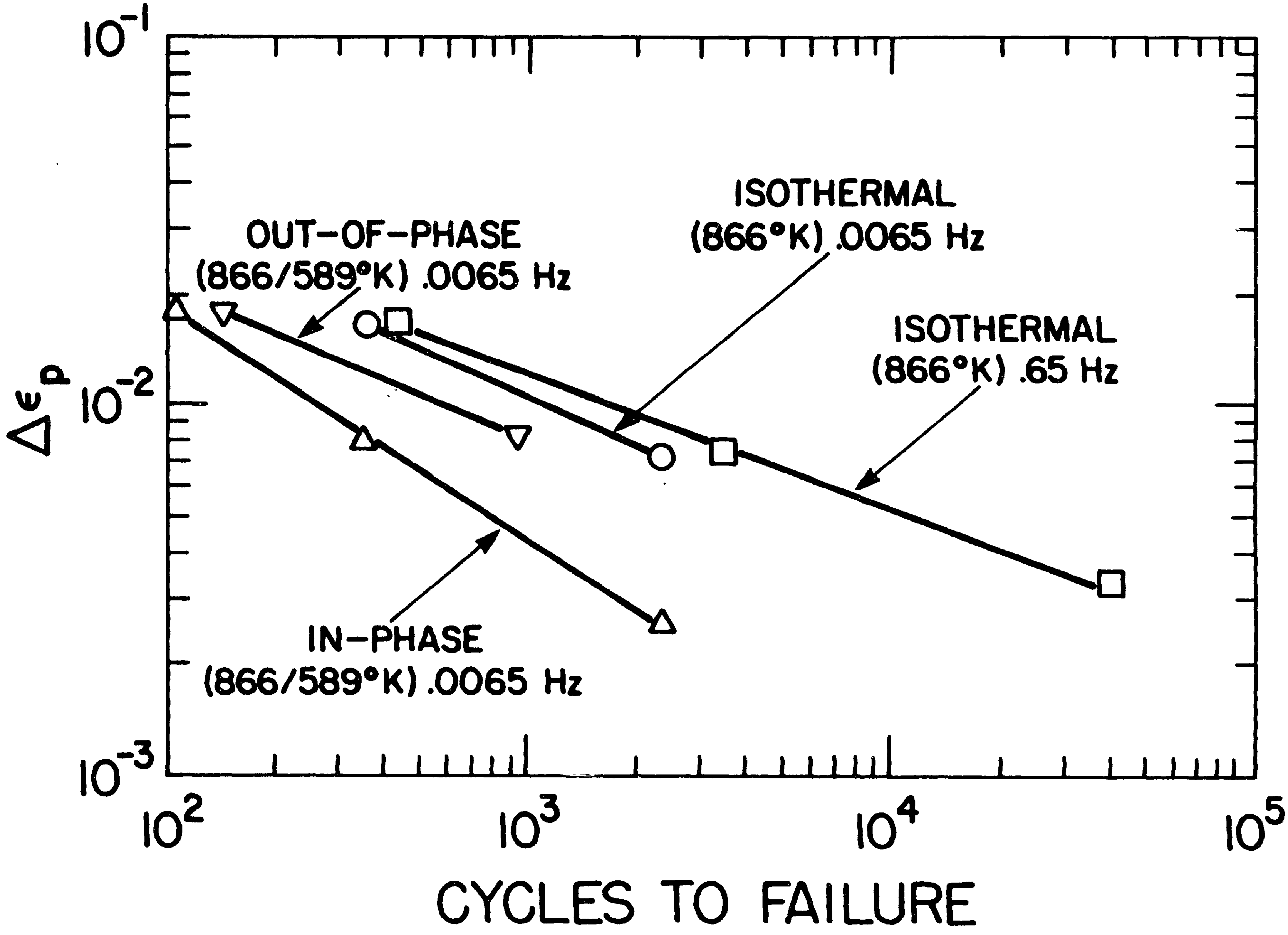


Figure 3.1 Isothermal and TMF lives in A-286[21].

3.3 Fatigue Crack Growth Data

Hardly any data has been gathered on crack growth during thermal-mechanical fatigue under conditions of small plastic strains [27-29, 41, 51] and even fewer in the fully plastic strain ranges [22, 68].

Steels

During TMF cycling of 1/2 Cr-Mo-V steel in vacuum between 250 and 550°C, crack growth rates as a function of ΔK_ε were one-third those of continuous cycling at 550°C at the same plastic strain [63]. In air tests, Koizumi and Okazaki [68] found very little difference in growth rates as a function of ΔJ for 12 Cr-Mo-V-W steel thermal-mechanically cycled between 300-550°C and 350-600°C at several strain ranges. The same conclusions were derived for a 304SS cycled between 300 and 650°C. However, the validity of ΔJ for crack growth under TMF conditions need to be proven [41].

Superalloys

TMFCG tests conducted on conventionally cast Co and Ni base superalloys [27-29, 41, 51] have shown faster crack growth rate then their counterparts in isothermal conditions (at T_{max}). Typical results for a cobalt-based alloy are shown in Figure 3.2 [27] which shows that out-of-phase cycling is more deleterious than in-phase cycling. The TMF crack growth data were expressed as

$$\frac{da}{dN} = C_1 \, (\Delta K_\varepsilon)^m$$

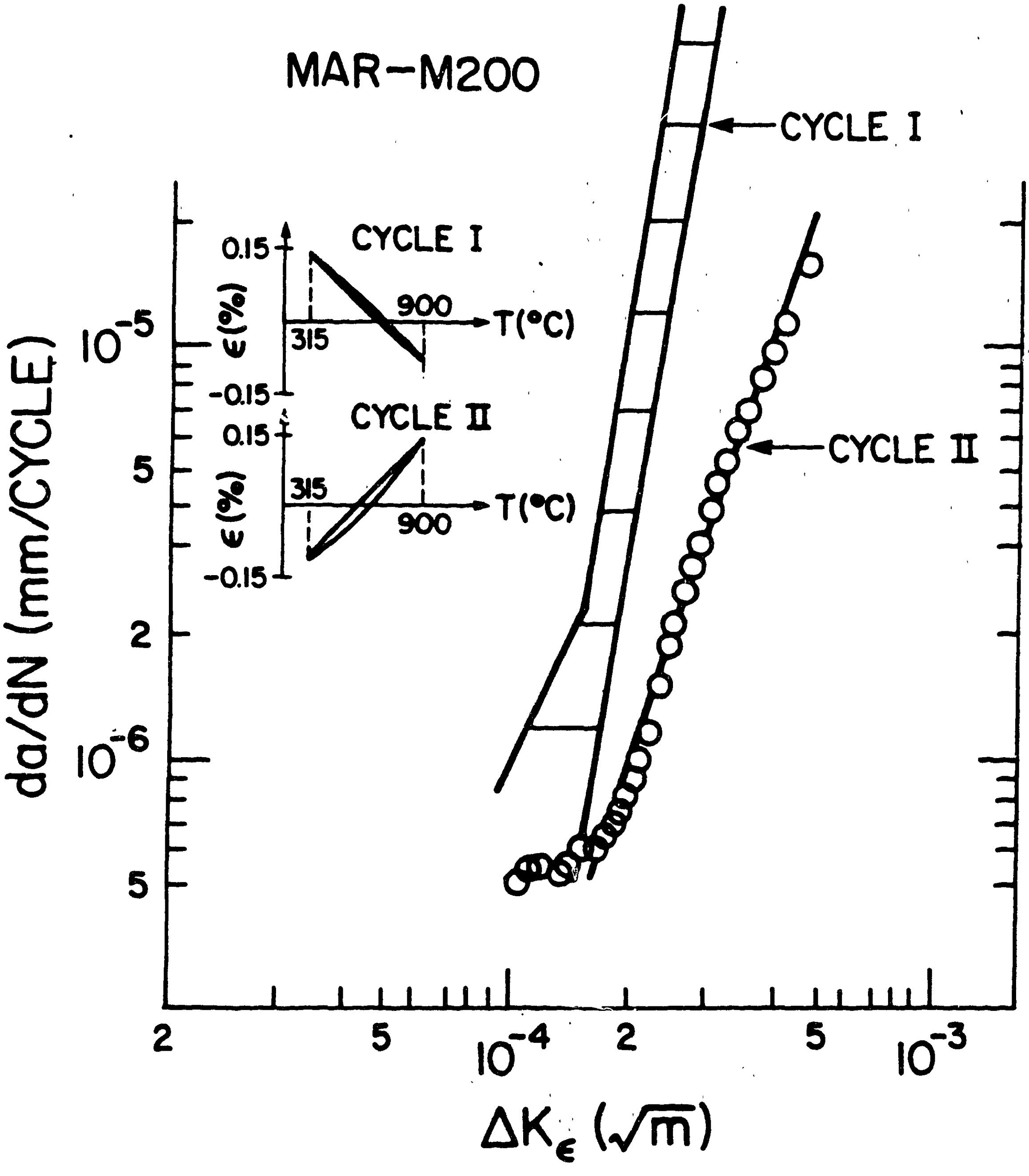


Figure 3.2 Comparison of TMFCG rates for out-of-phase and in-phase cycling in MAR-M200[27].

where ΔK_ε is the strain intensity factor. The index m was typically 3~4 for nickel-base alloy cycled from 980°C and ~6 for the cobalt-base alloy cycled from 425°C to 927°C [27].

Gemma et al [28] have also studied the TMF crack growth behavior of DS Mar-M200 and have found that crack growth rates were a minimum for a DS alloy tested parallel to the direction of grain growth. Crack growth rates prediction as a function of Θ (angle to direction of grain growth) was achieved by normalizing the ΔK_ε with the elastic modulus. Their work also indicated that under strain-controlled conditions, crack growth rates increase with increasing elastic modulus, in contrast to stress-controlled conditions for which the growth rates increase with increasing modulus [68].

In search of a correlation parameter to predict the crack propagation life of combustor liner material (Hastelloy-X), Meyers [41] has reduced the TMF data using ΔK, ΔK_ε, ΔJ, ΔCOD and Tomkin's model. All these approaches failed to correlate the TMF data within a factor of 5, which made them unattractive in predicting TMF crack growth. Furthermore, he showed that for such a parameter to exist, TMF must not involve mechanism of failure otherwise not present in isothermal testing.

<u>Crack Growth During Thermal Stress Fatigue</u>

Several temperature cycling experiments which consist of alternating plunging wedge-shaped disks into hot and cold fluidized beds have been carried out on various nickel-base alloys [8-12, 47, 70-71, 80, 82, 90]. For such geometries there is a fairly large (2.5-7 mm) range of uniform crack growth [12, 71]. Figure 3.3 shows that the

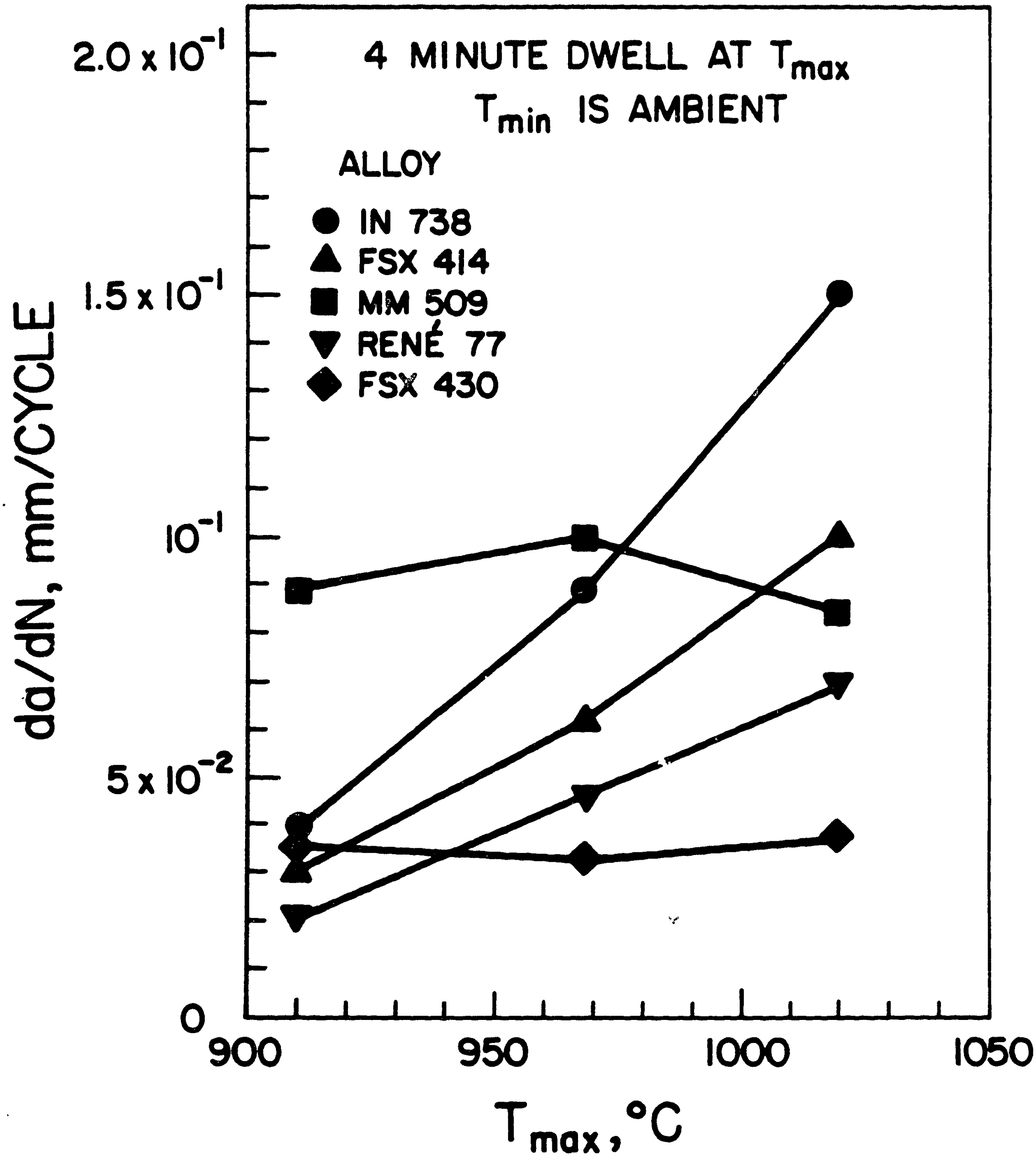


Figure 3.3 Variation of steady-state crack growth as a function of the maximum temperature in thermal shock experiment[70].

variation in uniform growth rate over the uniform range (in tapered disc experiments on several alloys) increases with increasing maximum temperature [70]. An increase in dwell time caused further crossovers in propagation rates. In these alloys, it has been shown that changes in growth rate reflects a transition in cracking mode [27, 29, 47] from trans to intergranular cracking.

3.4 Metallographic and Fractographic Data

Steels

Taira and Fujino [23-24] found from their microscopic observations on 304 SS cycled under TMF conditions that thermal cycling leaves residual grain boundary sliding which accumulates unidirectionally, in contrast to the case of isothermal fatigue. Also grain boundary cracking was often observed in the vicinity of triple points of boundaries which accomodate the strain concentration due to grain boundary sliding. Based on their finding, they concluded that the acceleration of damage is closely related to the accumulation of grain boundary cavitation in both in-phase and out-of-phase cycling. They proposed a mechanism of out-of-phase cavitation damage that involved accumulation of unreversed compressive grain boundary displacements in a low ductility material where the displacement could not be fully accommodated by intergranular deformation.

Superalloys -- cast

High temperature fatigue experiments of nickel-based superalloys have shown that crack initiation and growth behavior are very dependent upon metallurgical variables such as carbide spacing, interdendritic spacing and grain size. It is impossible to provide a comprehensive survey here. A useful starting point is the work of Pineau et al. [42, 50, 52] and Woodford and Mowbray [71].

Thermal fatigue experiments on tapered disks of several alloys [70-71] have suggested that coarse grains at the center of the cast disks may be responsible for a deceleration in crack growth rate (see Figure 3.3), but for two alloys (IN 738 and FSX-430), larger grain size initiated cracks sooner. Initiation was intergranular, otherwise mixed, and in the transgranular mode an interdendritic path was preferred. On a $\gamma/\gamma'-\zeta$ eutectic alloy cycled under TMF conditions, Sheffler and Jackson [72] reported similar observations, i.e. preferential cracking along Ni_3Nb dendrites. In FSX-414 and FSX-430 cracking took place between $M_{23}C_6$ carbides, whereas in others (Mar-M509 and In-738) cracking of MC carbides occurred. Rene 77 failed in a transgranular manner.

Beck and Santhanan [73] demonstrated that thermal-stress fatigue resistance of a cast cobalt-base alloy can be significantly improved by changing the casting variables so as to increase the spacing of the dendritic arms in the microstructure. It is noteworthy that crack initiation periods were lengthened and propagation rates were reduced by this technique. Their work and similar work by Bizon and Spera [9] and Howes [10, 47] have shown that it is possible to deal directly with methods for improving the thermal-fatigue resistance of materials. It should be pointed out here that fluidized bed tests, as in the above, are undertaken because with these alloys, it is not yet possible to predict thermal fatigue lives from isothermal data [70].

Superalloys--directionally solidified

In a directionally solidified ingot, IN738, the greatest resistance to crack growth occurred at 45° to the columnar grain direction because there were no easy paths linking the carbides. This is in contrast with the thermal-mechanical fatigue tests performed on directionally solidified (DS) Mar-M200 which show [28] that the growth rates were a minimum for the alloy tested parallel to the direction of grain growth (i.e., crack propagation normal to the direction of grain growth). Also, the growth rate was shown to increase for the orientations in the sequence θ =15°, 30° and 90°, and 45° until equivalence was reached with rates for conventionally cast B-1900. Similarly to Rene 77, DS Mar-M200 crack growth was transgranular for all orientations. At low strain intensity ranges (ΔK_ε) the crack propagation path was smooth but as ΔK_ε was increased the amount of interdendritic cracking increased, resulting in a much rougher fracture surface. The smooth-to-rough fracture surface transitions were rationalized in terms fo the orientation dependence of crack opening displacement for a given ΔK_ε. In addition, it was observed for Rene 77 that precipitate in the bulk specimen cycled in and out of solution and that a zone denuded of γ' precipitate formed by a diffusion process along the crack faces [71]. Other work on oxide-dispersion strengthened materials showed that cavitation developed at oxide particles so that cracking along grain alignment was less than the growth rate perpendicular to it [70].

It has been mentioned before that creep/TMF and simulated creep/TMF tests on DS Mar-M200 and conventionally cast B-1900 were phenomenologically indistinguishable [29]. However, the crack morphology of the simulated creep/TMF specimens differed from that of the creep/TMF specimens. The former had a crack morphology that was typical of plane stress conditions, that is, short hairline cracks that surrounded the dominant crack. In addition the fracture surface showed changes in crack planes and growth along steep angular planes. The fracture surfaces of the latter were more typical of crack growth under plane strain conditions.

Superalloys--crack morphology

In TMF tests of nickel-based alloys, Nitta et al. [66] and Meyers [41] have found that the crack path for in-phase cycling was planar, rough and intergranular, which is similar to isothermal tests at the peak temperature. Out-of-phase cycling has shown transgranular fracture with fatigue striations and a small amount of intergranular growth. Isothermal tests at low temperature showed transgranular fracture. The degree of nonplanar growth, the extent of surface roughness and the mode of growth for the out-of-phase TMF tests fall between the features observed for the high and low temperature isothermal tests described above. This observation suggests that crack growth under TMF conditions is, in some sense, an average of that experienced in isothermal tests over the temperature range of the TMF tests [41]. This evidence offers hope that some type of superposition model may eventually predict TMF crack growth. Crack growth in in-service combustor liners tends to be nonplanar, of a moderate level of surface roughness, and chiefly transgranular, similar to the TMF tests [41]. However, as seen before, high temperature isothermal growth tends to be planar, very rough, and intergranular. This observation reinforces the notion that isothermal tests run at the peak service temperature do not duplicate service conditions as well as TMF tests.

3.5 Discussion

The endurance and fatigue crack growth data presented and reviewed in the previous section have shown there is no general trend as to the severity of damage associated to in-phase and out-of-phase cycling. There is no hard and fast rule for relating the thermal-mechanical data to the isothermal testing data. There are many uncertainties in comparing total endurances only; for example, the number of cycles to initiation may differ, and cracking may be intergranular in isothermal tests but transgranular in the cyclic temperature tests. Obviously low endurance data will depend on initiation/propagation ratios, ductility including sensitivity to strain rate effects, T_{max} and ΔT, σ_{max} and σ_{mean} and microstructure.

We have seen that for all materials tested under TMF conditions, the mean stress associated with in-phase cycling is compressive whereas it is tensile for out-of-phase cycling. As also discussed earlier, the presence of a mean stress is not solely responsible for the shortening of fatigue life and the mean stress has to be taken into account with the strain ranges, that is the transition life (N_t) which is defined as the point of intersection of the elastic and plastic lines ($\varepsilon_p = \varepsilon_e$) on a strain cycle to failure plot. If the plastic strain range is large with respect to the elastic strain, the magnitude of the stress is likely to be controlled by plastic flow considerations and the magnitude of the mean stress (σ_m) is inconsequential. At small plastic strain (with respect to ε_e), elastic effects dominate and a compressive mean stress is likely to retard the amount of opening of the crack or lower the effective stress for crack initiation and the converse is true for tensile mean stress. On the basis of the previous arguments, life predictions in terms of ε_{tot} - N_f should be more adequate than prediction based on ε_{in} vs N_f. This has been demonstrated for nickel-base alloys by Nitta et al. [66]. This fact implies that thermal fatigue life of superalloys depends not only upon the ductility but also upon the strength (elastic strain). Several reasons can be given of for the dependence of the thermal fatigue of superalloys on tensile strength [66]:

1. The transition life, which is the life from the inelastic strain dependence to the elastic strain dependence, is of the order of 10^1 for high strength superalloys. This means that life strongly depends upon strength even in the low-cycle life region ($N_f < 10^4$ cycles).

2. For high-temperature low-cycle fatigue of superalloys failure has often been shown to be governed by the product of tensile stress and inelastic strain range, i.e., the inelastic strain energy [74].

3. The failure life of superalloys depends more upon the maximum tensile stress and the depth of oxides, than on ε_p, because crack initiation is caused by grain boundary oxide cracking on the surface [49-50].

The dependence of thermal fatigue on strength may not be restricted to superalloys. Jaske [61] found that in order to correlate in-phase, out-of-phase, and isothermal fatigue test data for a low carbon steel, a knowledge of both cyclic stress and cyclic strain was necessary. On the basis of stabilized stress range versus cycles to failure, little difference was observed in the behavior of specimens subjected to in-phase and out-of-phase temperature and strain cycling.

For most stainless steel, N_t is of the order of 8000 cycles in the temperature range covered by TMF tests. This explains why the mean stress has little effect on the fatigue life and why in-phase cycling is more damaging than out-of-phase cycling, as for the case of 304 and 316 SS. Grain boundary sliding is the predominant flow mechanism in 304 and 316 SS (in the temperature range 300-650°C) and it has been shown to be very important for in-phase cycling [23-24]. As the strain amplitude decreases, GBS will be less important and σ_m is likely to be of little influence until $\varepsilon_e \simeq \varepsilon_p$. Nitta et al. [25-26, 66] and Taira et al. [23, 24] have indeed observed that the difference between the in-phase and out-of-phase life is decreasing with decreasing strain; being the same near the transition life.

The behavior of Hastelloy-X is also consistent with the above arguments. At high strain ($\varepsilon_p > 0.005$) in-phase cycling leads to shorter life [30] and fast crack growth rates, whereas out-of-phase cycling gives rise to faster crack growth rate and shorter lives at low strain ($\varepsilon_p < 0.0035$) [75]. It should be pointed out that the transition life is approximately equal to 800 cycles ($\varepsilon_e \simeq 0.0035$) at 800°C and that Hastelloy-X doesn't present a ductility minimum vs temperature as for most high strength nickel-base alloys.

We have seen that the influence of σ_m depends on the values of ε_p at the transition life (N_t) and that σ_m is of little influence if plastic flow dominates ($\varepsilon_p/\varepsilon_e > 1$). Therefore any mechanism which reduces the efficiency at which plastic flow occurs will lead to an increased importance of σ_m because it is more difficult to relax it. This is consistent with the fact that it has been observed that when the temperature range embraces a ductility minimum in the material, out-of-phase thermal cycling gives lower endurances than isothermal and in-phase cycling [63]. This last observation also serves to explain why high strength nickel-base alloy displays faster crack growth rates and shorter life under out-of-phase TMF cycling; most of them having a ductility minimum in the temperature range covered by the tests.

The foregoing discussion has dealt with the requirement for predicting the total number of cycles to failure (N_f), and it is pertinent to question whether the predicted N_f should be taken, not as cycles to fracture but as cycles to some arbitrary load drop. In the case of thick specimens, clearly the N_f values would be different, since they would incorporate both initiation and propagation of a macroscopic crack. Thin components, however, are likely to limit crack propagation severely, so that the N_f value may be reasonably defined as cycles-to-initiation.

Considerable research effort has been directed at measuring rates of fatigue crack propagation in appropriate materials. The effect of increasing temperature upon the fatigue crack propagation rate is generally to cause an increase at a given ΔK, the extent depending on the materials. In order to obtain some insight into the factors governing the temperature dependence of fatigue crack propagation, the crack growth rates at fixed ΔK are often plotted as a function of reciprocal temperature and normalized with respect to the elastic modulus [4-5, 28, 76] (see Figure 3.4). The rate dependence at low temperatures has been shown to be much lower than at high temperatures. Conventional fracture mechanics indicates that crack growth rates are inversely proportional to elastic modulus and to fatigue flow stress. As shown by

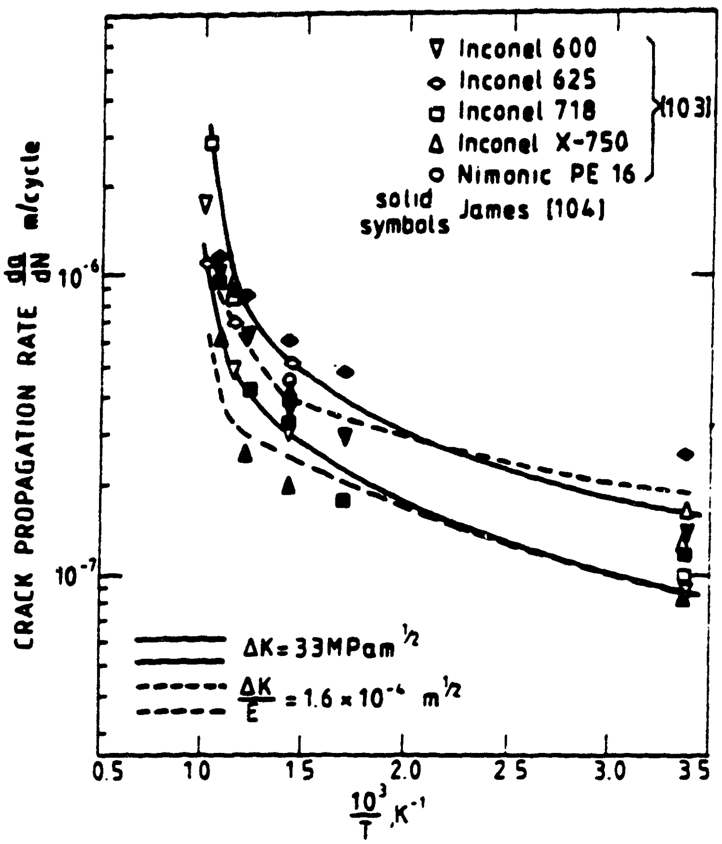


Figure 3.4 Temperature dependence of fatigue crack propagation in nickel-base alloys[5].

Lloyd [5] and Shahinian [76], the da/dN versus 1/T curves at fixed $\Delta K/E$, do not increase as rapidly as do similar curves at fixed ΔK. This indicates that the drop in flow stress with temperature is making a contribution at high temperatures.

Changes in the flow properties of the materials at the crack tip may be expected to induce changes in the propagation rate. A decrease in the work hardening exponent of the crack tip material caused by a decrease in strain rate or by an increase in temperature, would cause an increase in crack propagation rate. Similarly, stress relaxation reduces the effective value of σ_y. As seen before, the environment might increase the flow stress at the crack tip by oxide strengthening or reduce it by solid solution element depletion. Also, any of the possible effect of environment on the fatigue behavior might mask or enhance the effect of σ_m and ductility minimum.

Analysis of the data by Nitta et al. [26, 66], Jaske [61] and others, shows that thermal-mechanical life depends upon the tensile and fatigue ductilities as well as on the strength of the alloys. Hence, thermal fatigue life may be expected to be correctly predicted by a method which would take into account both the cyclic strength and ductility.

4.0 Experimental Procedure

4.1 Thermal-Mechanical Fatigue Testing Requirement

The testing capability and testing conditions for TMF testing of specimens are those that allow the duplication of in-service conditions. The requirements for testing facilities and test specimens for crack growth TMF testing are:

1. Strain control to simulate the thermal loading of the component since most high temperature components are under strain-controlled conditions. The mechanical strain (total strain minus thermal strain) should be controlled.
2. Strain hold time to simulate steady state conditions undergone during normal operation of nuclear reactor and gas turbine engines. During this period materials are undergoing creep and stress relaxation behavior.
3. Crack length measurement capability to obtain crack growth data in TMF conditions.
4. Controlled transient heating and cooling to simulate a component strain/temperature phase relationship (Figure 4.1).
5. Compressive load-carrying capability of the specimen to sustain compressive stresses which develop in the heating period of the cycle.

TMF crack propagation should mostly include testing of the base metal, since this type of cracking is readily studied by a fracture mechanics approach. The amount of component life spent in starting a

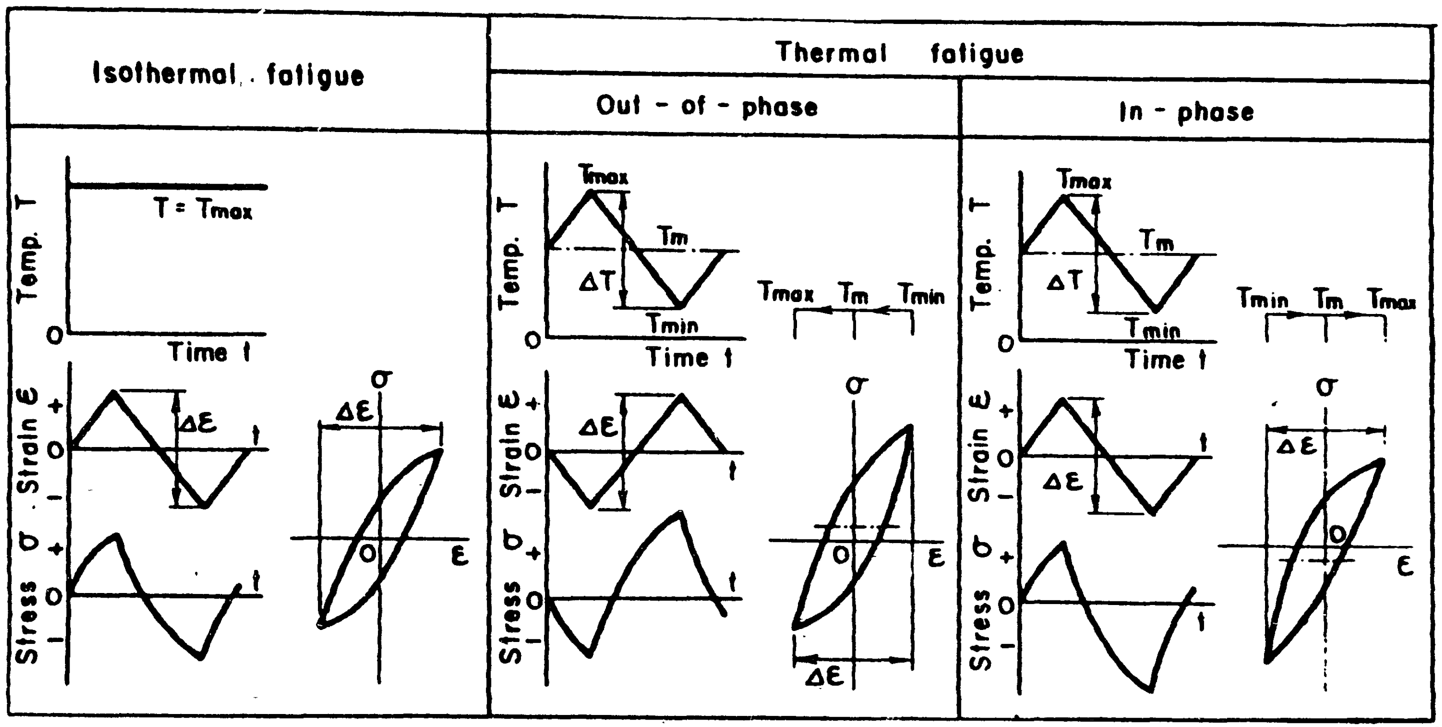


Figure 4.1 Schematic diagram showing waveforms of temperature, strain and stress in thermal and isothermal fatigue tests.

crack in the coating has to be addressed by crack initiation techniques. TMF tests should include several testing conditions in order to simulate the various strain-temperature relations at various locations in an actual component, as for example, a turbine blade.

Testing should also include both isothermal tests at the minimum and maximum temperatures of the TMF tests. The following variables should also possibly be included in the testing procedures:

1. total strain range, 0.10 to 1.0%
2. mean strain, -0.5 to 0.5%
3. crystal orientation for directionally solidified and/or single crystal materials
4. strain rate, set by the maximum transient heating and cooling rates that can be experimentally obtained.

In an extensive review, Hales [77] has described the important criteria which used to be considered when designing an experimental program to investigate fatigue at elevated temperature. He also pointed out some exhaustive reviews [78-79] which covered the various techniques available and critically examined some of them in the light of the demands placed on any testing program by modern design requirements and recent developments in understanding the physical processes in the phenomenon of high temperature fatigue. However, the present authors believe that there is still a need for a critical review of the various methods of testing to take into account the peculiarities of thermal-mechanical fatigue with a special emphasis on the recording of cyclic stress-strain behavior of various specimen geometries. The reader is referred to the Appendix I for a detailed discussion of the thermal-mechanical fatigue methods of testing.

4.2 Materials

The materials selected for this study of TMF are Hastelloy-X, Inconel X-750 and B-1900+Hf. These materials were chosen because they covered a broad range of strength and ductility. Table 4.1 gives the chemical composition and heat treatment conditions of the materials tested.

Hastelloy-X is a nickel-base solid solution strengthened superalloy. The material has a FCC gamma matrix and is strengthened by solid solution strengthening and grain boundary carbides. The solution strengthening elements are Cr and Mo. Hastelloy-X has a very high Mo content, which causes the carbides to be mainly M_6C carbides. The material also contains $M_{23}C_6$ carbides which form at temperatures up to 982°C and during cooling. The $M_{23}C_6$ carbides usually form in the grain boundaries. The material used in this study was in the form of 1 inch (25.4 mm) rod. The structure of the material was documented in the fully heat treated condition. Figure 4.2 shows the microstructure of the rod in the transverse direction. There was no major difference between the transverse and longitudinal directions. The microstructure is made up of two different grain sizes. There are large grains, 130 μm in diameter, and small grains, 50 μm in diameter. The average grain size is about 80 μm. The small grains are not recrystallized. This type of grain structure is common in solid solution strengthened nickel-base alloys because it is difficult to obtain a fuly recrystallized grain structure and prevent the formation of embrittling phases at the same

Table 4.1

(a) Chemical Composition

Material	C	Si	Mn	P	S	Al	B	Nb	Co	Cr	Fe	Hf	Mo	W	Ta	Ti	Zr	Bi+Pb
Hastelloy-X	0.10	1.0	1.0	0.04	0.03	-	-	-	1.5	22	18.5	-	9.0	0.6	-	-	-	-
Inconel X-750	0.04	0.2	0.5	-	-	0.7	-	1	1.0	12.5	7	-	-	-	-	-	-	-
B-1900+Hf	0.09	-	-	-	-	6.07	0.016	0.08	9.91	7.72	0.7	1.19	5.97	0.04	4.21	0.99	0.04	0.2

(b) Heat Treatment

Material	Heat Treatment
Hastelloy-X	1177°C x 2 hrs., AC
Inconel X-750	1150°C x 4 hrs. AC, 850°C x 24 hrs. AC, 700°C x 20 hrs. AC
B-1900+Hf	1080°C x 4 hrs. AC, 900°C x 10 hrs. AC

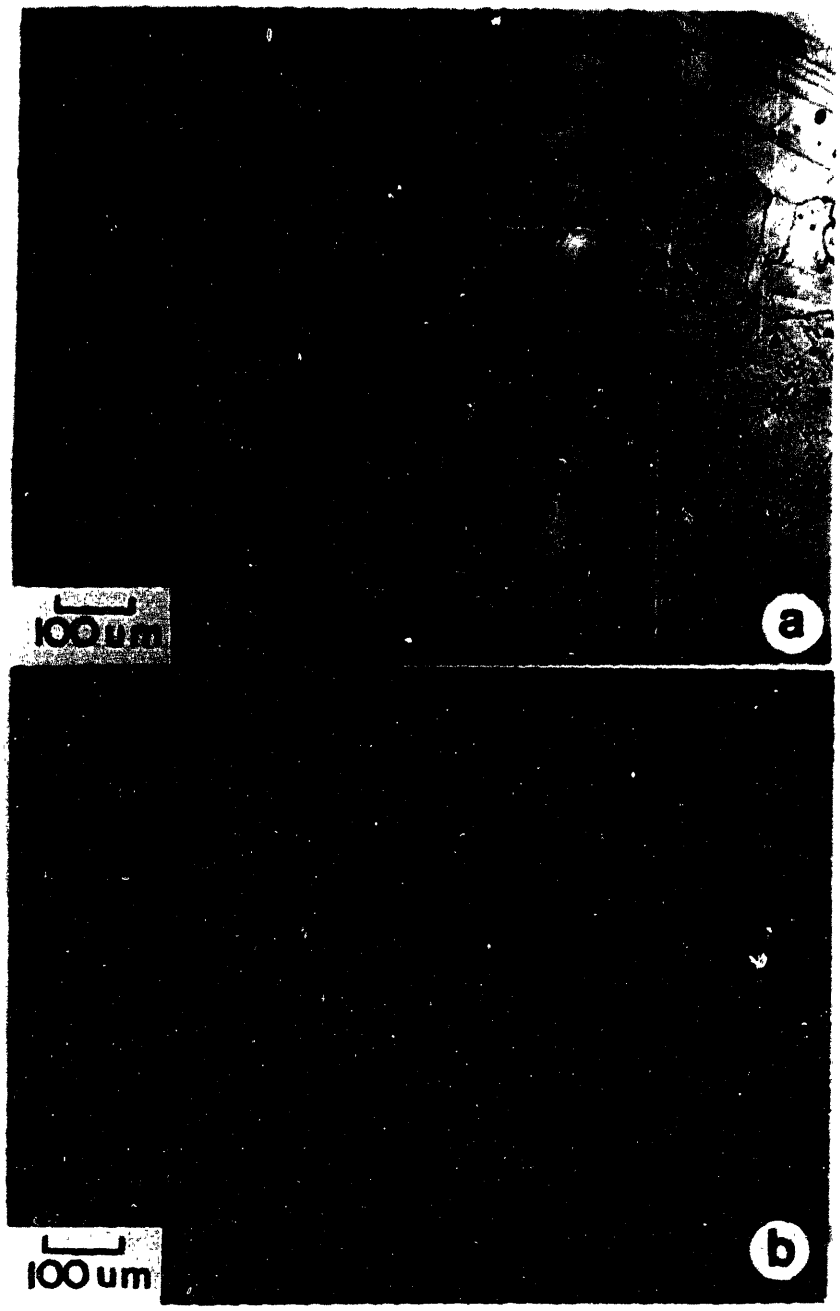


Figure 4.2 Microstructure of Hastelloy-X. Grain size in (a) the transverse direction, (b) longitudinal section.

time. In Figure 4.2, the large 6-10 μm particles are M_6C prime carbides.

Inconel X-750 is a corrosion and oxidation resistant material with good tensile and creep properties at elevated temperatures. Typical applications include land-based gas turbine parts, nuclear reactor springs, bolts, bellows and forming tools. The temperature range of interest is from 300 to 650°C. The material was supplied by International Nickel Company [122] in the form of12.7mm thick plate in the annealed condition. It was then given a two-stage heat treatment (see Table 4.1). The grain size, as seen on micrograph 4.3, is about 130 μm and is uniform in all directions. The two-stage heat treatment produces a bimodal distribution of γ′ phase: a coarse γ′ phase distributed uniformly in the grains and about 0.6 μm in size, and a fine distribution of γ′ about 300Å in size. The total volume fraction of γ′ phase is estimated at 18%. MC carbides with little or no orientation with the matrix can be seen in micrograph 4.3. At the grain boundaries, there is a continuous network of $M_{23}C_6$ carbides.

The B-1900+Hf material for this program was taken from the same heat of a special quality melt of B-1900+Hf obtained from Certified Alloys Products Inc. [123] Long Beach, California. The structure of the material was documented in both the as-cast and fully heat-treated conditions. The following observations were made:

1. The grain size is about 1 to 2 mm (Figure 4.4a).
2. The fully heat treated material showed the γ′ size to be about 0.9 μm (Figure 4.4c).
3. The structure has an interdendritic spacing of about 100 μm

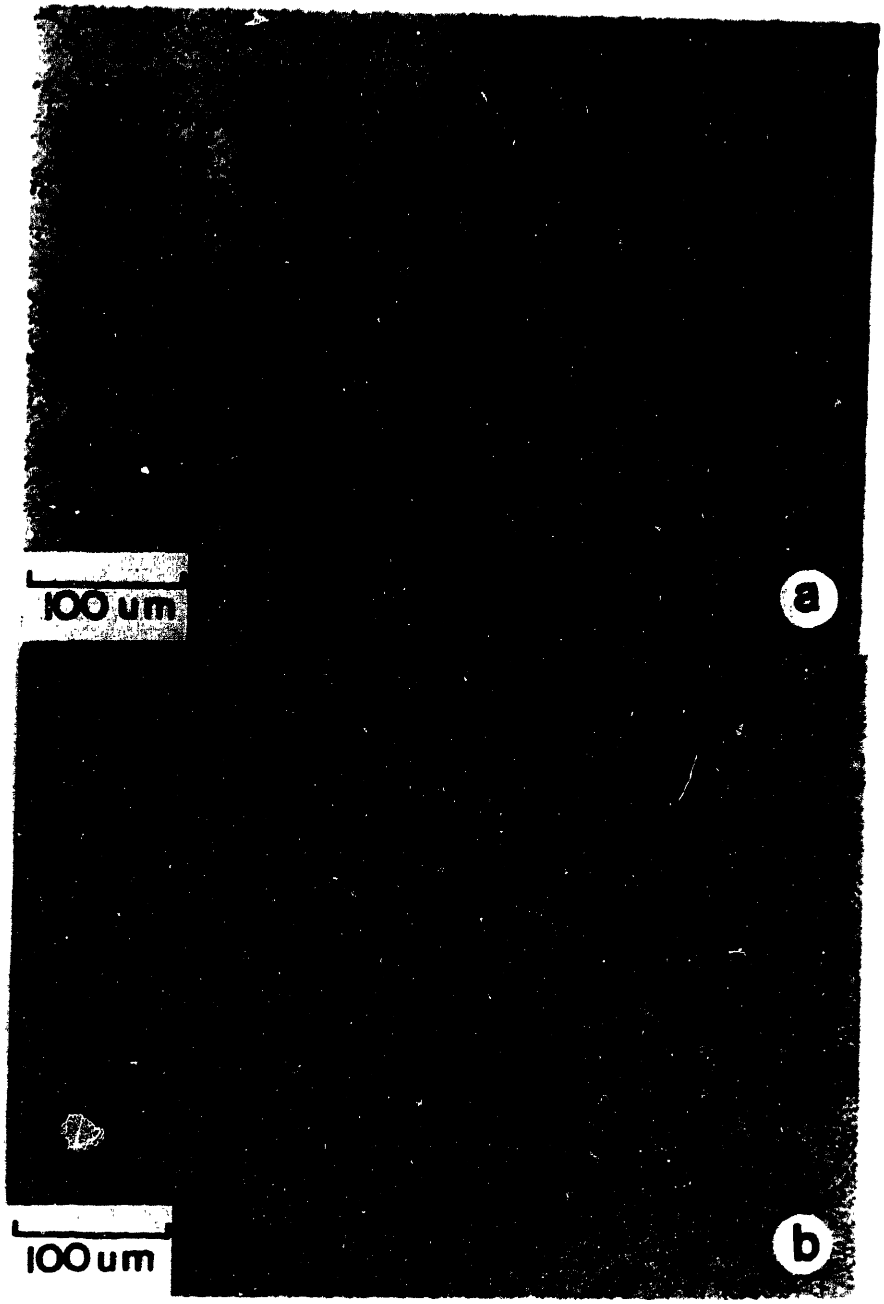


Figure 4.3 Microstructure of Inconel X-750(fully heat treated) Grain size and carbides in (a) transverse direction, (b) longitudinal section.

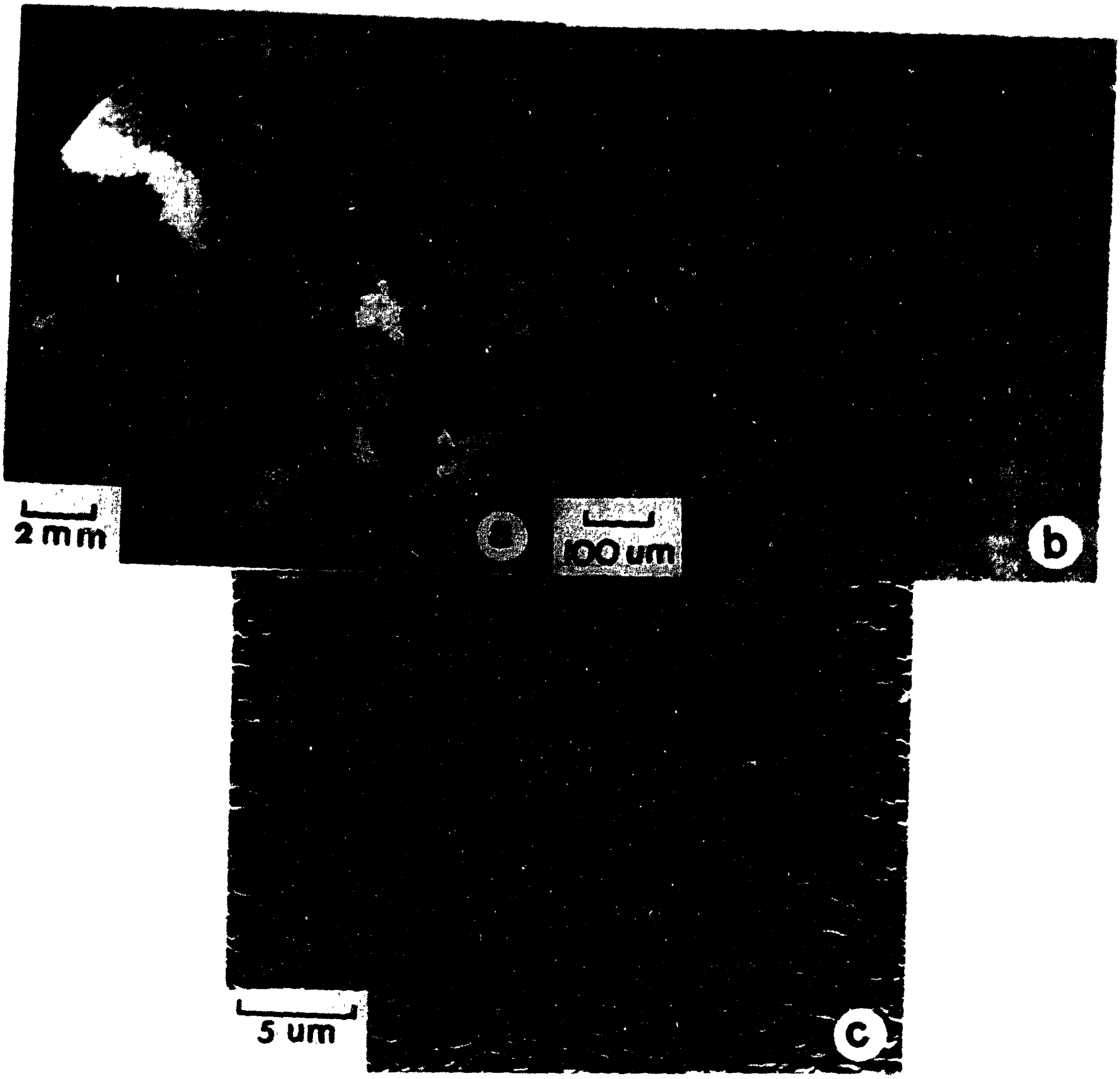


Figure 4.4 General microstructure of B-1900+Hf fully heat treated. (a)Grain size, (b) interdendritic spacing, fine and coarse γ , γ' size(0.9 μm).

with islands of γ'-eutectic surrounded by zones of fine γ' (Figure 4.4b).

4. MC carbides near the coarse γ' islands are present.

4.3 Specimen Geometry

Specimen geometries used for tensile and creep tests are shown in Figure 4.5 [123, 125]. The isothermal strain controlled fatigue specimen geometry used to provide a baseline for life prediction (initiation) is shown in Figure 4.6 [123, 125].

All the specimens used for TMF low cycle fatigue and TMFCG had a rectangular cross section of 11.7 x 4.4 mm^2 (Figure 4.7). A starter notch approximately 1 mm deep cut by electro-discharge machining was used for the TMF crack growth tests. These specimens were then precracked in fatigue at 10 Hz at room temperature under a ΔK of about 20-25 MPa $\sqrt{m}$. The test section of each specimen used for cyclic properties measurements was polished with successively finer grades of silicon-carbides paper to produce a bright finish, with finishing marks parallel to the longitudinal axis of the specimen. Specimens were degreased with trichloroethylene, followed by reagent grade acetone before being heated to temperature.

4.4 Apparatus and Testing Conditions

The apparatus used in this study was a computer-controlled thermal fatigue testing system, which consisted of a closed-loop servo-controlled electro-hydraulic tension-compression fatigue machine, a

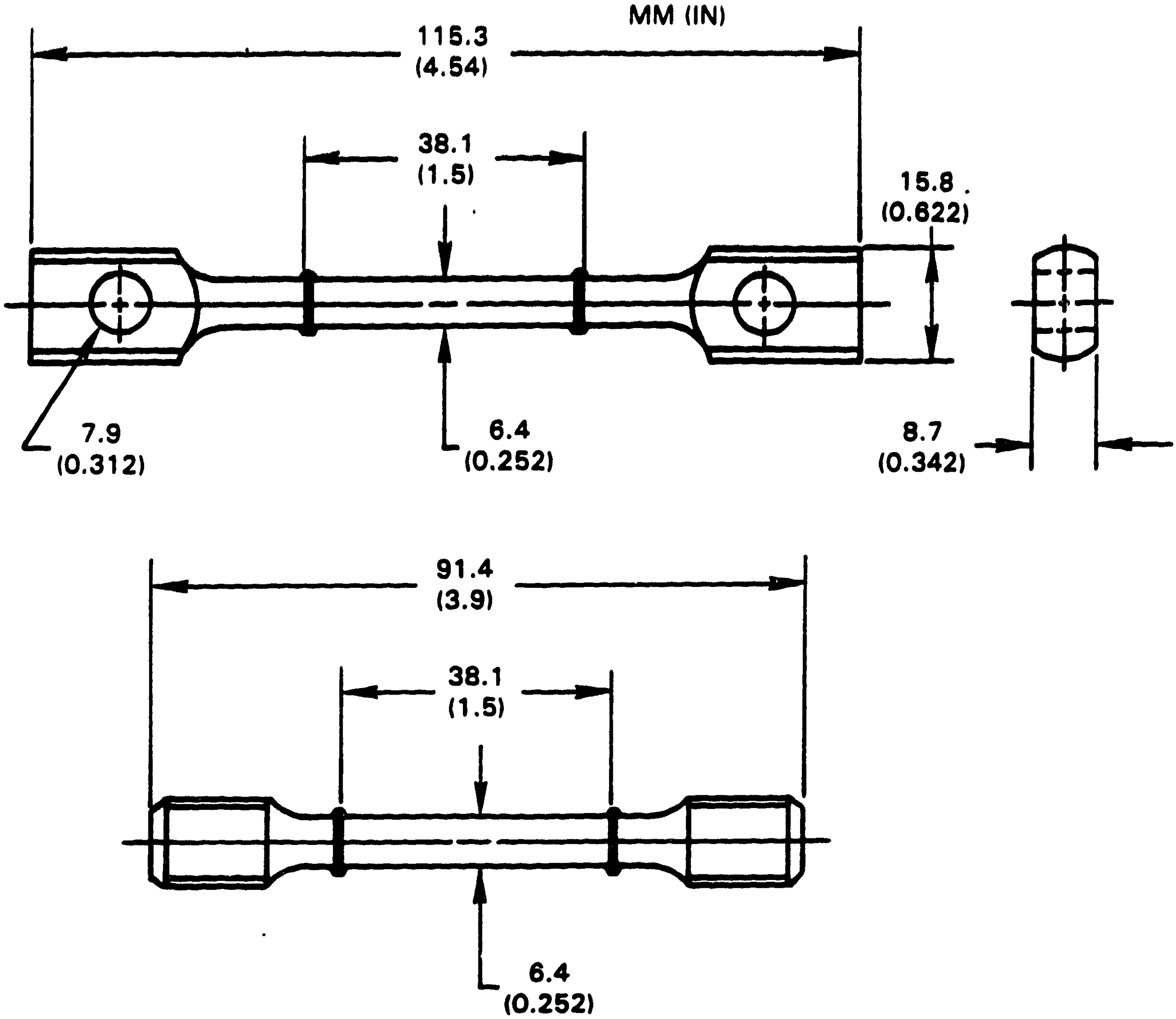


Figure 4.5 Test specimens for monotonic tensile and creep testing.

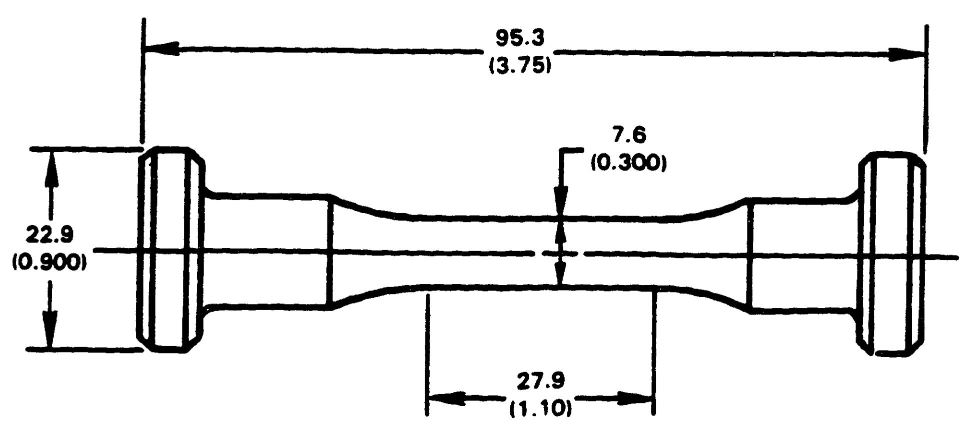


Figure 4.6 Test specimen geometry for low cycle fatigue testing.

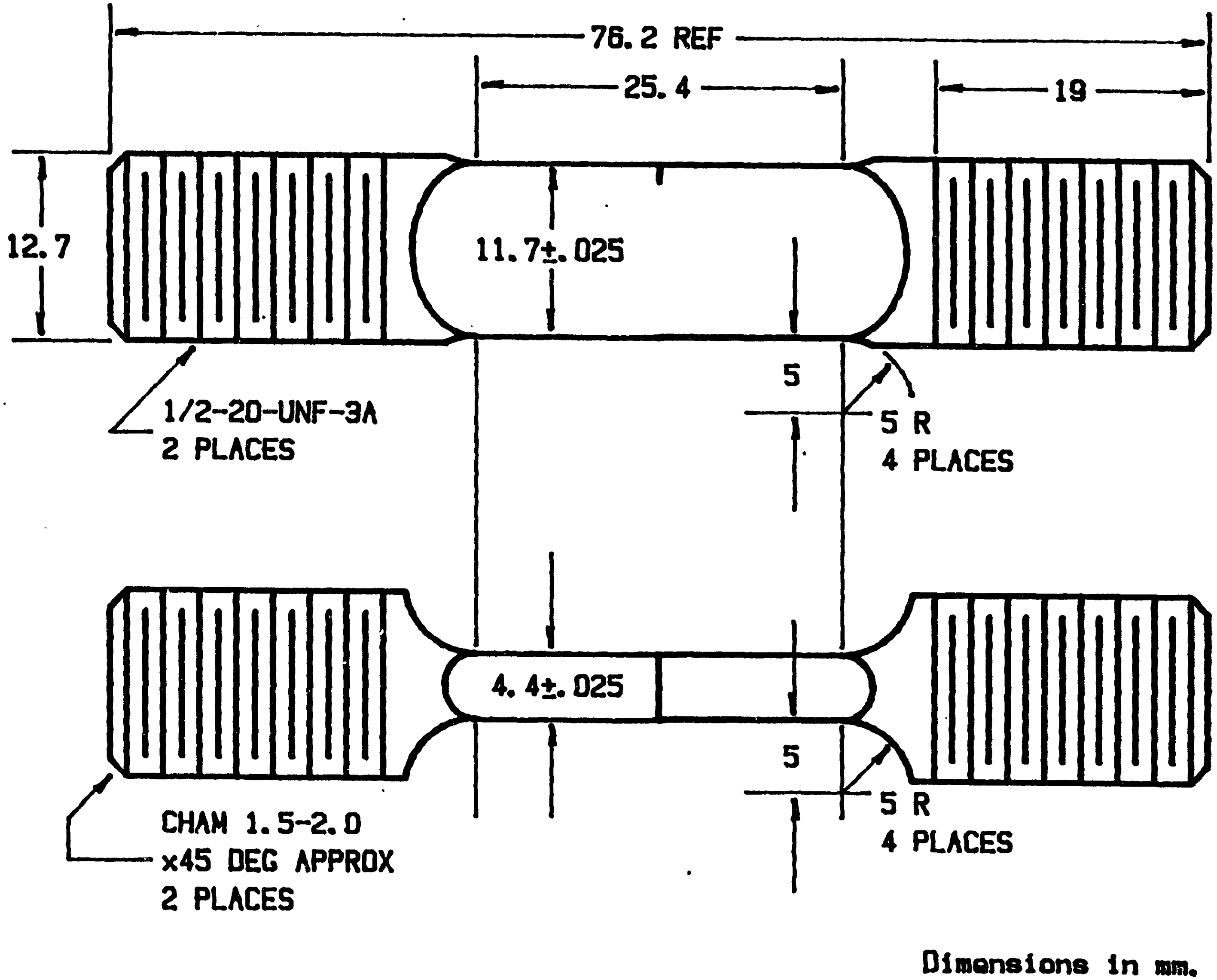


Figure 4.7 Test specimen geometry for TMF testing.

high frequency oscillator for induction heating, an air compressor for cooling and a mini-computer. Appendix II describes in detail the system and its components.

The system is capable of testing specimens of different sizes and configuration (SEN, CT, hollow tube, etc.) up to loads of 25,000 lbs. The specimen alignment is assured by the use of a Wood's metal pot. Because a D.C. potential drop technique was used to monitor crack growth, the lower grip is electrically insulated from the system by means of a ceramic coating. The ends of the grips are water cooled by copper coils.

Temperature was measured with 0.2 mm diameter chromel-alumel thermocouples which were spot welded along the gauge length. By computer controlling, the temperature in the gauge length was maintained within ±5°C of the desired temperature for both axial and transverse directions over the entire period of the test. Temperature and strain (or stress) were computer controlled so that they were in-phase or out-of-phase for the same triangular wave shape. Therefore, specimens were in tension at low temperature and in compression at high temperature under the out-of-phase cycling, and vice versa under the in-phase fatigue. The temperature range in the TMF tests was 300 to 650°C for Inconel X-750, and 400 to 925°C for Hastelloy-X and B-1900+Hf. The tests were carried out at a frequency of 0.0056 Hz (1/3 cpm) and were run at a R-ratio ($\sigma_{min}/\sigma_{max}$ or $\varepsilon_{min}/\varepsilon_{max}$) of -1 or 0.05. Isothermal fatigue tests were also conducted under the same frequency at T_{max} for comparison with the results of TMF tests. All the tests were carried out in air. Table 4.2 summarizes the experimental conditions. When possible, at least two tests at each condition were performed to insure repeatibility of the results. The software developed to run the isothermal and TMF tests, as well as the programs used to analyze the data, are described in Appendix III.

To measure the cyclic stress-strain behavior of B-1900+Hf, the strain increment technique was used. In this technique, the strain range is increased by about 10% after reaching saturation, defined as no change in stress range in 50 consecutive cycles.

There have been few attempts to measure crack length in TMF cycling using the potential drop technique [22, 49]. The method has proven to be satisfactory in isothermal conditions and it can be used for TMF testing, provided that the electrical noise is adequately filtered and the calibration curve properly corrected to take into account changes of potential with temperature. The procedures to: (1) correct the potential signals; (2) get the a/w vs N curve; and (3) analyze the potential signal to measure the closure stress, are described in detail in Appendix II.

This system, by contrast to optical measurements and compliance methods, has the capability of monitoring the crack extension during a single cycle (see Appendix II). Therefore, detailed analysis of the crack growth process can be performed and this is particularly important in trying to determine the mechanisms involved in TMFCG.

Table 4.2

Materials and Their Testing Conditions

(a) Hastelloy-X (TMFCG)

Frequency (Hz)	Temperature (°C)	Strain Range (ε_{tot} %)	R-ratio ($\varepsilon_{min}/\varepsilon_{max}$)
0.0056	400	0.25	-1
0.0056	400	0.50	-1
0.0056	925	0.25	-1
0.0056	925	0.50	-1
0.0056	400-925(In-phase)	0.25	-1
0.0056	400-925(In-phase)	0.50	-1
0.0056	925-400(Out-of-phase)	0.25	-1
0.0056	925-400(Out-of-phase)	0.50	-1

(b) Inconel X-750 (TMFCG)

Frequency (Hz)	Temperature (°C)	Stress Range (MPa)	R-ratio ($\sigma_{min}/\sigma_{max}$)
1.0	25	385	0.05
0.0056	650	385	0.05
0.0056	650	600	-1
0.0056	300-650(In-phase)	285	0.05
0.0056	300-650(In-phase)	600	-1
0.0056	650-300(Out-of-phase)	600	-1

(c) B-1900+Hf (Cyclic Properties)

Frequency (Hz)	Temperature (°C)	Strain Range (%)	N (Cycles)	R-ratio ($\varepsilon_{min}/\varepsilon_{max}$)
0.0056	400-925(In-phase)	0.2000	347	-1
0.0056	400-925 "	0.2515	337	-1
0.0056	400-925 "	0.3030	556	-1
0.0056	400-925 "	0.3580	388	-1
0.0056	400-925 "	0.3850	490	-1
0.0056	400-925 "	0.4075	383	-1
0.0056	400-925 "	0.4332	340	-1
0.0056	400-925 "	0.4525	288	-1
0.0056	400-925 "	0.4825	260	-1
0.0056	400-925 "	0.5650	54	-1
0.0056	925-400(Out-of-phase)	0.1765	475	-1
0.0056	925-400 "	0.1923	426	-1
0.0056	925-400 "	0.2153	312	-1
0.0056	925-400 "	0.2498	360	-1
0.0056	925-400 "	0.276	260	-1
0.0056	925-400 "	0.298	274	-1
0.0056	925-400 "	0.3279	274	-1
0.0056	925-400 "	0.3654	316	-1
0.0056	925-400 "	0.404	374	-1
0.0056	925-400 "	0.4370	326	-1
0.0056	925-400 "	0.4675	342	-1
0.0056	925-400 "	0.537	284	-1
0.0056	925-400 "	0.600	320	-1

(d) B-1900+Hf (TMFCG)

Frequency (Hz)	Temperature (°C)	Strain Range (%)	R-ratio ($\varepsilon_{min}/\varepsilon_{max}$)
0.10	925	0.25	-1
0.0056	925	0.25	-1
0.0056	925	0.50	-1
0.0056	400	0.50	-1
0.0056	400-925(In-phase)	0.25	-1
0.0056	400-925(In-phase)	0.50	-1
0.0056	925-400(Out-of-phase)	0.25	-1
0.0056	925-400(Out-of-phase)	0.50	-1

5.0 Results

5.1 Tensile and Creep Properties

Monotonic and creep tests were conducted to document typical engineering properties and to assess the deformation and failure mechanisms. Tensile and creep tests for B-1900+Hf and Hastelloy-X were run at Pratt & Whitney Aircraft, Hartford, Connecticut. A summary of all tensile tests and observed properties is presented in Table 5.1. Figures 5.1 to 5.4 show the monotonic tensile responses ($\dot{\varepsilon}$ = 0.005 min^{-1}) and the 0.2% yield stress of B-1900+Hf and Hastelloy-X, respectively [41, 123]. Comparison of the 0.2% yield strength data (Figure 5.2 and 5.4) have shown that the results fall within the anticipated scatter. Tensile data of Inconel X-750 (Table 5.1) were shown to compare well with the published data on Inconel X-750 with similar heat treatment [124].

A summary of the test conditions and observed properties for monotonic creep tests conducted on B-1900+Hf is presented in Table 5.2. Figure 5.5 shows the observed rupture lives and the anticipated scatter for this material. Specimens tested at 871°C and 982°C have similar rupture lives for the same normalized stress, while specimens tested at 760°C show a significantly longer life for the same, or higher, normalized stress level. Examination of fracture has shown that at higher temperature (871 and 982°C) the specimens failed in an intergranular cracking mode, while the specimens tested at 760°C failed predominantly by transgranular cracking.

The short-time Hastelloy-X creep response at 871°C and 982°C are shown in Figures 5.6 and 5.7. Table 5.3 summarizes the results for various test temperatures.

Table 5.1

Materials Tested and Their Tensile Properties
at Room and Elevated Temperature

(a) Hastelloy-X

T (°C)	$\dot{\varepsilon}$ (min^{-1})	Ex10^3 (MPa)	0.2% Yield (MPa)	UTS (MPa)	Elong. (%)
21	0.008	207	367.5	787	40
505	0.008	178	294	649	43
649	0.008	161	274	572	36
760	0.008	152	259	435	38
816	0.008	145	231	----	----
871	0.008	137	175	255	50
927	0.008	127	140	----	----
982	0.008	116	91	144	48

(b) Inconel X-750

T (%)	$\dot{\varepsilon}$ (min^{-1})	Ex10^3 (MPa)	0.2% Yield (MPa)	UTS (MPa)	Elong. (%)
24	0.005	295	610	1000	30
300	0.005	303	616	1000	32
650	0.005	260	550	820	7

(c) B-1900+Hf

T (°C)	$\dot{\varepsilon}$ (min^{-1})	Ex10^3 (MPa)	0.2% Yield (MPa)	UTS (MPa)	Elong. (%)	RA (%)
RT	0.005	187.5	714	----	4.9	5.9
260	0.005	169.6	702	888	8.3	10.7
538	0.005	149.6	727	----	----	-----
649	0.005	143.4	701	----	7.7	7.2
760	0.005	146.8	709	950	7.9	8.4
871	0.005	138.9	633	785	5.7	6.1
982	0.005	123.4	345	480	7.1	6.9

Table 5.2

Summary of the Creep Test Results for
B-1900+Hf

Temp. (°C)	Stress (MPa)	%Min Yield	Secondary Life (Hrs.)	CreepElong. (Min^{-1})	Elong. (%)
982	234	75	20.6	$2.5x10^{-5}$	6.0
982	283	90	4.1	$0.7x10^{-4}$	3.0
982	283	90	3.1	$2.0x10^{-4}$	5.0
871	427	75	18.2	$2.5x10^{-5}$	3.2
871	427	75	20.3	$1.5x10^{-3}$	2.3
871	517	90	2.8	$1.0x10^{-4}$	3.2
871	517	90	2.5	$1.25x10^{-4}$	2.3
871	283	50	441	$6.0x10^{-7}$	2.6
760	600	90	134.8	--------	3.1
760	670	100	30.2	$7.9x10^{-6}$	2.9
760	670	100	49.8	$7.5x10^{-6}$	3.78

Table 5.3

Temperature-Dependent Representation of Short-Time
Hastelloy X Creep Response

Temperature, °C	Constants for creep equation* $\varepsilon_{cr} = (\sigma/A)^n\ (t)$	
	A	n
705	973	4.41
760	517	4.75
816	304	5.09
871	195	5.42
927	158	3.78
983	134	2.53

*Stress (σ) in MPa, creep strain (ε_{cr}) in percent, (t) in hours

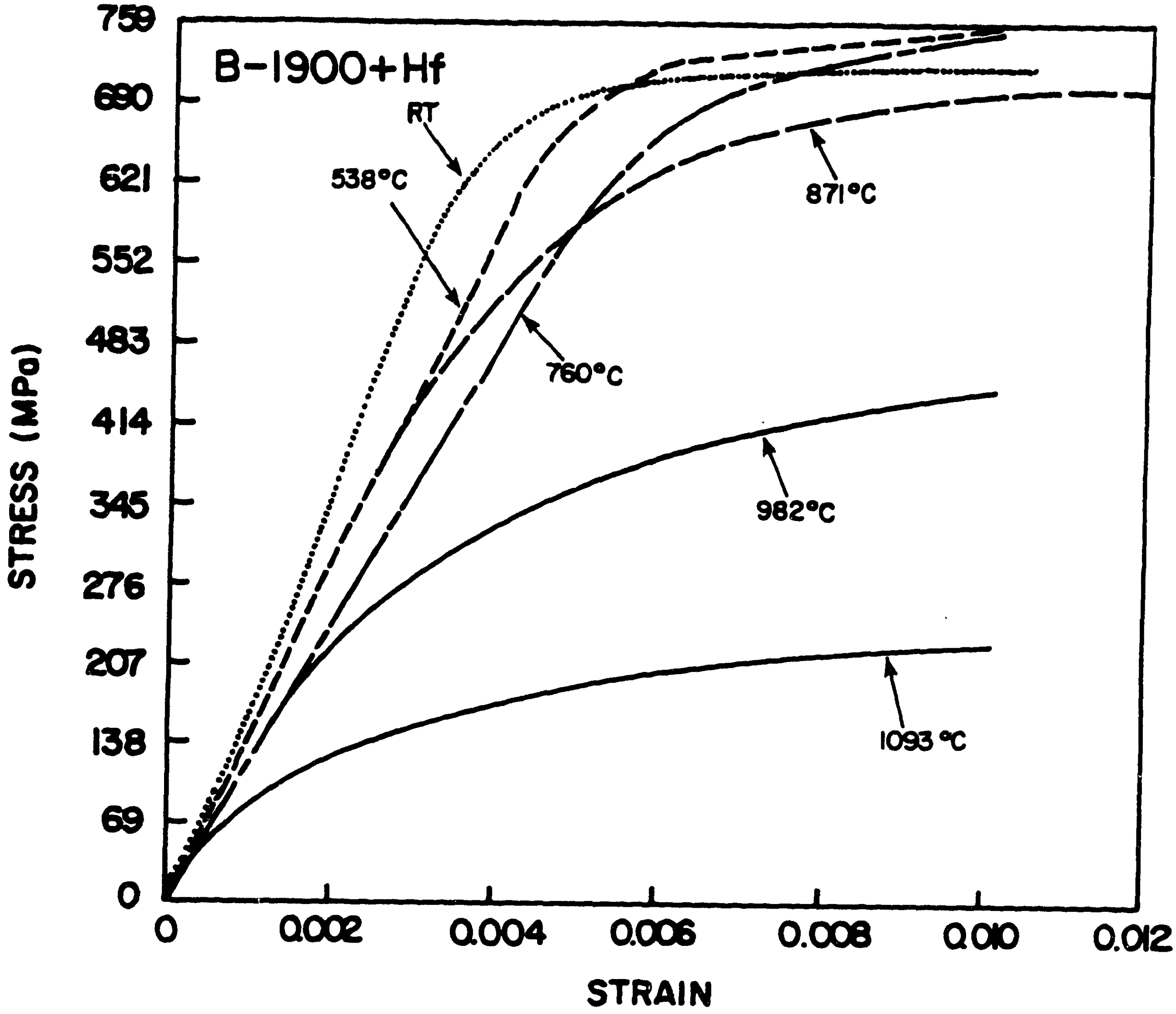


Figure 5.1 Monotonic tensile response of B-1900+Hf($\dot{\varepsilon}$= 0.005 min^{-1})

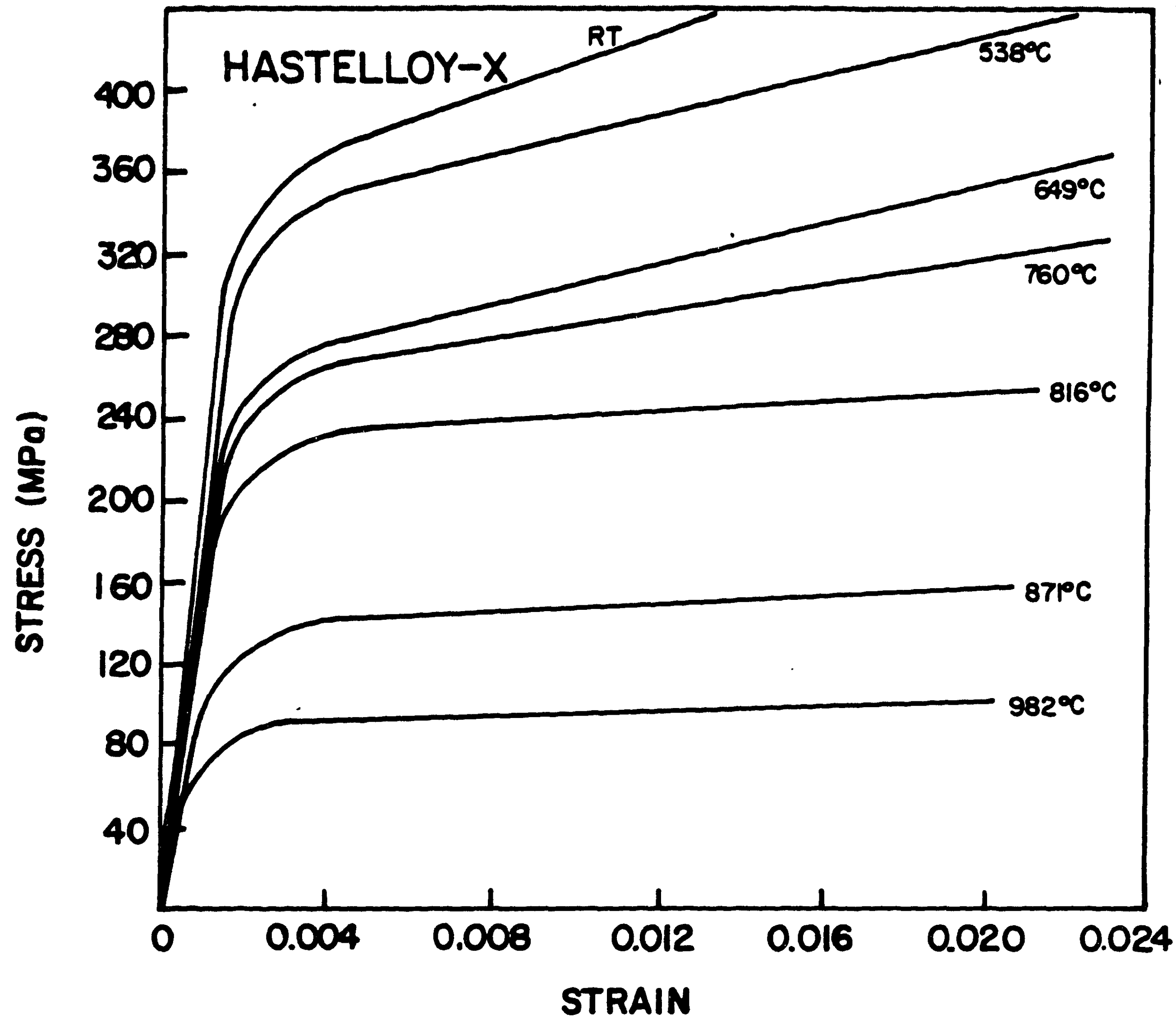


Figure 5.2 Monotonic tensile response of Hastelloy-X($\dot{\varepsilon}$= 0.005 min^{-1})

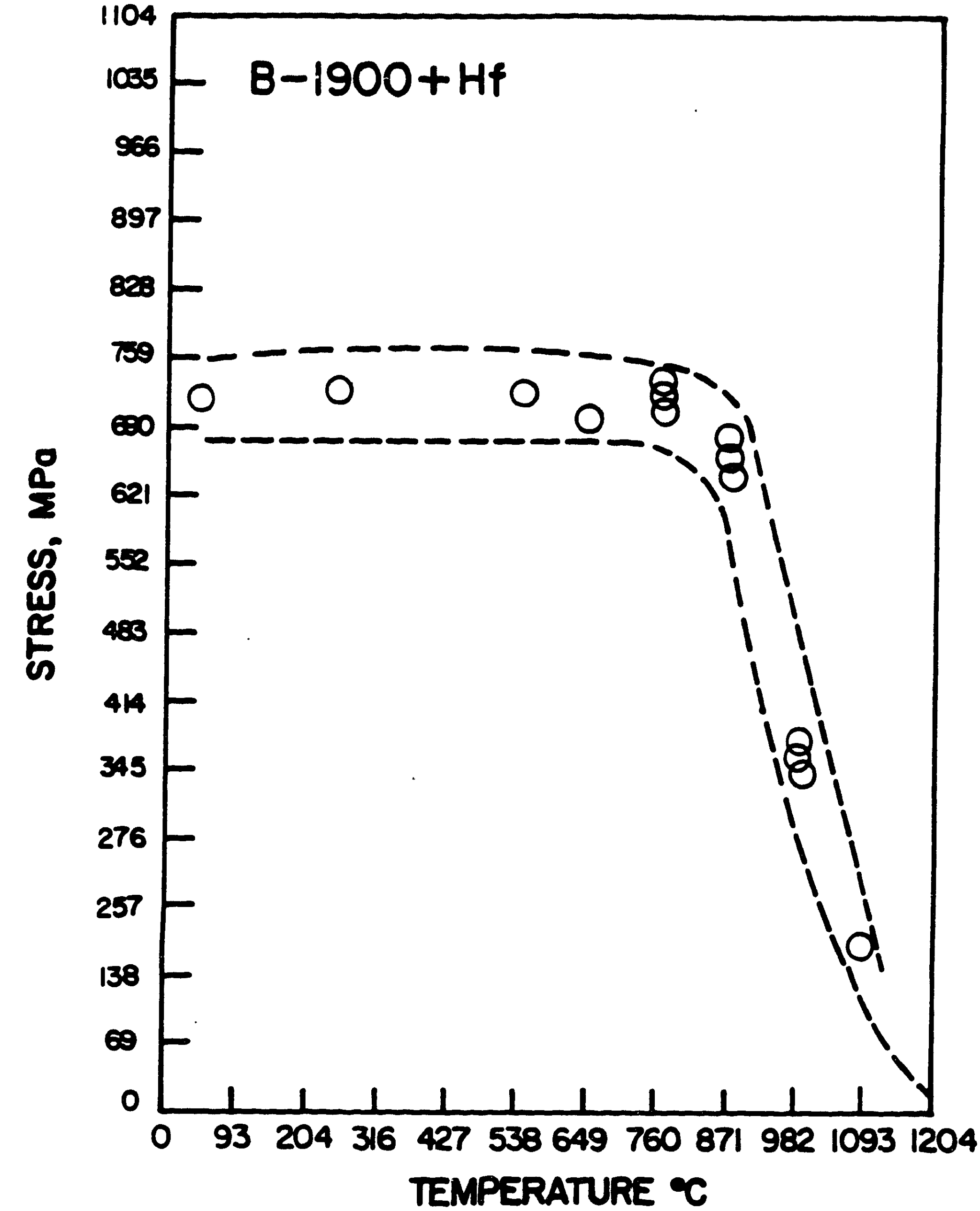


Figure 5.3 Yield stress(0.2 %) versus Representative scatter in B-1900+Hf.

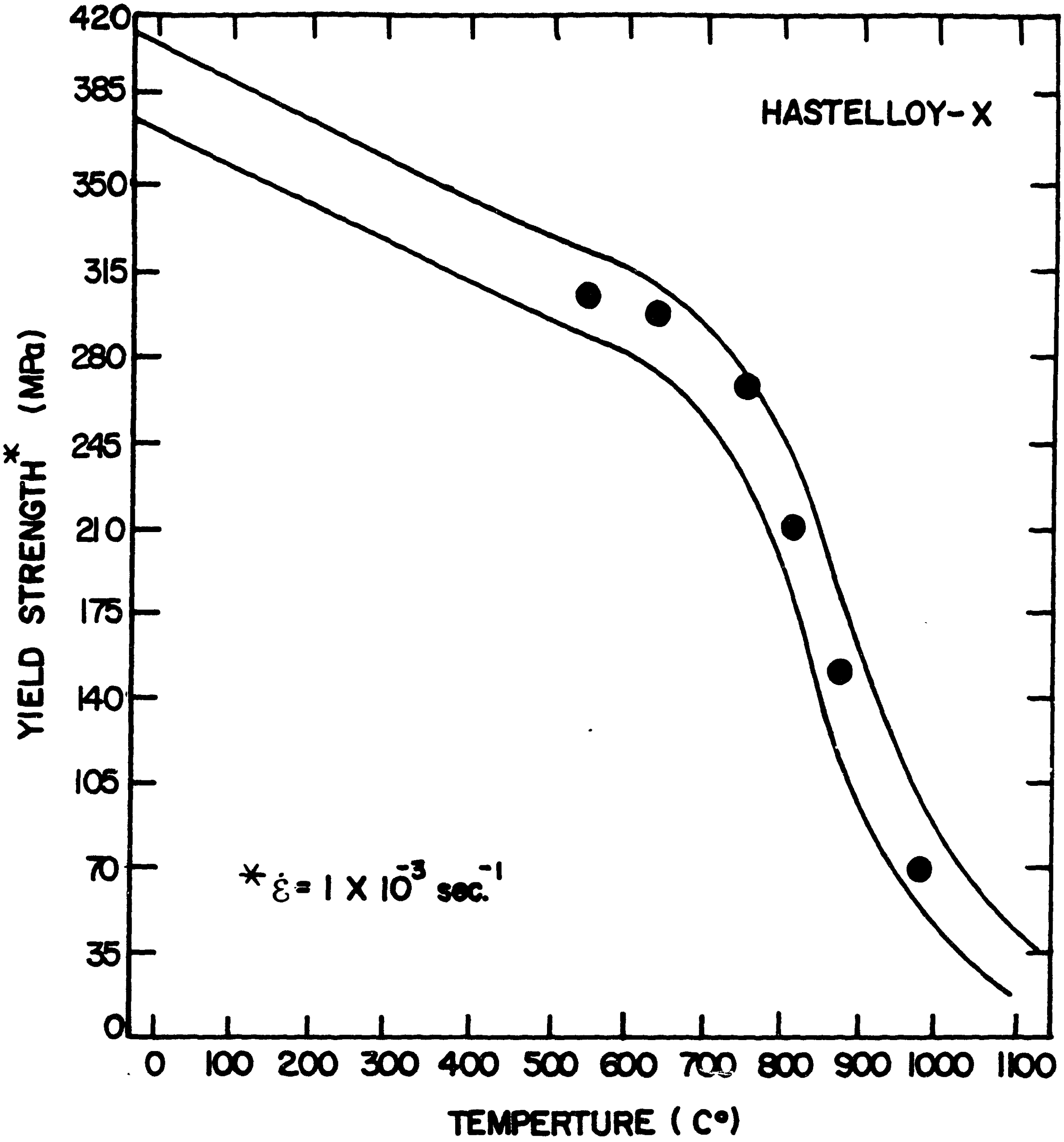


Figure 5.4 Yield stress(0.2 %) versus Representative scatter in Hastelloy-X.

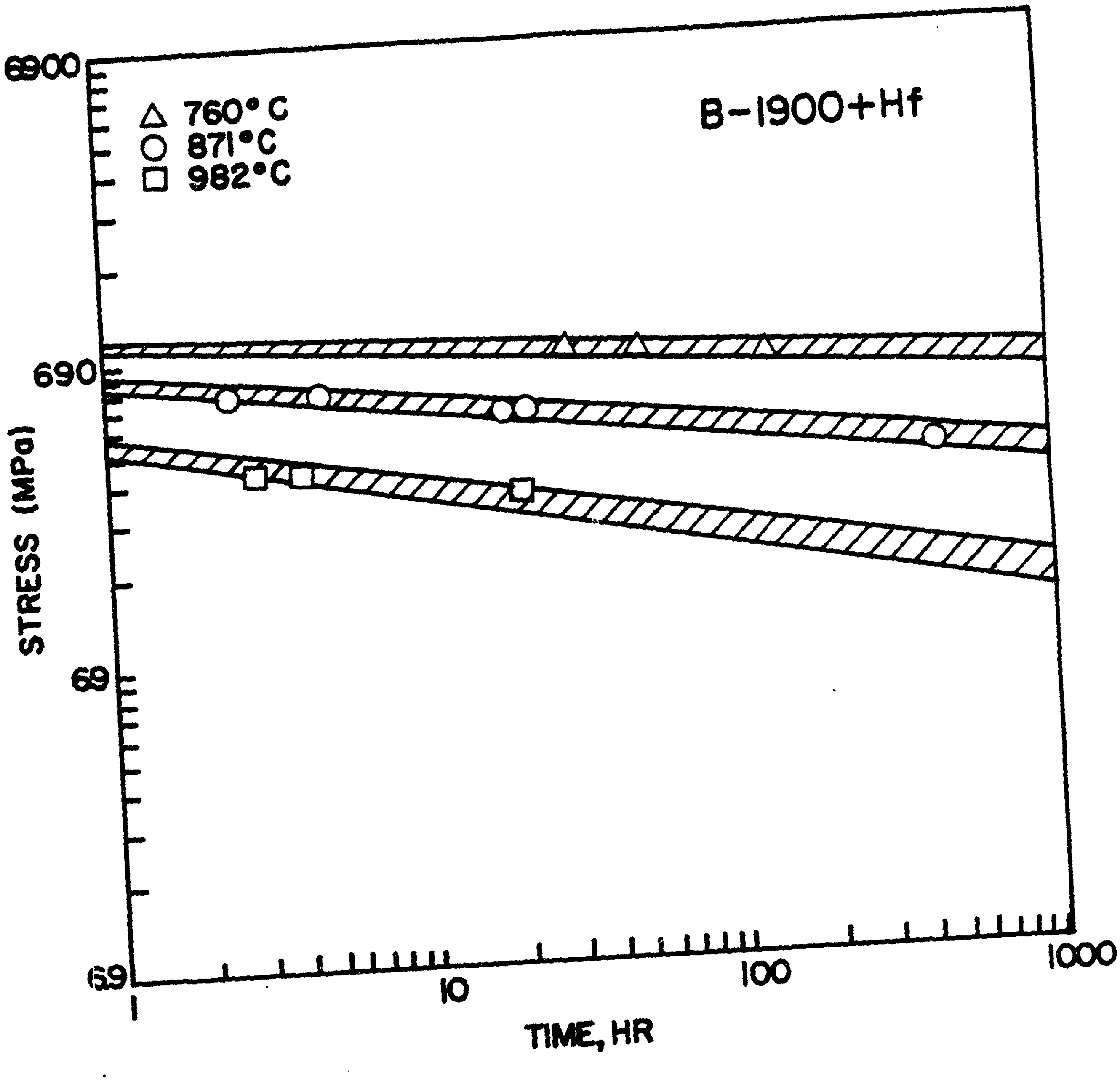


Figure 5.5 Specimen rupture life versus Representative rupture life scatter bands in B-1900+Hf.

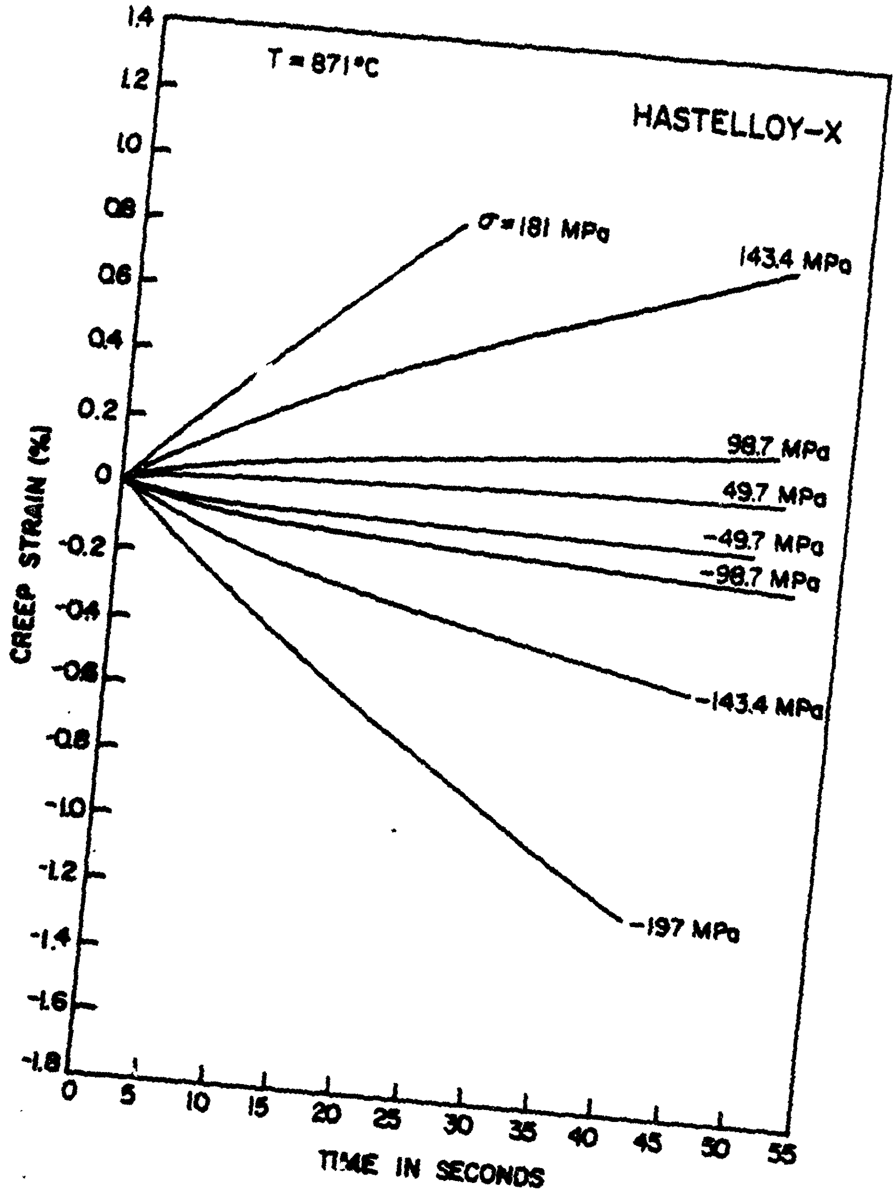


Figure 5.6 Creep responses of Hastelloy-X at 871 °C.

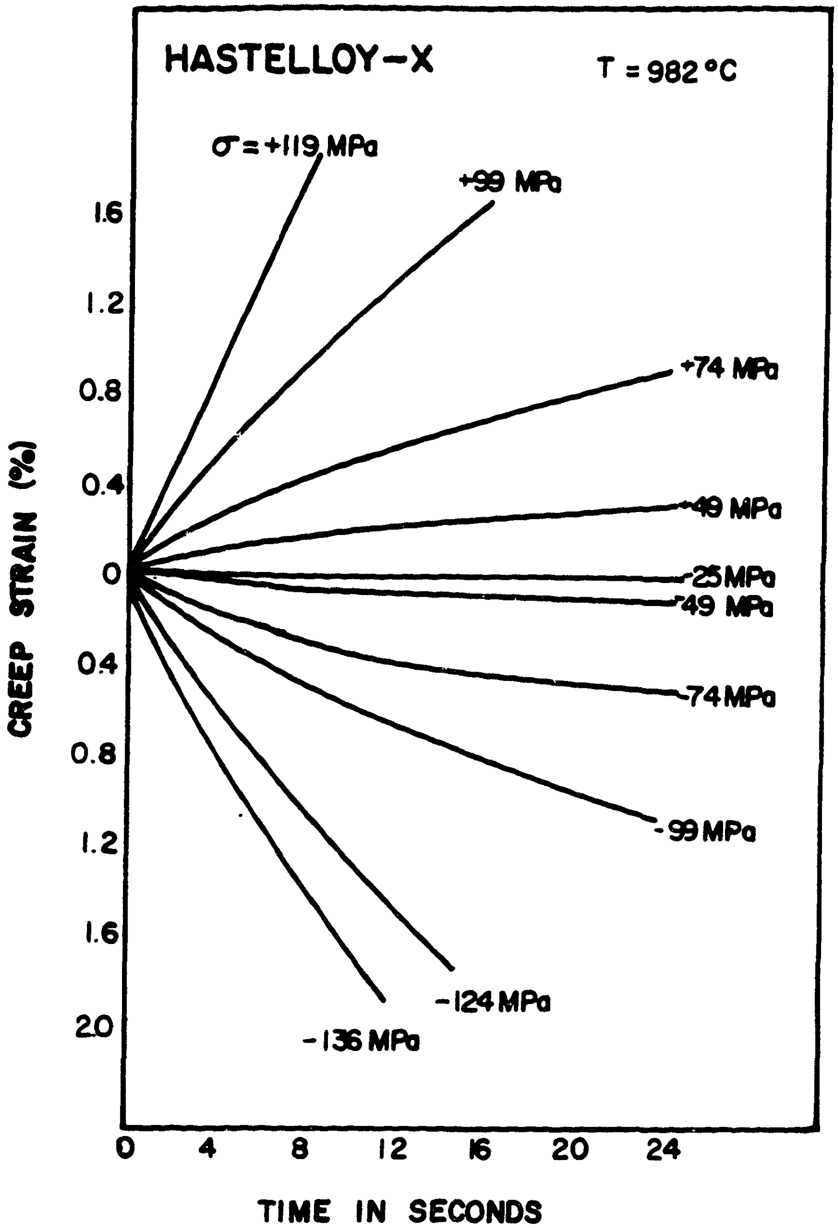


Figure 5.7 Creep responses of Hastelloy-X at 982 °C.

5.2 Low Cycle Fatigue Properties

Isothermal, strain-controlled fatigue tests were conducted to provide a baseline for life prediction model evaluation and to define crack initiation life in B-1900+Hf and Hastelloy-X [123, 125]. Major variables include strain range, strain rate (frequency) and temperature. Fully reversed ($R_\varepsilon = -1$) tests were conducted at 538°C, 650°C, 760°C, 871°C, and 982°C using a symmetrical sawtooth waveform. Tables 5.4 to 5.9 summarize the testing conditions and data for B-1900+Hf and Hastelloy-X, respectively. Failure was defined as 10% tensile load drop from the steady-state values.

A review of the cyclic response histories of the tests have indicated that the high temperature tests (871 and 982°C) display a small amount of softening, while the low temperature tests (760, 650, and 538°C) tests remain constant at the smaller strain ranges, but cyclically harden at the larger strain ranges [123, 126]. In most tests, multiple cracks were observed along the gauge length of the specimens. All specimens of B-1900+Hf were found to have surface initiated fatigue cracks associated with either porosity and/or carbides [123]. Analysis of the data have also shown that the nature of the initiation site (carbide vs. porosity) was not statistically significant in determining the fatigue life. Based on these results, carbide or porosity initiation is not considered a primary variable in the analysis of the fatigue tests. Examination of the failed specimens has shown varying degrees of transgranular vs. intergranular cracking, depending upon the temperature range. In most cases, the growth mode from initiation sites was transgranular. At low temperature ($T < 871$°C) the cracks propagated transgranularly throughout the cross-section of the specimens, whereas at higher temperature (T $\geq$ 871°C), the cracks start transgranularly and surprisingly become intergranular.

Table 5.4

Summary of B-1900+Hf Fatigue Tests
(T = 538°C, R_ε = -1)

Strain Range	Frequency (cpm)	Plastic Strain Range*	Stress Range (MPa)	Cyclic Life First crack	Cyclic Life 10% drop
0.005	10	0.005	855	7980	13000
0.005	10	0.005	807	7536	12560
0.005	10	0.002	868	6390	7775
0.005	10	0.003	910	4665	8877
0.005	10	0.002	965	5326	9050
0.005	10	0.003	945	6840	11400
0.005	10	0.002	945	5264	8774
0.008	6.25	0.002	1331	930	1550
0.008	6.25	0.002	1420	552	920
0.008	6.25	0.0024	1441	558	930
0.010	5.00	0.0065	1585	136	227
0.010	5.00	0.0040	1537	238	397
0.010	4.99	0.0048	1606	183	305

Table 5.5

Summary of B-1900+Hf Fatigue Tests
(T = 650°C, R_ε = -1)

Strain Range	Frequency (cpm)	Plastic Strain Range	Stress Range (MPa)	Cyclic Life First crack	Cyclic Life 10% drop
0.008	10	0.0021	1310	418	916

Table 5.6

Summary of B-1900+Hf Fatigue Tests
(T = 760°C, R_ε = -1)

Strain Range	Frequency (cpm)	Plastic Strain Range	Stress Range (MPa)	Cyclic Life First crack	Cyclic Life 10% drop
0.005	10	0.004	854	3283	7539
0.005	10	0.004	890	2911	6756
0.005	10	0.005	826	3776	8864
0.008	6.25	0.0016	1230	231	723
0.008	6.24	0.0023	1148	278	910
0.008	6.00	0.0014	1210	317	958
0.005	0.9	0.0016	749	4654	1139
0.005	0.9	0.008	765	2248	5620

Table 5.7

Summary of B-1900+Hf Fatigue Tests
(T = 871°C, R_ε = -1)

Strain Range	Frequency (cpm)	Plastic Strain Range	Stress Range (MPa)	Cyclic Life First crack	Cyclic Life 10% drop
0.005	10	0.00023	727	1554	3300
0.005	10	0.00023	750	808	2100
0.005	10	0.00032	773	1226	3050
0.005	10	0.00033	769	1257	3000
0.005	10	0.00022	725	1441	3500
0.005	10	0.00017	737	1032	2700
0.005	10	0.00025	726	1466	3600
0.005	10	0.00018	758	1127	2900
0.005	10	0.00028	770	912	2240
0.005	10	0.00020	753	1520	3780
0.005	10	0.00012	761	1337	3420
0.005	10	0.00021	819	1131	2488
0.005	1.0	0.00040	704	923	2656
0.005	1.0	0.00035	700	827	2110
0.005	1.0	0.00033	675	1019	2968
0.005	0.5	0.00067	645	757	2120
0.005	0.5	0.00062	630	787	2160
0.005	0.5	0.00059	625	499	1310
0.0035	14	0.00007	524	21760	67,580
0.0040	12.5	0.00009	647	3553	-----
0.0040	12.5	0.00010	629	3188	9252
0.0040	12.5	0.00006	677	3651	8100
0.0080	6.25	0.00130	998	81	383
0.0080	6.25	0.00103	1028.5	88	468
0.0080	6.25	0.00100	1012.2	93	454
0.0080	6.25	0.00129	1009.4	69	330

Table 5.7 (continued)

Strain Range	Frequency (cpm)	Plastic Strain Range	Stress Range (MPa)	Cyclic Life First crack	10% drop
0.0080	0.625	0.00140	1019	64	325
0.0080	0.625	0.00166	923	49	272
0.0080	0.625	0.00150	984	39	208
0.0080	0.624	0.00147	962	35	189

Table 5.8

Summary of B-1900+Hf Fatigue Tests
(T = 982°C, R_ε = -1)

Strain Range	Frequency (cpm)	Plastic Strain Range	Stress Range (MPa)	Cyclic Life First crack	Cyclic Life 10%drop
0.005	10.	0.00117	537	406	1254
0.005	10.	0.00094	536	460	1478
0.005	10.	0.00088	525	308	-----
0.005	10.	0.00092	541	389	1207
0.005	.10	0.00143	449	337	978
0.005	.10	0.00140	502	325	911
0.005	.10	0.00142	457	382	1332
0.008	6.25	0.00275	662	67	293
0.008	6.25	0.00263	636	48	233
0.008	6.25	0.00283	649	51	241
0.008	0.625	0.00311	583	44	207

Table 5.9

Summary of Hastelloy-X Fatigue Tests
($R_\varepsilon = -1$)

T (°C)	Strain Range	Frequency (cpm)	Plastic Strain Range	Stress Range (MPa)	Cyclic Life
538	0.005	10	0.0008	694	21,360
538	0.020	10	0.0120	1083	18,330
760	0.005	10	0.0010	622	6,100
760	0.005	10	0.0012	592	8,400
871	0.004	12.5	0.00085	403	12,600
871	0.004	12.5	0.00090	39	9,500
871	0.004	12.5	0.00080	410	12,800
871	0.005	10	0.0015	448	5,800
871	0.005	10	0.0019	397	3,800
871	0.005	10	0.0018	410	6,750
871	0.005	10	0.0017	423	4,400
871	0.005	10	0.0018	410	2,900
871	0.008	6.25	0.0050	384	1,250
871	0.008	6.25	0.0050	384	1,250
871	0.008	6.25	0.0047	423	1,120
982	0.005	10	0.00299	183	3,800
982	0.005	10	0.00280	200	3,805

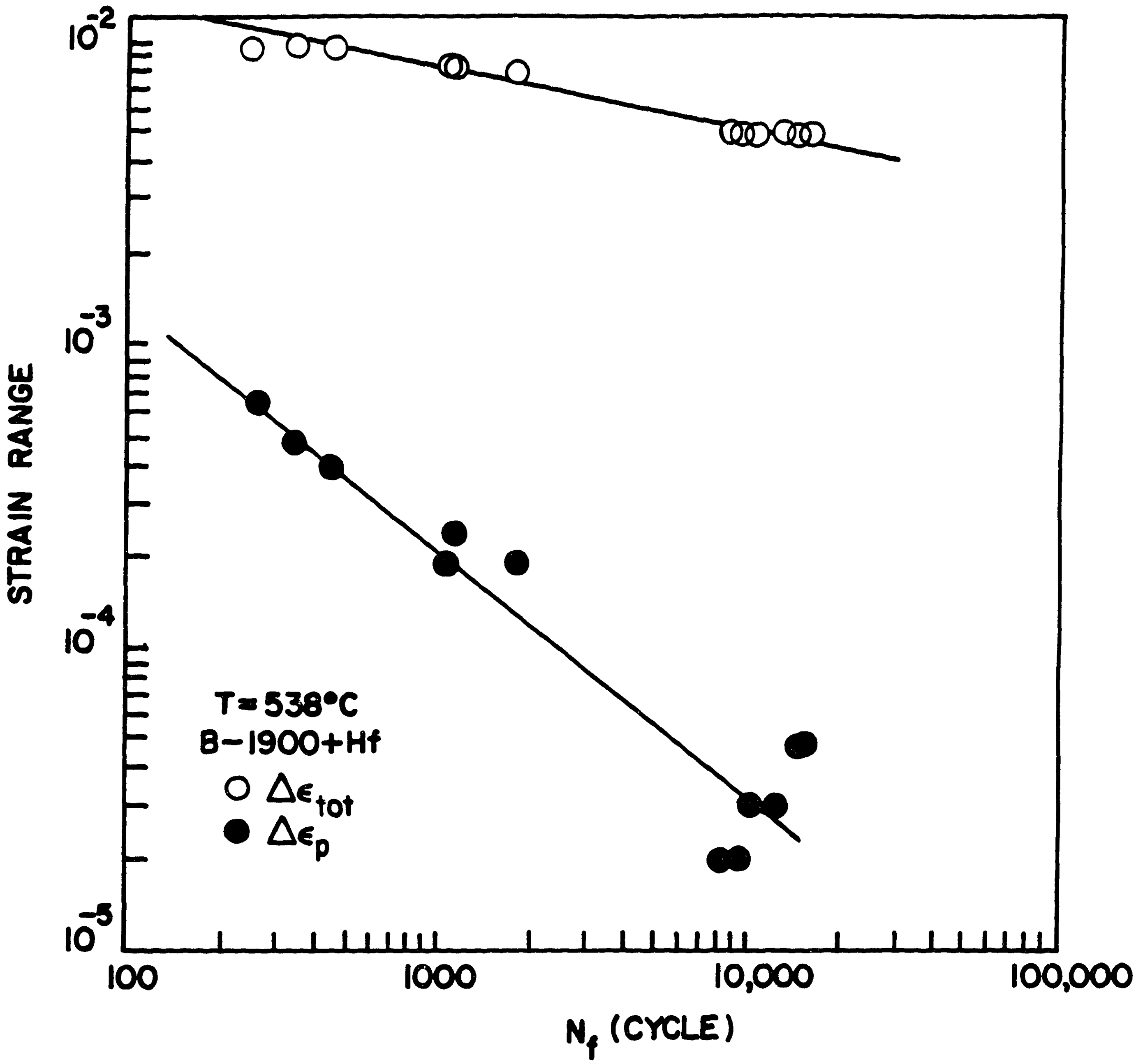


Figure 5.8 Strain range versus Number of cycles to failure in B-1900+Hf(T= 538 °C).

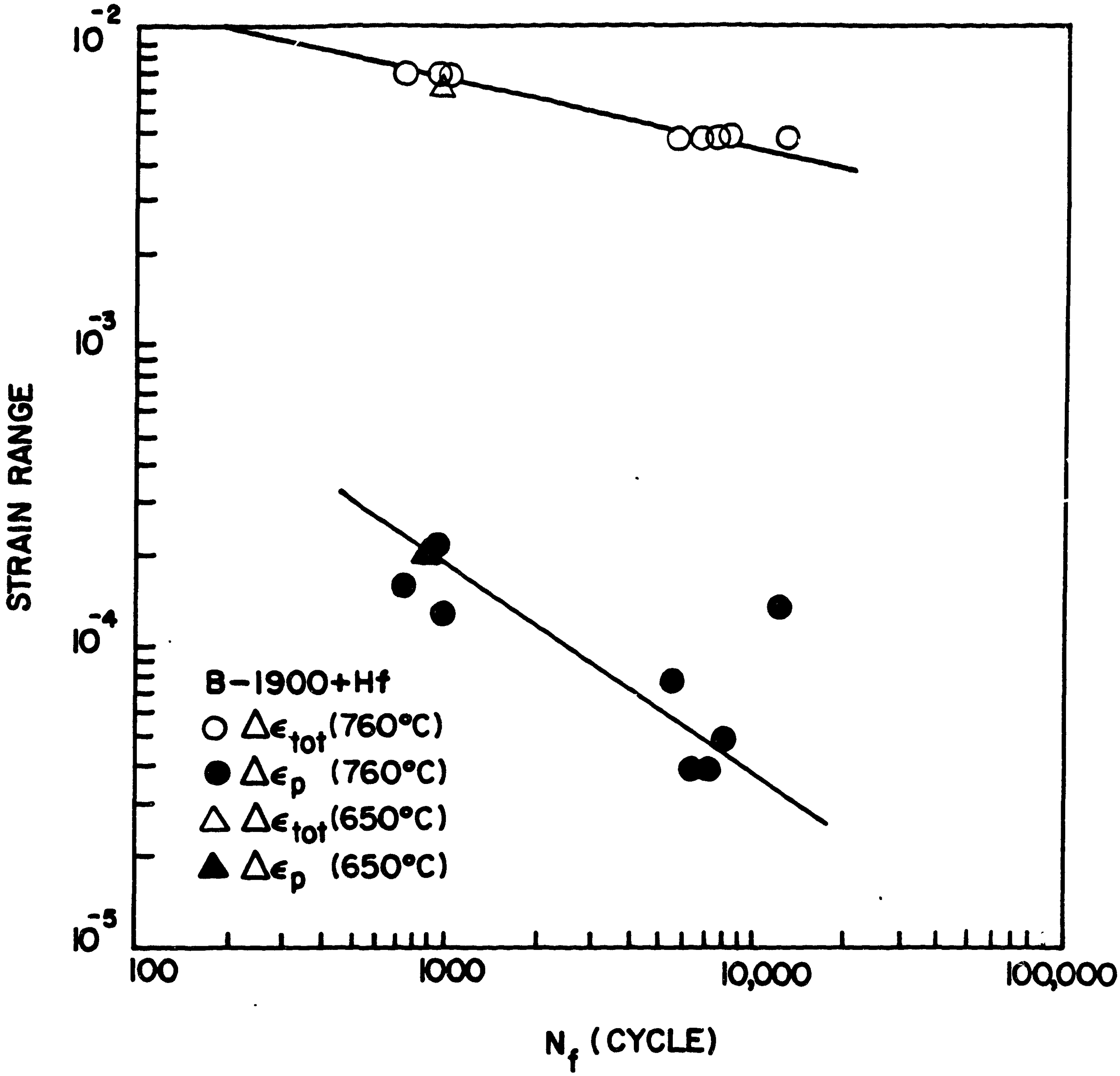


Figure 5.9 Strain range versus Number of cycles to failure in B-1900+Hf(T= 650 and 760 °C).

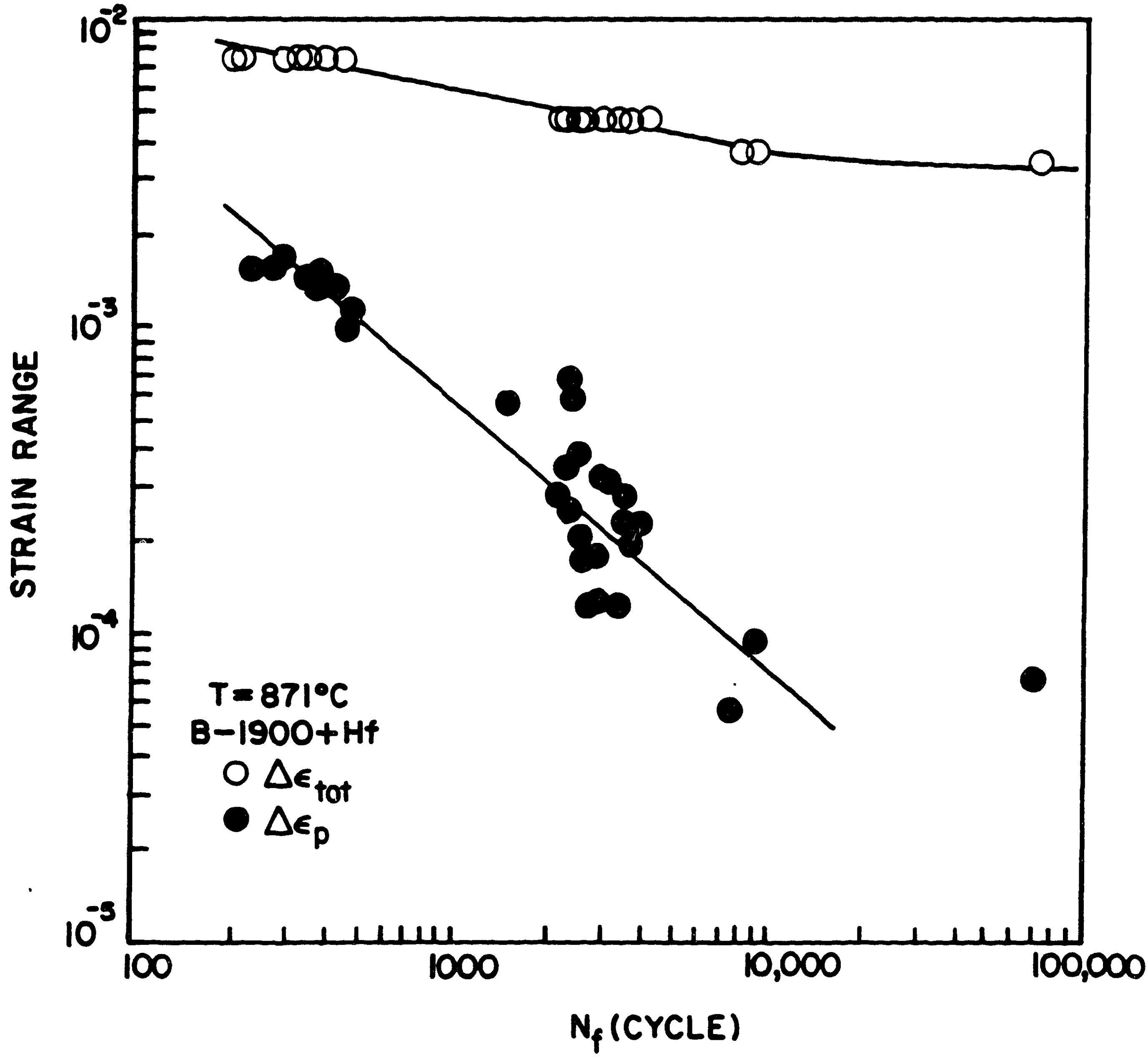


Figure 5.10 Strain range versus Number of cycles to failure in B-1900+Hf(T= 871 °C).

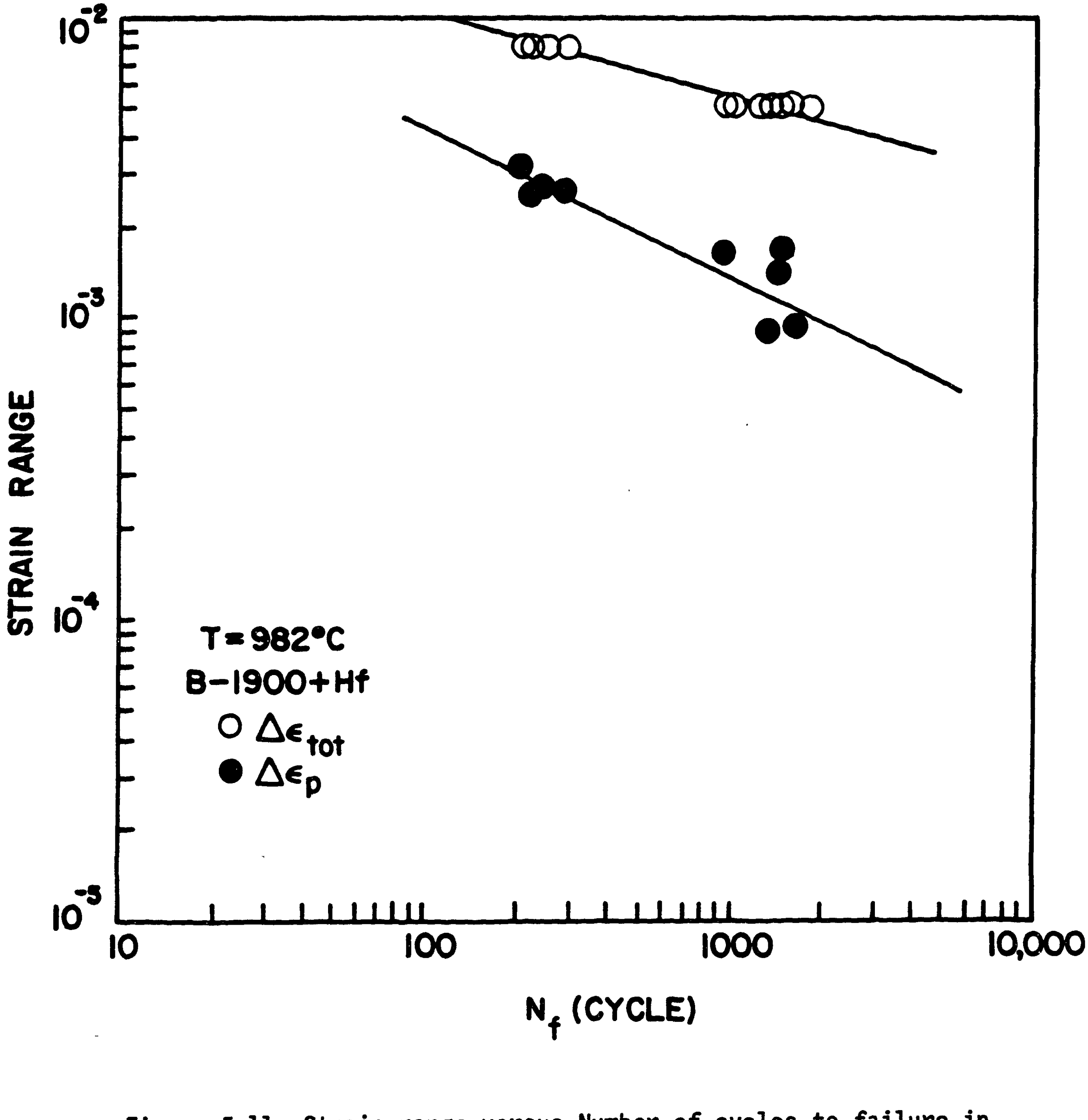


Figure 5.11 Strain range versus Number of cycles to failure in B-1900+Hf(T= 982 °C).

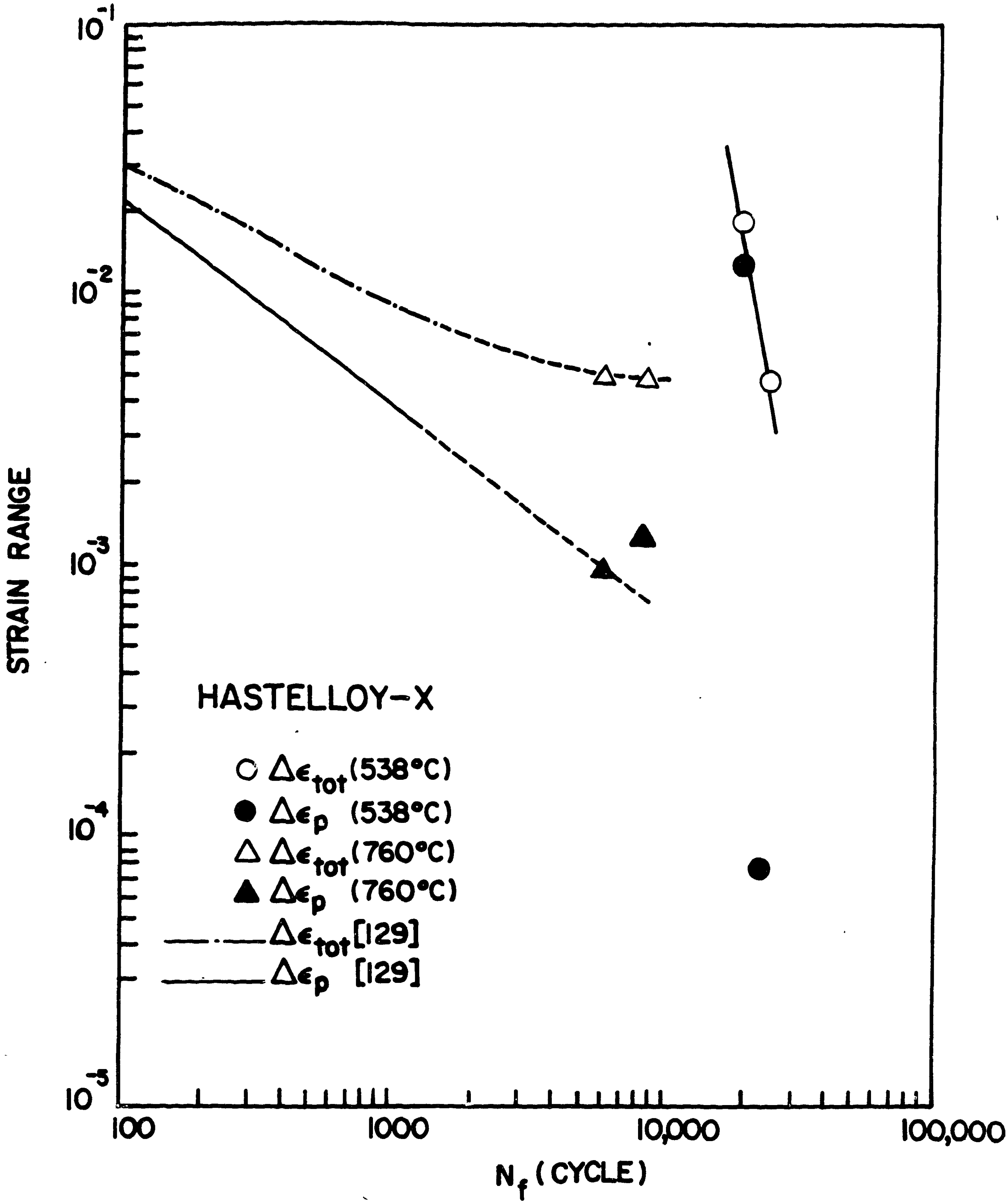


Figure 5.12 Strain range versus Number of cycles to failure in Hastelloy-X(T= 538 and 760 °C).

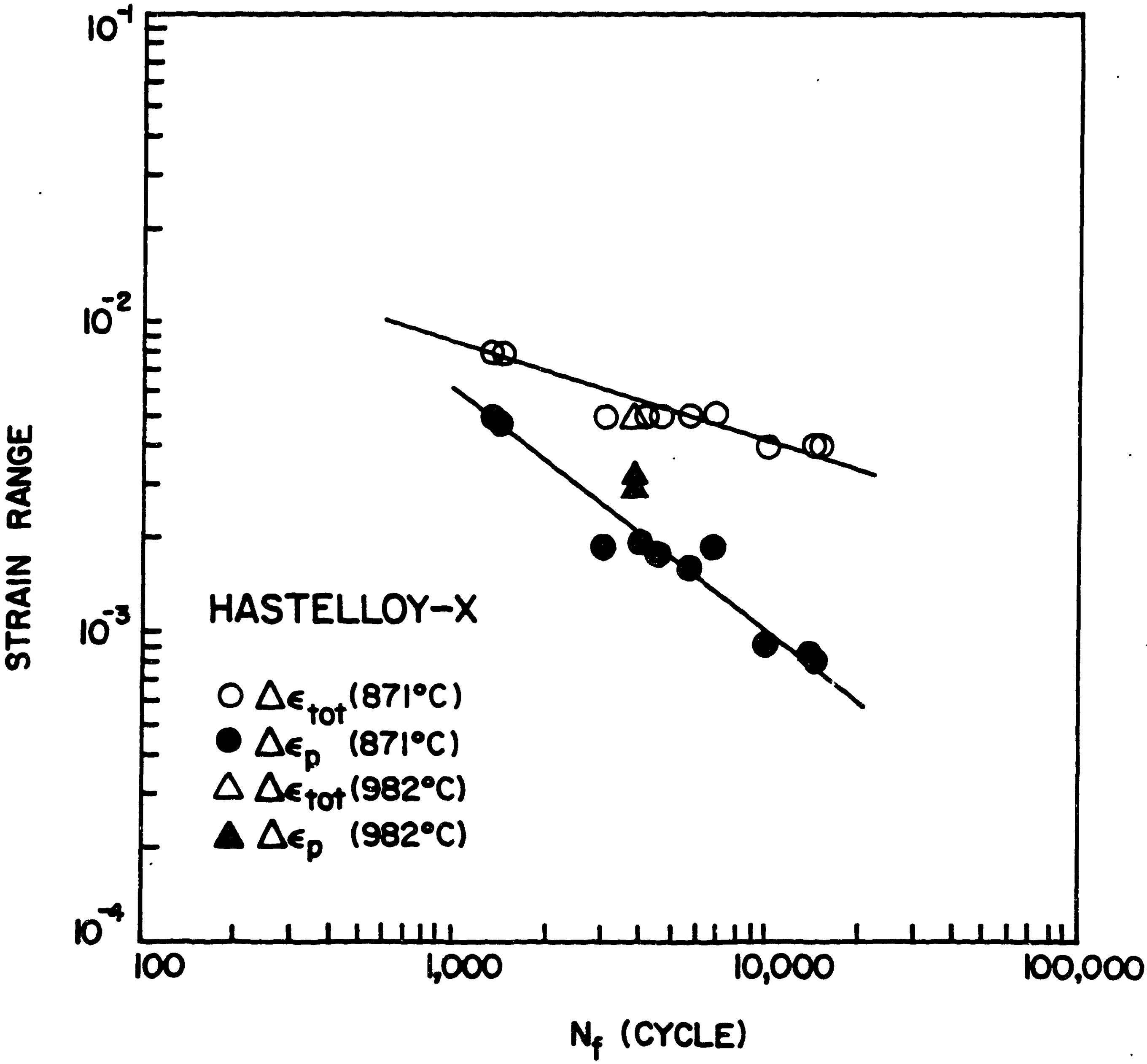


Figure 5.13 Strain range versus Number of cycles to failure in Hastelloy-X(T= 871 and 982 °C).

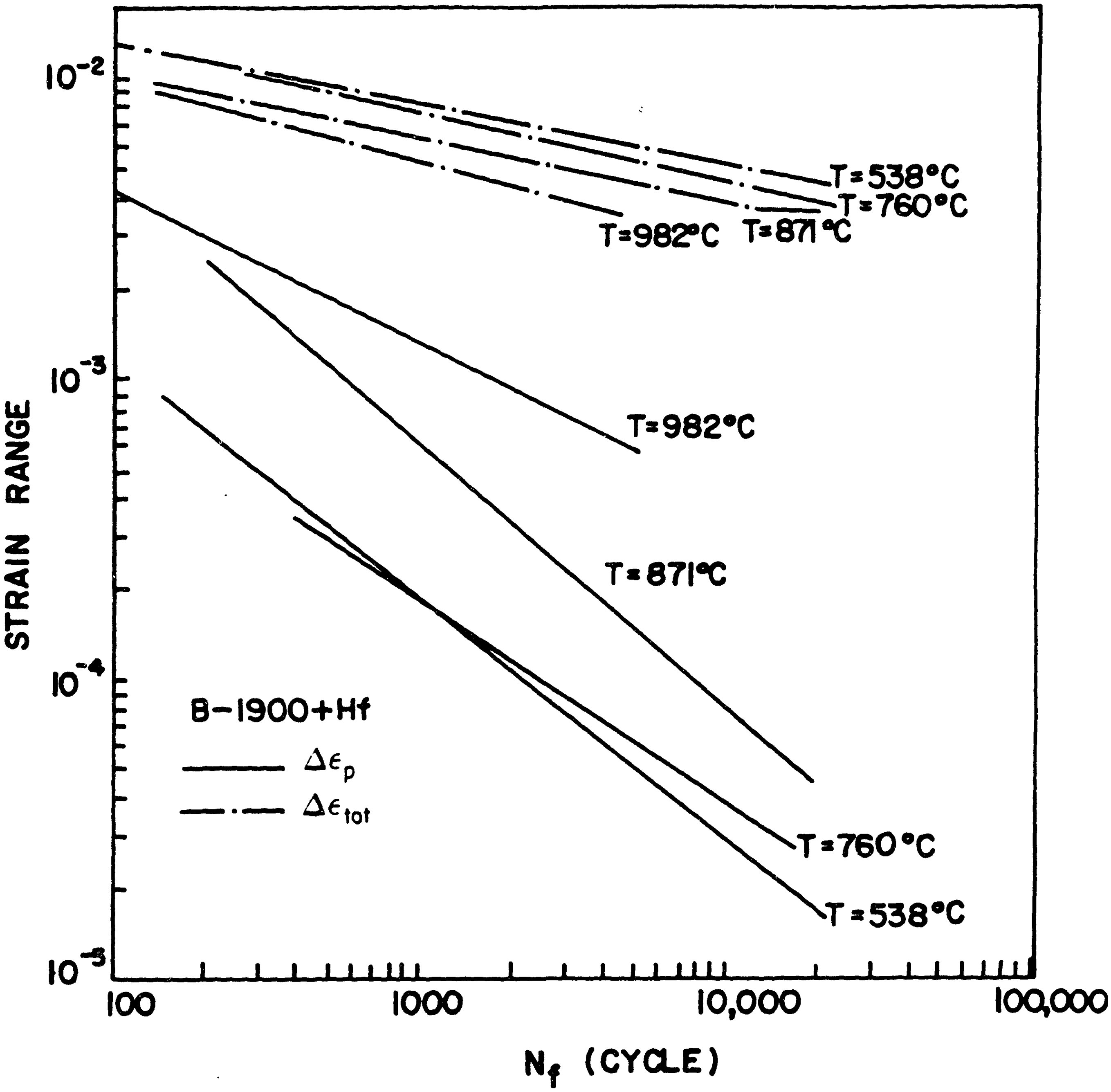


Figure 5.14 Strain range versus Number of cycles to failure as a function of temperature in B-1900+Hf.

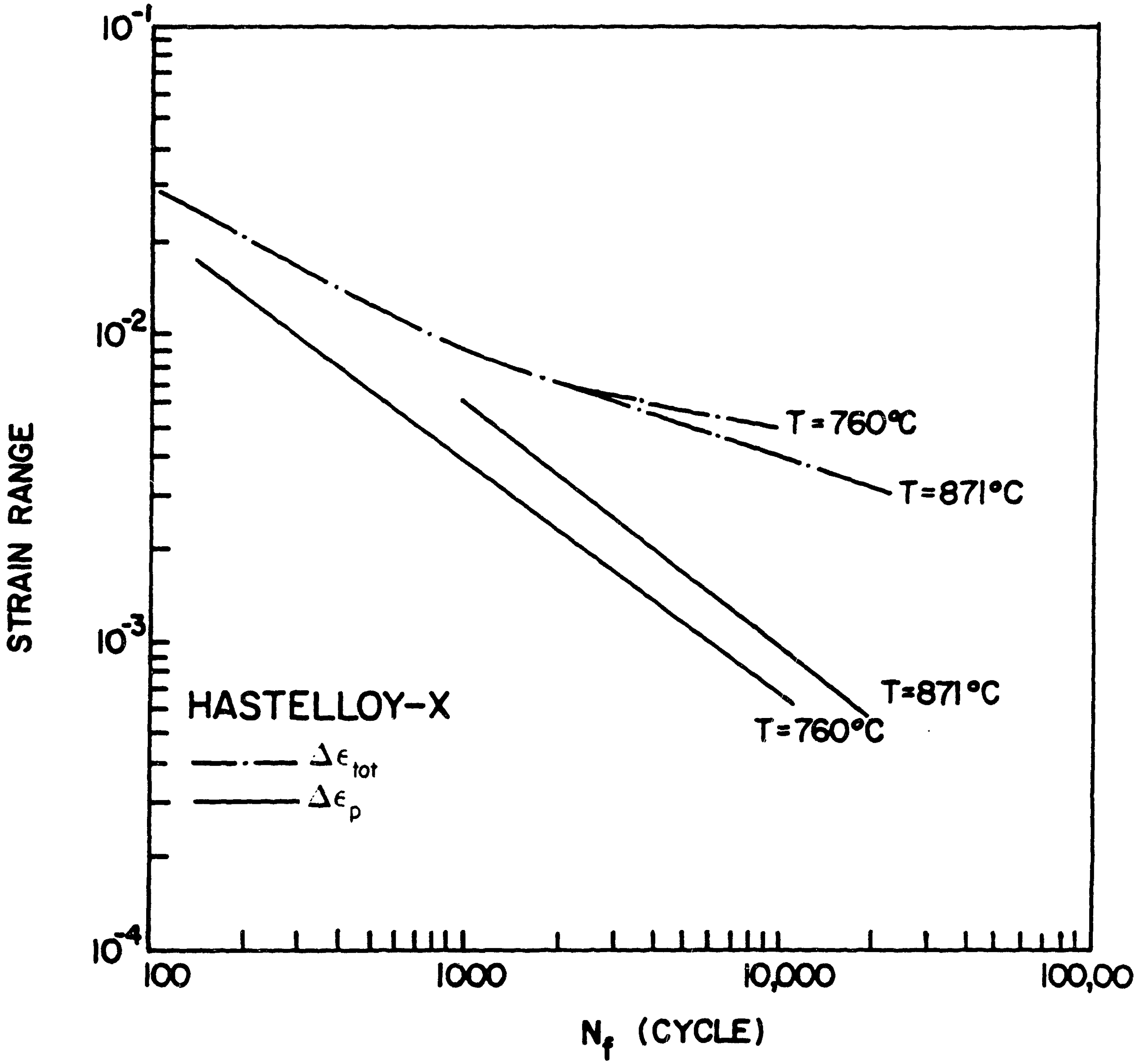


Figure 5.15 Strain range versus Number of cycles to failure as a function of temperature in Hastelloy-X.

The fatigue life in terms of total strain range ($\Delta \varepsilon_p$) at various temperatures is shown in Figures 5.8 to 5.13 for both B-1900+Hf and Hastelloy-X. These figures show that the total strain range ($\Delta \varepsilon_{tot}$) versus N_f correlates the data better than the inelastic strain range. This correlation is better supported by Figures 5.14 and 5.15, which show the effect of temperature on the fatigue life of B-1900+Hf and Hastelloy-X. At low temperature (538° and 760°C) there is no or little effect of the temperature, both in terms of $\Delta \varepsilon_{tot}$ vs. N_f or $\Delta \varepsilon_p$ vs. N_f. At higher temperature (T> 760°C) the life (N_f) decreases with increasing temperature at constant $\Delta \varepsilon_p$. Similar conclusions have also been reached by others [48-49, 66, 123, 127-128] with similar nickel-based alloys.

5.3 Thermal-Mechanical Fatigue Properties

Results of all the experiments conducted to measure the cyclic response of B-1900+Hf under TMF conditions are summarized in Tables 5.9 and 5.10, where the mechanical strain range ($\Delta \varepsilon_{tot}$), the number of applied cycles at each strain range, the initial and final stress range, and final plastic strain range ($\Delta \varepsilon_p$) are listed for both in-phase and out-of-phase cycling. The plastic strain ranges were taken as the width of the hysteresis loop at zero stress and, therefore, include both time-independent and time-dependent inelastic strain components. Figures 5.16 and 5.17 show the thermal strain (ε_{th}), the mechanical strain (ε_{mec}), the total strain (ε_{tot}), and the stress amplitude as a function of time (one cycle) for both in-phase and out-of-phase cycling. From the

Table 5.10

B-1900+Hf
In-Phase
T_{max} = 925°C
T_{min} = 400°C
0.0056 Hz (1/3 cpm)

$\Delta\varepsilon$ (%)	N	$\Delta\sigma_i$ (MPa)	$\Delta\sigma_f$ (MPa)	$\Delta\varepsilon_{pf}$ (%)
.2000	347	328	356	.000
.2515	337	426	436	.000
.3030	556	527	524	.000
.3590	388	611	620	.000
.3830	490	657	664	.000
.4075	303	786	708	.012
.43215	340	745	748	.015
.45250	280	777	782	.0275
.4925	160	881	842	.0475
.5650	54	862	098	.0700

Table 5.11

B-1900+Hf
Out-of-Phase
T_{max} = 925°C
T_{min} = 400°C
0.0056 Hz (1/3 cpm)

$\Delta\varepsilon$ (%)	N	$\Delta\sigma_i$ (MPa)	$\Delta\sigma_f$ (MPa)	$\Delta\varepsilon_{pf}$ (%)
.1765	475	304	310	.000
.1929	426	450	369	.000
.2153	312	416	408	.000
.2498	360	460	455	.000
.2760	260	492	485	.000
.2880	274	585	558	.000
.3279	274	605	585	.012
.3654	316	.650	640	.0075
.4040	374	711	684	.0145
.4370	326	754	745	.0145
.4675	222	794	787	.0325
.5370	142	885	876	.0405
.6000	60	981	950	0.0880

load-time and mechanical strain-time (Figures 5.16 - 5.17), the hysteresis loops were obtained.

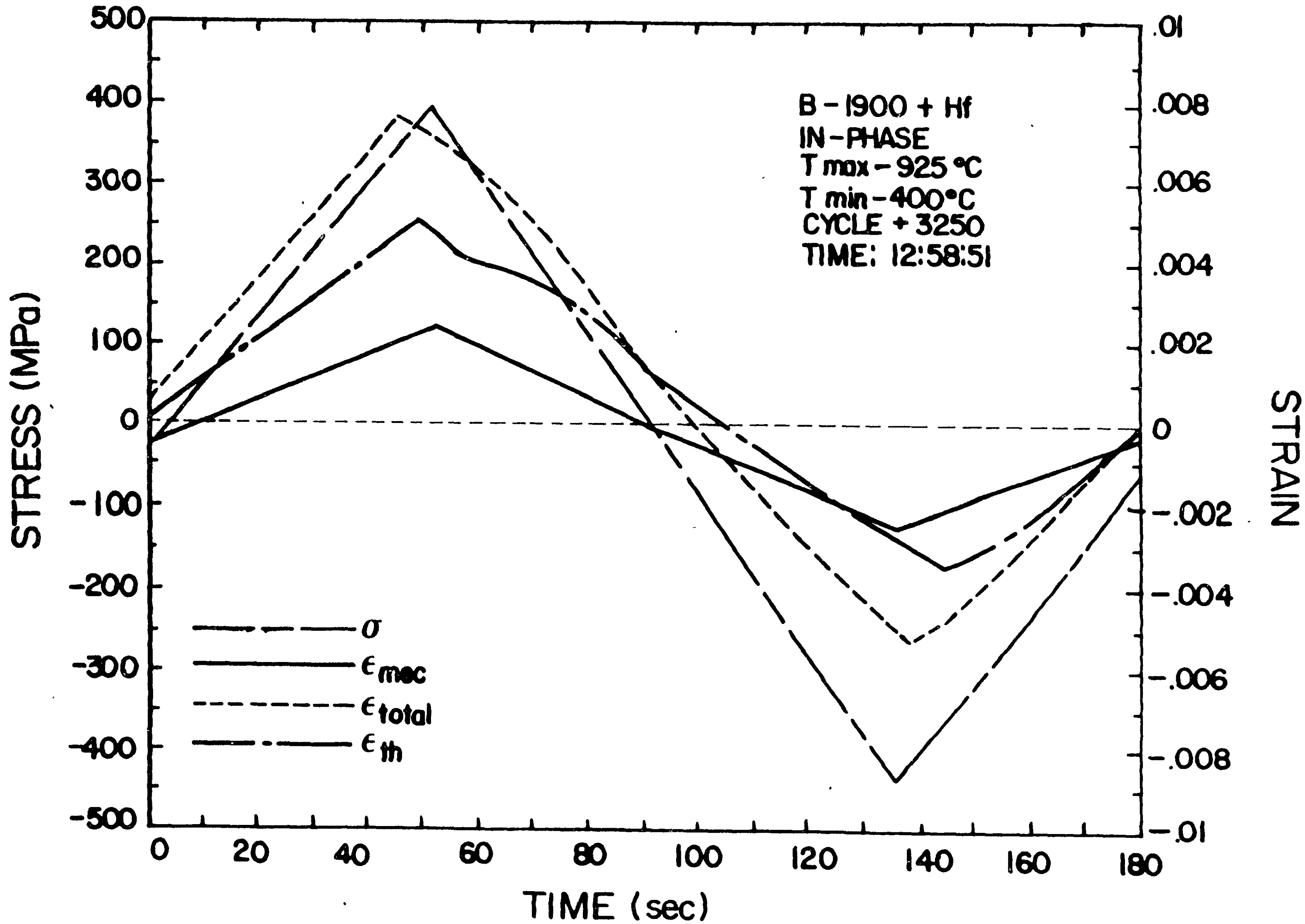


Figure 5.16 Various strain components and stress amplitude in one cycle (B-1900+Hf, in-phase).

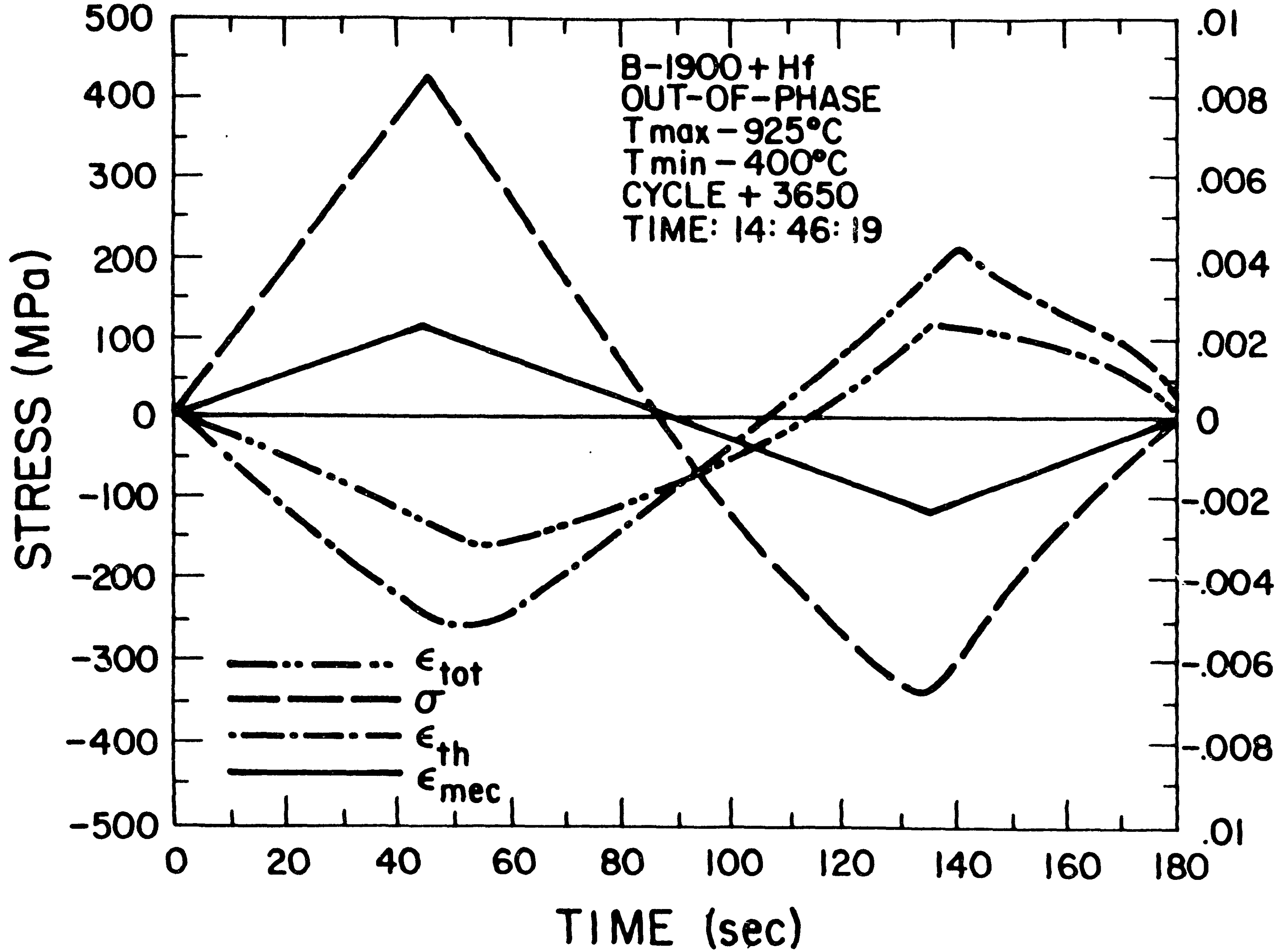


Figure 5.17 Various strain components and stress amplitude in one cycle (B-1900+Hf, out-of-phase).

Even though the mechanical strain cycling was fully reversed, the stress cycle was not symmetric about zero because the temperature was different at each extreme of the cycle. Figures 5.18 and 5.19 show examples of the loops obtained for both in-phase and out-of-phase cycling. For out-of-phase cycling, a positive (tensile) mean stress was observed and a negative mean stress was observed for the in-phase cycling. That is, for the in-phase cycle the magnitude of peak compressive stress was greater than the magnitude of peak tensile stress, or $|\sigma_{min}| > |\sigma_{max}|$. The opposite was true for the out-of-phase cycle where $|\sigma_{max}| > |\sigma_{min}|$. Figures 5.20 and 5.21 show the type of stress response for in-phase and out-of-phase cycling at fixed strain range (see Appendix IV). Figures 5.22 and 5.23 summarize the fatigue results at all strain ranges.

From Figures 5.20 to 5.23 the following conclusions can be drawn. First, one can conclude that the mean stress ($\bar{\sigma}$) does not vary much with the number of applied cycle and strain range for in-phase cycling. However, σ does vary with N and $\Delta \varepsilon_t$ for out-of-phase cycling. It is also interesting to notice that for similar strain range, the absolute value of σ is higher for out-of-phase than for in-phase cycling. Another conclusion that can be drawn is that for all $\Delta \varepsilon_{tot}$, in-phase cycling show σ_{max} to harden, whereas σ_{min} stayed almost unchanged except at high applied strain range. For out-of-phase cycling, the hardening-softening behavior depends on the applied strain range. At $\Delta \varepsilon_{tot} = 0.1765\%$, σ_{max} softened whereas σ_{min} hardened. At $\Delta \varepsilon_{tot} = 0.1923\%$, both σ_{max} and σ_{min} softened with the applied number of cycle. At higher $\Delta \varepsilon_{tot}$, σ_{max} hardened and σ_{min} softened. The softening of σ_{min} being more important than the hardening of σ_{max}, results in a drift of $\bar{\sigma}$ to higher

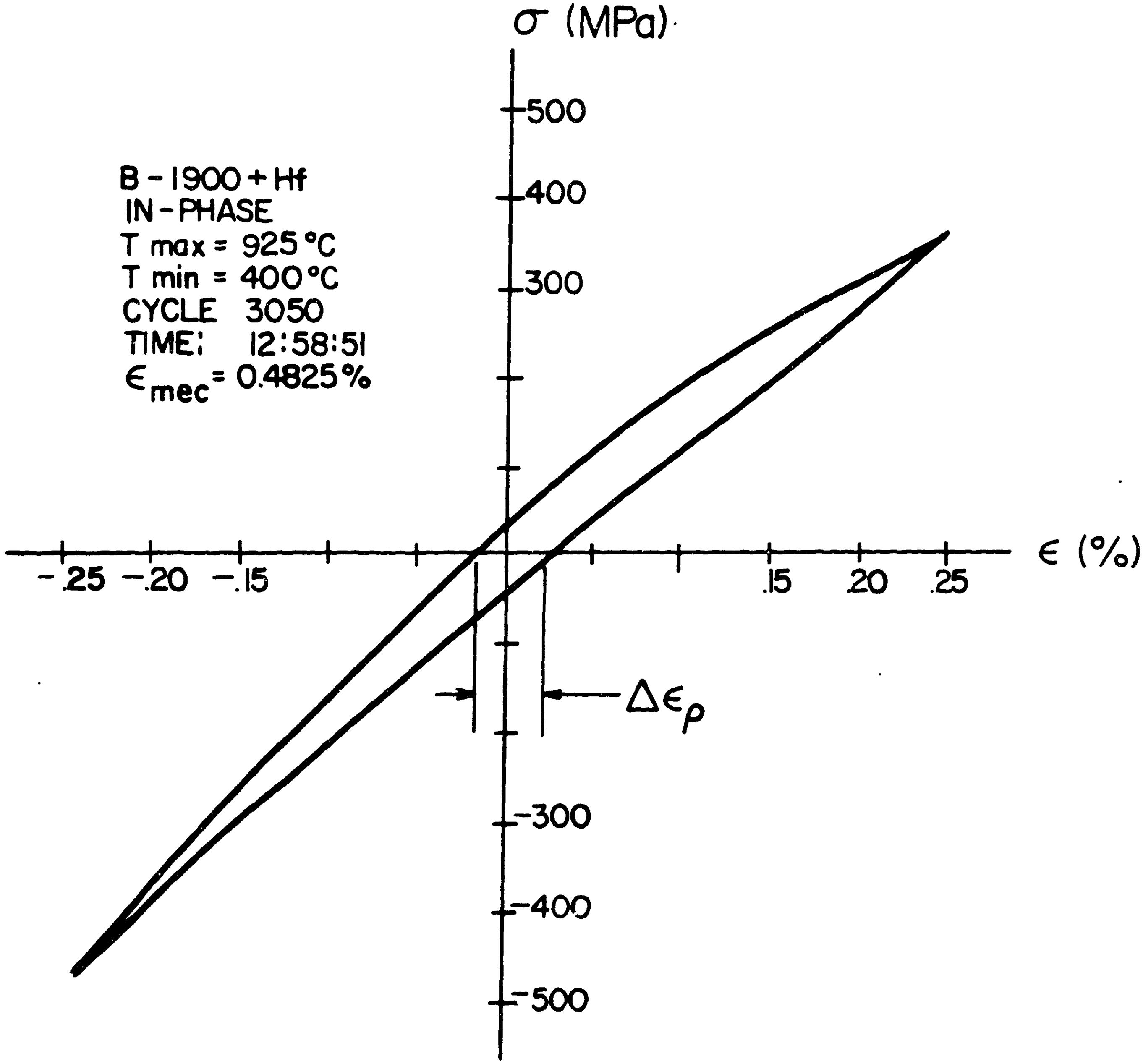


Figure 5.18 Hysteresis loop obtained in B-1900+Hf (in-phase).

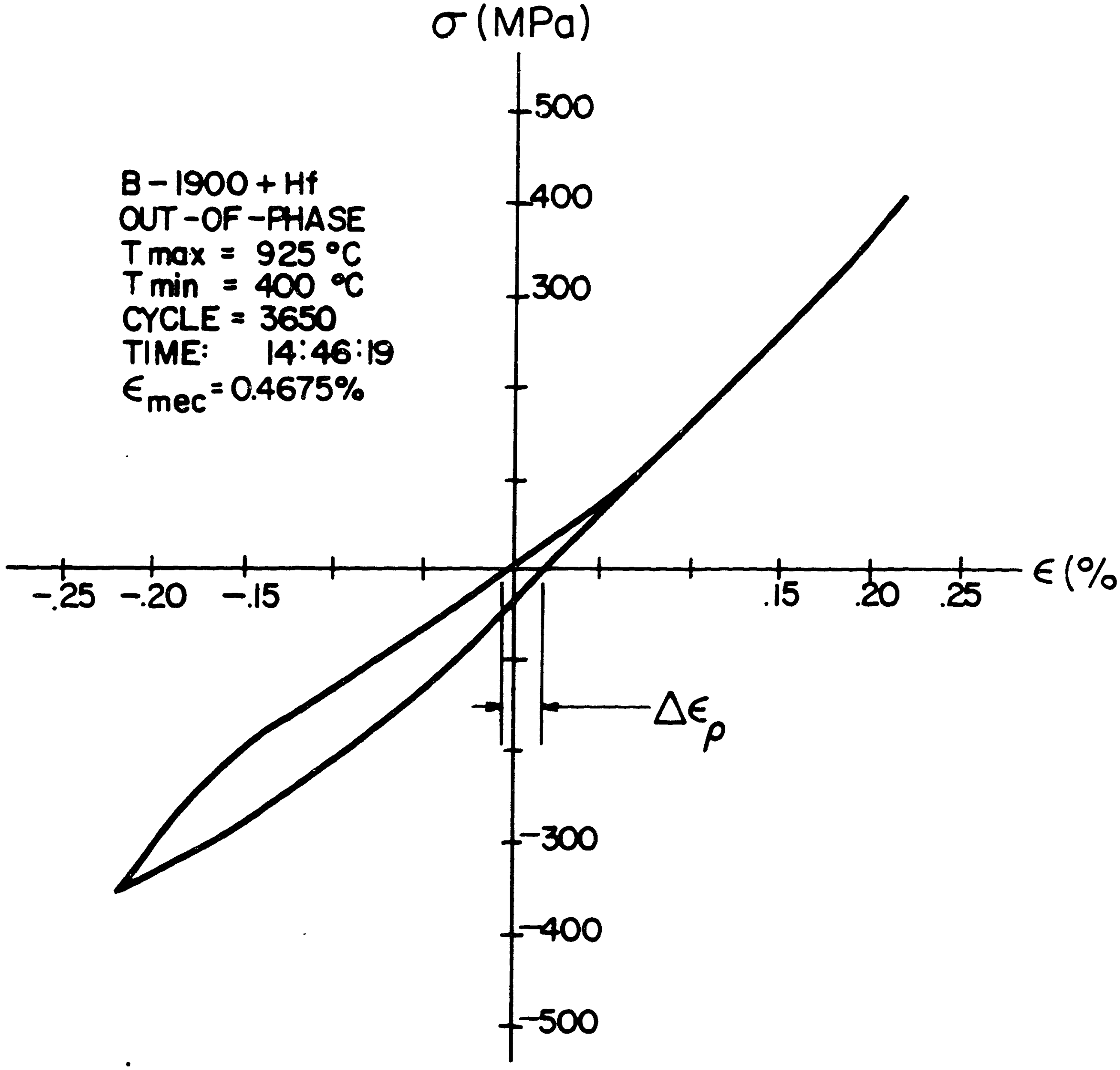


Figure 5.19 Hysteresis loop obtained in B-1900+Hf. (out-of-phase).

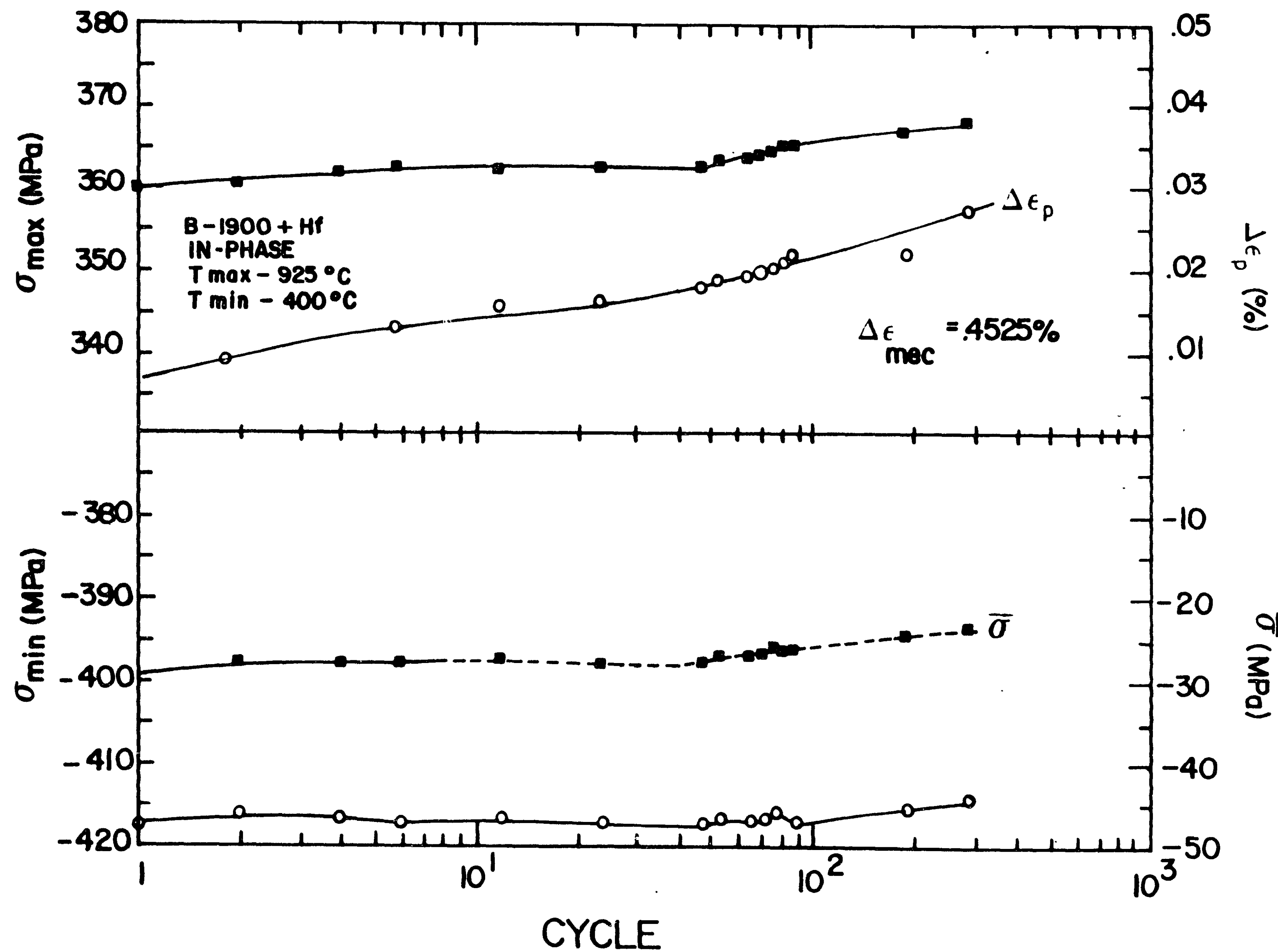


Figure 5.20 Cyclic hardening curves in B-1900+Hf (in-phase, $\Delta\varepsilon_{mec}$ = 0.452 %).

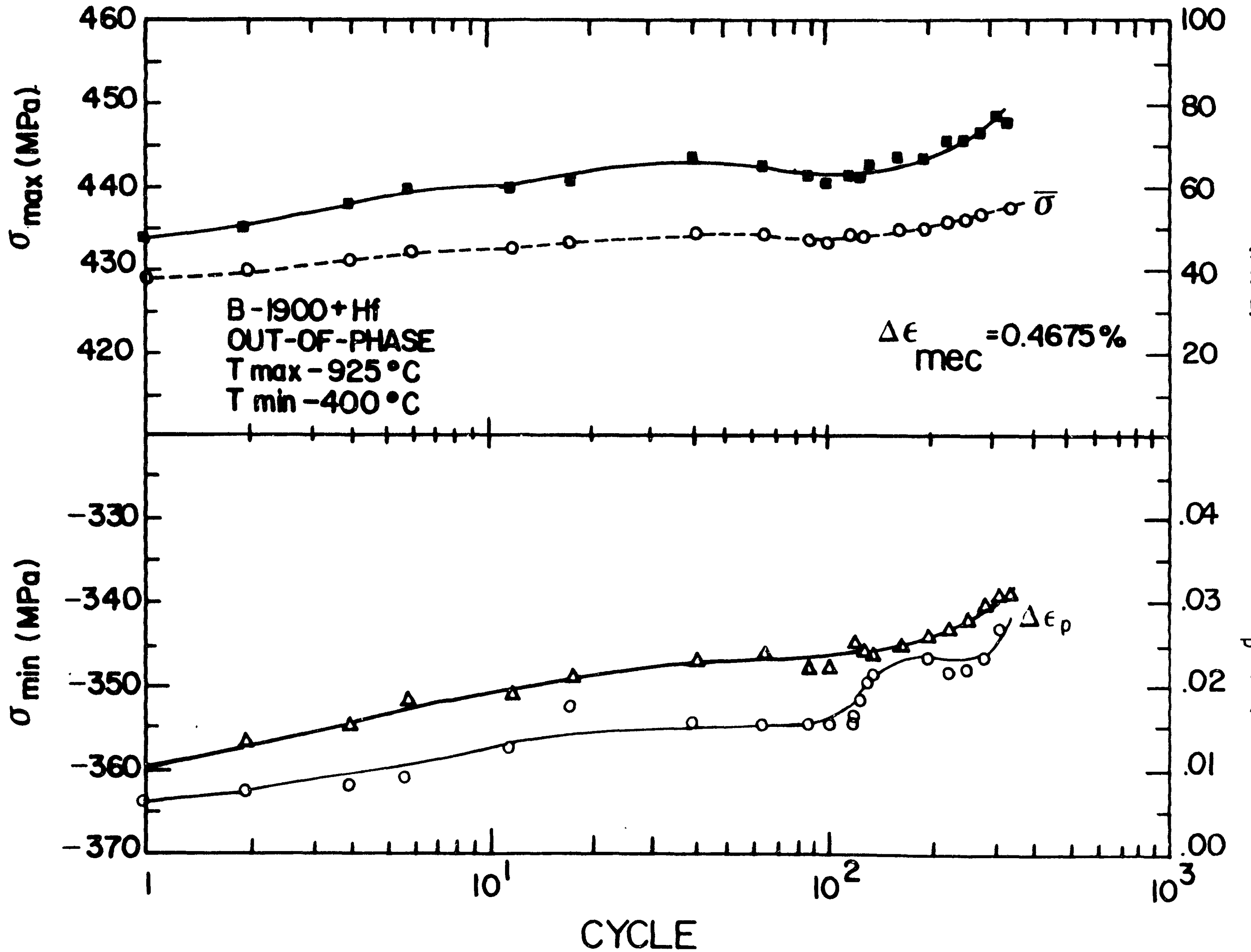


Figure 5.21 Cyclic hardening curves in B-1900+Hf (out-of-phase, $\Delta\epsilon_{mec}$ = 0.4675 %).

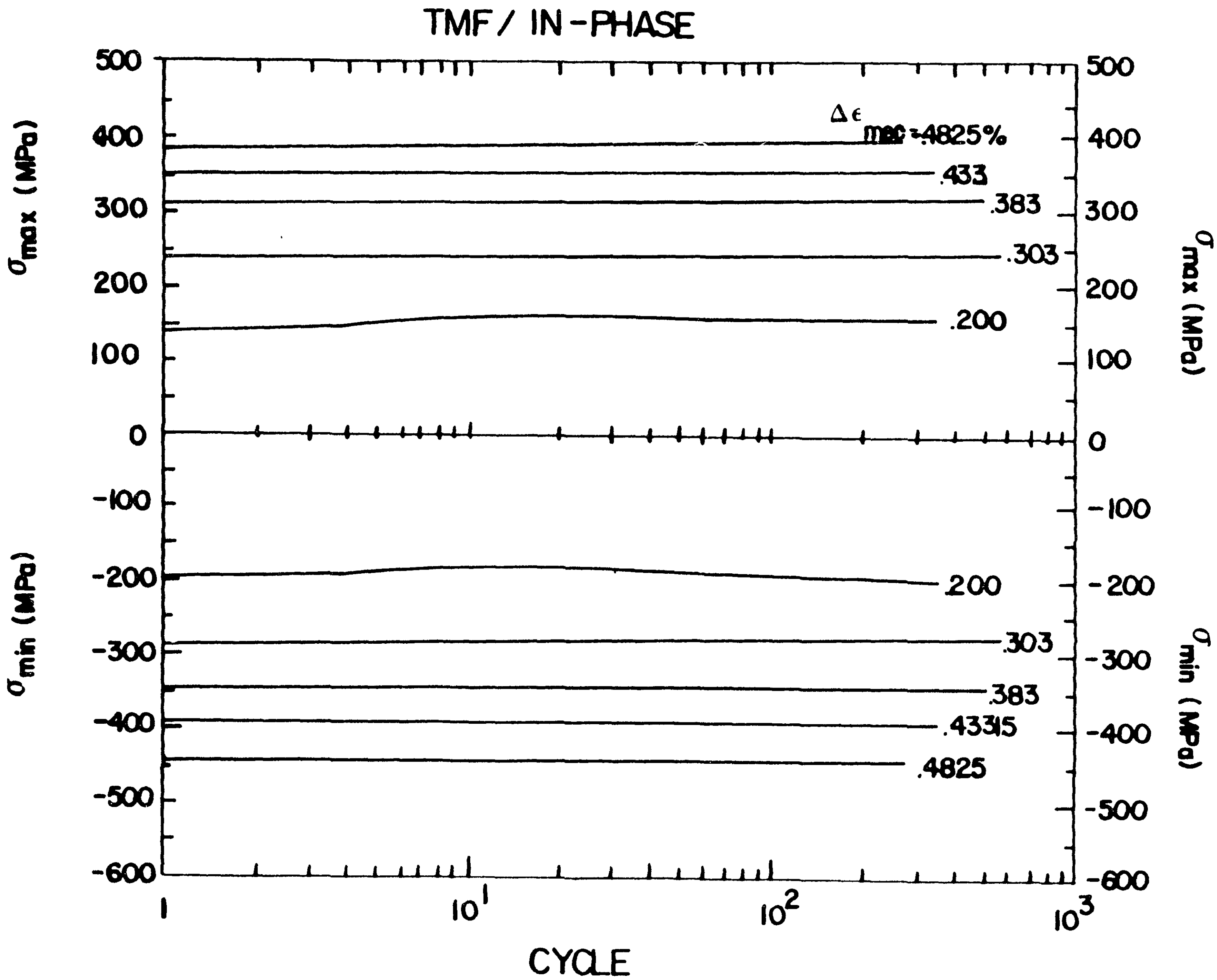


Figure 5.22 Cyclic hardening curves in B-1900+Hf(in-phase).

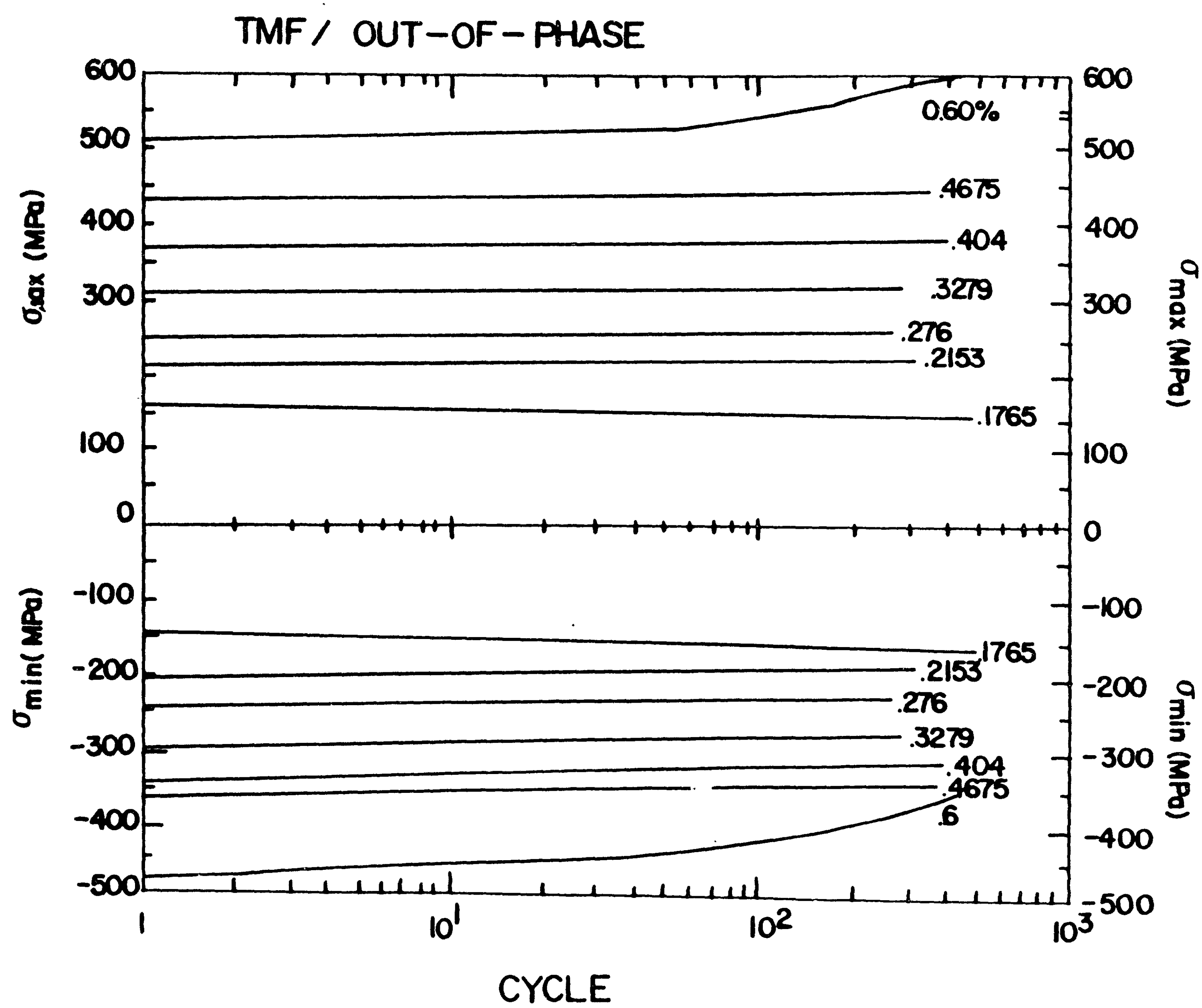


Figure 5.23 Cyclic hardening curves in B-1900+Hf(out-of-phase).

values of tensile stress. On the other hand, because σ_{min} stayed almost unchanged and σ_{max} hardened for in-phase cycling, the net result is a softening of the mean stress, that is $|\bar{\sigma}|$ decreases.

In this series of tests, saturation was defined such that the change in stress range ($\Delta\sigma$), in about fifty cycles, was less than or equal to two percent. The values of σ_{max}, σ_{min} and σ were plotted against the strain amplitude ($\Delta\varepsilon_{tot}/2$) to obtain the cyclic stress-strain (CSS) curves for in-phase and out-of-phase cycling. These curves are shown in Figures 5.24 and 5.25 along with the iso-thermal data obtained at 538 and 871°C [123, 126]. Interesting conclusions can be drawn from Figures 5.24 and 5.25. First, one can see that the CSS curves of in-phase, out-of-phase and isothermal testing converge at low $\Delta\varepsilon_{tot}/2$ (< 0.0012) but diverge as $\Delta\varepsilon_{tot}/2$ increases. At higher strain amplitude but lower than 0.28%, the maximum stress (σ_{max} at 925°C) for in-phase cycling is higher than isothermal fatigue at 871°C. For $\Delta\varepsilon_{tot}$ > 0.28%, the inverse is observed, that is, a higher hardening rate for isothermal fatigue than for σ_{max} of in-phase cycling. The hardening rate of σ_{min} of in-phase cycling (T = 400°C) is identical to the hardening measured for isothermal fatigue at 538°C. For out-of-phase cycling, σ_{min} (at T = 925°C) also show higher hardening rate than isothermal fatigue at 871°C ($\Delta\varepsilon_{tot}/2$ < 0.25%), and a lower hardening rate for $\Delta\varepsilon_{tot}/2$ > 0.25% than isothermal fatigue. However, σ_{max} (at T = 400°C) shows a higher hardening rate than isothermal fatigue at 538°C.

The hardening behavior of in-phase and out-of-phase cycling were compared by plotting. σ_{max} of in-phase, and $|\sigma_{min}|$ of out-of-phase on the same plot (both measured at 925°C). σ_{min} of in-phase and σ_{max} of out-of-phase cycling (both measured at 400°C) were also plotted.

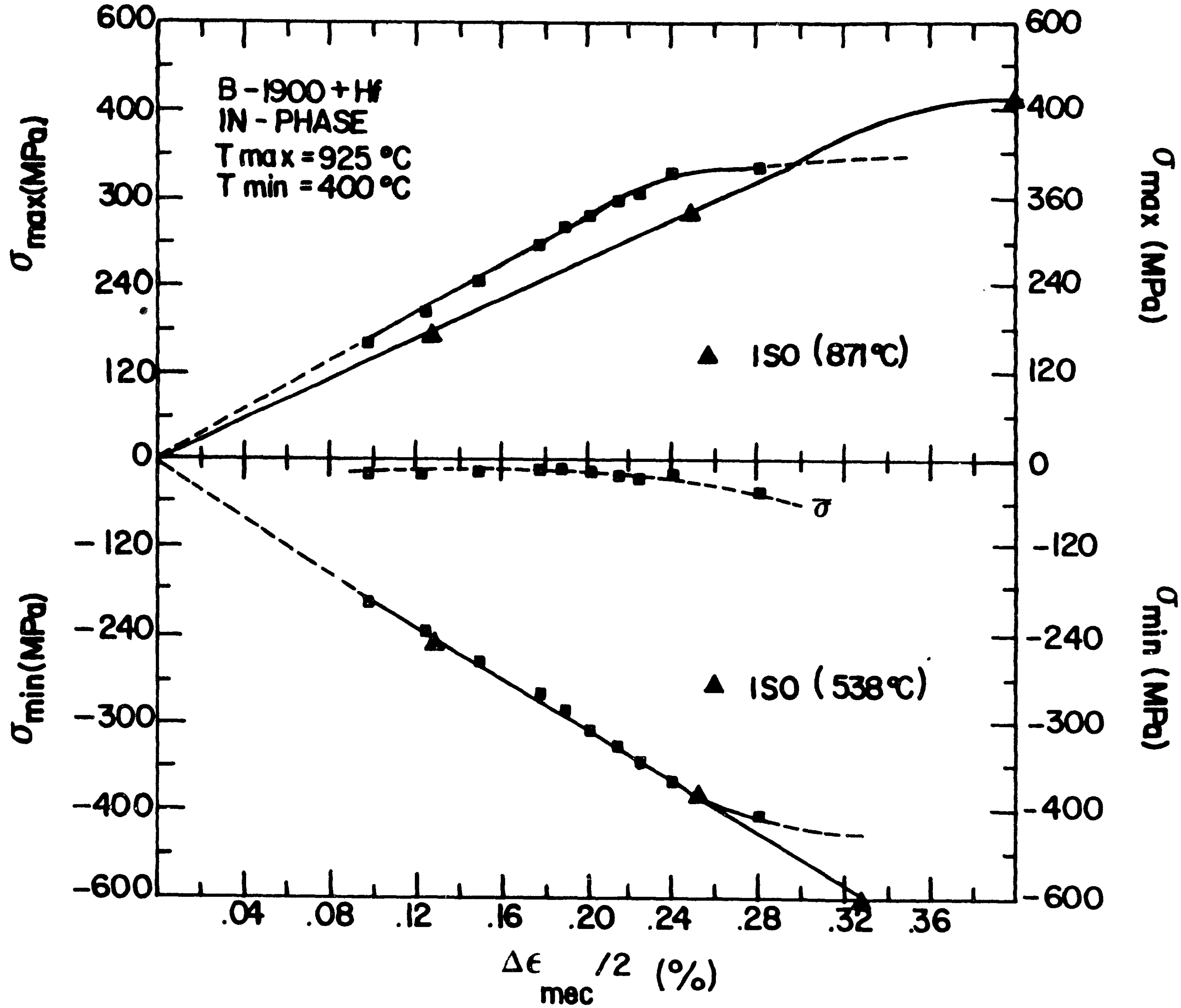


Figure 5.24 Cyclic stress-strain curves in B-1900+Hf. Isothermal (538 and 871 °C) versus in-phase cycling.

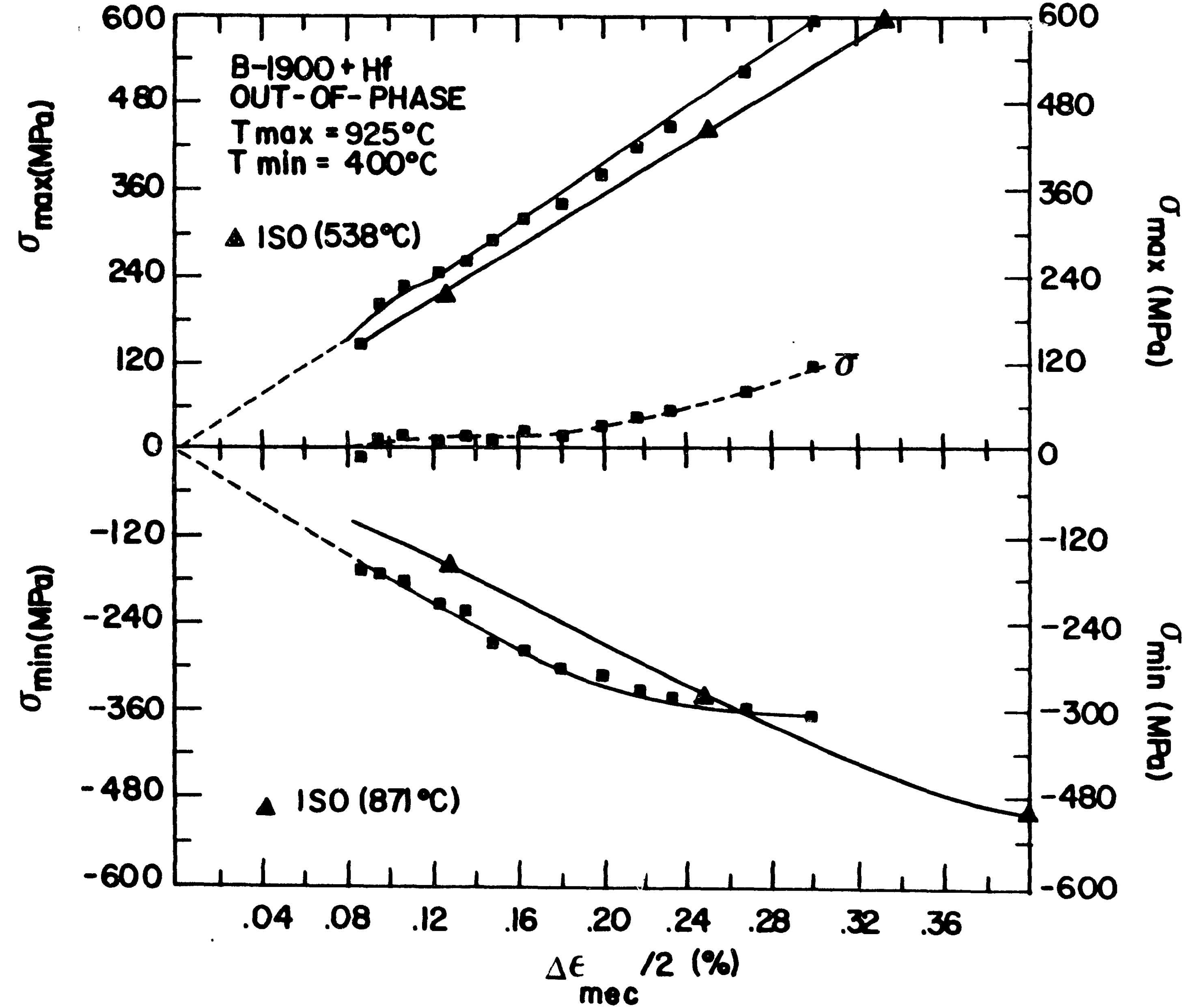


Figure 5.25 Cyclic stress-strain curves in B-1900+Hf. Isothermal (538 and 871 °C) versus out-of-phase cycling.

Figure 5.26 shows the results. One can see that the hardening at 925°C is higher for in-phase (tension) than for out-of-phase cycling (compression). However, the hardening rate at 400°C was higher for out-of-phase (tension) than for in-phase (compression) or isothermal cycling.

In order to clearly identify the process of crack growth, the long transverse and longitudinal sections perpendicular to the fracture surface were mounted for metallographic observation. For out-of-phase cycling, multiple cracks were observed along the gauge length (Figures 5.27a and 5.27b). The propagation path is transgranular and had proceeded interdendritically (Figure 5.27c). Examination of the specimen failed under in-phase cycling revealed a varying degree of transgranular and intergranular cracking (Figures 5.28a and 5.28b) with a density of surface cracks much lower than out-of-phase cycling. The fracture path, however, appears mainly intergranular (Figure 5.28c). These conclusions were supported by SEM fractographic observations.

5.4 Thermal-Mechanical Fatigue Crack Growth Properties

5.4.1 Inconel X-750

The Inconel X-750 specimens were precracked at room temperature under a cyclic stress intensity factor (ΔK_σ) of about 15 MPa $\sqrt{m}$. The ΔK_σ's were calculated with the following expression

$$\Delta K_\sigma = \frac{\Delta N}{BW} (\pi a)^{1/2} \left\{ F_1(\zeta) - 6F_2(\zeta) \cdot \left(\frac{C_{12}(\zeta)}{12(L/W)+C_{22}(\zeta)} \right) \right\} \qquad 5.1$$

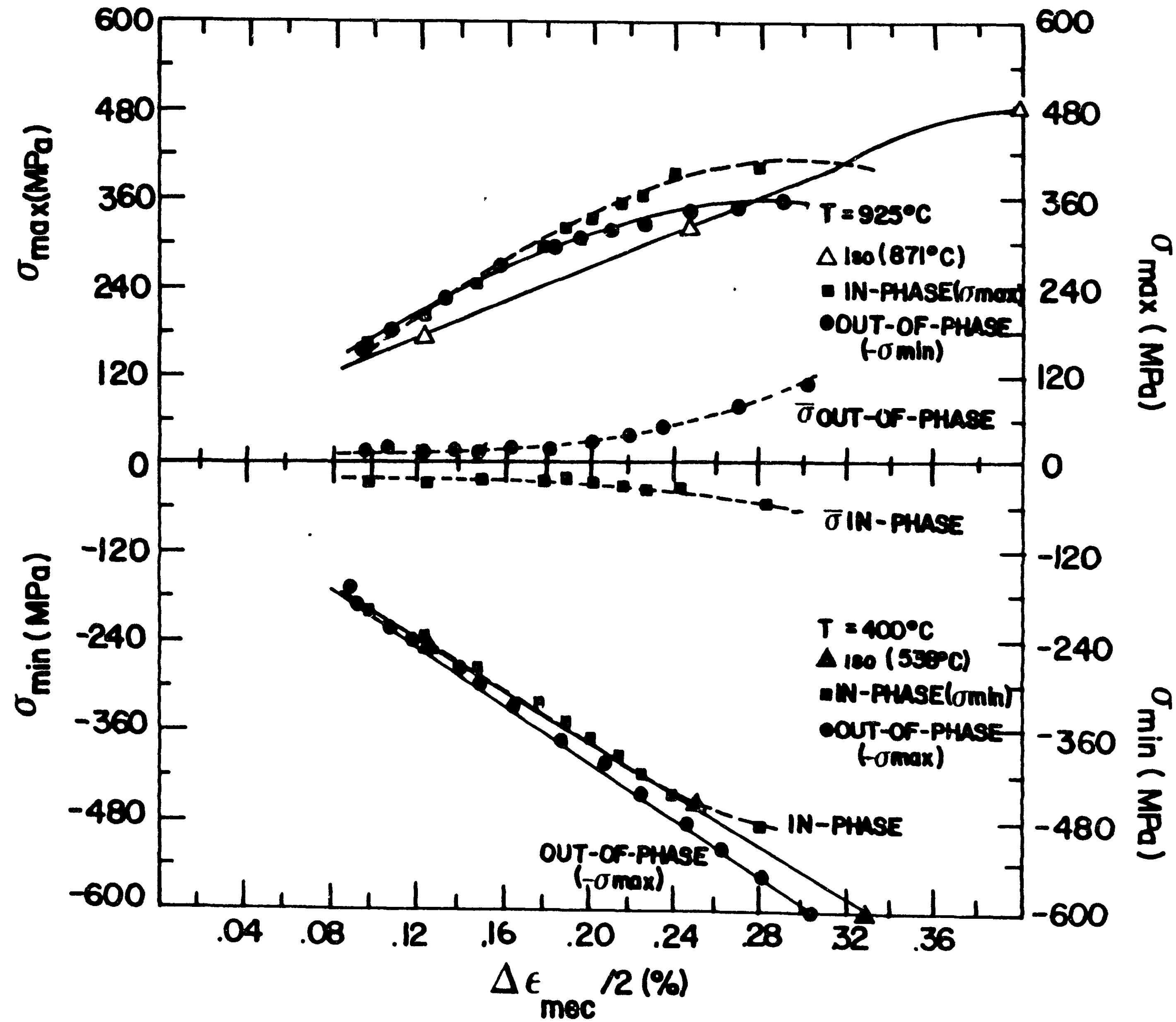


Figure 5.26 Cyclic stress-strain curves in B-1900+Hf. In-phase, out-of-phase and isothermal (538 and 871 °C).

Figure 5.27 Intergranular cracking in B-1900+Hf cycled under in-phase conditions. (a) Longitudinal section, (b) transverse section, (c) intergranular cracks.

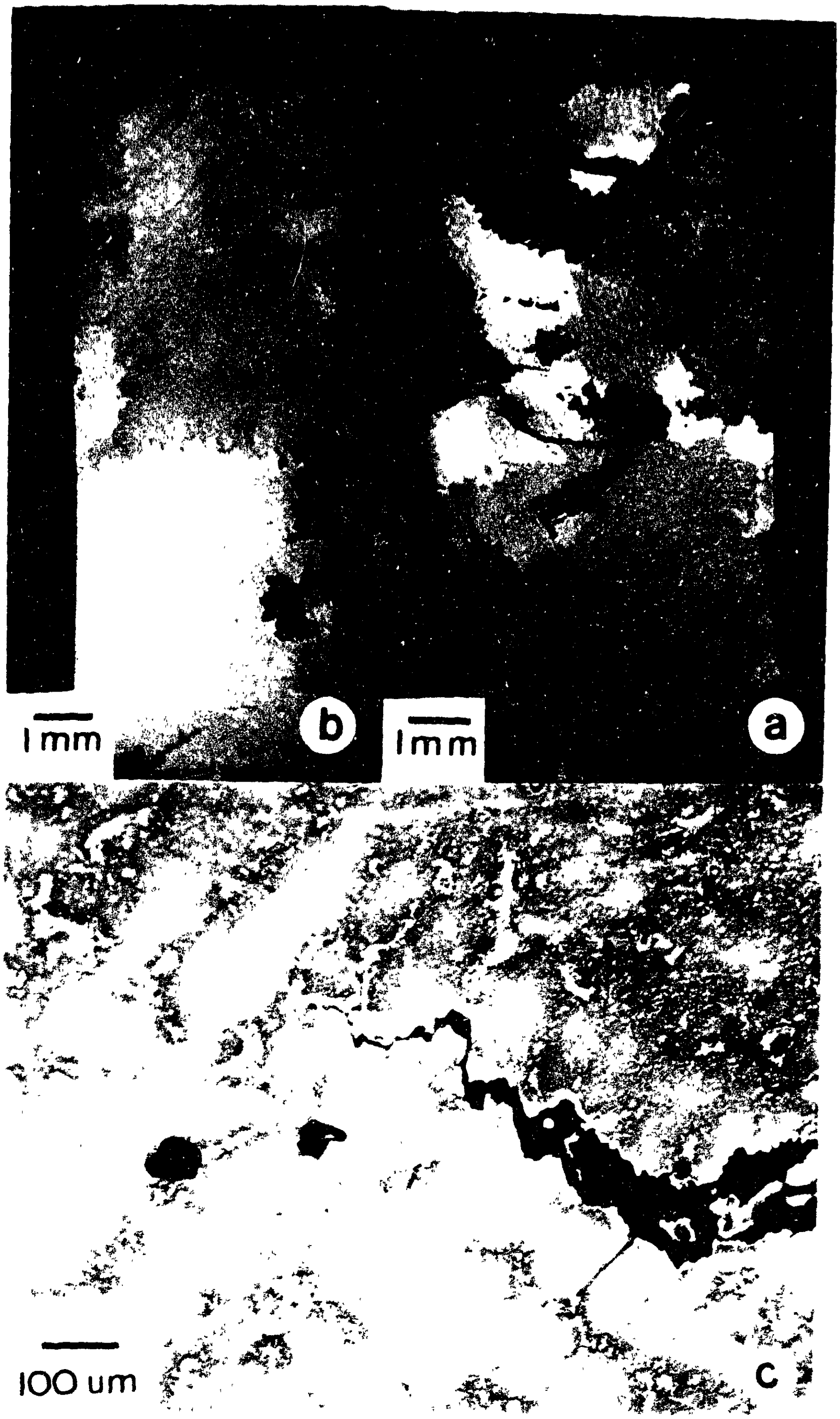


Figure 5.28 Transgranular cracking in B-1900+Hf cycled under out-of-phase conditions. (a) longitudinal section. (b) transverse section. (c) interdendritic cracking.

where $\zeta = a/w$, $F_1(\zeta)$, $F_2(\zeta)$, $C_{11}(\zeta)$, $C_{12}(\zeta)$ and $C_{22}(\zeta)$ geometry correction factors (see Appendix V). This expression was derived for an edge crack in a SEN plate with no bending. For long cracks ($a/w > 0.3$), the error between Eq. 5.1 and Harris' equation [145]

$$\Delta K_\sigma = \frac{\Delta N}{BW} (\pi a)^{1/2} \left\{ \frac{25}{20-13(a/w)-7(a/w)^2} \right\}^{1/2} \qquad 5.2$$

is about 15% and increases steadily with increasing crack length. From the peak potential versus number of curve (see Appendix II), the crack lengths versus N_c (number of cycle) curves were obtained. The crack growth rates were computed using a seven-point incremental polynomial method. Figure 5.29 shows the crack growth data (da/dn) as a function of ΔK_σ ($R = 0.05$) or K_{max} ($R = -1$). This figure shows that TMF cycling is more damaging than isothermal cycling. Furthermore, it shows that fully cycling ($R = -1$) is more damaging than tension-tension cycling ($R = 0.05$) which indicates that compressive straining enhances crack growth by enhancing fracture at the crack tip during the tensile going part of the cycle. The implication of this observation will be discussed further.

Examination of the fractured specimens showed that intergranular cracking was the predominant mode of fracture for the isothermal and in-phase cycled specimens. Out-of-phase cycling shows a transgranular path of fracture.

5.4.2 B-1900+Hf

All TMFCG tests on B-1900+Hf were conducted under strain-controlled conditions. Prior to testing, the specimens were fatigue-

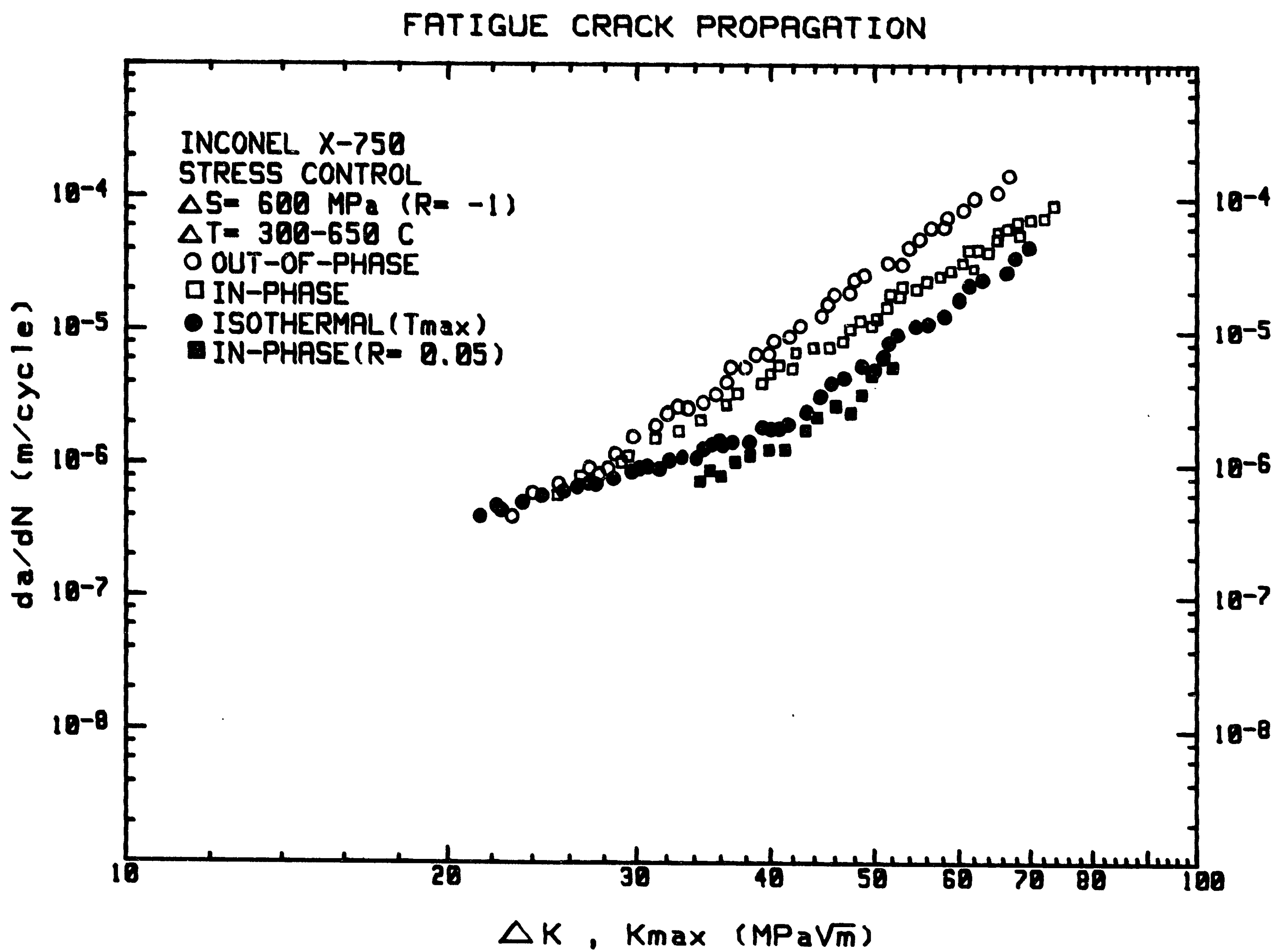


Figure 5.29 Summary of the TMFCG in Inconel X-750 cycled under load-control.

precracked in load control at 10 Hz and room temperature up to a ΔK_σ of about 20 MPa $\sqrt{m}$. Figures 5.30 to 5.32 show the crack growth rates as a function of the strain intensity factor (ΔK_ε) computed from

$$\Delta K_\epsilon = \Delta\epsilon \cdot (\pi a)^{1/2} \cdot \frac{F_1(\xi)\left\{1 - \frac{6F_2(\xi)}{F_1(\xi)} \cdot \frac{\hat{C}_{12}(\xi)}{(12\eta_b + \hat{C}_{22}(\xi))}\right\}}{\left\{1 + \frac{1}{\eta}\left[\hat{C}_{11}(\xi) - \frac{\hat{C}_{12}^2(\xi)}{12\eta_b + \hat{C}_{22}(\xi)} + \frac{6\hat{C}_{12}(\xi)(\eta-\eta_b)}{12\eta_b + \hat{C}_{22}(\xi)}\right]\right\}} \qquad 4.2$$

also derived in Appendix V. There ζ = a/w, η = L/W, η_b = L_b/W and $C_{11}(\zeta)$, $C_{12}(\zeta)$, $C_{22}(\zeta)$, $F_1(\zeta)$ and $F_2(\zeta)$ geometry correction factors. The ΔK_ε were used for convenience because TMF is a strain-controlled process. Although ΔK_ε, as defined by Eq. 4.2, lacks physical meaning under TMF conditions, it is for the case of isothermal and elastic conditions

$$\Delta K_\varepsilon = \Delta K/E' \qquad 4.3$$

where ΔK is the stress-intensity factor derived for fixed-ends displacement loading (see Appendix V). This definition of ΔK_ε was chosen because the conventional definition lacks of mechanistic understanding and overestimates the actual ΔK_ε even under elastic conditions (see Appendix V).

Figure 5.30 shows the TMFCG rates measured during out-of-phase cycling at $\Delta\varepsilon$ = 0.25%. All the data are plotted as to provide an indication of the scatter in the testing procedure. Figures 5.31 and 5.32 summarize the results for $\Delta\varepsilon$ = 0.25% and $\Delta\varepsilon$ = 0.50% at ν = 0.0055 Hz (1/3 cpm). All the FCGR data measured isothermally and during TMF cycling are shown in Appendix IV.

Figure 5.31 shows that TMFCG rates are faster than their isothermal counterpart under fully elastic cycling. At higher strain ranges (Figure 5.32), the opposite behavior is observed, i.e. faster crack

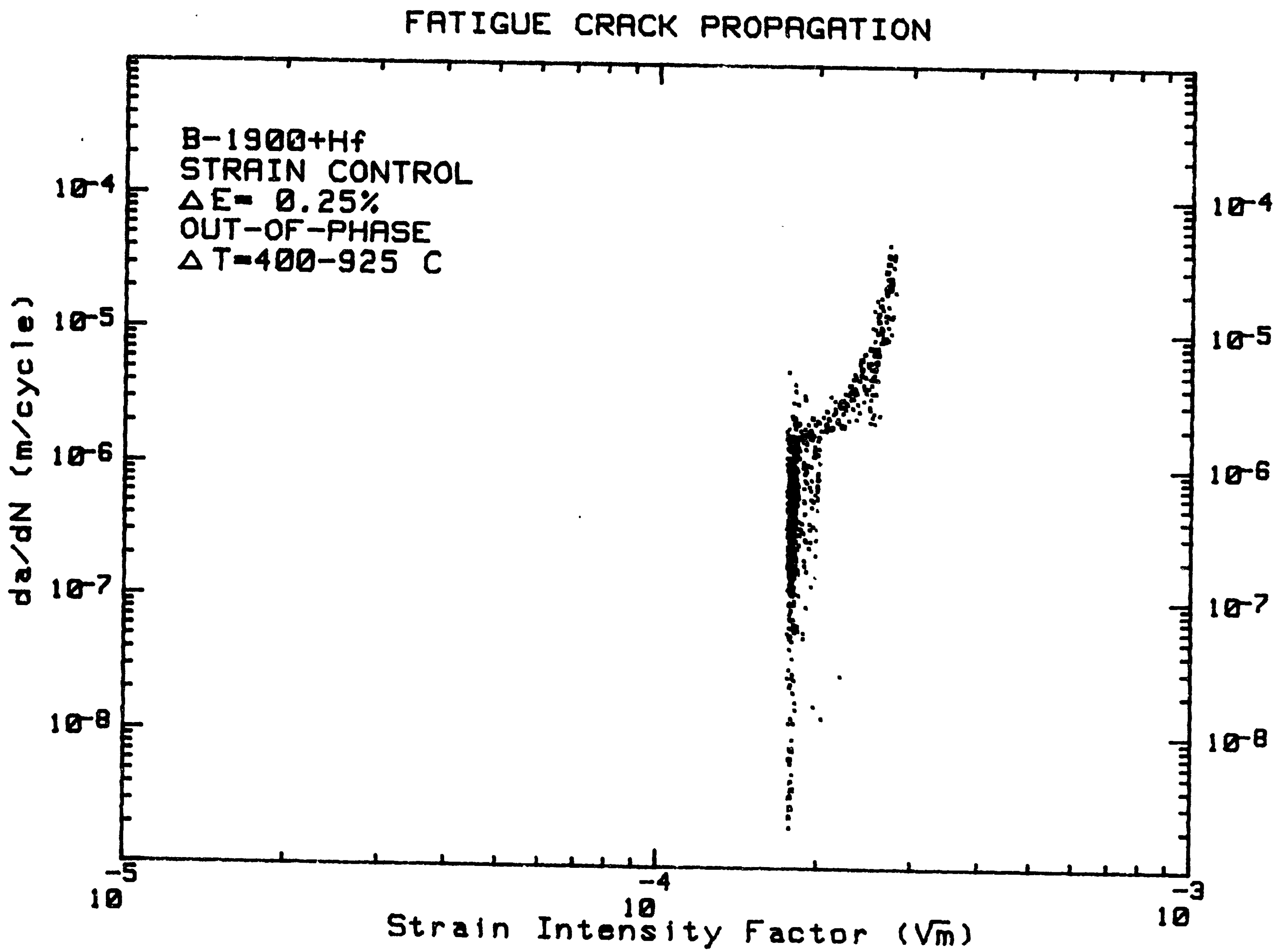


Figure 5.30 TMFCG rates in B-1900+Hf under out-of-phase cycling ($\Delta\varepsilon_{mec}$ = 0.25 %).

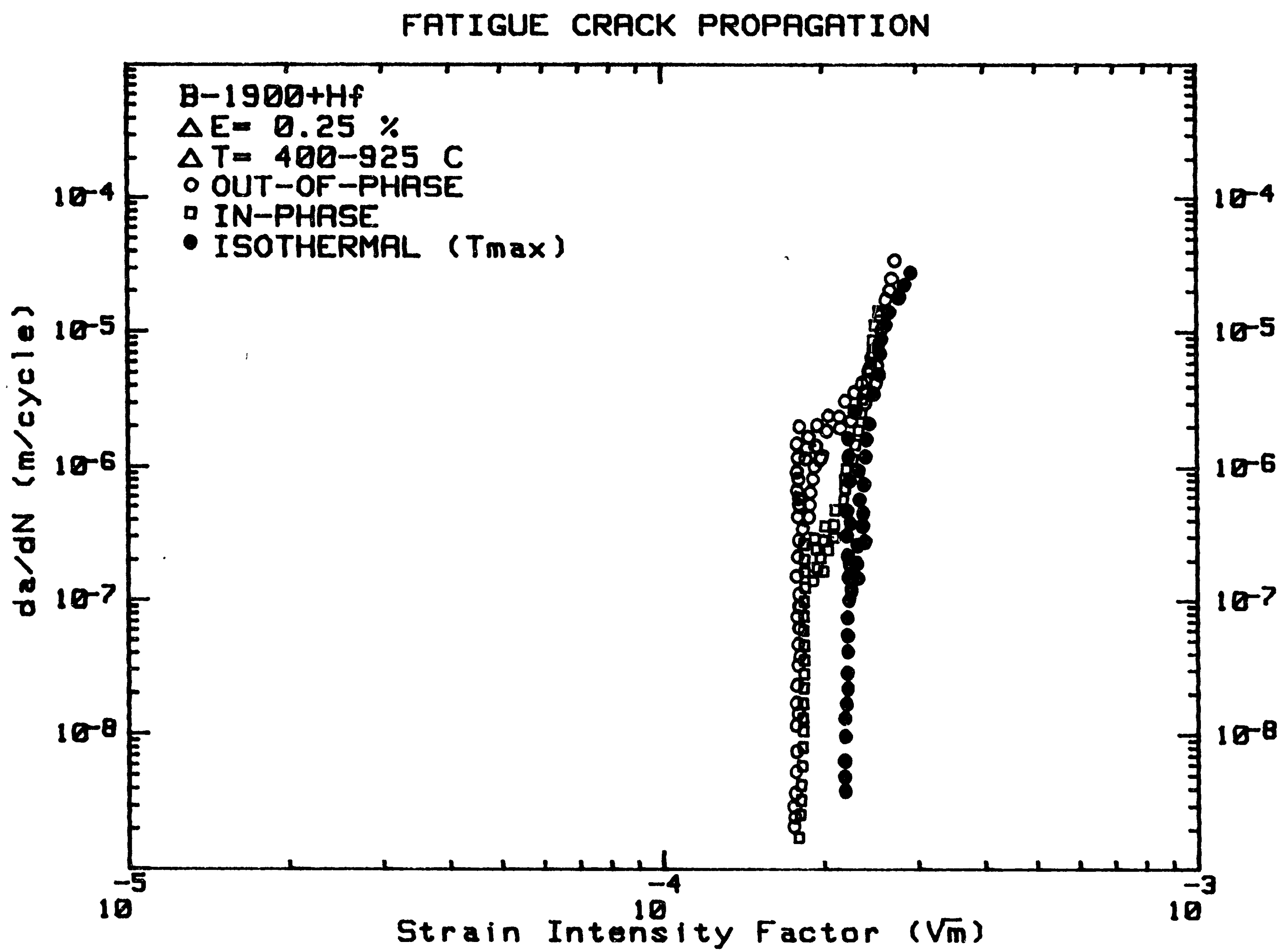


Figure 5.31 Summary of the TMFCG rates in B-1900+Hf cycled under strain-controll ($\Delta\varepsilon_{mec}$ = 0.25 %).

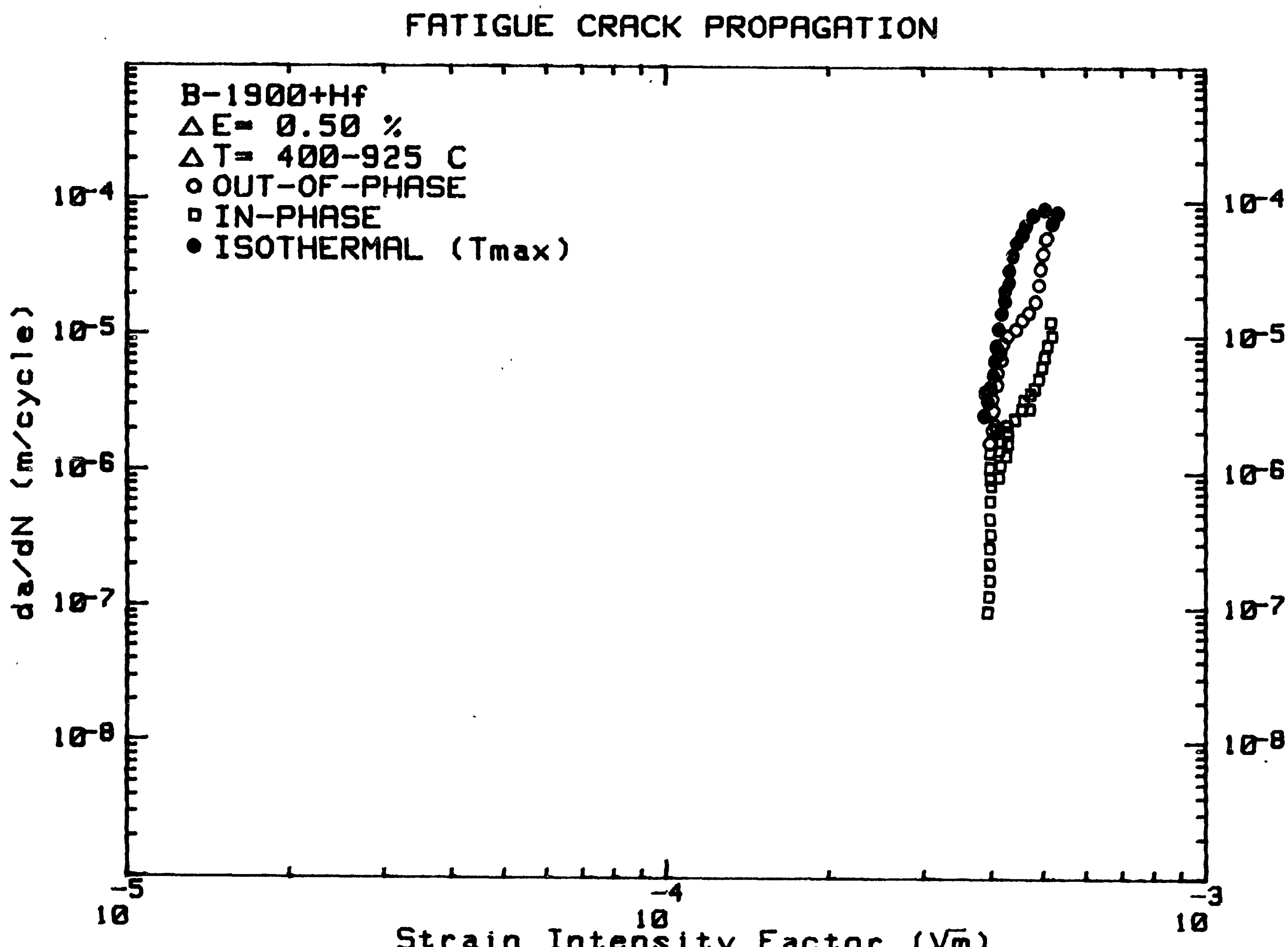


Figure 5.32 Summary of the TMFCG rates in B-1900+Hf cycled under strain-control ($\Delta\varepsilon_{mec}$ = 0.50 %).

growth rates are observed under isothermal conditions than under TMF conditions.

Comparison of Figures 5.31 and 5.32 shows that there is a strain range effect, i.e. the strain intensity factor does not correlate the TMF and isothermal data. In other words, if ΔK_{ε} is the appropriate driving force, then the growth rates data measured at low and high strain ranges would lie on a master curve.

At low strain range (Figure 5.31), a threshold, which delineates a domain of propagation and non-propagation, is observed in all cases. Under fully plastic cycling, a threshold is also observed, although less apparent. The TMF crack growth curves (Figures 5.31-5.32) show three distinct regimes characterized by their respective slopes. The isothermal tests always show two regimes: a threshold and a regime of rapid growth.

Figure 5.33 shows the effect of frequency on the crack propagation rates under displacement-controlled and isothermal conditions (T = 925°C). Clearly, the crack growth rates increase with increasing frequency. As for Inconel X-750, the fracture mode is intergranular under in-phase and isothermal cycling at low frequency. For out-of-phase and isothermal cycling at high frequency (0.1 Hz), the mode of fracture is transgranular.

5.4.3 Hastelloy-X

As for B-1900+Hf, all tests were run under displacement-controlled conditions. The specimens were precracked in fatigue at room temperature prior to testing. Figure 5.34 and 5.35 summarize the

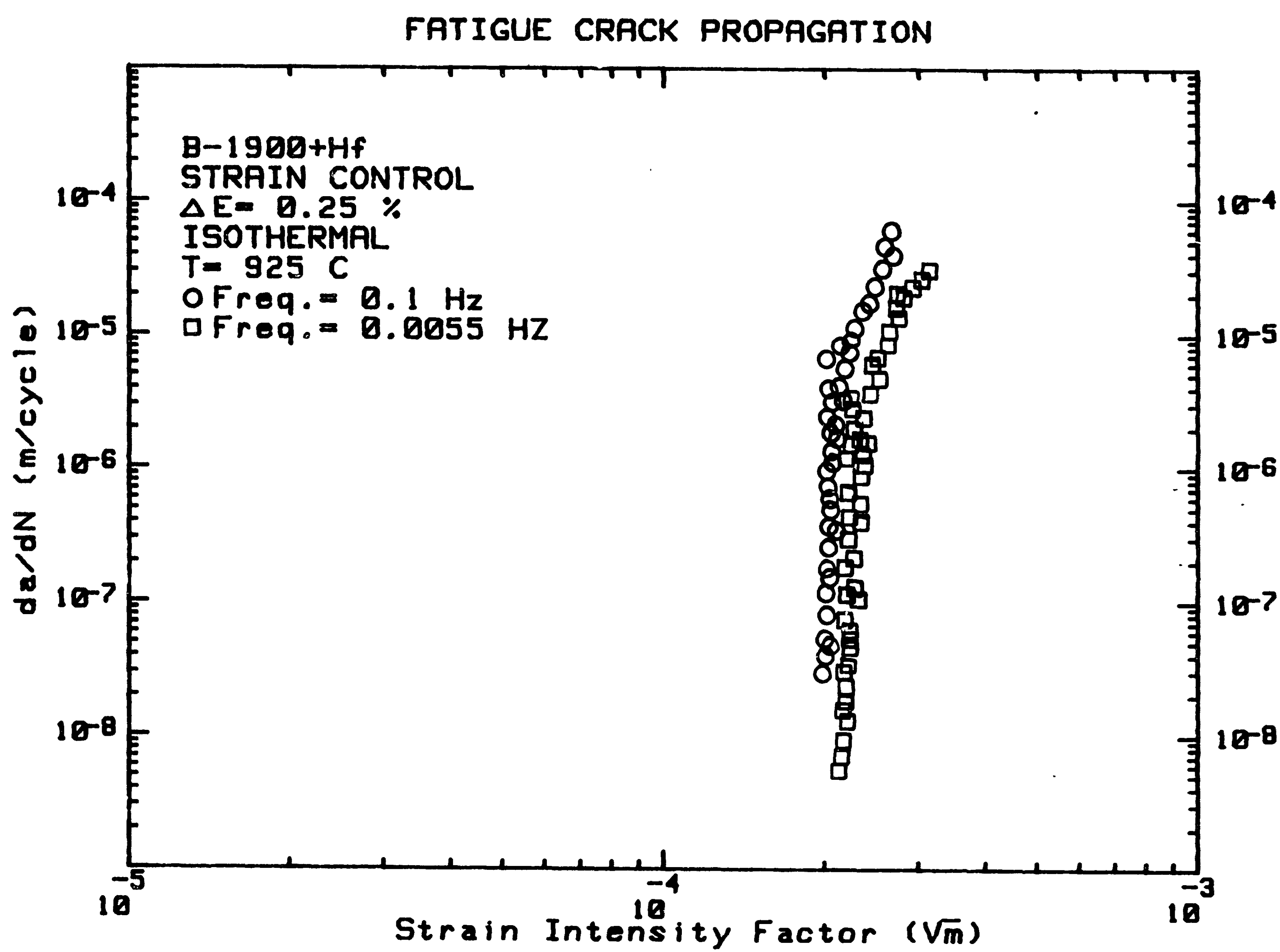


Figure 5.33 Effect of frequency on the FCG rates of B-1900+Hf cycled under strain-control.

crack growth rates as a function of the strain intensity computed with Eq. 4.2. All the FCGR curves measured isothermally and during TMF cycling are shown in Appendix IV.

Figure 5.34 shows that the crack growth rates under TMF cycling are faster than under isothermal cycling at T_{max}. Under fully plastic cycling (see Figure 5.35), the opposite behavior is observed. The crack growth data (Figures 5.34 and 5.35) show that there is a strain range effect indicating, as for the case of B-1900+Hf, that the strain intensity factor is not the universal driving force for TMF crack growth.

The fracture mode at low temperature (isothermal) is transgranular, whereas it is intergranular at high temperature. For TMF cycling, transgranular cracking is observed under out-of-phase conditions, whereas intergranular cracking is observed under in-phase cycling. These observations are consistent with what is observed for Inconel X-750 and B-1900+Hf. This indicates that there is hope to define a parameter that describes the growth rates of a crack under elastic and fully plastic conditions, for both isothermal and thermal-mechanical fatigue.

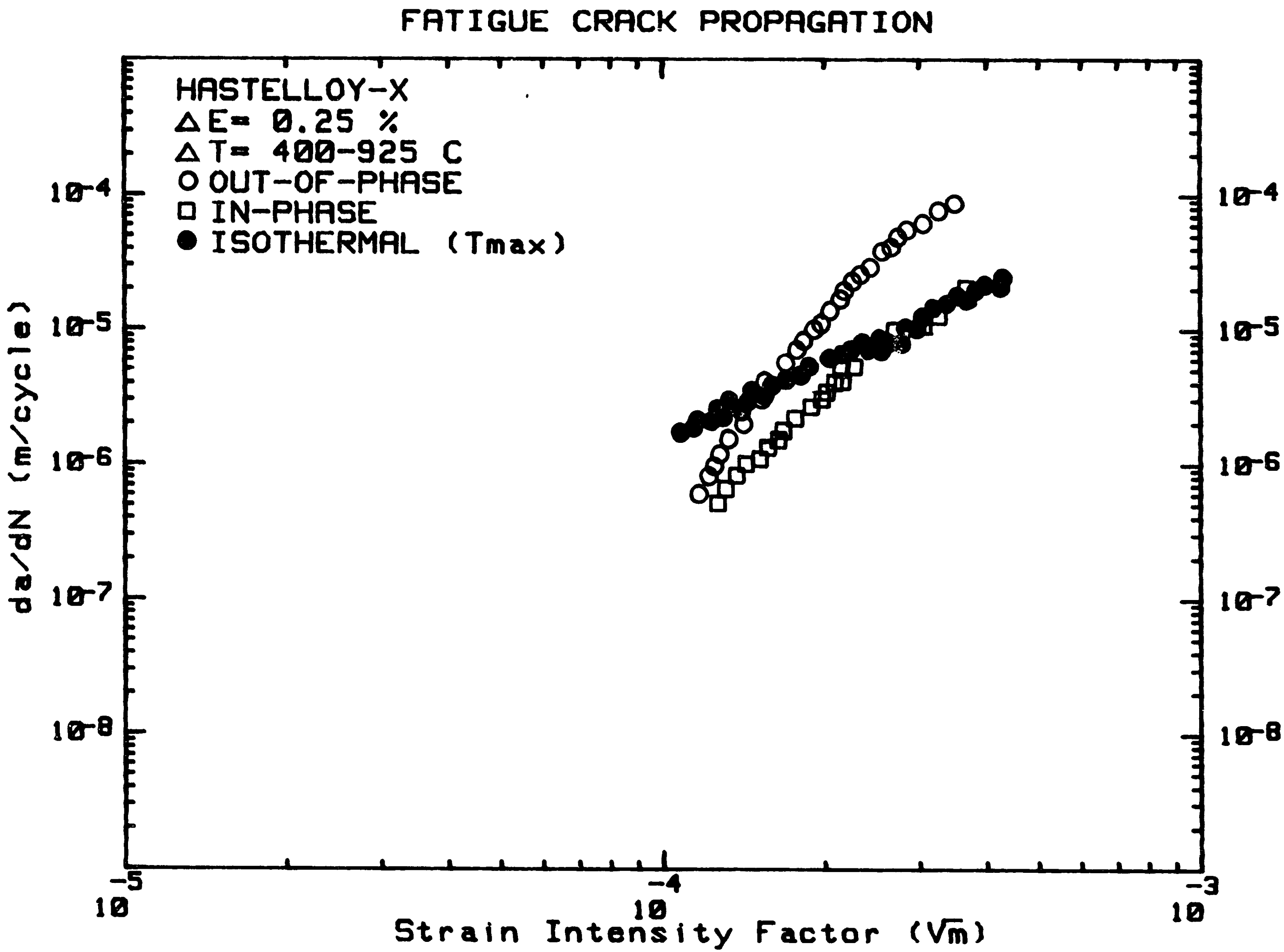


Figure 5.34 Summary of the TMFCG rates in Hastelloy-X cycled under strain-control ($\Delta\epsilon_{mec}$ = 0.25 %).

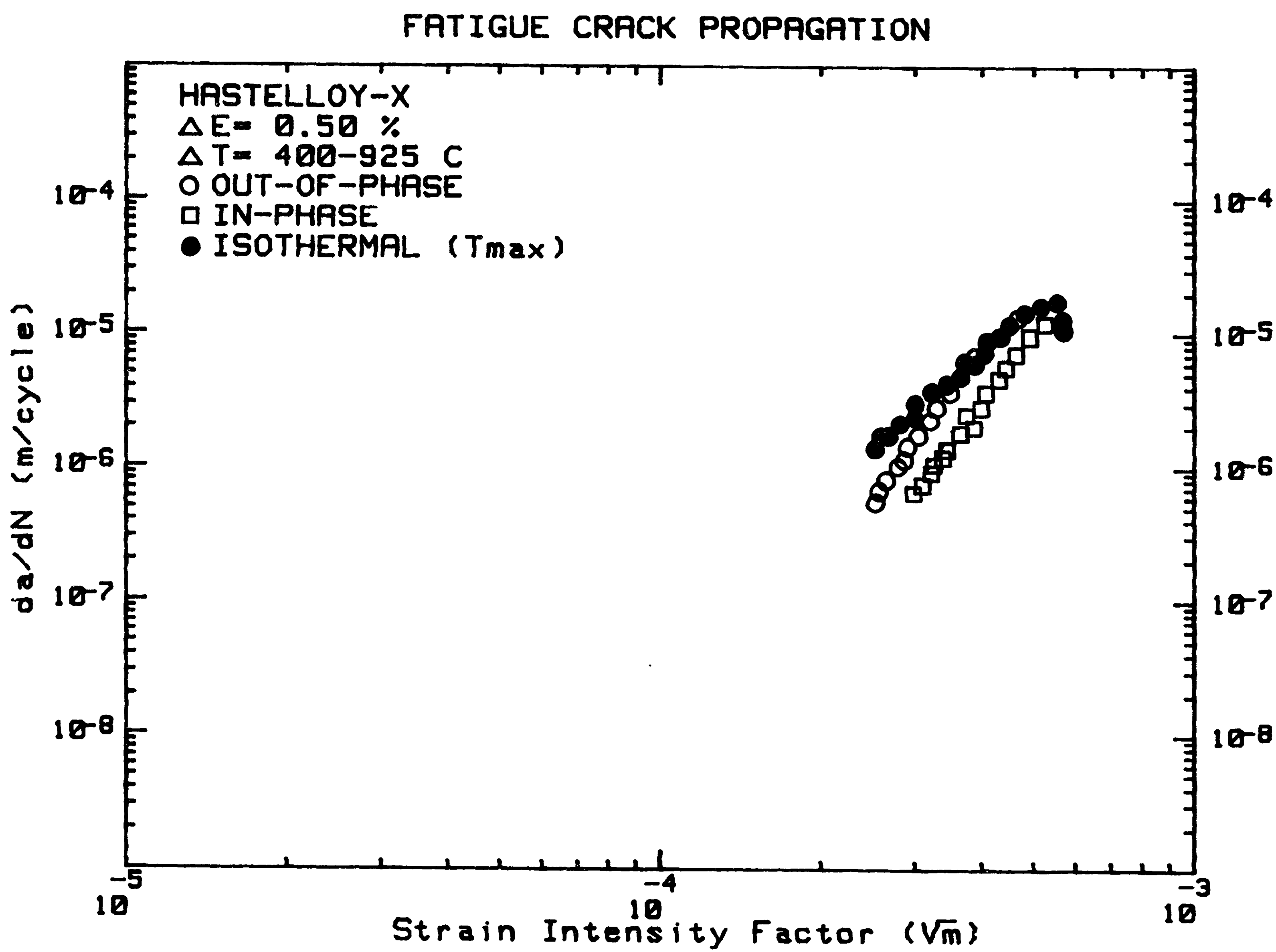


Figure 5.35 Summary of the TMFCG rates in Hastelloy-X cycled under strain-control ($\Delta\varepsilon_{mec}$ = 0.50 %).

6.0 Discussion

6.1 Isothermal Creep and Low Cycle Fatigue Behaviors (Life)

The creep data as shown in Figure 5.5 fall within the anticipated scatter. Extrapolation of the results to stress levels typical of those achieved during fatigue cycling (i.e., $\sigma_{max} < 200$ MPa at T = 925°C) leads to time to fracture of 10,000 hours or longer. Because the corresponding fatigue lives are in the order of 200 hours or lower, it is likely that the creep strains experienced by B-1900+Hf (or Inconel X-750) are negligible, and that most of the time-dependent damage is caused by environmental degradation rather than by creep. In other words, creep is not a significant problem for isothermal and TMF testing of B-1900+Hf and Inconel X-750 as far as the strain and temperature ranges used in this investigation are concerned. Because these strain ranges and temperature ranges are typical of those encountered in many commercial aircraft engines, the same conclusion concerning the relevance of creep damage to life predictions of these components are likely to apply.

For other materials such as Hastelloy-X, creep strains can be of significant importance because even the low stresses achieved during cycling ($\sigma_{max} \approx 100$ MPa at 925°C) can cause creep damage as can be seen in Figures 5.6 and 5.7. In that case, changing the applied strain range not only changes the fatigue response of the material (time-dependent component), but also changes the creep response of the materials. This extra variable will add to the scatter in the results.

The results showing the isothermal life behavior of B-1900+Hf and Hastelloy-X at various temperatures (Figures 5.8-5.13) clearly indicate that low cycle fatigue, expressed in terms of strain range vs number of cycle to failure, is a stochastic process having considerable variability. It is worth noting the scatter in Figure 5.10. As mentioned before (see section 3.5), this is expected since life data depend on initiation/propagation ratios, sensitivity to strain rate, ductility, which in turn depend on the geometry of the specimen, surface finish, etc. It is obvious then, that life prediction based on life data requires the use of a stochastic parameter to take into account this variability. However, the quantitative determination of this parameter requires extensive testing. Because the models based on life data failed to recognize the fact that they are dealing with a stochastic process, the poor correlation between their predictions and the actual life is not surprising [26, 66, 128, 130]. In particular, the SRP (Strain Range Partioning) and FS (Frequency Separation) approaches not only give poor predictions [26, 41, 66, 128, 130], but failed to predict the effect of temperature as shown in Figures 5.14 and 5.15. There are numerous new evidences that the Coffin-Manson approach (SRP, FS, etc.) is virtually identical to the growth law of small crack [149-153]. Furthermore, all these results suggest that we must pay attention to the behavior of small crack in order to solve low-cycle fatigue problems. In other words, a fracture mechanics approach is required for more adequate life predictions.

As pointed out earlier, the life data plotted in terms of total strain ranges instead of the inelastic strain ranges, provide a rationale for the effect of temperature on the fatigue life, i.e. the higher the temperature, the shorter the fatigue life. This indicates that stress-based fracture criteria are likely to be more successful for life prediction that strain-based fracture criteria.

The fact that life data do not provide a safe way to life prediction, does not mean that these tests should not be performed. The value of these tests lies in the fact that they provide valuable information on how a material responds (hardening/softening) to some imposed cyclic conditions. From these tests, appropriate constitutive equations relating cyclic stresses and strains are derived which in turn are used in more advanced life prediction schemes like the damage-tolerant approach.

6.2 The TMF Cyclic Responses of B-1900+Hf

Typical hardening/softening behavior observed under TMF cycling are shown in Figures 5.20 and 5.21. These curves show that σ_{max} (T = 400°C) hardened and σ_{min} (T = 925°C) softened for out-of-phase cycling. During in-phase cycling, σ_{max} (T = 925°C) hardened, whereas σ_{min} (T = 400°C) remains unchanged. During isothermal fatigue, the cyclic flow stress was either unchanged or showed a small amount of softening [123] at high temperatures. Obviously, the damage occurring during TMF and isothermal fatigue are different. To rationalize the observed behaviors, transmission electron microscopy (TEM) of the failed specimens were performed with a JEOL 100CX operated at 120 KeV. The thin foils were taken parallel to the loading axis and at least two foils per specimen were made in order to get a better understanding of the average damage in the bulk specimen. Figures 6.1 and 6.2 show typical dislocations substructures obtained under in-phase and out-of-phase conditions. In all cases, coarsening of

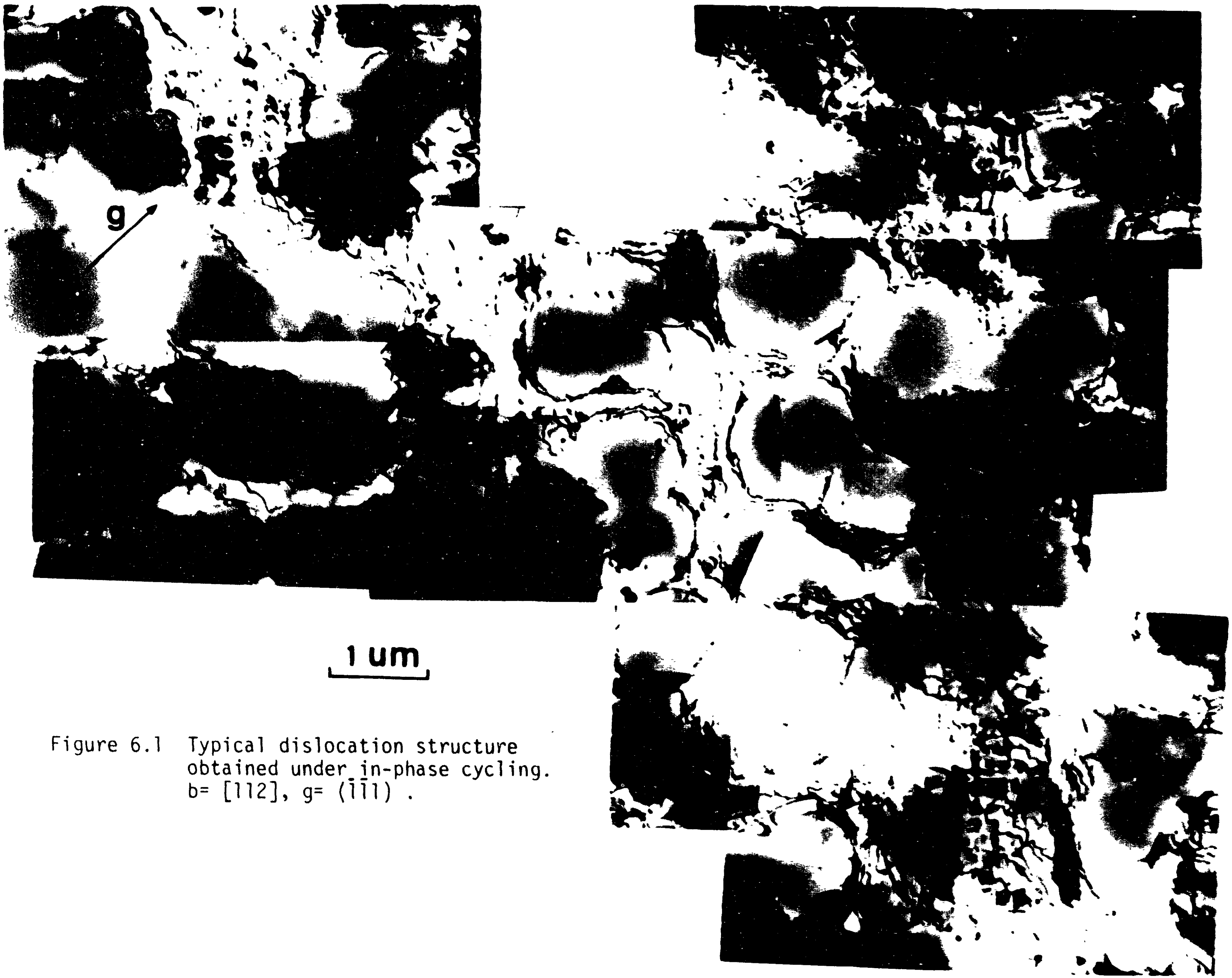


Figure 6.1 Typical dislocation structure obtained under in-phase cycling. b= [112], g= $(\bar{1}\bar{1}1)$.

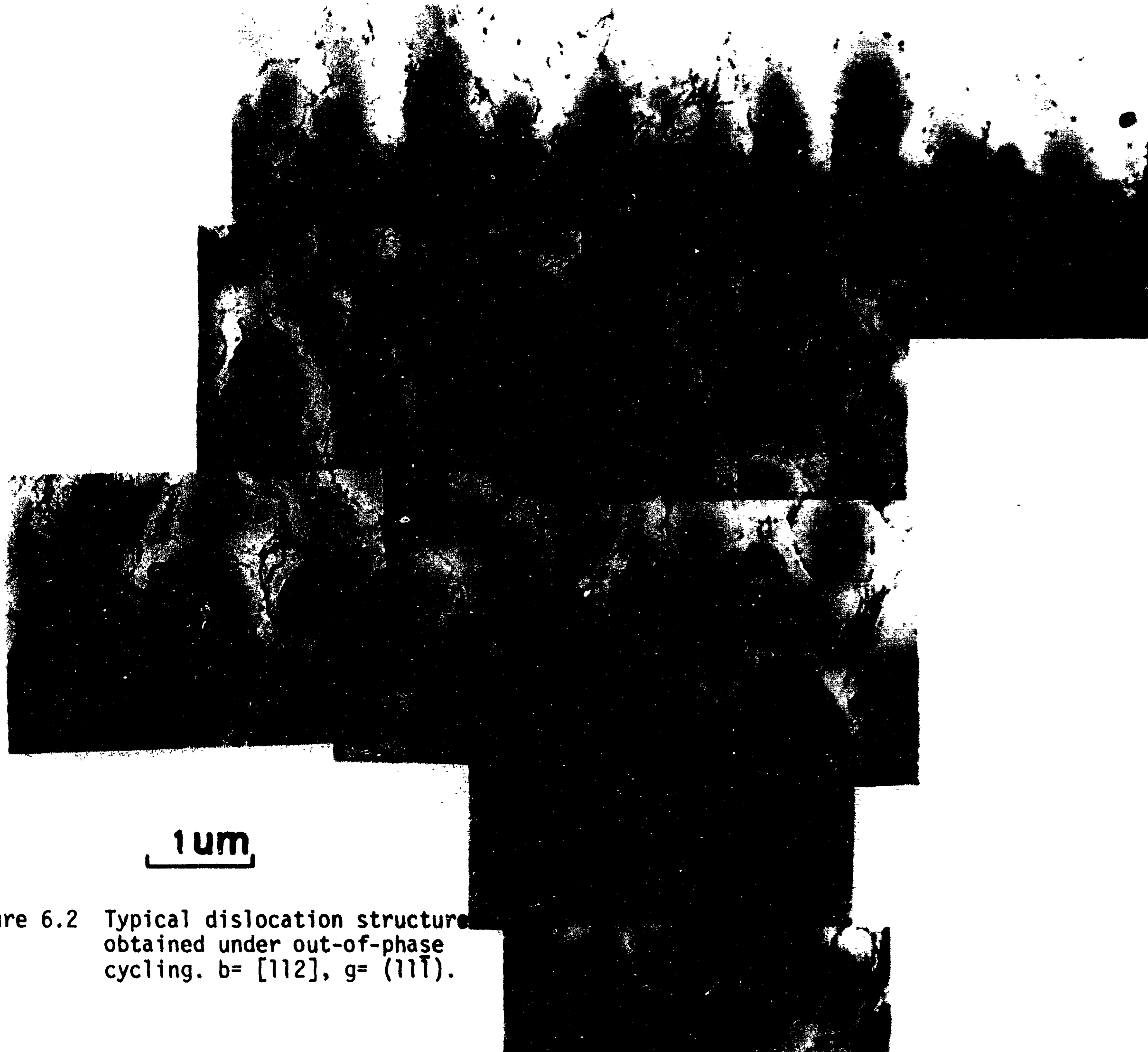


Figure 6.2 Typical dislocation structure obtained under out-of-phase cycling. b= [112], g= $(11\bar{1})$.

the γ' phase has taken place, becoming more pronounced under in-phase conditions. In some grains of the specimens cycled under in-phase conditions, directional coarsening (rafting) is observed (Figure 6.3). Little or no rafting was observed under out-of-phase cycling. A tight dislocation network encapsulating the γ' can be observed in Figures 6.1 and 6.3 (in-phase cycling), whereas a loose dislocation network is observed in Figure 6.2 (out-of-phase cycling) as can be seen from the dislocation spacing. These observations show that the dislocation density is higher under in-phase than under out-of-phase cycling. The octahedral active systems were found to be {111} [1̇10], {111} [101] and possibly {111} [011] in both cases. Three Burgers vectors were identified using the invisibility criterion and assuming that the dislocations were screw in character. These Burgers vectors are a/2[110], a/2[101], and a/2[101] indicating that at least three slip systems were operative. Comparison with specimens failed under isothermal cycling [21] shows that the dislocation density ranked in increasing order; isothermal (T_{min} and T_{max}), out-of-phase, and in-phase cycling.

The above results show that the two major microstructural features are: (1) changes in γ' precipitate morphology (Figures 6.1-6.3), and (2) introduction of a dislocation network about the γ' precipitates. It is well known [132-133] that coarsening of the γ' precipitates influences the mechanical behavior of nickel-base alloys. The task at hand is to separate the relative contribution of each structural feature to the cyclic behavior. In what follows, we will show that the increase in the cyclic flow stress during in-phase cycling is due to dislocation networks surrounding the γ', and the lower flow stress under out-of-

Figure 6.3 Dislocation structure observed in a sample cycled under in-phase conditions showing directional coarsening of the γ' phase. b= [112], g= (111).

phase cycling and pronounced softening behavior (Figure 5.21) is a consequence of not only the dislocation networks strengthening, but also of the directional strain field around the γ' precipitates.

Cyclic Hardening/Softening Behavior

It is generally accepted that coherent particles can be sheared by dislocations and consequently, the work done in forcing the first dislocations through the particles will be important in determining the flow stress. The resistance to shear is governed by several factors:

1. The interaction of the cutting dislocation with the stress field of the precipitates.

2. If the lattice parameters of matrix and precipitate differ, then during shearing of the particles, misfit dislocations must be created at the precipitate-matrix interface. The magnitude of the Burgers vector of the interface dislocation will be the difference between the Burgers vector of the slip dislocation in the matrix and in the precipitate, i.e., $(b_m - b_p)$.

3. If the matrix and precipitate possess different atomic volumes, a hydrostatic interaction would be expected between a moving dislocation and the precipitate.

In the case of superalloys with high volume fraction of γ' (>50%), resistance to γ'- shearing is the primary strengthening mechanism. With the mean free edge-to-edge distance between the precipitates being smaller than the average precipitate size itself, dislocation shearing of the particle is favored over dislocation looping around the particles.

As previously mentioned, coarsening and rafting of γ' develop in this alloy. This feature can be attributed to the large lattice misfit (misfit ~ -0.25% [134]), which generates sufficient interfacial strain to produce misfit dislocations at elevated temperature. Significant deformation can occur only by dislocation penetration of the γ' phase and it is postulated that the misfit dislocation nets at the interface retards this process. Therefore, more hardening should be expected when rafting takes place because the dislocation networks surrounding the γ' are more intense (see Figures 6.1 & 6.3). On the other hand, it has been shown by Shah and Duhl [135] that the flow stress decreases as γ' size increases provided the cubic shape of the γ' is conserved. These two superimposed phenomena, with opposite effects with regard to the flow stress, determine the apparent flow stress. Figures 5.20 and 5.21 shows that the maximum stress (σ_{max}) of in-phase cycling (T_{max}) display continuous hardening with little appearance of stabilization. On the other hand, the σ_{min} curve of out-of-phase (T_{max}) displays continuous softening. This behavior has been previously observed on B-1900+Hf cycled in TMF [126] where continuous hardening of σ_{max} and softening of σ_{min} occurred until final fracture without evidence of saturation. This raises the question of the validity of the strain increment technique for measuring the CSS curves of low stacking-fault energy materials where continuous hardening (or softening) is observed until fracture [126, 136-137] at low applied strain. This behavior is usually rationalized in terms of planar configuration of dislocations [136, 138]. At high strain, where cross-slip takes place, saturation of the CSS curves is observed. Although B-1900+Hf is also a low stacking fault energy material, the planar configuration of dislocations alone cannot explain the observed cyclic hardening/softening behavior (Figures 5.20-5.21).

At high temperature (T ≥ 600°C) the flow stress depends on the APB energy and thermally activated cross slip of the glide dislocations [135, 139]. The directionality of the internal stress field around the γ' is small because thermal activation is important. Therefore, the flow stress will depend on the factors controlling the internal stress. The density of misfit dislocations around the γ' particles, which depends on the γ' size and shape (rafting), affects the flow stress because it controls the internal stress on the glide dislocations. On the other hand, when coarsening takes place, the particle spacing increases, which leads to weakening because of the increased probability of avoiding shearing by Orowan-type mechanisms [140]. Under in-phase cycling, coarsening of the γ' takes place along with rafting, leading to high density of misfit dislocations without significant increase in interparticle spacing (Figure 6.3). The net result is an increase in the flow stress because the internal stress increases faster than the relaxation time required by the glide dislocation to overcome the barrier created by the misfit dislocations. During out-of-phase cycling, isotropic coarsening takes place leading to a smaller increase of misfit dislocations and a more significant increase of γ' particle spacing. The net result is a decrease in flow stress.

At low temperature the directionality of the resultant stress field around the γ' particles will also contribute to the internal stress acting on the glide dislocations [133, 135, 141]. With coarsening, the hydrostatic tensile stress field around the γ' increases and that is one of the reasons why misfit dislocations are required. It is well known that the resistance to the movement of glide dislocations (partials), i.e., the frictional force, increases with increasing hydrostatic stress [142]. Therefore, the superposition of an external hydrostatic stress field will increase or decrease the flow stress depending on the magnitude and sign of the applied stress. If an external tensile stress field is applied, the frictional stresses increase and so does the flow stress. This corresponds to the out-of-phase cycling case where the applied stresses are tensile at low temperature (see Figure 5.21). When the applied stress field is compressive, the flow stress decreases or remains unchanged, depending on the magnitude of the net stress field [135]. During in-phase cycling, the stresses are compressive at T_{min} and cancelled with the hydrostatic tensile stress field around the γ'. The net result is that the frictional forces on the glide dislocation are lower and the flow stress does not change much as cycling proceeds and coarsening takes place (at T_{max}), as can be seen in Figure 5.20.

To fully understand the flow behavior with change in sign of the applied stress, we must also consider the resolved constriction stress of partial dislocations [135, 139]. The direction of the glide force per unit length, $F_{g/L}$, will reverse upon reversing the applied stress, σ [135]. While this has no physical meaning in a macroscopic sense, since it only alters the direction of glide for a particular dislocation, reversing the direction of the glide forces acting on the partial dislocation, it leads to a distinctly different physical situation. The resulting force tends to constrict the partials under an applied tensile stress and extend them under a compressive stress [135]. Since constriction of the partials is required by the cross-slip process, the flow stress appears stronger in tension than in compression where the extended partials retard cross-slip activity. The previous argument implies that the flow behavior of B-1900+Hf is governed by octahedral slip activity. If such is the case, the flow stress can be written as

$$\sigma \propto (1/R + 1/\lambda) , \qquad (6.1)$$

where R is the particle size and λ the mean free distance between the two constricted nodes where the cross-slip event occurs [135]. At high temperature, λ is always smaller than R and the flow stress is governed by the dislocation network and internal stress which fixed λ [135, 143]. At low temperature, the flow stress depends on which of these two parameters (R, λ) is the smaller. If a tensile stress is applied, the partials tend to be constricted and the mean free distance between cross-slip event decreases. If a compressive stress is applied, the partials are pulled further apart (λ increases), and the flow stress is controlled by R, the particle size.

Cyclic Stress-Strain Behavior

The cyclic stress-strain curves (Figures 5.24-5.26) show that the flow stress, as a function of the strain amplitude ($\Delta \varepsilon_t/2$), is higher for in-phase than out-of-phase or isothermal cycling at the maximum temperature (T_{max}). This is consistent with the fact that dislocation density increases in the order of isothermal, out-of-phase and in-phase cycling (see Figures 6.1-6.3). At low temperature (T_{min}), because cross-slip is a function of the magnitude and sign of applied stress, the flow stress is higher under out-of-phase cycling than under in-phase or isothermal cycling (see Figure 5.26).

The issue that needs to be addressed now is the cracking mode. As shown in Figures 5.27 and 5.28, fracture is transgranular and proceeds interdentritically in out-of-phase cycling and intergranularly under in-phase cycling conditions. As expected, fracture is controlled by the favored mode of rupture in the tensile part of the cycle. Under in-phase cycling, tension occurs at high temperature where the cohesive strength of the grain boundary is low, which obviously promotes intergranular cracking. During out-of-phase cycling, the specimen is under tensile loading at low temperature and the weakest transgranular features (carbide film, secondary dendrites, inclusion stringers, etc.) control the rupture mode. The fact that the fracture mode depends on the maximum tensile stress, rather than on a critical plastic strain range, suggests that the failure criteria for B-1900+Hf is stress-based rather than strain-dependent. In other words, the testing condition (temperature and strain relationship) leading to the maximum tensile stress, will determine the number of cycles to initiation and the propagation rates.

As mentioned earlier, the major critical turbine components operate under strain-controlled conditions and, more specifically, under displacement-control. The stresses are not known a priori. Therefore, if the failure criteria in B-1900+Hf is stress-dependent, the relative crack growth rates for a given crack length can be determined from the CSS curves. Figure 5.26 shows that at low strain amplitude (<0.25%) the stress range for TMF cycling is higher than for isothermal fatigue. Consequently, faster crack growth rates should be obtained for TMF cycling. Under fully plastic conditions ($\Delta \varepsilon_t/2$ > 0.25%), Figure 5.26 also shows that the isothermal stress range is higher than the stress ranges obtained under TMF cycling and faster crack growth rates are expected under isothermal cycling (T_{max}).

6.3 Fatigue Crack Growth (FCG) of Inconel X-750

The results (Figure 5.29) have shown that the crack growth rates are higher for TMF cycling than for the equivalent isothermal condition ($T = 650°C$) which is in agreement with the results obtained on other nickel-base alloys [22, 27-29, 41, 58]. Secondly, it was observed that the crack growth rates are higher for $R = -1$ than for $R = 0.05$, which indicates that compressive stresses play an important role in the mechanics of TMFCG. Comparison between out-of-phase and in-phase cycling at $R = -1$ shows that out-of-phase cycling is more damaging than in-phase cycling at high ΔK, whereas at low ΔK, the crack growth rates are the same. The rationale of this behavior was found in the analysis of potential change within each cycle. At low ΔK, the potential curves $V(\sigma)$ which characterizes the crack geometry, i.e. crack length and crack configuration (see Appendix II), are identical for out-of-phase and in-phase cycling as can be seen in Figure 6.4. This indicates that the mechanical driving force (same stress, crack length and crack configuration) are similar and therefore identical crack growth rates are expected. As ΔK increases, the $V(\sigma)$ potential curves for both in-phase and out-of-phase cycling display different characteristic features. Figure 6.5 shows $V(\sigma)$, $V(\sigma,T)$ and $V(T)$ measured at $\Delta K \simeq 50$ MPa $\sqrt{m}$. The in-phase $V(\sigma)$ curve shows a smooth increase with σ_{net} (σ_{net} = N/BW) followed by a sharp increase near σ_{max}. The potential then remains quasi-stable as σ_{net} decreases and sharply falls as σ_{net} approaches zero. In the compression regime the potential smoothly decreases reaching a minimum at σ_{min}, and finally increases as the stress increases again. On the other hand, the out-of-phase $V(\sigma)$

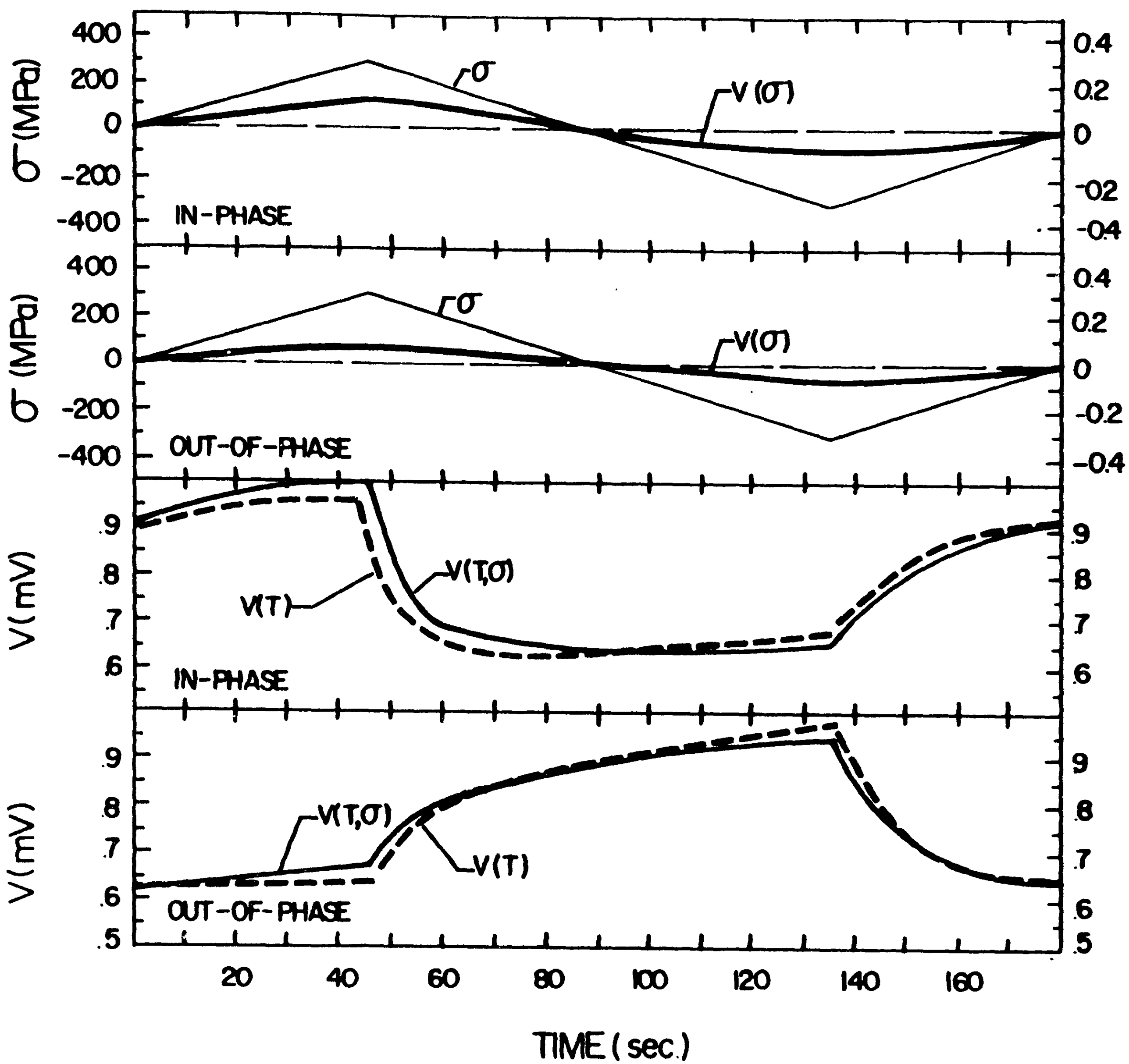


Figure 6.4 Potential curves $V(T,\sigma)$, $V(T)$, and $V(\sigma)$ for TMFCG in Inconel X-750. $\Delta K = 25\ MPa\sqrt{m}$.

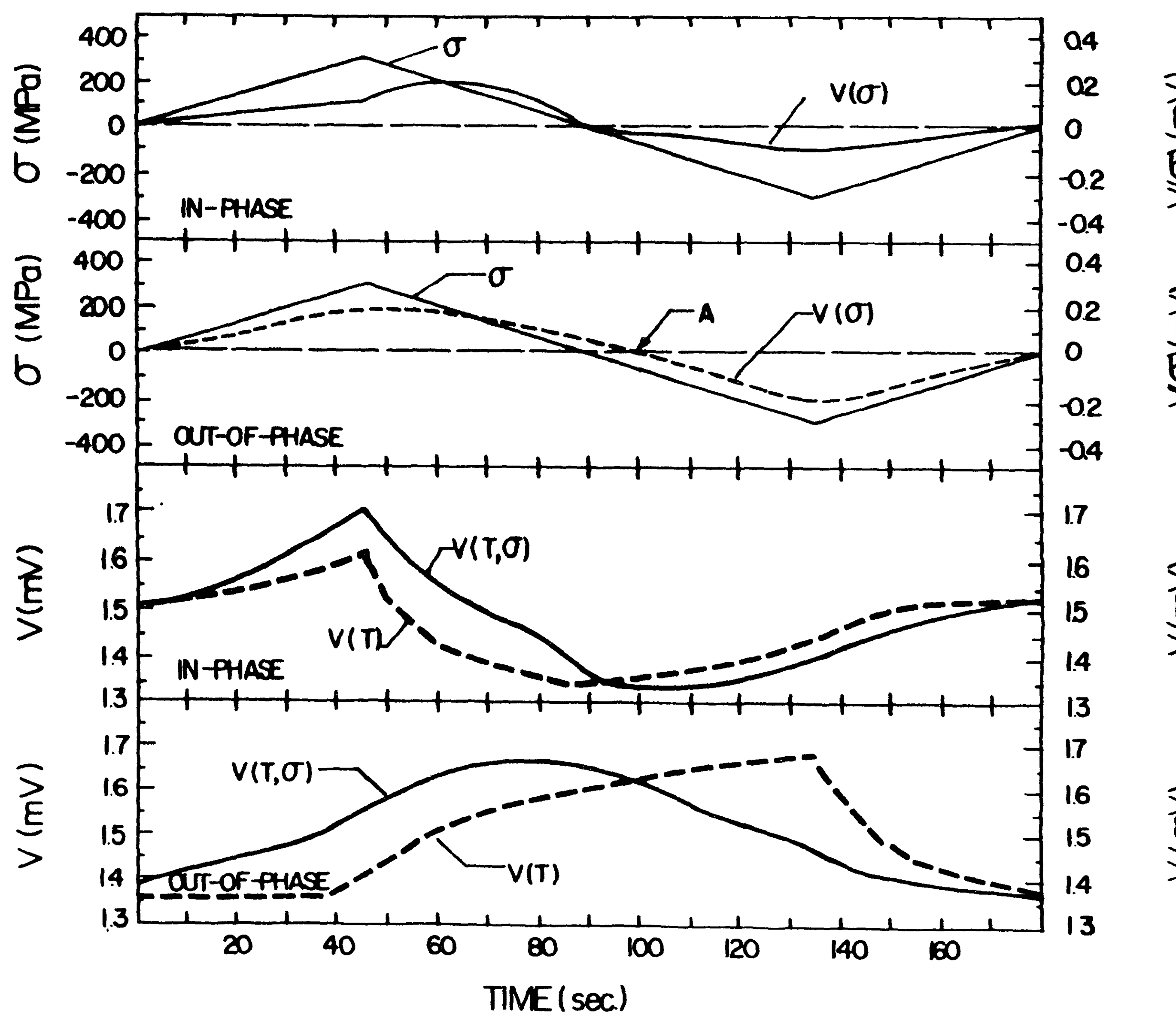


Figure 6.5 Potential curves $V(T,\sigma)$, $V(T)$, and $V(\sigma)$ for TMFCG in Inconel X-750. $\Delta K = 50\ MPa\sqrt{m}$.

potential curves show (at the same ΔK) a smooth increase with σ_{net} with a peak value at σ_{max}. The potential then decreases down to a minimum and finally increases again with σ_{net}. The most important feature of these signals is the crossover point (denoted A) for which $V(\sigma) = 0$. The crossover point represents the stress to apply to the specimen for the potential $V(\sigma,T)$ to equal $V(T)$. Because $V(T)$ is measured at zero applied stress ($\sigma_{net} = 0$) for the entire thermal cycle, the expected stress to apply is $\sigma = 0$. However, if the potential field near the crack tip is disturbed either by a non-zero residual stress-strain field or by geometrical events such as blunting, closure, etc., the $V(\sigma,T)$ potential at zero applied stress might not necessarily equal $V(T)$ and a non-zero stress is required to cancel out the contribution of this event (Δ). For out-of-phase potential, the crossover occurs at a negative stress (e.g. $\sigma = -75$ MPa at $\Delta K = 50$ MPa $\sqrt{m}$), whereas, it always occurs near zero stress for the in-phase potential. By first assuming that the crossover point occurs near an effective closure stress (σ_{cl}), it follows that the effective stress intensity factor (ΔK_{eff}) for crack growth will be higher for out-of-phase than for in-phase cycling (σ_{cl} is negative). If we plot the crack growth rates as a function of ΔK_{eff} ($\Delta\sigma_{eff} = \sigma_{max} - \sigma_{closure}$), we find that both in-phase and out-of-phase crack growth rates overlap as shown in Figure 6.6. This clearly shows that the stress intensity factor is a suitable parameter for correlating TMFCG data, provided the effective stress range is used.

It is important to note that the absolute amplitude of the $V(\sigma)$ signal (see Figure 6.5) in the compressive regime of the cycle, is much higher for out-of-phase than for in-phase cycling which indicates that the crack surfaces are in contact on a much larger scale than under in-

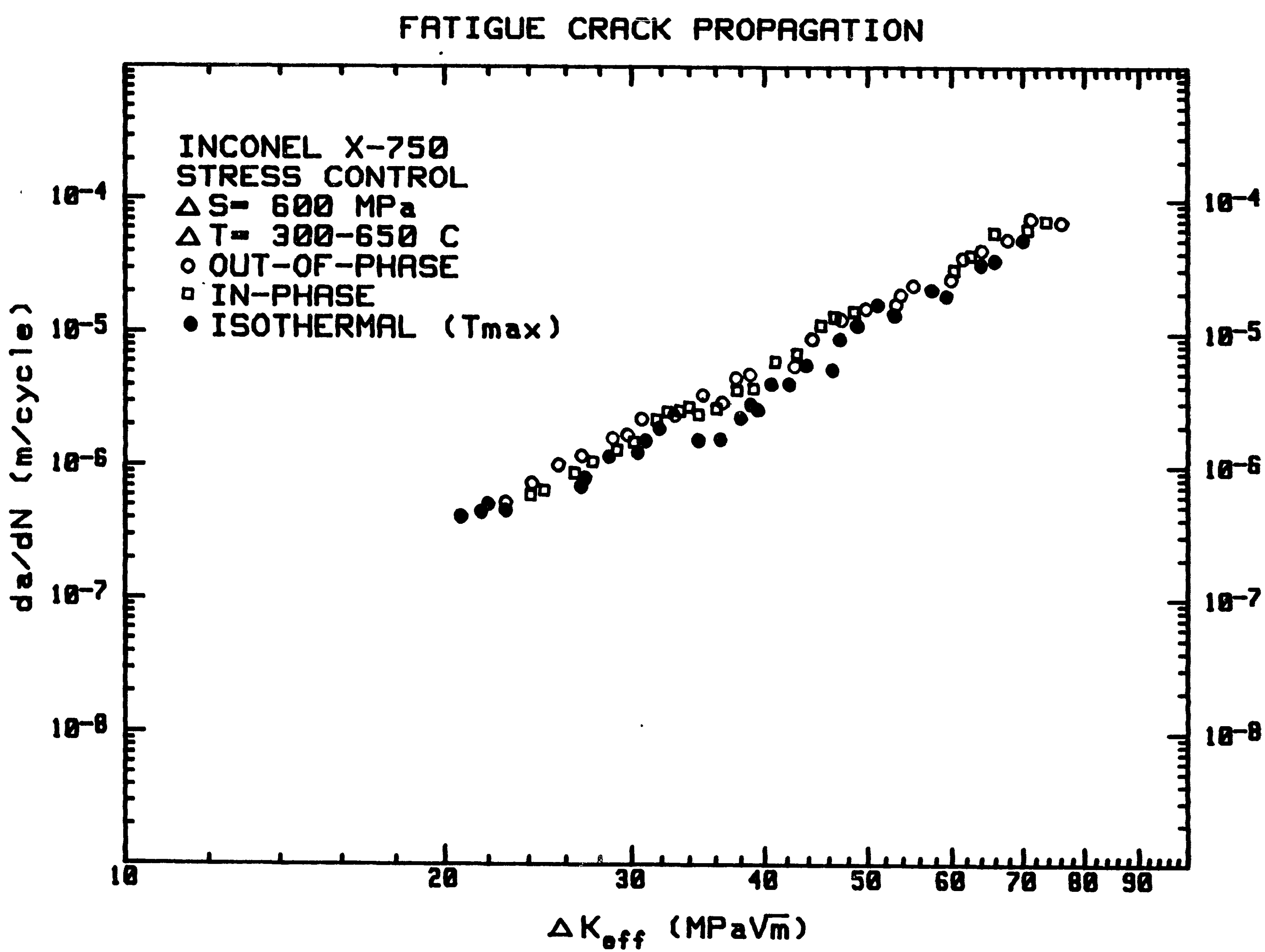


Figure 6.6 TMFCG rates in Inconel X-750 as a function of ΔK_{eff}.

phase cycling. This is in agreement with the fractographic observation which shows that out-of-phase cycling leads to transgranular crack with considerable mating, whereas, in-phase cycling leads to intergranular fracture with little evidence of mating. The higher crack growth rates at R = -1 than R = 0.05 and the fractographic observations lead to the conclusion that although there should be no cracking at the maximum temperature because of the compressive stress, some form of damage is taking place under compressive strain. Also, the surface oxide film found at high temperature will rupture at low temperature, under maximum tensile stress. This leads to a resharpening of the crack tip with each cycle and results in an increase in growth rate.

The rationale for the negative closure stress in out-of-phase cycling can be found by assuming that the residual stress field at the crack tip is of tensile nature at zero applied stress and that a compressive stress has to be applied in order to cancel it and to close the crack. Assuming that the CSS behavior of Inconel X-750 is similar (in trend) to B-1900+Hf, then the validity of the previous argument is supported by Figure 5.26, which shows that for out-of-phase cycling the mean stress σ increases with increasing strain ranges. At low ΔK, the strain range near the crack tip are small [144] and the mean stress are small for all cases (see Figure 5.26). As ΔK increases, the strain range increases and so does σ. However, the increase of σ with $\Delta \varepsilon_{mec}/2$ is much faster for out-of-phase than for in-phase or isothermal cycling which remain small (with respect to the stress range) for all strain amplitudes. In other words, the closure stress is becoming more and more compressive as ΔK increases, which is consistent with our measurements of σ_{cl}.

The difference in growth rates between the isothermal test (650°C) and the TMF tests can also be explained by looking at the $V(\sigma)$ potential curves at low and high ΔK. The $V(\sigma)$ curves were normalized by subtracting $(V_{max} - V_{min})/2$ from the $V(\sigma, T = cst)$ signal. At low ΔK no significant differences were observed between $V(\sigma)$ potential curves of isothermal and TMF tests. At high ΔK, however, the isothermal $V(\sigma)$ potential curves show a crossover point taking place at a tensile stress. This implies that the effective driving force for cracking is reduced with respect to in-phase and out-of-phase cycling. By taking into account this σ_{cl} in the computation of ΔK, one finds that the crack growth rates in terms of ΔK_{eff} for both isothermal and TMF tests are similar as shown in Figure 6.6. The positive closure stress observed is attributed to the buildup of oxides in the wake of the crack. This is supported by fractographic observation which have shown greater oxidation for isothermal testing than for TMF cycling.

6.4 Fatigue Crack Growth (FCG) of B-1900+Hf

The applicability of fracture mechanics to correlate TMFCG data under controlled elastic displacements is confirmed by comparing our data with those of Pratt & Whitney Aircraft (P&WA) obtained on tubular specimens with an EDM slot in the center of the gage section. Because the reported data [29] are plotted in terms of the conventional ΔK_{ε} given by

$$\Delta K_{\varepsilon} = \Delta\varepsilon . \sqrt{\pi a} . G(a/W) \tag{6.2}$$

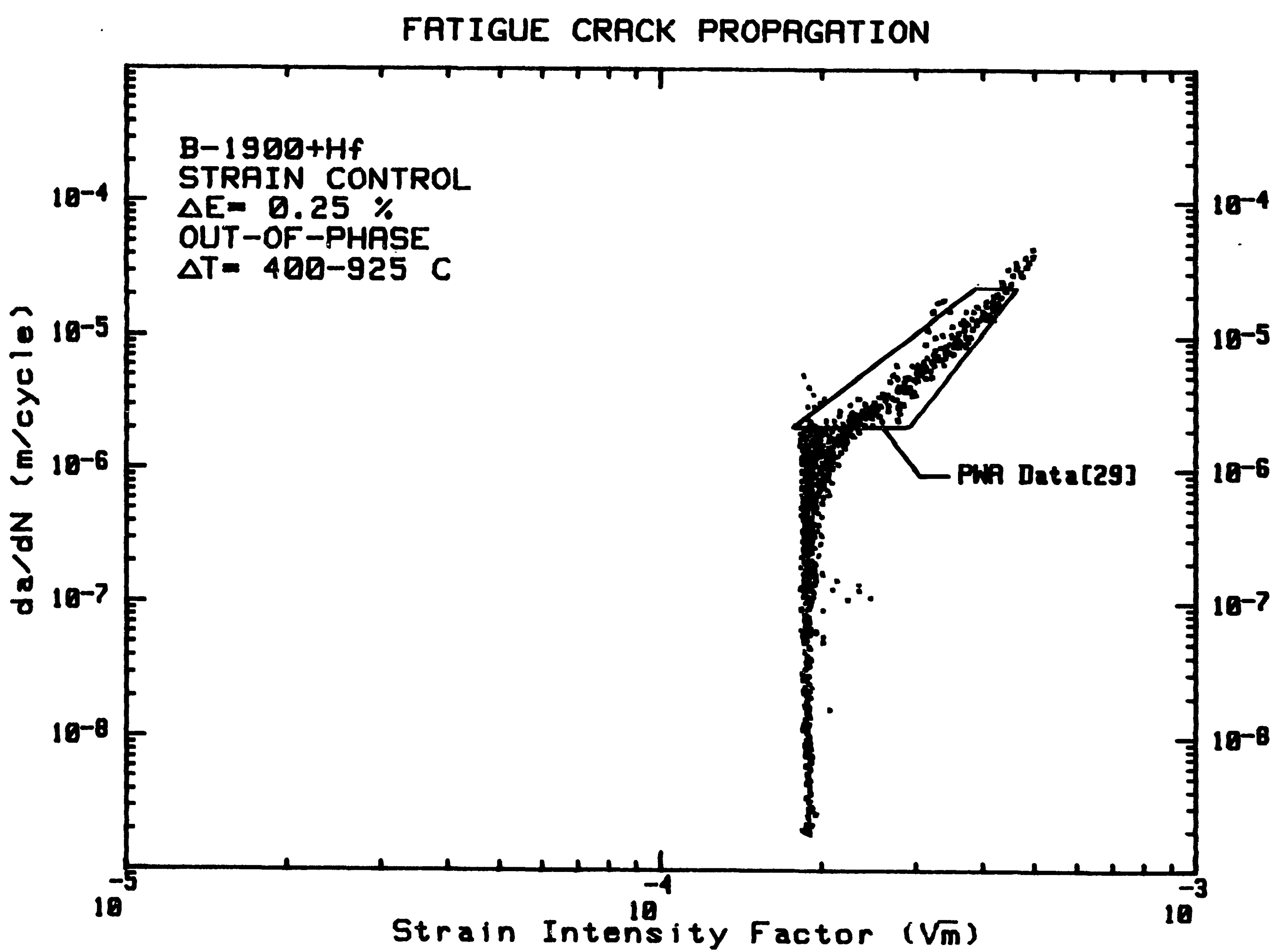


Figure 6.7 Out-of-phase crack growth rates in B-1900+Hf as a function of the conventional strain intensity factor for SEN and tubular specimens[29].

where G(a/W) is the correction factor derived for stress-controlled (free rotation) stress intensity factor, the following formula was used to compute the ΔK_{ε}'s

$$\Delta K_{\varepsilon} = \Delta\varepsilon . \sqrt{\pi a} \quad \{25/20-13(a/W) - 7(a/W)^2\}^{1/2} \tag{6.3}$$

Figure 6.7 shows the results for $\Delta\varepsilon_{mec}$ = 0.25%. As can be seen, the agreement between the two geometries is quite good which suggests that fracture mechanics is applicable provided the driving force is properly computed. Although Eq. 6.3 does not provide realistic quantitative values for the ΔK_{ε}'s (see Appendix V), it is used here to show that for a given material and testing conditions, equal driving forces calculated on different geometries yield identical crack growth rates. Figure 6.7 also shows that the computerized testing system has the capability of measuring crack growth rates data three orders of magnitude lower than those measured by more conventional TMF testing systems.

The results in Figure 5.31 show that TMF cycling is more damaging than isothermal cycling when the strain range in the bulk is fully elastic. This result is expected since for Inconel X-750 cycled under elastic stresses, the same behavior was observed (see Figure 5.29). As we know, the maximum stress (σ_{max}) achieved during cycling ranks (in increasing order) as isothermal, in-phase, and out-of-phase. Similarly, the crack growth rates also ranked in this order (see Figure 5.31). It is now logical to assume that the driving force for cracking is a stress-based parameter. In other words, even under strain-controlled cycling, the driving force for cracking is not the applied strain range as much as the maximum stress which results from the imposed cyclic straining. This point has been discussed in Section 6.2 where the modes of cracking in B-1900+Hf indicate that the fracture criterion under TMF cycling is stress-based. Further support to this conclusion was given by the study of the effect of frequency (see Figure 5.33) which shows that as the frequency increases (at fixed strain range and temperature) the crack growth rate increases. The rationale for this behavior is that as the frequency increases, the strain rate increases and so does the stress range. If indeed the fracture criterion is stress-based rather than strain-based, an increase in stress range results in an increase in crack growth rates (in terms of the ΔK_ε's).

Based on this idea and on the fact that the stress intensity factor was a successful parameter for Inconel X-750, all the crack growth rates data were replotted in terms of ΔK_σ, where ΔK_σ is given by Eq. 5.1. The measured loads (N_{max}) are used in the ΔK_σ calculations.

Figure 6.8 shows that effect of frequency disappears if the crack growth rates are plotted in term of ΔK_σ. The confirms the assumption that fracture in B-1900+Hf is stress-based rather than strain-based. Figures 6.9 and 6.10 show a summary of the results for $\Delta \varepsilon_{mec}$ = 0.25% and $\Delta \varepsilon_{mec}$ = 0.50%. The detail results are given in Appendix IV. As expected, an increase in maximum stress by changing the testing conditions results in shifting the fatigue crack growth curves to the right, i.e., higher values of ΔK_σ. The shift of the FCP curves causes an increase in the apparent spread of the data between the different testing conditions. Although the spread suggests that K_σ is not the proper correlation parameter, comparison of Figures 6.9 and 6.10 shows that ΔK_σ does provide a rationale for the effect of strain range. In other words, the considerable strain range effect which exists when the crack growth data are plotted in term of ΔK_ε (see Figures 5.31 and 5.32)

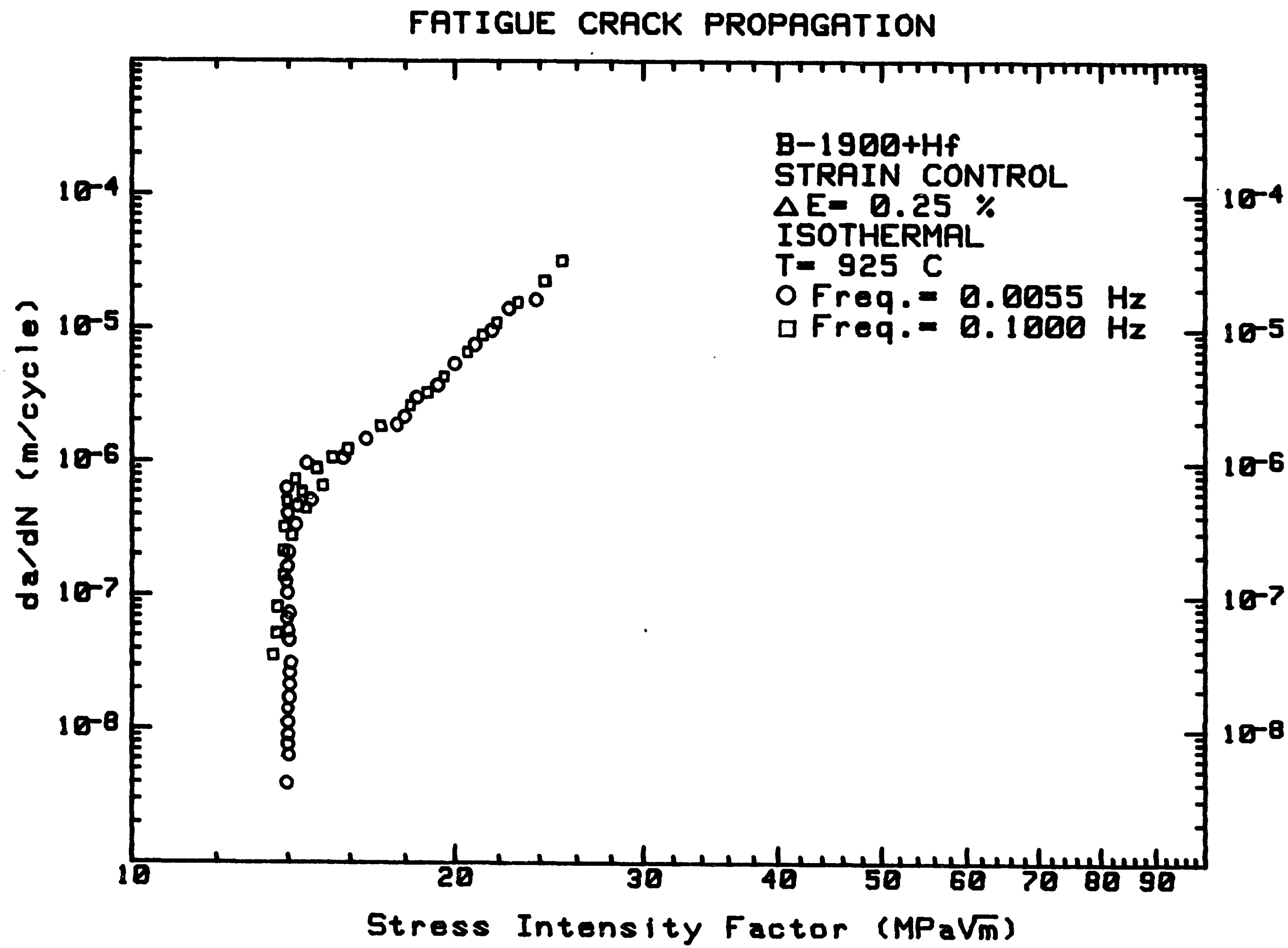


Figure 6.8 Effect of frequency on the FCG rates in B-1900+Hf as a function of ΔK_σ (T_{max}, $\Delta\varepsilon_{mec}$ = 0.25 %).

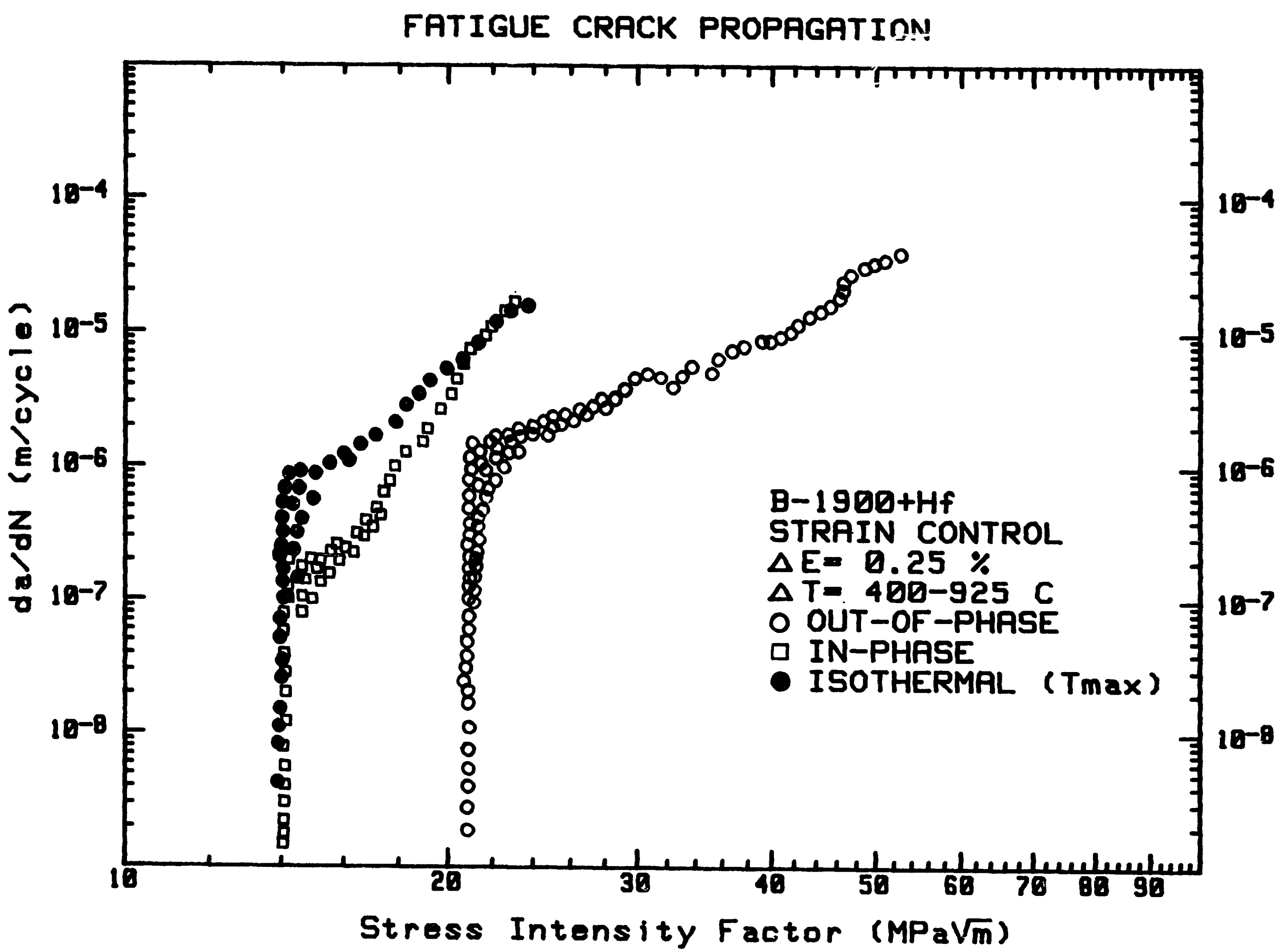


Figure 6.9 FCG rates in B-1900+Hf as a function of ΔK_σ ($\Delta\epsilon_{mec} = 0.25$ %)

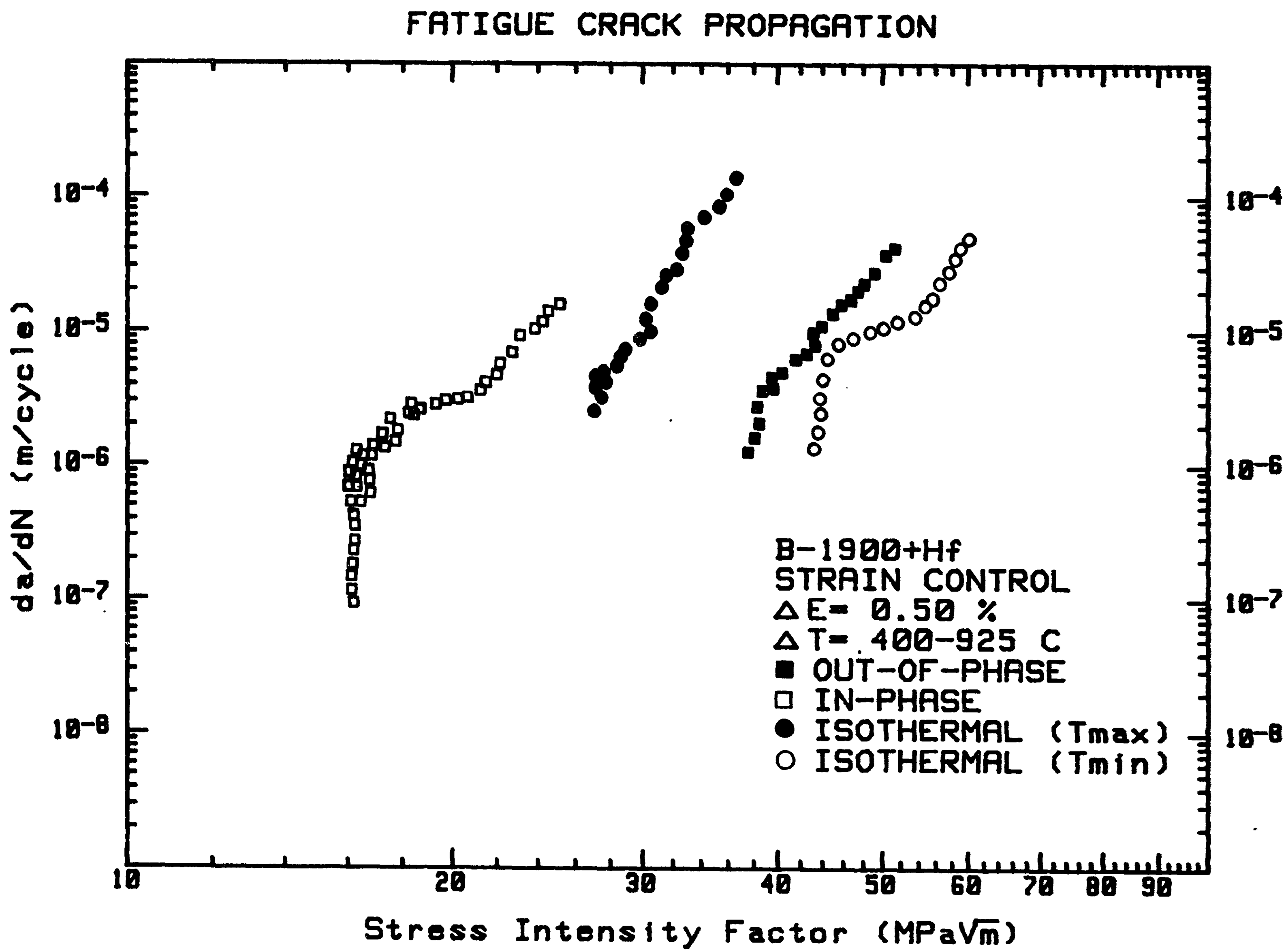


Figure 6.10 FCG rates in B-1900+Hf as a function of $\Delta K_{\sigma}(\Delta\varepsilon_{mec} = 0.50\ \%)$

disappears if the ΔK_σ is used to correlate the data. Figure 6.11 shows the correlation of the data for out-of-phase cycling at 0.25 and 0.50%. The ability of ΔK_σ to better rationalize the effect of strain range as compared to ΔK_ε stems from the fact that doubling the strain range does not necessarily result in doubling the stress range. Instead, an increase in $\Delta\varepsilon$ results in an increase in crack growth rates which is proportional to an increase in stress range. The use of Eq. 5.1 which takes into account any non-proportional increase in stress range caused by hardening or softening at the crack tip and in the bulk material, is therefore more appropriate than the use of Eq. 5.3 which assumes a proportional increase in $\Delta\sigma$ with an increase in $\Delta\varepsilon_{mec}$. As long as the plastic strain range remains small, Eq. 5.1 provides a fairly good estimate of the stress intensity factor in the material and thus a good correlation of the crack growth data with increasing strain range is expected. However, if the plastic strain range is significant (for a given $\Delta\varepsilon_{mec}$), the correction for bending use in Eq. 5.1 will be incorrect because the bending (M) and normal (N) components in the specimen are coupled and cannot be determined separately (see Appendix V). This explains why the ΔK_σ cannot collapse the crack growth data for isothermal testing to the same extent than it does for in-phase and out-of-phase cycling because under isothermal conditions (T_{max}) and $\Delta\varepsilon_{mec}$ = 0.50%, the plastic strain range is approximately 0.08% whereas it is less than 0.02% for in-phase and out-of-phase cycling.

As for Inconel X-750, the difference between in-phase, out-of-phase and isothermal crack growth data was found in the analysis of the potential drop signal. Figures 6.12, 6.13 and 6.14 show the corrected potential signal $V(\sigma)$ at two different crack lengths (a/W ~

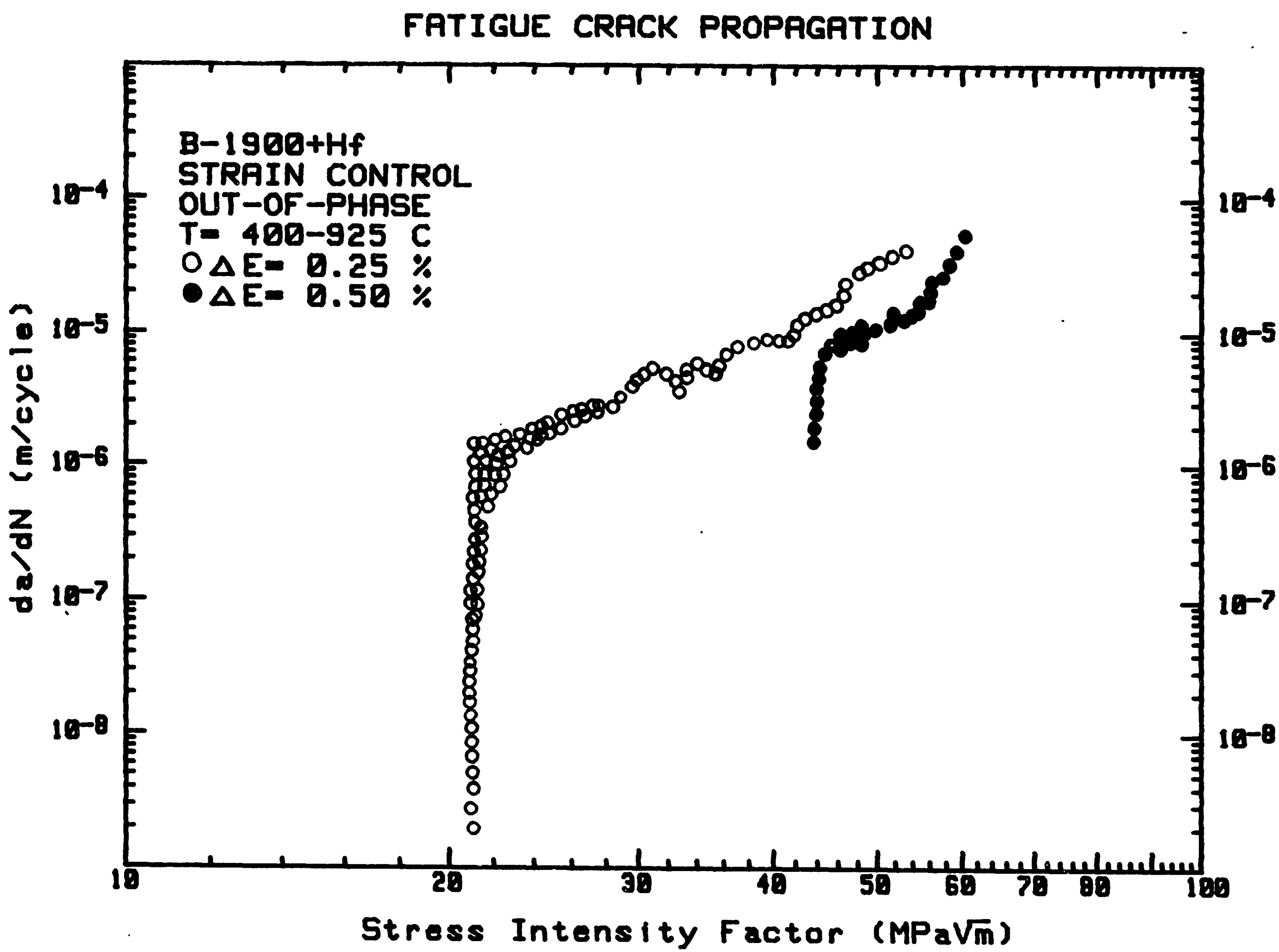


Figure 6.11 Effect of strain range on the FCG rates in B-1900+Hf cycled under out-of-phase conditions

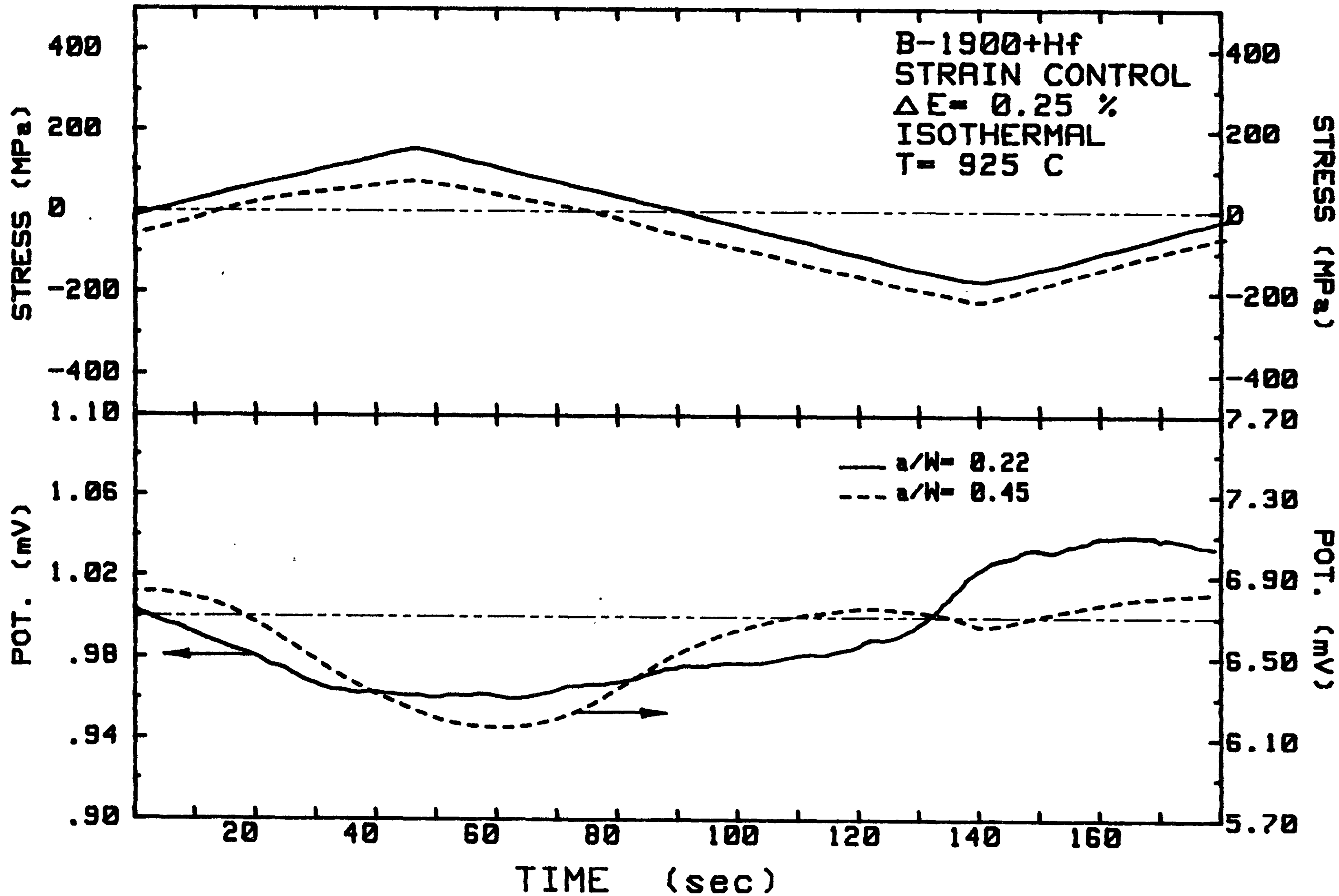


Figure 6.12 Potential curve V(σ) and stress amplitude in B-1900+Hf under isothermal cycling(T_{max}).

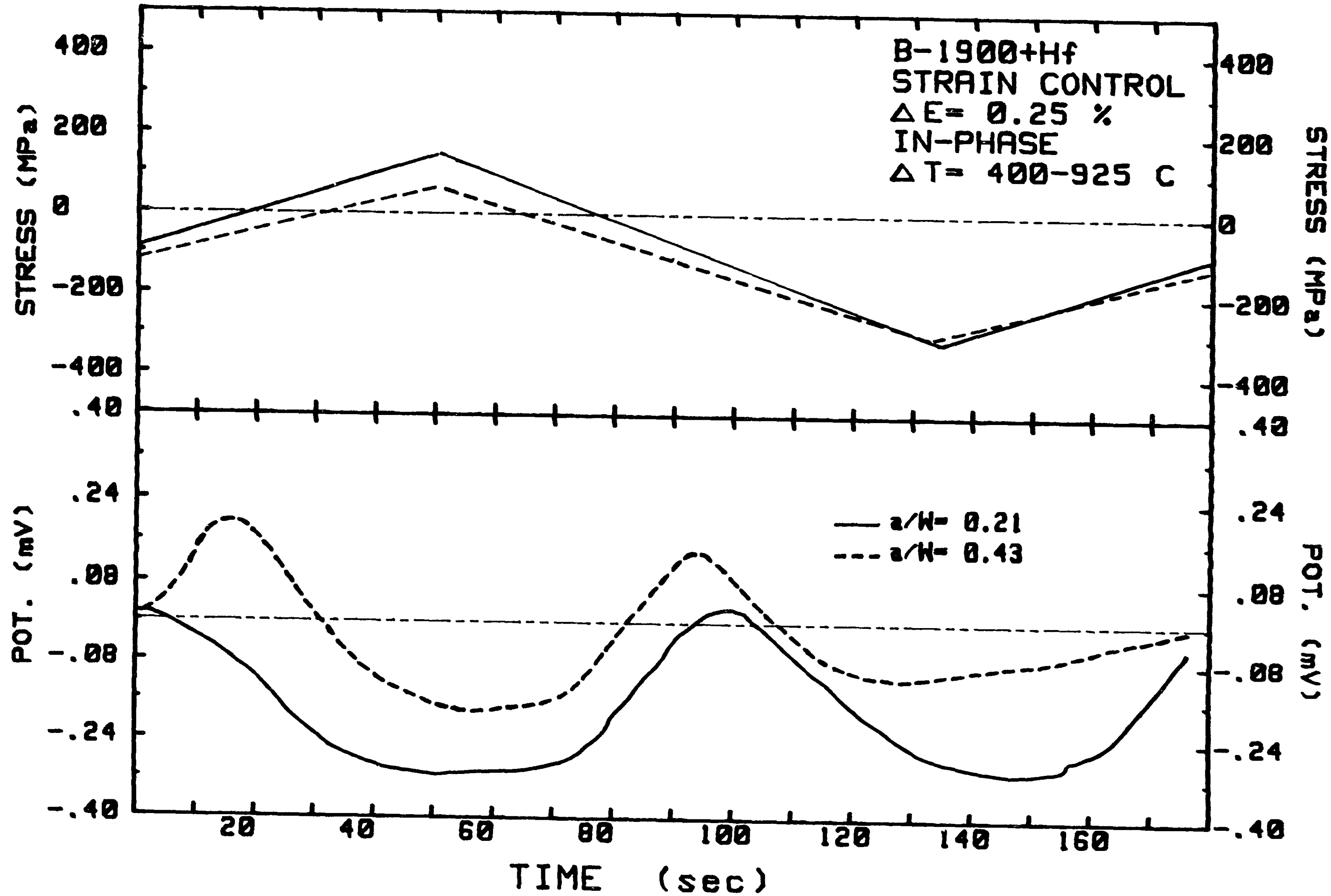


Figure 6.13 Potential curve V(σ) and stress amplitude in B-1900+Hf under in-phase cycling.

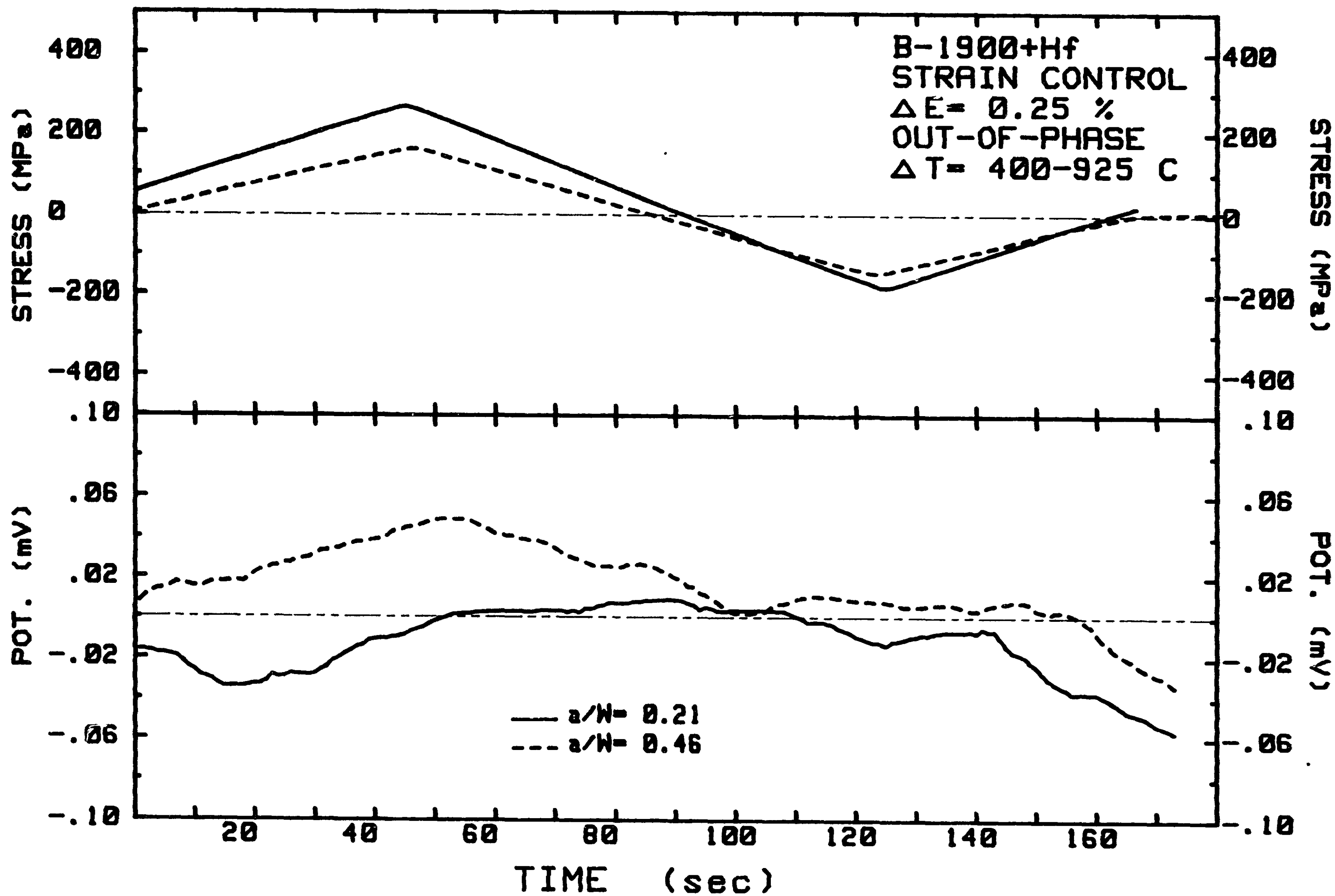


Figure 6.14 Potential curve V(σ) and stress amplitude in B-1900+Hf under out-of-phase cycling.

0.21 and 0.45) for isothermal, in-phase, and out-of-phase testings. The following conclusions can be drawn from these figures. First, the potential signal $V(\sigma)$) are more complex than for Inconel X-750 (see Figures 6.4 and 6.5). This is expected since in these tests the stresses are not imposed but rather depend on changes in geometry (crack length) and material properties (hardening/softening) and this introduces another variable in the $V(\sigma)$ signals. Another observation is that the opening stress (σ_{op}), which is the stress at which the crack is fully open, is different from the closure stress (σ_{cl}), the stress at which the crack is completely closed. Furthermore, the difference between σ_{op} and σ_{cl} increases as the crack length increases. Finally, the magnitude of σ_{op} varies with crack length being tensile for $a/W < 0.40$ and compressive for large crack lengths ($a/W > 0.45$). The influence of crack length on σ_{op} will be discussed later.

The crack growth rates were then plotted as a function of the effective stress intensity factor (ΔK_{eff}). In the calculation of ΔK_{eff} only the opening stress (σ_{op}) were used. Figure 6.15 shows the crack growth rates as a function of ΔK_{eff}. As can be seen, the ΔK_{eff} provides a reasonable correlation for the crack growth rates and all the data fall within one of the two master curves. All the test data for which σ_{max} occurs at T_{max} (isothermal and in-phase cycling) fall in one scatterband, whereas all the low temperature data for which σ_{max} occurs at T_{min} (isothermal and out-of-phase cycling) fall in the other scatterband. In terms of mode of fracture, all the tests for which fracture is transgranular (σ_{max} at T_{min}) are grouped in one scatterband of slope 2.8, whereas all the tests which failed intergranularly (σ_{max} at T_{max}) clustered in a scatterband of slope 5.0. This clearly indicates that the

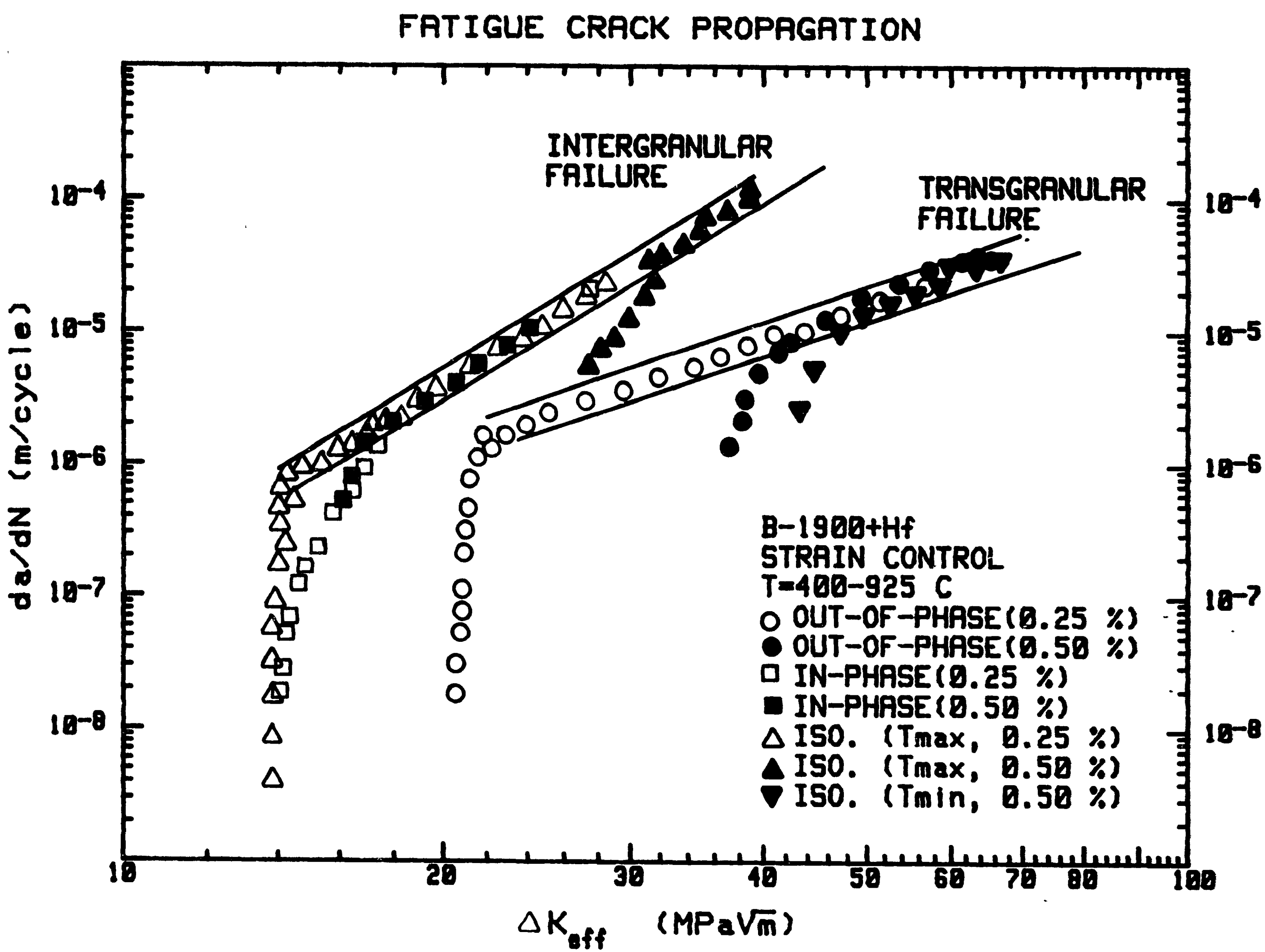


Figure 6.15 FCG rates in B-1900+Hf as a function of ΔK_{eff}

mode of fracture in B-1900+Hf is stress-based and depends only on the temperature at which σ_{max} occurs. In other words, for a given mode of fracture (fixed by the temperature at σ_{max}) there is a unique relationship between the crack growth rates and ΔK_{eff}. Furthermore, this relationship is independent of the applied strain range and cyclic history of the material.

However, the knowledge of ΔK_{eff} alone is not sufficient to allow prediction of the crack growth rates. The relationship between the strain and the temperature must be known such that the probable mode of fracture can be determined and the proper master curve (da/dN vs ΔK_{eff}) be used.

The last issue to be addressed is the variation of σ_{op} with crack length. Figure 6.16 shows the variation of σ_{op} as a function of a/W for an isothermal test (T_{max}, $\Delta \varepsilon_{mec}$ = 0.25%). Also shown in this figure is the function G(a/W) here defined as

$$G(a/W) = L \cdot \Delta K_{\varepsilon} / \Delta \delta \qquad (6.4)$$

with ΔK_{ε} given by Eq. 5.3, L the gauge length of the specimen, and $\Delta \delta$ the imposed displacement (see Appendix V). As can be seen in Figure 6.16, the σ_{op} is tensile when the normal component (N) dominates and ΔK_{ε} increases. On the other hand, when the bending component is significant, i.e. when ΔK_{ε} decreases with increasing crack length, the σ_{op} is compressive. It is interesting to note for this testing condition (the only one for which Eqs. 6.4 and 5.3 strictly apply), that the inflection point in the σ_{op} vs a/W results occurs at a value of a/W which yield a maximum in the G(a/W) vs a/W curve. Furthermore, the fact that σ_{op} monotonically decreases between $0.35 < a/W < 0.65$ whereas K shows a monotone increase followed by a decreased, clearly

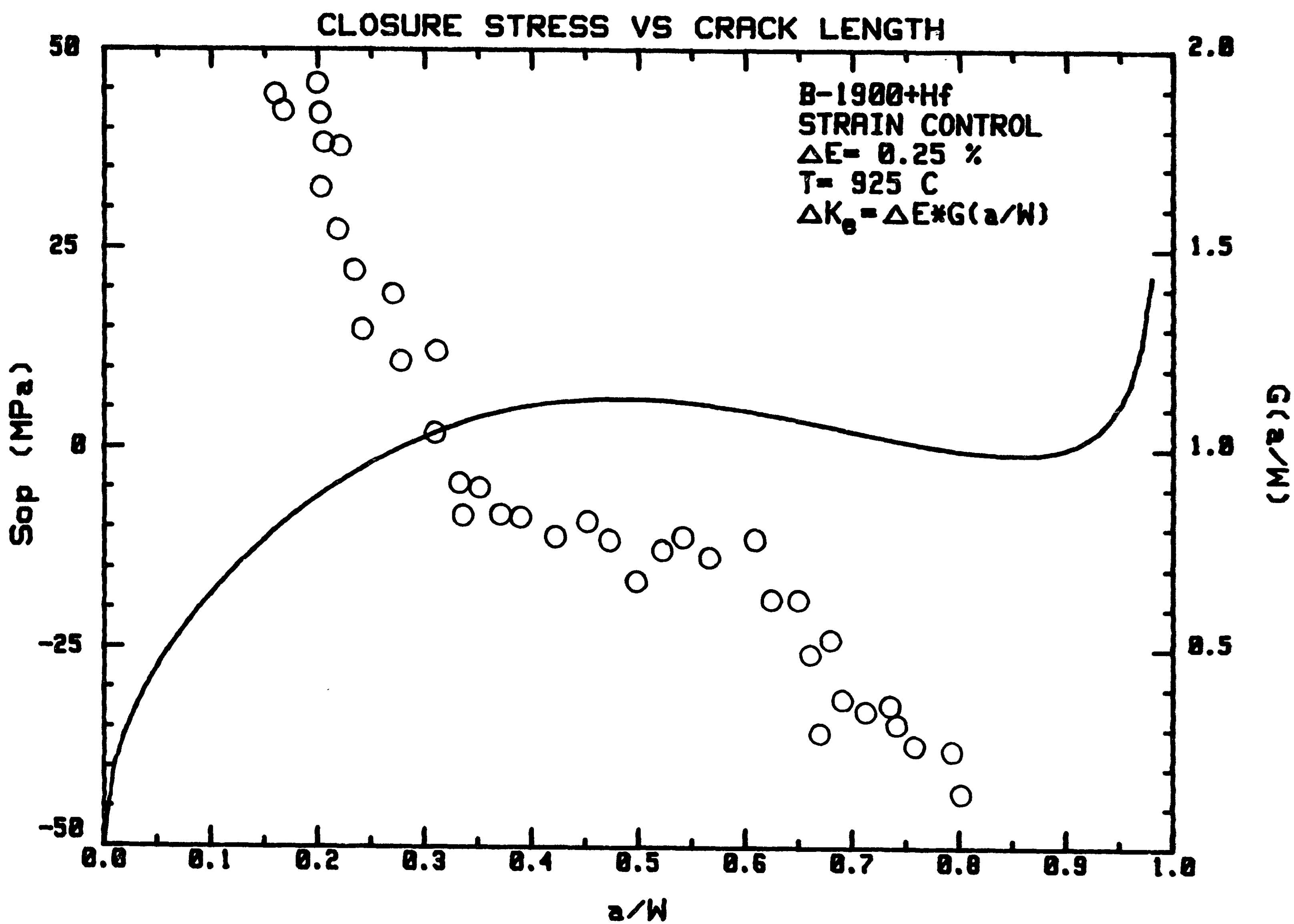


Figure 6.16 Variation of the opening stress(σ_{op}) as a function of crack length in B-1900+Hf(T= 925 °C, $\Delta\varepsilon_{mec}$ = 0.25 %).

indicates that σ_{op} is not controlled by K_{max} as sometimes observed in stress-controlled testing [147-148]. The results shown in Figure 6.16 support the argument that σ_{op} is a function of the dominant stress component (tensile or bending) in SEN specimens tested under strain controlled conditions.

An explanation for a decreasing σ_{op} with increasing crack length can be found in the work of Tanaka et al. [149-150] and Murakani et al. [151-152] which have shown (for a given crack length) that σ_{op} decreases when the plastic strain ranges increases. Furthermore, they showed that, under large cyclic plastic strains, cracks remain fully open even at the minimum stress (σ_{min}). For a center-cracked panel (CCP) specimen under remote uniform displacement (i.e., ε_0), Murakani [153] has obtained, using finite element analysis, an equation describing the singularity in strain distribution at the tip of a crack of length "a", i.e.,

$$\varepsilon_y = C(n) \cdot \varepsilon_0 \cdot (a/r)^{\alpha(n)} , \qquad (6.5)$$

where C(n) is a function which depends on the strain hardening exponent n, ε_0 the imposed remote far-field strain, and $\alpha(n)$ the signularity term which depends on the strain hardening exponent. For n = 5, the singularity term α is 0.52 and for n = 13, α is 0.64 [153]. Although not exact, Eq. 6.5 provides a rationale for the decrease in σ_{op} with increasing crack length. As "a" increases (at fixed ε_0), the plastic strain range ε_y at the crack increases and σ_{op} decreases, in agreement with the observation of Tanaka et al. [149-152]. A decrease in σ_{op} with increasing a/W has recently been reported in a CCP specimen of low carbon steel cycled under fully reversed stress conditions [149]. However, the decrease in σ_{op} with a/W is likely to be larger in SEN specimens than in CCP specimens, because the plastic strain range at the crack tip will be larger (for a given ε_o). In the case of SEN specimens, ε_y not only depends on ε_o and n as indicated by Eq. 6.5, but also will include a component which is a function of the bending moment in the specimen. In conclusion, the considerable drop of σ_{op} with increasing crack length, which leads to an increase in ΔK_{eff}, explains why the crack growth rates increase even if the material experiences a significant drop in the actual load (N_{max}) which tends to reduce ΔK_{eff}.

6.5 Fatigue Crack Growth (FCG) in Hastelloy-X

The applicability of fracture mechanics to correlate TMFCG data under displacement controlled conditions in Hastelloy-X is checked by comparing our data with those of Pratt &Whitney Aircraft (PWA) obtained on tubular specimens with an EDM slot in the center of the gage section [41]. As for B-1900+Hf, the conventional definition of the ΔK_ε was used (i.e., Eq. 6.3) for purpose of comparison. Figure 6.17 shows the results for out-of-phase cycling between 400-925°C at $\Delta\varepsilon_{mec}$= 0.25%. The results of PWA (i.e., the scatterband) for out-of-phase cycling between 426-871°C and 426-927°C at a frequency of 2 min. per cycle are shown. The following comments are made concerning the data presented in Figure 6.17..

First, the 426-871°C tests data show the fastest crack growth rates of all out-of-phase tests run at a frequency of 2 min/cycle. This does not agree with the intuitive result that the higher the peak temperature in the TMF cycle, the faster should be the crack growth rate.

Secondly, the 426-871°C tests data at 2 min/cycle are in good agreement with the 400-925°C tests data at 3 min/cycle. The

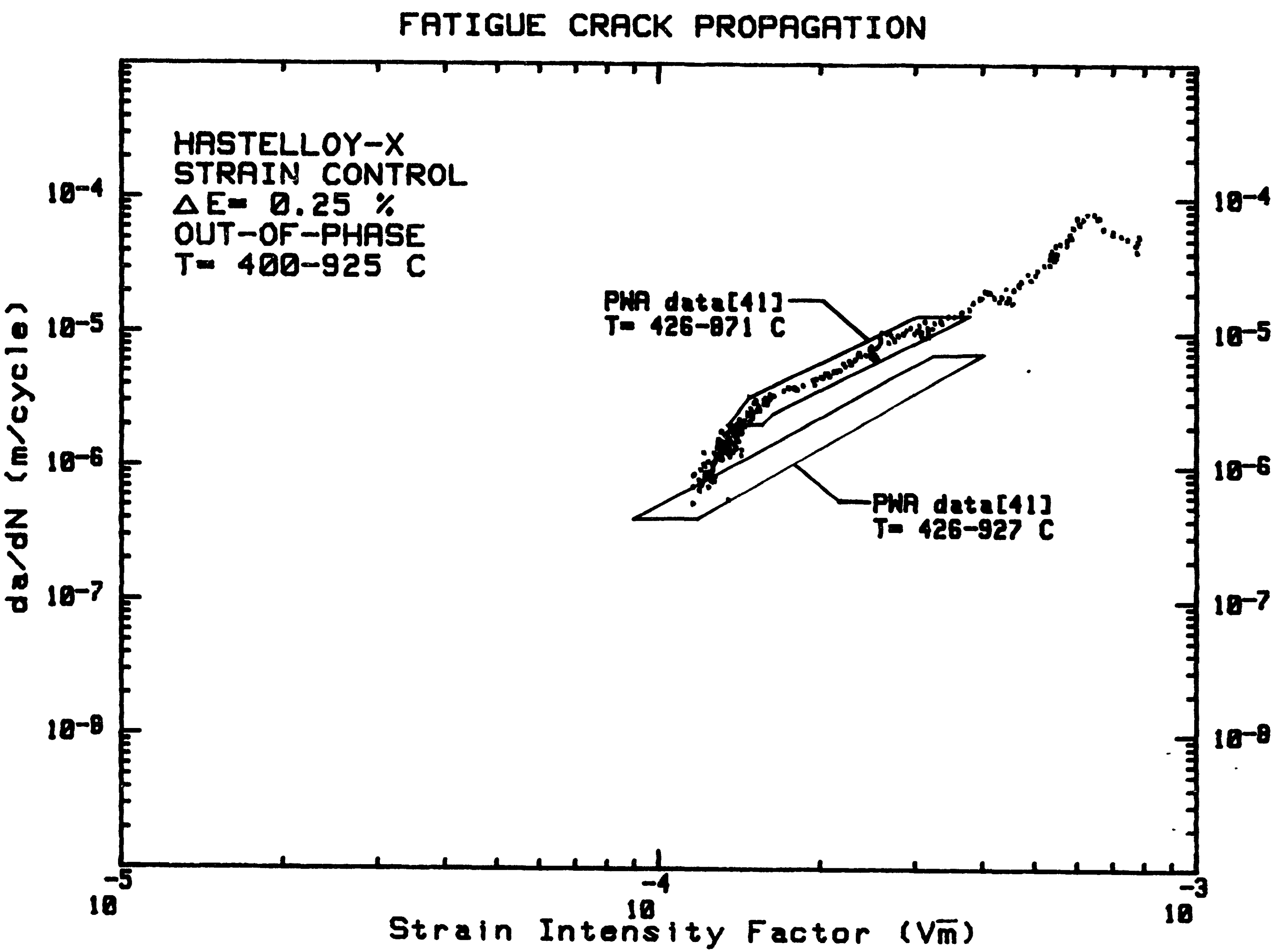


Figure 6.17 Out-of-phase crack growth rates in Hastelloy-X as a function the conventional strain intensity factor in SEN and tubular specimens[41].

explanation for these observations can be attributed to extensive ageing of the γ-matrix which takes place in the vicinity of 815-871°C [154-155]. At temperatures lower than 815°C the kinetic of $M_{23}C_6$ and Laves phase precipitation is slow because diffusion is slow. At temperatures above 875°C, the solubility of the γ-matrix is high enough and the driving force for precipitation (i.e., supersaturated concentration) is reduced. In other words, the main controlling parameter for ageing of the γ-matrix in fatigue will be the time spent in the temperature range of 815-871°C within each cycle. It is interesting to note that the time spent per cycle in the temperature range of 815-871°C is 15 sec. for the 426-871°C test, 12 sec. for the 426-927°C test, and 19 sec. for the 400-925°C test (3 min/cycle). Because the propeties of the γ-matrix controls the crack growth behavior, the results shown in Figure 6.17 are not surprising. In fact, based on the previous argument, one would expect faster crack growth rates for the 400-925°C test (3 min/cycle) than for the 426-871°C test 92 min/cycle) because the time spent per cycle in the range of 815-871°C is larger. Nevertheless, the agreement between the two geometries is quite good which suggests that fracture mechanics is applicable for TMF of ductile materials like Hastelloy-X.

Figure 5.34 shows that the TMFCG rates under elastic cycling ($\Delta \varepsilon_{mec}$ = 0.25%) are faster than those obtained under isothermal cycling at T_{max}. However, the TMFCG rates are lower than their isothermal counterpart under fully plastic cycling ($\Delta \varepsilon_{mec}$ = 0.25%) which is consistent with the crack growth behavior of B-1900+Hf (see section 5.4.3). Comparison of the crack growth rates as a function of ΔK_{ε} also shows a considerable strain range effect which clearly indicates that

ΔK_{ε} is not the proper driving force for TMF crack growth (see section 5.4.4).

As for Inconel X-750 and B-1900+Hf, the crack growth rates data were plotted in terms of ΔK_{σ} using Eq. 5.1. Figures 6.18 and 6.19 show the results for $\Delta \varepsilon_{mec}$ = 0.25% and $\Delta \varepsilon_{mec}$ = 0.50%. As can be seen in Figures 6.18 and 6.19, the ΔK_{σ} leads to an apparent increase in scatter between the various tests data. However, a close look at the results reveals, as for B-1900+Hf, that the stress intensity factor separates the tests data which failed intergranularly, and the tests which failed transgranularly in two distinct groups. Furthermore, comparison of Figure 6.18 and 6.19 shows that ΔK_{σ} provides a very good correlation of the crack growth rates with increasing strain range. This is unexpected since for the same applied strain range, the plastic strain range in Hastelloy-X are significantly larger than for B-1900+Hf. Therefore, the ability of ΔK_{σ} to correlate the crack growth data in Hastelloy-X should be limited as compared to B-1900+Hf.

The explanation for this dichotomy can be found in the work of Murakami [153] which have shown that the strain singularity term α at the crack tip (see Eq. 6.5) is a function of the strain hardening exponent (n) and remote strain (ε_0). More specifically for n = 5, α was shown to increase with increasing ε_0 up to a maximum value of 0.81 (the HRR singularity is 0.83) for $\varepsilon_0 \simeq 0.2\%$ and to further decrease to values about 0.5 under large plastic strain ($\varepsilon_0 > 2.0\%$). With ΔK_{σ}, the singularity of the strain at the crack tip is 0.5. At large plastic strain the singularity term is also close to 0.5 [153]. If, for a given mode of fracture, the response of a material to given local stress-strain distribution (at the crack tip) is unique, the observed agreement between the elastic and

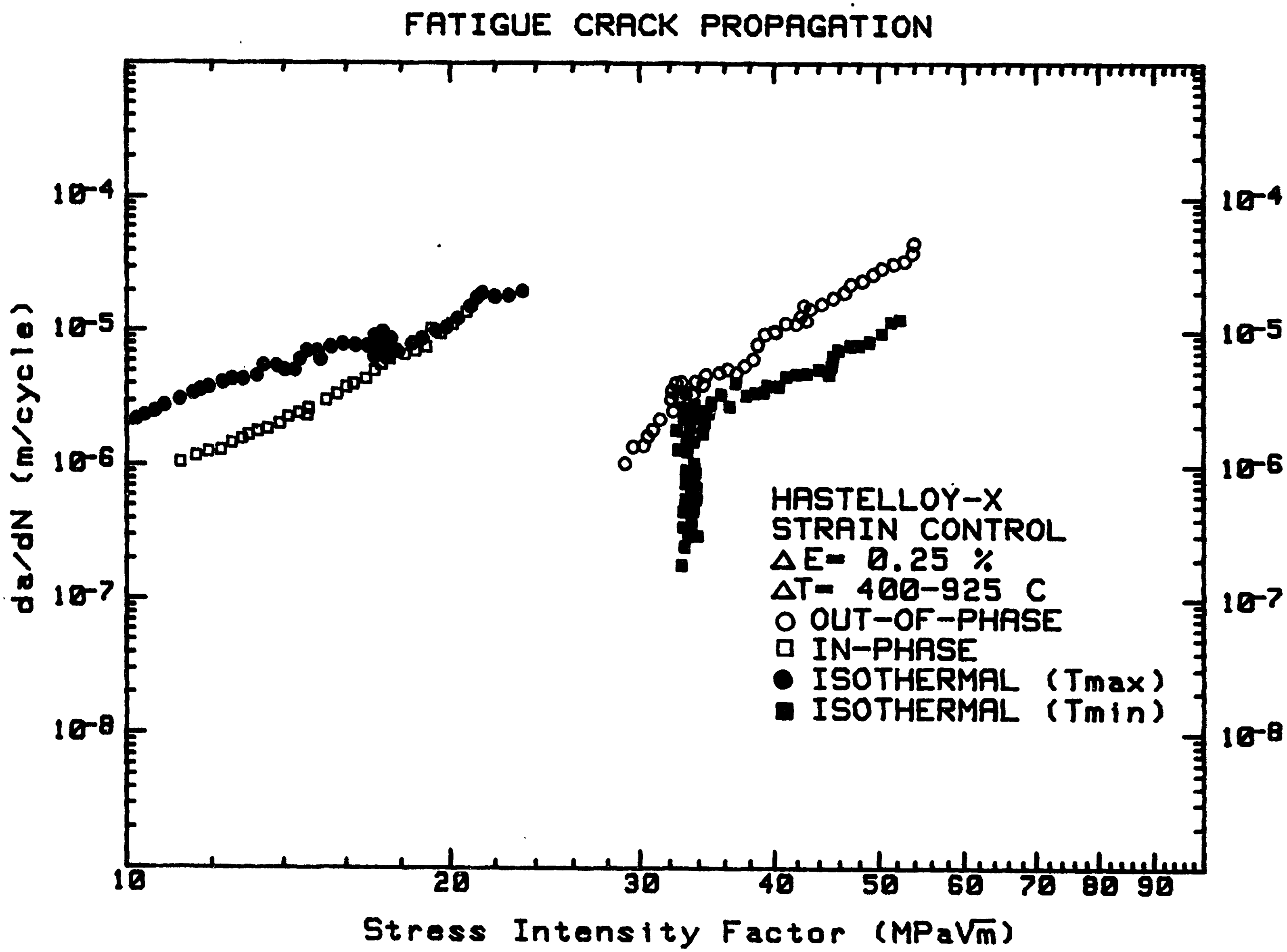


Figure 6.18 FCG rates in Hastelloy-X as a function of ΔK_{σ} ($\Delta\varepsilon_{mec}$ = 0.25 %)

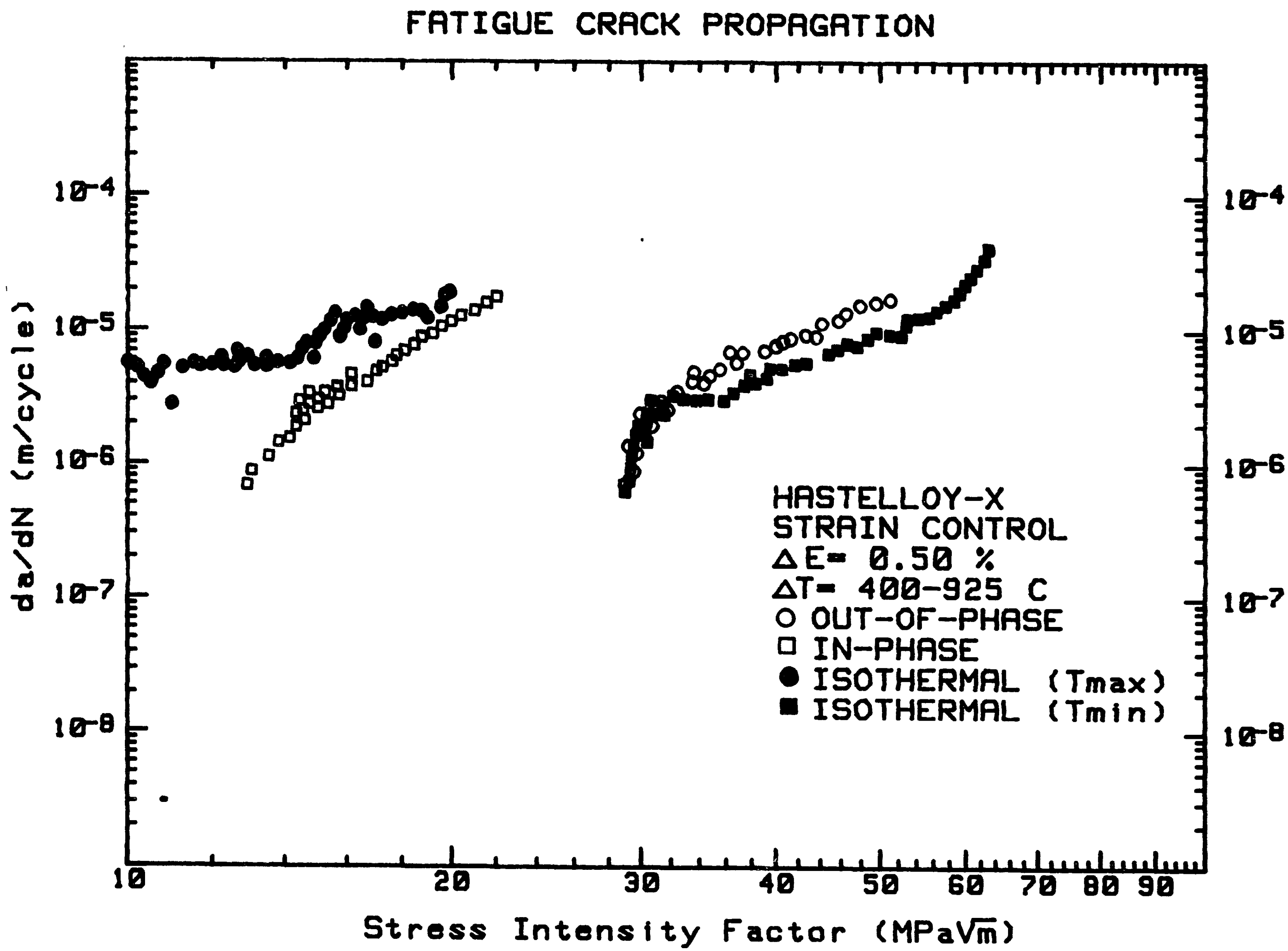


Figure 6.19 FCG rates in Hastelloy-X as a function of ΔK_σ ($\Delta\varepsilon_{mec} = 0.50$ %)

plastic strain cycling data is not surprising. For small plastic strains ΔK_σ provides a poor correlation of the data because the singularity at the crack tip is 0.83 (for n = 5) and the material response to this new stress-strain distribution will be different.

As for B-1900+Hf, an attempt to rationalize the effect of the various testing conditions on the crack growth rates was undertaken by using the measured values of the opening stress (σ_{op}) derived from the analysis of the potential drop signals. Figure 6.20, 6.21 and 6.22 shows the potential $V(\sigma)$ and the stress response of Hastelloy-X under isothermal, in-phase and out-of-phase cycling. A plot of σ_{op} as a function of crack length for $\Delta \varepsilon_{mec}$ = 0.25% and T = 925°C (see Figure 6.23) shows the same trend than in B-1900+Hf (see Figure 6.16). This suggests that the parameters which controlled σ_{op} are similar in both alloys. Using the value of σ_{op}, the crack growth rates as a function of ΔK_{eff} can be plotted. The result is shown in Figure 6.24. As can be seen, except for in-phase cycling, the ΔK_{eff} provides a very good correlation of the data. Furthermore, the correlation holds independently of the applied strain range and cyclic history. The implication of this result is the assumption that, for a given mode of fracture the response of the material to a particular stress-strain distribution at the crack tip is unique, is verified. On the other hand, the mode of fracture is controlled by the tensile strain rate ($\dot{\varepsilon}^+$) at the crack tip. If the $\dot{\varepsilon}^+$ which depends on the frequency, $\Delta \varepsilon$ and a/W, is below the critical value ($\dot{\varepsilon}^+_{crit}$) requires to cause intergranular fracture at temperature T, intergranular cracking occurs. The value of $\dot{\varepsilon}^+_{crit}$ is a function of only the temperature for a given material and environment.

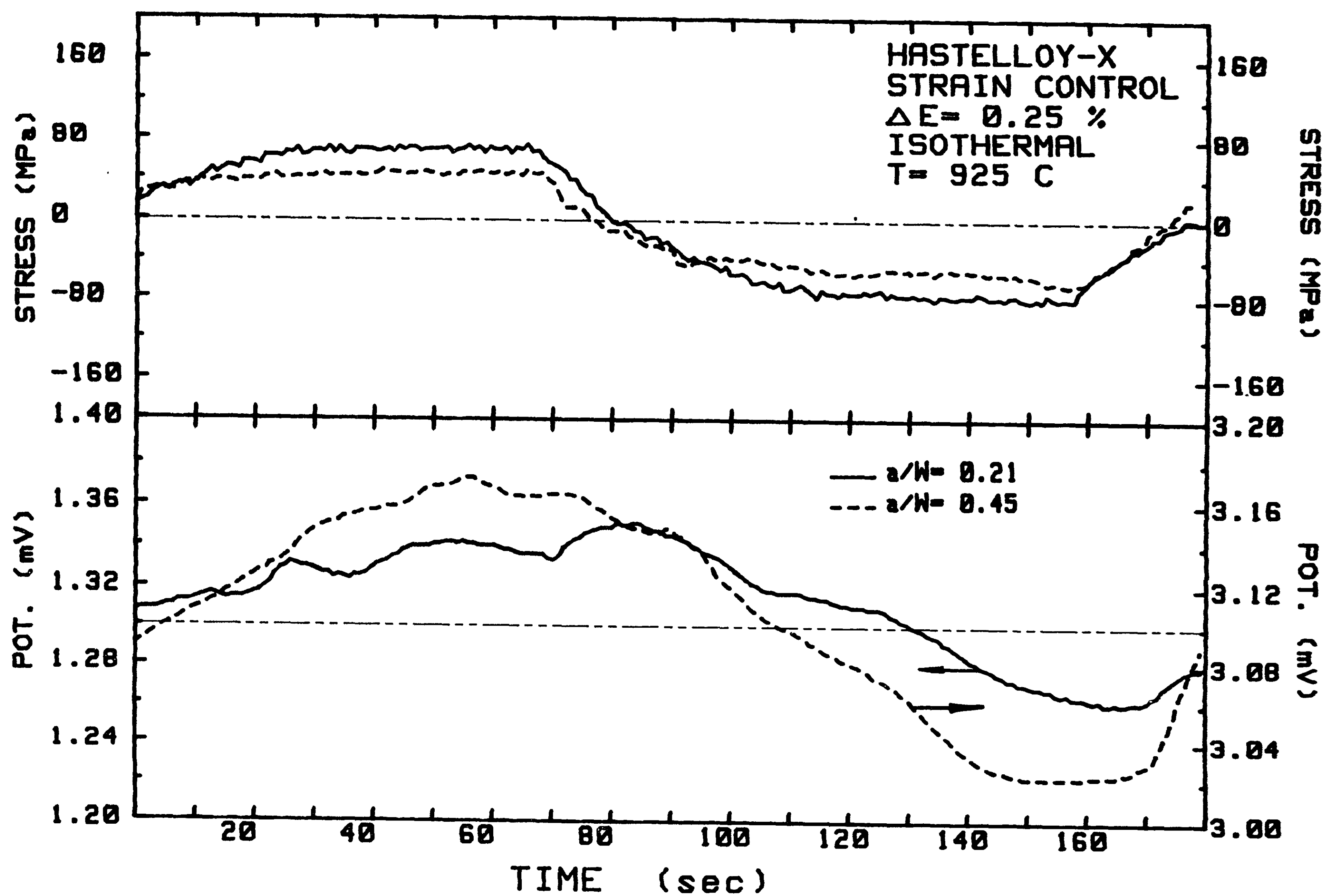


Figure 6.20 Potential curve V(σ) and stress amplitude in Hastelloy-X under isothermal cycling(T_{max}).

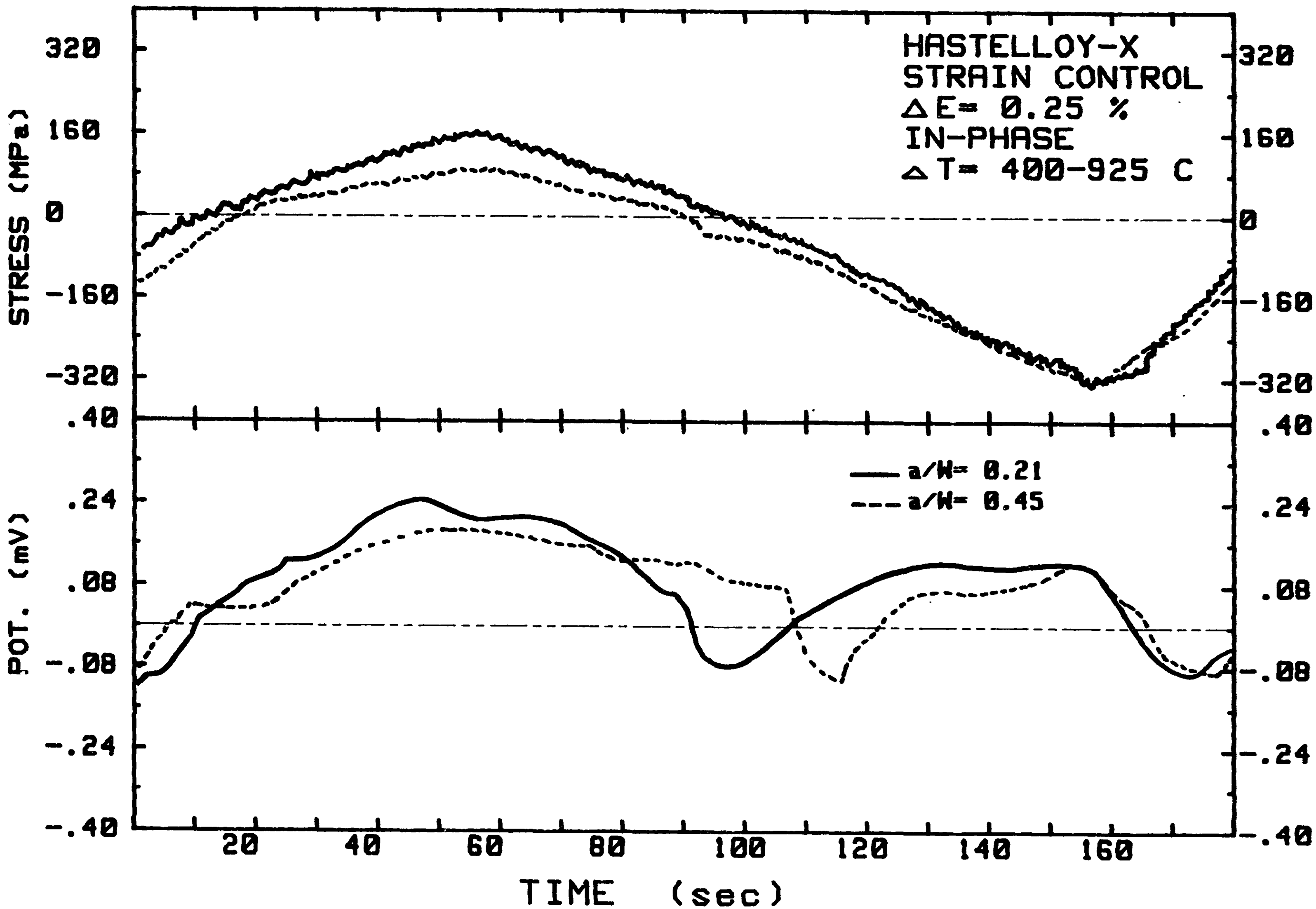


Figure 6.21 Potential curve V(σ) and stress amplitude in Hastelloy-X under in-phase cycling.

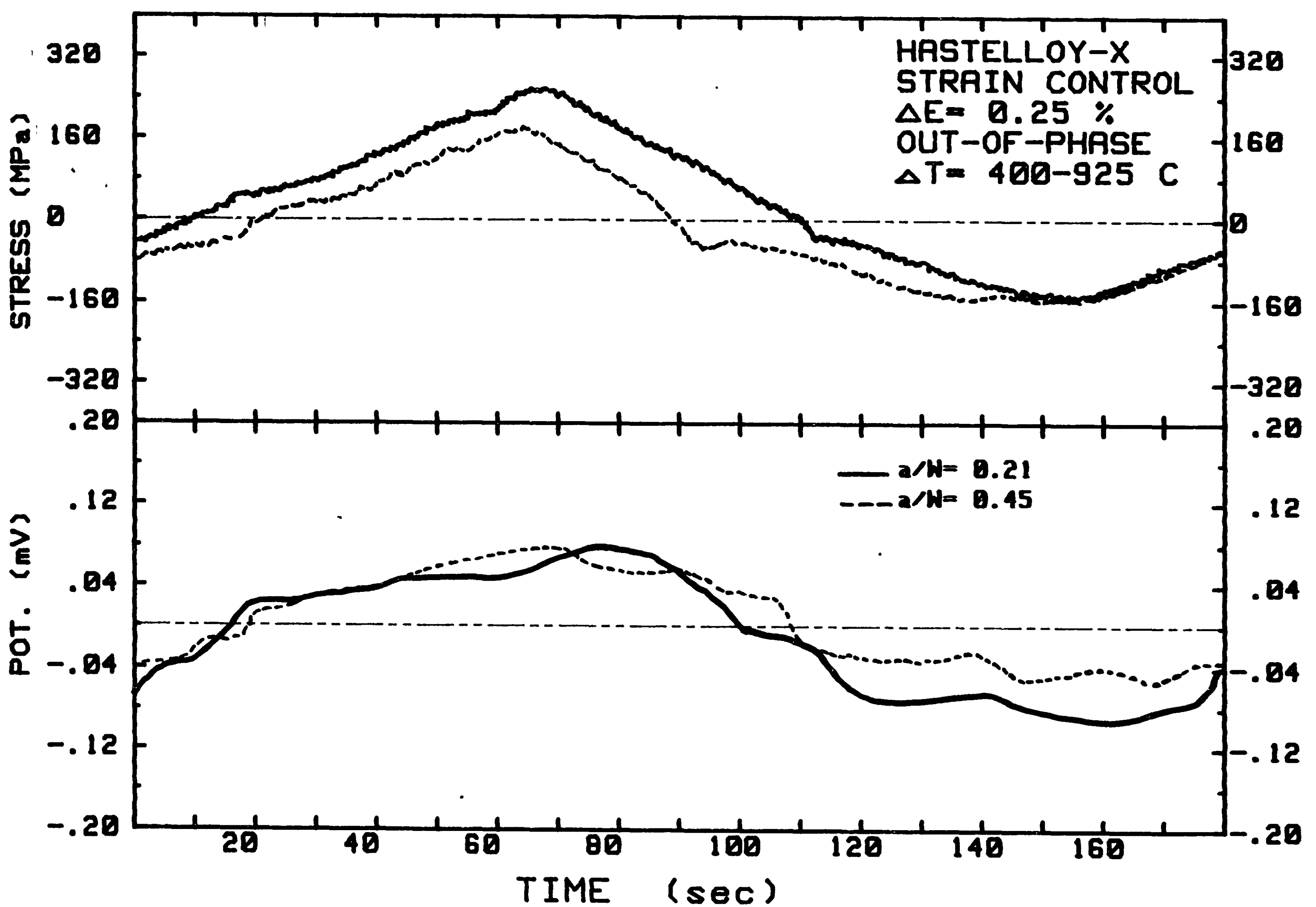


Figure 6.22 Potential curve $V(\sigma)$ and stress amplitude in Hastelloy-X under out-of-phase cycling.

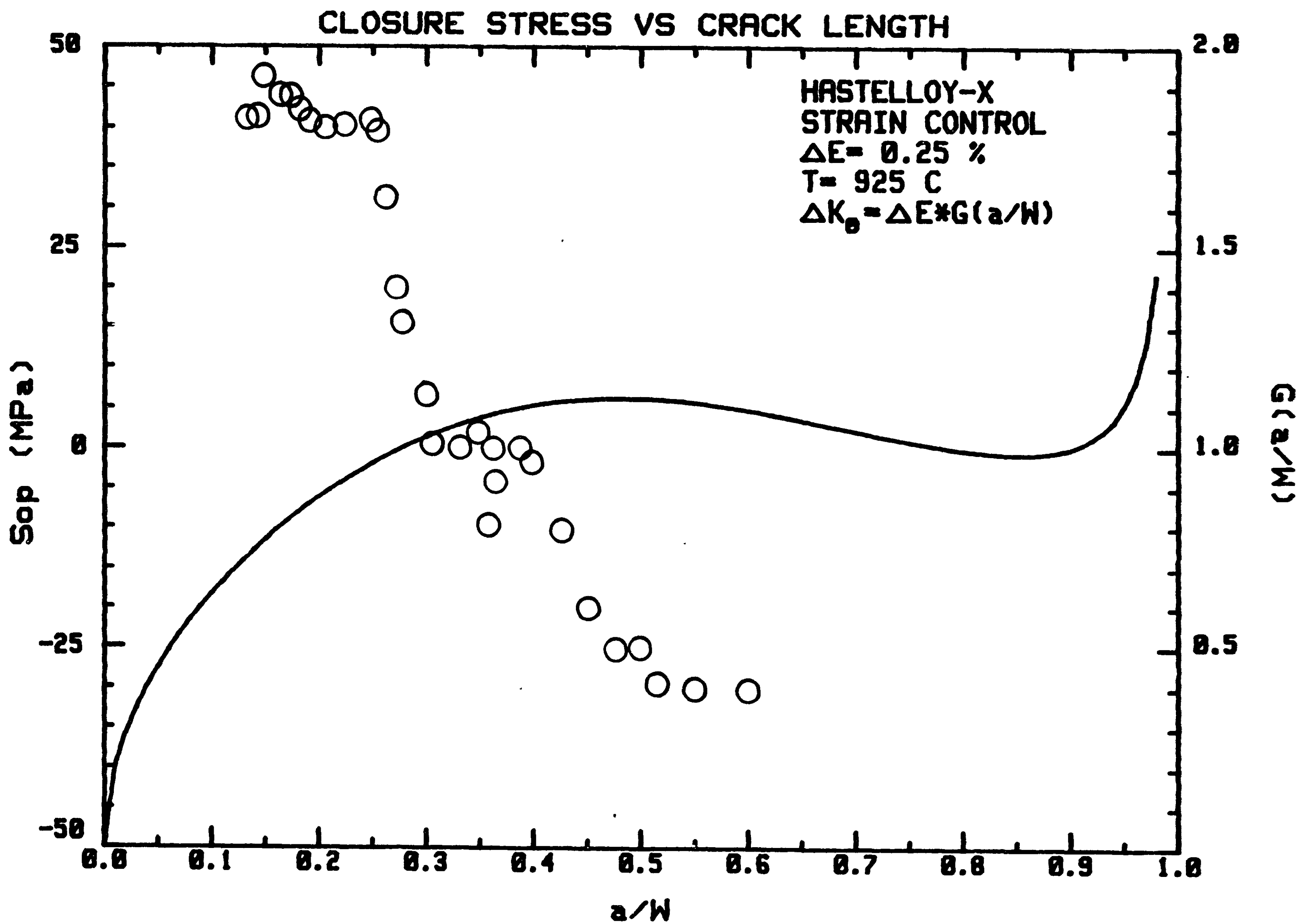


Figure 6.23 Variation of the opening stress(σ_{op}) as a function of crack length in Hastelloy-X(T= 925 °C, $\Delta\varepsilon_{mec}$ = 0.25 %)

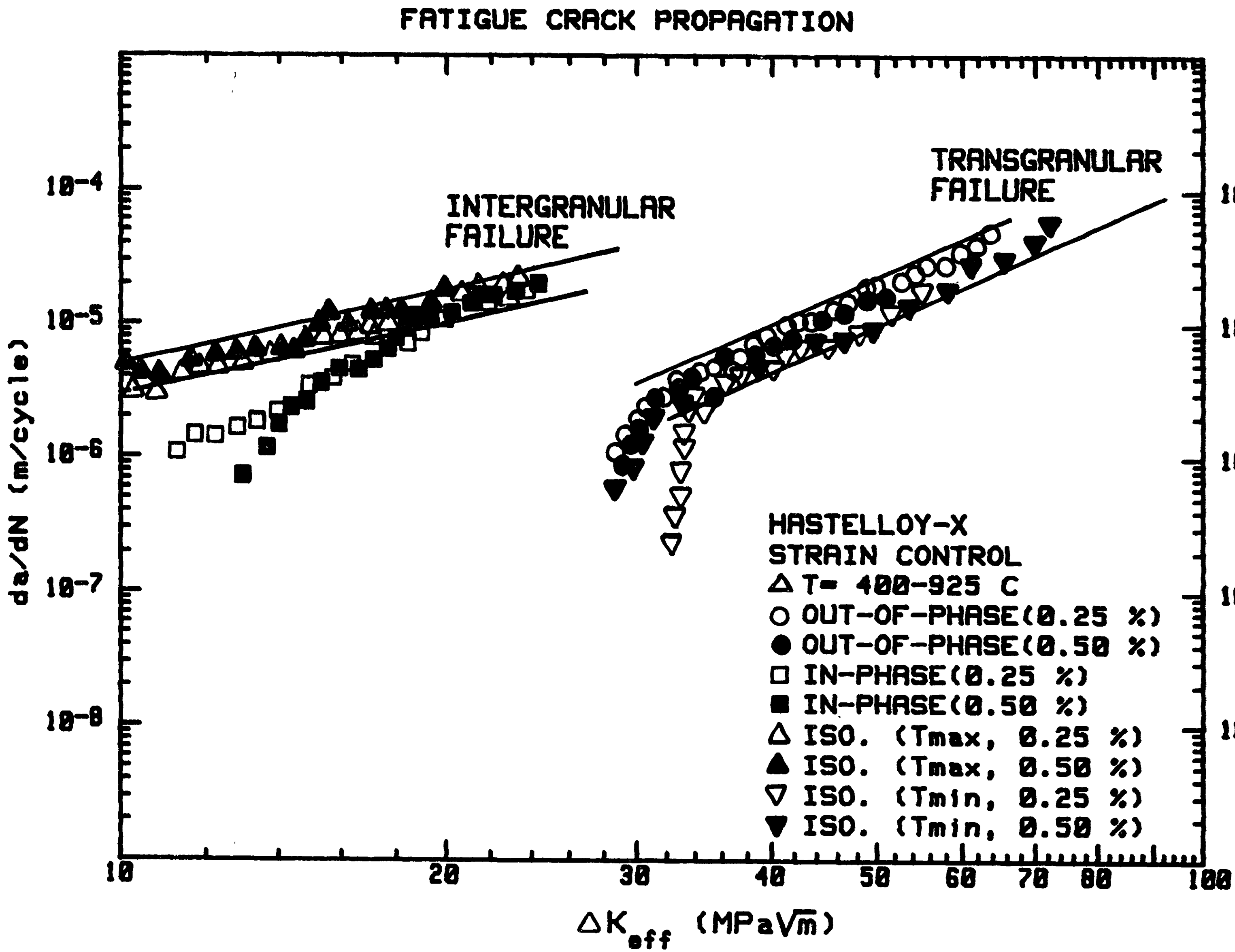


Figure 6.24 FCG rates in Hastelloy-X as a function of ΔK_{eff}

6.6 Correlation Parameters for In-Service Components

The complete description of a thermal-mechanical cycle requires a definition of the relationship between strain and temperature. The prediction of the crack propagation life of an engine component ideally requires the correct prediction of crack growth rates at all locations along the crack growth path. Cracks grow through regimes with a range of different TMF cycles. Solving an in-service component problem requires the ability to predict the crack growth rates for many different strain-temperature paths. If each different TMF cycle requires independent crack growth data, the total data base required for making life predictions would be prohibitively large. Two strategies for dealing with this problem can be used:

1. Methods of predicting TMF crack growth rates from isothermal data,

2. Develop a way of using TMF data for one type of TMF cycle (e.g., out-of-phase) to predict growth rates occurring in more complex strain-temperature (ε-T) cycles (see Figure 1.1).

The first approach is indeed more desirable than strategy 2 because isothermal data are more commonly available and less expensive to generate.

We have shown that ΔK_σ and ΔK_{eff} collapse the data to a great extent which indicates that prediction of complex ε-T cycles from isothermal data is possible. From isothermal testing the da/dN vs ΔK_{eff} curve is obtained, which in turn is used in prediction of crack growth rates of component (assuming that the ΔK_{eff} for the component is known). Although the results in Figures 6.6, 6.10, 6.15 and 6.24 are promising there are still some unaccounted scatter in the results of which suggests that ΔK_{eff} might not be the universal parameter. The differences between the TMF and isothermal data might be the results of one or more of the following factors:

1. The effect of material ageing (coarsening of γ' in the case of B-1900+Hf and Inconel X-750, and carbides precipitation in Hastelloy-X),

2. Temperature change during straining might result in crack growth mechanisms that do not occur isothermally. Damage due to differential expansion of the oxide layer relative to the base metal is one possible mechanism of this type.

3. The success of the correlation parameter may depend on the path and method used for its calculation.

Although our results indicate tha the best parameter for predicting TMF crack growth from isothermal data may be ΔK_{eff}, the application of ΔK_{eff}, in practice, is inconvenient because we must measure or assume the opening ratio of a crack in real components and this measurement is very difficult. Therefore, it is preferable in practice to determine the value of ΔK as a function of crack size and geometry and then estimate the crack growth rates rather than estimating ΔK_{eff} (i.e., σ_{cl}). With this respect, the prediction of complex ε-T cycles from TMF data is an advancement over using isothermal data at the peak temperature. This strategy involves the decomposition of complex ε-T cycles into more simple TMF cycles, and the use of the crack growth data (in terms of ΔK_{σ}) measured under TMF conditions to predict the growth rates of complex ε-T cycles. Models based on the scheme cited above might not give improved crack growth predictions when compared to strategy 1, but they would not require the knowledge of opening ratios in complex components. The ultimate accuracy of these predictions is only limited by the extent to which complex ε-T crack growth is produced by unique damage mechanisms.

7.0 Conclusions

The main achievements of this work can be summarized as follows:

1. A thermal-mechanical fatigue test rig has been built around a conventional servo-hydraulic machine to simulate complex stress-strain-temperature cycles as experienced for in-service components. The system and its capabilities have been described in detail. A corrected DC potential drop technique used to accurately measure crack lengths was shown to yield better capabilities for low crack growth rate measurements (< 10^{-6} m/cycles) than the conventional methods used in thermal-mechanical fatigue crack growth testing.

2. It was shown that solid specimen geometries can be used to simulate in-service thermal-mechanical fatigue crack growth conditions. The SEN geometry has some advantages over the tubular geometry in that it allows deterministic crack length measurements and the use of an accurate fracture mechanics solution.

3. Exact K-solutions for SEN specimens under fixed-end displacements controlled with no-free rotation were derived. It was shown that for this case, the use of conventional fracture mechanics K-solutions can seriously overestimate the actual K's. The conventional definition of the strain intensity factor was shown to be in error because it fails to take into account both load shedding and the closing bending moments.

4. The thermal-mechanical crack growth (TMFCG) rates in Inconel X-750 were measured under stress-controlled conditions in the temperature range from 300 to 650°C. On a stress intensity basis (ΔK_{σ})

the TMFCG rates were faster than their isothermal counterpart. Using an effective stress range ($\Delta\sigma_{eff}$) computed using measured values of the opening stress, good correlation between isothermal and TMF crack growth rates were obtained on a ΔK_{eff}-basis.

5. The cyclic stress-strain behavior for B-1900+Hf under TMF cycling differs from the isothermal behavior by showing more cyclic hardening (> 15%), both at high and low temperatures. Thus it is difficult to predict the cyclic stress-strain behavior under realistic conditions from isothermal data. The synergistic coupling between the cyclic stress-strain behavior and temperatures cannot be ignored.

6. The cyclic flow stress at elevated temperature (T_{max}) is primarily controlled by the density of misfit dislocations, which depends on the amount of isotropic and directional coarsening. At low temperature (T_{min}) the flow stress is controlled by the directionality of the stress field around the γ', the magnitude of which depends on the sign of the applied stress.

7. Using a strain-based approach (ΔK_{ε}), a poor correlation between the isothermal and the TMFCG rates under elastic ($\Delta\varepsilon_{mec}$ = 0.25%) and/or fully plastic ($\Delta\varepsilon_{mec}$ = 0.50%) conditions were observed. A stress-based approach (ΔK_{σ}) which takes into account the hardening/softening behavior of the materials as well as the load shedding caused by extra-compliance provided by the presence of the crack was shown to rationalize the isothermal and TMF crack growth rates. Furthermore, based upon the ΔK_{eff}, the isothermal crack growth rates at T_{min} and T_{max} were shown to provide an upper and lower bounds for the TMFCG rates, respectively.

8. The cracking modes (transgranular vs. intergranular) in B-1900+Hf and Hastelloy-X are controlled by the occurrence of a critical tensile stress (σ_c) for intergranular or transgranular fracture, rather than by a critical tensile strain. Each mode of fracture is shown to be characterized by a unique crack growth rate vs. ΔK_{eff} curve.

9. The opening stress (σ_{op}) was shown to be a complex function of crack length and strain range.

10. For the materials and testing conditions used in this investigation, the crack growth rates are controlled by the mechanical driving force (ΔK_σ, ΔK_{eff}) only. The time-dependent components (oxidation and creep) were shown not to affect significantly the crack growth rates.

8.0 Recommendations for Future Work

The problem of thermal-mechanical fatigue in hot section engine components was extensively examined. The major areas investigated included a problem survey, experimental fracture mechanics techniques, evaluation of data correlation parameters, and prediction of crack growth under thermal-mechanical cycling. The following are the major observations and recommendations from this effort.

1. The crack propagation testing and data reduction for the various materials tested showed the necessity of the use of a physically sound data correlation parameter. The conventional strain intensity factor shows several results which make its use undesirable for TMF crack growth prediction. There was some strain range dependence on crack growth rates using both ΔK_{ε} and ΔK_{σ} parameters. Of the parameters extensively studied, the ΔK_{eff} was the best all-around approach for correlating high temperature and TMF data. Extensive evaluation of the ΔK_{eff} for TMF cycling is recommended.

2. The fracture mechanics analysis used a compliance approach to calculate a value for the correlation parameter. The compliance approach was originally developed for isothermal testing to obtain an experimental value of the K_{ε}. The compliance approach was extended for a more complicated situation, which includes spatially varying temperatures, and temperature-dependent material properties. Thus, the parameter calculated cannot be termed a "K" in the strictest sense. Further assessment of other parameters which are both theoretically justified and calculable for structural components is recommended.

3. Development of cyclic nonlinear fracture mechanics capability is recommended to better understand TMF crack growth, especially when substantial plastic strain develops.

4. There were marked differences in the specimen crack growth surface features as a function of temperature and TMF cycle. Low temperature isothermal growth was smooth and transgranular. High temperature isothermal growth was rough and intergranular. Out-of-phase TMF crack growth was moderately rough and chiefly transgranular whereas in-phase cycling was rough and intergranular. This evidence shows that not only is the temperature range and strain range important in TMF tests, but the cycle shape is also important. Crack growth in service tends to be of moderate level of surface roughness and chiefly transgranular, similar to out-of-phase TMF tests. This observation lends hope to the ultimate success of using simple TMF data to predict complex ε-T scheme. However, the final degree of success will be determined by the degree to which complex ε-T crack growth is governed by unique mechanisms not present under simple TMF conditions. Further work is recommended to further compare the damage which takes place under TMF and complex ε-T cycles.

9.0 References

1. D.A. Spera, "What is Thermal Fatigue?", ASTM STP 612, 1976, pp. 3-9.

2. R.P. Skelton, "The Growth of Short Cracks During High Strain Fatigue and Thermal Cycling", ASTM STP 770, 1982, pp. 337-381.

3. J. Wareing, "Mechanisms of High Temperature Fatigue and Creep-Fatigue Failure in Engineering Materials", in Fatigue at High Temperature, ed. by R.P. Skelton, Applied Sci. Pub., 1983, pp. 135-185.

4. W.J. Mills and L.A. James, "Effect of Temperature on the Fatigue Crack Propagation Behavior of Inconel X-750", Fat. Eng. Mat. Struct., Vol. 3, 1980, pp. 159-175.

5. G.J. Lloyd, "High Temperature Fatigue and Creep-Fatigue Crack Propagation: Mechanics, Mechanisms and Observed Behavior in Structural Materials", in Fatigue at High Temperature, ed. by R.P. Skelton, Applied Sci. Pub., 1983, pp. 187-258.

6. L.F. Coffin, "Damage Processes in Time Dependent Fatigue--A Review", in Creep-Fatigue-Environment Interactions, ed. by R.M. Pelloux and N.S. Stoloff, AIME Pub., 1980, pp. 1-23.

7. S. Taira, "Relationship Between Thermal Fatigue and Low Cycle Fatigue at Elevated Temperature", ASTM STP 520, 1973, pp. 80-101.

8. E. Glenny, J.E. Northwood, S.W. Shaw and T.A. Taylor, "A Technique for Thermal-Shock and Thermal-Fatigue Testing Based on the Use of Fluidized Solids", J. Inst. Met., Vol. 87, 1959, pp. 194-202.

9. P. Bizon and D.A. Spera, "Thermal Stress Fatigue Behavior of 26 Superalloys", ASTM STP 612, 1976, pp. 106-122.

10. M.A. Howes, "Thermal Fatigue and Oxidation Data on TAZ-8A, Marm-200 and Udimet 700 Superalloys", NASA CR-134775, 1975.

11. K.E. Hofer and V.E. Humphreys, "Thermal Fatigue and Oxidation Data of TAZ-8A and M22 Alloys and Variations", NASA-3-17787, 1981.

12. D.F. Mowbray and J.E. McConnelee, "Nonlinear Analysis of a Tapered Disk Thermal Fatigue Specimen", ASTM STP 612, 1976, pp. 10-29.

13. V. Moreno, G.J. Meyers, A. Kaufman and G.R. Halford, "Nonlinear Structural and Life Analyses of a Combustor Liner", NASA TM-82846, 1982, 21 pgs.

14. D.A. Spera and E.C. Cox, "Description of a Computerized Method for Predicting Thermal Fatigue Life of Metals", ASTM STP 612, 1976, pp. 69-85.

15. L.F. Coffin and R.P. Wesley, "Apparatus for Study of Effects of Cyclic Thermal Stresses on Ductile Metals", Trans. ASME, 1954, pp. 123-130.

16. E.E. Baldwin, G.J. Sokol, and L.F. Coffin, "Cyclic Strain Fatigue Studies on AISI Type 347 Stainless Steel", Trans. ASTM, Vol. 57, 1957, pp. 567-576.

17. T. Udoguchi and T. Wada, "Thermal Effect on Low Cycle Fatigue Strength of Steels", Thermal Stresses and Thermal Fatigue, ed. D.J. Littler, Butterworth, London, 1971, pp. 109-123.

18. R.H. Stentz, J.T. Berling and J.B. Conway, "A Comparison of Combined Temperature and Mechanical Strain Cycling Data With Isothermal Fatigue Results", Proc. 1st Int. Conf. on Structural Mechanics in Reactor Technology, Berlin, Vol. 6, 1971, pp. 391-411.

19. U.S. Lindholm and D.L. Davidson, "Low Cycle Fatigue With Combined Thermal and Strain Cycling", ASTM STP 520, 1973, pp. 473-481.

20. K.D. Sheffler and G.S. Doble, "Thermal Fatigue Behavior of T-111 and ASTAR 811C in Ultra High Vacuum", ASTM STP 520, 1973, pp. 491-499.

21. K.D. Sheffler, "Vacuum Thermal-Mechanical Fatigue Behavior of Two Iron-Base Alloys", ASTM STP 612, 1976, pp. 214-226. (See also NASA-CR-134534.)

22. R.P. Skelton, "Environmental Crack Growth in 1/2 Cr-Mo-V Steel During Isothermal High Strain Fatigue and Temperature Cycling", Mat. Sci. Eng., Vol. 35, 1978, p. 287-298.

23. S. Taira, M. Fujino and H. Higaki, "In-Phase Thermal Fatigue Strength of Steels", J. Soc. Mat. Sci. Japan, Vol. 25, 1976, pp. 375-381.

24. N. Fujino and S. Taira, "Effect of Thermal Cycle on Low Cycle Fatigue Life of Steels and Grain Boundary Sliding Characteristics", ICM 3, Vol. 2, 1979, pp. 49-58.

25. K. Kuwabara and A. Nitta, "Effect of High Temperature Tensile Strain Holding in Thermal Fatigue Fracture", in Symposium on Creep-Fatigue Interaction, ed. R.M. Curran, ASME-MPC, New York, 1976, pp. 161-177.

26. K. Kuwabara and A. Nitta, "Thermal-Mechanical Low Cycle Fatigue Under Creep-Fatigue Interaction in Type 304 Stainless Steel", ICM 3, Vol. 2, 1979, pp. 69-78.

27. C.A. Rau, A.E. Gemma and G.R. Leverant, "Thermal-Mechanical Fatigue Crack Propagation in Nickel and Cobalt-Base Superalloys Under Various Strain-Temperature Cycles", ASTM STP 520, 1973, pp. 166-178.

28. A.E. Gemma, B.S. Langer and G.R. Leverant, "Thermal Mechanical Fatigue Crack Propagation in Anisotropic Nickel-Base Superalloy", ASTM STP 612, 1976, pp. 199-213.

29. A.E. Gemma, F.X. Ashland, and R.M. Masci, "The Effects of Stress Dewlls and Varying Mean Strain on Crack Growth During Thermal Mechanical Fatigue", J. Test. Eval., Vol. 9, No. 4, 1981, pp. 209-215.

30. S. Taira, M. Fujino and R. Ohtani, "Collaborative Study of Thermal Fatigue Properties of High Temperature Alloys in Japan", Fat. Eng. Mater. Struct., Vol. 1, 1979, pp. 495-508.

31. M.J. Westwood, "Tensile Hold Time Effects on Isothermal and

Thermal Low Cycle Fatigue of 304 Stainless Steel", ICM 3, Vol. 2, 1976, pp. 59-67.

32. V.T. Troshchenko and L.A. Zaslotskaya, "Fatigue Strength of Superalloys Subjected to Combined Mechanical and Thermal Loading", ICM 3, Vol. 2, pp. 3-12.

33. A.P. Gussenkov and A.G. Kasantsev, "A Kinetic Deformation Criterion for Creep Low Cycle Fatigue Under Nonisothermal Loading", ICM 3, Vol. 2, 1979, pp. 43-49.

34. V.N. Filatov, A.S. Kruglov, S.V. Evropin and O.G. Yuren, "Nonisothermal Fatigue Testing of 12 kh18N9-Steel Within a Reactor", Zav. Labo., Vol. 47, No. 1, 1981, pp. 77-78.

35. G.A. Tulyakov, V.A. Plekhanov and V.N. Skorobogatykh, "Installation for Low Cycle Isothermal and Nonisothermal Fatigue Tests With Complex Loading", Zav. Labo., Vol. 44, No. 9, 1978, pp. 1133-1136.

36. S.W. Hopkins, "Low Cycle Thermal Mechanical Fatigue Testing", STM STP ;612, 1976, pp. 157-169.

37. N.D. Sobolev and I. Egorov, "Recording Cycle Strain Diagrams Under Nonisothermal Conditions (Review)", Zav. Labo., Vol. 45, No. 5, 1979, pp. 449-452.

38. E.G. Ellison, "Thermal-Mechanical Strain Cycling and Testing at Higher Temperatures", in Measurement of High Temperature Mechanical Properties of Materials, ed. by M.S. Loveday, M.F. Day and B.F. Dyson, HMSO Pub., 1982, pp. 204-224.

39. L.F. Coffin, "Fatigue at Elevated Temperature", ASTM STP 520, 1973, pp. 5-34.

40. R.L. McKnight, J.H. Laflen and G.T. Spamer, "Turbine Blade Tip Durability Analysis", NASA CR-165268, 1981, 112 pgs.

41. G.J. Meyers, "Fracture Mechanics Criteria for Turbine Engines and Hot Section Components", NASA CR-167896, 1982, 115 pgs.

42. M. Clavel, C. Levaillant and A. Pineau, "Influence of Micromechanisms of Cyclic Deformation at Elevated Temperature on

Fatigue Behavior", in Creep-Fatigue-Environment Interactions, ed. by R.M. Pelloux and N.S. Stoloff, AIME Pub., 1980, pp. 24-45.

43. R.H. Cook and R.P. Skelton, "Environment-Dependence of the Mechanical Properties of Metals at High Temperature", Int. Met. Rev., Vol. 19, Rev. 187, 1974, pp. 199-222.

44. H.W. Grunling, K.H. Keienburg and K.K. Schweitzer, "The Interaction of High Temperature Corrosion and Mechanical Properties of Alloys" in High Temperature Alloys for Gas Turbines, ed. by Brunetand, Coutsouradis, Gilbons, Lindblom, Meadowcraft and Stickley, Reidel Pub., 1982, pp. 507-543.

45. J.M. Davidson, K. Aning and J.K. Tien, "Hot Environment Effects on Alloy Mechanical Properties", in Symp. on Properties of High Temperature Alloys with Emphasis on Environmental Effects, Pub. by Electrochemical Soc., Princeton, NJ, 1976, pp. 175-198.

46. P. Marshall, "The Influence of Environment in Fatigue", in Fatigue at High Temperature, ed. by R.P. Skelton, Appl. Sci. Pub., 1983, pp. 253-303.

47. M.A.H. Howes, "Thermal Fatigue and Oxidation of Two Iron Base Alloys", ASTM STP 612, 1976, pp. 86-105.

48. M.O. Speidel and A. Pineau, "Fatigue of High Temperature Alloys for Gas Turbines", in High Temperature Alloys for Gas Turbines, ed. by Coutsouradis, Felix, Tischmeister, Habraken, Lindblom and Speidel, Appl. Sci. Pub., 1978, pp. 469-512.

49. S.D. Antolovich, "Aspects Metallurgiques de la Fatigue a Haute Temperature", in La Fatigue des Materiaux et Structures, ed. by C. Bathias and J.P. Bailon, Montreal University Press, 1980, pp. 465-496.

50. A. Pineau, "High Temperature Fatigue: Creep-Fatigue Oxidation Interactions in Relation to Microstructure", in Subcritical Crack Growth Due to Fatigue, Stress Corrosion and Creep, ed. by ISPRA, Appl. Sci. Pub., in press.

51. G.R. Leverant, T.E. Strangman and B.S. Langer, "Parameters Controlling the Thermal Fatigue Properties of Conventional Cast and Directionally Solidified Turbine Alloys", in Superalloys:

Metallurgy and Manufacture, ed. by Kear, Muzuka, Tien and Wlodek, Claxtors Pub., 1976, pp. 285-295.

52. A. Pineau, "High Temperature Fatigue Behavior of Engineering Materials in Relation to Microstructure", in Fatigue at High Temperature, ed. by R.P. Skelton, Appl. Sci. Pub., 1983, pp. 305-364.

53. D.H. Boone and C.P. Sullivan, "Effect of Several Metallurgical Variables on the Thermal Fatigue Behavior of Superalloys", ASTM STP 520, 1973, pp. 401-415.

54. M. Gell and G.R. Leverant, "Mechanisms of High Temperature Fatigue", ASTM STP 520, 1973, pp. 37-67.

55. F.E. Fujita, "Oxidation and Dislocation Mechanisms in Fatigue Crack Formation", in Fracture of Solids, ed. by D.S. Drucker and J.J. Gilman, AIME-TMS, 1963, pp. 657-670.

56. J. Wareing and H.G. Vaughan, "The Relationship Between Striation Spacing, Macroscopic Crack Growth Rate and the Low Cycle Fatigue Life of a Type 316 Stainless Steel at 625°C", Met. Sci., Vol. 11, 1977, pp. 414-424.

57. R.P. Skelton and J.I. Bucklow, "Cyclic Oxidation and Crack Growth During High Strain Fatigue of Low Alloy Steels", Met. Sci., Vol. 12, 1978, pp. 64-70.

58. T.E. Strangman, "Thermal Fatigue of Oxidation Resistant Overlay Coatings for Superalloys", Ph.D. Dissertation, Univ. of Conn., Storrs, Conn., 1978.

59. R.P. Skelton, "Crack Initiation and Growth in Simple Metals Components During Thermal Cycling", Fatigue at High Temperature, ed. by R.P. Skelton, Appl. Sci. Pub., 1983, pp. 1-62.

60. H.J. Westwood and W.K. Lee, "Creep-Fatigue Crack Initiation in 1/2 Cr-Mo-V Steel", in Creep and Fracture of Engineering Materials and Structure, ed. by B. Wilshire and D.J.R. Owen, Pineridge Press, 1982, pp. 517-530.

61. C.E. Jaske, "Thermal-Mechanical Low Cycle Fatigue of AISI 1010 Steel", ASTM STP 612, 1976, pp. 170-198.

62. A.A. Sheinker, "Exploratory Thermal-Mechanical Fatigue Results for Rene 80 in Ultrahigh Vacuum", NASA CR-159444, 1978, 16 pages.

63. S. Bhongbhobhat, "The Effect of Simultaneously Alternating Temperature and Hold Time in the Low Cycle Fatigue Behavior of Steels", in Low Cycle Fatigue Strength and Elasto-plastic Behavior of Materials, ed. by K.T. Rie and E. Harbach, DVM Pub., 1979, pp. 73-82.

64. W.R. Adams and P. Stanley, "A Programmable Machine for Simulated Thermal Fatigue Testing", J. Phys. E. Sci. Inst., Vol. 7, 1974, pp. 669-673.

65. R.A.T. Dawson, W.J. Elder, G.J. Hill and A.T. Price, "High-Strain Fatigue of Austenitic Steels", in Thermal and High Strain Fatigue, ed. by The Metals and Metallurgy Trust, London, 1967, pp. 239-269.

66. A. Nitta, K. Kuwabara and T. Kitamura, "The Characteristic of Thermal-Mechanical Faltigue Strength in Superalloys for Gas Turbines", in Proc. of the Int. Gas Turbine Congress, Tokyo, 23-28 Oct., 1983.

67. T. Koizumi and M. Okazaki, "Crack Growth and Prediction of Endurance in Thermal-Mechanical Fatigue of 12 Cr-Mo-V-W Steel", Fat. Eng. Mat. & Struct., Vol. 1, 1979, pp. 509-520.

68. K. Sadananda and P. Shahinian, "Effect of Environment on Crack Growth Behavior in Austenitic Stainless Steels Under Creep and Fatigue Conditions", Met. Trans., Vol. 11A, 1980, pp. 267-276.

69. M. Okazaki and T. Koizumi, "Crack Propagation During Low Cycle Thermal-Mechanical and Isothermal Fatigue at Elevated Temperatures", Met. Trans., Vol. 14A, 1983, pp. 1641-1648.

70. D.A. Woodford, "Thermal Fatigue Testing of Gas Turbine Materials", ICM 3, Vol. 2, 1979, pp. 33-42.

71. D.A. Woodford and D.F. Mowbray, "Effect of Material Characteristics and Test Variables on Thermal Fatigue of Cast Superalloys", Mat. Sci. & Eng., Vol. 16, 1974, pp. 5-43.

72. K.D. Scheffler and J.J. Jackson, "Stress Analysis, Thermo-Mechanical Fatigue Evaluation and Root Subcomponent Testing of Gamma/Gamma Prime-Delta Eutectic Alloy", NASA CR-135005, 1976, 52 pages.

73. C. Berk and A.T. Santhanan, "Effect of Microstructure on the Thermal Fatigue Resistance of Cast Cobalt Alloy Mar-M509", ASTM STP 612, 1976, pp. 123-140.

74. W.J. Ostergren, "A Damage Function and Associated Failure Equations for Predicting Hold Time Frequency Effects in Elevated Temperature Low-Cycle Fatigue", J. Test. Eval., Vol. 4, 1976, pp. 327-339.

75. A.E. Carden and T.B. Slade, "High Temperature Low Cycle Fatigue Experiments on Hastelloy-X", ASTM STP 459, 1969, pp. 111-129.

76. P. Shahinian, "Fatigue Crack Growth Characteristics of High Temperature Alloys", Met. Techno., Vol. 5, 1978, pp. 372-380.

77. R. Hales, "Fatigue Testing Methods at Elevated Temperature" in Fatigue High Temperature, ed. by R.P. Skelton, Appl. Sci. Pub., 1983, pp. 63-96.

78. A.E. Carden, "Thermal Fatigue Evaluation", ASTM STP 465, 1969, pp. 163-188.

79. A.E. Carden, "Fatigue at Elevated Temperature: A Review of Test Methods", ASTM STP 520, 1973, pp. 195-223.

80. T.C. Yen, "Thermal Fatigue--A Critical Review", Bull. No. 72, Welding Research Council Pub., WRCBA, 1961, 25 pages.

81. A.E. Carden, "Thermal Fatigue--An Analysis of the Conventional Experimental Method", ASTM, Vol. 63, 1963, pp. 735-758.

82. E. Glenny, "Thermal Fatigue", in High Temperature Materials in Gas Turbines, ed. by P. Sahn and M.O. Speidel, Elsevier Sci. Pub., 1974, pp. 387-395.

83. T. Slot, R.M. Stenz and J.T. Berling, "Controlled Strain Testing Procedures", ASTM STP 465, 1969, pp. 100-128.

84. E.A. Averchenkov, V.I. Egorov and N.D. Sobolev, "Method for Low Cycle Fatigue Testing of Isothermal and Nonisothermal Loading", Zav. Lab., Vol. 48, No. 8, 1982, pp. 59-61.

85. N.W. Brown and K.J. Miller, "A Biaxial Fatigue Machine for Elevated Temperature Testing", J. Test. Eval., Vol. 9, 1981, pp. 202-209.

86. M.W. Brown, "Low Cycle Fatigue Testing Under Multiaxial Stresses at Elevated Temperature", in Measurement of High Temperature Mechanical Properties of Materials, ed. by M.S. Loveday, M.F. Day and B.F. Dyson, HMSO, 1982, pp. 185-203.

87. J.M. Andrews and E.G. Ellison, "A Testing Rig for Cycling at High Biaxial Strains", J. Strain Analysis, Vol. 8, 1981, pp. 168-175.

88. C.H. Wells, "Elevated Temperature Testing Methods", ASTM STP 465, 1969, pp. 87-99.

89. L.A. James, "The Effect of Stress Ratio on the Elevated Temperature Fatigue Crack Propagation of Type 304 Stainless Steel", Nucl. Tech., Vol. 14, 1972, pp. 163-170.

90. J.D. Whittenberger and P.T. Bizon, "Comparative Thermal Fatigue Resistance of Several Oxide Dispersion Strengthened Alloys", Int. J. Fat., 1981, pp. 174-180.

91. C. Levaillant and A. Pineau, "Assessment of High Temperature LCF Life of Austenitic Stainless Steel by Using Intergranular Damage as Correlating Parameter", ASTM STP 770, 1982, pp. 169-193.

92. P.S. Prevey and W.P. Koster, "Effect of Surface Integrity on Fatigue of Structural Alloys at Elevated Temperature", ASTM STP 520, 1973, pp. 522-536.

93. Y. Asada, R. Yuuki, D. Sumamoto, T. Sokon, K. Tokimasa, Y. Makino, M. Kitagawa and K. Shingai, "Fatigue Crack Propagation Under Elastic-Plastic Medium at Elevated Temperature", in Proc. of the 4th Int. Conf. Press. Vessel Technology, pub. by Inst. of Mech. Engrs., UK, Vol. 1, 1981, pp. 347-353.

94. G.K. Harritos, D.L. Miller and T. Nicholas, "Sustained Load Crack Growth in Inconel 718 Under Non-isothermal Conditions", J.

Eng. Mat. Tech., in press.

95. F. Mudry, "Etude la Rupture Ductile et de la Rupture par Clicage d'Ariers Faiblement Allies", These de Doctorat d'Etat, Universite de Technologie de Compiegne, 1982.

96. H.O. VanLeeven, "Heating Methods in Materials Testing at Elevated Temperature", ed. by Nat. Lucht., Netherlands, TMM 2087, 1961, 23 pages.

97. Numerical Methods in Thermal Problems, Proc. of the 2nd Int. Conf., ed. by G.H. Morgen and B.S. Schiefler, Pineridge Press Pub., 1982, 1383 pages.

98. ABAQUS (User's Manual - Version 4), Hibbilt, Karlsson and Sorensen, Inc., 1983.

99. P.T. Raske and W.F. Burke, "An Extensometer for Low Cycle Fatigue Tests on Anisotropic Materials at Elevated Temperature", J. Phys. E. Sci. Instrum., Vol. 12, 1979, pp. 175-176.

100. G. Sunner, "A Diametral Extensometer for Elevated Temperature High Strain Fatigue", J. Phys. E. Sci. Instrum., Vol. 1, 1968, pp. 652-654.

101. L.F. Coffin, "The Stability of Metals Under Cyclic Plastic Strain", Trans. of the ASME, J. of Basic Engr., Vol. 82, No. 3, 1960, pp. 671-682.

102. L.F. Coffin, "Instability Effects in Thermal Fatigue", ASTM STP 612, 1976, pp. 227-238.

103. K.J. Miller, Discussion of the paper by Saheb and Bui Quoc (paper C186/73) and the results by Toor (paper C168/73), in Creep and Fatigue in Elevated Temperature Applications, ed. by Inst. Mech. Engrs. and ASTM, Vol. 2, 1973, p. 23.

104. N.J. Marchand, "Fatigue Oligocyclique et Fatigue-Propagation du Cuivre et Laiton-Alpha", MS Thesis, Ecole Polytechnique de Montreal, 1983.

105. H. Teranishi and A.J. McEvily, "The Effect of Oxidation on Hold Time Fatigue Behavior of 2 1/4 Cr-Mo Steel", Met. Trans., Vol.

10A, 1979, pp. 1806-1808.

106. R. Hales, "The High Temperature Oxidation Behavior of Austenitic Stainless Steel", Wekstoffe und Korrosion, Vol. 29, 1978, pp. 393-399.

107. J.S. Huang and R.M. Pelloux, "Low Cycle Fatigue Crack Propagation in Hastelloy-X at 25 and 760°C", Met. Trans., Vol. 11A, 1982, pp. 899-904.

108. B.A. Leich, "Microstructural Effects on the Room and Elevated Temperature Low Cycle Fatigue Behavior of Waspaloy", MS Thesis, University of Cincinnati, 1982.

109. M.F. DAy and G.F. Harrison, "Design and Calibration of Extensometers and Transducers", in Measurement of High Temperature Mechanical Properties of Materials, ed. by M.S. Loveday, M.F. Day and B.F. Dyson, HMSO, 1982, pp. 225-240.

110. D.J. Walter and R. Hales, "An Extensometer for Creep-Fatigue Testing at Elevated Temperatures and Low Strain Ranges", J. Strain Anal., Vol. 16, 1981, pp. 145-147.

111. R. Hales and D.J. Walter, "Measurement of Strain in High Temperature Fatigue", in Measurement of High Temperature Mechanical Properties of Materials, ed. by M.S. Loveday, M.F. Day and B.F. Dyson, HMSO, 1982, pp. 241-254.

112. C. Levaillant, B. Rezgui and A. Pineau, "Effects of Environment and Hold Times on High Temperature Low Cycle Fatigue Behavior of 316 LC Stainless Steel", ICM3, Vol. 2, 1979, pp. 163-172.

113. A.J. Perry, "Review: Cavitation in Creep", J. Mat. Sci., Vol. 9, 1974, pp. 1016-1039.

114. G. Greenwood, "Creep and Cavity Growth Under Multiaxial Stresses", Phil. Mag., Vol. 43, 1981, pp. 281-290.

115. M.F. Day and G.B. Thomas, "Microstructural Assessment of Fractional Life Approach to Low Cycle Fatigue at High Temperatures", Met. Sci., Vol. 13, 1979, pp. 25-33.

116. M.S. Starkey and R.P. Skelton, "A Comparison of the Strain Inten-

sity and Cyclic J Approaches to Crack Growth", Fat. Eng. Mat. Struct., Vol. 5, 1982, pp. 329-341.

117. L.P. Wynn, "TPE 331/T76 Turboprop Propulsion Engine Durability", AFWAL-TR-82-4069, 1982, 53 pages.

118. F. Erdogan and M. Ratwani, "Fatigue and Fracture of Cylindrical Shells Containing a Circumferential Crack", Int. J. Fract. Mech., Vol. 6, 1970, pp. 379-392.

119. R.P. Skelton, "The Effect of Microstructure and Tensile Dwell on the Growth of Short Fatigue Cracks in 316 Steel at 625°C", Proc. of the Int. Conf. on Mechanical Behavior and Nuclear Application of Stainless Steels at Elevated Temperature, ed. by The Metals Soc., UK, 1981, pp. 191-203.

120. A.C. Pickard and M.A. Hicks, "Crack Length Determination Using the D.C. Electrical Potential Drop Technique--A Comparison of Calibration Methods", in Advances in Crack Length Measurement, ed. by C.J. Beevers, EMAS Pub., 1982, pp. 97-113.

121. K. Krompholz, E.D. Grosser, K. Ewart and E. Moritz, "Application of the D.C. Potential Drop Technique in the High Temperature Regime", in Advances in Crack Length Measurement, ed. by. C.J. Beevers, EMAS Pub., 1982, pp. 231-251.

122. F. Gabrielli and R.M. Pelloux, "Effect of Environment on Fatigue and Creep Crack Growth in Inconel X-750 at Elevated Temperature", Met. Trans., Vol. 13A, 1982, pp. 1083-1089.

123. V. Moreno, "Creep-Fatigue Life Prediction for Engine Hot Section Materials (Isotropic)", NASA CR-168228, 1983, 85 pages.

124. "Engineering Properties of Inconel X-750", Technical Bulletin T-38, Huntington Alloy Products Division of the International Nickel Company, Inc., Huntington, WV.

125. K.P. Walker, "Research and Development Program for Nonlinear Structural Modeling with Advanced Time-Temperature Dependent Constitutive Relationships", NASA CR-165533, 1982, 187 pages.

126. Vito Moreno and J.T. Hill, Private Communication, 1984.

127. S.D. Antolovich, S. Liu and R. Baur, "Low Cycle Fatigue Behavior of Rene 80 at Elevated Temperature", Met. Trans., Vol. 12A, 1981, pp. 473-481.

128. K. Kuwabara, A. Nitta and T. Kitamura, "Thermal-Mechanical Fatigue Life Prediction in High-Temperature Component Materials for Power Plants", in Advances in Life Prediction Methods, ed. by D.A. Woodford and J.R. Whitehead, ASME MCP, 1983, pp. 131-142.

129. D.A. Jablonski, "Fatigue Behavior of Hastelloy-X at Elevated Temperature in Air, Vacuum and Oxygen Environments", Ph.D. Thesis, M.I.T., January 1978.

130. R.C. Bill, M.J. Verrille, M.A. McGaw and G.R. Halford, "Preliminary Study of Thermomechanical Fatigue of Polycrystalline MAR-M200", NASA TP-2280 AVSCOM TR 83-C-6, 1984, 16 pages.

131. V. Moreno, D.M. Nissley and L.S. Lin, "Creep-Fatigue Life Prediction for Engine Hot Section Materials", NASA CR-174844, 1984, 137 pages.

132. J.K. Tien and R.P. Gamble, "Effects of Stress Coarsening on Coherent Particle Strengthening", Met. Trans., Vol. 3, 1972, pp. 2157-2162.

133. D.O. Pearson, F.O. Lemkey and B.H. Kear, "Stress Coarsening of γ' and its Influence on Creep Properties of a Single Crystal Superalloy", in Proc. of the 4th Intl. Symp. on Superalloys, ed. by J.K. Tien, ASM, Metals Park, 1980, pp. 513-520.

134. L.S. Lin and J.M. Walsh, "Lattice Misfit by Convergent-Beam Electron Diffraction--Part 1: Geometry Measurement", in Proc. of the 42nd Annual Meeting of the EMSA, 1984, pp. 520-524.

135. D.M. Shah and D.N. Duhl, "The Effect of Orientation, Temperature and Gamma Prime Size on the Yield Strength of a Single Crystal Nickel-Base Superalloy", in Superalloy 1984, ed. by M. Gell et al, AIME, Warrendale, PA, 1984, pp. 107-116.

136. J.O. Nilson, "The Influence of Nitrogen on High Temperature Low Cycle Fatigue Behavior of Austenitic Stainless Steels", Fat. Eng. Mater. Struct., Vol. 7, No. 1, 1984, pp. 35-64.

137. K. Hatanaka and T. Yamada, "Effect of Grain Size on Low Cycle

Fatigue in Low Carbon Steel", Bull. of JSME, VOl. 24, No. 196, 1981, pp. 1692-1699.

138. T. Magnin, J. Driver, J. Lepinoux and L.P. Kubin, "Aspects Microstructuraux de la Deformation Cyclique dans les Metaux et Alliages C.C. et C.F.C..1 Consolidation Cyclique", Rev. Phy. Appl., Vol. 19, 1984, pp. 467-482.

139. G.R. Leverant, B.H. Kears and J.M. Oblack, "Creep of Precipitation-Hardened Nickel-Base Alloy Single Crystals at High Temperatures", Met. Trans., Vol. 4, 1973, pp. 355-362.

140. R.R. Jensen and J.K. Tien, "Temperature and Strain Rate Dependence of Stress-Strain Behavior in a Nickel-Base Superalloy", Met. Trans., Vol. 16A, 1985, pp. 1049-1068.

141. T. Miyazaki, K. Nakamura and H. Mori, "Experimental and Theorical Investigations on Morphological Changes of γ'-Precipitates in Ni-Al Single Crystals During Uniaxial Stress-Annealing", J. Mater. Sci., Vol. 14, 1979, pp. 1827-1837.

142. R.W.K. Honeycombe, The Plastic Deformation of Metals, 2nd Ed., Edward Arnold Pub., 1984.

143. M. Gell and D.N. Duhl, "The Development of Single Crystal Superalloy Turbine Blades", Proc. of the N.J. Grant Symposium on Processing and Properties of Advanced High-Temperature Alloys, June 17-18, 1985, ASM Pub., in press.

144. P. Lukas, L. Kunz, Z. Knesl, B. Weiss and R. Stickler, "Fatigue Crack Propagation Rate and the Crack Tip Plastic Strain Amplitude in Polycrystalline Copper", Mater. Sci. & Eng., Vol. 70, 1985, pp. 91-100.

145. D.O. Harris, Trans. of ASME, J. of Basic Eng., Vol. 89, 1969, pp. 519-526.

146. G.T. Embley and E.S. Russel, "Thermal-Mechanical Fatigue of Gas Turbine Bucket Alloys", in the Proc. of the First Parsons International Turbine Conference, Dublin, Ireland, June 1984, pp. 157-164.

147. W. Elber, "Significance of Fatigue Crack Closure, Damage Tolerance

in Aircraft Structures", ASTM STP 486, 1971, pp. 230-242.

148. P.C. Paris and H. Herrmann, "Twenty Years of Reflection on Questions Involving Fatigue Crack Growth, Part II: Some Observations of Crack Closure", in Fatigue Threshold, EMAS Pub., Warely, U.K., 1981, pp. 11-32.

149. T. Hoshide and K. Tanaka, "Analysis of Fatigue Crack Growth Propagation of Surface Flaws by Elastic-Plastic Fracture Mechanics", Trans. Jap. Soc. of Mech. Engrs., Vol. 48, 1982, pp. 1102-1110.

150. K. Tanaka, "Short-Crack Fracture Mechanics in Fatigue Conditions", in Current Research on Fatigue Research on Fatigue Cracks, ed. by T. Tanaka, M. Jono, and K. Komai, JSMS Pub., Kyoto, Japan, 1985, pp. 79-100.

151. Y. Murakami, S. Hrada, T. Endo, H. Tani-Ishi, and Y. Fukushima, "Correlations Among Growth Law of Small Cracks, Low Cycle Fatigue Law and Applicability of Miner's Rule", Eng. Frac. Mech., Vol. 18, 1983, pp. 909-924.

152. T. Yamada, T. Moshide, S. Fujimura and M. Manabe, "Investigation of Fatigue Life Prediction Based on Analysis of Surface-Crack Growth in Low Cycle Fatigue of Smooth Specimen of Medium Carbon Steel", Trans. of the JSME, Vol. 49A, 1983, pp. 441-451.

153. Y. Murakami, "Correlation Between Strain Singularity at Crack Tip Under Overall Plastic Deformation and the Exponent of the Coffin-Manson Law", to appear in the Proc. of the Conf. on Low Cycle Faltigue--Directions for the Future, Lake George, NY, September 30 - October 4, 1985.

154. E.L. Wagoner, "Physical Metallurgy and Mechanical Properties of Hastelloy-X", Technical Report, Haynes-Stellite Corp., June 23, 1961.

155. J.T. Hill, Private Communication, Pratt & Whitney Aircraft Corp., 1985.

THERMAL-MECHANICAL FATIGUE BEHAVIOR OF NICKEL-BASE SUPERALLOYS

VOL. 2

by

NORMAN J. MARCHAND

B. Ing., Ecole Polytechnique de Montreal, 1980
M.Sc.A., Ecole Polytechnique de Montreal, 1982

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF

DOCTOR OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
January 1986

Signature of Author . . . .
Department of Materials Science & Engineering
January 10, 1986

Certified by . . . .
R. M. Pelloux
Thesis Supervisor

Accepted by . . . .
Bernhardt J. Wuensch
Chairman, Departmental Graduate Committee

Appendix I

Thermal-Mechanical Fatigue Methods of Testing

1. Calibration and Recording of High Temperature Behavior

The first attempt at plotting stress-strain diagrams under thermal fatigue loading were made on the basis of Coffin's method. For instance, the diagram may be plotted by points corresponding to the half-cycles of heating and cooling. The stress is recorded by a load cell (transducers, etc.) and the strain is determined by calculation or measuring thermal or total strain at different instants of loading. Carden [78-79] explains the method of recording the parameters of thermal fatigue loading which employs the recording of the diagram of force (stress) vs mean temperature of the specimen and which makes it possible to determine the plastic strain of the specimen in any cycle. The most complicated feature in selecting the method of recording the strain parameters is the determination of the true strain at every instant of loading of the specimen. In many earlier investigations [80-82], the mean plastic strain per cycle was determined regardless of its localization caused by irregular temperature distribution and regardless of the deformation process.

In connection with the above, it is worth giving some attention to methods of direct experimental determination of strains in the critical section of the specimen at all stages of its loading.

Carden [79] suggested a method of obtaining stress-strain diagrams in thermal fatigue with the aid of two extensometers mounted on the specimen for measuring strain in the longitudinal and transverse directions; this involves the recording of the difference in the readings of the strain gauges, and in isotropic thermal extension the component of thermal strain is automatically eliminated. This method undoubtedly permits obtaining a more accurate strain pattern in thermal fatigue; however, it has its limitations. The longitudinal extensometer measures the strain on some base, and if there is a longitudinal temperature gradient along the specimen, an error may occur in the determination of local strain. Besides, it is assumed that the Poisson ratio ν does not depend on the temperature; this is not always correct; e.g., in the case of a steel 1Kh2M, changes from 0.27 to 0.31 when the temperature changes within the range of 100-600°C [37].

In our opinion, the method of automatic recording of the stress-strain diagram in thermal fatigue tests using one diametral extensometer [18, 20-21, 31, 33, 61-62, 83-84] has some advantages compared with Carden's method. Similar extensometers are widely used in low cycle fatigue tests; they consist of a system of two levers connected in the central part by an elastic hinge. The ends of the levers encircle the specimen (tubular or solid) diametrically; after the other ends have been displaced and the displacement recorded by a transducer, the diametral strain of the specimen is recorded. On an installation of the Coffin type, this strain in cyclic heating and cooling of the specimen consists of a mechanical and a thermal component. To eliminate the thermal part of the strain, an electrical method of compensation is used [36, 83-84]. From the signal proportional to the total strain (transducer-induced by the extensometer), the signal proportional to the thermal strain (induced by the thermocouple welded on at the investigated seciton of the specimen) is subtracted. One input of the x-y plotter receives a signal from the load-cell, the other receives the difference between the two signals proportional to the mechanical strain, and this yields the stress-strain diagram under nonisothermal conditions in coordinates of stress versus diametral strain. To obtain a diagram of the cyclic strains in terms of stress versus longitudinal strain, it is necessary to rearrange the recorded diagram and to convert the transverse strain into longitudinal strain. The total longitudinal axial strain ε_a can be expressed by the component of transverse strain at each instant of deformation:

$$\varepsilon_a = P/A \quad (1-2\nu(T))/E(T) \;-\; 2\Delta d/d$$

where d is the diameter and Δd the change in diameter of the specimen. A is its cross-sectional area; P is the force applied to the specimen; E(T) and ν(T) are the modulus of elasticity and the Poisson ratio, respectively, as functions of the temperature T.

There are similar methods of compensating for thermal strains as suggested by Slot et al. [83]; all of them involve the use of an analogue computer to obtain stress-strain diagrams in coordinates of longitudinal stress versus longitudinal strain. This is an important methodological advantage because these methods were used under conditions of independent thermal mechanical loads. However, it is generally accepted in these methods that the characteristics of elasticity E and ν remain constant at all stages of the strain cycle (i.e., their temperature dependence is not taken into account). This undoubtedly introduces an error. To calculate the change in cross-sectional area of the specimen and the elastic characteristics, a functional generator in combination with the analogue computer can be used [36].

At this point, it is important to notice that a change in the thermal loading program requires a change in the program which issues the compensating signals, whereas at moderate temperatures the previously examined method requires no correction. This problem can be ameliorated by the method described by Hopkins [36], which suggests varying the independent mechanical and thermal operating regimes. To compensate for thermal expansion and to obtain data on the magnitude of the mechanical strains, a method is used which is analogous to that described previously. The strain measuring channel receives, together with the signal from the strain meter but in opposite phase, a signal from the setting mechanisms whose program corresponds to the steady-state thermal strain of the specimen in zero-load-controlled cyclic temperature change.

Attention should be given to the arguments of Hopkins [36] and Sobolev and Egorov [37] in substantiating the method which used an analogue computer: they pointed out that it is possible that the functions expressing the temperature of thermal strain do not coincide at the heating and cooling stages. This noncoincidence can be particularly important in tests conducted with a thick-walled specimen whose deformation occurs under conditions of a nonuniform state of stress and strain and an inhomogeneous temperature field, where thermal stresses exist which were not taken into account.

The method of making torsional tests with tubular specimens has a number of methodological advantages from the point of view of recording the diagrams of nonisothermal cyclic strain, because independent programs of mechanical and thermal loading can be realized [32-33, 35]. Tulyakov et al. [35], Brown et al. [85-86], and

Andrews and Ellison [87] have described an installation with follower control system with load and temperature feedback based on this principle. The cyclic strain diagrams recorded in these tests take into account the nonuniformity of heating of the specimen which determines the change in the measuring base of the deformer because of the thermal axial strain of the specimen. We note that the use of thin-walled tubular specimens in cyclic torsion makes it possible to investigate the deformation process at very large strains where specimens in push-pull tests lose their stability.

It is difficult to reduce the hysteresis loop of thermal strain versus temperature to exactly zero. As pointed out by Hopkins [36], small changes in temperature can translate into large changes in mechanical strain if the total strain is left unchanged, whereas a small change in temperature during a cycle without changing the mechanical strain usually does not change the fatigue response of the materials. With the use of an analogue computer, accuracy is only limited by how closely it is adjusted to zero mechanical strain during the thermal cycling. Obviously, for a given thermal strain error, the test accuracy increases as the mechanical strain increases. However, to conduct these types of tests with an analogue computer is difficult and great care must be provided to insure that the proper heating and cooling compensation functions are used. With a digital computer as the programmer, this is not necessary. The computer can acquire and store the thermal strain as a function of both temperature and direction, whereas the thermal expansion compensator's output is only a function of temperature and not direction. A digital computer also allows more complex TMF testing to be conducted with no additional effort from the operator. Such tests may include mode transfers, strains for a given time or to a given creep strain [25-26, 29, 31]. Digital computers can almost continuously read the instantaneous temperature and adjust its total strain output to compensate for any temperature error. The frequency at which the computer does this can be programmed to the maximum heating rate of the temperature equipment and the accuracy desired from the test.

These control problems never surface in isothermal tests because static temperature control is much better than dynamic temperature control, and any change in the temperature which does occur only reflects in a mean strain shift rather than a strain range shift. Fatigue life, either measured as a number of cycles to crack initiation or as a crack propagation life, is affected more by a small change in strain range than it is by the same change in mean strain [29].

2. Heating and Cooling Methods

As pointed out by Carden [79], there are several categories of heating facilities for high temperature fatigue. They are: (1) Combustion furnace heating, (2) Electric resistance heating, and (3) Electric induction heating. Because of the experimental difficulties associated with the control of temperature, combustion furnaces are seldom used in TMF experiments. However, they are sometimes used [79] in testing which tends to reproduce the environmental conditions found in the combustion stage of aircraft engines. The electrical resistance category can be subdivided into four types of methods, respectively: (a) direct resistance (self heating) [19-21, 33, 62, 78-79],

(b) electrical resistance furnace [88-89], (c) fluidized bed [8-11, 90], (d) incandescent lamps [91-94], and (e) electric blankets [95]. The electric induction heating category can be divided into the direct induction method and indirect induction heating (susceptors).

As pointed out by Van Leeuwen [96], there is a relationship between the heating method, the design of the specimen, and temperature measurement and control. Radio frequency induction requires close coil-to-specimen spacing and small grip ends, which makes difficult the mounting of additional measurement devices and obscures access to and visibility of the surface for replication, etc; also the coil has a high voltage RF potential requiring insulation and safety precaution. A temperature disturbance occurs at the thermocouple mass in the field, as well as at specimen locations unequally spaced to the coil [83]. There are also RF induction-induced ground loop problems in the instrumentation cabling.

Resistance heating induces a longitudinal temperature distribution which must be minimized by the incorporation of internal heat sinks and longer lengths of uniform cross-section. Specimen cracks normal to the current flow are accompanied by local heating at the crack tips which is not acceptable for crack propagation testing. No heating can occur after the specimen fractures as can occur with induction heating, but local melting as a result of arcing can occur during stage III final fracture. However, neither direct RF induction nor resistance heating is good for very high electrical conductivity materials, and for these materials the impedance must be matched to the output circuits in both methods. Notwithstanding these problems, both methods are still the most frequently used for TMF testing and for strain life experiments, both methods have been proven satisfactory [19-35]. However, as mentioned before, direct resistance heating is not acceptable for crack propagation testing due to local heating at crack tips, and RF heating is used instead [27-29]. For these tests, low frequency heaters (10 KHz) are often used [28-29] because they produce lower wall thickness gradients than do the higher frequency (450 KHz) heaters and because there are fewer ground-loop problems with the 10 KHz heater than with the 450 KHz heaters. Nevertheless, it is the present author's belief that high frequency induction heating is preferable because of its fast response, making it possible to provide high heating rates and induce considerable thermal stress in specimens of small cross-section. In this case, the heating mode is close to that encountered in-service, since the heat is generated within the surface layer of the specimen. Analysis of the temperature fields in the specimens sometimes needs to be carried out with the high-frequency heaters, but numerical methods of non-stationary problems of heat conductivity using finite elements are now in our hands [12-14, 97-98]. Temperature measurement is usually accomplished by thermocouples welded onto the specimen. However, some cases of premature specimen failure due to the presence of a thermocouple have been reported [88], and optical pyrometry has been used for temperature feedback control [27-29, 36]. Few reports give calibration methods or internal analysis of the uncertainty of the readings [83].

3. Specimen Design and Measurement of Strain

Two types of machines have been used over the last 20 years for studying high temperature low cycle fatigue. One group is a natural evolution of machines used for uniaxial tensile testing, but these are generally operated in fundamentally different modes from their predecessors. The other type of machines achieves the reversed plastic strain required by deforming the material in bending. In the following, we are concerned with the specimen and appropriate extensometry used in uniaxial testing. The reader is referred to Hales [77] for a review of reverse bending testing.

An accurate measurement of strain is fundamental to mechanical fatigue testing since strain is usually the control parameter in such experiments. Also, some materials exhibit grain boundary cavitation as a result of creep-fatigue cycling and the cavities are believed to nucleate and grow as a result of deformation at slow strain rates [99]. This phenomenon has imposed extra conditions on strain measuring systems. More important, they must be stable at least over the period of the hold time and the stress relaxation behavior under nominally constant strain conditions must reflect the true behavior of the material. To perform strain-controlled tests, two methods are commonly adopted, axial and diametral strain control. These commonly employed test methods are critically assessed in the light of the need to perform tests at low (<0.5 %) strain ranges on cyclically hardening or softening behaviors.

3a. Diametral Strain Control

The majority of mechanical property tests are performed under uniaxial stress. This is true of the present state of creep-fatigue studies. Because of the difficulties experienced in attaching axial extensometers to some notch sensitive materials, many workers have used dimensional changes of the diameter to control the tests. Their measurements are usually converted to equivalent axial strain for purposes of comparison with the uniaxial data. This technique is usually used in conjuction with "hour-glass" shaped specimens. Only the strain in the narrowest portion of the specimen is considered. The diametral strain is measured by two probes which are held in contact with opposite sides of the specimen, the displacement being measured either directly or via a hinged arm, again by means fo LVDTs. A full description of these techniques is given by Slot et al. [83] and by Sunner [100].

As we have seen before, diametral strain is usually converted to axial strain since this is the parameter of interest. However, this conversion is not simple since the contribution of elastic and plastic deformation changes as the system cyclically hardens and also as the axial stress relaxes during a hold period. Therefore, to control axial strain with diametral strain measurement, continuous computation of the relationship

$$\varepsilon_a = \sigma / E \ (1/\nu_p - 1/\nu_e) - \varepsilon_d / \nu_p$$

is necessary [83].

Diametral strain control with the use of "hour-glass" specimens is open to a number of serious criticisms, some of a fundamental and some of a practical nature.

Criticism of a fundamental nature

(1) The above equation for computation of axial strain requires that the volume of the material remains constant, which is not the case when creep cavitation occurs.

(2) Diametral strain measurement is inappropriate for testing weldment and other materials which have directionally variable properties [99].

(3) The method is intrinsically insensitive. For example, a typical length-to-diameter ratio is 2:1 and for a specimen 25 mm - 12 mm diameter strained to 0.2 per cent, the axial elongation is approximately 50 μm, whereas in the same test the diameter contracts by only 7-12 μm (i.e., a factor of 5 smaller than a corresponding measurement made axially).

(4) Hour-glass specimens are used because the gently changing cross-section along the gauge length does not appear to generate unacceptable stress concentration gradients and the strain at the plane of maximum cross-section is believed to be uniform. However, it is observed that cyclically strained hour-glass specimens change their shape (Figure I.1) during the progress of fatigue testing, particularly at high cyclic strains [20-21, 31, 39, 60, 79, 101-102]. These important geometrical instabilities of solid and tubular hour-glass specimens have been pointed out by Coffin [102] and Miller [103] but received surprisingly little attention. Coffin has also tried to explain these results in terms of a positional variation or gradient in the cyclic stress-strain field caused by geometry, temperature or material variations. However, his analysis relied on a simple monotonic strain distribution which assumes constant strain in every slice of the cross-section. Miller [103] and Marchand [104] have performed fully-plastic finite element

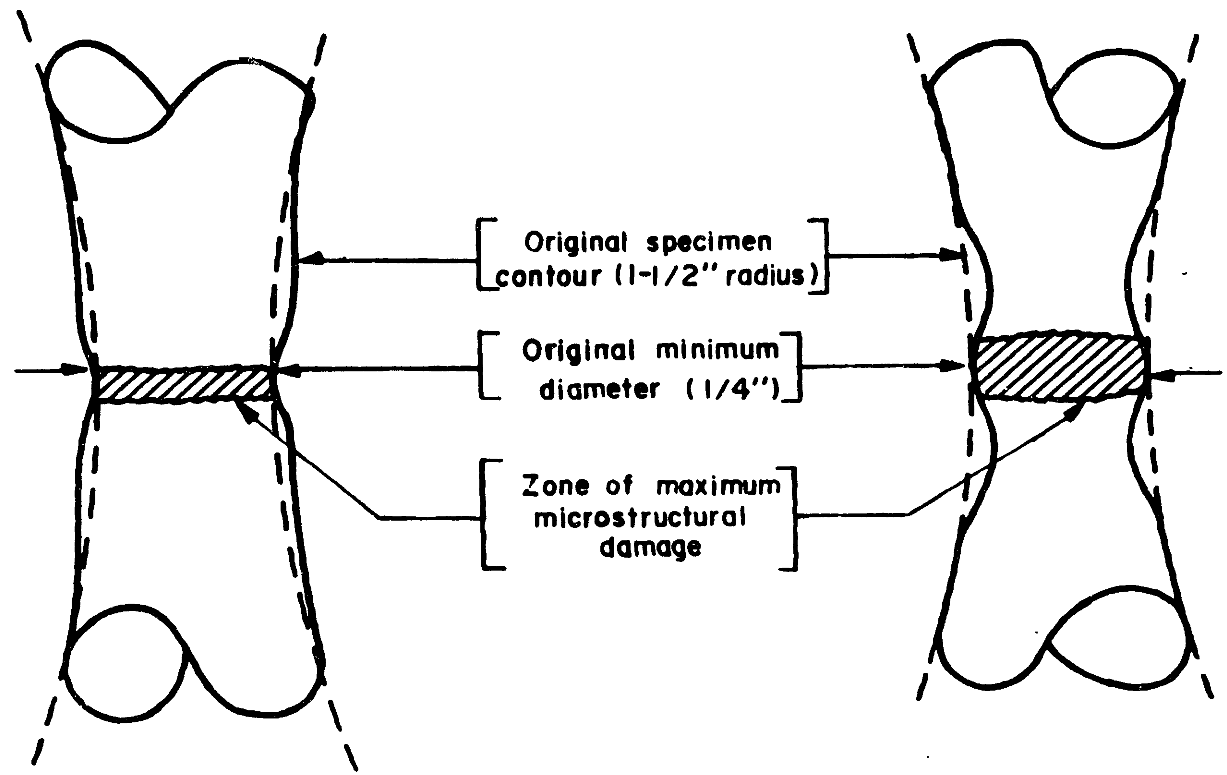


Figure I.1 Schematic illustrations of geometric changes in thermal-mechanical fatigue. (a) In-phase cycling, (b) Out-of-phase cycling [102].

analysis which show that the hour-glass profile acts as a notch, even at large radii of 40 mm (1 1/2") and clearly show the non-linearity between diametral and axial strains. Further, at high strain levels, the maximum strain will not be at the outer surface but at the core of minimum section (Figure I.2). The fact that pronounced geometric changes occurred as shown in Figure I.1 and that the final fracture takes place away from the temperature-controlled mid-point of the gauge length [21-22, 101-102], renders doubtful numerous already published results.

Criticisms of a Practical Nature

(1) Oxide growth on high temperature alloys is usually small but rapid oxide growth can initiate and produce 20-30 μm (greater than the strain being measured) of oxide in 500 hrs. [57, 105-106]. Because most oxides grow outward from the initial metal surface, lightly loaded probes of the type used in diametral extensometers are displaced by such growth. However, this displacement is interpreted as a compressive strain in the longitudinal axis of the specimen and the effect is equivalent to imposing a ratcheting strain on the system, since the servo-machine will impose a tensile mechanical strain to compensate for the diametral growth. Although examination of the specimen would show that this had occurred, it could cause the loss of a long and expensive test.

(2) Only a small volume of material is subjected to the chosen test conditions (Figure I.2). At failure, the final crack propagates through this material destroying the microstructure which had developed during the test. Consequently, metallographic examination and other measurements are difficult, thus reducing the value of the results and

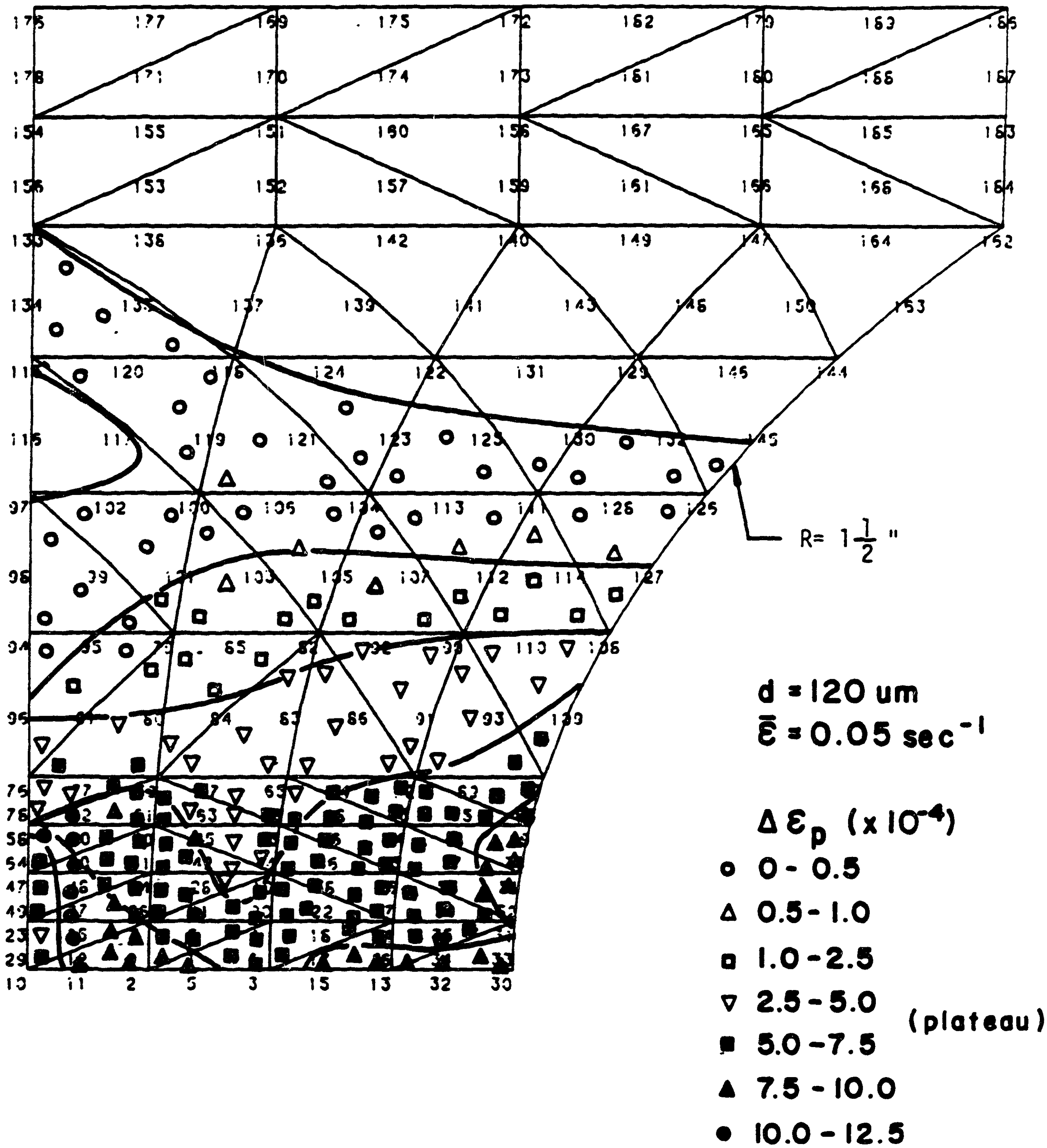


Figure I.2 Domain of iso-plastic strain in an hour-glass specimen of polycrystalline α-brass.

the testing program in developing life prediction models which may be used with confidence.

(3) The recording of crack growth data under strain-controlled conditions is nearly impossible since the crack initiates and grows in the zone where the knife-edges of the extensometer are in contact with the hour-glass specimen. There has been one attempt at measuring low cycle fatigue crack propagation rates under diametral strain-control using SEN specimens [107]. However, as in the case of hour-glass specimens, the relationship between the measured diametral strain and the axial strain was found to be non-linear under fully plastic conditions [77]. As seen before, although the measurement of diametral strain in conjunction with axial load affords in theory a method of measuring axial strain, caution is needed in interpretation of the results.

3B. Axial Strain Control

The obvious method to overcome the problems associated with the diametral strain control outlined above is to measure directly the changes of length of a defined gauge length. Axial strains are usually measured by attaching extensometer legs to knife edges machined onto the parallel section of the specimen, thereby defining the limits of the gauge length. The extensometer legs are positioned parallel to the stress axis and are extended beyond the furnace or RF coil [77, 99, 108] where these relative displacements are measured by LVDTs. This technique has proved convenient and successful on a range of materials which cyclically soften. However, certain materials, notably austenitic stainless steels [99, 109] and nickel-based alloys [111], are notch-sensitive at elevated temperatures. The presence of the built-up notches acts as a stress-raiser, localizing strain in a small volume underneath the knife edges [77], and failure often occurs at the base of the knife edges. Changes in the profile of the knife edge can alter the propensity for fatigue crack nucleation at the base of the knife edge, but it has not proved possible to prevent this behavior. However, several authors have described some alternative methods for measuring axial strain which minimized failure at the knife edges [91-92, 99, 109-111]. All these methods rely on the high coefficient of friction of metals at high temperatures to allow side arms to be placed in contact with the specimen gauge length under low loads without slipping.

We have seen that although diametral measurements can give a measure of axial strain, there are a number of features of the method which make it less desirable than the direct measurement of axial strain. However, in order to fully appreciate the difference between the stress-strain recorded in axial and diametral strain controlled tests, it is necessary to consider the nature of both the volume changes and the closed-loop feedback system.

It is now established that during creep-fatigue testing, the creep component of the test cycle usually produces intergranular cavitation [3, 5, 42, 48-50, 91, 112]. If cavities grow purely by diffusion mechanisms, the strain developed is only in a direction which allows the applied stress to do work. It is now well-known [113-114] that under low uniaxial stresses, strain accumulates in one direction only. Therefore, the nature of the feedback system causes a difference in relaxation behavior, according to the method of strain control employed. If we consider that relaxation arises from strain conversion due to the diffuse growth of cavities, then in the case of axial strain measurement the volume change will be detected as a change in length and stress will consequently be reduced. Reduced stress will cause the growth of cavities to decelerate and the drop in stress will be a true reflection of the creep strain accumulated. On the other hand, the diameter of the specimen, being orthogonal to the applied stress, will not change during the growth of cavities by diffusion. Because the stress is not reduced during the hold time (with no detectable change in strain), the cavities continue to grow at a rate determined by the peak stress but since no stress drop occurs, there is no method available for estimating the creep strain accumulated. However, even with the use of axial extensometers, interpretation of th stress relaxation behavior is difficult [31, 50], there are very few experimental studies devoted to the quantitative determination of the damage underlying this behavior [112, 115]. Detailed information on the microstructural process taking place is necessary in order to fully understand the relaxation behavior.

4. Fatigue Crack Growth Testing and Crack Length Measurement

The techniques and specimen geometries used for isothermal crack growth measurements at elevated temperature and under LEFM conditions are now well-established and numerous results have been obtained in the last ten years [2-4, 42, 48-50, 59, 89, 91]. Crack growth data in the high strain regime (HSF), i.e., under conditions of reversed gross plasticity in isothermal conditions, are relatively new [2, 59, 116]. Most HSF crack growth data have been obtained in push-pull testing [59] on smooth longitudinal LCF specimens. Crack propagation rates were evaluated, either by subsequent striation measurement [2-3, 5, 59, 116] or by introducing a small starter notch and monitoring the crack growth by the potential drop technique [2, 59]. Optical measurement of crack progress has also been made on SEN specimens [107], solid center notched specimens [93, 117], and tubular center notched specimens [27-28, 41, 93]. Hardly any data have been gathered on crack growth during thermal-fatigue cycling in the plastic strain range [22, 27-29, 41, 48] and crack length measurements have been performed mostly by optical measurement [27-29, 41, 48] and by cellulose acetate replica [27] on tubular centered notched specimens. In these experiments, the surface crack lengths must be converted to a mean crack length for the tube used [41]

$$a = R \sin^{-1} (2a_p/D_o)$$

where a = mean crack length, R = mean radius of the tube, D_o = outer diameter of the tube, and $2a_p$ = total projected length measured by microscope.

Also a curvature correction factor $G(a/t)$ from Erdogan and Ratwani [118] must be used for calculating ΔK or ΔK_ε, the strain intensity factor. It should be noted that $G(a/t)$ was derived assuming no bulk plasticity, and the validity of this correction factor for TMF crack growth data remains to be proven. The resolution claimed [41] for optical measurements is in the range of 20-120 μm, depending on the crack tip clarity and providing the measurements are taken at maximum tensile load. The slowest growth rates reported on tubular notched specimens have been on the order of 1×10^{-6} m/cycle.

There have been few attempts to measure crack length in TMF conditions using the potential drop technique [22, 119] and 12.7 mm diameter push-pull specimens. Using a starter notch 200 μm deep, experimenters have been able to resolve accurately a crack increment of 40 μm, and the corresponding growth rates were measured to better than 10^{-5} mm/cycle [119]. The method of potential drop has been proven satisfactory in isothermal conditions for numerous materials [120-121], and it has been shown (see Appendix II) that it can be used satisfactorily provided that the electrical noise is adequately filtered and that the calibration curve is properly corrected in order to take into account changes of potential with temperature.

Appendix II

Design of Test Unit

The apparatus which was built is a computer-controlled thermal fatigue testing system consisting primarily of a closed-loop servo-controlled electro-hydraulic tension compression fatigue machine (MTS 810 model 906.06), a high frequency oscillator for industrial heating (Lepel Model T-2.5-1-KO1-BW), an air compressor for cooling, a low frequency function generator (Exact model 504), a data acquisition and control system (HP-6942A), a programmable high resolution digital voltmeter (DVM Model 3478A), and a main frame computer (HP 9816S). Figure II.1 shows the control block diagram of this system. The system is capable of testing specimens of different size and configuration (SEN, CT, hollow tube, etc.) and to withstand a fatigue loading of 25,000 lbf (11,350 Kgf). The test specimen is mounted in a vertical plane into two end grips of high tensile steel. The specimen alignment is insured by the use of a Wood's metal pot. Because a DC potential drop technique is used to monitor crack growth, the lower grip is electrically insulated from the system by means of a ceramic coating. The ends of the end grips are water cooled by means of copper coils surrounding them.

1. The Demand-signal Generators

The demand-signal generation system provides separate demand signal for the control of specimen strain or applied stress and temperature. A programmable temperature controller is used to

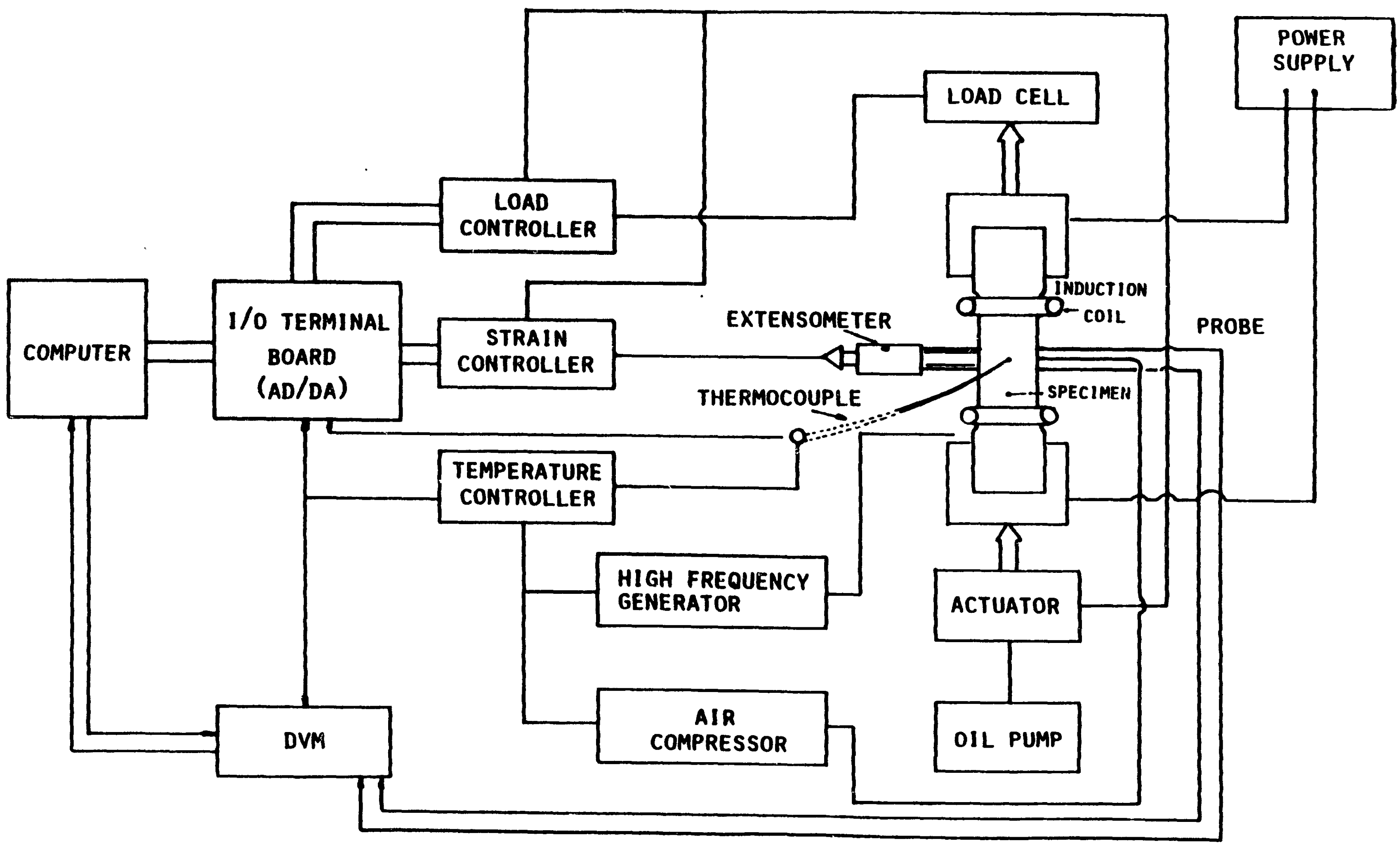


Figure II.1 Control block diagram of the testing system.

generate the driving signal to the high frequency generator, the DVM, the computer, the air valve and the frequency generator. The computer and the data acquisition and control system provide the command signals for the stress or strain controlled cycling. Proper coding and decoding of trigger signals from the temperature controller to the computer allow the system to generate any strain/stress/temperature cycling. Feedback signals from measuring points on the machine and specimen are compared continuously with the demand signals in the electronic control equipment. The difference between corresponding signals (i.e. the error) controls the power delivered to the hydraulic actuator or the heating circuit unit in such a way that the error in the test variable is reduced; careful choice of relative signal magnitudes (amplification) allows adequate control of each variable.

2. Control of Strain or Stress

Control of specimen strain is achieved by control of the displacement between two knife-edges of the strain detector attached to the specimen surface. The axial strain extensometer, machined in-house, consists of a strain gauge extensometer (SG) and a transmitter consisting of two shims which link together two sets of Invar arms, each supporting an alumina rod. The feedback signal from the SG is compared with the demand signal and the difference between the two, the error signal, is used to control the driving signal to the servovalve. The command signals (mean strain and amplitude) are sent by the Multiprogrammer (HP 6942A), which is controlled by the HP 9816S. Figure II.2 shows a block diagram of the strain controlled closed-loop

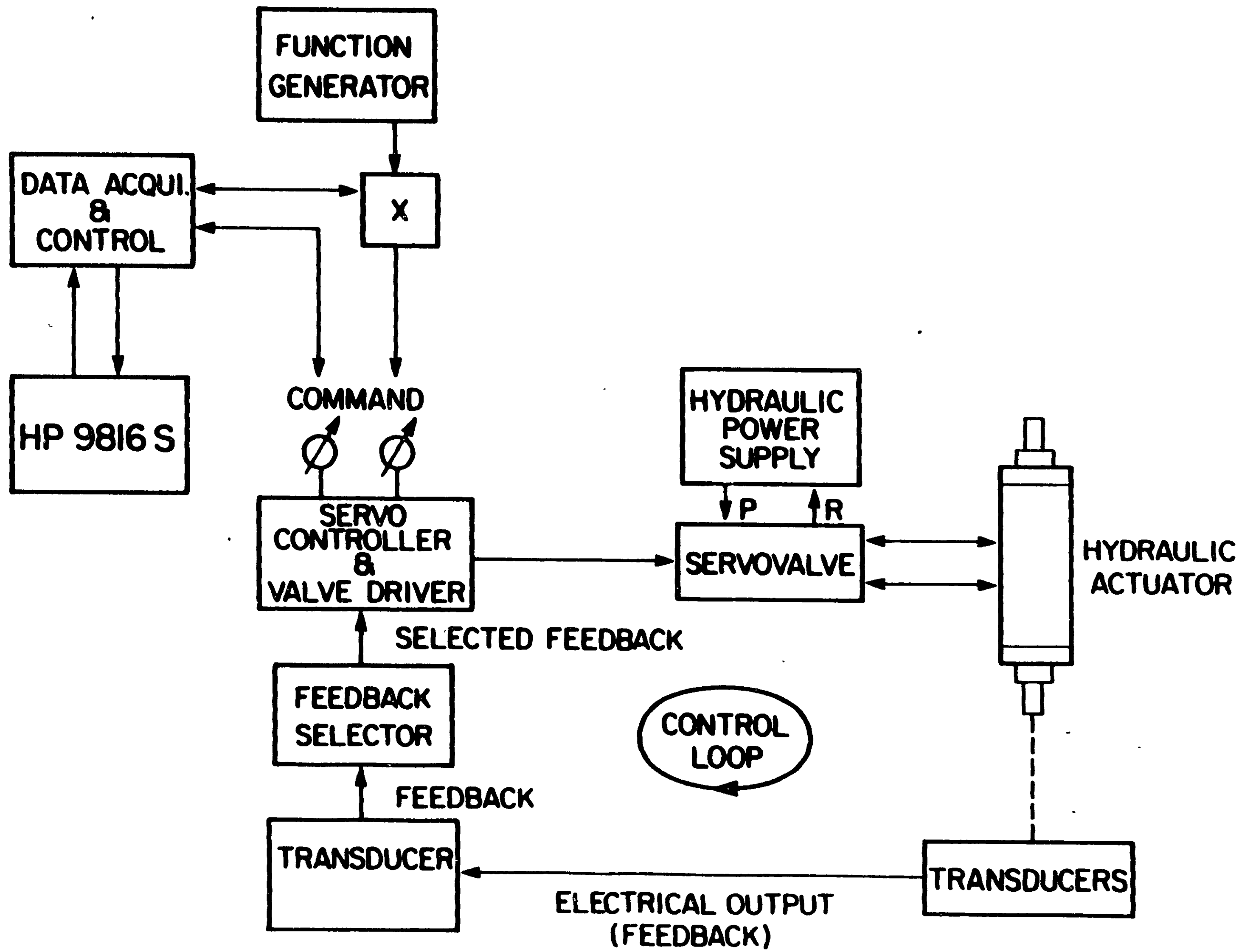


Figure II.2 Block diagram of the strain-control closed-loop unit

unit. In the system, the set point command (mean stress or strain) is fed directly to the servo-controller of the MTS unit. The amplitude signal command is also obtained by feeding the output of a D/A converter directly into the servo-controller. The procedure to continuously update the command is outlined in Section 3c.

The control of the stress applied is similar to the strain controlled except that the feedback signal is obtained from the load cell.

3. Data Acquisition and Control Unit

The command signals are sent by this data acquisition and control unit called Multiprogrammer or HP 6942A. It is composed of an internal main microprocessor, memory buffers for I/O and data storage, a real time clock and a backplane where separate plug-in cards are connected. Each plug-in I/O card is equipped with its own microprocessor.

Instructions to perform the test and values of the stress (in MPa) or strain (in %) for the mean and amplitude are typed-in on the HP 9816S at the beginning of the test. These values are then converted into their corresponding binary code and voltage and sent to the Multiprogrammer, which in turn decodes them and re-programs the appropriate I/O cards.

With the proper I/O cards, the system has the following capabilities:

1. Data acquisition
2. Measurement ranges
3. Control

4. Synthesis

3a. Acquisition

Analog measurement from up to 16 channels (can be increased to 960 channels) may be acquired, depending upon the scanner system configuration. Random access to any channel, as well as continuous scanning, are easily accomplished. Figure II.3 shows a diagram of the scanning system configuration implemented in our system. The signal may be digitized at rates up to 33 KHz by the A/D and stored on a memory card. Each memory card can store up to 64 Kbytes. The digitizing process takes place independently of all other Multiprogrammer activity, that is, data acquisition is achieved in parallel without any interruption from the control activities.

3b. Measurement Ranges

Proper switches on the A/D converters are used to select measure voltage range from ± 25mV to ± 10V full scale in the presence of 250V of common-mode voltage. The resolution can then be varied from 12 μV to 5 mV.

3c. Control and Synthesis

The twelve bits voltage D/A converters provide outputs for analog programmable instrument and stimulus of units under test (mean stress/strain and amplitude). A memory card can be used to

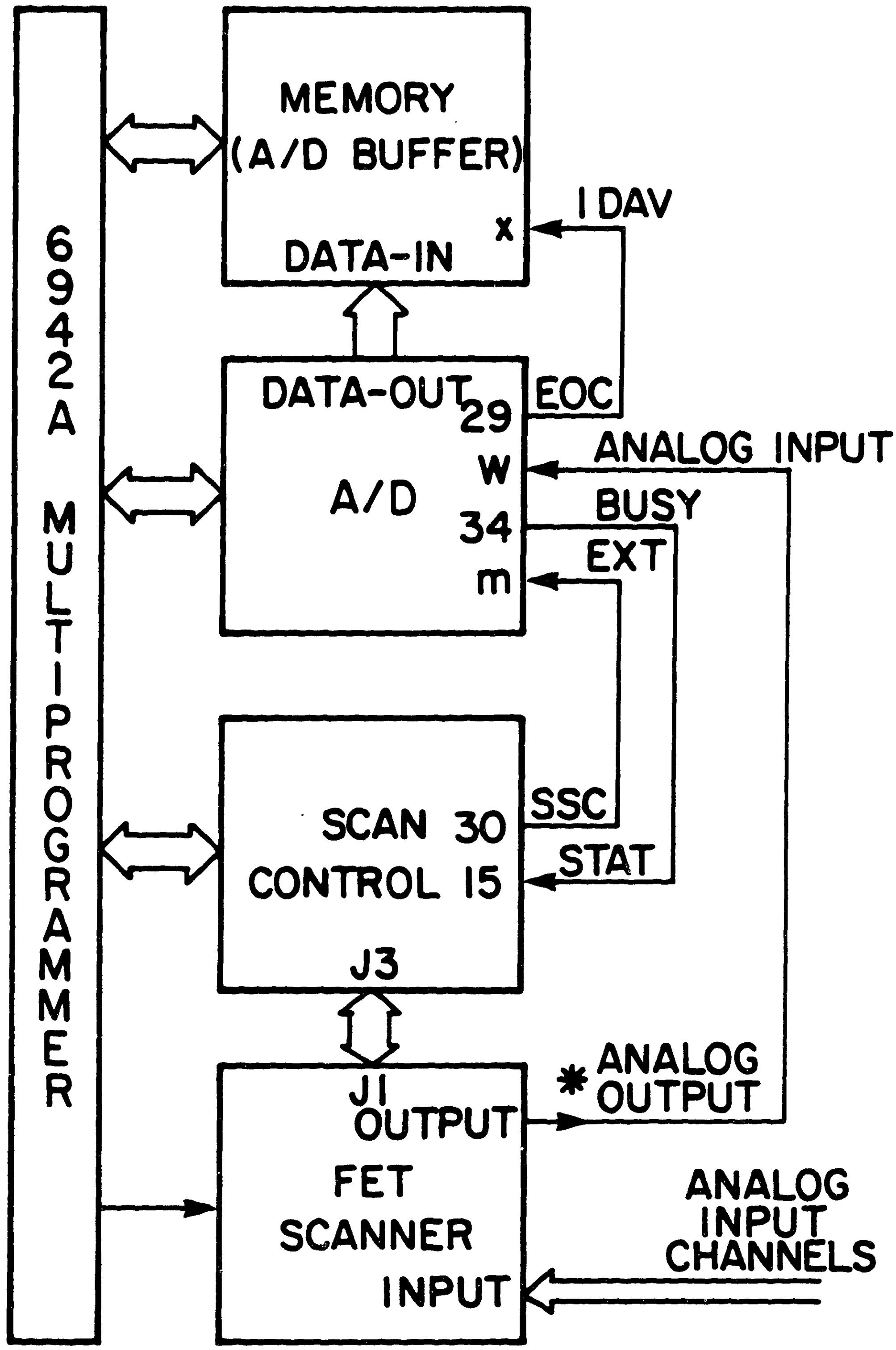


Figure II.3 Configuration for sequential scanning with buffered A/D continuously supply preloaded data to the D/A card at rates of up to 100 KHz. Any type of waveforms may be loaded into the memory card from the computer and used as stimuli for tests. Any repeatable wave shape (sine, square, triangular, etc.) can also be preloaded into a memory card. A timer card then determined the time between each analog voltage change such that the changes are perceived in the test as continuous as possible and not as a series of step changes.

4. Control of Temperature

The specimens are heated by a radio frequency unit. For TMF tests, low frequency heaters (10 KHz) are often used [28-29] because they produce lower wall thickness gradients than the higher frequency (450 KHz) heaters and because there are fewer ground-loop problems with the 10 KHz heaters than with the 450 KHz heaters. However, it is the present author's belief that high frequency induction heating is preferable because of its fast response, making it possible to provide high heating rates capable of inducing considerable thermal stresses in specimens of small cross-section. In this case, the heating mode is close to that encountered in-service, since the heat is generated within the surface layer of the specimen. Analysis of the temperature fields in the specimens needs to be carried out with the high frequency heaters, but numerical methods of non-stationary problems of heat conductivity using finite elements are now in our hands. Temperature measurement is accomplished by thermocouples welded onto the specimen. A thermistor is used with the temperature controller to provide a stabilized cold junction.

There is a relationship between the heating method, the design of the specimen, and temperature measurement and control. Radio frequency induction requires close coil-to-specimen spacing and small grip ends which makes difficult the mounting of additional measurement devices and obscures access to and visibility of the surface for replication, etc.; also, the coil has a high voltage RF potential requiring insulation and safety precaution. A temperature disturbance occurs at the thermocouple mass in the field, as well as at specimen locations spaced to the coil. Finally, there are possible RF induction-induced group loop problems in the instrumentation cabling.

In order to minimize these potential problems, great care was taken to use properly shielded cables. The coils were machined as to maintain constant the coil-to-specimen spacing. This was achieved by using a mold having over-sized dimensions of the specimen. Before each test, the coil must be carefully positioned around the specimen as to minimize the temperature difference between the four faces of the specimen. The temperature variation along the gauge length of the specimen has been measured to be less than 10°C and accurate to ±1°C.

The amplified signal from the thermocouple (Figure II.4) is compared with the demand signal which was programmed. A difference between the demand and feedback signals caused by the specimen temperature being too low, causes an increase in the power supplied to the coil. The increased power passing through the coil increases the temperature of the specimen and hence reduces the error. If the temperature is too high, the power to the coil is reduced. Compressed air is used to cool the specimen during the cool down portion of the temperature cycle. The air valve is best controlled by the

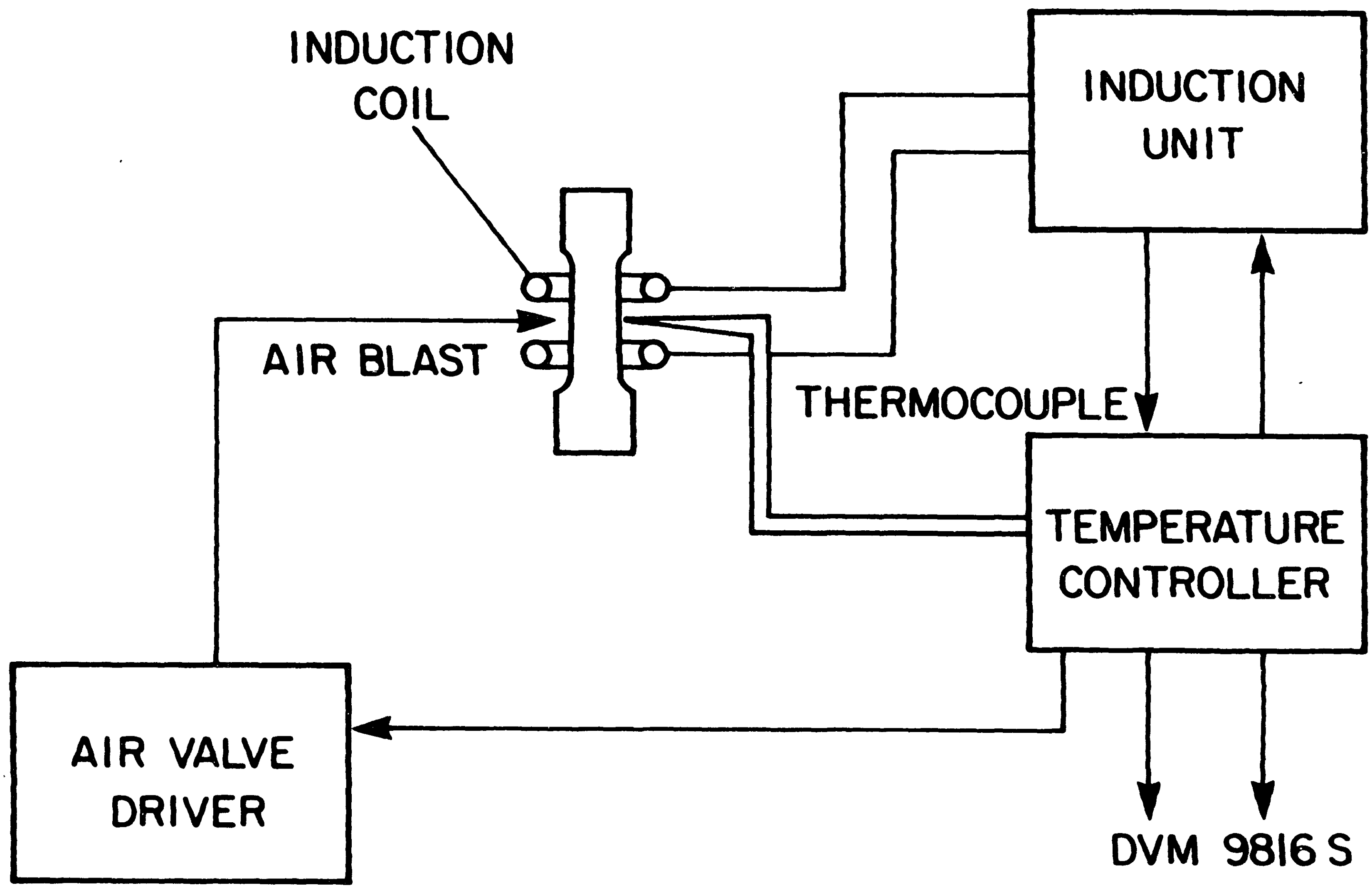


Figure II.4 Block diagram of the temperature control unit

There is a relationship between the heating method, the design of the specimen, and temperature measurement and control. Radio frequency induction requires close coil-to-specimen spacing and small grip ends which makes difficult the mounting of additional measurement devices and obscures access to and visibility of the surface for replication, etc.; also, the coil has a high voltage RF potential requiring insulation and safety precaution. A temperature disturbance occurs at the thermocouple mass in the field, as well as at specimen locations spaced to the coil. Finally, there are possible RF induction-induced group loop problems in the instrumentation cabling.

In order to minimize these potential problems, great care was taken to use properly shielded cables. The coils were machined as to maintain constant the coil-to-specimen spacing. This was achieved by using a mold having over-sized dimensions of the specimen. Before each test, the coil must be carefully positioned around the specimen as to minimize the temperature difference between the four faces of the specimen. The temperature variation along the gauge length of the specimen has been measured to be less than 10°C and accurate to ± 1°C.

The amplified signal from the thermocouple (Figure II.4) is compared with the demand signal which was programmed. A difference between the demand and feedback signals caused by the specimen temperature being too low, causes an increase in the power supplied to the coil. The increased power passing through the coil increases the temperature of the specimen and hence reduces the error. If the temperature is too high, the power to the coil is reduced. Compresses air is used to cool the specimen during the cool down portion of the temperature cycle. The air valve is best controlled by the temperature programmer, which opened or closed a solid state relay that drove the power to the air valve. With this technique, the air is always synchronized to the temperature cycle and is repeatable. The cooling rate is controlled by the heater ballasting the air blast cooling.

The limits of the system, in terms of heating and maximum temperature, were also tested. Heating rates in excess of 20°C/sec. have been obtained with the test rig and single edge notch specimen geometry. For faster rates, some control accurcy is sacrificed and a detailed thermal analysis has to be performed to determine the time-temperature distribution in the specimen. Using different specimen geometry and coil design, control heating rates in excess of 200°C/sec. were achieved with an identical heating/cooling unit. The maximum uniform temperature that can be reached is a function of the specimen geometry and specimen-to-coil distance. With the present configuration of specimen and coil design, the upper limit of the temperature cycles is 1050°C. In the development work, specimens were tested between 400 and 925°C. A six degree per second heating and cooling ramp was programmed and run. The dynamic performance of the system is shown in Figure II.5. Five thermocouples were used and the temperature of each thermocouple was always within ten degrees of the requested profile throughout the test.

5. Crack Growth Measurement

Hardly any data have been gathered on crack growth during thermal fatigue cycling in the plastic strain range and crack length measurements have been performed mostly by optical measurements

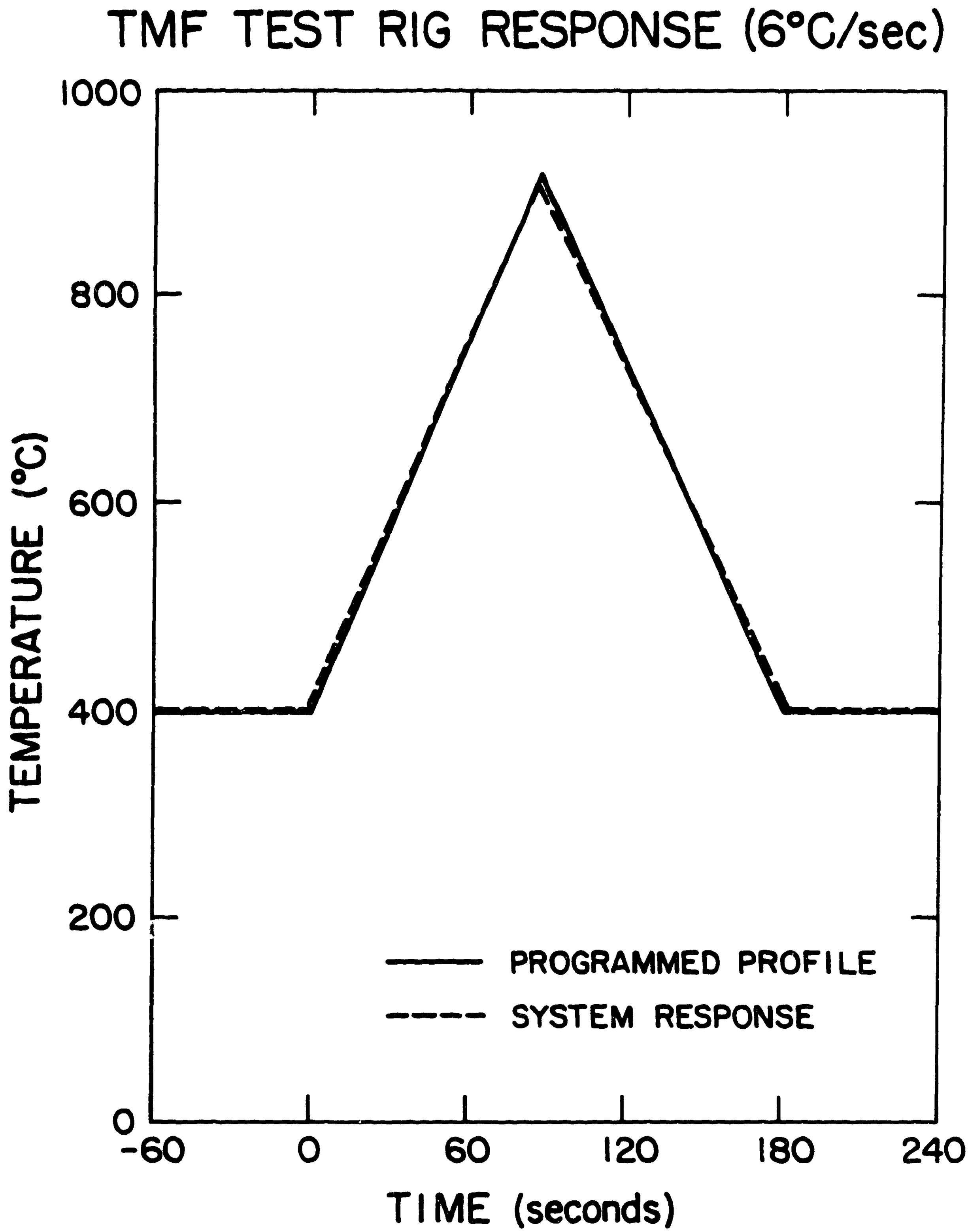


Figure II.5 Programmed profile and system response for a 6 °C/sec triangular waveform.

and by cellulose acetate replica on tubular centered notched specimens. In these experiments, the surface crack lengths must be converted to a mean crack length for the tube used (see Appendix I). The resolution claimed [41] for optical measurements is in the range of 40-120 μm, depending on the crack tip clarity and providing the measurements are taken at maximum tensile load.

There have been few attempts to measure crack length in TMF conditions using the potential drop technique [22, 119] on a 12.7 mm diameter push-pull specimen. Using a starter notch 200 μm deep, experimenters have been able to resolve accurately a crack increment of 40 μm, and the corresponding growth rates were measured to better than 10^{-8} m/cycle [119]. The method of potential drop has been used satisfactorily for isothermal conditions for numerous materials [120-121] and preliminary experiments by the authors have shown that it can be used satisfactorily provided that the electrical noises are adequately filtered and that the calibration curve is properly corrected in order to take into account changes of potential with temperature.

The electrical noises induced in the potential probes by RF heating can be eliminated by the use of a band-pass filter centered on a 430-460 KHz frequency range. Power line surges, transients, and RF induced power-line noises can be taken care of by the following technique. First, a 10 KHz band-pass filter is used to eliminate the induced-noise of the 10 KHz Master oscillator of the MTS console usd to provide excitation to the servo-valves LVDT's. Second, the DVM is programmed to convert the analog signal in 10 power line cycles (PLC). In this mode, 1 PLC is used for the runup time, the A/D operation repeated ten times. The resulting ten readings are then averaged and the answer becomes a single reading. This, with the built-in broad band filter, greatly reduces the emf noise induced by the power-line fluctuations. A schematic of the DC potential system is shown in Figure II.6. Because the potential, the temperature, the load, and the strain are recorded synchronously, it is possible to take into account the thermal variation of the potential. The procedure is the following. First, the changes in potential as a function of the temperature and strain or stress ($V(T,\sigma)$) is measured. Then a thermal cycle at zero stress is recorded ($V(T,\sigma = 0)$). The effect of the applied stress or strain is obtained by subtracting $V(T,\sigma=0)$ from $V(T,\sigma)$) to yield $V(\sigma)$. The procedure is outlined in Figure II.7. To assess crack growth, the peak potential of each cycle ($V(T,\sigma)$) is recorded, and plots of the voltage versus applied number of cycles are obtained. Using well-known solutions [120-121] relating (V/Vo) to (a/ao) for the single-edge notch specimen geometry, the crack lengths versus N curves are derived. Improved sensitivity can be achieved through the use of reference probes where the ratio ($V(t)/Vo$) is corrected by multiplying by (V^{o}_{ref}/V_{ref}), the ratio of the initial reference probe signal to the reference probe signal at time t. The resolution achieved under TMF conditions on single edge notch specimens is better than 10 μm.

A close look at the potential signals shown in Figure II.7 indicates that they can be used to generate information on the mechanisms taking place at the crack tip. By noting that $V(T,\sigma)$ and $V(T,\sigma = 0)$ are also functions of the geometry, we can write

$$V(\sigma) = V(T, \sigma. \text{geo}) - V(T, \sigma = 0, \text{geo})$$

Therefore, for a given temperature T and zero applied stress, the expected value of $V(\sigma)$ is zero is the geometry of the crack, which

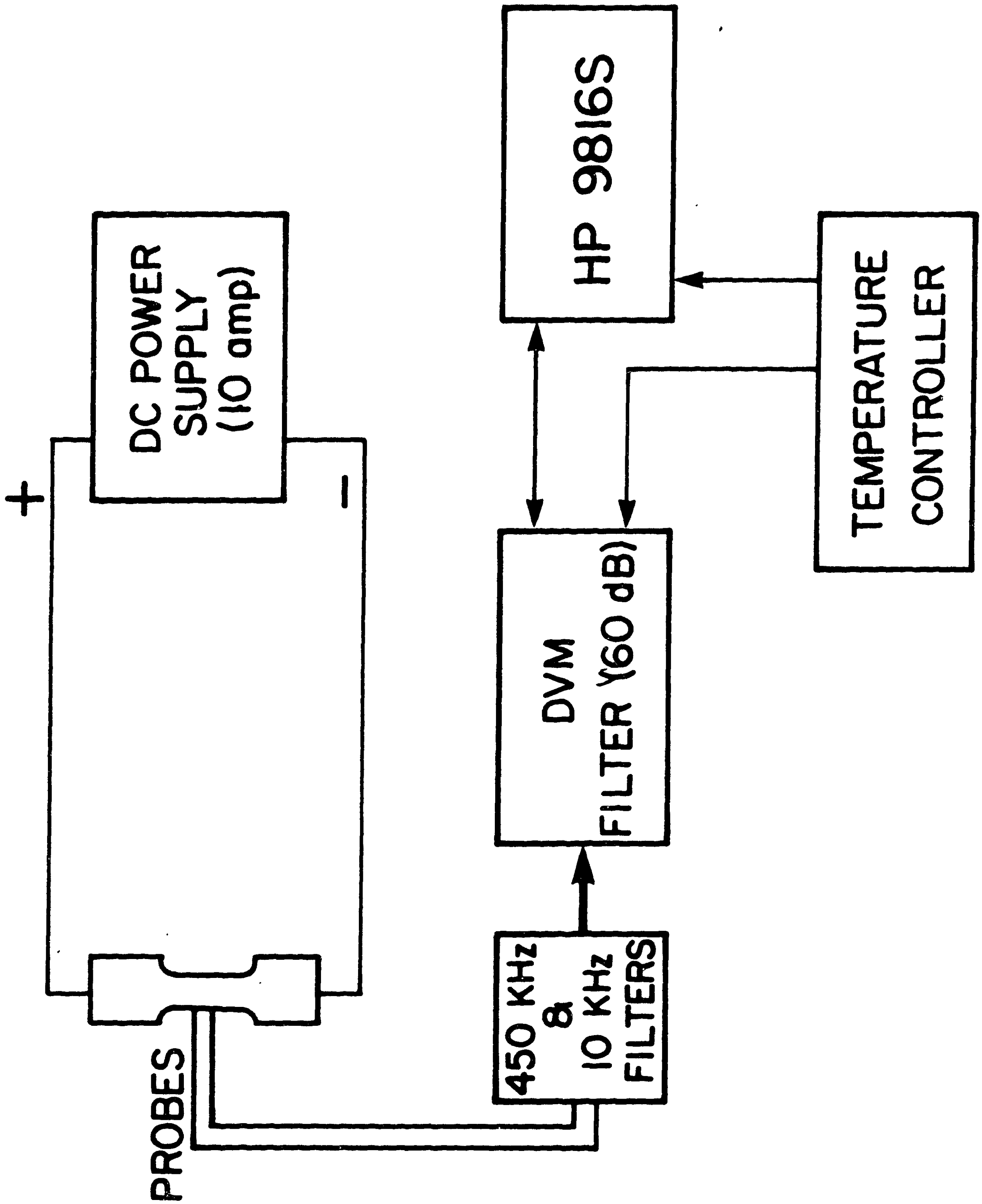
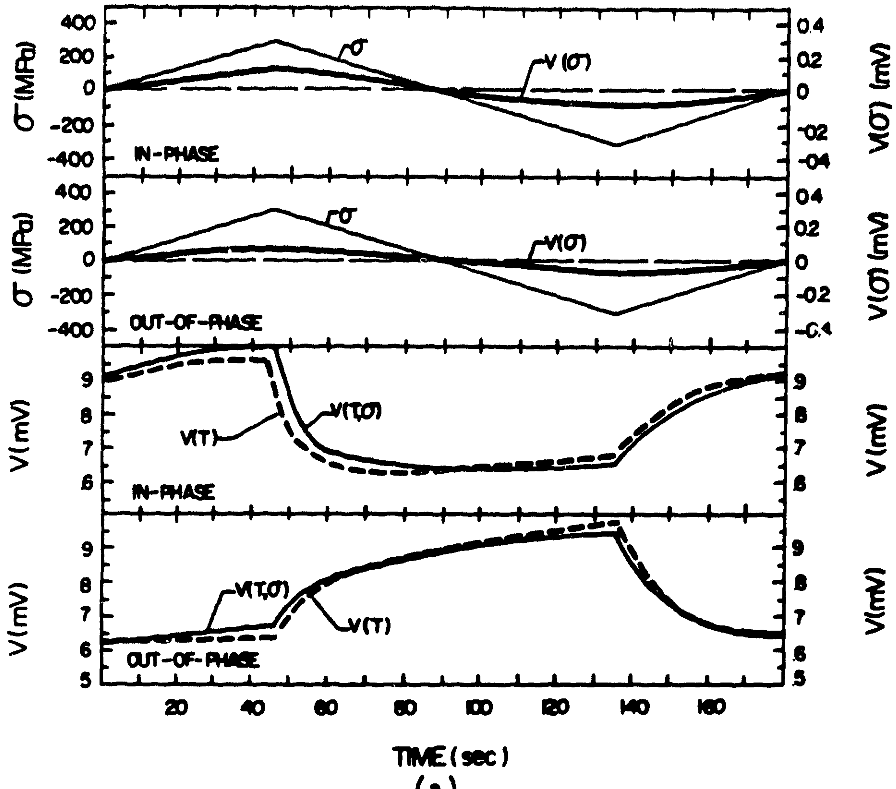


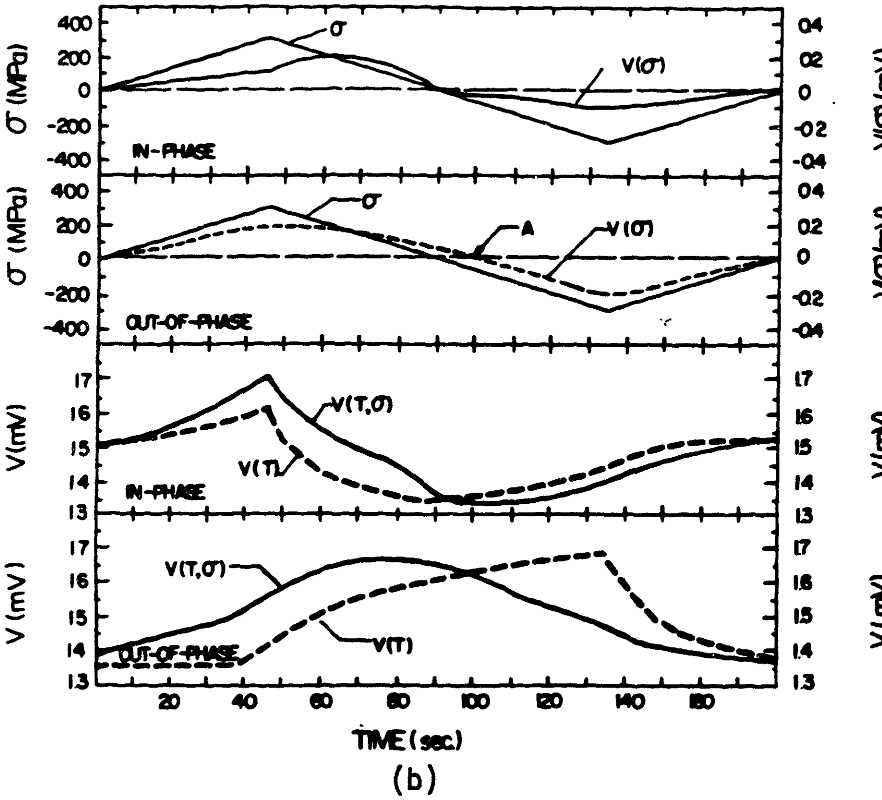
Figure II.6 Schematic of the DC potential drop system.

include crack length and configuration, in unchanged. However, if $V(\sigma)$ is not equal to zero at zero applied stress, then the geometry terms in $V(T, \sigma, geo)$ and $V(T, \sigma = 0, geo)$ must be different (see Equation 2). Assuming that the crack increment during the thermal cycle is negligible, then the difference associated with the geometry term is solely a function of the configuration. The applied stress required (see Figure II.7b) to give $V(\sigma) = 0$ represents the minimum necessary stress to re-establish in the specimen the same crack configuration. Because the transition point where $V(\sigma)$ equal zero corresponds to the transition between the tensile and compressive part of the cycle, it is logical to assume that this change in configuration is a change in the opening of the crack.

From Figure II.7 other information can be derived from the analysis of the $V(\sigma)$ potential curves. That is, to determine when the crack is growing within one cycle. In Figure II.7b, the $V(\sigma)$ curve for in-phase cycling shows a hump near the maximum stress, whereas at low ΔK (Figure II.7a) and for out-of-phase cycling, the $V(\sigma)$ curve is continuous without discontinuity. Such an increase in potential usually reflects a sudden increase in crack length. Therefore, it can be concluded that under this particular condition and specific loading, the crack growth was happening at the peak applied load/strain. On the other hand, for the other conditions shown in Figure II.7, it is logical to assume that the crack was growing in a continuous fashion in the tensile going part of the cycle ($0 < \sigma < \sigma_{max}$). The crack extension during a single cycle can be estimated by comparing the values of $V(T,\sigma)$ at the beginning and at the end of the cycle. Local estimate of the crack growth rate within one cycle can be computed by dividing the



(a)



(b)

Figure II.7 Potential curves $V(T,\sigma)$, $V(T)$ and $V(\sigma)$. (a) ΔK= 25 MPa$\sqrt{m}$
(b) ΔK= 50 MPa$\sqrt{m}$

increment in crack length by the time spent in the tension going part of the cycle (time interval for which $V(\sigma) > 0$). If a hump is seen on the $V(\sigma)$ the time interval between the hump and $V(\sigma) = 0$ must be taken.

From the above analysis of the $V(\sigma)$ curves, it can be concluded that the potential drop method has the capability of monitoring crack extension during a cycle, whereas the optical measurement and compliance methods are limited for growth measurement to no less than one cycle. Therefore, detailed analysis of the crack growth process can be achieved and this is particularly important in trying to determine the mechanisms involved with TMF crack growth.

Appendix III

Software for Control and Data Analysis

The software developed to run the isothermal and TMF tests are presented in this appendix. The programs used for data reduction and for plotting the data are also presented.

```
10!   ***************************************************************************
20!   *                                                                         *
30!   *                          PROGRAM ISO/TMF                                *
40!   *                          ***************                                *
50!   *                                                                         *
60!   *                                                                         *
70!   *      THIS PROGRAM PERFORM THE REAL-TIME CONTROL AND DATA ACQUI-         *
80!   *      SITION OF ISOTHERMAL OR THERMAL-MECHANICAL FATIGUE TEST.           *
90 !  *                                                                         *
100!  *      THE OPTIONS (Iso, TMF, CG, LCF) AND THE TEST PARAMETERS            *
110!  *      (Load and strain scale, Mean and amplitude, and frequency)         *
120!  *      ARE ENTERED BY THE USER.                                           *
130!  *                                                                         *
140!  *      THE MULTI. IS PROGRAMMED ACCORDINGLY AND DISPLAYED IN REAL-        *
150!  *      TIME THE HYSTERISIS LOOP AND POTENTIAL.                            *
160!  *                                                                         *
170!  *                                                                         *
180!  ***************************************************************************
190 !
200 !_______________SELECT OPTIONS AND READ PARAMETERS________________________
210 !               ---------------------------------
220 LOADSUB ALL FROM "READ_PARAM"
230 CALL Read_param(Material$,Mode$,Contr_variable,Scale,Mean,Span,Gauge,Frequen
cy)
240 DELSUB Read_param
250 !
260 IF Mode$="TMLCF" THEN Option_tmlcf
270 IF Mode$="TMFCG" THEN Option_tmfcg
280 IF Mode$="ISOLCF" THEN Option_isolcf
290 IF Mode$="ISOCG" THEN Option_isocg
300 !
310 GOTO Finish
320 !
330 !_______________________________________________________________________
340 Option_tmlcf:  !
350 !              OPTION THERMAL-MECHANICAL LOW CYCLE FATIGUE
360 !              -------------------------------------------
370       LOADSUB ALL FROM "TMLCF"
380       Tmlcf(Material$,Contr_variable,Scale,Mean,Span,Gauge)
390       GOTO Finish
400 !_______________________________________________________________________
410 Option_tmfcg:  !
420 !             OPTION THERMAL-MECHANICAL FATIGUE CRACK GROWTH
430 !             ---------------------------------------------
440       LOADSUB ALL FROM "TMFCG"
450       Tmfcg(Material$,Contr_variable,Scale,Mean,Span,Gauge)
460       GOTO Finish
470 !_______________________________________________________________________
480 Option_isolcf: !
490 !              OPTION ISOTHERMAL LOW CYCLE FATIGUE
500 !              -----------------------------------
510       LOADSUB ALL FROM "ISOLCF"
520       Isolcf(Material$,Contr_variable,Scale,Mean,Span,Gauge,Frequency)
530       GOTO Finish
540 !_______________________________________________________________________
```

```
550 Option_isocg:  !
560 !             OPTION ISOTHERMAL FATIGUE CRACK GROWTH
570 !             --------------------------------------
580         LOADSUB ALL FROM "ISOCG"
590         Isocg(Material$,Contr_variable,Scale,Mean,Span,Gauge,Frequency)
600         GOTO Finish
610 !_______________________________________________________________________
620 Finish:     !
630             END
640 !_______________________________________________________________________
```

```
10!    **********************************************************************
20!    *                                                                    *
30!    *                    PROGRAM READ_PARAM                              *
40!    *                    ******************                              *
50!    *                                                                    *
60!    *     THIS PROGRAM IS USED BY ISO_TMF TO READ THE OPTIONS AND        *
70!    *     PARAMETERS REQUIRE TO RUN THE DESIRED TEST.                    *
80!    *                                                                    *
90!    *     THE OPTIONS ARE:      1. ISOTHERMAL LCF (ISOLCF)               *
100!   *                           2. ISOTHERMAL CG (ISOCG)                 *
110!   *                           3. THERMAL-MECHANICAL LCF (TMLCF)        *
120!   *                           4. THERMAL-MECHANICAL FCG (TMFCG)        *
130!   *                                                                    *
140!   *     THE INPUTTED PARAMETERS ARE:                                   *
150!   *                           1. ALLOY DESIGNATION                     *
160!   *                           2. LOAD AND STRAIN SCALES                *
170!   *                           3. CONTROL VARIABLE (Stress or Strain)   *
180!   *                           4. GAUGE LENGTH                          *
190!   *                           5. MEAN (Mean stress or strain           *
200!   *                           6. AMPLITUDE (Stress or strain)          *
210!   *                           7. FREQUENCY                             *
220!   *                                                                    *
230!   *     THE PROGRAM ALSO CHECKS THE VALIDITY OF THE DATA.              *
240!   *                                                                    *
250!   **********************************************************************
260    !
270    OPTION BASE 0
280    CALL Read_param(Material$,Mode$,Contr_variable,Scale,Mean,Span,Gauge,Frequ
ency)
290    PRINT USING "2/,23X,K";"Program READ_PARAM terminated"
300    END
310    !
320    SUB Read_param(Material$,Mode$,Contr_variable,Scale,Mean,Span,Gauge,Freque
ncy)
330    OPTION BASE 0
340    DIM Dummy$[40],Dummy1$[80]
350    Dummy$="****************************************"
360    Dummy1$="*****************************************************************
***************"
370    !
380!-------------------------------------------------------------------------
390    PRINT USING 460;Dummy1$
400    PRINT USING 410
410    IMAGE 2/,20X,"ENTER TYPE OF ALLOY (10 Characters max)"
420    BEEP
430    INPUT Material$
440!-------------------------------------------------------------------------
450    PRINT USING 460;Dummy1$
460    IMAGE 2/,80A
470    PRINT USING 480
480    IMAGE 2/,20X,"ENTER TYPE OF TESTING",/,25X,"1. ISOTHERMAL FATIGUE",/,25X,"
2. THERMAL-MECHANICAL FATIGUE"
490    BEEP
500    INPUT Mode_type
510    IF Mode_type<>1 AND Mode_type<>2 THEN
```

```
520        PRINT USING 530
530        IMAGE 2/,20X,"INCORRECT MODE TYPE. Please re-enter"
540        BEEP 3000,1.0
550        GOTO 470
560     ELSE
570     END IF
580     !-----------------------------------------------------------------------
590     PRINT USING 460;Dummy1$
600     PRINT USING 610
610     IMAGE 2/,20X,"ENTER MODE OF TESTING",/,25X,"1. LCF (Initiation)",/,25X,"2.
 CRACK PROPAGATION (Propagation)"
620     BEEP
630     INPUT Mode_of_testing
640     IF Mode_of_testing<>1 AND Mode_of_testing<>2 THEN
650        PRINT USING 660
660        IMAGE 2/,20X,"UNDEFINED MODE OF TESTING. Please re-enter"
670        BEEP 3000,1.0
680        GOTO 600
690     ELSE
700     END IF
710     !-----------------------------------------------------------------------
720     IF Mode_type=1 THEN
730        Mode$="ISOCG"
740        IF Mode_of_testing=1 THEN Mode$="ISOLCF"
750     ELSE
760        Mode$="TMFCG"
770        IF Mode_of_testing=1 THEN Mode$="TMLCF"
780     END IF
790        PRINT USING 460;Dummy1$
800        PRINT USING 810;Dummy$,Mode$
810        IMAGE 2/,15X,40A,/,15X,"*",38X,"*",/,15X,"*",7X,6A," MODE WAS SELECTED.
",6X,"*"
820        IF Mode$="ISOLCF" OR Mode$="ISOCG" THEN
830           PRINT USING 840;Dummy$
840           IMAGE 15X,"*",7X,"NO DEFAULT VALUE.",14X,"*",/,15X,"*",38X,"*",/,15X
,"*",38X,"*",/,15X,40A
850        ELSE
860           PRINT USING 870
870           IMAGE 15X,"*",7X,"DEFAULT VALUES ARE:",12X,"*",/,15X,"*",12X,"A.  3
min/cycle",11X,"*"
880           PRINT USING 890;Dummy$
890           IMAGE 15X,"*",12X,"B.  3 or 4 Thermo.",8X,"*",/,15X,"*",38X,"*",/,15
X,40A
900        END IF
910     !-----------------------------------------------------------------------
920     PRINT USING 460;Dummy1$
930     PRINT USING 940
940     IMAGE 2/,20X,"ENTER CONTROL VARIABLE?",/,25X,"1. LOAD CONTROL",/,25X,"2. S
TRAIN CONTROL"
950     BEEP
960     INPUT Contr_variable
970     IF Contr_variable<>1 AND Contr_variable<>2 THEN
980        PRINT USING 990
990        IMAGE 2/,20X,"UNDEFINED CONTROL VARIABLE. Please re-enter"
1000       BEEP 3000,1.0
```

```
1000      BEEP 3000,1.0
1010      GOTO 930
1020  ELSE
1030  END IF
1040  !_____________________________________________________________________
1050  IF Contr_variable=1 THEN
1060       PRINT USING 460;Dummy1$
1070       PRINT USING 1080
1080       IMAGE 2/,20X,"ENTER LOAD SCALE? (100,50,20 or 10 %)"
1090       BEEP
1100       INPUT Load_scale
1110       Load_scale=Load_scale
1120       IF Load_scale<>100 AND Load_scale<>50 AND Load_scale<>20 AND Load_scal
e<>10 THEN
1130           PRINT USING 1140
1140           IMAGE 2/,20X,"Undefined value. Please re-enter."
1150           BEEP 3000,1.0
1160           GOTO 1070
1170       ELSE
1180       END IF
1190  !_____________________________________________________________________
1200    PRINT USING 460;Dummy1$
1210    PRINT USING 1220
1220    IMAGE 2/,20X, "ENTER MEAN STRESS (in MPa)?"
1230    BEEP
1240    INPUT Mean
1250    IF (Mean*11.569896)<=-(250*Load_scale) OR Mean>=(250*Load_scale) THEN
1260        PRINT USING 1270
1270        IMAGE 2/,20X,"VALUE OUT-OF-BOUND. Please re-enter."
1280        BEEP 3000,1.0
1290        GOTO 1210
1300    ELSE
1310    END IF
1320  !_____________________________________________________________________
1330      PRINT USING 460;Dummy1$
1340      PRINT USING 1350
1350      IMAGE 2/,20X, "ENTER STRESS AMPLITUDE (in MPa)"
1360      BEEP
1370      INPUT Span
1380      IF Span<=0 THEN
1390          PRINT USING 1400
1400          IMAGE 2/,20X,"UNDEFINED VALUE. Please re-enter"
1410          BEEP 3000,1.0
1420          GOTO 1340
1430      ELSE
1440          IF (ABS(Mean)+Span)*11.569896>=250*Load_scale THEN
1450              PRINT USING 1460
1460              IMAGE 2/,20X,"VALUE OUT-OF-BOUND,  Please re-enter"
1470              BEEP 3000,1.0
1480              GOTO 1340
1490          ELSE
1500          END IF
1510      END IF
1520  !_____________________________________________________________________
1530      PRINT USING 460;Dummy1$
1540      PRINT USING 1550
```

```
2070       END IF
2080 !------------------------------------------------------------------------
2090       PRINT USING 460;Dummy1$
2100       Emax=.05493*Strain_scale/Gauge_lenght
2110       PRINT USING 2120
2120       IMAGE 2/,20X,"ENTER MEAN STRAIN (in %)"
2130       BEEP
2140       INPUT Mean
2150       Mean=Mean/100
2160       IF Mean<=-Emax OR Mean>=Emax THEN
2170          PRINT USING 2180
2180          IMAGE 2/,20X,"VALUE OUT-OF-BOUND. Please re-enter"
2190          BEEP 3000,1.0
2200          GOTO 2110
2210       ELSE
2220       END IF
2230 !------------------------------------------------------------------------
2240          PRINT USING 460;Dummy1$
2250          PRINT USING 2260
2260          IMAGE 2/,20X,"ENTER STRAIN AMPLITUDE (in %)"
2270          BEEP
2280          INPUT Span
2290          Span=Span/100
2300          IF Span<=0 THEN
2310             PRINT USING 2320
2320             IMAGE 2/,20X,"UNDEFINED VALUE.  Please re-enter."
2330             BEEP 3000,1.0
2340             GOTO 2250
2350          ELSE
2360             IF (ABS(Mean)+Span)>=Emax THEN
2370                PRINT USING 2380
2380                IMAGE 2/,20X,"VALUE OUT-OF-BOUND. Please re-enter."
2390                BEEP 3000,1.0
2400                GOTO 2110
2410             ELSE
2420             END IF
2430          END IF
2440  !-----------------------------------------------------------------------
2450          PRINT USING 460;Dummy1$
2460          PRINT USING 2470
2470          IMAGE 2/,20X,"ENTER LOAD SCALE (100,50,20 or 10%)"
2480          BEEP
2490          INPUT Load_scale
2500          Load_scale=Load_scale
2510          IF Load_scale<>100 AND Load_scale<>50 AND Load_scale<>20 AND Load_s
cale<>10 THEN
2520             PRINT USING 2530
2530             IMAGE 2/,20X,"UNDEFINED VALUE. Please re-enter"
2540             BEEP 3000,1.0
2550             GOTO 2460
2560          ELSE
2570          END IF
2580  !-----------------------------------------------------------------------
2590  END IF
2600  !-----------------------------------------------------------------------
```

```
1550      IMAGE 2/,20X,"ENTER THE GAUGE LENGTH (in inch)"
1560      BEEP
1570      INPUT Gauge_length
1580      IF Gauge_length<.400 OR Gauge_length>.6 THEN
1590         PRINT USING 1600
1600         IMAGE 2/,20X,"GAUGE LENGHT VALUE OUT-OF-BOUND. Please re-enter."
1610         BEEP 3000,1.0
1620         GOTO 1540
1630      ELSE
1640      END IF
1650  !____________________________________________________________________
1660      PRINT USING 460;Dummy1$
1670      PRINT USING 1680
1680      IMAGE 2/,20X,"ENTER STRAIN SCALE (100,50,20 or 10 %)"
1690      BEEP
1700      INPUT Strain_scale
1710      Strain_scale=Strain_scale/100
1720      IF Strain_scale<>1. AND Strain_scale<>.5 AND Strain_scale<>.2 AND Stra
in_scale<>.1 THEN
1730        PRINT USING 1740
1740        IMAGE 2/,20X,"UNDEFINED VALUE. Please re-enter."
1750        BEEP 3000,1.0
1760        GOTO 1670
1770      ELSE
1780      END IF
1790 !---------------------------------------------------------------------
1800   ELSE
1810 !_____________________________________________________________________
1820      PRINT USING 460;Dummy1$
1830      PRINT USING 1840
1840      IMAGE 2/,20X,"ENTER STRAIN SCALE (100,50,20 or 10 %)"
1850      BEEP
1860      INPUT Strain_scale
1870      Strain_scale=Strain_scale/100
1880      IF Strain_scale<>1. AND Strain_scale<>.5 AND Strain_scale<>.2 AND Stra
in_scale<>.1 THEN
1890         PRINT USING 1900
1900         IMAGE 2/,20X, "Undefined value.  Please re-enter"
1910         BEEP 3000,1.0
1920         GOTO 1830
1930      ELSE
1940      END IF
1950 !_____________________________________________________________________
1960      PRINT USING 460;Dummy1$
1970      PRINT USING 1980
1980      IMAGE 2/,20X,"ENTER THE GAUGE LENGHT (in inch)"
1990      BEEP
2000      INPUT Gauge_lenght
2010      IF Gauge_lenght<.4 OR Gauge_lenght>.6 THEN
2020         PRINT USING 2030
2030         IMAGE 2/,20X,"GAUGE LENGHT VALUE OUT-OF-BOUND. Please re-enter."
2040         BEEP 3000,1.0
2050         GOTO 1970
2060      ELSE
```

```
2610        IF Mode_type=1 THEN
2620           PRINT USING 460;Dummy1$
2630           PRINT USING 2640
2640           IMAGE 2/,20X,"ENTER THE FREQUENCY (in Hz)"
2650           BEEP
2660           INPUT Frequency
2670           PRINT USING 2680;Frequency
2680           IMAGE 2/,25X,"INPUTTED Frequency IS ",K," Hz",/
2690           PRINT USING 2700
2700           IMAGE 30X,"1.  CONTINUE",/,30X,"2.  CHANGE THE FREQUENCY"
2710           BEEP
2720           INPUT Change
2730           IF Change<>1 AND Change<>2 THEN
2740              PRINT USING 2750
2750              IMAGE /,30X,"Incorrect value. Please re-enter"
2760              BEEP 3000,1.0
2770              GOTO 2690
2780           END IF
2790           IF Change=2 THEN GOTO 2630
2800           GOTO Continue
2810        END IF
2820   !
2830 Continue: !
2840   Scale=Load_scale+Strain_scale
2850   Gauge=Gauge_lenght
2860   PRINT USING 460;Dummy1$
2870   BEEP 800,2
2880   PRINT USING "/,20X,K";"** PROGRAM THE Micro_Data Track **"
2890   !
2900   SUBEND
```

```
10!    *******************************************************************
20!    *                                                                 *
30!    *                      PROGRAM TMLCF                              *
40!    *                      *************                              *
50!    *                                                                 *
60!    *      THIS PROGRAM PERFORMS THE REAL-TIME CONTROL AND DATA       *
70!    *      ACQUISITION OF THERMAL-MECHANICAL LOW CYCLE FATIGUE        *
80!    *      TESTS.                                                     *
90!    *                                                                 *
100!   *      THIS PROGRAM IS CALLED BY ISO_TMF AFTER SELECTION OF THE   *
110!   *      TMFLCF OPTION.                                             *
120!   *                                                                 *
130!   *      THE MICRO_DATA_TRACK MUST BE PROGRAMMED FOR 180 sec        *
140!   *      PERIOD (1/3 cpm).                                          *
150!   *                                                                 *
160!   *      THE CYCLIC THERMAL STRAINS ARE FIRST MEASURED BY CYCLING   *
170!   *      THE TEMPERATURE AT ZERO STRESS.                            *
180!   *                                                                 *
190!   *      THE CYCLIC DATA(Stresses, strains, and temperatures)       *
200!   *      ARE RECORDED EVERY (N2/N1)=1.25 CYCLES.                    *
210!   *                                                                 *
220!   *******************************************************************
230    !
240    OPTION BASE 0
250    CALL Tmlcf(Material$,Contr_variable,Scale,Mean,Span,Gauge)
260    END
270!*********************************************************************
280!*********************************************************************
290    !
300    SUB Tmlcf(Material$,Contr_variable,Scale,Mean,Span,Gauge)
310    DIM Load(179),Strain(179),Temp1(179),Temp2(179),Temp3(179)
320    MASS STORAGE IS ":HP8290X,700,1"
330    !
340    !______ ______________DEFINE I/O CARD FUNCTION________________________
350    !                     -----------------------
360    ASSIGN @Dvm TO 720
370    ASSIGN @Multi TO 723
380    ASSIGN @Disc TO "MRZ"
390    Disc_drive=0
400    Dvm=720
410    Multi=723
420    Scanner=0
430    A_to_d=3
440    Memory1=7
450    D_to_a1=10
460    Timer=11
470    D_to_a2=12
480    Memory2=13
490    !
500    !________________DEFAULT VALUES FOR THE PARAMETERS___________________
510    !                ------------------------------
520    Start=1                                           !Start channel
530    Stop=5                                            !Stop channel
540    Pointer=4                                         !Interrupt word
```

```
550   !
560   !____________________INITIALIZE THE MULTI______________________________
570   !                    --------------------
580   CALL Init_multi(@Multi,@Disc)
590   CALL Init_mem(@Multi,Memory1)
600   !
610   !___________________INITIALIZE CONTROL PROCEDURE______________________
620   !                   ----------------------------
630   CALL Init_control(@Multi,Memory2,Timer,D_to_a1,Contr_variable,Scale,Mean,S
pan,Gauge)
640   !
650   !_______________SET INTERRUPT BRANCH FOR TRIGGER SIGNAL_______________
660   !               --------------------------------------
670   Cycle=1                                 !Set cycle counter
680   G=SPOLL(723)                            !Clear Multi SRQ
690   P=SPOLL(720)                            !Clear DVM SRQ
700   OUTPUT @Dvm;"F1R-2N5Z0"                 !Set DVM ranges
710   OUTPUT @Dvm;"T2D3 TMF TEST"             !Set ext trigger and display
720   OUTPUT @Dvm;"KM01"                      !Set mask for Data Ready
730   ON INTR 7,2 GOTO Start_tmf              !Set interrupt branch
740   ENABLE INTR 7;2                         !Enable interrupt SRQ
750      BEEP
760      PRINT USING 770
770      IMAGE 2/,15X,"REMOVE CONTROL AND UTILITY PAC DISCS",2/,15X,"INSERT TWO
DISCS FOR DATA STORAGE",2/,15X,"PRESS K0(Continue) TO CONTINUE"
780      ON KEY 0 LABEL "**CONTINUE**" GOTO 800
790 Spin: GOTO Spin
800       BEEP
810       OFF KEY
820       PRINT USING 830
830       IMAGE 20/,25X,"PROGRAM AND START THE LEPEL",2/,20X,"****WAITING FOR A
TRIGGER SIGNAL****"
840       CREATE BDAT "TEST_PARAM",1,100
850       ASSIGN @Disc TO "TEST_PARAM"
860       OUTPUT @Disc;Material$,"TMLCF",Contr_variable,Scale,Mean,Span,Gauge
870       ASSIGN @Disc TO *
880       ENTER Dvm;P
890   !
900   !____________________START CYCLING AND TAKING DATA_____________________
910   !                    -----------------------------
920 Start_tmf:      !
930   OUTPUT @Multi;"CY",Timer,"T"                 !Start cycling
940   OUTPUT @Multi;"SC,0,0,0,0T"                  !Reset Clock of Multi
950   PRINT USING 960
960   IMAGE 10/,20X,"*****   TMFLCF TEST IN PROGRESS   *****"
970   !
980   !__________READ DATA____________
990   !
1000  CALL Read_data(@Multi,@Dvm,Multi,Dvm,Scanner,Memory1,Load(*),Strain(*),Tem
p1(*),Temp2(*),Temp3(*))
1010  !
1020  !__________STORE DATA___________
1030  !          ----------
1040  CALL Store_data(@Multi,@Disc,Disc_drive,Cycle,Scale,Load(*),Strain(*),Temp
1(*),Temp2(*),Temp3(*))
```

```
1050  !
1060  !__________DISPLAY DATA__________
1070  !         ------------
1080  CALL Plot_data(Material$,Contr_variable,Scale,Span,Gauge,Cycle,Load(*),Str
ain(*))
1090  !
1100  !___________SET INTERRUPT FOR SYNCHRO. SIGNAL_________
1110  !           --------------------------------
1120  IF Cycle<9 THEN
1130       Increment=2
1140       GOTO Synchro
1150  END IF
1160  Cycle2=INT(Cycle*1.25)
1170  Increment=Cycle2-Cycle
1180  !
1190 Synchro:                                   !
1200  Waiting=Increment-1.5
1210  Cycle=Cycle+Increment                      !Update cycle counter
1220  WAIT (180*Waiting)                         !Set waiting time
1230  OUTPUT @Dvm;"T2KM01"                       !Ext. trigger and DVM mask
1240  ON INTR 7,2 GOTO 990                       !Set interrupt branch
1250  ENABLE INTR 7;2                            !Enable interrupt
1260  ENTER Dvm;P
1270  SUBEND
1280!_____________________________________________________________________________
1290!
1300!  ******************************************************************
1310!  *                                                                *
1320!  *        SUBROUTINES USED BY SUB TMFCG(Param1,Param2,...)        *
1330!  *                                                                *
1340!  ******************************************************************
1350!
1360!_____________________________________________________________________________
1370  SUB Init_control(@Multi,Memory,Timer,D_to_a,Control,Scale,Mean,Span,Gauge)
1380  ALLOCATE A(4095)
1390    !__GENERATE WAVEFORM__
1400    !  ----------------
1410    Load_scale=INT(Scale)/100
1420    Strain_scale=Scale-(Load_scale*100)
1430    IF Control=1 THEN
1440        Pmean=Mean/(Load_scale*216.078)      !Convert MPA into voltage
1450        Pmax=Span/(Load_scale*216.078)
1460    ELSE
1470        Emax=.05493*Strain_scale/Gauge
1480        Pmean=10*Mean/Emax                   !Convert Strain into voltage
1490        Pmax=10*Span/Emax
1500    END IF
1510    Coeff=800
1520    IF Pmax>5.12 THEN Coeff=400
1530    Ndata=INT(Coeff*Pmax+.5)-1
1540    Ndata2=INT(Ndata/4+.5)
1550    Ndata3=Ndata-Ndata2
1560    FOR I=0 TO Ndata2-1
1570      A(I)=Pmax*I/(Ndata2-1)
1580      A(I+Ndata3)=Pmax*(I/(Ndata2-1)-1)
```

```
1590     NEXT I
1600     FOR I=Ndata2 TO Ndata3
1610       A(I)=(Ndata-1-2*I)*Pmax/(Ndata+1-2*Ndata2)
1620     NEXT I
1630     FOR I=Ndata+1 TO 4095
1640       A(I)=0.
1650     NEXT I
1660     !
1670     !__CALCULATE THE PACE__
1680     !  ------------------
1690     Time_int=180/Ndata
1700     Pace=INT(Time_int*500000)/1000             !Pace in msec
1710     !
1720     !__PROGRAM MULTI__AMPLITUDE
1730     !  ----------------------
1740     OUTPUT @Multi;"CC",Memory+1,"T"            !Clear memory card
1750     OUTPUT @Multi;"SF",Memory,3,1,.005,12,"T" !Set format(LSB=0.005)
1760     OUTPUT @Multi;"WF",Memory+1.0,4095,"T"     !Truncate memory
1770     OUTPUT @Multi;"MO",Memory,A(*),"T"         !Load data into memory
1780     OUTPUT @Multi;"WF",Memory+1,Ndata,"T"      !Truncate memory card
1790     OUTPUT @Multi;"WF",Memory+.1,10,"T"        !Set re-circulate mode
1800     OUTPUT @Multi;"WF",Timer+.2,1,"T"          !Set timer for continuous
1810     OUTPUT @Multi;"WF",Timer,Pace,"T"          !output. Set Pace
1820     !
1830     !__PROGRAM MULTI-- MEAN
1840     !  -------------------
1850     Incr=.005
1860     IF Pmean=0 THEN GOTO 1940
1870     IF Pmean<0 THEN Incr=-.005
1880     J=0
1890     OUTPUT @Multi;"OP",D_to_a,J,"T"
1900     IF ABS(J)<ABS(Pmean) THEN
1910        J=J+Incr
1920        GOTO 1890
1930     END IF
1940     DEALLOCATE A(*)
1950     SUBEND
1960!_____________________________________________________________________________
1970  SUB Read_data(@Multi,@Dvm,Multi,Dvm,Scanner,Memory,Load(*),Strain(*),Temp1
(*),Temp2(*),Temp3(*))
1980  ALLOCATE Scan(4)
1990  OUTPUT @Dvm;"T1M00"               !Set internal trigger and clear mask
2000     G=SPOLL(@Multi)
2010     ENTER 72310;A,B,C,D,E,F
2020     ON INTR 7,2 GOTO Interrupt
2030     ENABLE INTR 7;2
2040     OUTPUT @Multi;"CY",Scanner,"T"             !Enable scanner
2050     OUTPUT @Multi;"WF",Scanner+.3,5,"T"        !Set sequential mode
2060     !
2070     FOR K=0 TO 179
2080       OUTPUT @Multi;"WF",Scanner,1,"T"         !Set start channel
2090       OUTPUT @Multi;"WF",Scanner+.1,5,"T"      !set stop channel
2100       OUTPUT @Multi;"WF",Scanner+.2,40,"T"     !Set pace(40 us)
2110       Prog_diff(@Multi,Memory,0)               !Reset diff. counter
2120       Prog_write(@Multi,Memory,0)              !Reset write pointer
```

```
2130      OUTPUT @Multi;"CC";Memory,"T"              !Clear Memory card
2140      Prog_mode(@Multi,Memory,56)                !Set FIFO mode
2150      Prog_refi(@Multi,Memory,4)                 !Set stop pointer
2160      OUTPUT @Multi;"AC";Memory;"T"              !Armed memory card
2170      OUTPUT @Multi;"CY",Scanner,"T"             !Start pacer
2180 Wait:   GOTO Wait
2190 Interrupt:      !
2200      G=SPOLL(@Multi)
2210      IF G=64 THEN
2220        ENTER 72310;A,B,C
2230        IF C<>0 THEN
2240           ENTER 72312;Address
2250           IF Address<>Memory THEN
2260              PRINT "NOT MEMORY CARD"
2270              PAUSE
2280           END IF
2290           OUTPUT @Multi;"WF",Scanner+.2,0,"T"      !Stop scanner
2300           Prog_mode(@Multi,Memory,76)              !Set FIFO lockout
2310           OUTPUT @Multi;"MR";Memory;5;"T"          !MR command to get data
2320           ENTER 72305 USING "%,W";Scan(*)          !Entering the data
2330           Strain(K)=Scan(0)*.005                   !Convert the data into
2340           Load(K)=Scan(1)*.005                     !engineering units by
2350           Temp1(K)=Scan(2)*.005                    !multiplying with the
2360           Temp2(K)=Scan(3)*.005                    !LSB value.
2370           Temp3(K)=Scan(4)*.005
2380           BEEP
2390           WAIT .685                                !Wait set for 180
2400           ENABLE INTR 7;2
2410        ELSE
2420           BEEP 2000,.3
2430           PRINT "          ***  SRQ NOT SET BY ARMED CARD  ***"
2440           PAUSE
2450        END IF
2460     ELSE
2470        BEEP 2000,.3
2480        PRINT "          ***  MULTI DID NOT INTERRUPT  ***"
2490        PAUSE
2500     END IF
2510   NEXT K
2520   DEALLOCATE Scan(*)
2530   OUTPUT @Multi;"RC"                              !Read Real-time clock
2540   SUBEND
2550!______________________________________________________________________________
2560   SUB Store_data(@Multi,@Disc,Disc_drive,Cycle,Scale,Load(*),Strain(*),Temp
1(*),Temp2(*),Temp3(*))
2570   ALLOCATE X(179),Y(179)
2580   ENTER 72314;T1,T2,T3,T4                         !Enter Clock values
2590   Time$=VAL$(T1*24+T2)&":"&VAL$(T3)&":"&VAL$(T4)
2600   FOR I=0 TO 179
2610     X(I)=INT((Load(I)+10)*1000)+(Strain(I)+10)/100
2620     Y(I)=DROUND(Temp1(I),3)*1000000+DROUND(Temp2(I),3)*1000+Temp3(I)
2630  NEXT I
2640  !
2650  ON ERROR GOTO Recover
2660  CREATE BDAT "COCLE"&VAL$(Cycle),1,5000
```

```
2670  ASSIGN @Disc TO "COCLE"&VAL$(Cycle)
2680  OUTPUT @Disc;Cycle,Time$,Scale,X(*),Y(*)
2690  ASSIGN @Disc TO *
2700  GOTO Out
2710  !
2720 Recover: !
2730    IF ERRN=59 OR ERRN=64 THEN
2740       IF Disc_drive=0 THEN
2750          Disc_drive=1
2760          MASS STORAGE IS ":HP8290X,700,1"
2770       ELSE
2780          Disc_drive=0
2790          MASS STORAGE IS ":HP8290X,700,0"
2800       END IF
2810    ELSE
2820       PRINT USING 2830;ERRN
2830       IMAGE 2/,25X,"ERROR NUMBER ",K,/,25X,"CHECK STORAGE UNIT",/,25X,"PROG
RAM ABORTED"
2840       PAUSE
2850    END IF
2860 Out: !
2870    DEALLOCATE X(*),Y(*)
2880    SUBEND
2890!_________________________________________________________________________
2900 SUB Init_multi(@Multi,@Disc)
2910   CLEAR 7
2920   WAIT 4.0
2930   G=SPOLL(@Multi)
2940   IF G<>64 THEN
2950      PRINT "       ***MULTI DID NOT INTERRUPT***"
2960      PAUSE
2970   END IF
2980   STATUS @Multi,3;Multi
2990   ENTER Multi*100+10;A
3000   IF A<>16384 THEN
3010     PRINT "       ***SELF TEST DID NOT SET SRQ***"
3020     PAUSE
3030   END IF
3040  ALLOCATE Ascii$[80]
3050  ON END @Disc GOTO Eof
3060 Rd_file:    ENTER @Disc;Ascii$
3070    OUTPUT @Multi;Ascii$
3080    GOTO Rd_file
3090 Eof: OFF END @Disc
3100    ASSIGN @Disc TO *
3110    DEALLOCATE Ascii$
3120    MASS STORAGE IS ":HP8290X,700,0"
3130 SUBEND
3140!_________________________________________________________________________
3150 SUB Init_mem(@Multi,Memory)
3160   Prog_mode(@Multi,Memory,254)
3170   Prog_mode(@Multi,Memory,54)
3180   Prog_ref1(@Multi,Memory,1048575)
3190   Prog_ref2(@Multi,Memory,0)
3200   Prog_read(@Multi,Memory,0)
3210   Prog_diff(@Multi,Memory,0)
```

```
3220   Prog_write(@Multi,Memory,0)
3230   OUTPUT @Multi;"CC";Memory;"T"
3240 SUBEND
3250!________________________________________________________________________
3260 SUB Prog_read(@Multi,Memory,Read_pointer)
3270   Read_low=Read_pointer MOD 65536
3280   Read_high=Read_pointer DIV 65536
3290   Read_low_oct=FNDec_to_octal(Read_low)
3300   Read_high_oct=FNDec_to_octal(Read_high)
3310   OUTPUT @Multi;"WF";Memory+.1;22;Memory+.2;Read_low_oct;"T"
3320   OUTPUT @Multi;"WF";Memory+.1;23;Memory+.2;Read_high_oct;"T"
3330 SUBEND
3340!________________________________________________________________________
3350 SUB Prog_write(@Multi,Memory,Write_pointer)
3360   Write_low=Write_pointer MOD 65536
3370   Write_high=Write_pointer DIV 65536
3380   Write_low_oct=FNDec_to_octal(Write_low)
3390   Write_high_oct=FNDec_to_octal(Write_high)
3400   OUTPUT @Multi;"WF";Memory+.1;24;Memory+.2,Write_low_oct;"T"
3410   OUTPUT @Multi;"WF";Memory+.1;25;Memory+.2;Write_high_oct;"T"
3420 SUBEND
3430!________________________________________________________________________
3440 SUB Prog_diff(@Multi,Memory,Diff_counter)
3450   Diff_low=Diff_counter MOD 65536
3460   Diff_high=Diff_counter DIV 65536
3470   Diff_low_oct=FNDec_to_octal(Diff_low)
3480   Diff_high_oct=FNDec_to_octal(Diff_high)
3490   OUTPUT @Multi;"WF";Memory+.1;20;Memory+.2;Diff_low_oct;"T"
3500   OUTPUT @Multi;"WF";Memory+.1;21;Memory+.2;Diff_high_oct;"T"
3510 SUBEND
3520!________________________________________________________________________
3530 SUB Prog_ref1(@Multi,Memory,Ref1_reg)
3540   Ref1_low=Ref1_reg MOD 65536
3550   Ref1_high=Ref1_reg DIV 65536
3560   Ref1_low_oct=FNDec_to_octal(Ref1_low)
3570   Ref1_high_oct=FNDec_to_octal(Ref1_high)
3580   OUTPUT @Multi;"WF";Memory+.1,26;Memory+.2;Ref1_low_oct;"T"
3590   OUTPUT @Multi;"WF";Memory+.1,27;Memory+.2;Ref1_high_oct;"T"
3600 SUBEND
3610!________________________________________________________________________
3620 SUB Prog_ref2(@Multi,Memory,Ref2_reg)
3630   Ref2_low=Ref2_reg MOD 65536
3640   Ref2_high=Ref2_reg DIV 65536
3650   Ref2_low_oct=FNDec_to_octal(Ref2_low)
3660   Ref2_high_oct=FNDec_to_octal(Ref2_high)
3670   OUTPUT @Multi;"WF";Memory+.1;30;Memory+.2;Ref2_low_oct;"T"
3680   OUTPUT @Multi;"WF";Memory+.1;31;Memory+.2;Ref2_high_oct;"T"
3690 SUBEND
3700!________________________________________________________________________
3710 SUB Prog_mode(@Multi,Memory,Mode)
3720   OUTPUT @Multi;"WF";Memory+.3;Mode;"T"
3730 SUBEND
3740!________________________________________________________________________
3750 SUB Read_status(@Multi,Memory,Status)
```

```
3760   STATUS @Multi,3;Multi
3770   OUTPUT @Multi;"WF";Memory+.1;1;"T RV";Memory+.2;"T"
3780   ENTER Multi*100+6;Status
3790 SUBEND
3800!-----------------------------------------------------------------------------
3810 SUB Read_read(@Multi,Memory,Read_pointer)
3820   STATUS @Multi,3;Multi
3830   OUTPUT @Multi;"WF";Memory+.1;22;"T RV";Memory+.2;"T"
3840   ENTER Multi*100+6;Read_low_oct
3850   OUTPUT @Multi;"WF";Memory+.1;23;"T RV";Memory+.2;"T"
3860   ENTER Multi*100+6;Read_high_oct
3870   Read_low=FNOctal_to_dec(Read_low_oct)
3880   Read_high=FNOctal_to_dec(Read_high_oct)
3890   Read_pointer=65536*Read_high+Read_low
3900 SUBEND
3910!-----------------------------------------------------------------------------
3920 SUB Read_write(@Multi,Memory,Write_pointer)
3930   STATUS @Multi,3;Multi
3940   OUTPUT @Multi;"WF";Memory+.1;24;"T RV";Memory+.2;"T"
3950   ENTER Multi*100+6;Write_low_oct
3960   OUTPUT @Multi;"WF";Memory+.1;25;"T RV";Memory+.2;"T"
3970   ENTER Multi*100+6;Write_high_oct
3980   Write_low=FNOctal_to_dec(Write_low_oct)
3990   Write_high=FNOctal_to_dec(Write_high_oct)
4000   Write_pointer=65536*Write_high+Write_low
4010 SUBEND
4020!-----------------------------------------------------------------------------
4030 SUB Read_diff(@Multi,Memory,Diff_counter)
4040   STATUS @Multi,3;Multi
4050   OUTPUT @Multi;"WF";Memory+.1;20;"T RV";Memory+.2;"T"
4060   ENTER Multi*100+6;Diff_low_oct
4070   OUTPUT @Multi;"WF";Memory+.1;21;"T RV";Memory+.2;"T"
4080   ENTER Multi*100+6;Diff_high_oct
4090   Diff_low=FNOctal_to_dec(Diff_low_oct)
4100   Diff_high=FNOctal_to_dec(Diff_high_oct)
4110   Diff_counter=65536*Diff_high+Diff_low
4120 SUBEND
4130!-----------------------------------------------------------------------------
4140 DEF FNDec_to_octal(Dec)
4150   Oct=Dec+2*INT(Dec/8)+20*INT(Dec/64)+200*INT(Dec/512)+2000*INT(Dec/4096)+2
0000*INT(Dec/32768)+200000*INT(Dec/262144)
4160   RETURN Oct
4170 FNEND
4180!-----------------------------------------------------------------------------
4190 DEF FNOctal_to_dec(Octal)
4200   X9=0
4210   X7=Octal
4220   X=0
4230 More:   !
4240   X=X+(X7-INT(X7/10)*10)*8^X9
4250   X7=INT(X7/10)
4260   X9=X9+1
4270   IF X7<>0 THEN GOTO More
4280   RETURN X
```

```
4290 FNEND
4300!_______________________________________________________________________________
4310  SUB Plot_data(Material$,Contr_variable,Scale,Span,Gauge,Cycle,Load(*),Stra
in(*))
4320  ALLOCATE A(20),B(20)
4330  Load_range=INT(Scale)/100
4340  Strain_range=Scale-(Load_range*100)
4350  Conv_fac=216.078                  !Factor use to convert volt into MPa for
4360  Emax=.05493*Strain_range/Gauge !a SEN geometry. Scale strain range
4370  FOR I=0 TO 20
4380    A(I)=-50+I*50/9
4390    B(I)=-40+I*2.5
4400  NEXT I
4410  DEG
4420  GINIT
4430  GRAPHICS ON
4440  GCLEAR
4450  ALPHA OFF
4460  FRAME
4470  WINDOW -66.7224080268,66.7224080268,-50,50
4480  MOVE -50,-40
4490  FOR I=0 TO 18
4500    DRAW A(I),-40
4510    IDRAW 0,1
4520    IMOVE 0,-1
4530  NEXT I
4540  C=1.
4550  D=.5
4560  FOR I=0 TO 20
4570    DRAW 50,B(I)
4580    IF 2*INT(I/2)=I THEN
4590       IDRAW -D,0
4600       IMOVE D,0
4610    ELSE
4620       IDRAW -C,0
4630       IMOVE C,0
4640    END IF
4650  NEXT I
4660  DRAW 50,40
4670  DRAW -50,40
4680  DRAW -50,10
4690  FOR I=20 TO 0 STEP -1
4700    DRAW -50,B(I)
4710    IF 2*INT(I/2)=I THEN
4720       IDRAW D,0
4730       IMOVE -D,0
4740    ELSE
4750       IDRAW C,0
4760       IMOVE -C,0
4770    END IF
4780  NEXT I
4790  MOVE -50,15
4800  FOR I=0 TO 18
4810    DRAW A(I),-15
```

```
4820     IDRAW 0,1
4830     IDRAW 0,-2
4840     IMOVE 0,1
4850  NEXT I
4860  MOVE -50,10
4870  FOR I=0 TO 18
4880     DRAW A(I),10
4890     IDRAW 0,-1
4900     IMOVE 0,1
4910  NEXT I
4920  MOVE 10,10
4930  DRAW 10,40
4940  MOVE -40,25
4950  FOR I=0 TO 20 STEP 2
4960     DRAW (-40+I),25
4970     IDRAW 0,.5
4980     IDRAW 0,-1
4990     IMOVE 0,.5
5000  NEXT I
5010  MOVE -30,15
5020  FOR I=0 TO 20 STEP 2
5030     DRAW -30,15+I
5040     IDRAW .5,0
5050     IDRAW -1,0
5060     IMOVE .5,0
5070  NEXT I
5080  CSIZE 3,.6
5090  MOVE -50,-40
5100  FOR I=2 TO 17 STEP 2
5110     MOVE A(I)-3,-43
5120     LABEL VAL$(I*10)
5130  NEXT I
5140  FOR I=1 TO 10 STEP 2
5150       MOVE 50.5,B(I)-1.5
5160       LABEL VAL$(.2*I-1.0)
5170       MOVE -55,B(I)-1.5
5180       LABEL VAL$(.2*I-1.0)
5190  NEXT I
5200  FOR I=11 TO 20 STEP 2
5210       MOVE 50.5,B(I)-1.5
5220       LABEL VAL$((I-15)*100)
5230       MOVE -58,B(I)-1.5
5240       LABEL VAL$((I-15)*100)
5250  NEXT I
5260  CSIZE 4,.6
5270  MOVE -20,-48
5280  LABEL "TIME  (sec)"
5290  CSIZE 3,.6
5300  LDIR 90
5310  MOVE -56,-35
5320  LABEL "Etot (%)"
5330  MOVE 62,-35
5340  LABEL "Etot (%)"
5350  MOVE -58,-14
```

```
5360  LABEL "STRESS (MPa)"
5370  MOVE 62,-14
5380  LABEL "STRESS (MPa)"
5390  CSIZE 3.5,.6
5400  LDIR 0
5410  MOVE 15,32
5420  LABEL Material$
5430  MOVE 15,29
5440  LABEL "TMLCF"
5450  MOVE 20,26
5460  LABEL "Tmax= 926 C"
5470  MOVE 20,23
5480  LABEL "Tmin= 400 C"
5490  MOVE 15,20
5500  LABEL "CYCLE #"&VAL$(Cycle)
5510  CSIZE 3,.6
5520  MOVE -15,29
5530  DRAW -13,29
5540  DRAW -14,30
5550  DRAW -15,29
5560  MOVE -12,28
5570  IF Contr_variable=1 THEN
5580     LABEL "S= "&VAL$(2*Span)&"MPa"
5590     MOVE -15,32
5600     LABEL "STRESS CONTROL"
5610  ELSE
5620     LABEL "E= "&VAL$(2*Span)
5630     MOVE -15,32
5640     LABEL "STRAIN CONTROL"
5650  END IF
5660  MOVE -15,24
5670  LABEL "X-DIV=0.05%"
5680  MOVE -15,20
5690  LABEL "Y-DIV=100 MPa"
5700  MOVE -50,-2.5
5710  FOR I=0 TO 179                                  !Convert volt to Mpa with
5720    B1=(((Load(I)*Load_range*Conv_fac)+500)/40)-15 !the assumption that the
5730    A1=(I*5/9)-50                                 !speciment cross section
5740    DRAW A1,B1                                    !is (11.7*4.4)mm2.
5750  NEXT I
5760  MOVE -50,-27.5
5770  FOR I=0 TO 179
5780     B1=((Strain(I)*Emax/10)+.01)*1250-40         !Plot total strain
5790     A1=(I*5/9)-50
5800     DRAW A1,B1
5810  NEXT I
5820  A1=((Strain(0)*Emax/10)+.0025)*4000-40
5830  B1=(((Load(0)*Range*Conv_fac)+500)/50)+15
5840  MOVE A1,B1
5850  FOR I=1 TO 179
5860     A1=((Strain(I)*Emax/10)+.0025)*4000-40      !Plot the hysterisis
5870     B1=(((Load(I)*Range*Conv_fac)+500)/50)+15   !loop.
5880     DRAW A1,B1
5890  NEXT I
5900  SUBEND
```

```
10!    ***********************************************************************
20!    *                                                                     *
30!    *                      PROGRAM TMFCG                                  *
40!    *                      *************                                  *
50!    *                                                                     *
60!    *       THIS PROGRAM PERFORMS THE REAL-TIME CONTROL AND DATA          *
70!    *       ACQUISITION OF THERMAL-MECHANICAL FATIGUE CRACK GROWTH        *
80!    *       TESTS.                                                        *
90!    *                                                                     *
100!   *       THIS PROGRAM IS CALLED BY ISO_TMF AFTER SELECTION OF THE      *
110!   *       TMFCG OPTION.                                                 *
120!   *                                                                     *
130!   *       THE MICRO_DATA_TRACK MUST BE PROGRAMMED FOR 180 sec           *
140!   *       PERIOD (1/3 cpm).                                             *
150!   *                                                                     *
160!   *       THE CYCLIC THERMAL STRAINS ARE FIRST MEASURED BY CYCLING      *
170!   *       THE TEMPERATURE AT ZERO STRESS.                               *
180!   *                                                                     *
190!   *       THE CYCLIC DATA(Stresses, strains, temperatures, and          *
200!   *       potentials) ARE RECORDED EVERY 5 CYCLES.                      *
210!   *                                                                     *
220!   ***********************************************************************
230    !
240    OPTION BASE 0
250    Material$="B-1900+Hf"
260    Contr_variable=2
270    Scale=50.10
280    Mean=0.
290    Span=.00250
300    Gauge=.515
310    CALL Tmfcg(Material$,Contr_variable,Scale,Mean,Span,Gauge)
320    END
330!**************************************************************************
340!**************************************************************************
350    !
360    SUB Tmfcg(Material$,Contr_variable,Scale,Mean,Span,Gauge)
370    DIM Load(179),Strain(179),Temp1(179),Temp2(179),Temp3(179),Pot(179)
380    MASS STORAGE IS ":HP8290X,700,1"
390    !
400    !______ _______________DEFINE I/O CARD FUNCTION__________________________
410    !                     -----------------------
420    ASSIGN @Dvm TO 720
430    ASSIGN @Multi TO 723
440    ASSIGN @Disc TO "MRZ"
450    Disc_drive=0
460    Dvm=720
470    Multi=723
480    Scanner=0
490    A_to_d=3
500    Memory1=7
510    D_to_a1=10
520    Timer=11
530    D_to_a2=12
540    Memory2=13
```

```
550    !
560    !__________________DEFAULT VALUES FOR THE PARAMETERS__________________
570    !                  --------------------------------
580    Start=1                                              !Start channel
590    Stop=5                                               !Stop channel
600    Pointer=4                                            !Interrupt word
610    !
620    !______________________INITIALIZE THE MULTI___________________________
630    !                      --------------------
640    CALL Init_multi(@Multi,@Disc)
650    CALL Init_mem(@Multi,Memory1)
660    !
670    !____________________INITIALIZE CONTROL PROCEDURE_____________________
680    !                    ----------------------------
690    CALL Init_control(@Multi,Memory2,Timer,D_to_a1,Contr_variable,Scale,Mean,S
pan,Gauge)
700    !
710    !________________SET INTERRUPT BRANCH FOR TRIGGER SIGNAL_______________
720    !                ---------------------------------------
730    Cycle=1                                !Set cycle counter
740    G=SPOLL(723)                           !Clear Multi SRQ
750    P=SPOLL(720)                           !Clear DVM SRQ
760    OUTPUT @Dvm;"F1R-2N5Z0"                !Set DVM ranges
770    OUTPUT @Dvm;"T2D3 TMF TEST"            !Set ext trigger and display
780    OUTPUT @Dvm;"KM01"                     !Set mask for Data Ready
790    ON INTR 7,2 GOTO Start_tmf             !Set interrupt branch
800    ENABLE INTR 7;2                        !Enable interrupt SRQ
810       BEEP
820       PRINT USING 830
830       IMAGE 2/,15X,"REMOVE CONTROL AND UTILITY PAC DISCS",2/,15X,"INSERT TWO
DISCS FOR DATA STORAGE",2/,15X,"PRESS K0(Continue) TO CONTINUE"
840        ON KEY 0 LABEL "**CONTINUE**" GOTO 860
850 Spin: GOTO Spin
860        BEEP
870        OFF KEY
880        PRINT USING 890
890        IMAGE 2/,15X,"PROGRAM AND START THE LEPEL",2/,15X,"****WAITING FOR A T
RIGGER SIGNAL****"
900        CREATE BDAT "TEST_PARAM",1,100
910        ASSIGN @Disc TO "TEST_PARAM"
920        OUTPUT @Disc;Material$,"TMFCG",Contr_variable,Scale,Mean,Span,Gauge
930        ASSIGN @Disc TO *
940        ENTER Dvm;P
950    !
960    !_____________________START CYCLING AND TAKING DATA____________________
970    !                     ----------------------------
980 Start_tmf:       !
990    OUTPUT @Multi;"CY",Timer,"T"                 !Start cycling
1000   OUTPUT @Multi;"SC,0,0,0,0T"                  !Reset Clock of Multi
1010   !
1020   !__________READ DATA___________
1030   !
```

```
1040  CALL Read_data(@Multi,@Dvm,Multi,Dvm,Scanner,Memory1,Load(*),Strain(*),Tem
p1(*),Temp2(*),Temp3(*),Pot(*))
1050  !
1060  !__________STORE DATA__________
1070  !          ----------
1080  CALL Store_data(@Multi,@Disc,Disc_drive,Cycle,Scale,Load(*),Strain(*),Temp
1(*),Temp2(*),Temp3(*),Pot(*))
1090  !
1100  !__________DISPLAY DATA__________
1110  !          ------------
1120  CALL Plot_data(Material$,Contr_variable,Scale,Span,Gauge,Cycle,Load(*),Str
ain(*),Pot(*))
1130  !
1140  !___________SET INTERRUPT FOR SYNCHRO. SIGNAL_________
1150  !           --------------------------------
1160  IF Cycle>11 THEN GOTO Synchro
1170  IF Cycle<=5 THEN
1180     Increment=2
1190     Waiting=.5
1200  ELSE
1210     IF Cycle=7 THEN
1220        Increment=3
1230        Waiting=1.5
1240     ELSE
1250        Increment=5
1260        Waiting=3.5
1270     END IF
1280  END IF
1290  !
1300 Synchro:                                  !
1310  Cycle=Cycle+Increment                    !Update cycle counter
1320  WAIT (180*Waiting)                       !Set waiting time
1330  OUTPUT @Dvm;"T2KM01"                     !Ext. trigger and DVM mask
1340  ON INTR 7,2 GOTO 1030                    !Set interrupt branch
1350  ENABLE INTR 7;2                          !Enable interrupt
1360  GRAPHICS OFF
1370  PRINT USING 1380
1380  IMAGE 10/,20X,"*****   TMFCG TEST IN PROGRESS   *****"
1390  ENTER Dvm;P
1400  SUBEND
1410!_________________________________________________________________________
1420!
1430!  ******************************************************************
1440!  *                                                                *
1450!  *        SUBROUTINES USED BY SUB TMFCG(Param1,Param2,...)        *
1460!  *                                                                *
1470!  ******************************************************************
1480!
1490!_________________________________________________________________________
1500  SUB Init_control(@Multi,Memory,Timer,D_to_a,Control,Scale,Mean,Span,Gauge)
1510  ALLOCATE A(4095)
1520    !__GENERATE WAVEFORM__
1530    !  -----------------
1540    Load_scale=INT(Scale)/100
1550    Strain_scale=Scale-(Load_scale*100)
1560    IF Control=1 THEN
```

```
1570        Pmean=Mean/(Load_scale*216.078)       !Convert MPA into voltage
1580        Pmax=Span/(Load_scale*216.078)
1590     ELSE
1600        Emax=.05493*Strain_scale/Gauge
1610        Pmean=10*Mean/Emax                    !Convert Strain into voltage
1620        Pmax=10*Span/Emax
1630     END IF
1640     Coeff=800
1650     IF Pmax>5.12 THEN Coeff=400
1660     Ndata=INT(Coeff*Pmax+.5)-1
1670     Ndata2=INT(Ndata/4+.5)
1680     Ndata3=Ndata-Ndata2
1690     FOR I=0 TO Ndata2-1
1700       A(I)=Pmax*I/(Ndata2-1)
1710       A(I+Ndata3)=Pmax*(I/(Ndata2-1)-1)
1720     NEXT I
1730     FOR I=Ndata2 TO Ndata3
1740       A(I)=(Ndata-1-2*I)*Pmax/(Ndata+1-2*Ndata2)
1750     NEXT I
1760     FOR I=Ndata+1 TO 4095
1770       A(I)=0.
1780     NEXT I
1790     !
1800     !__CALCULATE THE PACE__
1810     ! -----------------
1820     Time_int=180/Ndata
1830     Pace=INT(Time_int*500000)/1000           !Pace in msec
1840     !
1850     !__PROGRAM MULTI__AMPLITUDE
1860     ! ----------------------
1870     OUTPUT @Multi;"CC",Memory+1,"T"          !Clear memory card
1880     OUTPUT @Multi;"SF",Memory,3,1,.005,12,"T" !Set format(LSB=0.005)
1890     OUTPUT @Multi;"WF",Memory+1.0,4095,"T"   !Truncate memory
1900     OUTPUT @Multi;"MO",Memory,A(*),"T"       !Load data into memory
1910     OUTPUT @Multi;"WF",Memory+1,Ndata,"T"    !Truncate memory card
1920     OUTPUT @Multi;"WF",Memory+.1,10,"T"      !Set re-circulate mode
1930     OUTPUT @Multi;"WF",Timer+.2,1,"T"        !Set timer for continuous
1940     OUTPUT @Multi;"WF",Timer,Pace,"T"        !output. Set Pace
1950     !
1960     !__PROGRAM MULTI-- MEAN
1970     ! --------------------
1980     Incr=.005
1990     IF Pmean=0 THEN GOTO 2070
2000     IF Pmean<0 THEN Incr=-.005
2010     J=0
2020     OUTPUT @Multi;"OP",D_to_a,J,"T"
2030     IF ABS(J)<ABS(Pmean) THEN
2040        J=J+Incr
2050        GOTO 2020
2060     END IF
2070     DEALLOCATE A(*)
2080     SUBEND
2090!______________________________________________________________
```

```
2100  SUB Read_data(@Multi,@Dvm,Multi,Dvm,Scanner,Memory,Load(*),Strain(*),Temp1
(*),Temp2(*),Temp3(*),Pot(*))
2110  ALLOCATE Scan(4)
2120  OUTPUT @Dvm;"T1M00"                !Set internal trigger and clear mask
2130    G=SPOLL(@Multi)
2140    ENTER 72310;A,B,C,D,E,F
2150    ON INTR 7,2 GOTO Interrupt
2160    ENABLE INTR 7;2
2170    OUTPUT @Multi;"CY",Scanner,"T"             !Enable scanner
2180    OUTPUT @Multi;"WF",Scanner+.3,5,"T"        !Set sequential mode
2190    !
2200    FOR K=0 TO 179
2210      OUTPUT @Multi;"WF",Scanner,1,"T"         !Set start channel
2220      OUTPUT @Multi;"WF",Scanner+.1,5,"T"      !set stop channel
2230      OUTPUT @Multi;"WF",Scanner+.2,40,"T"     !Set pace(40 us)
2240      Prog_diff(@Multi,Memory,0)               !Reset diff. counter
2250      Prog_write(@Multi,Memory,0)              !Reset write pointer
2260      OUTPUT @Multi;"CC";Memory,"T"            !Clear Memory card
2270      Prog_mode(@Multi,Memory,56)              !Set FIFO mode
2280      Prog_ref1(@Multi,Memory,4)               !Set stop pointer
2290      OUTPUT @Multi;"AC";Memory;"T"            !Armed memory card
2300      OUTPUT @Multi;"CY",Scanner,"T"           !Start pacer
2310 Wait:   GOTO Wait
2320 Interrupt:      !
2330      G=SPOLL(@Multi)
2340      IF G=64 THEN
2350        ENTER 72310;A,B,C
2360        IF C<>0 THEN
2370           ENTER 72312;Address
2380           IF Address<>Memory THEN
2390              PRINT "NOT MEMORY CARD"
2400              PAUSE
2410           END IF
2420           OUTPUT @Multi;"WF",Scanner+.2,0,"T"      !Stop scanner
2430           Prog_mode(@Multi,Memory,76)              !Set FIFO lockout
2440           OUTPUT @Multi;"MR";Memory;5;"T"          !MR command to get data
2450           ENTER 72305 USING "%,W";Scan(*)          !Entering the data
2460           Strain(K)=Scan(0)*.005                   !Convert the data into
2470           Load(K)=Scan(1)*.005                     !engineering units by
2480           Temp1(K)=Scan(2)*.005                    !multiplying with the
2490           Temp2(K)=Scan(3)*.005                    !LSB value.
2500           Temp3(K)=Scan(4)*.005
2510           ENTER Dvm;Pot(K)                         !Enter potential
2520           BEEP
2530           WAIT .685                                !Wait set for 180
2540           ENABLE INTR 7;2
2550        ELSE
2560           BEEP 2000,.3
2570           PRINT "        ***  SRQ NOT SET BY ARMED CARD  ***"
2580           PAUSE
2590        END IF
2600      ELSE
2610         BEEP 2000,.3
2620         PRINT "        ***  MULTI DID NOT INTERRUPT  ***"
2630         PAUSE
```

```
2640     END IF
2650    NEXT K
2660    DEALLOCATE Scan(*)
2670    OUTPUT @Multi;"RC"                                 !Read Real-time clock
2680    SUBEND
2690!____________________________________________________________________________
2700    SUB Store_data(@Multi,@Disc,Disc_drive,Cycle,Scale,Load(*),Strain(*),Temp
1(*),Temp2(*),Temp3(*),Pot(*))
2710    ALLOCATE X(179),Y(179)
2720    ENTER 72314;T1,T2,T3,T4                            !Enter Clock values
2730    Time$=VAL$(T1*24+T2)&":"&VAL$(T3)&":"&VAL$(T4)
2740    FOR I=0 TO 179
2750      X(I)=INT((Load(I)+10)*1000)+(Strain(I)+10)/100
2760      Y(I)=DROUND(Temp1(I),3)*1000000+DROUND(Temp2(I),3)*1000+Temp3(I)
2770   NEXT I
2780   !
2790   ON ERROR GOTO Recover
2800   CREATE BDAT "CYCLE"&VAL$(Cycle),1,5000
2810   ASSIGN @Disc TO "CYCLE"&VAL$(Cycle)
2820   OUTPUT @Disc;Cycle,Time$,Scale,X(*),Y(*),Pot(*)
2830   ASSIGN @Disc TO *
2840   GOTO Out
2850   !
2860 Recover: !
2870     IF ERRN=59 OR ERRN=64 THEN
2880        IF Disc_drive=0 THEN
2890           Disc_drive=1
2900           MASS STORAGE IS ":HP8290X,700,1"
2910        ELSE
2920           Disc_drive=0
2930           MASS STORAGE IS ":HP8290X,700,0"
2940        END IF
2950     ELSE
2960        PRINT USING 2970;ERRN
2970        IMAGE 2/,25X,"ERROR NUMBER ",K,/,25X,"CHECK STORAGE UNIT",/,25X,"PROG
RAM ABORTED"
2980        PAUSE
2990     END IF
3000 Out: !
3010     DEALLOCATE X(*),Y(*)
3020     SUBEND
3030!____________________________________________________________________________
3040 SUB Init_multi(@Multi,@Disc)
3050   CLEAR 7
3060   WAIT 4.0
3070   G=SPOLL(@Multi)
3080   IF G<>64 THEN
3090     PRINT "      ***MULTI DID NOT INTERRUPT***"
3100     PAUSE
3110   END IF
3120   STATUS @Multi,3;Multi
3130   ENTER Multi*100+10;A
3140   IF A<>16384 THEN
3150     PRINT "      ***SELF TEST DID NOT SET SRQ***"
```

```
3160      PAUSE
3170    END IF
3180   ALLOCATE Ascii$[80]
3190   ON END @Disc GOTO Eof
3200 Rd_file:    ENTER @Disc;Ascii$
3210      OUTPUT @Multi;Ascii$
3220      GOTO Rd_file
3230 Eof: OFF END @Disc
3240      ASSIGN @Disc TO *
3250      DEALLOCATE Ascii$
3260      MASS STORAGE IS ":HP8290X,700,0"
3270 SUBEND
3280!_____________________________________________________________________
3290 SUB Init_mem(@Multi,Memory)
3300    Prog_mode(@Multi,Memory,254)
3310    Prog_mode(@Multi,Memory,54)
3320    Prog_ref1(@Multi,Memory,1048575)
3330    Prog_ref2(@Multi,Memory,0)
3340    Prog_read(@Multi,Memory,0)
3350    Prog_diff(@Multi,Memory,0)
3360    Prog_write(@Multi,Memory,0)
3370    OUTPUT @Multi;"CC";Memory;"T"
3380 SUBEND
3390!_____________________________________________________________________
3400 SUB Prog_read(@Multi,Memory,Read_pointer)
3410    Read_low=Read_pointer MOD 65536
3420    Read_high=Read_pointer DIV 65536
3430    Read_low_oct=FNDec_to_octal(Read_low)
3440    Read_high_oct=FNDec_to_octal(Read_high)
3450    OUTPUT @Multi;"WF";Memory+.1;22;Memory+.2;Read_low_oct;"T"
3460    OUTPUT @Multi;"WF";Memory+.1;23;Memory+.2;Read_high_oct;"T"
3470 SUBEND
3480!_____________________________________________________________________
3490 SUB Prog_write(@Multi,Memory,Write_pointer)
3500    Write_low=Write_pointer MOD 65536
3510    Write_high=Write_pointer DIV 65536
3520    Write_low_oct=FNDec_to_octal(Write_low)
3530    Write_high_oct=FNDec_to_octal(Write_high)
3540    OUTPUT @Multi;"WF";Memory+.1;24;Memory+.2,Write_low_oct;"T"
3550    OUTPUT @Multi;"WF";Memory+.1;25;Memory+.2;Write_high_oct;"T"
3560 SUBEND
3570!_____________________________________________________________________
3580 SUB Prog_diff(@Multi,Memory,Diff_counter)
3590    Diff_low=Diff_counter MOD 65536
3600    Diff_high=Diff_counter DIV 65536
3610    Diff_low_oct=FNDec_to_octal(Diff_low)
3620    Diff_high_oct=FNDec_to_octal(Diff_high)
3630    OUTPUT @Multi;"WF";Memory+.1;20;Memory+.2;Diff_low_oct;"T"
3640    OUTPUT @Multi;"WF";Memory+.1;21;Memory+.2;Diff_high_oct;"T"
3650 SUBEND
3660!_____________________________________________________________________
3670 SUB Prog_ref1(@Multi,Memory,Ref1_reg)
3680    Ref1_low=Ref1_reg MOD 65536
3690    Ref1_high=Ref1_reg DIV 65536
```

```
3700   Ref1_low_oct=FNDec_to_octal(Ref1_low)
3710   Ref1_high_oct=FNDec_to_octal(Ref1_high)
3720   OUTPUT @Multi;"WF";Memory+.1,26;Memory+.2;Ref1_low_oct;"T"
3730   OUTPUT @Multi;"WF";Memory+.1,27;Memory+.2;Ref1_high_oct;"T"
3740 SUBEND
3750!_____________________________________________________________________
3760 SUB Prog_ref2(@Multi,Memory,Ref2_reg)
3770   Ref2_low=Ref2_reg MOD 65536
3780   Ref2_high=Ref2_reg DIV 65536
3790   Ref2_low_oct=FNDec_to_octal(Ref2_low)
3800   Ref2_high_oct=FNDec_to_octal(Ref2_high)
3810   OUTPUT @Multi;"WF";Memory+.1;30;Memory+.2;Ref2_low_oct;"T"
3820   OUTPUT @Multi;"WF";Memory+.1;31;Memory+.2;Ref2_high_oct;"T"
3830 SUBEND
3840!_____________________________________________________________________
3850 SUB Prog_mode(@Multi,Memory,Mode)
3860   OUTPUT @Multi;"WF";Memory+.3;Mode;"T"
3870 SUBEND
3880!_____________________________________________________________________
3890 SUB Read_status(@Multi,Memory,Status)
3900   STATUS @Multi,3;Multi
3910   OUTPUT @Multi;"WF";Memory+.1;1;"T RV";Memory+.2;"T"
3920   ENTER Multi*100+6;Status
3930 SUBEND
3940!_____________________________________________________________________
3950 SUB Read_read(@Multi,Memory,Read_pointer)
3960   STATUS @Multi,3;Multi
3970   OUTPUT @Multi;"WF";Memory+.1;22;"T RV";Memory+.2;"T"
3980   ENTER Multi*100+6;Read_low_oct
3990   OUTPUT @Multi;"WF";Memory+.1;23;"T RV";Memory+.2;"T"
4000   ENTER Multi*100+6;Read_high_oct
4010   Read_low=FNOctal_to_dec(Read_low_oct)
4020   Read_high=FNOctal_to_dec(Read_high_oct)
4030   Read_pointer=65536*Read_high+Read_low
4040 SUBEND
4050!_____________________________________________________________________
4060 SUB Read_write(@Multi,Memory,Write_pointer)
4070   STATUS @Multi,3;Multi
4080   OUTPUT @Multi;"WF";Memory+.1;24;"T RV";Memory+.2;"T"
4090   ENTER Multi*100+6;Write_low_oct
4100   OUTPUT @Multi;"WF";Memory+.1;25;"T RV";Memory+.2;"T"
4110   ENTER Multi*100+6;Write_high_oct
4120   Write_low=FNOctal_to_dec(Write_low_oct)
4130   Write_high=FNOctal_to_dec(Write_high_oct)
4140   Write_pointer=65536*Write_high+Write_low
4150 SUBEND
4160!_____________________________________________________________________
4170 SUB Read_diff(@Multi,Memory,Diff_counter)
4180   STATUS @Multi,3;Multi
4190   OUTPUT @Multi;"WF";Memory+.1;20;"T RV";Memory+.2;"T"
4200   ENTER Multi*100+6;Diff_low_oct
4210   OUTPUT @Multi;"WF";Memory+.1;21;"T RV";Memory+.2;"T"
```

```
4220   ENTER Multi*100+6;Diff_high_oct
4230   Diff_low=FNOctal_to_dec(Diff_low_oct)
4240   Diff_high=FNOctal_to_dec(Diff_high_oct)
4250   Diff_counter=65536*Diff_high+Diff_low
4260 SUBEND
4270!_______________________________________________________________________________
4280 DEF FNDec_to_octal(Dec)
4290   Oct=Dec+2*INT(Dec/8)+20*INT(Dec/64)+200*INT(Dec/512)+2000*INT(Dec/4096)+2
0000*INT(Dec/32768)+200000*INT(Dec/262144)
4300   RETURN Oct
4310 FNEND
4320!_______________________________________________________________________________
4330 DEF FNOctal_to_dec(Octal)
4340   X9=0
4350   X7=Octal
4360   X=0
4370 More:   !
4380   X=X+(X7-INT(X7/10)*10)*8^X9
4390   X7=INT(X7/10)
4400   X9=X9+1
4410   IF X7<>0 THEN GOTO More
4420   RETURN X
4430 FNEND
4440!_______________________________________________________________________________
4450  SUB Plot_data(Material$,Contr_variable,Scale,Span,Gauge,Cycle,Load(*),Stra
in(*),Pot(*))
4460  ALLOCATE A(20),B(20)
4470  Load_range=INT(Scale)/100
4480  Strain_range=Scale-(Load_range*100)
4490  Conv_fac=216.078                 !Factor use to convert volt into MPa for
4500  Emax=.05493*Strain_range/Gauge !a SEN geometry. Scale strain range
4510  FOR I=0 TO 20
4520    A(I)=-50+I*50/9
4530    B(I)=-40+I*2.5
4540  NEXT I
4550  DEG
4560  GINIT
4570  GRAPHICS ON
4580  GCLEAR
4590  ALPHA OFF
4600  FRAME
4610  WINDOW -66.7224080268,66.7224080268,-50,50
4620  MOVE -50,-40
4630  FOR I=0 TO 18
4640    DRAW A(I),-40
4650    IDRAW 0,1
4660    IMOVE 0,-1
4670  NEXT I
4680  C=.5
4690  D=1
4700  FOR I=0 TO 20
4710    DRAW 50,B(I)
4720    IF I>=10 THEN
4730       C=1
```

```
4740        D=.5
4750     ELSE
4760     END IF
4770     IF 2*INT(I/2)=I THEN
4780        IDRAW -D,0
4790        IMOVE D,0
4800     ELSE
4810        IDRAW -C,0
4820        IMOVE C,0
4830     END IF
4840  NEXT I
4850  DRAW 50,40
4860  DRAW -50,40
4870  DRAW -50,10
4880  FOR I=20 TO 0 STEP -1
4890     DRAW -50,B(I)
4900     IF I<=10 THEN
4910        C=.5
4920        D=1.0
4930     ELSE
4940     END IF
4950     IF 2*INT(I/2)=I THEN
4960        IDRAW D,0
4970        IMOVE -D,0
4980     ELSE
4990        IDRAW C,0
5000        IMOVE -C,0
5010     END IF
5020  NEXT I
5030  MOVE -50,15
5040  FOR I=0 TO 18
5050     DRAW A(I),-15
5060     IDRAW 0,1
5070     IDRAW 0,-2
5080     IMOVE 0,1
5090  NEXT I
5100  MOVE -50,10
5110  FOR I=0 TO 18
5120     DRAW A(I),10
5130     IDRAW 0,-1
5140     IMOVE 0,1
5150  NEXT I
5160  MOVE 10,10
5170  DRAW 10,40
5180  MOVE -40,25
5190  FOR I=0 TO 20 STEP 2
5200     DRAW (-40+I),25
5210     IDRAW 0,.5
5220     IDRAW 0,-1
5230     IMOVE 0,.5
5240  NEXT I
5250  MOVE -30,15
5260  FOR I=0 TO 20 STEP 2
5270     DRAW -30,15+I
```

```
5280     IDRAW .5,0
5290     IDRAW -1,0
5300     IMOVE .5,0
5310  NEXT I
5320  CSIZE 3,.6
5330  MOVE -50,-40
5340  FOR I=2 TO 17 STEP 2
5350    MOVE A(I)-3,-43
5360    LABEL VAL$(I*10)
5370  NEXT I
5380  FOR I=0 TO 10 STEP 2
5390     MOVE 50.5,B(I)-1.5
5400     LABEL VAL$(.5+I*.50)
5410     MOVE -55,B(I)-1.5
5420     LABEL VAL$(.5+I*.50)
5430  NEXT I
5440  FOR I=11 TO 20 STEP 2
5450     MOVE 50.5,B(I)-1.5
5460     LABEL VAL$((I-15)*100)
5470     MOVE -58,B(I)-1.5
5480     LABEL VAL$((I-15)*100)
5490  NEXT I
5500  CSIZE 4,.6
5510  MOVE -20,-48
5520  LABEL "TIME  (sec)"
5530  CSIZE 3,.6
5540  LDIR 90
5550  MOVE -56,-38
5560  LABEL "POT. (mV)"
5570  MOVE 58,-38
5580  LABEL "POT. (mV)"
5590  MOVE -58,-14
5600  LABEL "STRESS (MPa)"
5610  MOVE 62,-14
5620  LABEL "STRESS (MPa)"
5630  CSIZE 3.5,.6
5640  LDIR 0
5650  MOVE 15,32
5660  LABEL Material$
5670  MOVE 15,29
5680  LABEL "TMFCG"
5690  MOVE 20,26
5700  LABEL "Tmax= 926 C"
5710  MOVE 20,23
5720  LABEL "Tmin= 400 C"
5730  MOVE 15,20
5740  LABEL "CYCLE #"&VAL$(Cycle)
5750  CSIZE 3,.6
5760  MOVE -15,29
5770  DRAW -13,29
5780  DRAW -14,30
5790  DRAW -15,29
5800  MOVE -12,28
5810  IF Contr_variable=1 THEN
```

```
5820     LABEL "S= "&VAL$(2*Span)&"MPa"
5830     MOVE -15,32
5840     LABEL "STRESS CONTROL"
5850  ELSE
5860     LABEL "E= "&VAL$(2*Span)
5870     MOVE -15,32
5880     LABEL "STRAIN CONTROL"
5890  END IF
5900  MOVE -15,24
5910  LABEL "X-DIV=0.05%"
5920  MOVE -15,20
5930  LABEL "Y-DIV=100 MPa"
5940  MOVE -50,-2.5
5950  FOR I=0 TO 179                                      !Convert volt to Mpa with
5960    B1=(((Load(I)*Load_range*Conv_fac)+500)/40)-15 !the assumption that the
5970    A1=(I*5/9)-50                                   !speciment cross section
5980    DRAW A1,B1                                      !is (11.7*4.4)mm2.
5990  NEXT I
6000  MOVE -50,-27.5
6010  FOR I=0 TO 179                                      !Plot potential
6020     B1=(Pot(I)*1000-.5)*5-40
6030     A1=(I*5/9)-50
6040     DRAW A1,B1
6050  NEXT I
6060  A1=((Strain(0)*Emax/10)+.0025)*4000-40
6070  B1=(((Load(0)*Range*Conv_fac)+500)/50)+15
6080  MOVE A1,B1
6090  FOR I=1 TO 179
6100     A1=((Strain(I)*Emax/10)+.0025)*4000-40        !Plot the hysterisis
6110     B1=(((Load(I)*Range*Conv_fac)+500)/50)+15     !loop.
6120     DRAW A1,B1
6130  NEXT I
6140  SUBEND
```

```
10!     ************************************************************************
20!     *                                                                      *
30!     *                        PROGRAM ISOCG                                 *
40!     *                        *************                                 *
50!     *                                                                      *
60!     *       THIS PROGRAM PERFORMS THE REAL-TIME CONTROL AND DATA           *
70!     *       ACQUISITION OF ISOTHERMAL LOW CYCLE FATIGUE CRACK              *
80!     *       GROWTH TESTS.                                                  *
90!     *                                                                      *
100!    *       THIS PROGRAM IS CALLED BY ISO_TMF AFTER SELECTION OF THE       *
110!    *       ISOCG OPTION.                                                  *
120!    *                                                                      *
130!    *       THE CYCLIC DATA(Stresses, strains and potentials) ARE          *
140!    *       RECORDED EVERY 5 CYCLES.                                       *
150!    *                                                                      *
160!    ************************************************************************
170     !
180     OPTION BASE 0
190     CALL Isocg(Material$,Contr_variable,Scale,Mean,Span,Gauge,Frequency)
200     END
210!***************************************************************************
220!***************************************************************************
230     !
240     SUB Isocg(Material$,Contr_variable,Scale,Mean,Span,Gauge,Frequency)
250     MASS STORAGE IS ":HP8290X,700,1"
260     !
270     !______ ______________DEFINE I/O CARD FUNCTION__________________________
280     !                    ------------------------
290     ASSIGN @Multi TO 723
300     ASSIGN @Dvm TO 720
310     ASSIGN @Disc TO "MRZ"
320     Disc_drive=0
330     Multi=723
340     Scanner=0
350     A_to_d=3
360     Memory1=7
370     D_to_a1=10
380     Timer=11
390     D_to_a2=12
400     Memory2=13
410     !
420     !_________________DEFAULT VALUES FOR THE PARAMETERS__________________
430     !                 --------------------------------
440     Start=1                                                  !Start channel
450     Stop=2                                                   !Stop channel
460     Cycle_per_block=1
470     Data_per_cycle=200
480     Gauge_factor=.05493
490     !
500     !_____________________INITIALIZE THE MULTI___________________________
510     !
520     CALL Init_multi(@Multi,@Disc)
530     CALL Init_mem(@Multi,Memory1)
540     !
```

```
550   !____________________INITIALIZE CONTROL PROCEDURE________________________
560   !                    ---------------------------
570   CALL Init_control(@Multi,Memory2,Timer,D_to_a1,Contr_variable,Scale,Mean,S
pan,Gauge,Gauge_factor,Frequency,Ndata,Pace,Paces)
580   OUTPUT @Dvm;"F1R-2N4T1Z0D3  ISOCG"
590   !
600   !____________________INITIALIZE READING PROCEDURE________________________
610   !                    ---------------------------
620   CALL Init_read(Cycle_per_block,Data_per_cycle,Frequency,Pace1,Pace2,Ref_co
unt)
630   ALLOCATE Scan(Ref_count),Pot((Ref_count+1)/2-1)
640   !                ----------------------------------------
650   BEEP
660   Cycle=1                               !Set cycle counter
670   Next_cycle=3                          !Set Next_cycle counter
680   G=SPOLL(723)                          !Clear Multi SRQ
690   PRINT USING 700
700   IMAGE 2/,15X,"REMOVE CONTROL AND UTILITY PAC DISCS",2/,15X,"INSERT TWO DIS
CS FOR DATA STORAGE",2/,15X,"PRESS K0(Continue) TO CONTINUE"
710   ON KEY 0 LABEL "**CONTINUE**" GOTO 730
720 Spin: GOTO Spin
730         OFF KEY
740         CREATE BDAT "TEST_PARAM",1,100
750         ASSIGN @Disc TO "TEST_PARAM"
760         OUTPUT @Disc;Material$,"ISOCG",Contr_variable,Scale,Mean,Span,Gauge,Fr
equency
770         ASSIGN @Disc TO *
780         CALL Plot_frame(Material$,Contr_variable,Mean,Span,Frequency)
790         DELSUB Plot_frame
800         PRINT USING 810
810         IMAGE 20/,25X,"MULTIPROGRAMMER IS READY",2/,20X,"****WAITING FOR THE S
TART COMMAND****"
820         ON KEY 0 LABEL " START CYCLING" GOTO Start_iso
830 Wait:   GOTO Wait
840   !
850   !_____________________START CYCLING AND TAKING DATA______________________
860   !                     ----------------------------
870 Start_iso:!
880   OFF KEY
890   !
900   !__________READ DATA____________
910   !          ---------
920   OUTPUT @Multi;"WF",Timer+.2,0,"T"
930   IF Frequency>.185 THEN OUTPUT @Multi;"WF",Timer,Paces,"T"
940   OUTPUT @Multi;"WF",Timer+.2,1,"T"
950   CALL Read_data(@Multi,@Dvm,Frequency,Timer,Pace,Paces,Scanner,Memory1,Pace
1,Pace2,Ref_count,Scan(*),Pot(*))
960   !__________STORE DATA___________
970   !          ----------
980   CALL Store_data(@Disc,Disc_drive,Cycle,Ref_count,Scan(*),Pot(*))
990   !
1000  !__________DISPLAY DATA__________
1010  !          ------------
1020  CALL Plot_data(Contr_variable,Scale,Span,Gauge,Gauge_factor,Cycle,Ref_coun
t,Scan(*),Pot(*),Exponent,Min,Vo)
```

```
1030  !
1040  !__________SET WAITING FOR SYNCHRO. SIGNAL_________
1050  !          ------------------------------
1060 Synchro:                                       !
1070  OUTPUT @Multi;"WF",Timer+.2,0,"T"             !Stop cycling
1080  ON INTR 7 GOTO Complete
1090  ENABLE INTR 7;2
1100  OUTPUT @Multi;"RV",Memory2+1,Memory2+1.1,Memory2+1.2,Memory2+1.3,"T"
1110  ENTER 72306;A1,B1,C1,D1                       !Read read and ref.pointers
1120  OUTPUT @Multi;"WF",Memory2+1.1,C1-A1,"T"      !Set diff. counter
1130  OUTPUT @Multi;"WF",Memory2+.1,2,"T"           !Set FIFO mode
1140  OUTPUT @Multi;"WF",Memory2+1.0,1,"T"          !Set ref. word
1150  OUTPUT @Multi;"WF",Memory2+1.3,A1,"T"         !set read pointer
1160  OUTPUT @Multi;"AC",Memory2+1,"T"              !Arm card
1170  OUTPUT @Multi;"WF",Timer+.2,1,"T"             !Set continuous mode
1180  OUTPUT @Multi;"CY",Timer,"T"                  !Resume cycling
1190 Spin2:    GOTO Spin2
1200 Complete: !
1210  OUTPUT @Multi;"WF",Timer+.2,0,"T"             !Set one-shot mode
1220  OUTPUT @Multi;"DC",Memory2+1,"T"              !Disarm card
1230  OUTPUT @Multi;"RC"                            !Read real-time clock(T2)
1240  ENTER 72314;A1,B1,C1,D1                       !Enter real-time clock(T1)
1250  ENTER 72314;A2,B2,C2,D2                       !Enter real-time clock(T2)
1260  T1=60*(A1*60*24+B1*60+C1)+D1
1270  T2=60*(A2*60*24+B2*60+C2)+D2
1280  Time_delay=T2-T1                              !Compute elapsed time
1290  Wait_time=(Next_cycle-Cycle-1)/Frequency      !Wait between read data
1300  IF Wait_time<Time_delay THEN
1310     Cycle=INT(Cycle+1.5+Time_delay*Frequency)
1320     Next_cycle=Cycle+5
1330     IF Cycle<7 THEN Next_cycle=Cycle+2
1340     OUTPUT @Multi;"CC",Memory2+1,"T"              !Clear card
1350     OUTPUT @Multi;"WF",Memory2+.1,10,"T"          !Set re-circulate mode
1360     OUTPUT @Multi;"WF",Memory2+1.0,Ndata,"T"      !Set ref. word
1370     OUTPUT @Multi;"WF",Memory2+1.2,Ndata+1,"T"    !Set write pointer
1380     OUTPUT @Multi;"WF",Memory2+1.3,0,"T"          !Set read pointer
1390     OUTPUT @Multi;"WF",Timer+.2,1,"T"             !Set continuous mode
1400     GOTO 910
1410  ELSE
1420     Cycle=Next_cycle
1430     Next_cycle=Cycle+5
1440     IF Cycle<7 THEN Next_cycle=Cycle+2
1450     Waitime=Wait_time-INT(Time_delay*Frequency+.5)/Frequency
1460     OUTPUT @Multi;"CC",Memory2+1,"T"              !Clear card
1470     OUTPUT @Multi;"WF",Memory2+.1,10,"T"          !Set re-circulate mode
1480     OUTPUT @Multi;"WF",Memory2+1.0,Ndata,"T"      !Set ref. word
1490     OUTPUT @Multi;"WF",Memory2+1.2,Ndata+1,"T"    !Set write pointer
1500     OUTPUT @Multi;"WF",Memory2+1.3,0,"T"          !Set read pointer
1510     OUTPUT @Multi;"WF",Timer+.2,1,"T"             !Set continuous mode
1520     OUTPUT @Multi;"CY",Timer,"T"                  !Resume cycling
1530     WAIT Waitime                                  !Wait next read data
1540     GOTO 910
1550  END IF
1560  SUBEND
```

```
1570!_______________________________________________________________________
1580!
1590!  ***************************************************************
1600!  *                                                             *
1610!  *       SUBROUTINES USED BY SUB ISOCG(Param1,Param2,...)      *
1620!  *                                                             *
1630!  ***************************************************************
1640!
1650!_______________________________________________________________________
1660  SUB Init_control(@Multi,Memory,Timer,D_to_a,Control,Scale,Mean,Span,Gauge,
Gauge_factor,Frequency,Ndata,Pace,Paces)
1670     !__GENERATE WAVEFORM__
1680     ! -----------------
1690     Load_scale=INT(Scale)/100
1700     Strain_scale=Scale-(Load_scale*100)
1710     IF Control=1 THEN
1720        Pmean=Mean/(Load_scale*216.078)       !Convert MPA into voltage
1730        Pmax=Span/(Load_scale*216.078)
1740     ELSE
1750        Emax=Gauge_factor*Strain_scale/Gauge
1760        Pmean=10*Mean/Emax                    !Convert Strain into voltage
1770        Pmax=10*Span/Emax
1780     END IF
1790     Coeff=800
1800     IF Pmax>5.12 THEN Coeff=400
1810     Ndata=INT(Coeff*Pmax+.5)-1
1820     ALLOCATE A(Ndata)
1830     Ndata2=INT(Ndata/4+.5)
1840     Ndata3=Ndata-Ndata2
1850     FOR I=0 TO Ndata2-1
1860       A(I)=Pmax*I/(Ndata2-1)
1870       A(I+Ndata3)=Pmax*(I/(Ndata2-1)-1)
1880     NEXT I
1890     FOR I=Ndata2 TO Ndata3
1900       A(I)=(Ndata-1-2*I)*Pmax/(Ndata+1-2*Ndata2)
1910     NEXT I
1920     !
1930     !__CALCULATE THE PACE__
1940     ! ------------------
1950     Time_int=1/(Ndata*Frequency)
1960     Pace=INT(Time_int*500000)/1000            !Pace in msec
1970     Paces=Pace
1980     IF Frequency>.185 THEN Paces=INT(500000/(Ndata*.185))/1000
1990     !
2000     !__PROGRAM MULTI__AMPLITUDE
2010     ! ------------------------
2020     OUTPUT @Multi;"CC",Memory+1,"T"           !Clear memory card
2030     OUTPUT @Multi;"SF",Memory,3,1,.005,12,"T" !Set format(LSB=0.005)
2040     OUTPUT @Multi;"WF",Memory+1.0,Ndata,"T"   !Truncate memory
2050     OUTPUT @Multi;"MO",Memory,A(*),"T"        !Load data into memory
2060     OUTPUT @Multi;"WF",Memory+.1,10,"T"       !Set re-circulate mode
2070     OUTPUT @Multi;"WF",Timer+.2,1,"T"         !Set timer for continuous
2080     OUTPUT @Multi;"WF",Timer,Pace,"T"         !output. Set Pace
2090     !
```

```
2100    !__PROGRAM MULTI-- MEAN
2110    ! -------------------
2120    Incr=.005
2130    IF Pmean=0 THEN GOTO 2210
2140    IF Pmean<0 THEN Incr=-.005
2150    J=0
2160    OUTPUT @Multi;"OP",D_to_a,J,"T"
2170    IF ABS(J)<ABS(Pmean) THEN
2180       J=J+Incr
2190       GOTO 2160
2200    END IF
2210    DEALLOCATE A(*)
2220    SUBEND
2230!-------------------------------------------------------------------------
2240  SUB Read_data(@Multi,@Dvm,Frequency,Timer,Pace,Paces,Scanner,Memory,Pace1,
Pace2,Ref_count,Scan(*),Pot(*))
2250    G=SPOLL(@Multi)
2260    ENTER 72310;A,B,C,D,E,F
2270    ON INTR 7 GOTO Interrupt                        !Set interrupt branch
2280    ENABLE INTR 7;2                                 !Enable Interrupt
2290    OUTPUT @Multi;"CY",Scanner,"T"                  !Enable scanner
2300    OUTPUT @Multi;"WF",Scanner+.3,5,"T"             !Set sequential mode
2310    !
2320    OUTPUT @Multi;"WF",Scanner,1,"T"                !Set start channel
2330    OUTPUT @Multi;"WF",Scanner+.1,2,"T"             !set stop channel
2340    OUTPUT @Multi;"WF",Scanner+.2,Pace1,"T"         !Set pace(in usec)
2350    Prog_diff(@Multi,Memory,0)                      !Reset diff. counter
2360    Prog_write(@Multi,Memory,0)                     !Reset write pointer
2370    OUTPUT @Multi;"CC";Memory,"T"                   !Clear Memory card
2380    Prog_mode(@Multi,Memory,56)                     !Set FIFO mode
2390    Prog_ref1(@Multi,Memory,Ref_count)              !Set stop pointer
2400    OUTPUT @Multi;"AC";Memory;"T"                   !Armed memory card
2410    OUTPUT @Multi;"SC,0,0,0,0T"                     !Reset clock
2420    OUTPUT @Multi;"CY",Timer,"T"                    !Start cycling
2430    OUTPUT @Multi;"CY",Scanner,"T"                  !Start pacer
2440    FOR I=0 TO (Ref_count+1)/2-1
2450      ENTER @Dvm;Pot(I)                             !Read potential
2460      WAIT Pace2
2470    NEXT I
2480 Spin:    GOTO Spin
2490 Interrupt:      !
2500      OUTPUT @Multi;"WF",Timer+.2,0,"T"             !Check pace
2510      OUTPUT @Multi;"RC"                            !Read real-time clock(T1)
2520      IF Frequency>.185 THEN OUTPUT @Multi;"WF",Timer,Pace,"T"
2530      OUTPUT @Multi;"WF",Timer+.2,1,"T"             !Set continuous mode
2540      OUTPUT @Multi;"CY",Timer,"T"                  !Continue cycling
2550      G=SPOLL(@Multi)
2560      IF G=64 THEN
2570        ENTER 72310;A,B,C
2580        IF C<>0 THEN
2590           ENTER 72312;Address
2600           IF Address<>Memory THEN
2610              PRINT "NOT MEMORY CARD"
2620              PAUSE
2630           END IF
```

```
2640            OUTPUT @Multi;"WF",Scanner+.2,0,"T"      !Stop scanner
2650            Prog_mode(@Multi,Memory,76)              !Set FIFO lockout
2660            OUTPUT @Multi;"MR";Memory;Ref_count+1;"T"!MR command to get data
2670            ENTER 72305 USING "%,W";Scan(*)          !Entering the data
2680         ELSE
2690            BEEP 2000,.3
2700            PRINT "          ***  SRQ NOT SET BY ARMED CARD  ***"
2710            PAUSE
2720         END IF
2730       ELSE
2740         BEEP 2000,.3
2750         PRINT "          ***  MULTI DID NOT INTERRUPT  ***"
2760         PAUSE
2770       END IF
2780     SUBEND
2790!_______________________________________________________________________
2800     SUB Store_data(@Disc,Disc_drive,Cycle,Ref_count,Scan(*),Pot(*))
2810     ALLOCATE X(199)
2820     FOR I=0 TO Ref_count STEP 2
2830       X(I/2)=INT((Scan(I)+10)*1000)+(Scan(I+1)+10)/100
2840     NEXT I
2850    ON ERROR GOTO Recover
2860    CREATE BDAT "CYCLE"&VAL$(Cycle),1,4000
2870    ASSIGN @Disc TO "CYCLE"&VAL$(Cycle)
2880    OUTPUT @Disc;Cycle,X(*),Pot(*)
2890    ASSIGN @Disc TO *
2900    GOTO Out
2910    !
2920 Recover: !
2930      IF ERRN=59 OR ERRN=64 THEN
2940         IF Disc_drive=0 THEN
2950            Disc_drive=1
2960            MASS STORAGE IS ":HP8290X,700,1"
2970         ELSE
2980            Disc_drive=0
2990            MASS STORAGE IS ":HP8290X,700,0"
3000         END IF
3010      ELSE
3020         BEEP 2000,1
3030         PRINT USING 3040;ERRN
3040         IMAGE 2/,25X,"ERROR NUMBER ",K,/,25X,"CHECK STORAGE UNIT",/,25X,"PROG
RAM ABORTED"
3050         PAUSE
3060      END IF
3070      GOTO 2860
3080 Out:         !
3090      DEALLOCATE X(*)
3100      SUBEND
3110!_______________________________________________________________________
3120 SUB Init_multi(@Multi,@Disc)
3130    CLEAR 7
3140    WAIT 4.0
3150    G=SPOLL(@Multi)
```

```
3160   IF B<>64 THEN
3170      PRINT "        ***MULTI DID NOT INTERRUPT***"
3180      PAUSE
3190   END IF
3200   STATUS @Multi,3;Multi
3210   ENTER Multi*100+10;A
3220   IF A<>16384 THEN
3230     PRINT "        ***SELF TEST DID NOT SET SRQ***"
3240     PAUSE
3250   END IF
3260  ALLOCATE Ascii$[80]
3270  ON END @Disc GOTO Eof
3280 Rd_file:    ENTER @Disc;Ascii$
3290    OUTPUT @Multi;Ascii$
3300    GOTO Rd_file
3310 Eof: OFF END @Disc
3320    ASSIGN @Disc TO *
3330    DEALLOCATE Ascii$
3340    MASS STORAGE IS ":HP8290X,700,0"
3350 SUBEND
3360!_______________________________________________________________________
3370 SUB Init_mem(@Multi,Memory)
3380   Prog_mode(@Multi,Memory,254)
3390   Prog_mode(@Multi,Memory,54)
3400   Prog_ref1(@Multi,Memory,1048575)
3410   Prog_ref2(@Multi,Memory,0)
3420   Prog_read(@Multi,Memory,0)
3430   Prog_diff(@Multi,Memory,0)
3440   Prog_write(@Multi,Memory,0)
3450   OUTPUT @Multi;"CC";Memory;"T"
3460 SUBEND
3470!_______________________________________________________________________
3480 SUB Prog_read(@Multi,Memory,Read_pointer)
3490   Read_low=Read_pointer MOD 65536
3500   Read_high=Read_pointer DIV 65536
3510   Read_low_oct=FNDec_to_octal(Read_low)
3520   Read_high_oct=FNDec_to_octal(Read_high)
3530   OUTPUT @Multi;"WF";Memory+.1;22;Memory+.2;Read_low_oct;"T"
3540   OUTPUT @Multi;"WF";Memory+.1;23;Memory+.2;Read_high_oct;"T"
3550 SUBEND
3560!_______________________________________________________________________
3570 SUB Prog_write(@Multi,Memory,Write_pointer)
3580   Write_low=Write_pointer MOD 65536
3590   Write_high=Write_pointer DIV 65536
3600   Write_low_oct=FNDec_to_octal(Write_low)
3610   Write_high_oct=FNDec_to_octal(Write_high)
3620   OUTPUT @Multi;"WF";Memory+.1;24;Memory+.2,Write_low_oct;"T"
3630   OUTPUT @Multi;"WF";Memory+.1;25;Memory+.2;Write_high_oct;"T"
3640 SUBEND
3650!_______________________________________________________________________
3660 SUB Prog_diff(@Multi,Memory,Diff_counter)
3670   Diff_low=Diff_counter MOD 65536
3680   Diff_high=Diff_counter DIV 65536
3690   Diff_low_oct=FNDec_to_octal(Diff_low)
```

```
3700   Diff_high_oct=FNDec_to_octal(Diff_high)
3710   OUTPUT @Multi;"WF";Memory+.1;20;Memory+.2;Diff_low_oct;"T"
3720   OUTPUT @Multi;"WF";Memory+.1;21;Memory+.2;Diff_high_oct;"T"
3730 SUBEND
3740!____________________________________________________________________
3750 SUB Prog_ref1(@Multi,Memory,Ref1_reg)
3760   Ref1_low=Ref1_reg MOD 65536
3770   Ref1_high=Ref1_reg DIV 65536
3780   Ref1_low_oct=FNDec_to_octal(Ref1_low)
3790   Ref1_high_oct=FNDec_to_octal(Ref1_high)
3800   OUTPUT @Multi;"WF";Memory+.1,26;Memory+.2;Ref1_low_oct;"T"
3810   OUTPUT @Multi;"WF";Memory+.1,27;Memory+.2;Ref1_high_oct;"T"
3820 SUBEND
3830!____________________________________________________________________
3840 SUB Prog_ref2(@Multi,Memory,Ref2_reg)
3850   Ref2_low=Ref2_reg MOD 65536
3860   Ref2_high=Ref2_reg DIV 65536
3870   Ref2_low_oct=FNDec_to_octal(Ref2_low)
3880   Ref2_high_oct=FNDec_to_octal(Ref2_high)
3890   OUTPUT @Multi;"WF";Memory+.1;30;Memory+.2;Ref2_low_oct;"T"
3900   OUTPUT @Multi;"WF";Memory+.1;31;Memory+.2;Ref2_high_oct;"T"
3910 SUBEND
3920!____________________________________________________________________
3930 SUB Prog_mode(@Multi,Memory,Mode)
3940   OUTPUT @Multi;"WF";Memory+.3;Mode;"T"
3950 SUBEND
3960!____________________________________________________________________
3970 SUB Read_status(@Multi,Memory,Status)
3980   STATUS @Multi,3;Multi
3990   OUTPUT @Multi;"WF";Memory+.1;1;"T RV";Memory+.2;"T"
4000   ENTER Multi*100+6;Status
4010 SUBEND
4020!____________________________________________________________________
4030 SUB Read_read(@Multi,Memory,Read_pointer)
4040   STATUS @Multi,3;Multi
4050   OUTPUT @Multi;"WF";Memory+.1;22;"T RV";Memory+.2;"T"
4060   ENTER Multi*100+6;Read_low_oct
4070   OUTPUT @Multi;"WF";Memory+.1;23;"T RV";Memory+.2;"T"
4080   ENTER Multi*100+6;Read_high_oct
4090   Read_low=FNOctal_to_dec(Read_low_oct)
4100   Read_high=FNOctal_to_dec(Read_high_oct)
4110   Read_pointer=65536*Read_high+Read_low
4120 SUBEND
4130!____________________________________________________________________
4140 SUB Read_write(@Multi,Memory,Write_pointer)
4150   STATUS @Multi,3;Multi
4160   OUTPUT @Multi;"WF";Memory+.1;24;"T RV";Memory+.2;"T"
4170   ENTER Multi*100+6;Write_low_oct
4180   OUTPUT @Multi;"WF";Memory+.1;25;"T RV";Memory+.2;"T"
4190   ENTER Multi*100+6;Write_high_oct
4200   Write_low=FNOctal_to_dec(Write_low_oct)
4210   Write_high=FNOctal_to_dec(Write_high_oct)
4220   Write_pointer=65536*Write_high+Write_low
```

```
4230 SUBEND
4240!_______________________________________________________________________________
4250 SUB Read_diff(@Multi,Memory,Diff_counter)
4260   STATUS @Multi,3;Multi
4270   OUTPUT @Multi;"WF";Memory+.1;20;"T RV";Memory+.2;"T"
4280   ENTER Multi*100+6;Diff_low_oct
4290   OUTPUT @Multi;"WF";Memory+.1;21;"T RV";Memory+.2;"T"
4300   ENTER Multi*100+6;Diff_high_oct
4310   Diff_low=FNOctal_to_dec(Diff_low_oct)
4320   Diff_high=FNOctal_to_dec(Diff_high_oct)
4330   Diff_counter=65536*Diff_high+Diff_low
4340 SUBEND
4350!_______________________________________________________________________________
4360 DEF FNDec_to_octal(Dec)
4370   Oct=Dec+2*INT(Dec/8)+20*INT(Dec/64)+200*INT(Dec/512)+2000*INT(Dec/4096)+2
0000*INT(Dec/32768)+200000*INT(Dec/262144)
4380   RETURN Oct
4390 FNEND
4400!_______________________________________________________________________________
4410 DEF FNOctal_to_dec(Octal)
4420   X9=0
4430   X7=Octal
4440   X=0
4450 More:   !
4460   X=X+(X7-INT(X7/10)*10)*8^X9
4470   X7=INT(X7/10)
4480   X9=X9+1
4490   IF X7<>0 THEN GOTO More
4500   RETURN X
4510 FNEND
4520!_______________________________________________________________________________
4530  SUB Plot_frame(Material$,Contr_variable,Mean,Span,Frequency)
4540  ALLOCATE A(20),B(20)
4550  FOR I=0 TO 4
4560    A(I)=25*I-50
4570  NEXT I
4580  FOR I=0 TO 20
4590    B(I)=3*I-40
4600  NEXT I
4610  DEG
4620  GINIT
4630  GCLEAR
4640  ALPHA ON
4650  GRAPHICS OFF
4660  FRAME
4670  WINDOW -66.7224080268,66.7224080268,-50,50
4680  MOVE -50,-40
4690  FOR I=0 TO 3
4700    FOR J=1 TO 9
4710      DRAW 25*LGT(J*10^I)-50,-40
4720      IF J=1 THEN
4730        IDRAW 0,1
4740        IMOVE 0,-1
```

```
4750      ELSE
4760         IDRAW 0,.5
4770         IMOVE 0,-.5
4780      END IF
4790    NEXT J
4800  NEXT I
4810  C=.5
4820  D=1.0
4830  FOR I=0 TO 20
4840    DRAW 50,B(I)
4850    IF 2*INT(I/2)=I THEN
4860       IDRAW -D,0
4870       IMOVE D,0
4880    ELSE
4890       IDRAW -C,0
4900       IMOVE C,0
4910    END IF
4920  NEXT I
4930  FOR I=3 TO 0 STEP -1
4940    FOR J=9 TO 1 STEP -1
4950      DRAW 25*LGT(J*10^I)-50,20
4960      IF J=1 THEN
4970        IDRAW 0,-1
4980        IMOVE 0,1
4990      ELSE
5000        IDRAW 0,-.5
5010        IMOVE 0,.5
5020      END IF
5030    NEXT J
5040  NEXT I
5050  FOR I=20 TO 0 STEP -1
5060    DRAW -50,B(I)
5070    IF 2*INT(I/2)=I THEN
5080       IDRAW D,0
5090       IMOVE -D,0
5100    ELSE
5110       IDRAW C,0
5120       IMOVE -C,0
5130    END IF
5140  NEXT I
5150  MOVE -50,-10
5160  FOR I=0 TO 3
5170    FOR J=1 TO 9
5180      DRAW 25*LGT(J*10^I)-50,-10
5190      IF J=1 THEN
5200        IDRAW 0,1
5210        IDRAW 0,-2
5220        IMOVE 0,1
5230      ELSE
5240        IDRAW 0,.5
5250        IDRAW 0,-1
5260        IMOVE 0,.5
5270      END IF
5280    NEXT J
```

```
5290  NEXT I
5300  MOVE 50,20
5310  DRAW 50,40
5320  DRAW -50,40
5330  DRAW -50,20
5340  CSIZE 3,.6
5350  MOVE -50,-44
5360  LABEL VAL$(1)
5370  FOR I=1 TO 4
5380    MOVE A(I)-3,-44
5390    LABEL VAL$(10^I)
5400  NEXT I
5410  FOR I=0 TO 8 STEP 2
5420    MOVE -56,3*I-41
5430    LABEL USING "D.D";I/2
5440    MOVE 51,3*I-41
5450    LABEL USING "D.D";I/2
5460  NEXT I
5470  CSIZE 4,.6
5480  MOVE -5,-48
5490  LABEL "CYCLE"
5500  CSIZE 3,.6
5510  LDIR 90
5520  MOVE -58,-32
5530  IDRAW 0,3
5540  IDRAW -1.5,-1.5
5550  IDRAW 1.5,-1.5
5560  MOVE -57,-28
5570  LABEL "V/Vo"
5580  MOVE -58,-3
5590  IDRAW 0,3
5600  IDRAW -1.5,-1.5
5610  IDRAW 1.5,-1.5
5620  MOVE -57,1
5630  IF Contr_variable=1 THEN
5640     LABEL "E/2(X 10  )"
5650     IMOVE -3.5,14.6
5660     LABEL "-"
5670  ELSE
5680     LABEL "S/2 (MPa)"
5690  END IF
5700  LDIR -90
5710  MOVE 59,16
5720  IDRAW 0,-3
5730  IDRAW 1.5,1.5
5740  IDRAW -1.5,1.5
5750  MOVE 59,12
5760  IF Contr_variable=1 THEN
5770    LABEL "E/2(X 10  )"
5780    IMOVE 3.5,-14.6
5790    LABEL "-"
5800  ELSE
5810    LABEL "S/2 (MPa)"
```

```
5820  END IF
5830  MOVE 59,-20
5840  IDRAW 0,-3
5850  IDRAW 1.5,1.5
5860  IDRAW -1.5,1.5
5870  MOVE 59,-24
5880  LABEL "V/Vo"
5890  LDIR 0
5900  CSIZE 3.5,.6
5910  MOVE -45,34
5920  LABEL Material$
5930  MOVE -45,30
5940  LABEL "ISOCG"
5950  MOVE -45,26
5960  LABEL "Frequency= "&VAL$(Frequency)&" Hz"
5970  IF Contr_variable=1 THEN
5980     MOVE 15,34
5990     LABEL "STRESS CONTROL"
6000     MOVE 15,31
6010     IDRAW 3,0
6020     IDRAW -1.5,1.5
6030     IDRAW -1.5,-1.5
6040     MOVE 19,30
6050     LABEL "S = "&VAL$(Span*2)&" (MPa)"
6060  ELSE
6070     MOVE 15,34
6080     LABEL "STRAIN CONTROL"
6090     MOVE 15,31
6100     IDRAW 3,0
6110     IDRAW -1.5,1.5
6120     IDRAW -1.5,-1.5
6130     MOVE 19,30
6140     LABEL "E = "&VAL$(Span*2)
6150  END IF
6160  MOVE 19,26
6170  LABEL "R = "&VAL$(DROUND((Mean-Span)/(Mean+Span),2))
6180  SUBEND
6190!------------------------------------------------------------------------
6200  SUB Init_read(Cycle_per_block,Data_per_cycle,Frequency,Pace1,Pace2,Ref_cou
nt)
6210  Ref_count=(Cycle_per_block*Data_per_cycle*2)-1
6220  IF Frequency>=.185 THEN
6230     Pace1=INT(1.E+6/(Data_per_cycle*2*.185)-30)        !Pace in usec
6240     Pace2=0.
6250  ELSE
6260     Pace1=INT(1.E+6/(Data_per_cycle*2*Frequency)-30)    !Pace in usec
6270     Pace2=DROUND((1/Frequency-1/37)/200,3)
6280  END IF
6290  SUBEND
6300!------------------------------------------------------------------------
6310  SUB Plot_data(Contr_variable,Scale,Span,Gauge,Gauge_factor,Cycle,Ref_count
,Scan(*),Pot(*),Exponent,Min,Vo)
6320  ALPHA OFF
6330  GRAPHICS ON
6340  CSIZE 3,.6
6350  Indice=1
```

```
6360  Max=0
6370  Maxim=0
6380    IF Contr_variable=1 THEN Indice=0
6390      FOR I=74 TO 124 STEP 2
6400        J=I+Indice
6410        IF Scan(J)*.005>Max THEN Maximum=Scan(J)*.005
6420        IF Scan(J+200)*.005<Min THEN Minimum=Scan(J+200)*.005
6430        IF Pot(I/2)>Maxim THEN Maxim=Pot(I/2)
6440        IF Pot(I/2+1)>Maxim THEN Maxim=Pot(I/2+1)
6450      NEXT I
6460    Range=Maximum-Minimum
6470    Vmax=Maxim
6480    Load_scale=INT(Scale)/100
6490    Strain_scale=Scale-(Load_scale*100)
6500    IF Cycle>1 THEN GOTO Data
6510    CALL Scale_frame(Contr_variable,Scale,Gauge,Gauge_factor,Range,Min,Expon
ent)
6520    Vo=Vmax
6530 Data: !
6540    X=25*LGT(Cycle)-50
6550    IF Contr_variable=1 THEN
6560       Strain_ampli=Range*Strain_scale*Gauge_factor/(Gauge*20)
6570       Y1=4*(Strain_ampli*(10^Exponent)-Min)
6580    ELSE
6590       Stress_ampli=Range*Load_scale*216.078/2
6600       Y1=.2*(Stress_ampli-Min)
6610    END IF
6620    MOVE X-.7,3*Y1-11.3
6630    LABEL "x"
6640    MOVE X-.7,6*(Vmax/Vo)-41.3
6650    LABEL "x"
6660   SUBEND
6670!-----------------------------------------------------------------------
6680   SUB Scale_frame(Contr_variable,Scale,Gauge,Gauge_factor,Range,Min,Exponen
t)
6690   Load_scale=INT(Scale)/100
6700   Strain_scale=Scale-(Load_scale*100)
6710   IF Contr_variable=1 THEN
6720      Strain_ampli=Range*Strain_scale*Gauge_factor/(Gauge*20)
6730      FOR Multiplier=0 TO 5
6740        IF Strain_ampli*10^Multiplier>1 AND Strain_ampli*10^Multiplier<10 TH
EN Exponent=Multiplier
6750      NEXT Multiplier
6760      A=Strain_ampli*10^Exponent
6770      IF A<2.5 THEN
6780        Min=0.
6790      ELSE
6800        IF A<5.0 THEN
6810          Min=2.5
6820        ELSE
6830          IF A<7.5 THEN
6840            Min=5.0
6850          ELSE
6860            Min=7.5
```

```
6870            END IF
6880          END IF
6890        END IF
6900      ELSE
6910        Stress_ampli=Range*Load_scale*216.078/2
6920        Norme=INT(Stress_ampli/100)*100
6930        A=Stress_ampli-Norme
6940        IF A<25 THEN
6950          Min=Norme
6960          IF A<12.5 THEN Min=Norme-25
6970        ELSE
6980          IF A<50 THEN
6990            Min=Norme+25
7000            IF A<37.5 THEN Min=Norme
7010          ELSE
7020            IF A<75 THEN
7030              Min=Norme+50
7040              IF A<62.5 THEN Min=Norme+25
7050            ELSE
7060              Min=Norme+75
7070              IF A<77.5 THEN Min=Norme+50
7080            END IF
7090          END IF
7100        END IF
7110      END IF
7120      A$="3D"
7130      Facto=5
7140      IF Contr_variable=1 THEN
7150        A$="D.D"
7160        Facto=.50
7170      END IF
7180      FOR I=2 TO 10 STEP 2
7190        MOVE -56,3*I-11.5
7200        LABEL USING A$;(Facto*I+Min)
7210        MOVE 51,3*I-11.5
7220        LABEL USING A$;(Facto*I+Min)
7230      NEXT I
7240      CSIZE 3,.6
7250      LDIR 90
7260      IF Contr_variable=1 THEN
7270        MOVE -57,1
7280        IMOVE -1,16.6
7290        LABEL USING "D";Exponent
7300        LDIR -90
7310        MOVE 59,12
7320        IMOVE .5,-16.6
7330        LABEL USING "D";Exponent
7340      END IF
7350      LDIR 0
7360    SUBEND
```

```
10!   ***************************************************************************
20!   *                                                                         *
30!   *                       PROGRAM ISOLCF                                    *
40!   *                       **************                                    *
50!   *                                                                         *
60!   *       THIS PROGRAM PERFORMS THE REAL-TIME CONTROL AND DATA              *
70!   *       ACQUISITION OF ISOTHERMAL LOW CYCLE FATIGUE TESTS.                *
80!   *                                                                         *
90 !  *       THIS PROGRAM IS CALLED BY ISO_TMF AFTER SELECTION OF THE          *
100!  *       ISOLCF OPTION.                                                    *
110!  *                                                                         *
120!  *       THE CYCLIC DATA(Stresses, and strains) ARE RECORDED               *
130!  *       (N2/N1)=1.25 CYCLES.                                              *
140!  *                                                                         *
150!  ***************************************************************************
160   !
170   OPTION BASE 0
180   CALL Isolcf(Material$,Contr_variable,Scale,Mean,Span,Gauge,Frequency)
190   END
200!******************************************************************************
210!******************************************************************************
220   !
230   SUB Isolcf(Material$,Contr_variable,Scale,Mean,Span,Gauge,Frequency)
240   MASS STORAGE IS ":HP8290X,700,1"
250   !
260   !______ _______________DEFINE I/O CARD FUNCTION_________________________
270   !                     -----------------------
280   ASSIGN @Multi TO 723
290   ASSIGN @Disc TO "MRZ"
300   Disc_drive=0
310   Multi=723
320   Scanner=0
330   A_to_d=3
340   Memory1=7
350   D_to_a1=10
360   Timer=11
370   D_to_a2=12
380   Memory2=13
390   Ndata=0
400   Pace=0
410   !
420   !_________________DEFAULT VALUES FOR THE PARAMETERS____________________
430   !                 --------------------------------
440   Start=1                                               !Start channel
450   Stop=2                                                !Stop channel
460   Pointer=1                                             !Interrupt word
470   Cycle_per_block=12
480   Data_per_cycle=200
490   Gauge_factor=.05493
500   !
510   !_____________________INITIALIZE THE MULTI______________________________
520   !
530   CALL Init_multi(@Multi,@Disc)
540   CALL Init_mem(@Multi,Memory1)
```

```
550    !
560    !____________________INITIALIZE CONTROL PROCEDURE______________________
570    !                    ----------------------------
580    CALL Init_control(@Multi,Memory2,Timer,D_to_a1,Contr_variable,Scale,Mean,S
pan,Gauge,Gauge_factor,Frequency,Ndata)
590    !
600    !____________________INITIALIZE READING PROCEDURE______________________
610    !                    ----------------------------
620    CALL Init_read(Frequency,Pace,Cycle_per_block,Data_per_cycle,Ref_count)
630    ALLOCATE Scan(Ref_count)
640    !
650    !_______________SET INTERRUPT BRANCH FOR THE START SIGNAL_____________
660    !               ----------------------------------------
670    BEEP
680    Cycle=1                              !Set cycle counter
690    Next_cycle=2                         !Set Next_cycle counter
700    PRINT USING 710
710    IMAGE 2/,15X,"REMOVE CONTROL AND UTILITY PAC DISCS",2/,15X,"INSERT TWO DIS
CS FOR DATA STORAGE",2/,15X,"PRESS K0(Continue) TO CONTINUE"
720    ON KEY 0 LABEL "**CONTINUE**" GOTO 740
730 Spin: GOTO Spin
740         OFF KEY
750         CREATE BDAT "TEST_PARAM",1,100
760         ASSIGN @Disc TO "TEST_PARAM"
770         OUTPUT @Disc;Material$,"ISOLCF",Contr_variable,Scale,Mean,Span,Gauge,F
requency
780         ASSIGN @Disc TO *
790         CALL Plot_frame(Material$,Contr_variable,Mean,Span,Frequency)
800         DELSUB Plot_frame
810         PRINT USING 820
820         IMAGE 20/,25X,"MULTIPROGRAMMER IS READY",2/,20X,"****WAITING FOR THE S
TART COMMAND****"
830         ON KEY 0 LABEL " START CYCLING" GOTO Start_iso
840 Wait:   GOTO Wait
850    !
860    !____________________START CYCLING AND TAKING DATA_____________________
870    !                    -----------------------------
880 Start_iso:!
890    OFF KEY
900    OUTPUT @Multi;"SC,0,0,0,0T"              !Reset Clock of Multi
910    OUTPUT @Multi;"CY",Timer,"T"             !Start cycling
920    !
930    !__________READ DATA___________
940    !          ---------
950    CALL Read_data(@Multi,Multi,Scanner,Memory1,Pace,Ref_count,Scan(*))
960    !__________STORE DATA__________
970    !          ----------
980    CALL Store_data(@Disc,Disc_drive,Cycle,Next_cycle,Ref_count,Scan(*))
990    !
1000   !__________DISPLAY DATA_________
1010   !          ------------
1020   CALL Plot_data(Contr_variable,Scale,Span,Gauge,Gauge_factor,Cycle,Next_cyc
le,Ref_count,Scan(*),Exponent,Exp,Min,Mini)
```

```
1030  !
1040  !__________SET WAITING FOR SYNCHRO. SIGNAL_________
1050  !          ------------------------------
1060 Synchro:!
1070  OUTPUT @Multi;"WF",Timer+.2,0,"T"          !Stop cycling
1080  ON INTR 7 GOTO Complete
1090  ENABLE INTR 7;2
1100  OUTPUT @Multi;"RV",Memory2+1,Memory2+1.1,Memory2+1.2,Memory2+1.3,"T"
1110  ENTER 72306;A1,B1,C1,D1                    !Read card pointers
1120  OUTPUT @Multi;"WF",Memory2+1.1,C1-A1,"T"   !Set differential counter
1130  OUTPUT @Multi;"WF",Memory2+.1,2,"T"        !Set FIFO output mode
1140  OUTPUT @Multi;"WF",Memory2+1.0,1,"T"       !Set reference word
1150  OUTPUT @Multi;"WF",Memory2+1.3,A1,"T"      !Set read pointer
1160  OUTPUT @Multi;"AC",Memory2+1,"T"           !Arm card for interrupt
1170  OUTPUT @Multi;"WF",Timer+.2,1,"T"          !Reset timer(continuous)
1180  OUTPUT @Multi;"CY",Timer,"T"               !Complete the cycle
1190 Spin2:   GOTO Spin2
1200 Complete: !
1210  OUTPUT @Multi;"WF",Timer+.2,0,"T"          !Stop Timer
1220  OUTPUT @Multi;"DC",Memory2+1,"T"           !Disarm card
1230  OUTPUT @Multi;"RC"                         !Set real-time clock(T2)
1240  ENTER 72314;A1,B1,C1,D1                    !Read real-time clock(T1)
1250  ENTER 72314;A2,B2,C2,D2                    !Read real-time clock(T2)
1260  T1=60*(A1*60*24+B1*60+C1)+D1
1270  T2=60*(A2*60*24+B2*60+C2)+D2
1280  Time_delay=T2-T1
1290  Wait_time=(Next_cycle-Cycle-1)/Frequency   !Wait between cycles
1300  IF Wait_time<Time_delay THEN
1310     Cycle=INT(Cycle+1.5+Time_delay*Frequency)
1320     Next_cycle=INT(Cycle*1.25)
1330     OUTPUT @Multi;"CC",Memory2+1,"T"        !Clear card
1340     OUTPUT @Multi;"WF",Memory2+.1,10,"T"    !Set re-circulate mode
1350     OUTPUT @Multi;"WF",Memory2+1.0,Ndata,"T" !Set ref.  word
1360     OUTPUT @Multi;"WF",Memory2+1.2,Ndata+1,"T"!Set write pointer
1370     OUTPUT @Multi;"WF",Memory2+1.3,0,"T"    !Set read pointer
1380     OUTPUT @Multi;"WF",Timer+.2,1,"T"       !Set continuous mode
1390     OUTPUT @Multi;"CY",Timer,"T"            !Resume cycling
1400     GOTO 940
1410  ELSE
1420     Cycle=Next_cycle
1430     Waitime=Wait_time-((T2-T1)*Frequency)   !Compute wait
1440     OUTPUT @Multi;"CC",Memory2+1,"T"        !Clear card
1450     OUTPUT @Multi;"WF",Memory2+.1,10,"T"    !Set re-circulate mode
1460     OUTPUT @Multi;"WF",Memory2+1.0,Ndata,"T" !Set ref.  word
1470     OUTPUT @Multi;"WF",Memory2+1.2,Ndata+1,"T"!Set write word
1480     OUTPUT @Multi;"WF",Memory2+1.3,0,"T"    !Set read pointer
1490     OUTPUT @Multi;"WF",Timer+.2,1,"T"       !Set continuous mode
1500     OUTPUT @Multi;"CY",Timer,"T"            !Resume cycling
1510     WAIT Waitime                            !Wait for next reading
1520     GOTO 940
1530  END IF
1540  SUBEND
1550!---------------------------------------------------------------------
1560!
```

```
1570! ****************************************************************
1580! *                                                              *
1590! *       SUBROUTINES USED BY SUB ISOLCF(Param1,Param2,...)      *
1600! *                                                              *
1610! ****************************************************************
1620!
1630!_____________________________________________________________________
1640  SUB Init_control(@Multi,Memory,Timer,D_to_a,Control,Scale,Mean,Span,Gauge,
Gauge_factor,Frequency,Ndata)
1650     !__GENERATE WAVEFORM__
1660     ! ----------------
1670     Load_scale=INT(Scale)/100
1680     Strain_scale=Scale-(Load_scale*100)
1690     IF Control=1 THEN
1700         Pmean=Mean/(Load_scale*216.078)      !Convert MPA into voltage
1710         Pmax=Span/(Load_scale*216.078)
1720     ELSE
1730         Emax=Gauge_factor*Strain_scale/Gauge
1740         Pmean=10*Mean/Emax                   !Convert Strain into voltage
1750         Pmax=10*Span/Emax
1760     END IF
1770     Coeff=800
1780     IF Pmax>5.12 THEN Coeff=400
1790     Ndata=INT(Coeff*Pmax+.5)-1
1800     ALLOCATE A(Ndata)
1810     Ndata2=INT(Ndata/4+.5)
1820     Ndata3=Ndata-Ndata2
1830     FOR I=0 TO Ndata2-1
1840       A(I)=Pmax*I/(Ndata2-1)
1850       A(I+Ndata3)=Pmax*(I/(Ndata2-1)-1)
1860     NEXT I
1870     FOR I=Ndata2 TO Ndata3
1880       A(I)=(Ndata-1-2*I)*Pmax/(Ndata+1-2*Ndata2)
1890     NEXT I
1900     !
1910     !__CALCULATE THE PACE__
1920     ! -----------------
1930     Time_int=1/(Ndata*Frequency)
1940     Pace=INT(Time_int*500000)/1000            !Pace in msec
1950     !__PROGRAM MULTI__AMPLITUDE
1960     ! ----------------------
1970     OUTPUT @Multi;"CC",Memory+1,"T"           !Clear memory card
1980     OUTPUT @Multi;"SF",Memory,3,1,.005,12,"T" !Set format(LSB=0.005)
1990     OUTPUT @Multi;"MO",Memory,A(*),"T"        !Load data into memory
2000     OUTPUT @Multi;"WF",Memory+1.0,Ndata,"T"   !Truncate memory
2010     OUTPUT @Multi;"WF",Memory+.1,10,"T"       !Set re-circulate mode
2020     OUTPUT @Multi;"WF",Timer+.2,1,"T"         !Set timer for continuous
2030     OUTPUT @Multi;"WF",Timer,Pace,"T"         !output. Set Pace
2040     !
2050     !__PROGRAM MULTI-- MEAN
2060     ! -------------------
2070     Incr=.005
2080     IF Pmean=0 THEN GOTO 2160
2090     IF Pmean<0 THEN Incr=-.005
```

```
2100    J=0
2110    OUTPUT @Multi;"OP",D_to_a,J,"T"
2120    IF ABS(J)<ABS(Pmean) THEN
2130       J=J+Incr
2140       GOTO 2110
2150    END IF
2160    DEALLOCATE A(*)
2170    SUBEND
2180!-----------------------------------------------------------------------------------
2190   SUB Read_data(@Multi,Multi,Scanner,Memory,Pace,Ref_count,Scan(*))
2200    G=SPOLL(@Multi)
2210    ENTER 72310;A,B,C,D,E,F
2220    ON INTR 7,2 GOTO Interrupt
2230    ENABLE INTR 7;2
2240    OUTPUT @Multi;"CY",Scanner,"T"                !Enable scanner
2250    OUTPUT @Multi;"WF",Scanner+.3,5,"T"           !Set sequential mode
2260    !
2270      OUTPUT @Multi;"WF",Scanner,1,"T"            !Set start channel
2280      OUTPUT @Multi;"WF",Scanner+.1,2,"T"         !set stop channel
2290      OUTPUT @Multi;"WF",Scanner+.2,Pace,"T"      !Set pace(in usec)
2300      Prog_diff(@Multi,Memory,0)                  !Reset diff. counter
2310      Prog_write(@Multi,Memory,0)                 !Reset write pointer
2320      OUTPUT @Multi;"CC";Memory,"T"               !Clear Memory card
2330      Prog_mode(@Multi,Memory,56)                 !Set FIFO mode
2340      Prog_ref1(@Multi,Memory,Ref_count)          !Set stop pointer
2350      OUTPUT @Multi;"AC";Memory;"T"               !Armed memory card
2360      OUTPUT @Multi;"CY",Scanner,"T"              !Start pacer
2370 Wait:   GOTO Wait
2380 Interrupt:       !
2390      OUTPUT @Multi;"RC"                          !Read real-time clock(T1)
2400      G=SPOLL(@Multi)
2410      IF G=64 THEN
2420        ENTER 72310;A,B,C
2430        IF C<>0 THEN
2440           ENTER 72312;Address
2450           IF Address<>Memory THEN
2460              PRINT "NOT MEMORY CARD"
2470              PAUSE
2480           END IF
2490           OUTPUT @Multi;"WF",Scanner+.2,0,"T"       !Stop scanner
2500           Prog_mode(@Multi,Memory,76)               !Set FIFO lockout
2510           OUTPUT @Multi;"MR";Memory;Ref_count+1;"T"!MR command to get data
2520           ENTER 72305 USING "%,W";Scan(*)           !Entering the data
2530        ELSE
2540           BEEP 2000,.3
2550           PRINT "          ***  SRQ NOT SET BY ARMED CARD  ***"
2560           PAUSE
2570        END IF
2580     ELSE
2590        BEEP 2000,.3
2600        PRINT "          ***  MULTI DID NOT INTERRUPT  ***"
2610        PAUSE
2620     END IF
```

```
2630   SUBEND
2640!----------------------------------------------------------------------
2650   SUB Store_data(@Disc,Disc_drive,Cycle,Next_cycle,Ref_count,Scan(*))
2660   ALLOCATE X(199)
2670   Start=0
2680   Stop=399
2690   Init_cycle=Cycle
2700 Transfer:    !
2710   Next_cycle=INT(Init_cycle*1.25)
2720   IF Next_cycle=Init_cycle THEN Next_cycle=Init_cycle+1
2730   FOR I=0 TO 399 STEP 2
2740     J=I+Start
2750     X(I/2)=INT((Scan(J)+10)*1000)+(Scan(J+1)+10)/100
2760  NEXT I
2770  ON ERROR GOTO Recover
2780  CREATE BDAT "CYCLE"&VAL$(Init_cycle),1,4000
2790  ASSIGN @Disc TO "CYCLE"&VAL$(Init_cycle)
2800  OUTPUT @Disc;Init_cycle,X(*)
2810  ASSIGN @Disc TO *
2820 Enough:     !
2830  Init_cycle=Init_cycle+1
2840  Start=Stop+1
2850  Stop=Stop+400
2860  IF Stop>Ref_count THEN GOTO Out
2870  IF Init_cycle<Next_cycle THEN GOTO Enough
2880  GOTO Transfer
2890  !
2900 Recover: !
2910    IF ERRN=59 OR ERRN=64 THEN
2920       IF Disc_drive=0 THEN
2930          Disc_drive=1
2940          MASS STORAGE IS ":HP8290X,700,1"
2950       ELSE
2960          Disc_drive=0
2970          MASS STORAGE IS ":HP8290X,700,0"
2980       END IF
2990    ELSE
3000       BEEP 2000,1
3010       PRINT USING 3020;ERRN
3020       IMAGE 2/,25X,"ERROR NUMBER ",K,/,25X,"CHECK STORAGE UNIT",/,25X,"PROG
RAM ABORTED"
3030       PAUSE
3040    END IF
3050    GOTO 2780
3060 Out:       !
3070    DEALLOCATE X(*)
3080    SUBEND
3090!----------------------------------------------------------------------
3100 SUB Init_multi(@Multi,@Disc)
3110   CLEAR 7
3120   WAIT 4.0
3130   G=SPOLL(@Multi)
3140   IF G<>64 THEN
3150      PRINT "      ***MULTI DID NOT INTERRUPT***"
```

```
3160        PAUSE
3170    END IF
3180    STATUS @Multi,3;Multi
3190    ENTER Multi*100+10;A
3200    IF A<>16384 THEN
3210      PRINT "         ***SELF TEST DID NOT SET SRQ***"
3220      PAUSE
3230    END IF
3240   ALLOCATE Ascii$[80]
3250   ON END @Disc GOTO Eof
3260 Rd_file:    ENTER @Disc;Ascii$
3270     OUTPUT @Multi;Ascii$
3280     GOTO Rd_file
3290 Eof: OFF END @Disc
3300     ASSIGN @Disc TO *
3310     DEALLOCATE Ascii$
3320     MASS STORAGE IS ":HP8290X,700,0"
3330 SUBEND
3340!____________________________________________________________________
3350 SUB Init_mem(@Multi,Memory)
3360    Prog_mode(@Multi,Memory,254)
3370    Prog_mode(@Multi,Memory,54)
3380    Prog_ref1(@Multi,Memory,1048575)
3390    Prog_ref2(@Multi,Memory,0)
3400    Prog_read(@Multi,Memory,0)
3410    Prog_diff(@Multi,Memory,0)
3420    Prog_write(@Multi,Memory,0)
3430    OUTPUT @Multi;"CC";Memory;"T"
3440 SUBEND
3450!____________________________________________________________________
3460 SUB Prog_read(@Multi,Memory,Read_pointer)
3470    Read_low=Read_pointer MOD 65536
3480    Read_high=Read_pointer DIV 65536
3490    Read_low_oct=FNDec_to_octal(Read_low)
3500    Read_high_oct=FNDec_to_octal(Read_high)
3510    OUTPUT @Multi;"WF";Memory+.1;22;Memory+.2;Read_low_oct;"T"
3520    OUTPUT @Multi;"WF";Memory+.1;23;Memory+.2;Read_high_oct;"T"
3530 SUBEND
3540!____________________________________________________________________
3550 SUB Prog_write(@Multi,Memory,Write_pointer)
3560    Write_low=Write_pointer MOD 65536
3570    Write_high=Write_pointer DIV 65536
3580    Write_low_oct=FNDec_to_octal(Write_low)
3590    Write_high_oct=FNDec_to_octal(Write_high)
3600    OUTPUT @Multi;"WF";Memory+.1;24;Memory+.2,Write_low_oct;"T"
3610    OUTPUT @Multi;"WF";Memory+.1;25;Memory+.2;Write_high_oct;"T"
3620 SUBEND
3630!____________________________________________________________________
3640 SUB Prog_diff(@Multi,Memory,Diff_counter)
3650    Diff_low=Diff_counter MOD 65536
3660    Diff_high=Diff_counter DIV 65536
3670    Diff_low_oct=FNDec_to_octal(Diff_low)
3680    Diff_high_oct=FNDec_to_octal(Diff_high)
3690    OUTPUT @Multi;"WF";Memory+.1;20;Memory+.2;Diff_low_oct;"T"
```

```
3700   OUTPUT @Multi;"WF";Memory+.1;21;Memory+.2;Diff_high_oct;"T"
3710 SUBEND
3720!______________________________________________________________________
3730 SUB Prog_ref1(@Multi,Memory,Ref1_reg)
3740   Ref1_low=Ref1_reg MOD 65536
3750   Ref1_high=Ref1_reg DIV 65536
3760   Ref1_low_oct=FNDec_to_octal(Ref1_low)
3770   Ref1_high_oct=FNDec_to_octal(Ref1_high)
3780   OUTPUT @Multi;"WF";Memory+.1,26;Memory+.2;Ref1_low_oct;"T"
3790   OUTPUT @Multi;"WF";Memory+.1,27;Memory+.2;Ref1_high_oct;"T"
3800 SUBEND
3810!______________________________________________________________________
3820 SUB Prog_ref2(@Multi,Memory,Ref2_reg)
3830   Ref2_low=Ref2_reg MOD 65536
3840   Ref2_high=Ref2_reg DIV 65536
3850   Ref2_low_oct=FNDec_to_octal(Ref2_low)
3860   Ref2_high_oct=FNDec_to_octal(Ref2_high)
3870   OUTPUT @Multi;"WF";Memory+.1;30;Memory+.2;Ref2_low_oct;"T"
3880   OUTPUT @Multi;"WF";Memory+.1;31;Memory+.2;Ref2_high_oct;"T"
3890 SUBEND
3900!______________________________________________________________________
3910 SUB Prog_mode(@Multi,Memory,Mode)
3920   OUTPUT @Multi;"WF";Memory+.3;Mode;"T"
3930 SUBEND
3940!______________________________________________________________________
3950 SUB Read_status(@Multi,Memory,Status)
3960   STATUS @Multi,3;Multi
3970   OUTPUT @Multi;"WF";Memory+.1;1;"T RV";Memory+.2;"T"
3980   ENTER Multi*100+6;Status
3990 SUBEND
4000!______________________________________________________________________
4010 SUB Read_read(@Multi,Memory,Read_pointer)
4020   STATUS @Multi,3;Multi
4030   OUTPUT @Multi;"WF";Memory+.1;22;"T RV";Memory+.2;"T"
4040   ENTER Multi*100+6;Read_low_oct
4050   OUTPUT @Multi;"WF";Memory+.1;23;"T RV";Memory+.2;"T"
4060   ENTER Multi*100+6;Read_high_oct
4070   Read_low=FNOctal_to_dec(Read_low_oct)
4080   Read_high=FNOctal_to_dec(Read_high_oct)
4090   Read_pointer=65536*Read_high+Read_low
4100 SUBEND
4110!______________________________________________________________________
4120 SUB Read_write(@Multi,Memory,Write_pointer)
4130   STATUS @Multi,3;Multi
4140   OUTPUT @Multi;"WF";Memory+.1;24;"T RV";Memory+.2;"T"
4150   ENTER Multi*100+6;Write_low_oct
4160   OUTPUT @Multi;"WF";Memory+.1;25;"T RV";Memory+.2;"T"
4170   ENTER Multi*100+6;Write_high_oct
4180   Write_low=FNOctal_to_dec(Write_low_oct)
4190   Write_high=FNOctal_to_dec(Write_high_oct)
4200   Write_pointer=65536*Write_high+Write_low
4210 SUBEND
4220!______________________________________________________________________
4230 SUB Read_diff(@Multi,Memory,Diff_counter)
4240   STATUS @Multi,3;Multi
```

```
4250   OUTPUT @Multi;"WF";Memory+.1;20;"T RV";Memory+.2;"T"
4260   ENTER Multi*100+6;Diff_low_oct
4270   OUTPUT @Multi;"WF";Memory+.1;21;"T RV";Memory+.2;"T"
4280   ENTER Multi*100+6;Diff_high_oct
4290   Diff_low=FNOctal_to_dec(Diff_low_oct)
4300   Diff_high=FNOctal_to_dec(Diff_high_oct)
4310   Diff_counter=65536*Diff_high+Diff_low
4320 SUBEND
4330!_______________________________________________________________________________
4340 DEF FNDec_to_octal(Dec)
4350   Oct=Dec+2*INT(Dec/8)+20*INT(Dec/64)+200*INT(Dec/512)+2000*INT(Dec/4096)+2
0000*INT(Dec/32768)+200000*INT(Dec/262144)
4360   RETURN Oct
4370 FNEND
4380!_______________________________________________________________________________
4390 DEF FNOctal_to_dec(Octal)
4400   X9=0
4410   X7=Octal
4420   X=0
4430 More:    !
4440   X=X+(X7-INT(X7/10)*10)*8^X9
4450   X7=INT(X7/10)
4460   X9=X9+1
4470   IF X7<>0 THEN GOTO More
4480   RETURN X
4490 FNEND
4500!_______________________________________________________________________________
4510  SUB Plot_frame(Material$,Contr_variable,Mean,Span,Frequency)
4520  ALLOCATE A(20),B(20)
4530  FOR I=0 TO 4
4540    A(I)=25*I-50
4550  NEXT I
4560  FOR I=0 TO 20
4570    B(I)=3*I-40
4580  NEXT I
4590  DEG
4600  GINIT
4610  GCLEAR
4620  ALPHA ON
4630  GRAPHICS OFF
4640  FRAME
4650  WINDOW -66.7224080268,66.7224080268,-50,50
4660  MOVE -50,-40
4670  FOR I=0 TO 3
4680    FOR J=1 TO 9
4690      DRAW 25*LGT(J*10^I)-50,-40
4700      IF J=1 THEN
4710         IDRAW 0,1
4720         IMOVE 0,-1
4730      ELSE
4740         IDRAW 0,.5
4750         IMOVE 0,-.5
4760      END IF
4770    NEXT J
```

```
4780  NEXT I
4790  C=.5
4800  D=1.0
4810  FOR I=0 TO 20
4820    DRAW 50,B(I)
4830    IF 2*INT(I/2)=I THEN
4840       IDRAW -D,0
4850       IMOVE D,0
4860    ELSE
4870       IDRAW -C,0
4880       IMOVE C,0
4890    END IF
4900  NEXT I
4910  FOR I=3 TO 0 STEP -1
4920    FOR J=9 TO 1 STEP -1
4930      DRAW 25*LGT(J*10^I)-50,20
4940      IF J=1 THEN
4950        IDRAW 0,-1
4960        IMOVE 0,1
4970      ELSE
4980        IDRAW 0,-.5
4990        IMOVE 0,.5
5000      END IF
5010    NEXT J
5020  NEXT I
5030  FOR I=20 TO 0 STEP -1
5040    DRAW -50,B(I)
5050    IF 2*INT(I/2)=I THEN
5060       IDRAW D,0
5070       IMOVE -D,0
5080    ELSE
5090       IDRAW C,0
5100       IMOVE -C,0
5110    END IF
5120  NEXT I
5130  MOVE -50,-10
5140  FOR I=0 TO 3
5150    FOR J=1 TO 9
5160      DRAW 25*LGT(J*10^I)-50,-10
5170      IF J=1 THEN
5180        IDRAW 0,1
5190        IDRAW 0,-2
5200        IMOVE 0,1
5210      ELSE
5220        IDRAW 0,.5
5230        IDRAW 0,-1
5240        IMOVE 0,.5
5250      END IF
5260    NEXT J
5270  NEXT I
5280  MOVE 50,20
5290  DRAW 50,40
5300  DRAW -50,40
5310  DRAW -50,20
5320  CSIZE 3,.6
```

```
5330  MOVE -50,-44
5340  LABEL VAL$(1)
5350  FOR I=1 TO 4
5360    MOVE A(I)-3,-44
5370    LABEL VAL$(10^I)
5380  NEXT I
5390  CSIZE 4,.6
5400  MOVE -5,-48
5410  LABEL "CYCLE"
5420  CSIZE 3,.6
5430  LDIR 90
5440  MOVE -58,-36
5450  IDRAW 0,3
5460  IDRAW -1.5,-1.5
5470  IDRAW 1.5,-1.5
5480  MOVE -57,-32
5490  LABEL "Ep(X 10   )"
5500  IMOVE -3.5,13
5510  LABEL "-"
5520  MOVE -58,-3
5530  IDRAW 0,3
5540  IDRAW -1.5,-1.5
5550  IDRAW 1.5,-1.5
5560  MOVE -57,1
5570  IF Contr_variable=1 THEN
5580     LABEL "E/2(X 10  )"
5590     IMOVE -3.5,14.6
5600     LABEL "-"
5610  ELSE
5620     LABEL "S/2 (MPa)"
5630  END IF
5640  LDIR -90
5650  MOVE 59,16
5660  IDRAW 0,-3
5670  IDRAW 1.5,1.5
5680  IDRAW -1.5,1.5
5690  MOVE 59,12
5700  IF Contr_variable=1 THEN
5710    LABEL "E/2(X 10  )"
5720    IMOVE 3.5,-14.6
5730    LABEL "-"
5740  ELSE
5750    LABEL "S/2 (MPa)"
5760  END IF
5770  MOVE 59,-16
5780  IDRAW 0,-3
5790  IDRAW 1.5,1.5
5800  IDRAW -1.5,1.5
5810  MOVE 59,-20
5820  LABEL "Ep(X 10  )"
5830  IMOVE 3.5,-13
5840  LABEL "-"
5850  LDIR 0
5860  CSIZE 3.5,.6
```

```
5870  MOVE -45,34
5880  LABEL Material$
5890  MOVE -45,30
5900  LABEL "ISOLCF"
5910  MOVE -45,26
5920  LABEL "Frequency= "&VAL$(Frequency)&" Hz"
5930  IF Contr_variable=1 THEN
5940     MOVE 15,34
5950     LABEL "STRESS CONTROL"
5960     MOVE 15,31
5970     IDRAW 3,0
5980     IDRAW -1.5,1.5
5990     IDRAW -1.5,-1.5
6000     MOVE 19,30
6010     LABEL "S = "&VAL$(Span*2)&" (MPa)"
6020  ELSE
6030     MOVE 15,34
6040     LABEL "STRAIN CONTROL"
6050     MOVE 15,31
6060     IDRAW 3,0
6070     IDRAW -1.5,1.5
6080     IDRAW -1.5,-1.5
6090     MOVE 19,30
6100     LABEL "E = "&VAL$(Span*2)
6110  END IF
6120  MOVE 19,26
6130  LABEL "R = "&VAL$(DROUND((Mean-Span)/(Mean+Span),2))
6140  SUBEND
6150!_____________________________________________________________________
6160  SUB Init_read(Frequency,Pace,Cycle_per_block,Data_per_cycle,Ref_count)
6170  Ref_count=(Cycle_per_block*Data_per_cycle*2)-1
6180  Pace=INT(1.E+6/(Data_per_cycle*2*Frequency)-30)       !Pace in usec
6190  SUBEND
6200!_____________________________________________________________________
6210  SUB Plot_data(Contr_variable,Scale,Span,Gauge,Gauge_factor,Cycle,Next_cycl
e,Ref_count,Scan(*),Exponent,Expo,Min,Mini)
6220  ALPHA OFF
6230  GRAPHICS ON
6240  CSIZE 3,.6
6250  Start=0
6260  Stop=399
6270  Indice=1
6280 Transfer:!
6290    Next_cycle=INT(Cycle*1.25)
6300    IF Next_cycle=Cycle THEN Next_cycle=Cycle+1
6310    IF Contr_variable=1 THEN Indice=0
6320      FOR I=Start+74 TO Start+124 STEP 2
6330        J=I+Indice
6340        IF Scan(J)*.005>Max THEN Maximum=Scan(J)*.005
6350        IF Scan(J+200)*.005<Min THEN Minimum=Scan(J+200)*.005
6360        IF ABS(Scan(J+50))*.005<.10 THEN Maxim=Scan(J+50)*.005
6370        IF ABS(Scan(J+250))*.005<.10 THEN Minim=Scan(J+250)*.005
6380      NEXT I
```

```
6390    Range=Maximum-Minimum
6400    Ep_range=Scan(Maxim+1)-Scan(Minim+1)
6410    Load_scale=INT(Scale)/100
6420    Strain_scale=Scale-(Load_scale*100)
6430    IF Cycle>1 THEN GOTO Data
6440    CALL Scale_frame(Contr_variable,Scale,Gauge,Gauge_factor,Range,Ep_range,
Min,Mini,Exponent,Expo)
6450 Data: !
6460    X=25*LGT(Cycle)-50
6470    IF Contr_variable=1 THEN
6480       Strain_ampli=Range*Strain_scale*Gauge_factor/(Gauge*20)
6490       Y1=4*(Strain_ampli*(10^Exponent)-Min)
6500    ELSE
6510       Stress_ampli=Range*Load_scale*216.078/2
6520       Y1=.2*(Stress_ampli-Min)
6530    END IF
6540    MOVE X-.7,3*Y1-11.3
6550    LABEL "x"
6560    Ep=Ep_range*Strain_scale*Gauge_factor/(Gauge*10)
6570    MOVE X-.7,12*(Ep*(10^Expo)-Mini)-41.3
6580    LABEL "x"
6590 Enough:  !
6600   Cycle=Cycle+1
6610   Start=Stop+1
6620   Stop=Stop+400
6630   IF Stop>Ref_count THEN GOTO Out
6640   IF Cycle<Next_cycle THEN GOTO Enough
6650   GOTO Transfer
6660 Out:      !
6670   Cycle=Cycle-1
6680   SUBEND
6690!-----------------------------------------------------------------------
6700   SUB Scale_frame(Contr_variable,Scale,Gauge,Gauge_factor,Range,Ep_range,Mi
n,Mini,Exponent,Expo)
6710   Load_scale=INT(Scale)/100
6720   Strain_scale=Scale-(Load_scale*100)
6730   IF Contr_variable=1 THEN
6740      Strain_ampli=Range*Strain_scale*Gauge_factor/(Gauge*20)
6750      FOR Multiplier=0 TO 5
6760        IF Strain_ampli*10^Multiplier>1 AND Strain_ampli*10^Multiplier<10 TH
EN Exponent=Multiplier
6770      NEXT Multiplier
6780      A=Strain_ampli*10^Exponent
6790      IF A<2.5 THEN
6800        Min=0.
6810      ELSE
6820        IF A<5.0 THEN
6830          Min=2.5
6840        ELSE
6850          IF A<7.5 THEN
6860            Min=5.0
6870          ELSE
6880            Min=7.5
6890          END IF
```

```
6900          END IF
6910        END IF
6920     ELSE
6930        Stress_ampli=Range*Load_scale*216.078/2
6940        Norme=INT(Stress_ampli/100)*100
6950        A=Stress_ampli-Norme
6960        IF A<25 THEN
6970          Min=Norme
6980          IF A<12.5 THEN Min=Norme-25
6990        ELSE
7000          IF A<50 THEN
7010            Min=Norme+25
7020            IF A<37.5 THEN Min=Norme
7030          ELSE
7040            IF A<75 THEN
7050              Min=Norme+50
7060              IF A<62.5 THEN Min=Norme+25
7070            ELSE
7080              Min=Norme+75
7090              IF A<77.5 THEN Min=Norme+50
7100            END IF
7110          END IF
7120        END IF
7130      END IF
7140      Ep=Ep_range*Strain_scale*Gauge_factor/(Gauge*10)
7150      FOR Multiplier=0 TO 5
7160        IF Ep*10^Multiplier>1 AND Ep*10^Multiplier<10 THEN Expo=Multiplier
7170      NEXT Multiplier
7180      A=Ep*10^Expo
7190      IF A<2.5 THEN
7200         Mini=0.
7210      ELSE
7220         IF A<5.0 THEN
7230            Mini=2.5
7240         ELSE
7250            IF A<7.5 THEN
7260               Mini=5.0
7270            ELSE
7280               Mini=7.5
7290            END IF
7300         END IF
7310      END IF
7320      CSIZE 3,.5
7330      FOR I=0 TO 8 STEP 2
7340        MOVE -56,3*I-41
7350        LABEL USING "D.D";(.25*I+Mini)
7360        MOVE 51,3*I-41
7370        LABEL USING "D.D";(.25*I+Mini)
7380      NEXT I
7390      A$="3D"
7400      Facto=5
7410      IF Contr_variable=1 THEN
7420         A$="D.D"
7430         Facto=.50
```

```
7440     END IF
7450     FOR I=2 TO 10 STEP 2
7460       MOVE -56,3*I-11.5
7470       LABEL USING A$;(Facto*I+Min)
7480       MOVE 51,3*I-11.5
7490       LABEL USING A$;(Facto*I+Min)
7500     NEXT I
7510     CSIZE 3,.6
7520     LDIR 90
7530     MOVE -57,-32
7540     IMOVE -.5,16
7550     LABEL USING "D";Expo
7560     IF Contr_variable=1 THEN
7570        MOVE -57,1
7580        IMOVE -1,16.6
7590        LABEL USING "D";Exponent
7600        LDIR -90
7610        MOVE 59,12
7620        IMOVE .5,-16.6
7630        LABEL USING "D";Exponent
7640     END IF
7650     LDIR -90
7660     MOVE 59,-20
7670     IMOVE .5,-14
7680     LABEL USING "D";Expo
7690     LDIR 0
7700   SUBEND
```

```
10!    *********************************************************************
20!    *                                                                   *
30!    *                    PROGRAM THERMAL_STRAINS                        *
40!    *                    ***********************                        *
50!    *                                                                   *
60!    *      THIS PROGRAM PERFORMS THE REAL-TIME CONTROL AND DATA         *
70!    *      ACQUISITION OF THERMAL STRAINS TESTS MEASUREMENTS.           *
80!    *                                                                   *
90 !   *      THE MICRO_DATA_TRACK MUST BE PROGRAMMED FOR 180 sec          *
100!   *      PERIOD (1/3 cpm).                                            *
110!   *                                                                   *
120!   *      THE CYCLIC THERMAL STRAINS ARE MEASURED BY CYCLING THE       *
130!   *      TEMPERATURE AT ZERO STRESS.                                  *
140!   *                                                                   *
150!   *********************************************************************
160    !
170    OPTION BASE 0
180    CALL Read_param(Material$,Mode,Scale,Mean,Span,Gauge)
190    DELSUB Read_param
200    CALL Thermal_strain(Material$,Mode,Scale,Mean,Span,Gauge)
210    END
220!************************************************************************
230!************************************************************************
240    !
250    SUB Thermal_strain(Material$,Mode,Scale,Mean,Span,Gauge)
260    ALLOCATE Load(4095),Strain(4095),Temp1(4095),Temp2(4095),Temp3(4095)
270    MASS STORAGE IS ":HP8290X,700,1"
280    !
290    !______ _______________DEFINE I/O CARD FUNCTION_________________________
300    !                     -----------------------
310    ASSIGN @Dvm TO 720
320    ASSIGN @Multi TO 723
330    ASSIGN @Disc TO "MRZ"
340    Disc_drive=0
350    Dvm=720
360    Multi=723
370    Scanner=0
380    A_to_d=3
390    Memory1=7
400    D_to_a1=10
410    Timer=11
420    D_to_a2=12
430    Memory2=13
440    !
450    !_________________DEFAULT VALUES FOR THE PARAMETERS____________________
460    !                 ---------------------------------
470    Start=1                                       !Start channel
480    Stop=5                                        !Stop channel
490    Pointer=20479                                 !Interrupt word(4906*5)
500    IF Mode=1 THEN
510      ALLOCATE Pot(1)
520    ELSE
530      ALLOCATE Pot(809)
540    END IF
```

```
550    !
560    !______________________INITIALIZE THE MULTI____________________________
570    !                      --------------------
580    CALL Init_multi(@Multi,@Disc)
590    CALL Init_mem(@Multi,Memory1)
600    !
610    !________________SET INTERRUPT BRANCH FOR TRIGGER SIGNAL________________
620    !                ----------------------------------------
630    Cycle=1                                  !Set cycle counter
640    G=SPOLL(723)                             !Clear Multi SRQ
650    P=SPOLL(720)                             !Clear DVM SRQ
660    BEEP
670    PRINT USING 680
680    IMAGE 20/,15X,"REMOVE CONTROL AND UTILITY PAC DISCS.",2/,15X,"INSERT ONE F
LOPPY(in right disc) FOR "
690    PRINT USING 700
700    IMAGE 1/,15X,"DATA STORAGE.  PRESS K0 TO CONTINUE."
710    ON KEY 0 LABEL "**CONTINUE**" GOTO 730
720 Spin: GOTO Spin
730    BEEP
740    OFF KEY
750    CREATE BDAT "Thermal",1,100
760    ASSIGN @Disc TO "Thermal"
770    OUTPUT @Disc;Material$,"THERMAL_STRAIN",Mode,Scale,Mean,Span,Gauge
780    ASSIGN @Disc TO *
790    PRINT USING 800
800    IMAGE 20/,25X,"PROGRAM AND START THE LEPEL",2/,20X,"****WAITING FOR A TRIG
GER SIGNAL****"
810    PRINT USING 820
820    IMAGE 5/,15X,"*****   THERMAL STRAINS MEASUREMENT IN PROGRESS   *****"
830        OUTPUT @Dvm;"F1R-2N5Z0"              !Set DVM range
840        OUTPUT @Dvm;"T2D3 TMF TEST"          !Set ext trigger and display
850        OUTPUT @Dvm;"KM01"                   !Set mask for Data Ready
860        ON INTR 7,2 GOTO Start_tmf           !Set interrupt branch
870        ENABLE INTR 7;2                      !Enable interrupt SRQ
880        ENTER Dvm;P
890    !
900    !____________________START CYCLING AND TAKING DATA____________________
910    !                    ----------------------------
920 Start_tmf:        !
930    BEEP
940    OUTPUT @Multi;"SC,0,0,0,0T"                 !Reset Clock of Multi
950    !
960    !__________READ DATA___________
970    !
980    CALL Read_data(@Multi,@Dvm,Multi,Dvm,Scanner,Memory1,Mode,Load(*),Strain(*
),Pot(*),Temp1(*),Temp2(*),Temp3(*))
990    !
1000   !__________STORE DATA__________
1010   !          ----------
1020   CALL Store_data(@Multi,@Disc,Disc_drive,Mode,Cycle,Scale,Load(*),Strain(*)
,Pot(*),Temp1(*),Temp2(*),Temp3(*))
1030   !
1040   !__________DISPLAY DATA__________
```

```
1050  !         ------------
1060  CALL Plot_data(Material$,Scale,Span,Gauge,Cycle,Load(*),Strain(*),Temp1(*)
,Temp2(*),Temp3(*))
1070  !
1080  !__________SET INTERRUPT FOR SYNCHRO. SIGNAL_________
1090  !         ----------------------------------
1100  IF Cycle<9 THEN
1110       Increment=2
1120       GOTO Synchro
1130  END IF
1140  Cycle2=INT(Cycle*1.25)
1150  Increment=Cycle2-Cycle
1160  !
1170 Synchro:                                    !
1180  Waiting=Increment-1.5
1190  Cycle=Cycle+Increment                      !Update cycle counter
1200  WAIT (180*Waiting)                         !Set waiting time
1210  CALL Init_mem(@Multi,Memory1)              !Reset memory card
1220  OUTPUT @Dvm;"T2KM01"                       !Ext. trigger and DVM mask
1230  ON INTR 7,2 GOTO 970                       !Set interrupt branch
1240  ENABLE INTR 7;2                            !Enable interrupt
1250  ENTER Dvm;P
1260  SUBEND
1270!_____________________________________________________________________
1280!
1290!  ****************************************************************
1300!  *                                                              *
1310!  *       SUBROUTINES USED BY SUB TMFCG(Param1,Param2,...)       *
1320!  *                                                              *
1330!  ****************************************************************
1340!
1350!_____________________________________________________________________
1360!
1370  SUB Read_data(@Multi,@Dvm,Multi,Dvm,Scanner,Memory,Mode,Load(*),Strain(*),
Pot(*),Temp1(*),Temp2(*),Temp3(*))
1380  ALLOCATE INTEGER Scan(20479)
1390  OUTPUT @Dvm;"T1M00"
1400    G=SPOLL(@Multi)
1410    ENTER 72310;A,B,C
1420    ON INTR 7,2 GOTO Interrupt
1430    ENABLE INTR 7;2
1440    OUTPUT @Multi;"CY",Scanner,"T"              !Enable scanner
1450    OUTPUT @Multi;"WF",Scanner+.3,5,"T"         !Set sequential mode
1460    !
1470      OUTPUT @Multi;"WF",Scanner,1,"T"          !Set start channel
1480      OUTPUT @Multi;"WF",Scanner+.1,5,"T"       !set stop channel
1490      OUTPUT @Multi;"WF",Scanner+.2,8759,"T"    !Set pace(40 us)
1500      Prog_diff(@Multi,Memory,0)                !Reset diff. counter
1510      Prog_write(@Multi,Memory,0)               !Reset write pointer
1520      OUTPUT @Multi;"CC";Memory,"T"             !Clear Memory card
1530      Prog_mode(@Multi,Memory,56)               !Set FIFO mode
1540      Prog_ref1(@Multi,Memory,20479)            !Set stop pointer
1550      OUTPUT @Multi;"AC";Memory;"T"             !Armed memory card
1560      OUTPUT @Multi;"CY",Scanner,"T"            !Start pacer
```

```
1570       IF Mode=1 THEN GOTO Wait
1580       FOR I=0 TO 809                                    !If require read
1590         ENTER @Dvm;Pot(I)                               !potential change
1600       NEXT I                                            !with temperature
1610 Wait:    GOTO Wait
1620 Interrupt:        !
1630       BEEP
1640       G=SPOLL(@Multi)
1650       IF G=64 THEN
1660         ENTER 72310;A,B,C
1670         IF C<>0 THEN
1680            ENTER 72312;Address
1690            IF Address<>Memory THEN
1700               PRINT "NOT MEMORY CARD"
1710               PAUSE
1720            END IF
1730            OUTPUT @Multi;"WF",Scanner+.2,0,"T"        !Stop scanner
1740            Prog_mode(@Multi,Memory,76)                !Set FIFO lockout
1750            OUTPUT @Multi;"MR";Memory;20480;"T"        !MR command to get data
1760            ENTER 72305 USING "%,W";Scan(*)            !Entering the data
1770            FOR K=0 TO 20479 STEP 5
1780              J=INT(K/5)
1790              Strain(J)=Scan(K)*.005
1800              Load(J)=Scan(K+1)*.005
1810              Temp1(J)=Scan(K+2)*.005
1820              Temp2(J)=Scan(K+3)*.005
1830              Temp3(J)=Scan(K+4)*.005
1840            NEXT K
1850            ENABLE INTR 7;2
1860         ELSE
1870            BEEP 2000,.3
1880            PRINT "          ***  SRQ NOT SET BY ARMED CARD  ***"
1890            PAUSE
1900         END IF
1910      ELSE
1920         BEEP 2000,.3
1930         PRINT "          ***  MULTI DID NOT INTERRUPT  ***"
1940         PAUSE
1950      END IF
1960   DEALLOCATE Scan(*)
1970   OUTPUT @Multi;"RC"                                   !Read Real-time clock
1980   SUBEND
1990!-----------------------------------------------------------------------------
2000   SUB Store_data(@Multi,@Disc,Disc_drive,Mode,Cycle,Scale,Load(*),Strain(*)
,Pot(*),Temp1(*),Temp2(*),Temp3(*))
2010   !
2020  FOR I=0 TO 4095
2030    Temp1(I)=(Temp1(I))!+Temp2(I)+Temp3(I))/3
2040    Temp2(I)=(Load(I)+10)*1000+(Strain(I)+10)/100
2050  NEXT I
2060  ON ERROR GOTO Recover
2070  MASS STORAGE IS ":HP8290X,700,0"
2080  CREATE BDAT "CYCLE"&VAL$(Cycle),15,5000
2090  ASSIGN @Disc TO "CYCLE"&VAL$(Cycle)
```

```
2100  IF Mode=1 THEN
2110    OUTPUT @Disc;Cycle,Scale,Temp1(*),Temp2(*)
2120  ELSE
2130    OUTPUT @Disc;Cycle,Scale,Temp1(*),Temp2(*),Pot(*)
2140  END IF
2150  ASSIGN @Disc TO *
2160  GOTO Out
2170 Recover:          IF ERRN=64 THEN MASS STORAGE IS ":HP8290X,700,1"
2180                   GOTO 2080
2190  !
2200 Out: !
2210    SUBEND
2220!_________________________________________________________________
2230 SUB Init_multi(@Multi,@Disc)
2240   CLEAR 7
2250   WAIT 4.0
2260   G=SPOLL(@Multi)
2270   IF G<>64 THEN
2280      PRINT "      ***MULTI DID NOT INTERRUPT***"
2290      PAUSE
2300   END IF
2310   STATUS @Multi,3;Multi
2320   ENTER Multi*100+10;A
2330   IF A<>16384 THEN
2340     PRINT "       ***SELF TEST DID NOT SET SRQ***"
2350     PAUSE
2360   END IF
2370  ALLOCATE Ascii$[80]
2380  ON END @Disc GOTO Eof
2390 Rd_file:    ENTER @Disc;Ascii$
2400    OUTPUT @Multi;Ascii$
2410    GOTO Rd_file
2420 Eof: OFF END @Disc
2430    ASSIGN @Disc TO *
2440    DEALLOCATE Ascii$
2450    MASS STORAGE IS ":HP8290X,700,0"
2460 SUBEND
2470!_________________________________________________________________
2480 SUB Init_mem(@Multi,Memory)
2490   Prog_mode(@Multi,Memory,254)
2500   Prog_mode(@Multi,Memory,54)
2510   Prog_ref1(@Multi,Memory,1048575)
2520   Prog_ref2(@Multi,Memory,0)
2530   Prog_read(@Multi,Memory,0)
2540   Prog_diff(@Multi,Memory,0)
2550   Prog_write(@Multi,Memory,0)
2560   OUTPUT @Multi;"CC";Memory;"T"
2570 SUBEND
2580!_________________________________________________________________
2590 SUB Prog_read(@Multi,Memory,Read_pointer)
2600   Read_low=Read_pointer MOD 65536
2610   Read_high=Read_pointer DIV 65536
2620   Read_low_oct=FNDec_to_octal(Read_low)
2630   Read_high_oct=FNDec_to_octal(Read_high)
```

```
2640   OUTPUT @Multi;"WF";Memory+.1;22;Memory+.2;Read_low_oct;"T"
2650   OUTPUT @Multi;"WF";Memory+.1;23;Memory+.2;Read_high_oct;"T"
2660 SUBEND
2670!_______________________________________________________________
2680 SUB Prog_write(@Multi,Memory,Write_pointer)
2690   Write_low=Write_pointer MOD 65536
2700   Write_high=Write_pointer DIV 65536
2710   Write_low_oct=FNDec_to_octal(Write_low)
2720   Write_high_oct=FNDec_to_octal(Write_high)
2730   OUTPUT @Multi;"WF";Memory+.1;24;Memory+.2,Write_low_oct;"T"
2740   OUTPUT @Multi;"WF";Memory+.1;25;Memory+.2;Write_high_oct;"T"
2750 SUBEND
2760!_______________________________________________________________
2770 SUB Prog_diff(@Multi,Memory,Diff_counter)
2780   Diff_low=Diff_counter MOD 65536
2790   Diff_high=Diff_counter DIV 65536
2800   Diff_low_oct=FNDec_to_octal(Diff_low)
2810   Diff_high_oct=FNDec_to_octal(Diff_high)
2820   OUTPUT @Multi;"WF";Memory+.1;20;Memory+.2;Diff_low_oct;"T"
2830   OUTPUT @Multi;"WF";Memory+.1;21;Memory+.2;Diff_high_oct;"T"
2840 SUBEND
2850!_______________________________________________________________
2860 SUB Prog_ref1(@Multi,Memory,Ref1_reg)
2870   Ref1_low=Ref1_reg MOD 65536
2880   Ref1_high=Ref1_reg DIV 65536
2890   Ref1_low_oct=FNDec_to_octal(Ref1_low)
2900   Ref1_high_oct=FNDec_to_octal(Ref1_high)
2910   OUTPUT @Multi;"WF";Memory+.1,26;Memory+.2;Ref1_low_oct;"T"
2920   OUTPUT @Multi;"WF";Memory+.1,27;Memory+.2;Ref1_high_oct;"T"
2930 SUBEND
2940!_______________________________________________________________
2950 SUB Prog_ref2(@Multi,Memory,Ref2_reg)
2960   Ref2_low=Ref2_reg MOD 65536
2970   Ref2_high=Ref2_reg DIV 65536
2980   Ref2_low_oct=FNDec_to_octal(Ref2_low)
2990   Ref2_high_oct=FNDec_to_octal(Ref2_high)
3000   OUTPUT @Multi;"WF";Memory+.1;30;Memory+.2;Ref2_low_oct;"T"
3010   OUTPUT @Multi;"WF";Memory+.1;31;Memory+.2;Ref2_high_oct;"T"
3020 SUBEND
3030!_______________________________________________________________
3040 SUB Prog_mode(@Multi,Memory,Mode)
3050   OUTPUT @Multi;"WF";Memory+.3;Mode;"T"
3060 SUBEND
3070!_______________________________________________________________
3080 SUB Read_status(@Multi,Memory,Status)
3090   STATUS @Multi,3;Multi
3100   OUTPUT @Multi;"WF";Memory+.1;1;"T RV";Memory+.2;"T"
3110   ENTER Multi*100+6;Status
3120 SUBEND
3130!_______________________________________________________________
3140 SUB Read_read(@Multi,Memory,Read_pointer)
3150   STATUS @Multi,3;Multi
3160   OUTPUT @Multi;"WF";Memory+.1;22;"T RV";Memory+.2;"T"
3170   ENTER Multi*100+6;Read_low_oct
```

```
3180   OUTPUT @Multi;"WF";Memory+.1;23;"T RV";Memory+.2;"T"
3190   ENTER Multi*100+6;Read_high_oct
3200   Read_low=FNOctal_to_dec(Read_low_oct)
3210   Read_high=FNOctal_to_dec(Read_high_oct)
3220   Read_pointer=65536*Read_high+Read_low
3230 SUBEND
3240!________________________________________________________________________
3250 SUB Read_write(@Multi,Memory,Write_pointer)
3260   STATUS @Multi,3;Multi
3270   OUTPUT @Multi;"WF";Memory+.1;24;"T RV";Memory+.2;"T"
3280   ENTER Multi*100+6;Write_low_oct
3290   OUTPUT @Multi;"WF";Memory+.1;25;"T RV";Memory+.2;"T"
3300   ENTER Multi*100+6;Write_high_oct
3310   Write_low=FNOctal_to_dec(Write_low_oct)
3320   Write_high=FNOctal_to_dec(Write_high_oct)
3330   Write_pointer=65536*Write_high+Write_low
3340 SUBEND
3350!________________________________________________________________________
3360 SUB Read_diff(@Multi,Memory,Diff_counter)
3370   STATUS @Multi,3;Multi
3380   OUTPUT @Multi;"WF";Memory+.1;20;"T RV";Memory+.2;"T"
3390   ENTER Multi*100+6;Diff_low_oct
3400   OUTPUT @Multi;"WF";Memory+.1;21;"T RV";Memory+.2;"T"
3410   ENTER Multi*100+6;Diff_high_oct
3420   Diff_low=FNOctal_to_dec(Diff_low_oct)
3430   Diff_high=FNOctal_to_dec(Diff_high_oct)
3440   Diff_counter=65536*Diff_high+Diff_low
3450 SUBEND
3460!________________________________________________________________________
3470 DEF FNDec_to_octal(Dec)
3480   Oct=Dec+2*INT(Dec/8)+20*INT(Dec/64)+200*INT(Dec/512)+2000*INT(Dec/4096)+2
0000*INT(Dec/32768)+200000*INT(Dec/262144)
3490   RETURN Oct
3500 FNEND
3510!________________________________________________________________________
3520 DEF FNOctal_to_dec(Octal)
3530   X9=0
3540   X7=Octal
3550   X=0
3560 More:   !
3570   X=X+(X7-INT(X7/10)*10)*8^X9
3580   X7=INT(X7/10)
3590   X9=X9+1
3600   IF X7<>0 THEN GOTO More
3610   RETURN X
3620 FNEND
3630!________________________________________________________________________
3640  SUB Plot_data(Material$,Scale,Span,Gauge,Cycle,Load(*),Strain(*),Temp1(*),
Temp2(*),Temp3(*))
3650  ALLOCATE A(20),B(20)
3660  Load_range=INT(Scale)/100
3670  Strain_range=Scale-(Load_range*100)
3680  Conv_fac=216.078
3690  Emax=.05555*Strain_range/Gauge
```

```
3700  FOR I=0 TO 20
3710    A(I)=-50+I*50/9
3720    B(I)=-40+I*2.5
3730  NEXT I
3740  DEG
3750  GINIT
3760  GRAPHICS ON
3770  GCLEAR
3780  ALPHA OFF
3790  FRAME
3800  WINDOW -66.7224080268,66.7224080268,-50,50
3810  MOVE -50,-40
3820  FOR I=0 TO 18
3830    DRAW A(I),-40
3840    IDRAW 0,1
3850    IMOVE 0,-1
3860  NEXT I
3870  C=1.
3880  D=.5
3890  FOR I=0 TO 20
3900    DRAW 50,B(I)
3910    IF 2*INT(I/2)=I THEN
3920       IDRAW -D,0
3930       IMOVE D,0
3940    ELSE
3950       IDRAW -C,0
3960       IMOVE C,0
3970    END IF
3980  NEXT I
3990  DRAW 50,40
4000  DRAW -50,40
4010  DRAW -50,10
4020  FOR I=20 TO 0 STEP -1
4030    DRAW -50,B(I)
4040    IF 2*INT(I/2)=I THEN
4050       IDRAW D,0
4060       IMOVE -D,0
4070    ELSE
4080       IDRAW C,0
4090       IMOVE -C,0
4100    END IF
4110  NEXT I
4120  MOVE -50,-15
4130  FOR I=0 TO 18
4140    DRAW A(I),-15
4150    IDRAW 0,1
4160    IDRAW 0,-2
4170    IMOVE 0,1
4180  NEXT I
4190  MOVE -50,10
4200  FOR I=0 TO 18
4210    DRAW A(I),10
4220    IDRAW 0,-1
4230    IMOVE 0,1
```

```
4240  NEXT I
4250  MOVE 10,10
4260  DRAW 10,40
4270  MOVE -40,25
4280  FOR I=0 TO 20 STEP 2
4290    DRAW (-40+I),25
4300    IDRAW 0,.5
4310    IDRAW 0,-1
4320    IMOVE 0,.5
4330  NEXT I
4340  MOVE -30,15
4350  FOR I=0 TO 20 STEP 2
4360    DRAW -30,15+I
4370    IDRAW .5,0
4380    IDRAW -1,0
4390    IMOVE .5,0
4400  NEXT I
4410  CSIZE 3,.6
4420  MOVE -50,-40
4430  FOR I=2 TO 17 STEP 2
4440    MOVE A(I)-3,-43
4450    LABEL VAL$(I*10)
4460  NEXT I
4470  FOR I=1 TO 10 STEP 2
4480     MOVE 50.5,B(I)-1.5
4490     LABEL USING "S.D";.2*I-1.0
4500     MOVE -55,B(I)-1.5
4510     LABEL USING "S.D";.2*I-1.0
4520  NEXT I
4530  FOR I=11 TO 20 STEP 2
4540     MOVE -58,B(I)-1.5
4550     LABEL VAL$((I-15)*100)
4560  NEXT I
4570  FOR I=11 TO 20 STEP 2
4580     MOVE 51.5,B(I)-1.5
4590     LABEL VAL$(70*(I-10)+300)
4600  NEXT I
4610  CSIZE 4,.6
4620  MOVE -20,-48
4630  LABEL "TIME  (sec)"
4640  CSIZE 3,.6
4650  LDIR 90
4660  MOVE -56,-35
4670  LABEL "Etot (%)"
4680  LDIR -90
4690  MOVE 58,-20
4700  LABEL "Etot (%)"
4710  LDIR 90
4720  MOVE -58,-14
4730  LABEL "STRESS (MPa)"
4740  LDIR -90
4750  MOVE 58,10
4760  LABEL "TEMPERATURE (C)"
4770  CSIZE 3.5,.6
```

```
4780  LDIR 0
4790  MOVE 15,32
4800  LABEL Material$
4810  MOVE 15,29
4820  LABEL "Thermal Strain"
4830  MOVE 20,26
4840  LABEL "Analysis"
4850  MOVE 15,23
4860  LABEL "Tmax= 926 C"
4870  MOVE 15,20
4880  LABEL "Tmin= 400 C"
4890  MOVE 15,17
4900  LABEL "CYCLE #"&VAL$(Cycle)
4910  CSIZE 3,.6
4920  MOVE -15,29
4930  DRAW -13,29
4940  DRAW -14,30
4950  DRAW -15,29
4960  MOVE -12,28
4970     LABEL "S= "&VAL$(2*Span)&"MPa"
4980     MOVE -15,32
4990     LABEL "STRESS CONTROL"
5000  MOVE -15,24
5010  LABEL "X-DIV=0.05%"
5020  MOVE -15,20
5030  LABEL "Y-DIV=100 MPa"
5040  MOVE -50,-2.5
5050  FOR I=0 TO 4095
5060    B1=(((Load(I)*Load_range*Conv_fac)+500)/40)-15
5070    A1=(I*100/4095)-50
5080    DRAW A1,B1
5090  NEXT I
5100  MOVE -50,-2.5
5110  FOR I=0 TO 4095
5120    B1=2.5*(Temp1(I)*1000-300)/70-15
5130    A1=(I*100/4095)-50
5140    DRAW A1,B1
5150  NEXT I
5160  MOVE -50,-27.5
5170  FOR I=0 TO 4095
5180     B1=((1.5*Strain(I)*Emax/10)+.01)*1250-40     !Plot thermal strain
5190     A1=(I*100/4095)-50
5200     DRAW A1,B1
5210  NEXT I
5220  A1=((1.5*Strain(0)*Emax/10)+.0025)*4000-40
5230  B1=(((Load(0)*Range*Conv_fac)+500)/50)+15
5240  MOVE A1,B1
5250  FOR I=1 TO 4095 STEP 5
5260     A1=((1.5*Strain(I)*Emax/10)+.0025)*4000-40       !Plot hysterisis loop
5270     B1=(((Load(I)*Range*Conv_fac)+500)/50)+15
5280     DRAW A1,B1
5290  NEXT I
5300  SUBEND
5310!_____________________________________________________________________________
```

```
5320  SUB Read_param(Material$,Mode,Scale,Mean,Span,Gauge)
5330  OPTION BASE 0
5340  ALLOCATE Dummy$[40],Dummy1$[80]
5350  Dummy$="****************************************"
5360  Dummy1$="*****************************************************************
***************"
5370  !
5380  PRINT USING 5440;Dummy1$
5390  PRINT USING 5400
5400  IMAGE 2/,20X,"ENTER TYPE OF ALLOY (10 Characters max)"
5410  BEEP
5420  INPUT Material$
5430!_________________________
5440  IMAGE 2/,80A
5450  PRINT USING 5440;Dummy1$
5460  PRINT USING 5470
5470  IMAGE 2/,20X,"ENTER MODE OF TESTING",/,25X,"1. LCF (Initiation)",/,25X,"2.
 CRACK PROPAGATION (Propagation)"
5480  BEEP
5490  INPUT Mode_of_testing
5500  IF Mode_of_testing<>1 AND Mode_of_testing<>2 THEN
5510     PRINT USING 5520
5520     IMAGE 2/,20X,"UNDEFINED MODE OF TESTING. Please re-enter"
5530     BEEP 3000,1.0
5540     GOTO 5460
5550  ELSE
5560  END IF
5570  !________________________
5580  PRINT USING 5440;Dummy1$
5590  PRINT USING 5600
5600  IMAGE 2/,20X,"ENTER CONTROL VARIABLE?",/,25X,"1. LOAD CONTROL",/,25X,"2. S
TRAIN CONTROL"
5610  BEEP
5620  INPUT Contr_variable
5630  IF Contr_variable<>1 AND Contr_variable<>2 THEN
5640     PRINT USING 5650
5650     IMAGE 2/,20X,"UNDEFINED CONTROL VARIABLE. Please re-enter"
5660     BEEP 3000,1.0
5670     GOTO 5590
5680  ELSE
5690  END IF
5700  !___________________________
5710  IF Contr_variable=1 THEN
5720      PRINT USING 5440;Dummy1$
5730      PRINT USING 5740
5740      IMAGE 2/,20X,"ENTER LOAD SCALE? (100,50,20 or 10 %)"
5750      BEEP
5760      INPUT Load_scale
5770      Load_scale=Load_scale
5780      IF Load_scale<>100 AND Load_scale<>50 AND Load_scale<>20 AND Load_scal
e<>10 THEN
5790          PRINT USING 5800
5800          IMAGE 2/,20X,"Undefined value. Please re-enter."
5810          BEEP 3000,1.0
5820          GOTO 5730
5830      ELSE
```

```
5840       END IF
5850    !________________________
5860      PRINT USING 5440;Dummy1$
5870      PRINT USING 5880
5880      IMAGE 2/,20X, "ENTER MEAN STRESS (in MPa)?"
5890      BEEP
5900      INPUT Mean
5910      IF (Mean*11.569896)<=-(250*Load_scale) OR Mean>=(250*Load_scale) THEN
5920        PRINT USING 5930
5930        IMAGE 2/,20X,"VALUE OUT-OF-BOUND. Please re-enter."
5940        BEEP 3000,1.0
5950        GOTO 5870
5960      ELSE
5970      END IF
5980    !___________________________
5990        PRINT USING 5440;Dummy1$
6000        PRINT USING 6010
6010        IMAGE 2/,20X, "ENTER STRESS AMPLITUDE (in MPa)"
6020        BEEP
6030        INPUT Span
6040        IF Span<=0 THEN
6050          PRINT USING 6060
6060          IMAGE 2/,20X,"UNDEFINED VALUE. Please re-enter"
6070          BEEP 3000,1.0
6080          GOTO 6000
6090        ELSE
6100          IF (ABS(Mean)+Span)*11.569896>=250*Load_scale THEN
6110            PRINT USING 6120
6120            IMAGE 2/,20X,"VALUE OUT-OF-BOUND,  Please re-enter"
6130            BEEP 3000,1.0
6140            GOTO 6000
6150          ELSE
6160          END IF
6170        END IF
6180    !___________________________
6190        PRINT USING 5440;Dummy1$
6200        PRINT USING 6210
6210        IMAGE 2/,20X,"ENTER THE GAUGE LENGTH (in inch)"
6220        BEEP
6230        INPUT Gauge_length
6240        IF Gauge_length<.400 OR Gauge_length>.6 THEN
6250          PRINT USING 6260
6260          IMAGE 2/,20X,"GAUGE LENGHT VALUE OUT-OF-BOUND. Please re-enter."
6270          BEEP 3000,1.0
6280          GOTO 6200
6290        ELSE
6300        END IF
6310    !___________________________
6320        PRINT USING 5440;Dummy1$
6330        PRINT USING 6340
6340        IMAGE 2/,20X,"ENTER STRAIN SCALE (100,50,20 or 10 %)"
6350        BEEP
6360        INPUT Strain_scale
6370        Strain_scale=Strain_scale/100
```

```
6380      IF Strain_scale<>1. AND Strain_scale<>.5 AND Strain_scale<>.2 AND Stra
in_scale<>.1 THEN
6390        PRINT USING 6400
6400        IMAGE 2/,20X,"UNDEFINED VALUE. Please re-enter."
6410        BEEP 3000,1.0
6420        GOTO 6330
6430      ELSE
6440      END IF
6450 !------------------------------
6460   ELSE
6470 !______________________________
6480      PRINT USING 5440;Dummy1$
6490      PRINT USING 6500
6500      IMAGE 2/,20X,"ENTER STRAIN SCALE (100,50,20 or 10 %)"
6510      BEEP
6520      INPUT Strain_scale
6530      Strain_scale=Strain_scale/100
6540      IF Strain_scale<>1. AND Strain_scale<>.5 AND Strain_scale<>.2 AND Stra
in_scale<>.1 THEN
6550         PRINT USING 6560
6560         IMAGE 2/,20X, "Undefined value.  Please re-enter"
6570         BEEP 3000,1.0
6580         GOTO 6490
6590      ELSE
6600      END IF
6610 !______________________________
6620      PRINT USING 5440;Dummy1$
6630      PRINT USING 6640
6640      IMAGE 2/,20X,"ENTER THE GAUGE LENGHT (in inch)"
6650      BEEP
6660      INPUT Gauge_lenght
6670      IF Gauge_lenght<.4 OR Gauge_lenght>.6 THEN
6680         PRINT USING 6690
6690         IMAGE 2/,20X,"GAUGE LENGHT VALUE OUT-OF-BOUND. Please re-enter."
6700         BEEP 3000,1.0
6710         GOTO 6630
6720      ELSE
6730      END IF
6740 !______________________________
6750      PRINT USING 5440;Dummy1$
6760      Emax=.05493*Strain_scale/Gauge_lenght
6770      PRINT USING 6780
6780      IMAGE 2/,20X,"ENTER MEAN STRAIN (in %)"
6790      BEEP
6800      INPUT Mean
6810      Mean=Mean/100
6820      IF Mean<=-Emax OR Mean>=Emax THEN
6830         PRINT USING 6840
6840         IMAGE 2/,20X,"VALUE OUT-OF-BOUND. Please re-enter"
6850         BEEP 3000,1.0
6860         GOTO 6770
6870      ELSE
6880      END IF
6890 !________________________________
6900         PRINT USING 5440;Dummy1$
```

```
6910          PRINT USING 6920
6920          IMAGE 2/,20X,"ENTER STRAIN AMPLITUDE (in %)"
6930          BEEP
6940          INPUT Span
6950          Span=Span/100
6960          IF Span<=0 THEN
6970             PRINT USING 6980
6980             IMAGE 2/,20X,"UNDEFINED VALUE.  Please re-enter."
6990             BEEP 3000,1.0
7000             GOTO 6910
7010          ELSE
7020             IF (ABS(Mean)+Span)>=Emax THEN
7030                PRINT USING 7040
7040                IMAGE 2/,20X,"VALUE OUT-OF-BOUND. Please re-enter."
7050                BEEP 3000,1.0
7060                GOTO 6770
7070             ELSE
7080             END IF
7090          END IF
7100   !_________________________________
7110          PRINT USING 5440;Dummy1$
7120          PRINT USING 7130
7130          IMAGE 2/,20X,"ENTER LOAD SCALE (100,50,20 or 10%)"
7140          BEEP
7150          INPUT Load_scale
7160          Load_scale=Load_scale
7170          IF Load_scale<>100 AND Load_scale<>50 AND Load_scale<>20 AND Load_s
cale<>10 THEN
7180             PRINT USING 7190
7190             IMAGE 2/,20X,"UNDEFINED VALUE. Please re-enter"
7200             BEEP 3000,1.0
7210             GOTO 7120
7220          ELSE
7230          END IF
7240   END IF
7250 Continue: !
7260   Scale=Load_scale+Strain_scale
7270   Gauge=Gauge_lenght
7280   Mode=Mode_of_testing
7290   PRINT USING 5440;Dummy1$
7300   BEEP 800,2
7310   PRINT USING "/,20X,K";"** PROGRAM THE Micro_Data Track **"
7320   !
7330   SUBEND
```

```
10!    *********************************************************************
20!    *                                                                   *
30!    *                         PROGRAM FCP                               *
40!    *                         ***********                               *
50!    *                                                                   *
60!    *       THIS PROGRAM PERFORMS THE REAL-TIME CONTROL AND DATA        *
70!    *       ACQUISITION OF FATIGUE CRACK GROWTH TESTS.                  *
80!    *                                                                   *
90 !   *       THE CYCLIC DATA(Stresses and potentials) ARE                *
100!   *       RECORDED EVERY 5 CYCLES.                                    *
110!   *                                                                   *
120!   *********************************************************************
130    !
140    OPTION BASE 0
150    CALL Read_param(Material$,Scale,Mean,Span,Frequency)
160    !DELSUB Read_param
170    CALL Fcp(Material$,Scale,Mean,Span,Frequency)
180    END
190!*************************************************************************
200!*************************************************************************
210    !
220    SUB Fcp(Material$,Scale,Mean,Span,Frequency)
230    MASS STORAGE IS ":HP8290X,700,1"
240    !
250    !______ _______________DEFINE I/O CARD FUNCTION_________________________
260    !                     ------------------------
270    ASSIGN @Multi TO 723
280    ASSIGN @Dvm TO 720
290    ASSIGN @Disc TO "MRZ"
300    Disc_drive=0
310    Multi=723
320    Scanner=0
330    A_to_d=3
340    Memory1=7
350    D_to_a1=10
360    Timer=11
370    D_to_a2=12
380    Memory2=13
390    !
400    !_________________DEFAULT VALUES FOR THE PARAMETERS____________________
410    !                 ---------------------------------
420    Start=2                                               !Start channel
430    Stop=2                                                !Stop channel
440    Cycle_per_block=1
450    Data_per_cycle=200
460    IF Frequency>=20. THEN
470       Increment=249
480    ELSE
490      IF Frequency>=10 THEN
500        Increment=149
510      ELSE
520        IF Frequency>=1 THEN
530          Increment=49
540        ELSE
```

```
550          Increment=4
560          IF Increment>=.1 THEN Increment=9
570        END IF
580      END IF
590    END IF
600    !
610    !______________________INITIALIZE THE MULTI_____________________________
620    !
630    CALL Init_multi(@Multi,@Disc)
640    CALL Init_mem(@Multi,Memory1)
650    !
660    !
670    !____________________INITIALIZE READING PROCEDURE_______________________
680    !                    ----------------------------
690    CALL Init_read(Cycle_per_block,Data_per_cycle,Frequency,Pace1,Pace2,Ref_co
unt)
700    ALLOCATE Scan(Ref_count),Pot(Ref_count)
710    !                 ----------------------------------------
720    BEEP
730    Cycle=1                                !Set cycle counter
740    Next_cycle=Cycle+Increment             !Set Next_cycle counter
750    G=SPOLL(723)                           !Clear Multi SRQ
760    PRINT USING 770
770    IMAGE 20/,15X,"REMOVE CONTROL AND UTILITY PAC DISCS",2/,15X,"INSERT TWO DI
SCS FOR DATA STORAGE",2/,15X,"PRESS K0(Continue) TO CONTINUE"
780    ON KEY 0 LABEL "**CONTINUE**" GOTO 800
790 Spin: GOTO Spin
800        OFF KEY
810        CREATE BDAT "TEST_PARAM",1,100
820        ASSIGN @Disc TO "TEST_PARAM"
830        OUTPUT @Disc;Material$,"FCP",Scale,Mean,Span,Frequency
840        ASSIGN @Disc TO *
850        CALL Plot_frame(Material$,Scale,Mean,Span,Frequency,Max)
860        !DELSUB Plot_frame
870        PRINT USING 880
880        IMAGE 20/,25X,"MULTIPROGRAMMER IS READY",2/,20X,"****WAITING FOR THE S
TART COMMAND****"
890        ON KEY 0 LABEL " START CYCLING" GOTO Start_fcp
900 Wait:   GOTO Wait
910    !
920    !____________________START CYCLING AND TAKING DATA______________________
930    !                    -----------------------------
940 Start_fcp:!
950    OFF KEY
960    Vmax1=0
970    !__________CONTROL PROCEDURE______
980    !          -----------------
990    OUTPUT @Dvm;"F1R-2N4T1Z0D3    *FCP*"
1000   CALL Init_control(@Multi,Memory2,Timer,D_to_a1,Scale,Mean,Span,Frequency,N
data,Pace,Paces)
1010   !
1020   !_________READ DATA___________
1030   !         ---------
1040   OUTPUT @Multi;"WF",Timer+.2,0,"T"
```

```
1050  IF Frequency>.185 THEN OUTPUT @Multi;"WF",Timer,Paces,"T"
1060  OUTPUT @Multi;"WF",Timer+.2,1,"T"
1070  CALL Read_data(@Multi,@Dvm,Frequency,Timer,Pace,Paces,Scanner,Memory1,Pace
1,Pace2,Ref_count,Scan(*),Pot(*))
1080  !__________DISPLAY DATA__________
1090  !          ------------
1100  CALL Plot_data(Frequency,Max,Cycle,Pot(*),Vo,Vmax)
1110  Vmax2=Vmax
1120  !
1130  !__________STORE DATA___________
1140  !          ----------
1150  CALL Store_data(@Disc,Disc_drive,Cycle,Scan(*),Pot(*),Vmax1,Vmax2)
1160  !
1170  !__________SET WAITING FOR SYNCHRO. SIGNAL_________
1180  !          --------------------------------
1190 Synchro:                                         !
1200  OUTPUT @Multi;"WF",Timer+.2,0,"T"               !Stop cycling
1210  OUTPUT @Multi;"RC"                              !Read real-time clock(T2)
1220  ENTER 72314;A1,B1,C1,D1                         !Enter real-time clock(T1)
1230  ENTER 72314;A2,B2,C2,D2                         !Enter real-time clock(T2)
1240  T1=60*(A1*60*24+B1*60+C1)+D1
1250  T2=60*(A2*60*24+B2*60+C2)+D2
1260  Time_delay=T2-T1                                !Compute elapsed time
1270  Wait_time=Increment/Frequency                   !Wait between read data
1280  IF Wait_time<Time_delay THEN
1290     Cycle=INT(Cycle+1.5+Time_delay*Frequency)
1300     Next_cycle=Cycle+Increment
1310     OUTPUT @Multi;"WF",Timer+.2,1,"T"              !Set continuous mode
1320     Vmax1=Vmax2
1330     GOTO 1030
1340  ELSE
1350     Cycle=Next_cycle
1360     Next_cycle=Cycle+Increment+INT(Time_delay*Frequency+.5)
1370     Waitime=Wait_time
1380     Vmax1=Vmax2
1390     OUTPUT @Multi;"WF",Timer+.2,1,"T"              !Set continuous mode
1400     OUTPUT @Multi;"CY",Timer,"T"                   !Resume cycling
1410     WAIT Waitime                                   !Wait next read data
1420     GOTO 1030
1430  END IF
1440  SUBEND
1450!______________________________________________________________________
```

```
1460!
1470!  ******************************************************************
1480!  *                                                                *
1490!  *         SUBROUTINES USED BY SUB FCP(Param1,Param2,...)         *
1500!  *                                                                *
1510!  ******************************************************************
1520!
1530!__________________________________________________________________________
1540  SUB Init_control(@Multi,Memory,Timer,D_to_a,Scale,Mean,Span,Frequency,Ndat
a,Pace,Paces)
1550    !__GENERATE WAVEFORM__
1560    !  ----------------
1570    Load_scale=INT(Scale)/100
1580    Pmean=Mean/(Load_scale*216.078)       !Convert MPA into voltage
1590    Pmax=Span/(Load_scale*216.078)        !Convert MPa into voltage
1600    Coeff=800
1610    IF Pmax>5.12 THEN Coeff=400
1620    Ndata=INT(Coeff*Pmax+.5)-1
1630    ALLOCATE A(Ndata)
1640    Ndata2=INT(Ndata/4+.5)
1650    Ndata3=Ndata-Ndata2
1660    FOR I=0 TO Ndata2-1
1670      A(I)=Pmax*I/(Ndata2-1)
1680      A(I+Ndata3)=Pmax*(I/(Ndata2-1)-1)
1690    NEXT I
1700    FOR I=Ndata2 TO Ndata3
1710      A(I)=(Ndata-1-2*I)*Pmax/(Ndata+1-2*Ndata2)
1720    NEXT I
1730    !
1740    !__CALCULATE THE PACE__
1750    !  ----------------
1760    Time_int=1/(Ndata*Frequency)
1770    Pace=INT(Time_int*500000)/1000              !Pace in msec
1780    Paces=Pace
1790    IF Frequency>.185 THEN Paces=INT(500000/(Ndata*.185))/1000
1800    !
1810    !__PROGRAM MULTI__AMPLITUDE
1820    !  -----------------------
1830    OUTPUT @Multi;"CC",Memory+1,"T"             !Clear memory card
1840    OUTPUT @Multi;"SF",Memory,3,1,.005,12,"T"  !Set format(LSB=0.005)
1850    OUTPUT @Multi;"WF",Memory+1.0,Ndata,"T"    !Truncate memory
1860    OUTPUT @Multi;"MO",Memory,A(*),"T"          !Load data into memory
1870    OUTPUT @Multi;"WF",Memory+.1,10,"T"         !Set re-circulate mode
1880    OUTPUT @Multi;"WF",Timer+.2,1,"T"           !Set timer for continuous
1890    OUTPUT @Multi;"WF",Timer,Pace,"T"           !output. Set Pace
1900    !
1910    !__PROGRAM MULTI-- MEAN
1920    !  --------------------
1930    Incr=.005
1940    IF Pmean=0 THEN GOTO 2020
1950    IF Pmean<0 THEN Incr=-.005
1960    J=0
1970    OUTPUT @Multi;"OP",D_to_a,J,"T"
1980    IF ABS(J)<ABS(Pmean) THEN
```

```
1990        J=J+Incr
2000        GOTO 1970
2010     END IF
2020     DEALLOCATE A(*)
2030     SUBEND
2040!------------------------------------------------------------------------------
2050  SUB Read_data(@Multi,@Dvm,Frequency,Timer,Pace,Paces,Scanner,Memory,Pace1,
Pace2,Ref_count,Scan(*),Pot(*))
2060     G=SPOLL(@Multi)
2070     ENTER 72310;A,B,C,D,E,F
2080     ON INTR 7 GOTO Interrupt                      !Set interrupt branch
2090     ENABLE INTR 7;2                               !Enable Interrupt
2100     OUTPUT @Multi;"CY",Scanner,"T"                !Enable scanner
2110     OUTPUT @Multi;"WF",Scanner+.3,5,"T"           !Set sequential mode
2120     !
2130     OUTPUT @Multi;"WF",Scanner,2,"T"              !Set start channel
2140     OUTPUT @Multi;"WF",Scanner+.1,2,"T"           !set stop channel
2150     OUTPUT @Multi;"WF",Scanner+.2,Pace1,"T"       !Set pace(in usec)
2160     Prog_diff(@Multi,Memory,0)                    !Reset diff. counter
2170     Prog_write(@Multi,Memory,0)                   !Reset write pointer
2180     OUTPUT @Multi;"CC";Memory,"T"                 !Clear Memory card
2190     Prog_mode(@Multi,Memory,56)                   !Set FIFO mode
2200     Prog_ref1(@Multi,Memory,Ref_count)            !Set stop pointer
2210     OUTPUT @Multi;"AC";Memory;"T"                 !Armed memory card
2220     OUTPUT @Multi;"SC,0,0,0,0T"                   !Reset clock
2230     OUTPUT @Multi;"CY",Timer,"T"                  !Start cycling
2240     OUTPUT @Multi;"CY",Scanner,"T"                !Start pacer
2250     FOR I=0 TO Ref_count
2260       ENTER @Dvm;Pot(I)                           !Read potential
2270       WAIT Pace2
2280     NEXT I
2290 Spin:    GOTO Spin
2300 Interrupt:        !
2310       OUTPUT @Multi;"WF",Timer+.2,0,"T"           !Check pace
2320       OUTPUT @Multi;"RC"                          !Read real-time clock(T1)
2330       IF Frequency>.185 THEN OUTPUT @Multi;"WF",Timer,Pace,"T"
2340       OUTPUT @Multi;"WF",Timer+.2,1,"T"           !Set continuous mode
2350       OUTPUT @Multi;"CY",Timer,"T"                !Continue cycling
2360       G=SPOLL(@Multi)
2370       IF G=64 THEN
2380         ENTER 72310;A,B,C
2390         IF C<>0 THEN
2400            ENTER 72312;Address
2410            IF Address<>Memory THEN
2420               PRINT "NOT MEMORY CARD"
2430               PAUSE
2440            END IF
2450            OUTPUT @Multi;"WF",Scanner+.2,0,"T"       !Stop scanner
2460            Prog_mode(@Multi,Memory,76)               !Set FIFO lockout
2470            OUTPUT @Multi;"MR";Memory;Ref_count+1;"T"!MR command to get data
2480            ENTER 72305 USING "%,W";Scan(*)           !Entering the data
2490         ELSE
2500            BEEP 2000,.3
2510            PRINT "          ***  SRQ NOT SET BY ARMED CARD  ***"
```

```
2520            PAUSE
2530         END IF
2540      ELSE
2550         BEEP 2000,.3
2560         PRINT "         ***  MULTI DID NOT INTERRUPT  ***"
2570         PAUSE
2580      END IF
2590    SUBEND
2600!_______________________________________________________________________
2610   SUB Store_data(@Disc,Disc_drive,Cycle,Scan(*),Pot(*),Vmax1,Vmax2)
2620   ON ERROR GOTO Recover
2630   IF ABS(Vmax2-Vmax1)*1000<.005 THEN GOTO Out
2640   CREATE BDAT "CYCLE"&VAL$(Cycle),1,4000
2650   ASSIGN @Disc TO "CYCLE"&VAL$(Cycle)
2660   OUTPUT @Disc;Cycle,Scan(*),Pot(*)
2670   ASSIGN @Disc TO *
2680   GOTO Out
2690   !
2700 Recover: !
2710     IF ERRN=59 OR ERRN=64 THEN
2720        IF Disc_drive=0 THEN
2730           Disc_drive=1
2740           MASS STORAGE IS ":HP8290X,700,1"
2750        ELSE
2760           Disc_drive=0
2770           MASS STORAGE IS ":HP8290X,700,0"
2780        END IF
2790     ELSE
2800        BEEP 2000,1
2810        PRINT USING 2820;ERRN
2820        IMAGE 2/,25X,"ERROR NUMBER ",K,/,25X,"CHECK STORAGE UNIT",/,25X,"PROG
RAM ABORTED"
2830        PAUSE
2840     END IF
2850     GOTO 2640
2860 Out:         !
2870     SUBEND
2880!_______________________________________________________________________
2890 SUB Init_multi(@Multi,@Disc)
2900   CLEAR 7
2910   WAIT 4.0
2920   G=SPOLL(@Multi)
2930   IF G<>64 THEN
2940      PRINT "      ***MULTI DID NOT INTERRUPT***"
2950      PAUSE
2960   END IF
2970   STATUS @Multi,3;Multi
2980   ENTER Multi*100+10;A
2990   IF A<>16384 THEN
3000     PRINT "      ***SELF TEST DID NOT SET SRQ***"
3010     PAUSE
3020   END IF
3030  ALLOCATE Ascii$[80]
3040  ON END @Disc GOTO Eof
```

```
3050 Rd_file:    ENTER @Disc;Ascii$
3060     OUTPUT @Multi;Ascii$
3070     GOTO Rd_file
3080 Eof: OFF END @Disc
3090     ASSIGN @Disc TO *
3100     DEALLOCATE Ascii$
3110     MASS STORAGE IS ":HP8290X,700,0"
3120 SUBEND
3130!-------------------------------------------------------------------
3140 SUB Init_mem(@Multi,Memory)
3150   Prog_mode(@Multi,Memory,254)
3160   Prog_mode(@Multi,Memory,54)
3170   Prog_ref1(@Multi,Memory,1048575)
3180   Prog_ref2(@Multi,Memory,0)
3190   Prog_read(@Multi,Memory,0)
3200   Prog_diff(@Multi,Memory,0)
3210   Prog_write(@Multi,Memory,0)
3220   OUTPUT @Multi;"CC";Memory;"T"
3230 SUBEND
3240!-------------------------------------------------------------------
3250 SUB Prog_read(@Multi,Memory,Read_pointer)
3260   Read_low=Read_pointer MOD 65536
3270   Read_high=Read_pointer DIV 65536
3280   Read_low_oct=FNDec_to_octal(Read_low)
3290   Read_high_oct=FNDec_to_octal(Read_high)
3300   OUTPUT @Multi;"WF";Memory+.1;22;Memory+.2;Read_low_oct;"T"
3310   OUTPUT @Multi;"WF";Memory+.1;23;Memory+.2;Read_high_oct;"T"
3320 SUBEND
3330!-------------------------------------------------------------------
3340 SUB Prog_write(@Multi,Memory,Write_pointer)
3350   Write_low=Write_pointer MOD 65536
3360   Write_high=Write_pointer DIV 65536
3370   Write_low_oct=FNDec_to_octal(Write_low)
3380   Write_high_oct=FNDec_to_octal(Write_high)
3390   OUTPUT @Multi;"WF";Memory+.1;24;Memory+.2,Write_low_oct;"T"
3400   OUTPUT @Multi;"WF";Memory+.1;25;Memory+.2;Write_high_oct;"T"
3410 SUBEND
3420!-------------------------------------------------------------------
3430 SUB Prog_diff(@Multi,Memory,Diff_counter)
3440   Diff_low=Diff_counter MOD 65536
3450   Diff_high=Diff_counter DIV 65536
3460   Diff_low_oct=FNDec_to_octal(Diff_low)
3470   Diff_high_oct=FNDec_to_octal(Diff_high)
3480   OUTPUT @Multi;"WF";Memory+.1;20;Memory+.2;Diff_low_oct;"T"
3490   OUTPUT @Multi;"WF";Memory+.1;21;Memory+.2;Diff_high_oct;"T"
3500 SUBEND
3510!-------------------------------------------------------------------
3520 SUB Prog_ref1(@Multi,Memory,Ref1_reg)
3530   Ref1_low=Ref1_reg MOD 65536
3540   Ref1_high=Ref1_reg DIV 65536
3550   Ref1_low_oct=FNDec_to_octal(Ref1_low)
3560   Ref1_high_oct=FNDec_to_octal(Ref1_high)
3570   OUTPUT @Multi;"WF";Memory+.1,26;Memory+.2;Ref1_low_oct;"T"
3580   OUTPUT @Multi;"WF";Memory+.1,27;Memory+.2;Ref1_high_oct;"T"
3590 SUBEND
```

```
3600!-----------------------------------------------------------------------------
3610 SUB Prog_ref2(@Multi,Memory,Ref2_reg)
3620   Ref2_low=Ref2_reg MOD 65536
3630   Ref2_high=Ref2_reg DIV 65536
3640   Ref2_low_oct=FNDec_to_octal(Ref2_low)
3650   Ref2_high_oct=FNDec_to_octal(Ref2_high)
3660   OUTPUT @Multi;"WF";Memory+.1;30;Memory+.2;Ref2_low_oct;"T"
3670   OUTPUT @Multi;"WF";Memory+.1;31;Memory+.2;Ref2_high_oct;"T"
3680 SUBEND
3690!-----------------------------------------------------------------------------
3700 SUB Prog_mode(@Multi,Memory,Mode)
3710   OUTPUT @Multi;"WF";Memory+.3;Mode;"T"
3720 SUBEND
3730!-----------------------------------------------------------------------------
3740 SUB Read_status(@Multi,Memory,Status)
3750   STATUS @Multi,3;Multi
3760   OUTPUT @Multi;"WF";Memory+.1;1;"T RV";Memory+.2;"T"
3770   ENTER Multi*100+6;Status
3780 SUBEND
3790!-----------------------------------------------------------------------------
3800 SUB Read_read(@Multi,Memory,Read_pointer)
3810   STATUS @Multi,3;Multi
3820   OUTPUT @Multi;"WF";Memory+.1;22;"T RV";Memory+.2;"T"
3830   ENTER Multi*100+6;Read_low_oct
3840   OUTPUT @Multi;"WF";Memory+.1;23;"T RV";Memory+.2;"T"
3850   ENTER Multi*100+6;Read_high_oct
3860   Read_low=FNOctal_to_dec(Read_low_oct)
3870   Read_high=FNOctal_to_dec(Read_high_oct)
3880   Read_pointer=65536*Read_high+Read_low
3890 SUBEND
3900!-----------------------------------------------------------------------------
3910 SUB Read_write(@Multi,Memory,Write_pointer)
3920   STATUS @Multi,3;Multi
3930   OUTPUT @Multi;"WF";Memory+.1;24;"T RV";Memory+.2;"T"
3940   ENTER Multi*100+6;Write_low_oct
3950   OUTPUT @Multi;"WF";Memory+.1;25;"T RV";Memory+.2;"T"
3960   ENTER Multi*100+6;Write_high_oct
3970   Write_low=FNOctal_to_dec(Write_low_oct)
3980   Write_high=FNOctal_to_dec(Write_high_oct)
3990   Write_pointer=65536*Write_high+Write_low
4000 SUBEND
4010!-----------------------------------------------------------------------------
4020 SUB Read_diff(@Multi,Memory,Diff_counter)
4030   STATUS @Multi,3;Multi
4040   OUTPUT @Multi;"WF";Memory+.1;20;"T RV";Memory+.2;"T"
4050   ENTER Multi*100+6;Diff_low_oct
4060   OUTPUT @Multi;"WF";Memory+.1;21;"T RV";Memory+.2;"T"
4070   ENTER Multi*100+6;Diff_high_oct
4080   Diff_low=FNOctal_to_dec(Diff_low_oct)
4090   Diff_high=FNOctal_to_dec(Diff_high_oct)
4100   Diff_counter=65536*Diff_high+Diff_low
4110 SUBEND
4120!-----------------------------------------------------------------------------
```

```
4120!-----------------------------------------------------------------------
4130 DEF FNDec_to_octal(Dec)
4140   Oct=Dec+2*INT(Dec/8)+20*INT(Dec/64)+200*INT(Dec/512)+2000*INT(Dec/4096)+2
0000*INT(Dec/32768)+200000*INT(Dec/262144)
4150   RETURN Oct
4160 FNEND
4170!-----------------------------------------------------------------------
4180 DEF FNOctal_to_dec(Octal)
4190   X9=0
4200   X7=Octal
4210   X=0
4220 More:    !
4230   X=X+(X7-INT(X7/10)*10)*8^X9
4240   X7=INT(X7/10)
4250   X9=X9+1
4260   IF X7<>0 THEN GOTO More
4270   RETURN X
4280 FNEND
4290!-----------------------------------------------------------------------
4300  SUB Plot_frame(Material$,Scale,Mean,Span,Frequency,Max)
4310  ALLOCATE A(20),B(20)
4320  IF Frequency>=10 THEN
4330    Max=7
4340  ELSE
4350    IF Frequency>=1 THEN
4360      Max=6
4370    ELSE
4380      IF Frequency>=.1 THEN
4390        Max=5
4400      ELSE
4410        Max=4
4420      END IF
4430    END IF
4440  END IF
4450  FOR I=0 TO Max
4460    A(I)=(100/Max)*I-50
4470  NEXT I
4480  FOR I=0 TO 20
4490    B(I)=3*I-40
4500  NEXT I
4510  DEG
4520  GINIT
4530  GCLEAR
4540  ALPHA ON
4550  GRAPHICS OFF
4560  FRAME
4570  WINDOW -66.7224080268,66.7224080268,-50,50
4580  LDIR 0
4590  C=.75
4600  D=1.5
4610  MOVE -50,-40
4620  FOR I=0 TO Max-1
4630    FOR J=2 TO 10 STEP 2
4640      DRAW (100/Max)*LGT(J*10^I)-50,-40
4650      IF J=10 THEN
```

```
4660         IDRAW 0,D
4670         IMOVE 0,-D
4680      ELSE
4690         IDRAW 0,C
4700         IMOVE 0,-C
4710      END IF
4720    NEXT J
4730  NEXT I
4740  FOR I=0 TO 20
4750    DRAW 50,B(I)
4760    IF 2*INT(I/2)=I THEN
4770       IDRAW -D,0
4780       IMOVE D,0
4790    ELSE
4800       IDRAW -C,0
4810       IMOVE C,0
4820    END IF
4830  NEXT I
4840  FOR I=Max-1 TO 0 STEP -1
4850    FOR J=10 TO 2 STEP -2
4860      DRAW (100/Max)*LGT(J*10^I)-50,20
4870      IF J=10 THEN
4880        IDRAW 0,-D
4890        IMOVE 0,D
4900      ELSE
4910        IDRAW 0,-C
4920        IMOVE 0,C
4930      END IF
4940    NEXT J
4950  NEXT I
4960  FOR I=20 TO 0 STEP -1
4970    DRAW -50,B(I)
4980    IF 2*INT(I/2)=I THEN
4990       IDRAW D,0
5000       IMOVE -D,0
5010    ELSE
5020       IDRAW C,0
5030       IMOVE -C,0
5040    END IF
5050  NEXT I
5060  MOVE 50,20
5070  DRAW 50,40
5080  DRAW -50,40
5090  DRAW -50,20
5100  CSIZE 3,.6
5110  MOVE -50,-43
5120  LABEL USING "D";1
5130  FOR I=1 TO Max
5140    MOVE A(I)+1,-43
5150    LABEL USING "D";I
5160    IMOVE -3,1
5170    LABEL USING "DD";10
5180  NEXT I
5190  FOR I=0 TO 18 STEP 2
```

```
5200     MOVE -56,3*I-41
5210     LABEL USING "D.D";I/4
5220     MOVE 51,3*I-41
5230     LABEL USING "D.D";I/4
5240  NEXT I
5250  CSIZE 4,.6
5260  MOVE -5,-49
5270  LABEL "CYCLE"
5280  CSIZE 3,.6
5290  LDIR 90
5300  MOVE -58,-13
5310  IDRAW 0,3
5320  IDRAW -1.5,-1.5
5330  IDRAW 1.5,-1.5
5340  MOVE -57,-9
5350  LABEL "V/Vo"
5360  LDIR -90
5370  MOVE 59,-4
5380  IDRAW 0,-3
5390  IDRAW 1.5,1.5
5400  IDRAW -1.5,1.5
5410  MOVE 59,-8
5420  LABEL "V/Vo"
5430  LDIR 0
5440  CSIZE 3.5,.6
5450  MOVE -45,34
5460  LABEL Material$
5470  MOVE -45,30
5480  LABEL "FCP"
5490  MOVE -45,26
5500  LABEL "Frequency= "&VAL$(Frequency)&" Hz"
5510  MOVE 15,34
5520  LABEL "STRESS CONTROL"
5530  MOVE 15,31
5540  IDRAW 3,0
5550  IDRAW -1.5,1.5
5560  IDRAW -1.5,-1.5
5570  MOVE 19,30
5580  LABEL "S = "&VAL$(Span*2)&" (MPa)"
5590  MOVE 19,26
5600  LABEL "R = "&VAL$(DROUND((Mean-Span)/(Mean+Span),2))
5610  SUBEND
5620!____________________________________________________________________________
5630  SUB Init_read(Cycle_per_block,Data_per_cycle,Frequency,Pace1,Pace2,Ref_cou
nt)
5640  Ref_count=(Cycle_per_block*Data_per_cycle)-1
5650  IF Frequency>=.185 THEN
5660     Pace1=INT(1.E+6/(Data_per_cycle*.185)-30)         !Pace in usec
5670     Pace2=0.
5680  ELSE
5690     Pace1=INT(1.E+6/(Data_per_cycle*Frequency)-30)    !Pace in usec
5700     Pace2=DROUND((1/Frequency-1/37)/200,3)
5710  END IF
5720  SUBEND
```

```
5730!_________________________________________________________________________
5740  SUB Plot_data(Frequency,Max,Cycle,Pot(*),Vo,Vmax)
5750  ALPHA OFF
5760  GRAPHICS ON
5770  CSIZE 3,.6
5780  Maximum=0
5790  FOR J=37 TO 62
5800     IF Pot(J)>Maximum THEN Maximum=Pot(J)
5810  NEXT J
5820    Vmax=Maximum
5830 Vmax=.55
5840    IF Cycle>1 THEN GOTO Data
5850    Vo=Vmax
5860    MOVE -50,0
5870 Data: !
5880    X=(100/Max)*LGT(Cycle)-50
5890    DRAW X,12*(Vmax/Vo)-40
5900   SUBEND
5910!_________________________________________________________________________
5920  SUB Read_param(Material$,Scale,Mean,Span,Frequency)
5930  ALLOCATE Dummy$[40],Dummy1$[80]
5940  Dummy$="****************************************"
5950  Dummy1$="*****************************************************************
***************"
5960  IMAGE 2/,80A
5970  !
5980  PRINT USING 5960;Dummy1$
5990  IMAGE 2/,80A
6000  PRINT USING 6010
6010  IMAGE 2/,20X,"ENTER TYPE OF ALLOY (10 Characters max)"
6020  BEEP
6030  INPUT Material$
6040      PRINT USING 5960;Dummy1$
6050      PRINT USING 6060
6060      IMAGE 2/,20X,"ENTER LOAD SCALE? (100,50,20 or 10 %)"
6070      BEEP
6080      INPUT Load_scale
6090      Load_scale=Load_scale
6100      IF Load_scale<>100 AND Load_scale<>50 AND Load_scale<>20 AND Load_scal
e<>10 THEN
6110          PRINT USING 6120
6120          IMAGE 2/,20X,"Undefined value. Please re-enter."
6130          BEEP 3000,1.0
6140          GOTO 6050
6150      ELSE
6160      END IF
6170    PRINT USING 5960;Dummy1$
6180    PRINT USING 6190
6190    IMAGE 2/,20X, "ENTER MEAN STRESS (in MPa)?"
6200    BEEP
6210    INPUT Mean
6220    IF (Mean*11.569896)<=-(250*Load_scale) OR Mean>=(250*Load_scale) THEN
6230       PRINT USING 6240
6240       IMAGE 2/,20X,"VALUE OUT-OF-BOUND. Please re-enter."
```

```
6250        BEEP 3000,1.0
6260        GOTO 6180
6270     ELSE
6280     END IF
6290       PRINT USING 5960;Dummy1$
6300       PRINT USING 6310
6310       IMAGE 2/,20X, "ENTER STRESS AMPLITUDE (in MPa)"
6320       BEEP
6330       INPUT Span
6340       IF Span<=0 THEN
6350          PRINT USING 6360
6360          IMAGE 2/,20X,"UNDEFINED VALUE. Please re-enter"
6370          BEEP 3000,1.0
6380          GOTO 6300
6390       ELSE
6400          IF (ABS(Mean)+Span)*11.569896>=250*Load_scale THEN
6410             PRINT USING 6420
6420             IMAGE 2/,20X,"VALUE OUT-OF-BOUND,  Please re-enter"
6430             BEEP 3000,1.0
6440             GOTO 6300
6450          ELSE
6460          END IF
6470       END IF
6480          PRINT USING 5960;Dummy1$
6490          PRINT USING 6500
6500          IMAGE 2/,20X,"ENTER THE FREQUENCY (in Hz)"
6510          BEEP
6520          INPUT Frequency
6530          PRINT USING 6540;Frequency
6540          IMAGE 2/,25X,"INPUTTED Frequency IS ",K," Hz",/
6550          PRINT USING 6560
6560          IMAGE 30X,"1.  CONTINUE",/,30X,"2.  CHANGE THE FREQUENCY"
6570          BEEP
6580          INPUT Change
6590          IF Change<>1 AND Change<>2 THEN
6600              PRINT USING 6610
6610              IMAGE /,30X,"Incorrect value. Please re-enter"
6620              BEEP 3000,1.0
6630              GOTO 6550
6640          END IF
6650          IF Change=2 THEN GOTO 6490
6660          GOTO Continue
6670  !
6680 Continue: !
6690  Scale=Load_scale
6700  PRINT USING 5960;Dummy1$
6710  BEEP 800,2
6720  PRINT USING "/,20X,K,/,30X,K";"** PROGRAM THE Micro_Data Track **","IF REQ
UIRED"
6730  !
6740  SUBEND
```

```
10    !                   PROGRAM DATA_ANALYSIS
20    !
30    !This program analyses the data obtained from isothermal
40    !and thermal-mechanical fatigue tests.  It analyses the
50    !potential signal and the corresponding hysteresis loop.
60    !The (potential vs cycles) curve is first obtained. Then
70    !the fatigue crack growth rates are derived and plotted.
80    !The cyclic stress and strain hardening curves are also
90    !obtained and stored in memory.
100   !
110   DUMP DEVICE IS 701,EXPANDED
120   OPTION BASE 0
130   COM Load(359),Strain(359),Pot(359),Temp(359),X(359),Y(359)
140   COM /Param/ Crack(999),Cycles(999),Range,INTEGER Cycle,W,Z,Nmax
150   COM /Time/ A$[10]
160   ALLOCATE Stress_max(999),Stress_min(999)
170   INTEGER Option
180   Material$="HASTELLOY-X"
190   Tmax$="401"
200   Tmin$="399"
210   Range$="0.50"
220   Counter=-1
230   Cycle=-3
240   W=0
250   MASS STORAGE IS ":HP8290X,700,0"
260   PRINTER IS 1
270   PRINT "  "
280   PRINT "  "
290   PRINT "     Enter one of the following options:"
300   PRINT "                        1. manual entering"
310   PRINT "                        2. automatic entering"
320   INPUT Option
330   PRINT "  "
340   PRINT "  "
350   PRINT "     Enter the final cycle number"
360   INPUT Nmax
370   PRINT "  "
380   PRINT "  "
390   PRINT "     DO YOU WANT A GRAPHICS DISPLAY OF THE"
400   PRINT "     CYCLIC DATA?  Y/N "
410   INPUT B$
420   IF Cycle>=Nmax THEN GOTO 770
430   IF Option=1 THEN
440      PRINT "  "
450      PRINT "  "
460      PRINT "     Enter Cycle number?"
470      INPUT Cycle
480   ELSE
490     Cycle=Cycle+4
500   END IF
510   IF B$="N" OR B$="n" THEN
520       CALL Read_data
530       CALL Pot_time(Stress_max(*),Stress_min(*),Counter)
540   ELSE
```

```
550        CALL Read_data
560        CALL Plot(Material$,Range$,Tmax$,Tmin$)
570        CALL Pot_time(Stress_max(*),Stress_min(*),Counter)
580        WAIT 2.0
590        ALPHA ON
600        GRAPHICS OFF
610        PRINT " "
620        PRINT " "
630        PRINT "      DO YOU WANT A PRINTED COPY OF THE"
640        PRINT "      PREVIOUS DISPLAY?   Y/N"
650        INPUT B1$
660        IF B1$="Y" OR B1$="y" THEN
670          ALPHA OFF
680          GRAPHICS ON
690          DUMP GRAPHICS #701
700          PRINTER IS 701
710          PRINT CHR$(12)
720          PRINTER IS 1
730        ELSE
740        END IF
750    END IF
760  GOTO 420
770  PRINTER IS 1
780  CALL Volt_time(Material$,Range$,Tmax$,Tmin$,Stress_max(*),Stress_min(*),Cou
nter)
790  CALL Fcgr(Material$,Range$,Tmax$,Tmin$,Stress_max(*),Stress_min(*),Counter)
800  CALL Store_dadn(Stress_max(*),Stress_min(*),Counter)
810  !
820  GRAPHICS OFF
830  PRINT "  "
840  PRINT "  "
850  PRINT "       PLEASE ENTER TYPE OF FCGR CURVE"
860  PRINT "              1.  da/dN = f(K-stress)"
870  PRINT "              2.  da/dN = f(K-strain)"
880  PRINT "              3.  da/dN = f(J)"
890  PRINT "              4.  da/dN = f(COD)"
900  INPUT A1
910  IF A1=1 OR A1=2 THEN
920      IF A1=1 THEN
930         LOADSUB ALL FROM "STRESS_INT"
940         CALL Stress_int
950         DELSUB Stress_int
960      ELSE
970         LOADSUB ALL FROM "STRAIN_INT"
980         CALL Strain_int
990         DELSUB Strain_int
1000     END IF
1010 ELSE
1020     IF A1=3 THEN
1030        LOADSUB ALL FROM "J_INTEGRAL"
1040        CALL J_integral
1050        DELSUB J_integral
1060     ELSE
1070        LOADSUB ALL FROM "DEL_COD"
```

```
1080          CALL Cod
1090          DELSUB Cod
1100       END IF
1110 END IF
1120 PRINT "  "
1130 PRINT "  "
1140 PRINT "      DO YOU WANT TO CONTINUE?"
1150 PRINT "                   Y/N"
1160 INPUT B$
1170 IF B$="Y" THEN
1180   GOTO 820
1190 ELSE
1200   FOR I=0 TO 10
1210      PRINT " "
1220   NEXT I
1230   PRINT "                  ****PROGRAM TERMINATED****"
1240 END IF
1250 END
1260!____________________________________________________________________
1270 SUB Read_data
1280 OPTION BASE 0
1290 COM Load(359),Strain(359),Pot(359),Temp(359),X(359),Y(359)
1300 COM /Time/ A$[10]
1310 COM /Param/ Crack(999),Cycles(999),Range,INTEGER Cycle,W,Z,Nmax
1320 INTEGER Dummy
1330 IF Cycle>1 THEN GOTO 1420
1340 PRINT "  "
1350 PRINT "  "
1360 PRINT "      INSERT the floppies containing the data. Make sure"
1370 PRINT "      that floppy#1 is in the left drive, floppy #2 is in"
1380 PRINT "      the right drive, floppy#3 is in the left drive,"
1390 PRINT "      floppy#4 in the right drive, etc."
1400 ON KEY 0 LABEL "CONTINUE" GOTO Proceed
1410 Idle:     GOTO Idle
1420 Proceed:  !
1430   ON ERROR GOTO Recover
1440   !
1450   ASSIGN @Disc TO "CYCLE"&VAL$(Cycle)
1460   ENTER @Disc;Z,Dummy,Range,A$,X(*),Y(*)
1470   ASSIGN @Disc TO *
1480   FOR I=0 TO 359
1490     Load(I)=((INT(X(I))/1000)-10)
1500     Strain(I)=((X(I)-INT(X(I)))*100)-10
1510     Temp(I)=INT(Y(I))/1000
1520     Pot(I)=Y(I)-INT(Y(I))
1530   NEXT I
1540   BEEP
1550   GOTO End
1560 Recover: !
1570   IF ERRN=56 OR ERRN=64 THEN
1580      IF W=0 THEN
1590         W=1
1600         MASS STORAGE IS ":HP8290X,700,1"
1610         GOTO 1440
```

```
1620      ELSE
1630         W=0
1640         MASS STORAGE IS ":HP8290X,700,0"
1650         PRINT "    INSERT floppy#3 into left drive and"
1660         PRINT "    floppy#4 into right drive."
1670         ON KEY 0 LABEL "CONTINUE" GOTO 1440
1680      END IF
1690  ELSE
1700      PRINT " "
1710      PRINT " "
1720      PRINT "   ERROR NUMBER =";ERRN
1730      PRINT "   PLEASE CHECK STORAGE UNITS."
1740      PRINT "   PROGRAM TERMINATED."
1750      PAUSE
1760  END IF
1770 End:     !
1780 SUBEND
1790!-----------------------------------------------------------------------------
1800  SUB Plot(Material$,Range$,Tmax$,Tmin$)
1810  OPTION BASE 0
1820  COM Load(359),Strain(359),Pot(359),Temp(359),X(359),Y(359)
1830  COM /Time/ A$[10]
1840  COM /Param/ Crack(999),Cycles(999),Range,INTEGER Cycle,W,Z,Nmax
1850  DIM A(20),B(20)
1860  DEG
1870  GINIT
1880  GRAPHICS ON
1890  GCLEAR
1900  ALPHA OFF
1910  FRAME
1920  WINDOW -66.7224080268,66.7224080268,-50,50
1930  FOR I=0 TO 20
1940    A(I)=-50+I*50/9
1950    B(I)=-40+I*2.5
1960  NEXT I
1970  MOVE -50,-40
1980  FOR I=0 TO 18
1990    DRAW A(I),-40
2000    IDRAW 0,1
2010    IDRAW 0,-1
2020  NEXT I
2030  C=.5
2040  D=1
2050  FOR I=0 TO 20
2060    DRAW 50,B(I)
2070    IF I>=10 THEN
2080       C=1
2090       D=.5
2100    ELSE
2110    END IF
2120    IF 2*INT(I/2)=I THEN
2130       IDRAW -D,0
2140       IDRAW D,0
2150    ELSE
```

```
2160        IDRAW -C,0
2170        IDRAW C,0
2180     END IF
2190  NEXT I
2200  DRAW 50,40
2210  DRAW -50,40
2220  DRAW -50,10
2230  FOR I=20 TO 0 STEP -1
2240     DRAW -50,B(I)
2250     IF I<=10 THEN
2260        C=.5
2270        D=1.0
2280     ELSE
2290     END IF
2300     IF 2*INT(I/2)=I THEN
2310        IDRAW D,0
2320        IDRAW -D,0
2330     ELSE
2340        IDRAW C,0
2350        IDRAW -C,0
2360     END IF
2370  NEXT I
2380  MOVE -50,15
2390  FOR I=0 TO 18
2400     DRAW A(I),-15
2410     IDRAW 0,1
2420     IDRAW 0,-2
2430     IDRAW 0,1
2440  NEXT I
2450  MOVE -50,10
2460  FOR I=0 TO 18
2470     DRAW A(I),10
2480     IDRAW 0,-1
2490     IDRAW 0,1
2500  NEXT I
2510  MOVE 10,10
2520  DRAW 10,40
2530  MOVE -40,25
2540  FOR I=0 TO 20 STEP 2
2550     DRAW (-40+I),25
2560     IDRAW 0,.5
2570     IDRAW 0,-1
2580     IDRAW 0,.5
2590  NEXT I
2600  MOVE -30,15
2610  FOR I=0 TO 20 STEP 2
2620     DRAW -30,15+I
2630     IDRAW .5,0
2640     IDRAW -1,0
2650     IDRAW .5,0
2660  NEXT I
2670  CSIZE 3,.6
2680  MOVE -50,-40
2690  FOR I=2 TO 17 STEP 2
```

```
2700     MOVE A(I)-3,-43
2710     LABEL USING "K";(I*20)
2720  NEXT I
2730  Mini=Pot(0)
2740  Maxi=Pot(0)
2750  FOR I=1 TO 359
2760     IF Pot(I)>Maxi THEN Maxi=Pot(I)
2770     IF Pot(I)<Mini THEN Mini=Pot(I)
2780  NEXT I
2790  Min_scale=.8
2800  Factor=1.0
2810  IF Mini*1000>Min_scale THEN
2820    Min_scale=Min_scale+.5
2830    GOTO 2810
2840  END IF
2850  Min_scale=Min_scale-.5
2860  IF Maxi*1000>Min_scale+1.0 THEN Factor=Factor*2.0
2870  FOR I=0 TO 10 STEP 2
2880     MOVE -55,B(I)-1.5
2890     LABEL USING "K";Factor*.1*I+Min_scale
2900  NEXT I
2910  FOR I=1 TO 10 STEP 2
2920     MOVE 52,B(I)-1.5
2930     LABEL USING "S.D";I/10-.5
2940  NEXT I
2950  FOR I=11 TO 20 STEP 2
2960     MOVE 50.5,B(I)-1.5
2970     LABEL VAL$((I-15)*100)
2980     MOVE -58,B(I)-1.5
2990     LABEL VAL$((I-15)*100)
3000  NEXT I
3010  CSIZE 4,.6
3020  MOVE -20,-48
3030  LABEL "TIME  (sec)"
3040  CSIZE 3,.6
3050  LDIR 90
3060  MOVE -56,-38
3070  LABEL "POT. (mV)"
3080  MOVE -58,-14
3090  LABEL "STRESS (MPa)"
3100  LDIR -90
3110  MOVE 62,8
3120  LABEL "STRESS (MPa)"
3130  MOVE 60,-18
3140  LABEL "STRAIN (%)"
3150  CSIZE 3.5,.6
3160  LDIR 0
3170  MOVE 15,32
3180  LABEL Material$
3190  MOVE 15,29
3200  LABEL "TEMPERATURE"
3210  MOVE 20,26
3220  LABEL "Tmax= "&Tmax$&" C"
3230  MOVE 20,23
```

```
3240 LABEL "Tmin= "&Tmin$&" C"
3250 MOVE 15,20
3260 LABEL "CYCLES #"&VAL$(Cycle)&" & #"&VAL$(Cycle+1)
3270 MOVE 15,17
3280 LABEL "TIME: "&A$
3290 CSIZE 3,.6
3300 MOVE -15,29
3310 DRAW -13,29
3320 DRAW -14,30
3330 DRAW -15,29
3340 MOVE -12,28
3350 IF Z=1 THEN
3360    LABEL "S=  800 MPa"
3370    MOVE -15,32
3380    LABEL "STRESS CONTROL"
3390 ELSE
3400    LABEL "E= "&Range$&"%"
3410    MOVE -15,32
3420    LABEL "STRAIN CONTROL"
3430 END IF
3440 MOVE -15,24
3450 LABEL "X-DIV=0.05%"
3460 MOVE -15,20
3470 LABEL "Y-DIV=100 MPa"
3480 MOVE -50,-2.5
3490 FOR I=0 TO 358
3500   IF ABS(Load(I)-Load(I+1))>.5 THEN GOTO 3540
3510   B1=(((Load(I)*Range*216.078)+500)/40)-15
3520   A1=(I*5/18)-50
3530   DRAW A1,B1
3540 NEXT I
3550 FOR I=1 TO 358
3560   IF ABS(Strain(I)-Strain(I-1))>.08 THEN GOTO 3640
3570   IF Strain(I)=Strain(I-1) THEN GOTO 3640
3580   IF ABS(Load(I)-Load(I+1))>.5 THEN GOTO 3640
3590   B1=((.15*Strain(I)*.005555/.515)+.0025)*5000-40
3600   A1=(I*5/18)-50
3610   MOVE A1,B1
3620   IMOVE -.667,-.8
3630   LABEL "."
3640 NEXT I
3650 MOVE -50,-27.5
3660 FOR I=0 TO 359
3670    A1=(I*5/18)-50
3680    B1=(Pot(I)*1000-Min_scale)*25/Factor-40
3690    DRAW A1,B1
3700 NEXT I
3710 A1=((.15*Strain(0)*.005555/.515)+.0025)*4000-40
3720 B1=(((Load(0)*Range*216.078)+500)/50)+15
3730 MOVE A1,B1
3740 FOR I=1 TO 359
3750    PRINT I,Load(I),Strain(I)
3760    IF ABS(Strain(I)-Strain(I-1))>.08 THEN GOTO 3820
3770    IF Strain(I)=Strain(I-1) THEN GOTO 3820
```

```
3780      A1=((.15*Strain(I)*.005555/.515)+.0025)*4000-40
3790      B1=(((Load(I)*Range*216.078)+500)/50)+15
3800      MOVE A1,B1
3810      LABEL "."
3820   NEXT I
3830   SUBEND
3840!-----------------------------------------------------------------------
3850 SUB Pot_time(Stress_max(*),Stress_min(*),K)
3860 OPTION BASE 0
3870 COM Load(359),Strain(359),Pot(359),Temp(359),X(359),Y(359)
3880 COM /Time/ A$[10]
3890 COM /Param/ Crack(999),Cycles(999),Range,INTEGER Cycle,W,Z,Nmax
3900 INTEGER I,J,L
3910 L=0
3920 K=K+1
3930 Max=Pot(L)
3940 FOR I=1 TO 179
3950   J=I+L
3960   IF Pot(J)>Max THEN Max=Pot(J)
3970 NEXT I
3980 Max_stress=Load(L+30)
3990 Min_stress=Load(179+L-29)
4000 FOR I=30 TO 70
4010   J=I+L
4020   IF Load(J)>Max_stress AND Load(J)<Max_stress+.25 THEN Max_stress=Load(J)
4030   IF Load(179+L-I)<Min_stress AND Load(179+L-I)>Min_stress-.25 THEN Min_str
ess=Load(179+L-I)
4040 NEXT I
4050 Cycles(K)=INT(Cycle+(L/180))
4060 Crack(K)=Max
4070 Stress_max(K)=Max_stress
4080 Stress_min(K)=Min_stress
4090 PRINTER IS 1
4100 PRINT USING 4110;K,Cycles(K),Crack(K),Stress_max(K),Stress_min(K)
4110 IMAGE 5X,K,5X,K,5X,K,5X,K,5X,K
4120 L=L+180
4130 IF L>180 THEN GOTO 4150
4140 GOTO 3920
4150 PRINT " "
4160 PRINTER IS 1
4170 SUBEND
4180!-----------------------------------------------------------------------
4190  SUB Fcgr(Material$,Tmax$,Range$,Tmin$,Stress_max(*),Stress_min(*),Counter)
4200  OPTION BASE 0
4210  COM Load(359),Strain(359),Pot(359),Temp(359),X(359),Y(359)
4220  COM /Time/ A$[10]
4230  COM /Param/ Crack(999),Cycles(999),Range,INTEGER Cycle,W,Z,Nmax
4240  DIM A(20),B(20)
4250  !
4260  PRINT "  "
4270  PRINT "  "
4280  PRINT "    DO YOU WANT THE EXPERIMENTAL CALIBRATION?"
4290  PRINT "                 Y/N"
4300  INPUT B$
4310  IF B$="Y" THEN
```

```
4320    PRINT "  "
4330    PRINT "  "
4340    PRINT "       PLEASE ENTER THE INITIAL AND FINAL"
4350    PRINT "       CRACK LENGHT (in mm)"
4360    INPUT Crack_init,Crack_final
4370    PRINT "  "
4380    PRINT "  "
4390    PRINT "       PLEASE ENTER THE INITIAL AND FINAL"
4400    PRINT "       VOLTAGE (in milivolt)"
4410    INPUT Volt_init,Volt_final
4420    Cal_slope=(Crack_init-Crack_final)/(Volt_init-Volt_final)
4430    Cal_coeff=Crack_init-Cal_slope*Volt_init
4440    FOR I=0 TO Counter
4450       Crack(I)=Cal_slope*Crack(I)*1000+Cal_coeff
4460    NEXT I
4470  ELSE
4480    PRINT "   "
4490    PRINT "   "
4500    PRINT "     PLEASE ENTER THE INITIAL CRACK LENGHT"
4510    PRINT "                (in mm)"
4520    INPUT Crack_init
4530    PRINT "   "
4540    PRINT "   "
4550    PRINT "     PLEASE ENTER THE INITIAL VOLTAGE"
4560    PRINT "                (in milivolt)"
4570    INPUT Volt_init
4580    PRINT "   "
4590    PRINT "   "
4600    PRINT "     PLEASE ENTER PROBES Y-SPACING"
4610    PRINT "                (in mm)"
4620    INPUT Y_spacing
4630    RAD
4640    P1=PI*Y_spacing/23.4
4650    P2=(EXP(P1)+EXP(-P1))/2
4660    P3=COS(PI*Crack_init/23.4)
4670    P4=P2/P3
4680    P5=LOG(P4+SQR(P4^2-1))
4690    P6=P5/Volt_init
4700    P1=23.4/P1
4710    P2=P2*2
4720    FOR I=0 TO Counter
4730       P3=Crack(I)*1000
4740       Crack(I)=P1*Arc(P2/(EXP(P3*P6)+EXP(-P3*P6)))
4750    NEXT I
4760  END IF
4770  DEG
4780  GINIT
4790  GRAPHICS ON
4800  GCLEAR
4810  ALPHA OFF
4820  WINDOW -66.7224080268,66.7224080268,-50,50
4830  FOR I=0 TO 20
4840    B(I)=-40+I*4.0
4850  NEXT I
```

```
4860  MOVE -50,-40
4870  FOR J=0 TO 3
4880    FOR I=2 TO 10 STEP 2
4890      A1=25*LGT(I*10^J)-50
4900      DRAW A1,-40
4910      IF I<10 THEN
4920         IDRAW 0,.7
4930         IDRAW 0,-.7
4940      ELSE
4950         IDRAW 0,1.4
4960         IDRAW 0,-1.4
4970      END IF
4980    NEXT I
4990  NEXT J
5000  C=.7
5010  D=1.4
5020  FOR I=0 TO 20
5030    DRAW 50,B(I)
5040    IF 2*INT(I/2)=I THEN
5050       IDRAW -D,0
5060       IDRAW D,0
5070    ELSE
5080       IDRAW -C,0
5090       IDRAW C,0
5100    END IF
5110  NEXT I
5120  FOR J=3 TO 0 STEP -1
5130    FOR I=10 TO 2 STEP -2
5140       A1=25*LGT(I*10^J)-50
5150       DRAW A1,40
5160       IF I<10 THEN
5170          IDRAW 0,-C
5180          IDRAW 0,C
5190       ELSE
5200          IDRAW 0,-D
5210          IDRAW 0,D
5220       END IF
5230    NEXT I
5240  NEXT J
5250  DRAW -50,40
5260  FOR I=20 TO 0 STEP -1
5270    DRAW -50,B(I)
5280    IF 2*INT(I/2)=I THEN
5290       IDRAW D,0
5300       IDRAW -D,0
5310    ELSE
5320       IDRAW C,0
5330       IDRAW -C,0
5340    END IF
5350  NEXT I
5360  CSIZE 3,.6
5370  MOVE -51,-45
5380  LABEL "1"
5390  FOR I=1 TO 4
```

```
5400   MOVE LGT(10^I)*25-53,-45
5410   LABEL VAL$(10)
5420   IMOVE 4,5
5430   LABEL VAL$(I)
5440 NEXT I
5450 FOR I=2 TO 18 STEP 2
5460    MOVE 50.5,B(I)-1.5
5470    LABEL VAL$(I*.05)
5480    MOVE -55,B(I)-1.5
5490    LABEL VAL$(I*.05)
5500 NEXT I
5510 CSIZE 4,.6
5520 MOVE -58,0
5530 LABEL "A"
5540 LDIR 90
5550 MOVE -55,-.5
5560 LABEL "|"
5570 LDIR 0
5580 MOVE -58,-4
5590 LABEL "W"
5600 MOVE 56,-0
5610 LABEL "A"
5620 LDIR 90
5630 MOVE 59,-.5
5640 LABEL "|"
5650 LDIR 0
5660 MOVE 56,-4
5670 LABEL "W"
5680 MOVE -5,-49
5690 LABEL "CYCLE"
5700 MOVE 15,32
5710 LABEL Material$
5720 MOVE 15,29
5730 LABEL "TEMPERATURE"
5740 MOVE 20,26
5750 LABEL "Tmax= "&Tmax$&" C"
5760 MOVE 20,23
5770 LABEL "Tmin= "&Tmin$&" C"
5780 MOVE -40,29
5790 DRAW -38,29
5800 DRAW -39,30
5810 DRAW -40,29
5820 MOVE -37,28
5830 IF Z=1 THEN
5840   LABEL "S=  800 MPa"
5850   MOVE -40,32
5860   LABEL "STRESS CONTROL"
5870 ELSE
5880   LABEL "E= "&Range$&"%"
5890   MOVE -40,32
5900   LABEL "STRAIN CONTROL"
5910 END IF
5920 !
5930 FOR I=0 TO Counter
```

```
5940      A1=LGT(Cycles(I))*25-50
5950      B1=(Crack(I)/11.7)*80-40
5960      MOVE A1-1,B1-1.667
5970      LABEL "."
5980  NEXT I
5990  SUBEND
6000!------------------------------------------------------------------------
6010  SUB Store_dadn(Stress_max(*),Stress_min(*),Counter)
6020  OPTION BASE 0
6030  COM Load(359),Strain(359),Pot(359),Temp(359),X(359),Y(359)
6040  COM /Time/ A$[10]
6050  COM /Param/ Crack(999),Cycles(999),Range,INTEGER Cycle,W,Z,Nmax
6060  ALLOCATE Z$[19],Z1$[19],Z2$[19],Z3$[19],Z4$[19],Dummy1(399),Dummy2(399)
6070  GRAPHICS OFF
6080  ALPHA ON
6090  PRINT "  "
6100  PRINT "  "
6110  PRINT "       DO YOU WANT TO SAVE THE A VS N DATA?"
6120  PRINT "                          (Y/N)"
6130  INPUT B$
6140  IF B$<>"Y" AND B$<>"N" THEN GOTO 6090
6150  IF B$="Y" THEN
6160     PRINT "  "
6170     PRINT "  "
6180     PRINT "       INSERT THE FLOPPY DISC INTO DRIVE#0.  MAKE"
6190     PRINT "       SURE THAT IT HAS BEEN INITIALIZED "
6200     ON KEY 0 LABEL "CONTINUE" GOTO Proceed
6210 Idle:            GOTO Idle
6220 Proceed:         !
6230     PRINT "  "
6240     PRINT "  "
6250     PRINT "       ENTER FILE NAME?"
6260     INPUT Z$
6270     PRINT "  "
6280     PRINT "  "
6290     PRINT "       ENTER MATERIAL?"
6300     INPUT Z1$
6310     PRINT "  "
6320     PRINT "  "
6330     PRINT "       ENTER TESTING CONDITION?"
6340     PRINT "       (In-phase, out-of-phase, isothermal)"
6350     INPUT Z2$
6360     PRINT "  "
6370     PRINT "  "
6380     PRINT "       ENTER TEMPERATURE RANGE (in  C)"
6390     INPUT Z3$
6400     PRINT "  "
6410     PRINT "  "
6420     PRINT "       ENTER STRAIN RANGE (in %)"
6430     INPUT Z4$
6440     FOR I=0 TO Counter
6450        Dummy1(I)=Cycles(I)+(Crack(I)/100)
6460        Dummy2(I)=((Stress_max(I)+10)*1000)+((Stress_min(I)+10)/100)
6470     NEXT I
```

```
6480      MASS STORAGE IS ":HP8290X,700,0"
6490      CREATE BDAT Z$,1,8400
6500      ASSIGN @Disc TO Z$
6510      OUTPUT @Disc;Z1$,Z2$,Z3$,Z4$,Counter,Dummy1(*),Dummy2(*)
6520      ASSIGN @Disc TO *
6530      GOTO End_sub
6540   ELSE
6550      GOTO End_sub
6560   END IF
6570 End_sub:    !
6580      DEALLOCATE Z$,Z1$,Z2$,Z3$,Z4$,Dummy1(*),Dummy2(*)
6590      SUBEND
6600!____________________________________________________________________________
6610  SUB Volt_time(Material$,Range$,Tmax$,Tmin$,Stress_max(*),Stress_min(*),Cou
nter)
6620  OPTION BASE 0
6630  COM Load(359),Strain(359),Pot(359),Temp(359),X(359),Y(359)
6640  COM /Time/ A$[10]
6650  COM /Param/ Crack(999),Cycles(999),Range,INTEGER Cycle,W,Z,Nmax
6660  DIM A(20),B(20)
6670  DEG
6680  GINIT
6690  GRAPHICS ON
6700  GCLEAR
6710  ALPHA OFF
6720  WINDOW -66.7224080268,66.7224080268,-50,50
6730  FOR I=0 TO 20
6740    B(I)=-40+I*4.0
6750  NEXT I
6760  MOVE -50,-40
6770  FOR J=0 TO 3
6780    FOR I=2 TO 10 STEP 2
6790      A1=25*LGT(I*10^J)-50
6800      DRAW A1,-40
6810      IF I<10 THEN
6820         IDRAW 0,.7
6830         IDRAW 0,-.7
6840      ELSE
6850         IDRAW 0,1.4
6860         IDRAW 0,-1.4
6870      END IF
6880    NEXT I
6890  NEXT J
6900  C=.7
6910  D=1.4
6920  FOR I=0 TO 20
6930    DRAW 50,B(I)
6940    IF 2*INT(I/2)=I THEN
6950       IDRAW -D,0
6960       IDRAW D,0
6970    ELSE
6980       IDRAW -C,0
6990       IDRAW C,0
7000    END IF
```

```
7010  NEXT I
7020  FOR J=3 TO 0 STEP -1
7030    FOR I=10 TO 2 STEP -2
7040       A1=25*LGT(I*10^J)-50
7050       DRAW A1,40
7060       IF I<10 THEN
7070          IDRAW 0,-C
7080          IDRAW 0,C
7090       ELSE
7100          IDRAW 0,-D
7110          IDRAW 0,D
7120       END IF
7130    NEXT I
7140  NEXT J
7150  DRAW -50,40
7160  FOR I=20 TO 0 STEP -1
7170    DRAW -50,B(I)
7180    IF 2*INT(I/2)=I THEN
7190       IDRAW D,0
7200       IDRAW -D,0
7210    ELSE
7220       IDRAW C,0
7230       IDRAW -C,0
7240    END IF
7250  NEXT I
7260  CSIZE 3,.6
7270  MOVE -51,-45
7280  LABEL "1"
7290  FOR I=1 TO 4
7300    MOVE LGT(10^I)*25-53,-45
7310    LABEL "10"
7320    IMOVE 4,5
7330    LABEL VAL$(I)
7340  NEXT I
7350  FOR I=2 TO 18 STEP 2
7360     MOVE 50.5,B(I)-1.5
7370     LABEL VAL$(I*.25)
7380     MOVE -56,B(I)-1.5
7390     LABEL VAL$(I*.25)
7400  NEXT I
7410  CSIZE 4,.6
7420  MOVE -58,-20
7430  LDIR 90
7440  LABEL "VOLTAGE  (in mV)"
7450  MOVE 58,20
7460  LDIR -90
7470  LABEL "VOLTAGE  (in mV)"
7480  LDIR 0
7490  MOVE -5,-49
7500  LABEL "CYCLE"
7510  MOVE 15,32
7520  LABEL Material$
7530  MOVE 15,29
7540  LABEL "TEMPERATURE"
```

```
7550  MOVE 20,26
7560  LABEL "Tmax= "&Tmax$&" C"
7570  MOVE 20,23
7580  LABEL "Tmin= "&Tmin$&" C"
7590  MOVE -40,29
7600  DRAW -38,29
7610  DRAW -39,30
7620  DRAW -40,29
7630  MOVE -37,28
7640  IF Z=1 THEN
7650    LABEL "S=  800 MPa"
7660    MOVE -40,32
7670    LABEL "STRESS CONTROL"
7680  ELSE
7690    LABEL "E= "&Range$&"%"
7700    MOVE -40,32
7710    LABEL "STRAIN CONTROL"
7720  END IF
7730  !
7740  FOR I=0 TO Counter
7750     A1=LGT(Cycles(I))*25-51
7760     B1=Crack(I)*16000-41.667
7770     MOVE A1,B1
7780     LABEL "."
7790  NEXT I
7800  WAIT 3.0
7810  SUBEND
7820!_____________________________________________________________________
```

```
10    !                    PROGRAM  STRAIN_INT
20    OPTION BASE 0
30    COM /Param/ Crack(999),Cycles(999),Range,INTEGER Cycle,W,Z,Nmax
40    Z=2
50    CALL Strain_int
60    END
70    !
80    SUB Strain_int
90    OPTION BASE 0
100   COM /Param/ Crack(999),Cycles(999),Range,INTEGER Cycle,W,Z,Nmax
110   ALLOCATE Delk(999),Dadn(999)
120   ALLOCATE Bb(3),A(10),N(10),Z1$[19],Z2$[19],Z3$[19],Z4$[19]
130   ALLOCATE C11(99),C12(99),C22(99)
140   !
150 Select:                 !****************************************
160   PRINT " "
170   PRINT " "
180   PRINT "        ENTER FILE NAME?"
190   INPUT Z$
200   !
210 Read_data:              !****************************************
220 !______________________________
230   CALL Read_const(C11(*),C12(*),C22(*))
240   MASS STORAGE IS ":HP8290X,700,1"
250   ASSIGN @Disc TO Z$
260   ENTER @Disc;Z1$,Z2$,Z3$,Z4$,Nmaxx,Crack(*)
270   ASSIGN @Disc TO *
280   MASS STORAGE IS ":HP8290X,700,0"
290   Nmax=Nmaxx
300   FOR I=0 TO Nmax
310      Cycles(I)=INT(Crack(I))
320      Crack(I)=(Crack(I)-Cycles(I))*100
330   NEXT I
340 !________________________________
350 Plot_frame:             !****************************************
360   DEG
370   GINIT
380   !PLOTTER IS 705,"HPGL"
390   GRAPHICS ON
400   GCLEAR
410   LDIR 0
420   WINDOW -66.7224080268,66.7224080268,-50,50
430   CSIZE 3,.6
440   MOVE -50,-40
450   FOR I=-5 TO -4
460      FOR J=1 TO 10
470         A1=50*(LGT(J*10^I)+5)-50
480         DRAW A1,-40
490         IF J=10 THEN
500            IDRAW 0,1.8
510            IDRAW 0,-1.8
520         ELSE
530            IDRAW 0,.8
540            IDRAW 0,-.8
```

```
550         END IF
560       NEXT J
570    NEXT I
580    FOR J=-9 TO -4
590       FOR I=1 TO 9
600          A1=(LGT(I*10^J)+9)*80/6-40
610          DRAW 50,A1
620          IF I>1 THEN
630             IDRAW -.8,0
640             IDRAW .8,0
650          ELSE
660             IDRAW -1.8,0
670             IDRAW 1.8,0
680          END IF
690       NEXT I
700    NEXT J
710    FOR I=-4 TO -5 STEP -1
720       FOR J=10 TO 1 STEP -1
730          A1=50*(LGT(J*10^I)+5)-50
740          DRAW A1,40
750          IF J=10 THEN
760            IDRAW 0,-1.8
770            IDRAW 0,1.8
780          ELSE
790            IDRAW 0,-.8
800            IDRAW 0,.8
810          END IF
820       NEXT J
830    NEXT I
840    FOR J=-4 TO -9 STEP -1
850      FOR I=9 TO 1 STEP -1
860        A1=(LGT(I*10^J)+9)*80/6-40
870        DRAW -50,A1
880        IF I>1 THEN
890           IDRAW .8,0
900           IDRAW -.8,0
910        ELSE
920           IDRAW 1.8,0
930           IDRAW -1.8,0
940        END IF
950      NEXT I
960    NEXT J
970    DRAW -50,-40
980    FOR I=-5 TO -3
990       A1=(LGT(10^I)+5)*50-50
1000      MOVE A1,-40
1010      IMOVE -1.5,-3.0
1020      LABEL VAL$(I)
1030      IMOVE -1.6,.4
1040      LABEL "10"
1050   NEXT I
1060   MOVE 50,-40
1070   FOR I=-8 TO -4
1080      A1=(LGT(10^I)+9)*80/6-40
```

```
1090     MOVE 50,A1
1100     IMOVE 1.5,-1.5
1110     LABEL "10"
1120     IMOVE 2.8,3.5
1130     LABEL VAL$(I)
1140  NEXT I
1150  MOVE -50,40
1160  FOR I=-4 TO -8 STEP -1
1170     A1=(LGT(10^I)+9)*80/6-40
1180     MOVE -50,A1
1190     IMOVE -7.0,-1.5
1200     LABEL "10"
1210     IMOVE 2.8,3.5
1220     LABEL VAL$(I)
1230  NEXT I
1240  CSIZE 3.5,.6
1250  MOVE -25,-48
1260  LABEL "Strain Intensity Factor (Vm)"
1270  MOVE -25,-48
1280  IMOVE 51,3.1
1290  IDRAW 1.6,0
1300  LDIR 90
1310  MOVE -58,-15
1320  LABEL "da/dN (m/cycle)"
1330  LDIR 0
1340  MOVE -24,42
1350  LABEL "FATIGUE CRACK PROPAGATION"
1360  MOVE -45,30
1370  LABEL Z1$
1380  MOVE -45,27
1390  LABEL "STRAIN CONTROL"
1400  MOVE -45,25
1410  IDRAW 2,0
1420  IDRAW -1,1.5
1430  IDRAW -1,-1.5
1440  MOVE -42,24
1450  LABEL "E= "&Z4$&"%"
1460  MOVE -45,21
1470  LABEL Z2$
1480  MOVE -45,18
1490  LABEL "T="&Z3$&" C"
1500  MOVE -45,15
1510  LABEL "Frequency= 0.1 Hz"
1520  MOVE -45,13
1530  IDRAW 2,0
1540  IDRAW -1,1.5
1550  IDRAW -1,-1.5
1560  MOVE -42,12
1570  LABEL "K from Eq. 32"
1580  MOVE -45,9
1590  LABEL "L /W= 2.5"
1600  MOVE -43,8
1610  LABEL "b"
1620  MOVE -50,-40
```

```
1630  !
1640 Plot_data:                !****************************************
1650  Lw=2.17
1660  Lb=2.50
1670  K=-1
1680  Deltae=VAL(Z4$)/100
1690  Npts=Nmax-6
1700  FOR I=0 TO Npts
1710    L=0
1720    K=K+1
1730    K1=K+6
1740    FOR J=K TO K1
1750      L=L+1
1760      A(L)=Crack(J)
1770      N(L)=Cycles(J)
1780    NEXT J
1790    C1=.5*(N(1)+N(7))
1800    C2=.5*(N(7)-N(1))
1810    Sx=0
1820    Sx2=0
1830    Sx3=0
1840    Sx4=0
1850    Sy=0
1860    Syx=0
1870    Syx2=0
1880    FOR J=1 TO 7
1890      Xx=(N(J)-C1)/C2
1900      Yy=A(J)
1910      Sx=Sx+Xx
1920      Sx2=Sx2+Xx^2
1930      Sx3=Sx3+Xx^3
1940      Sx4=Sx4+Xx^4
1950      Sy=Sy+Yy
1960      Syx=Syx+Xx*Yy
1970      Syx2=Syx2+Yy*Xx^2
1980    NEXT J
1990    Den=7.0*(Sx2*Sx4-Sx3^2)-Sx*(Sx*Sx4-Sx2*Sx3)+Sx2*(Sx*Sx3-Sx2^2)
2000    T2=Sy*(Sx2*Sx4-Sx3^2)-Syx*(Sx*Sx4-Sx2*Sx3)+Syx2*(Sx*Sx3-Sx2^2)
2010    Bb(1)=T2/Den
2020    T3=7.0*(Syx*Sx4-Syx2*Sx3)-Sx*(Sy*Sx4-Syx2*Sx2)+Sx2*(Sy*Sx3-Syx*Sx2)
2030    Bb(2)=T3/Den
2040    T4=7.0*(Sx2*Syx2-Sx3*Syx)-Sx*(Sx*Syx2-Sx3*Sy)+Sx2*(Sx*Syx-Sx2*Sy)
2050    Bb(3)=T4/Den
2060    Yb=Sy/7.0
2070    Rss=0
2080    Tss=0
2090    FOR J=1 TO 7
2100      Xx=(N(J)-C1)/C2
2110      Yhat=Bb(1)+Bb(2)*Xx+Bb(3)*Xx^2
2120      Rss=Rss+(A(J)-Yhat)^2
2130      Tss=Tss+(A(J)-Yb)^2
2140    NEXT J
2150    R2=1.0-Rss/Tss
2160    Dadn(I)=Bb(2)/C2+2.0*Bb(3)*(N(4)-C1)/C2^2
```

```
2170     Xx=(N(4)-C1)/C2
2180     Ar=Bb(1)+Bb(2)*Xx+Bb(3)*Xx^2
2190 !___________________________________
2200     T=Ar/11.7
2210     K11=INT((T+.005)*100)-1
2220     F11=FNF11(T)
2230     F22=FNF22(T)
2240     Factor1=1-(6*(F22/F11)*C12(K11)/(C22(K11)+12*Lb))
2250     A11=C12(K11)^2/(12*Lb+C22(K11))
2260     A22=6*C12(K11)*(Lw-Lb)/(12*Lb+C22(K11))
2270     Factor2=(1+(C11(K11)-A11+A22)/Lw)
2280     Ft=SQR(PI*Ar/1000)*F11*Factor1/Factor2
2290     !Ft=SQR(25/(20-13*T-7*T^2))         !Harris'formùla
2300     Delk(I)=Deltae*Ft
2310     IF T>=.55 THEN Delk(I)=100
2320  NEXT I
2330 BEEP 3000,1
2340 ALPHA OFF
2350  !
2360  FOR I=0 TO Npts
2370     A1=(LGT(Delk(I))+5)*50-50
2380     A2=(LGT(ABS(Dadn(I))/1000)+9)*80/6-40
2390     MOVE A1,A2
2400     IMOVE -1.3344481605,-1
2410     LABEL "."
2420  NEXT I
2430 PENUP
2440 MOVE -50,-40
2450 PAUSE
2460 SUBEND
2470!_________________________________________________________________________
2480  DEF FNF11(I)
2490  RAD
2500   IF I=0 THEN
2510     F11=2/SQR(PI)
2520     GOTO 2570
2530   END IF
2540   A=SQR(2*TAN(PI*I/2)/(PI*I))
2550   B=(.752+2.02*I+.37*(1-SIN(PI*I/2))^3)/COS(PI*I/2)
2560   F11=A*B
2570   RETURN F11
2580  FNEND
2590!_________________________________________________________________________
2600  DEF FNF22(I)
2610  RAD
2620   IF I=0 THEN
2630     F22=2/SQR(PI)
2640     GOTO 2690
2650   END IF
2660   A=SQR(2*TAN(PI*I/2)/(PI*I))
2670   B=(.923+.199*(1-SIN(PI*I/2))^4)/COS(PI*I/2)
2680   F22=A*B
2690   RETURN F22
2700  FNEND
```

```
2710!_____________________________________________________________________
2720  SUB Read_const(C11(*),C12(*),C22(*))
2730  DATA .39655E-03,.1594E-02,.36060E-02,.64504E-02,.10149E-01,.14728E-0
1,.20217E-01,.26652E-01,.34072E-01,.42522E-01
2740  DATA .52051E-01,.62715E-01,.74576E-01,.87699E-01,.10216,.11804,.1354
2,.15441,.17510,.19761
2750  DATA .22207,.24861,.27738,.30854,.34225,.37872,.41813,.46073,.50674,
.55643
2760  DATA .61009,.66802,.73057,.79811,.87105,.94981,1.0349,1.1268,1.2262,
1.3335
2770  DATA 1.4497,1.5753,1.7114,1.8586,2.0182,2.1912,2.3789,2.5827,2.8040,
3.0447
2780  DATA 3.3065,3.5917,3.9026,4.2418,4.6124,5.0177,5.4615,5.9481,6.4823,
7.0697
2790  DATA 7.7166,8.4301,9.2186,10.091,11.060,12.136,13.336,14.676,16.176,
17.861
2800  DATA 19.759,21.905,24.337,27.106,30.271,33.905,38.097,42.958,48.629,
55.284
2810  DATA 63.151,72.522,83.782,97.441,114.19,135.00,161.19,194.73,238.50,
296.92
2820  DATA 377.05,490.68,658.66,920.72,1359.9,2174.0,3930.5,8822.6,31809.,
31809.
2830!____________
2840  READ C11(*)
2850!------------
2860  DATA .23631E-02,.9402E-02,.21068E-01,.37333E-01,.58192E-01,.83658E-0
1,.11377,.14857,.18815,.23259
2870  DATA .28201,.33655,.39637,.46164,.53256,.60936,.69228,.78161,.87763,
.98069
2880  DATA 1.0911,1.2093,1.3358,1.4708,1.6151,1.7690,1.9332,2.1083,2.2950,
2.4940
2890  DATA 2.7062,2.9324,3.1735,3.4305,3.7047,3.9970,4.3090,4.6419,4.9973,
5.3769
2900  DATA 5.7826,6.2163,6.6802,7.1768,7.7085,8.2785,8.8898,9.5459,10.251,
11.008
2910  DATA 11.824,12.702,13.650,14.672,15.777,16.974,18.269,19.676,21.204,
22.867
2920  DATA 24.680,26.659,28.826,31.200,33.808,36.680,39.849,43.356,47.246,
51.573
2930  DATA 56.403,61.811,67.869,74.744,82.510,91.347,101.45,113.07,126.50,
142.13
2940  DATA 160.45,182.09,207.87,238.89,276.62,323.09,381.14,454.87,550.30,
676.68
2950  DATA 848.67,1090.7,1445.7,1995.3,2909.4,4591.0,8192.0,18147.,64577.,
64577.
2960!___________
```

```
2970  READ C12(*)
2980!------------
2990  DATA .14082E-01,.55459E-01,.12310,.21611,.33375,.47539,.64052,.82877
,1.0399,1.2736
3000  DATA 1.5299,1.8089,2.1105,2.4352,2.7830,3.1545,3.5500,3.9703,4.4159,
4.8876
3010  DATA 5.3863,5.9128,6.4683,7.0540,7.6710,8.3207,9.0047,9.7247,10.482,
11.280
3020  DATA 12.118,13.001,13.931,14.909,15.939,17.025,18.168,19.374,20.646,
21.988
3030  DATA 23.405,24.901,26.484,28.158,29.930,31.807,33.798,35.911,38.155,
40.541
3040  DATA 43.081,45.787,48.673,51.756,55.052,58.582,62.366,66.430,70.801,
75.509
3050  DATA 80.590,86.083,92.034,98.493,105.52,113.18,121.55,130.73,140.81,
151.93
3060  DATA 164.22,177.85,193.03,210.01,229.06,250.55,274.91,302.67,334.49,
371.20
3070  DATA 413.86,463.83,522.87,593.31,678.29,782.09,910.72,1072.8,1280.9,
1554.3
3080  DATA 1923.4,2438.8,3189.1,4341.7,6244.1,9717.6,17100.,37358.,131140,
131140
3090!____________
3100  READ C22(*)
3110!------------
3120  SUBEND
```

```
10    !                       PROGRAM  STRESS_INT
20    OPTION BASE 0
30    COM /Param/ Crack(999),Cycles(999),Range,INTEGER Cycle,W,Z,Nmax
40    Z=2
50    CALL Stress_int
60    END
70    !
80    SUB Stress_int
90    OPTION BASE 0
100   COM /Param/ Crack(999),Cycles(999),Range,INTEGER Cycle,W,Z,Nmax
110   ALLOCATE Delk(1300),Dadn(1300)
120   ALLOCATE Bb(3),A(10),N(10),Z1$[19],Z2$[19],Z3$[19],Z4$[19]
130   ALLOCATE C11(99),C12(99),C22(99)
140   ALLOCATE Dummy1(199),Dummy2(199)
150   ALLOCATE Stress_max(199),Stress_min(199)
160   !
170 Select:                 !****************************************
180   PRINT " "
190   PRINT " "
200   PRINT "       ENTER FILE NAME?"
210   INPUT Z$
220   PRINT "  "
230   PRINT "  "
240   PRINT "       ENTER MODULUS(in MPa)?"
250   INPUT Modulus
260   PRINT "  "
270   PRINT "  "
280   PRINT "       ENTER POISSON RATIO?"
290   INPUT Poisson
300   !
310 Read_data:              !****************************************
320 !______________________________
330   CALL Read_const(C11(*),C12(*),C22(*))
340   MASS STORAGE IS ":HP8290X,700,1"
350   ASSIGN @Disc TO Z$
360   ENTER @Disc;Z1$,Z2$,Z3$,Z4$,Counter,Dummy1(*),Dummy2(*)
370   ASSIGN @Disc TO *
380   MASS STORAGE IS ":HP8290X,700,0"
390   FOR I=0 TO Counter
400      Cycles(I)=INT(Dummy1(I))
410      Crack(I)=(Dummy1(I)-Cycles(I))*100
420      Stress_max(I)=INT(Dummy2(I))/1000-10
430      Stress_min(I)=(Dummy2(I)-INT(Dummy2(I)))*100-10
440   NEXT I
450   DEALLOCATE Dummy1(*),Dummy2(*)
460   CALL Stress_cycle(Counter,Stress_max(*),Stress_min(*),Cycles(*),Crack(*),Z
1$,Z2$,Z3$,Z4$)
470 !________________________________
480 Plot_frame:             !****************************************
490   DEG
500   GINIT
510   !PLOTTER IS 705,"HPGL"
520   GRAPHICS ON
530   GCLEAR
```

```
540   LDIR 0
550   WINDOW -66.7224080268,66.7224080268,-50,50
560   CSIZE 3,.6
570   MOVE -50,-40
580   FOR I=10 TO 100 STEP 2
590          A1=100*(LGT(I/10))-50
600          DRAW A1,-40
610          IF INT(I/10)<(I/10) THEN
620             IDRAW 0,.8
630             IDRAW 0,-.8
640          ELSE
650             IDRAW 0,1.8
660             IDRAW 0,-1.8
670          END IF
680   NEXT I
690   FOR J=-9 TO -4
700      FOR I=1 TO 9
710          A1=(LGT(I*10^J)+9)*80/6-40
720          DRAW 50,A1
730          IF I>1 THEN
740             IDRAW -.8,0
750             IDRAW .8,0
760          ELSE
770             IDRAW -1.8,0
780             IDRAW 1.8,0
790          END IF
800      NEXT I
810   NEXT J
820   FOR I=100 TO 10 STEP -2
830          A1=100*(LGT(I/10))-50
840          DRAW A1,40
850          IF INT(I/10)<(I/10) THEN
860            IDRAW 0,-.8
870            IDRAW 0,.8
880          ELSE
890            IDRAW 0,-1.8
900            IDRAW 0,1.8
910          END IF
920   NEXT I
930   FOR J=-4 TO -9 STEP -1
940     FOR I=9 TO 1 STEP -1
950       A1=(LGT(I*10^J)+9)*80/6-40
960       DRAW -50,A1
970       IF I>1 THEN
980          IDRAW .8,0
990          IDRAW -.8,0
1000      ELSE
1010         IDRAW 1.8,0
1020         IDRAW -1.8,0
1030      END IF
1040    NEXT I
1050  NEXT J
1060  DRAW -50,-40
1070  FOR I=1 TO 9
```

```
1080      A1=LGT(I)*100-50
1090      MOVE A1,-40
1100      IMOVE -1.5,-3.0
1110      LABEL VAL$(I*10)
1120  NEXT I
1130  MOVE 50,-40
1140  FOR I=-8 TO -4
1150      A1=(LGT(10^I)+9)*80/6-40
1160      MOVE 50,A1
1170      IMOVE 1.5,-1.5
1180      LABEL "10"
1190      IMOVE 2.8,3.5
1200      LABEL VAL$(I)
1210  NEXT I
1220  MOVE -50,40
1230  FOR I=-4 TO -8 STEP -1
1240      A1=(LGT(10^I)+9)*80/6-40
1250      MOVE -50,A1
1260      IMOVE -7.0,-1.5
1270      LABEL "10"
1280      IMOVE 2.8,3.5
1290      LABEL VAL$(I)
1300  NEXT I
1310  CSIZE 3.5,.6
1320  MOVE -28,-48
1330  LABEL "Stress Intensity Factor (MPaVm)"
1340  MOVE -28,-48
1350  IMOVE 57,3.1
1360  IDRAW 1.6,0
1370  LDIR 90
1380  MOVE -58,-15
1390  LABEL "da/dN (m/cycle)"
1400  LDIR 0
1410  MOVE -24,42
1420  LABEL "FATIGUE CRACK PROPAGATION"
1430  MOVE -45,30
1440  LABEL Z1$
1450  MOVE -45,27
1460  LABEL "STRAIN CONTROL"
1470  MOVE -45,25
1480  IDRAW 2,0
1490  IDRAW -1,1.5
1500  IDRAW -1,-1.5
1510  MOVE -42,24
1520  LABEL "E= "&Z4$&"%"
1530  MOVE -45,21
1540  LABEL Z2$
1550  MOVE -45,18
1560  LABEL "T="&Z3$&" C"
1570  MOVE -50,-40
1580  PENUP
1590  !PAUSE
1600  !
```

```
1610 Plot_data:              !********************************************
1620  Lw=2.17
1630  Lb=2.5
1640  K=-1
1650  Npts=Counter-6
1660  FOR I=0 TO Npts
1670   !Deltap=(Stress_max(I+3)-Stress_min(I+3))*21.6078*5!K using stress range
1680    Deltap=Modulus*VAL(Z4$)/(100*(1-Poisson)) !K using Modulus
1690    L=0
1700    K=K+1
1710    K1=K+6
1720    FOR J=K TO K1
1730      L=L+1
1740      A(L)=Crack(J)
1750      N(L)=Cycles(J)
1760    NEXT J
1770    C1=.5*(N(1)+N(7))
1780    C2=.5*(N(7)-N(1))
1790    Sx=0
1800    Sx2=0
1810    Sx3=0
1820    Sx4=0
1830    Sy=0
1840    Syx=0
1850    Syx2=0
1860    FOR J=1 TO 7
1870      Xx=(N(J)-C1)/C2
1880      Yy=A(J)
1890      Sx=Sx+Xx
1900      Sx2=Sx2+Xx^2
1910      Sx3=Sx3+Xx^3
1920      Sx4=Sx4+Xx^4
1930      Sy=Sy+Yy
1940      Syx=Syx+Xx*Yy
1950      Syx2=Syx2+Yy*Xx^2
1960    NEXT J
1970    Den=7.0*(Sx2*Sx4-Sx3^2)-Sx*(Sx*Sx4-Sx2*Sx3)+Sx2*(Sx*Sx3-Sx2^2)
1980    T2=Sy*(Sx2*Sx4-Sx3^2)-Syx*(Sx*Sx4-Sx2*Sx3)+Syx2*(Sx*Sx3-Sx2^2)
1990    Bb(1)=T2/Den
2000    T3=7.0*(Syx*Sx4-Syx2*Sx3)-Sx*(Sy*Sx4-Syx2*Sx2)+Sx2*(Sy*Sx3-Syx*Sx2)
2010    Bb(2)=T3/Den
2020    T4=7.0*(Sx2*Syx2-Sx3*Syx)-Sx*(Sx*Syx2-Sx3*Sy)+Sx2*(Sx*Syx-Sx2*Sy)
2030    Bb(3)=T4/Den
2040    Yb=Sy/7.0
2050    Rss=0
2060    Tss=0
2070    FOR J=1 TO 7
2080      Xx=(N(J)-C1)/C2
2090      Yhat=Bb(1)+Bb(2)*Xx+Bb(3)*Xx^2
2100      Rss=Rss+(A(J)-Yhat)^2
2110      Tss=Tss+(A(J)-Yb)^2
2120    NEXT J
2130    R2=1.0-Rss/Tss
2140    Dadn(I)=Bb(2)/C2+2.0*Bb(3)*(N(4)-C1)/C2^2
2150    Xx=(N(4)-C1)/C2
```

```
2160     Ar=Bb(1)+Bb(2)*Xx+Bb(3)*Xx^2
2170 !___________________________________
2180     T=Ar/11.7
2190     K11=INT((T+.005)*100)-1
2200     F11=FNF11(T)
2210     F22=FNF22(T)
2220     Factor1=1-(6*(F22/F11)*C12(K11)/(C22(K11)+12*Lb))
2230     A11=C12(K11)^2/(12*Lb+C22(K11))
2240     A22=6*C12(K11)*(Lw-Lb)/(12*Lb+C22(K11))
2250     Factor2=(1+(C11(K11)-A11+A22)/Lw)
2260     Ft=SQR(PI*Ar/1000)*F11*Factor1/Factor2  !K using modulus
2270  ! Ft=SQR(PI*Ar/1000)*F11*Factor1          !K using stress range
2280     Delk(I)=Deltap*Ft
2290     PRINT Delk(I),Dadn(I)
2300     IF T>=.60 THEN Delk(I)=100000
2310  NEXT I
2320 BEEP 3000,1
2330  ALPHA OFF
2340  !
2350  FOR I=0 TO Npts
2360     A1=LGT(Delk(I)/10)*100-50
2370     A2=(LGT(ABS(Dadn(I))/1000)+9)*80/6-40
2380     MOVE A1,A2
2390     IMOVE -1.3344481605,-1
2400     LABEL "."
2410  NEXT I
2420 PENUP
2430 MOVE -50,-40
2440 PAUSE
2450 SUBEND
2460!_____________________________________________________________________
2470  DEF FNF11(I)
2480  RAD
2490   IF I=0 THEN
2500     F11=2/SQR(PI)
2510     GOTO 2560
2520   END IF
2530   A=SQR(2*TAN(PI*I/2)/(PI*I))
2540   B=(.752+2.02*I+.37*(1-SIN(PI*I/2))^3)/COS(PI*I/2)
2550   F11=A*B
2560   RETURN F11
2570  FNEND
2580!_____________________________________________________________________
2590  DEF FNF22(I)
2600  RAD
2610   IF I=0 THEN
2620     F22=2/SQR(PI)
2630     GOTO 2680
2640   END IF
2650   A=SQR(2*TAN(PI*I/2)/(PI*I))
2660   B=(.923+.199*(1-SIN(PI*I/2))^4)/COS(PI*I/2)
2670   F22=A*B
2680   RETURN F22
2690  FNEND
```

```
2700!----------------------------------------------------------------------------
2710  SUB Read_const(C11(*),C12(*),C22(*))
2720  DATA .39655E-03,.1594E-02,.36060E-02,.64504E-02,.10149E-01,.14728E-01,.202
17E-01,.26652E-01,.34072E-01,.42522E-01
2730  DATA .52051E-01,.62715E-01,.74576E-01,.87699E-01,.10216,.11804,.13542,.154
41,.17510,.19761
2740  DATA .22207,.24861,.27738,.30854,.34225,.37872,.41813,.46073,.50674,.55643
2750  DATA .61009,.66802,.73057,.79811,.87105,.94981,1.0349,1.1268,1.2262,1.3335
2760  DATA 1.4497,1.5753,1.7114,1.8586,2.0182,2.1912,2.3789,2.5827,2.8040,3.0447
2770  DATA 3.3065,3.5917,3.9026,4.2418,4.6124,5.0177,5.4615,5.9481,6.4823,7.0697
2780  DATA 7.7166,8.4301,9.2186,10.091,11.060,12.136,13.336,14.676,16.176,17.861
2790  DATA 19.759,21.905,24.337,27.106,30.271,33.905,38.097,42.958,48.629,55.284
2800  DATA 63.151,72.522,83.782,97.441,114.19,135.00,161.19,194.73,238.50,296.92
2810  DATA 377.05,490.68,658.66,920.72,1359.9,2174.0,3930.5,8822.6,31809.,31809.
2820  READ C11(*)
2830!------------
2840  DATA .23631E-02,.9402E-02,.21068E-01,.37333E-01,.58192E-01,.83658E-01,.113
77,.14857,.18815,.23259
2850  DATA .28201,.33655,.39637,.46164,.53256,.60936,.69228,.78161,.87763,.98069
2860  DATA 1.0911,1.2093,1.3358,1.4708,1.6151,1.7690,1.9332,2.1083,2.2950,2.4940
2870  DATA 2.7062,2.9324,3.1735,3.4305,3.7047,3.9970,4.3090,4.6419,4.9973,5.3769
2880  DATA 5.7826,6.2163,6.6802,7.1768,7.7085,8.2785,8.8898,9.5459,10.251,11.008
2890  DATA 11.824,12.702,13.650,14.672,15.777,16.974,18.269,19.676,21.204,22.867
2900  DATA 24.680,26.659,28.826,31.200,33.808,36.680,39.849,43.356,47.246,51.573
2910  DATA 56.403,61.811,67.889,74.744,82.510,91.347,101.45,113.07,126.50,142.13
2920  DATA 160.45,182.09,207.87,238.89,276.62,323.09,381.14,454.87,550.30,676.68
2930  DATA 848.67,1090.7,1445.7,1995.3,2909.4,4591.0,8192.0,18147.,64577.,64577.
2940  READ C12(*)
2950!------------
2960  DATA .14082E-01,.55459E-01,.12310,.21611,.33375,.47539,.64052,.82877,1.039
9,1.2736
2970  DATA 1.5299,1.8089,2.1105,2.4352,2.7830,3.1545,3.5500,3.9703,4.4159,4.8876
2980  DATA 5.3863,5.9128,6.4683,7.0540,7.6710,8.3207,9.0047,9.7247,10.482,11.280
2990  DATA 12.118,13.001,13.931,14.909,15.939,17.025,18.168,19.374,20.646,21.988
3000  DATA 23.405,24.901,26.484,28.158,29.930,31.807,33.798,35.911,38.155,40.541
3010  DATA 43.081,45.787,48.673,51.756,55.052,58.582,62.366,66.430,70.801,75.509
3020  DATA 80.590,86.083,92.034,98.493,105.52,113.18,121.55,130.73,140.81,151.93
3030  DATA 164.22,177.85,193.03,210.01,229.06,250.55,274.91,302.67,334.49,371.20
3040  DATA 413.86,463.83,522.87,593.31,678.29,782.09,910.72,1072.8,1280.9,1554.3
3050  DATA 1923.4,2438.8,3189.1,4341.7,6244.1,9717.6,17100.,37358.,131140,131140
3060  READ C22(*)
3070  SUBEND
3080!----------------------------------------------------------------------------
```

```
3090  SUB Stress_cycle(Counter,Stress_max(*),Stress_min(*),Cycles(*),Crack(*),Z1
$,Z2$,Z3$,Z4$)
3100  ALLOCATE A(20),B(20)
3110  FOR I=0 TO 20
3120    A(I)=25*I-50
3130    B(I)=3*I-40
3140  NEXT I
3150  DEG
3160  GINIT
3170  GCLEAR
3180  ALPHA OFF
3190  GRAPHICS ON
3200  FRAME
3210  WINDOW -66.722,66.722,-50,50
3220  MOVE -50,-40
3230  FOR I=0 TO 3
3240    FOR J=1 TO 9
3250      DRAW 25*LGT(J*10^I)-50,-40
3260      IF J=1 THEN
3270        IDRAW 0,1
3280        IDRAW 0,-1
3290      ELSE
3300        IDRAW 0,.5
3310        IDRAW 0,-.5
3320      END IF
3330    NEXT J
3340  NEXT I
3350  C=.5
3360  D=1.0
3370  FOR I=0 TO 20
3380    DRAW 50,B(I)
3390    IF 2*INT(I/2)=I THEN
3400      IDRAW -D,0
3410      IDRAW D,0
3420    ELSE
3430      IDRAW -C,0
3440      IDRAW C,0
3450    END IF
3460  NEXT I
3470  FOR I=3 TO 0 STEP -1
3480    FOR J=9 TO 1 STEP -1
3490      DRAW 25*LGT(J*10^I)-50,20
3500      IF J=1 THEN
3510        IDRAW 0,-1
3520        IDRAW 0,1
3530      ELSE
3540        IDRAW 0,-.5
3550        IDRAW 0,.5
3560      END IF
3570    NEXT J
3580  NEXT I
3590  FOR I=20 TO 0 STEP -1
3600    DRAW -50,B(I)
3610    IF 2*INT(I/2)=I THEN
3620      IDRAW D,0
```

```
3630       IDRAW -D,0
3640     ELSE
3650       IDRAW C,0
3660       IDRAW -C,0
3670     END IF
3680   NEXT I
3690   MOVE -50,-10
3700   FOR I=0 TO 3
3710     FOR J=1 TO 9
3720       DRAW 25*LGT(J*10^I)-50,-10
3730       IF J=1 THEN
3740         IDRAW 0,1
3750         IDRAW 0,-2
3760         IDRAW 0,1
3770       ELSE
3780         IDRAW 0,.5
3790         IDRAW 0,-1
3800         IDRAW 0,.5
3810       END IF
3820     NEXT J
3830   NEXT I
3840   MOVE 50,20
3850   DRAW 50,40
3860   DRAW -50,40
3870   DRAW -50,20
3880   CSIZE 3,.6
3890   MOVE -50,-44
3900   LABEL VAL$(1)
3910   FOR I=1 TO 4
3920     MOVE A(I)-3,-44
3930     LABEL VAL$(10^I)
3940   NEXT I
3950   FOR I=0 TO 8 STEP 2
3960     MOVE -56,B(I)-1
3970     LABEL USING "Z.D";I/10
3980     MOVE 51,B(I)-1
3990     LABEL USING "Z.D";I/10
4000   NEXT I
4010   FOR I=10 TO 20 STEP 2
4020     MOVE -56,B(I)-1
4030     LABEL USING "DDD";500*(I-10)/10
4040     MOVE 51,B(I)-1
4050     IF I=10 THEN LABEL USING "D";500*(I-10)/10
4060     IF I=12 THEN LABEL USING "DDD";500*(I-10)/10
4070     IF I>12 THEN LABEL USING "DDD";500*(I-10)/10
4080   NEXT I
4090   CSIZE 4,.6
4100   MOVE -5,-48
4110   LABEL "CYCLE"
4120   CSIZE 3,.6
4130   LDIR 90
4140   MOVE -58,-27
4150   LABEL "a/W"
4160   MOVE -58,-3
```

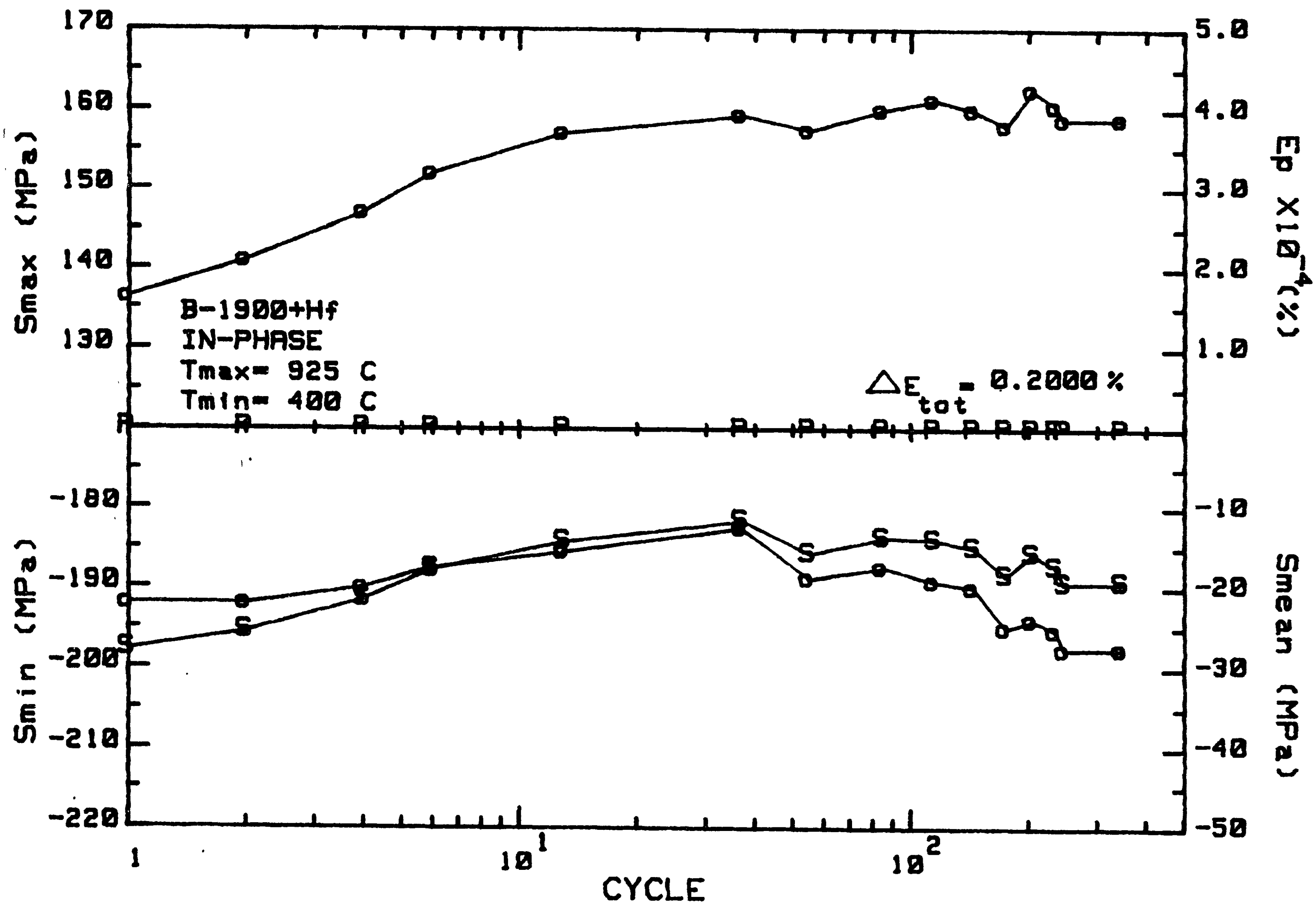
```
4170  LABEL "Smax (MPa)"
4180  LDIR -90
4190  MOVE 59,13
4200  LABEL "Smin (MPa)"
4210  MOVE 59,-22
4220  LABEL "a/W"
4230  LDIR 0
4240  CSIZE 3.5,.6
4250  MOVE -45,34
4260  LABEL Z1$
4270  MOVE -45,31
4280  LABEL "STRAIN CONTROL"
4290  MOVE -45,28
4300  LABEL Z2$
4310  MOVE -45,26
4320  IDRAW 3,0
4330  IDRAW -1.5,1.5
4340  IDRAW -1.5,-1.5
4350  MOVE -41,25
4360  LABEL "T= "&Z3$&" C"
4370  MOVE -45,23
4380  IDRAW 3,0
4390  IDRAW -1.5,1.5
4400  IDRAW -1.5,-1.5
4410  MOVE -41,22
4420  LABEL "E= "&Z4$&" %"
4430!___________________________________
4440  FOR I=0 TO Counter
4450    X=25*LGT(Cycles(I))-50
4460    Y=30*(Crack(I)/11.7)-40
4470    MOVE X,Y
4480    IMOVE -1,-2
4490    LABEL "."
4500  NEXT I
4510!___________________________________
4520  MOVE -50,0
4530  Previous1=Stress_max(0)*21.6078*5
4540  Previous2=Stress_max(0)*21.6078*5
4550  FOR I=0 TO Counter
4560    X=25*LGT(Cycles(I))-50
4570    Load=Stress_max(I)*21.6078*5
4580    IF Load>Previous1+75 OR Load<Previous1-75 THEN Alpha1
4590    IF Load>Previous2+75 OR Load<Previous2-75 THEN Alpha1
4600    GOTO 4640
4610 Alpha1:!
4620      Load=Previous1
4630      Stress_max(I)=Load/(21.6078*5)
4640      !___________________________________
4650    Y=30*Load/500-10
4660    MOVE X-1,Y-2
4670    LABEL "+"
4680    Previous2=Previous1
4690    Previous1=Load
4700  NEXT I
```

```
4710!________________________________
4720  MOVE -50,0
4730  Previous1=-Stress_min(0)*21.6078*5
4740  Previous2=-Stress_min(0)*21.6078*5
4750  FOR I=0 TO Counter
4760    X=25*LGT(Cycles(I))-50
4770    Load=-Stress_min(I)*21.6078*5
4780    IF Load>Previous1+50 OR Load<Previous1-50 THEN Alpha
4790    IF Load>Previous2+50 OR Load<Previous2-50 THEN Alpha
4800    GOTO 4840
4810 Alpha:!
4820          Load=Previous1
4830          Stress_min(I)=-Load/(21.6078*5)
4840          !____________________________________________
4850    Y=30*Load/500-10
4860    MOVE X-1,Y-1
4870    LABEL "-"
4880    Previous2=Previous1
4890    Previous1=Load
4900  NEXT I
4910  PAUSE
4920  SUBEND
```

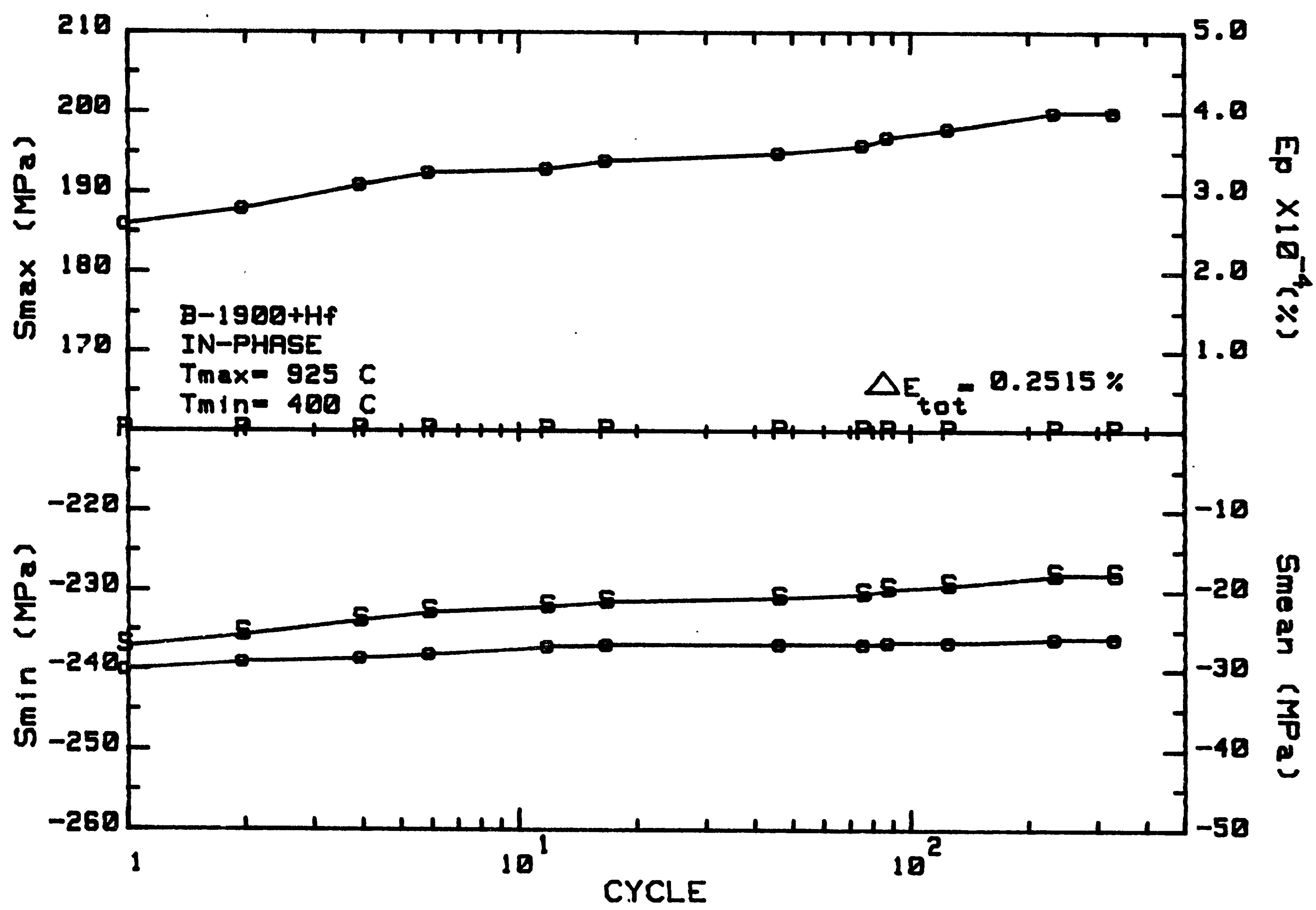
Appendix IV

In this appendix the cyclic stress-strain curves obtained under TMF conditions for B-1900+Hf are presented. The crack growth rates as a function of the strain intensity factor (ΔK_{ε}) are also presented for $\Delta\varepsilon$ = 0.25% and 0.50%.

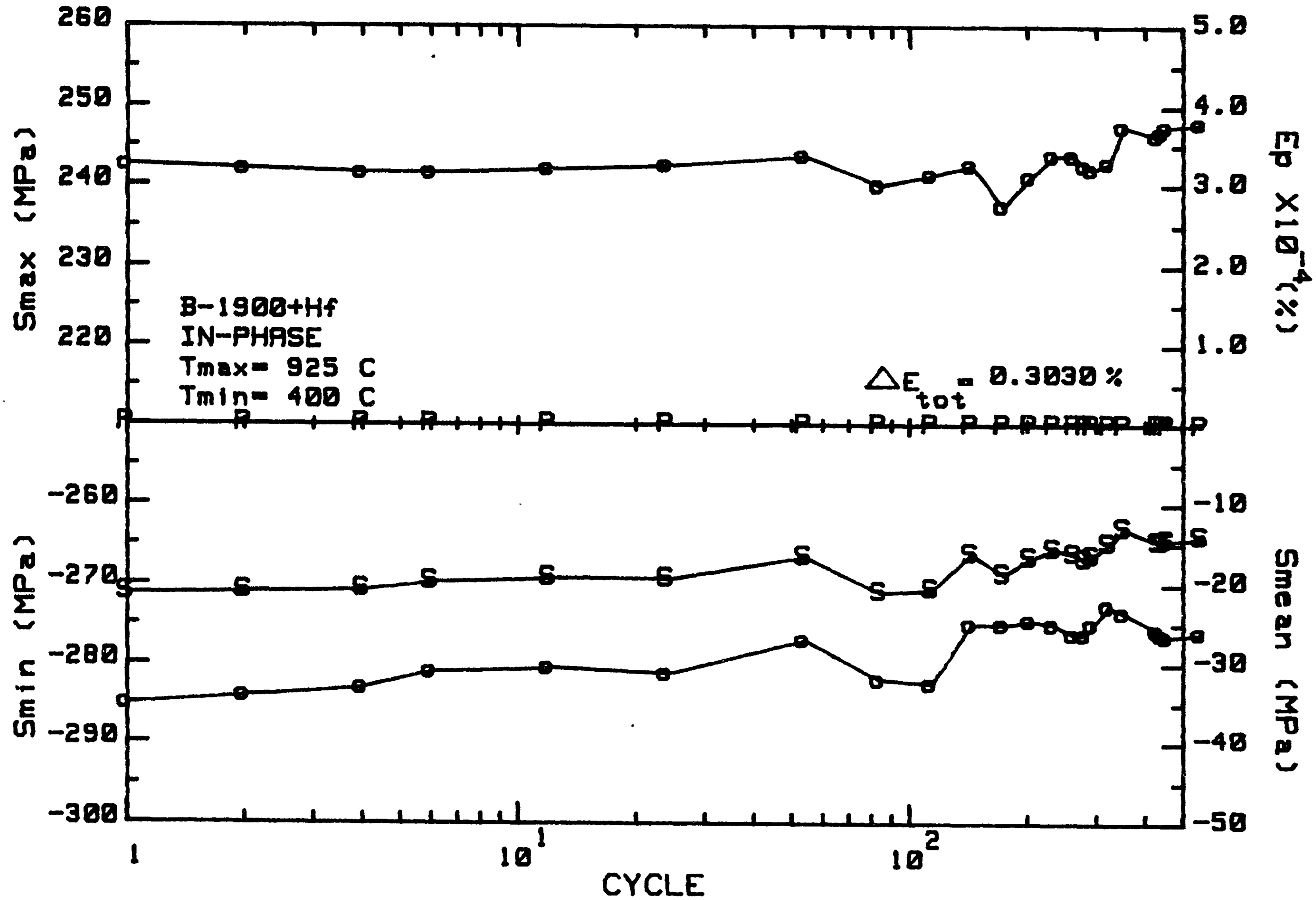
Each cyclic hardening/softening curves show the maximum stress, minimum stress, plastic strain range and mean stress as a function of the applied number of cycles. All data points on the FCG curves are shown as an indication of the scatter and resolution of the testing procedure.



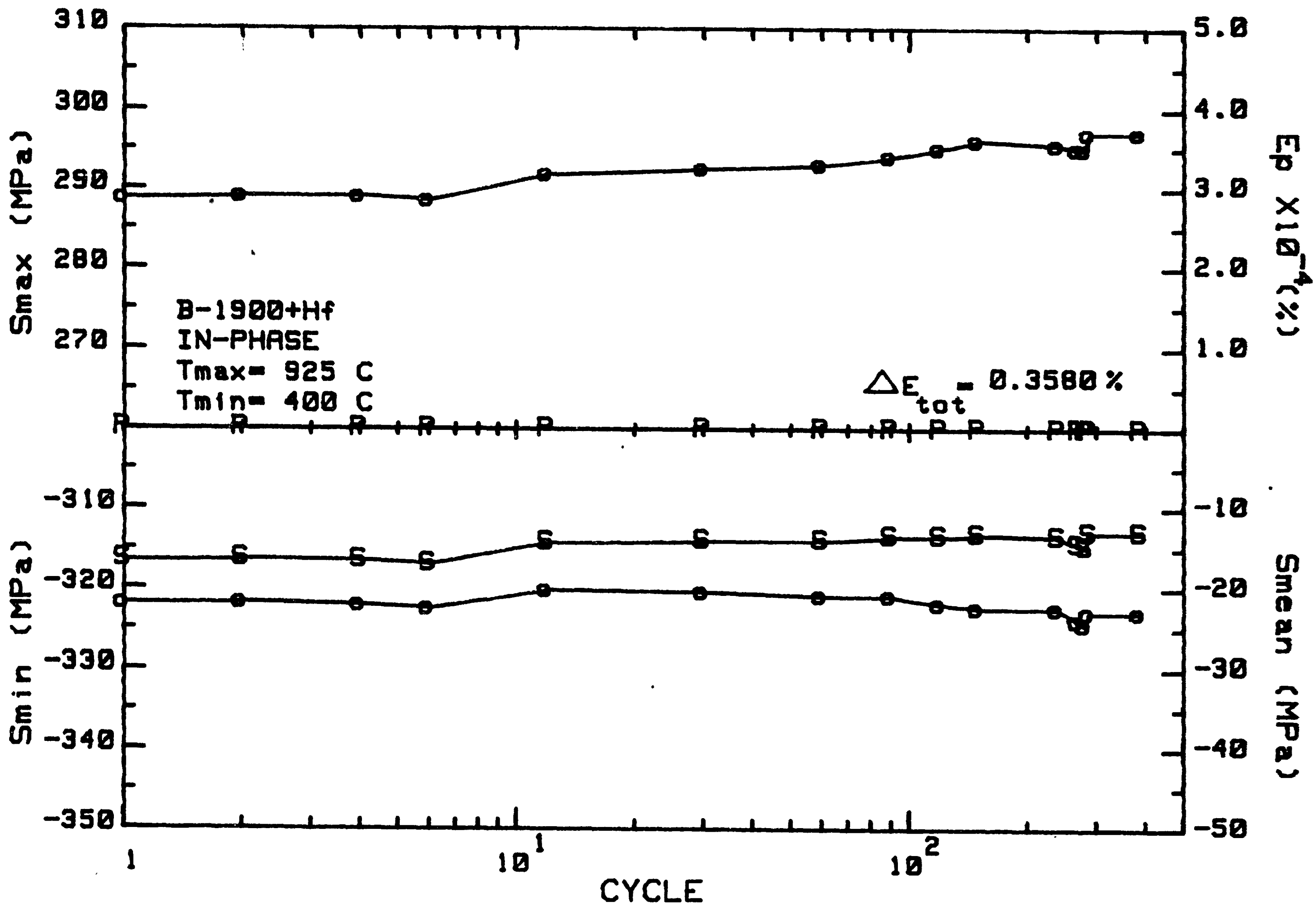
B-1900+Hf
IN-PHASE
Tmax= 925 C
Tmin= 400 C
ΔE_{tot} = 0.2000 %
Smax (MPa)
Ep X10^{-4}(%)
Smin (MPa)
Smean (MPa)
CYCLE



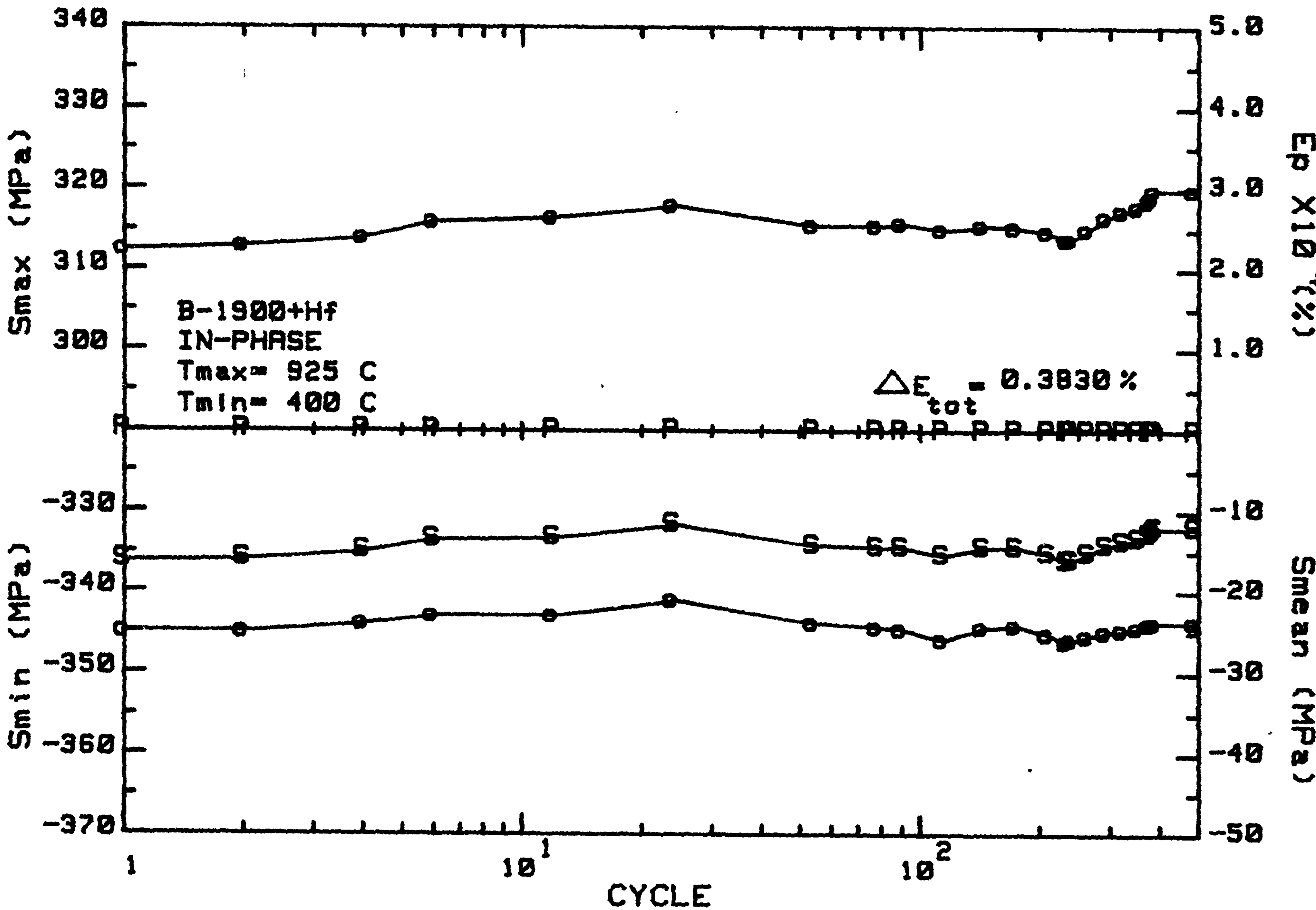
Smax (MPa)
210
200
190
180
170
Ep X10^-4 (%)
5.0
4.0
3.0
2.0
1.0
B-1900+Hf
IN-PHASE
Tmax= 925 C
Tmin= 400 C
ΔE_tot = 0.2515 %
Smin (MPa)
-220
-230
-240
-250
-260
Smean (MPa)
-10
-20
-30
-40
-50
1
10^1
10^2
CYCLE

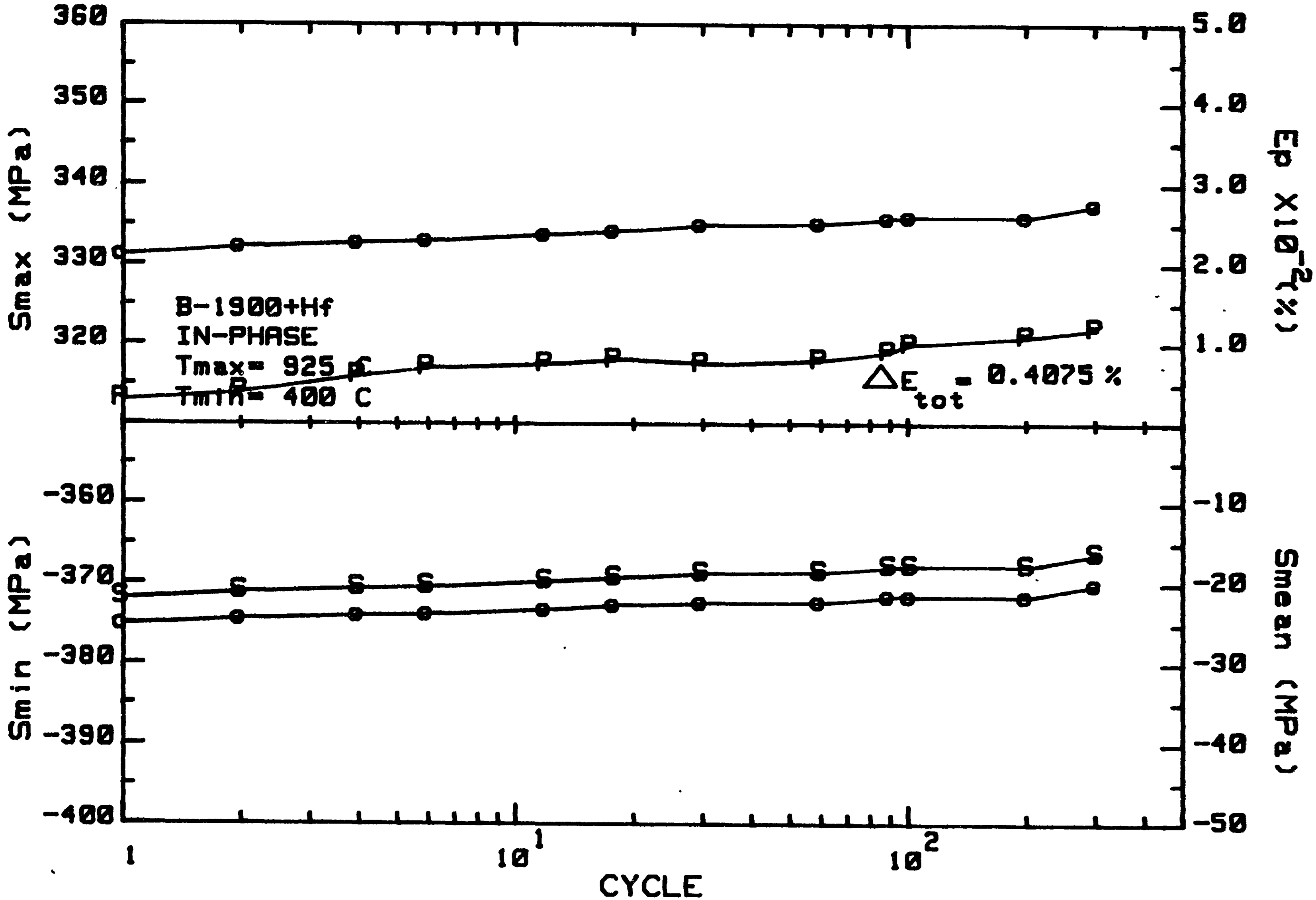


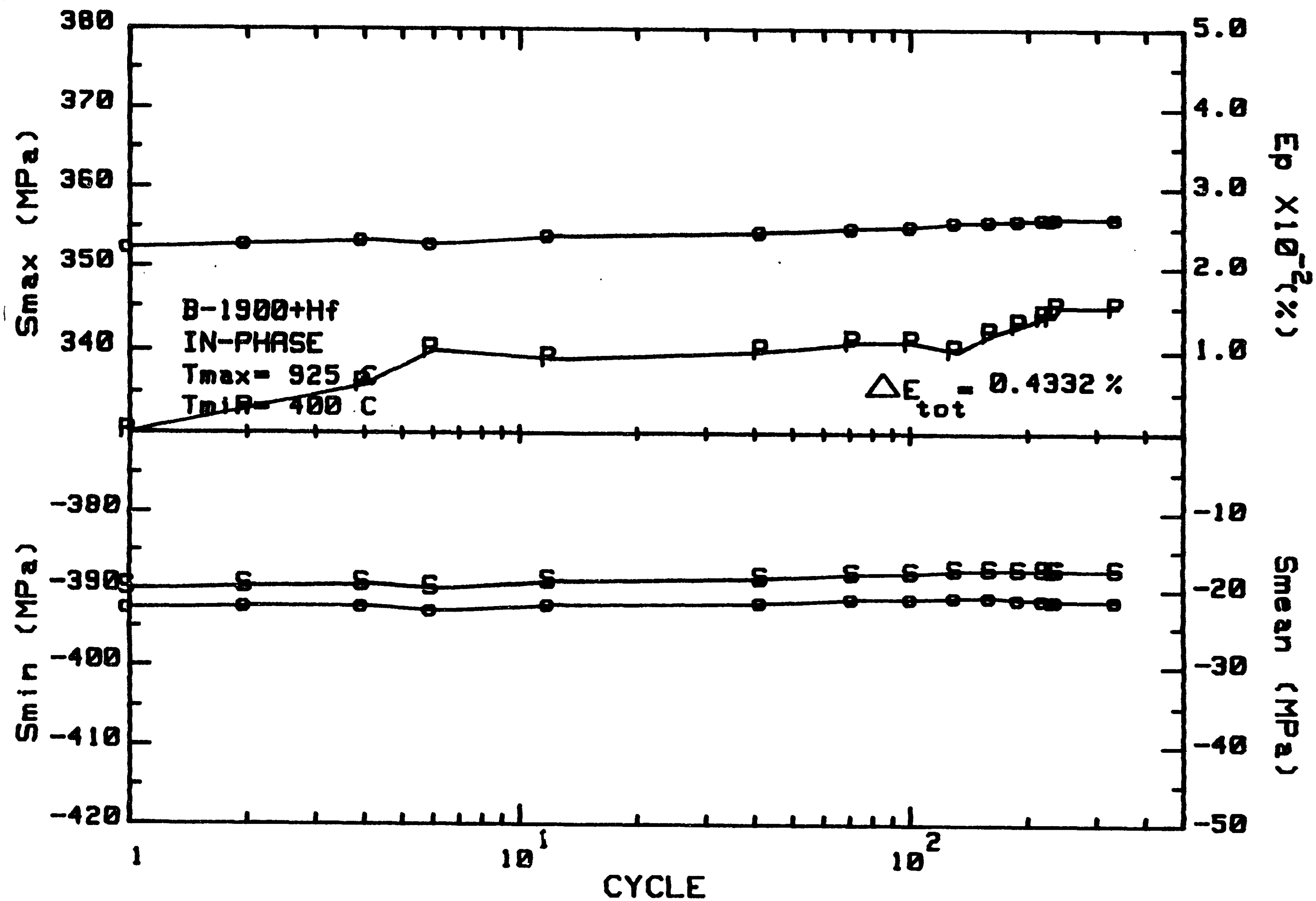
B-1900+Hf
IN-PHASE
Tmax= 925 C
Tmin= 400 C
ΔE_{tot} = 0.3030 %
Smax (MPa)
Ep X10^{-4}(%)
Smin (MPa)
Smean (MPa)
CYCLE



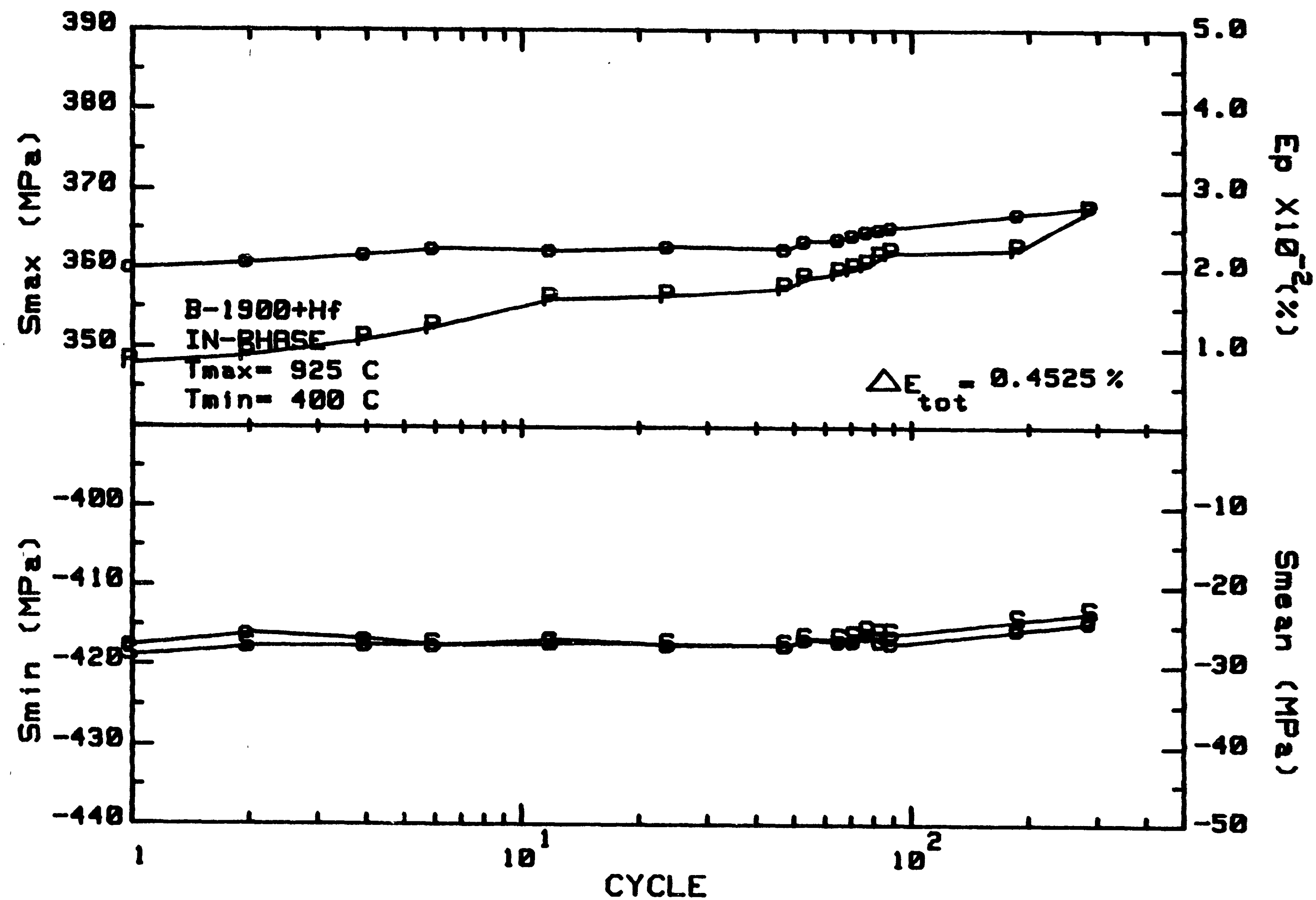
Smax (MPa)
Ep X10^-4 (%)
Smin (MPa)
Smean (MPa)
B-1900+Hf
IN-PHASE
Tmax= 925 C
Tmin= 400 C
ΔE_tot = 0.3580 %
CYCLE


Smax (MPa)
340
330
320
310
300
Smin (MPa)
-330
-340
-350
-360
-370
Ep X10^-4 (%)
5.0
4.0
3.0
2.0
1.0
Smean (MPa)
-10
-20
-30
-40
-50
B-1900+Hf
IN-PHASE
Tmax= 925 C
Tmin= 400 C
ΔE_tot = 0.3830 %
1
10^1
10^2
CYCLE

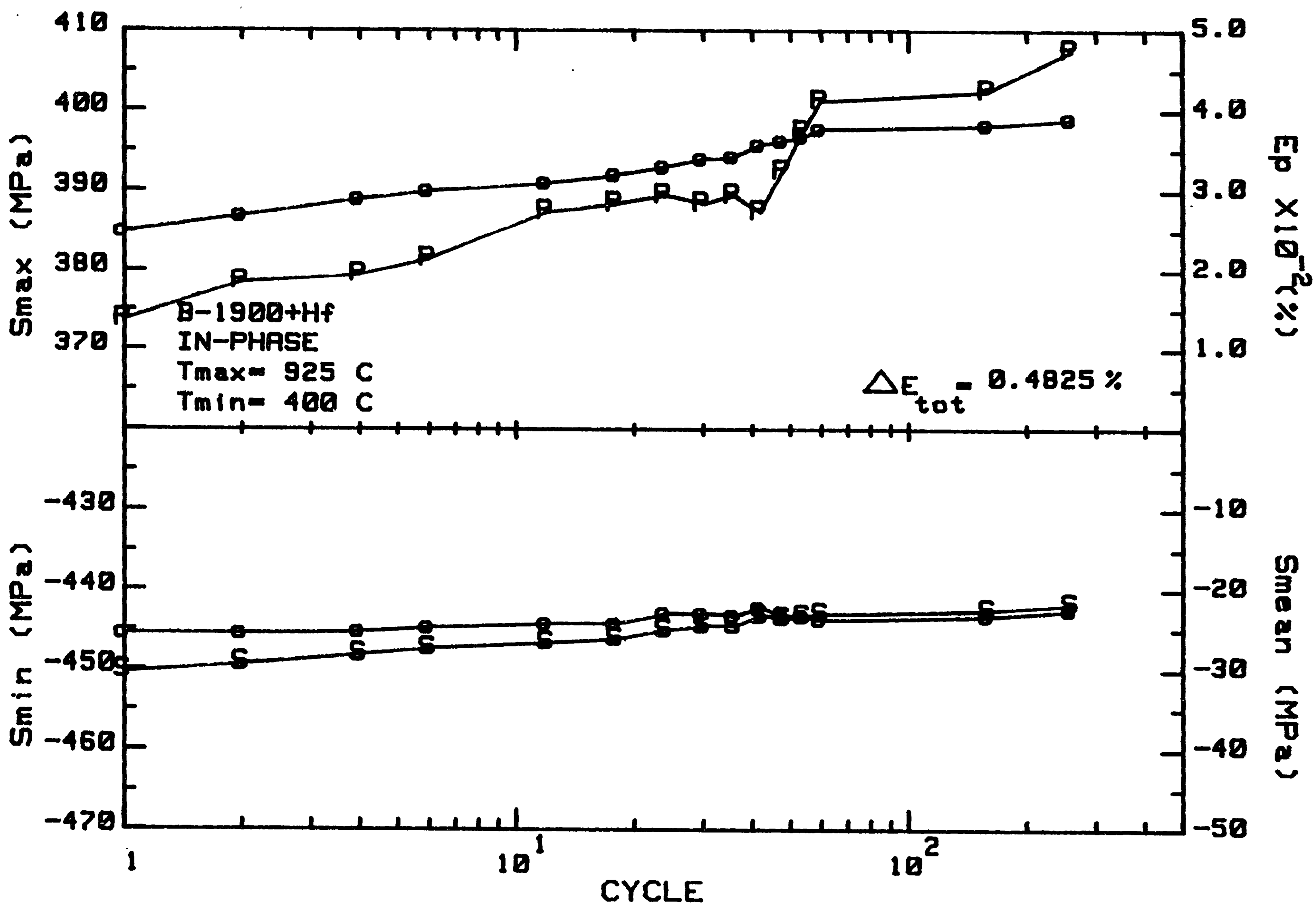

B-1900+Hf
IN-PHASE
Tmax= 925 C
Tmin= 400 C
ΔE_{tot} = 0.4075 %
Smax (MPa)
Smin (MPa)
Ep X10^{-2}(%)
Smean (MPa)
CYCLE
360
350
340
330
320
-360
-370
-380
-390
-400
5.0
4.0
3.0
2.0
1.0
-10
-20
-30
-40
-50
1
10^1
10^2

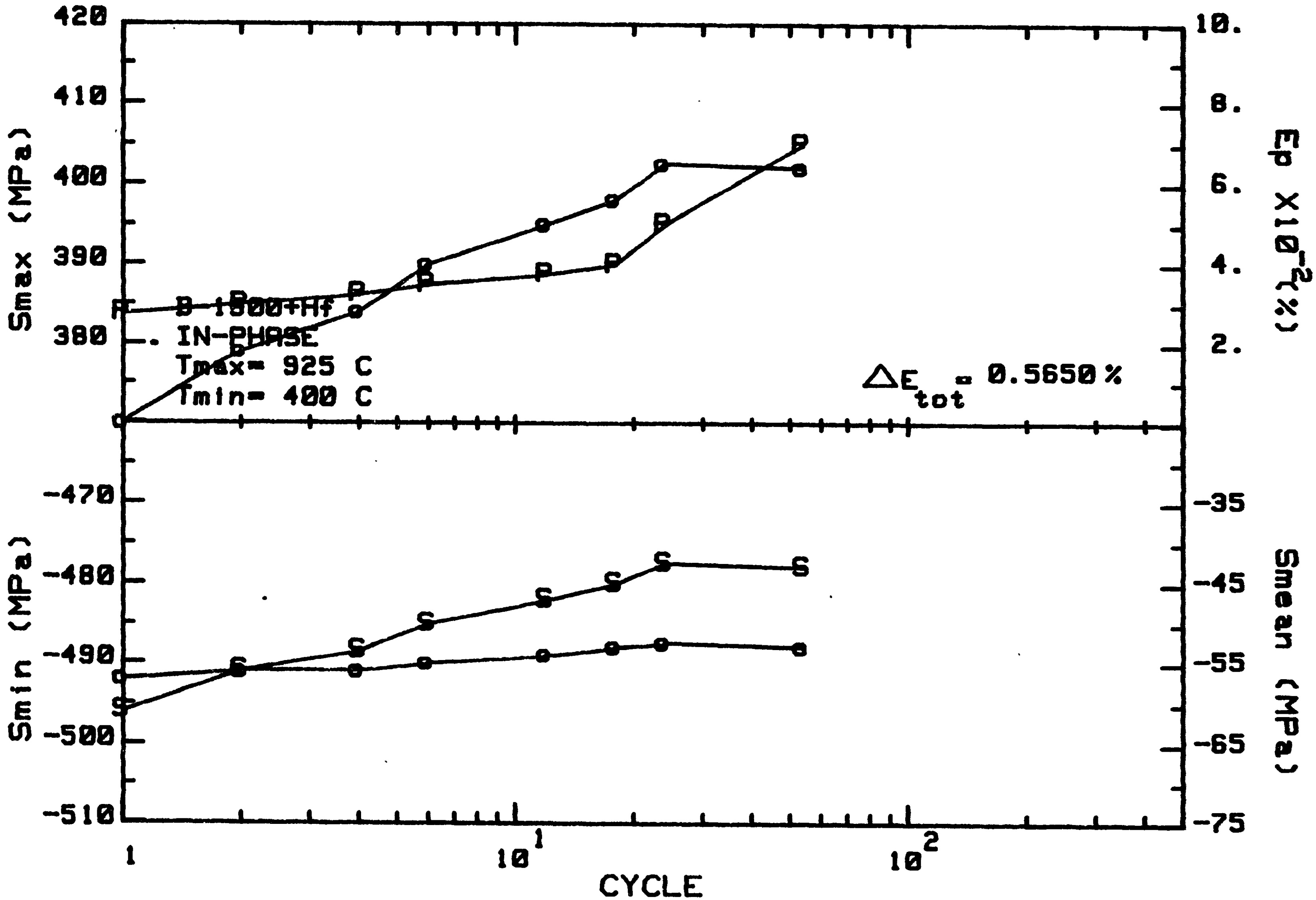


B-1900+Hf
IN-PHASE
Tmax= 925 C
Tmin= 400 C
ΔE_{tot} = 0.4332 %
Smax (MPa)
Smin (MPa)
Ep X10^{-2}(%)
Smean (MPa)
CYCLE
380
370
360
350
340
-380
-390
-400
-410
-420
5.0
4.0
3.0
2.0
1.0
-10
-20
-30
-40
-50
1
10^{1}
10^{2}

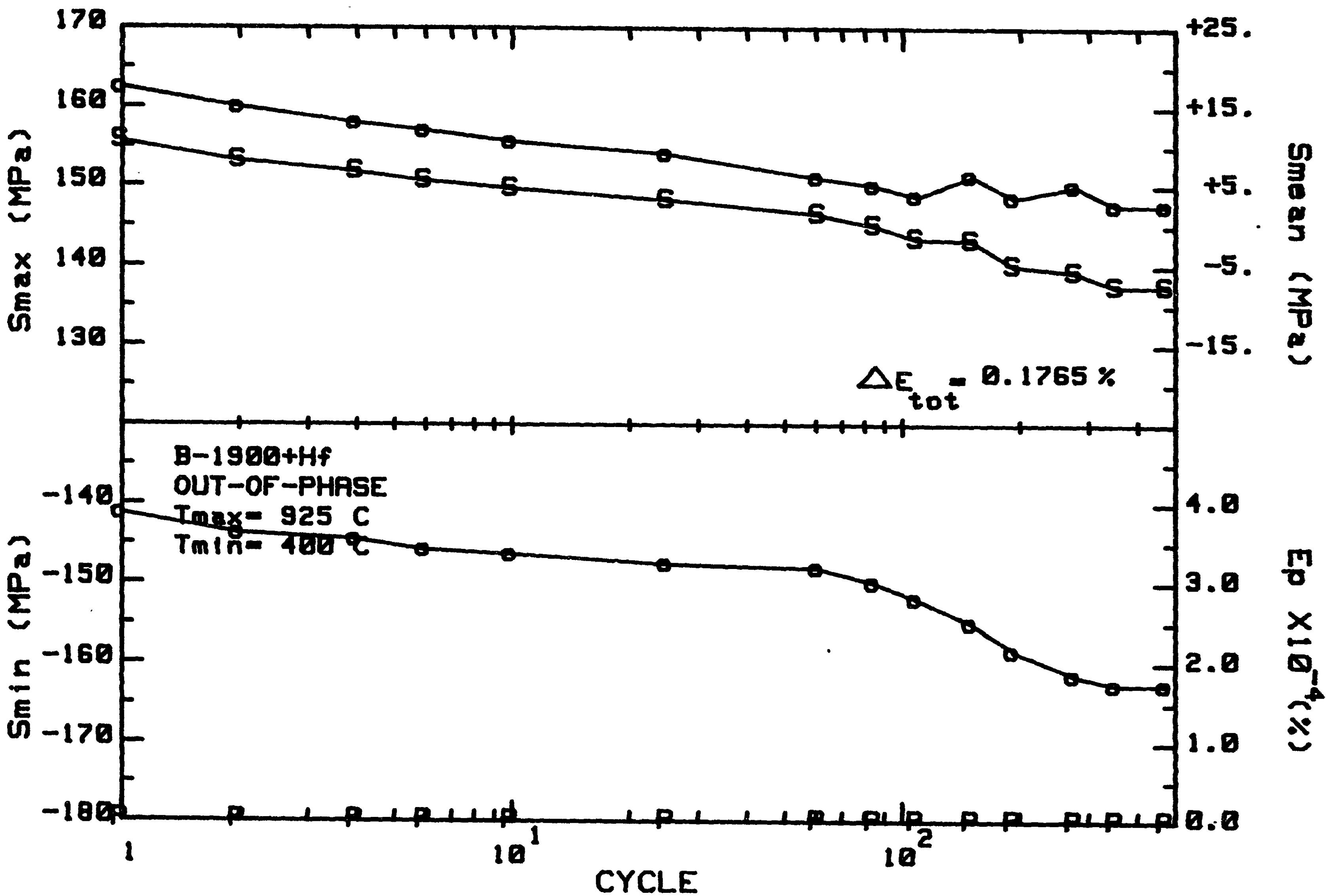


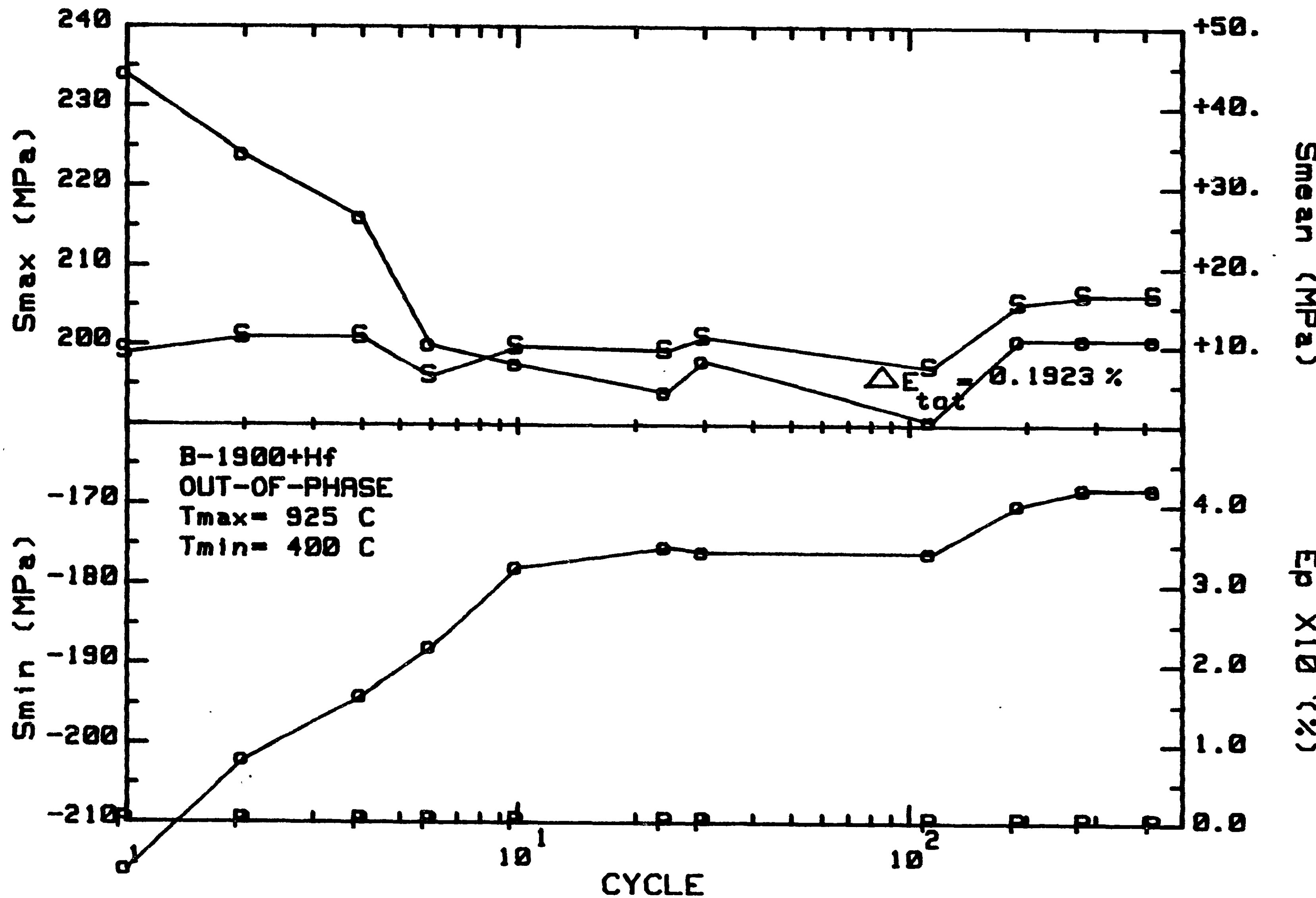
B-1900+Hf
IN-PHASE
Tmax= 925 C
Tmin= 400 C
ΔE_{tot} = 0.4525 %
Smax (MPa)
Smin (MPa)
Ep X10^{-2}(%)
Smean (MPa)
CYCLE
390
380
370
360
350
-400
-410
-420
-430
-440
5.0
4.0
3.0
2.0
1.0
-10
-20
-30
-40
-50
1
10^1
10^2


B-1900+Hf
IN-PHASE
Tmax= 925 C
Tmin= 400 C
ΔE_{tot} = 0.4825 %
Smax (MPa)
Ep X10^{-2}(%)
Smin (MPa)
Smean (MPa)
CYCLE

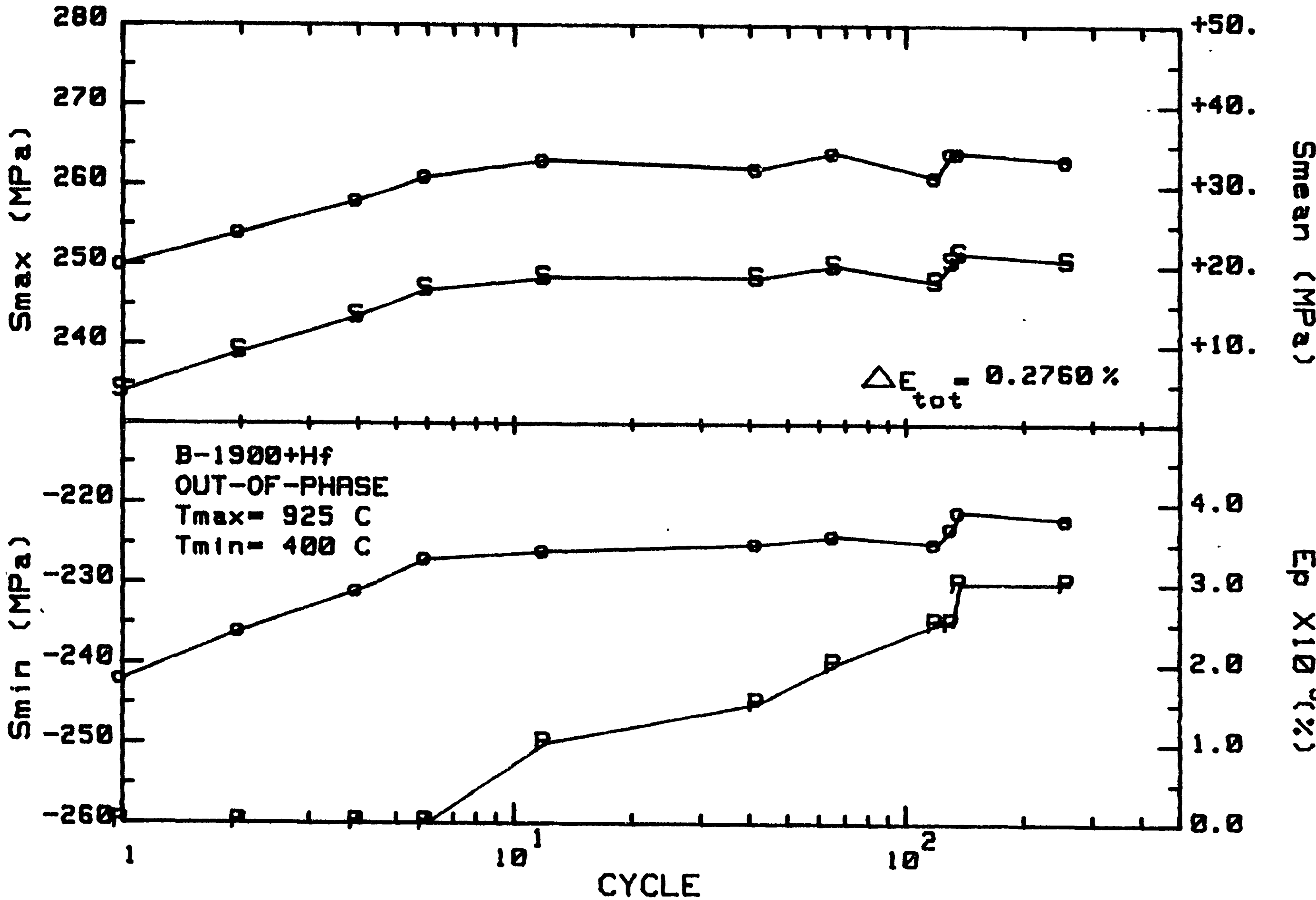


Smax (MPa)
Ep X10^{-2}(%)
Smin (MPa)
Smean (MPa)
B-1900+Hf
IN-PHASE
Tmax= 925 C
Tmin= 400 C
ΔE_{tot} = 0.5650 %
CYCLE

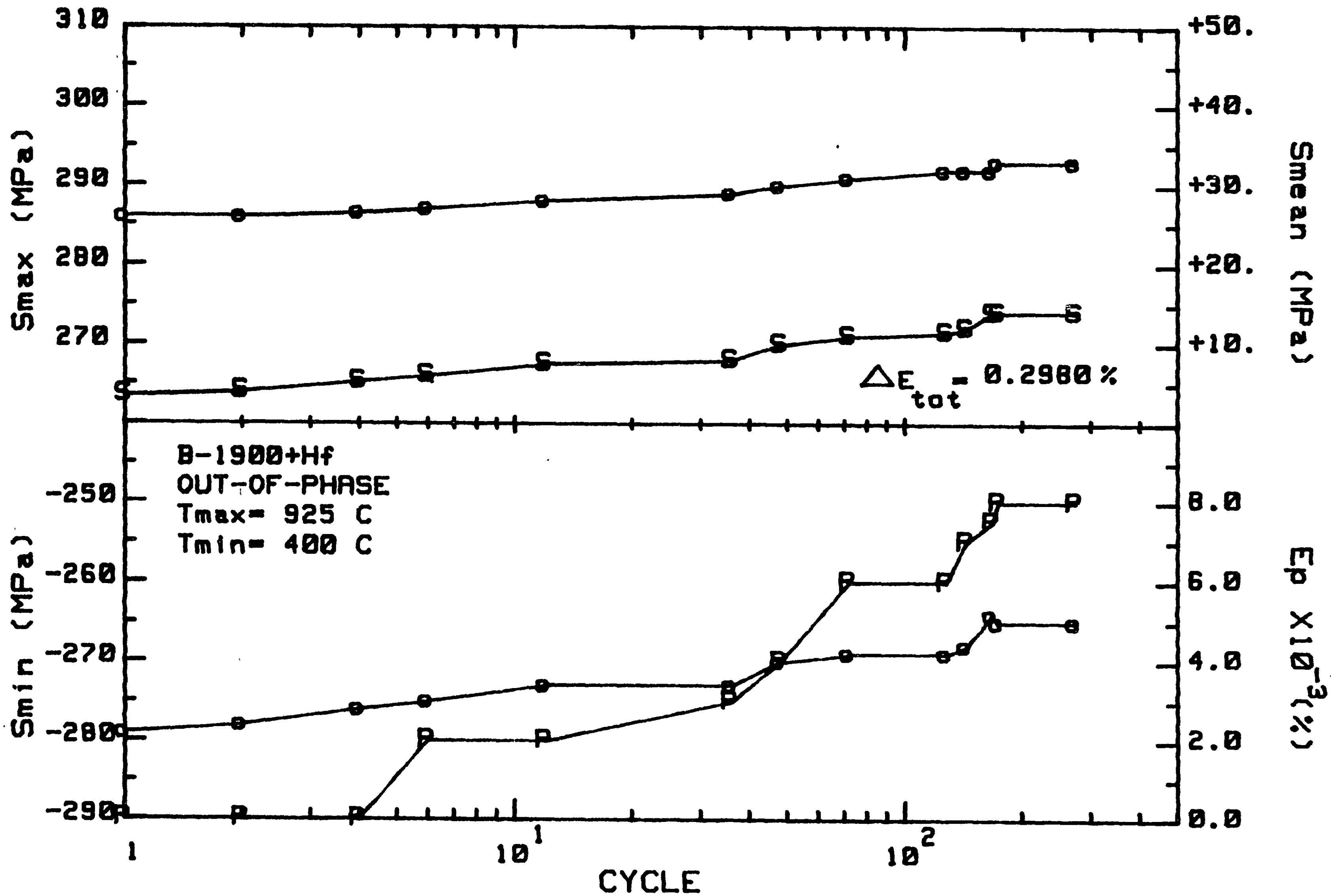

Smax (MPa)
Smin (MPa)
Smean (MPa)
Ep X10^-4 (%)
ΔE_tot = 0.1765 %
B-1900+Hf
OUT-OF-PHASE
Tmax= 925 C
Tmin= 400 C
CYCLE
170
160
150
140
130
-140
-150
-160
-170
-180
+25.
+15.
+5.
-5.
-15.
4.0
3.0
2.0
1.0
0.0
1
10^1
10^2



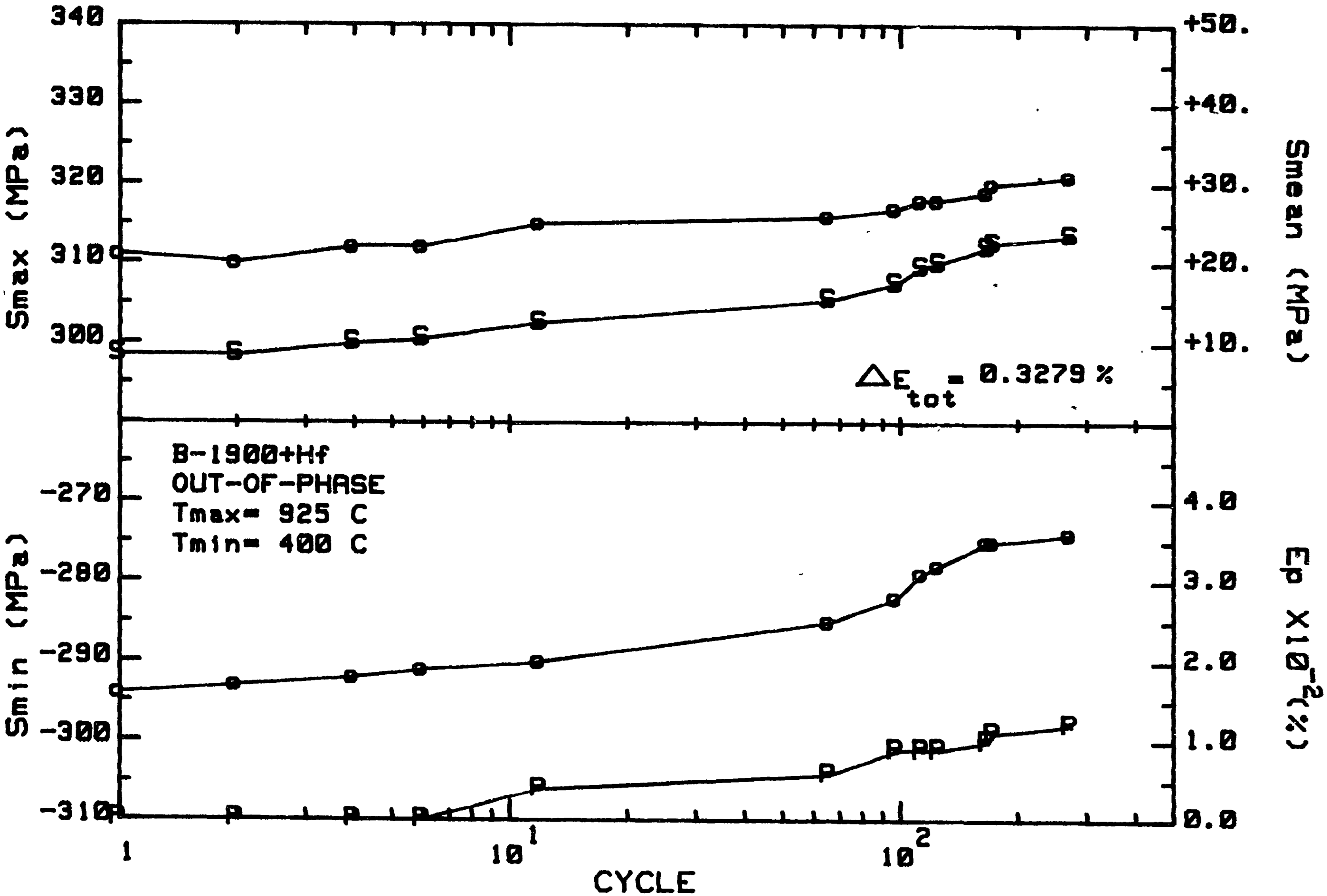
Smax (MPa)
240
230
220
210
200
Smean (MPa)
+50.
+40.
+30.
+20.
+10.
ΔE tot = 0.1923 %
B-1900+Hf
OUT-OF-PHASE
Tmax= 925 C
Tmin= 400 C
Smin (MPa)
-170
-180
-190
-200
-210
Ep X10^-4 (%)
4.0
3.0
2.0
1.0
0.0
1
10^1
10^2
CYCLE



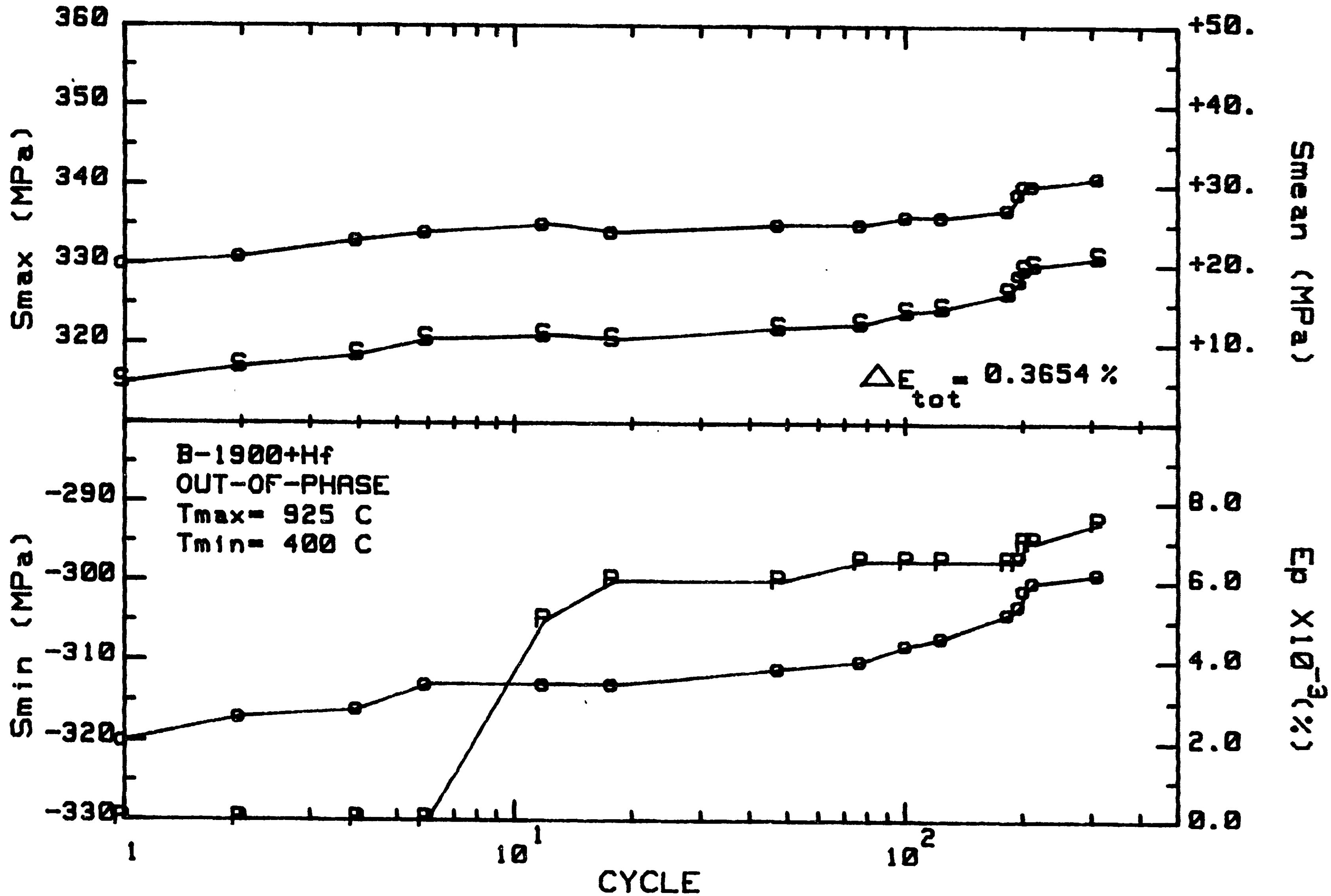
Smax (MPa)
280
270
260
250
240
Smean (MPa)
+50.
+40.
+30.
+20.
+10.
ΔE_{tot} = 0.2760 %
B-1900+Hf
OUT-OF-PHASE
Tmax= 925 C
Tmin= 400 C
Smin (MPa)
-220
-230
-240
-250
-260
Ep X10^{-3}(%)
4.0
3.0
2.0
1.0
0.0
1
10^{1}
10^{2}
CYCLE



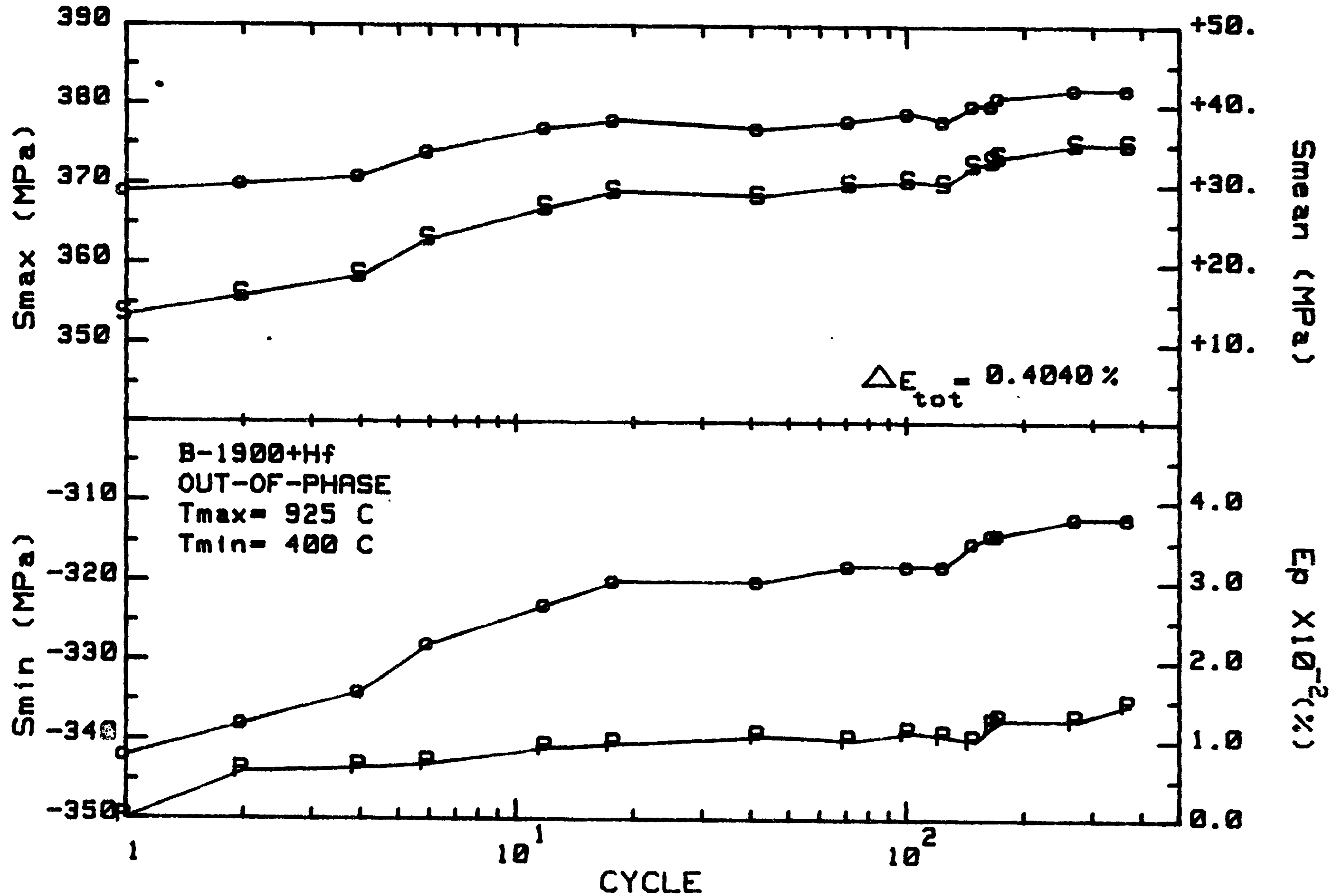
Smax (MPa)
310
300
290
280
270
Smean (MPa)
+50.
+40.
+30.
+20.
+10.
ΔE_{tot} = 0.2980 %
B-1900+Hf
OUT-OF-PHASE
Tmax= 925 C
Tmin= 400 C
Smin (MPa)
-250
-260
-270
-280
-290
Ep X10^{-3}(%)
8.0
6.0
4.0
2.0
0.0
1
10^1
10^2
CYCLE



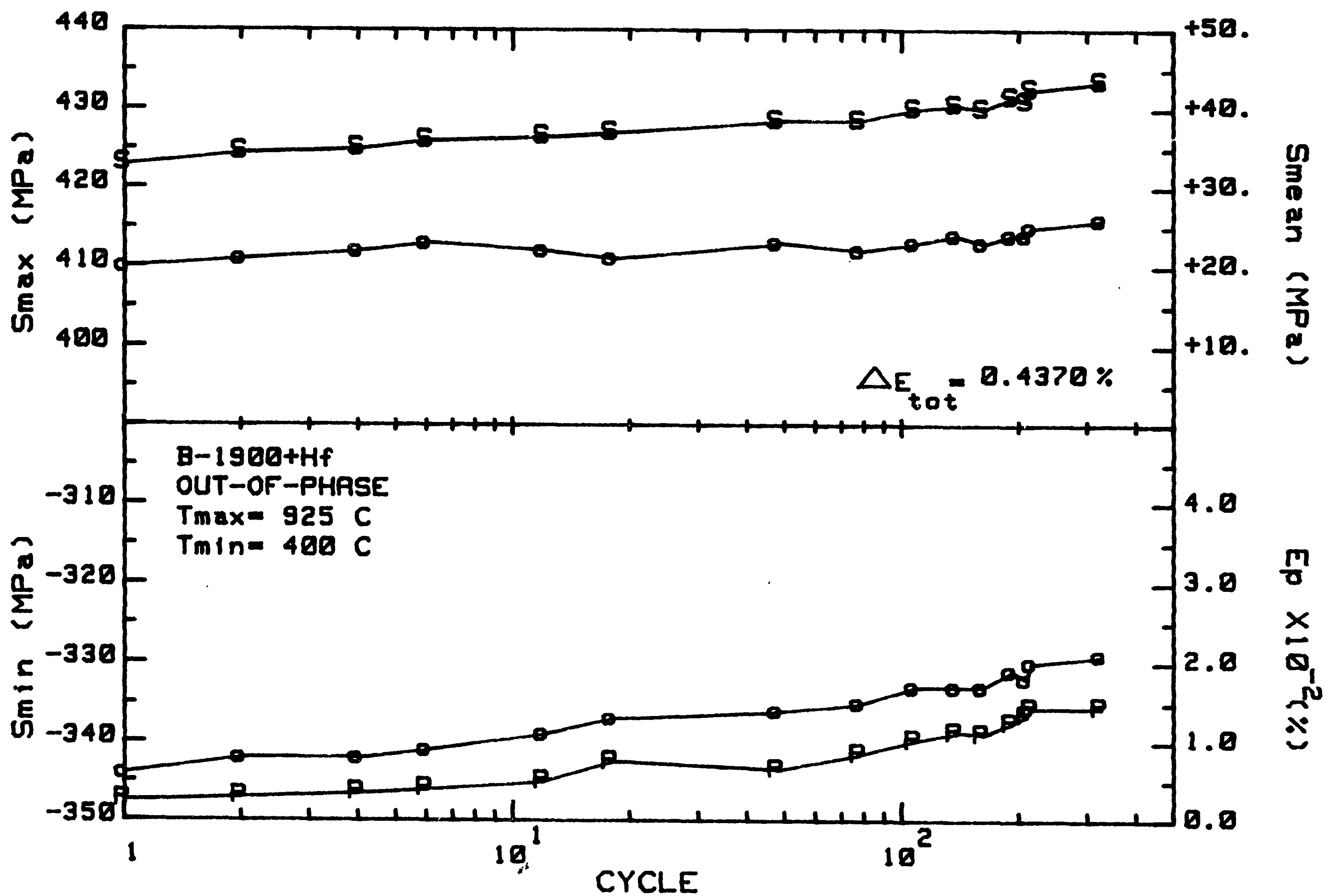
Smax (MPa)
340
330
320
310
300
Smean (MPa)
+50.
+40.
+30.
+20.
+10.
ΔE_tot = 0.3279 %
B-1900+Hf
OUT-OF-PHASE
Tmax= 925 C
Tmin= 400 C
Smin (MPa)
-270
-280
-290
-300
-310
Ep X10^-2 (%)
4.0
3.0
2.0
1.0
0.0
1
10^1
10^2
CYCLE

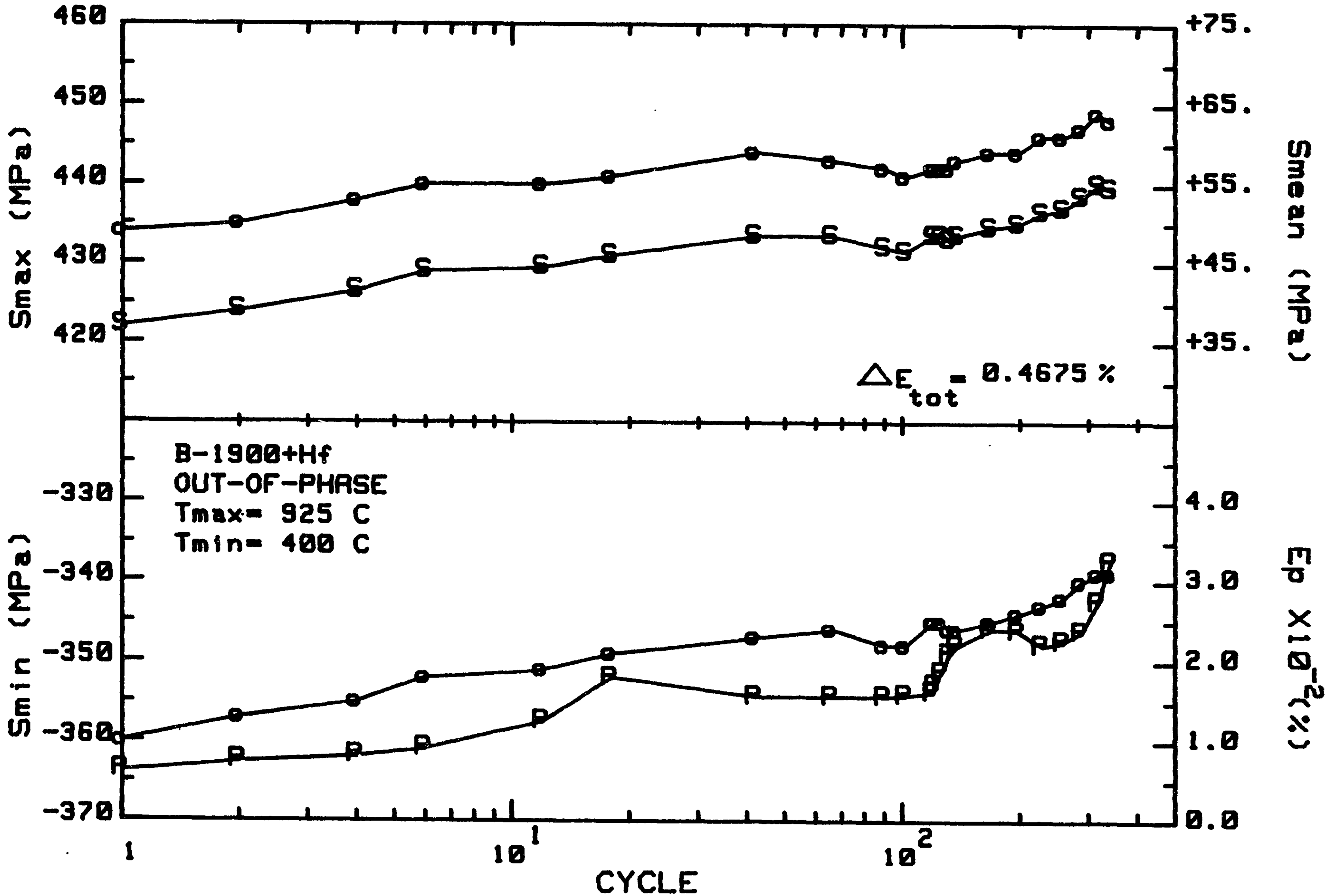


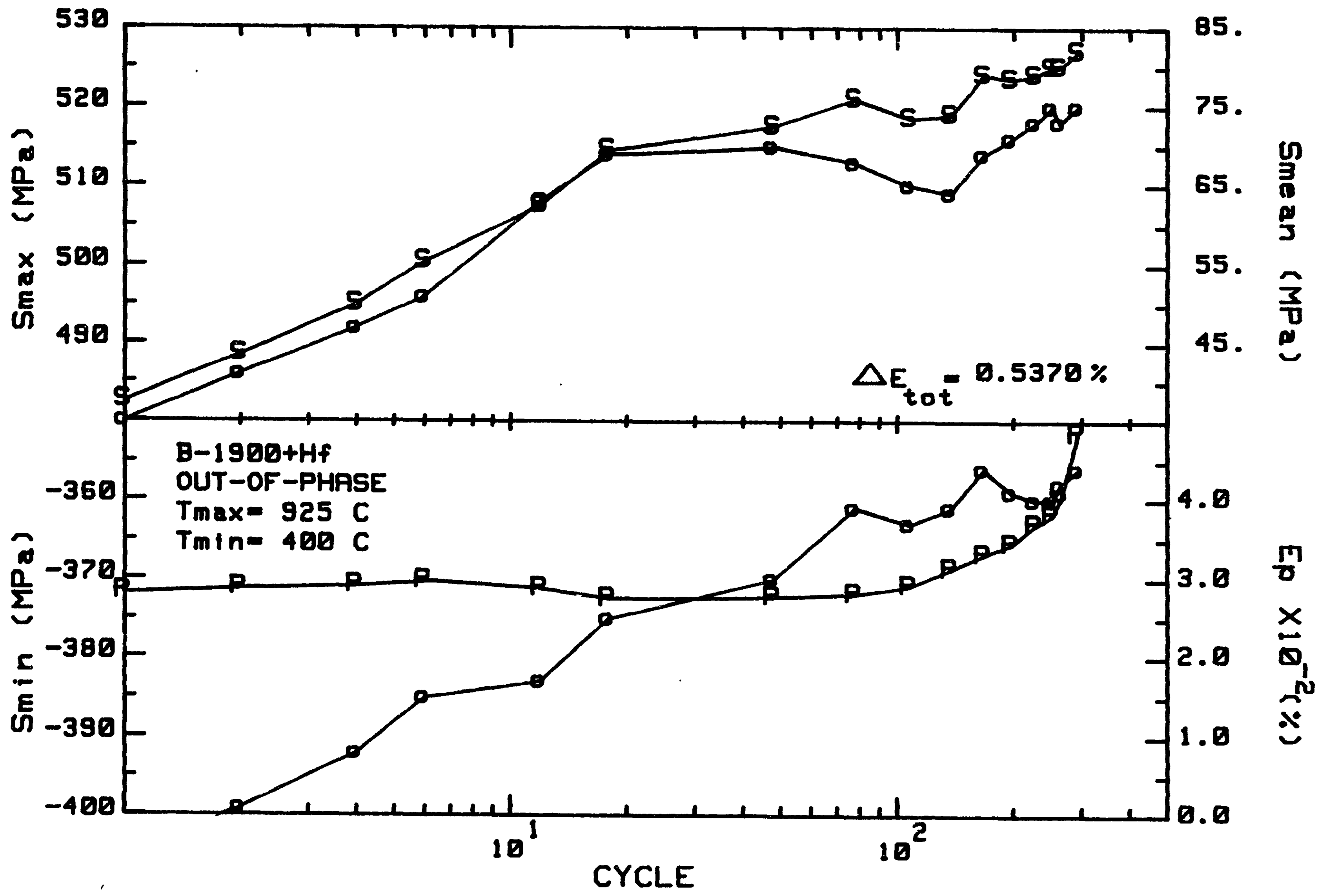
Smax (MPa)
Smean (MPa)
Smin (MPa)
Ep X10^-3 (%)
CYCLE
ΔE_tot = 0.3654 %
B-1900+Hf
OUT-OF-PHASE
Tmax= 925 C
Tmin= 400 C



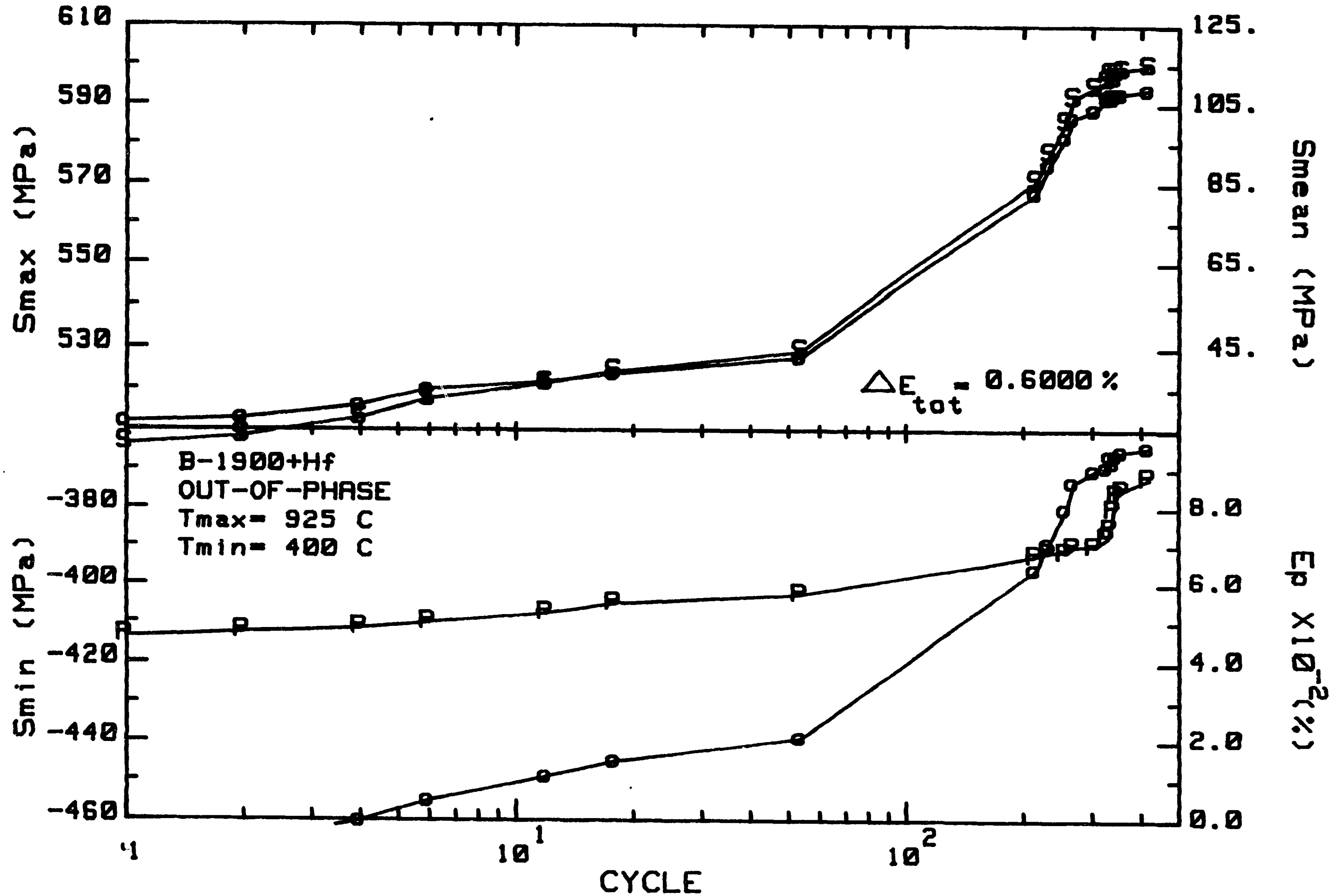
Smax (MPa)
390
380
370
360
350
Smean (MPa)
+50.
+40.
+30.
+20.
+10.
ΔE tot = 0.4040 %
B-1900+Hf
OUT-OF-PHASE
Tmax= 925 C
Tmin= 400 C
Smin (MPa)
-310
-320
-330
-340
-350
Ep X10^-2 (%)
4.0
3.0
2.0
1.0
0.0
1
10^1
10^2
CYCLE


Smax (MPa)
440
430
420
410
400
Smean (MPa)
+50.
+40.
+30.
+20.
+10.
ΔE tot = 0.4370 %
B-1900+Hf
OUT-OF-PHASE
Tmax= 925 C
Tmin= 400 C
Smin (MPa)
-310
-320
-330
-340
-350
Ep X10^-2 (%)
4.0
3.0
2.0
1.0
0.0
1
10^1
10^2
CYCLE

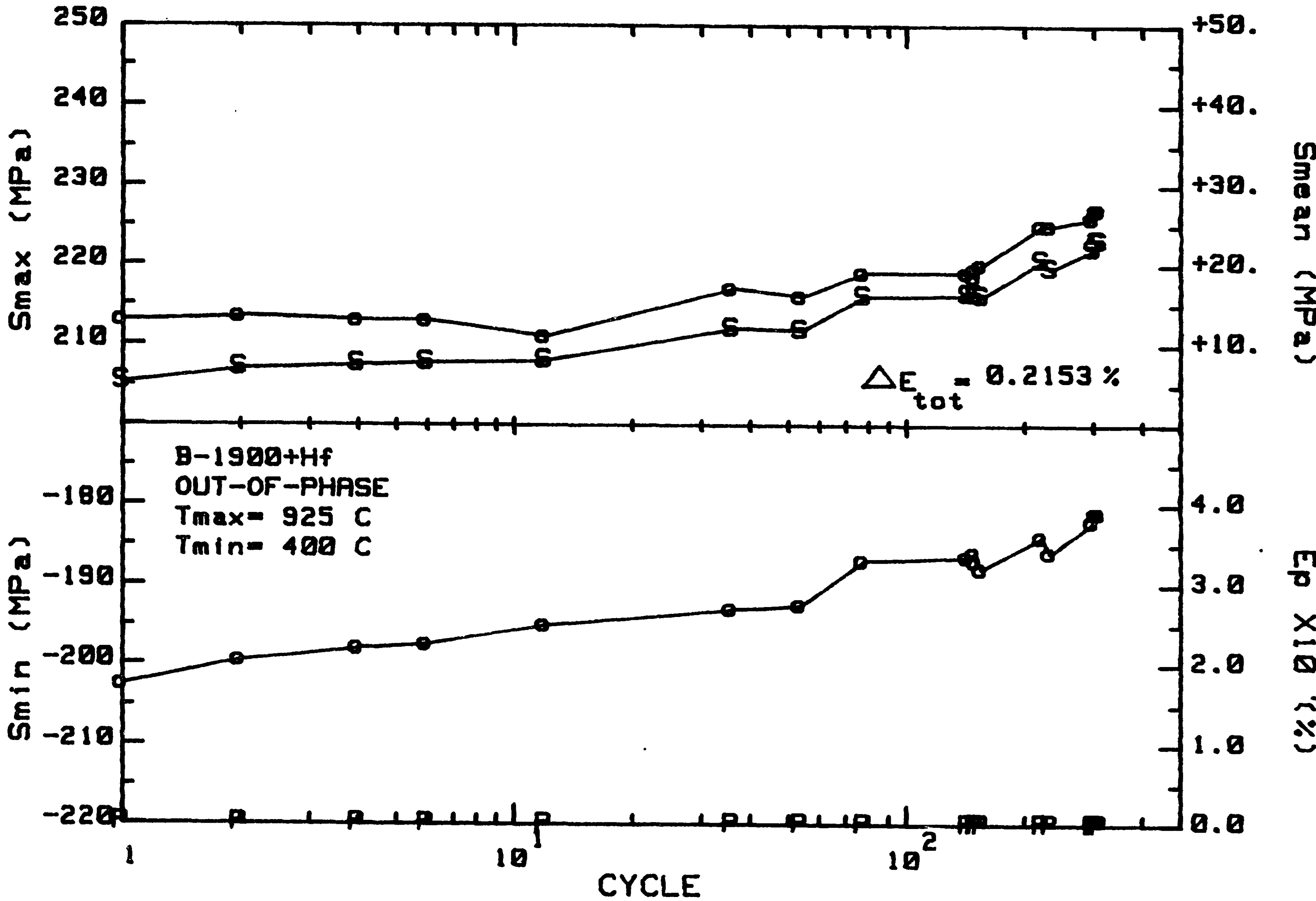

Smax (MPa)
Smean (MPa)
Smin (MPa)
Ep X10^-2(%)
ΔE_{tot} = 0.4675 %
B-1900+Hf
OUT-OF-PHASE
Tmax= 925 C
Tmin= 400 C
CYCLE



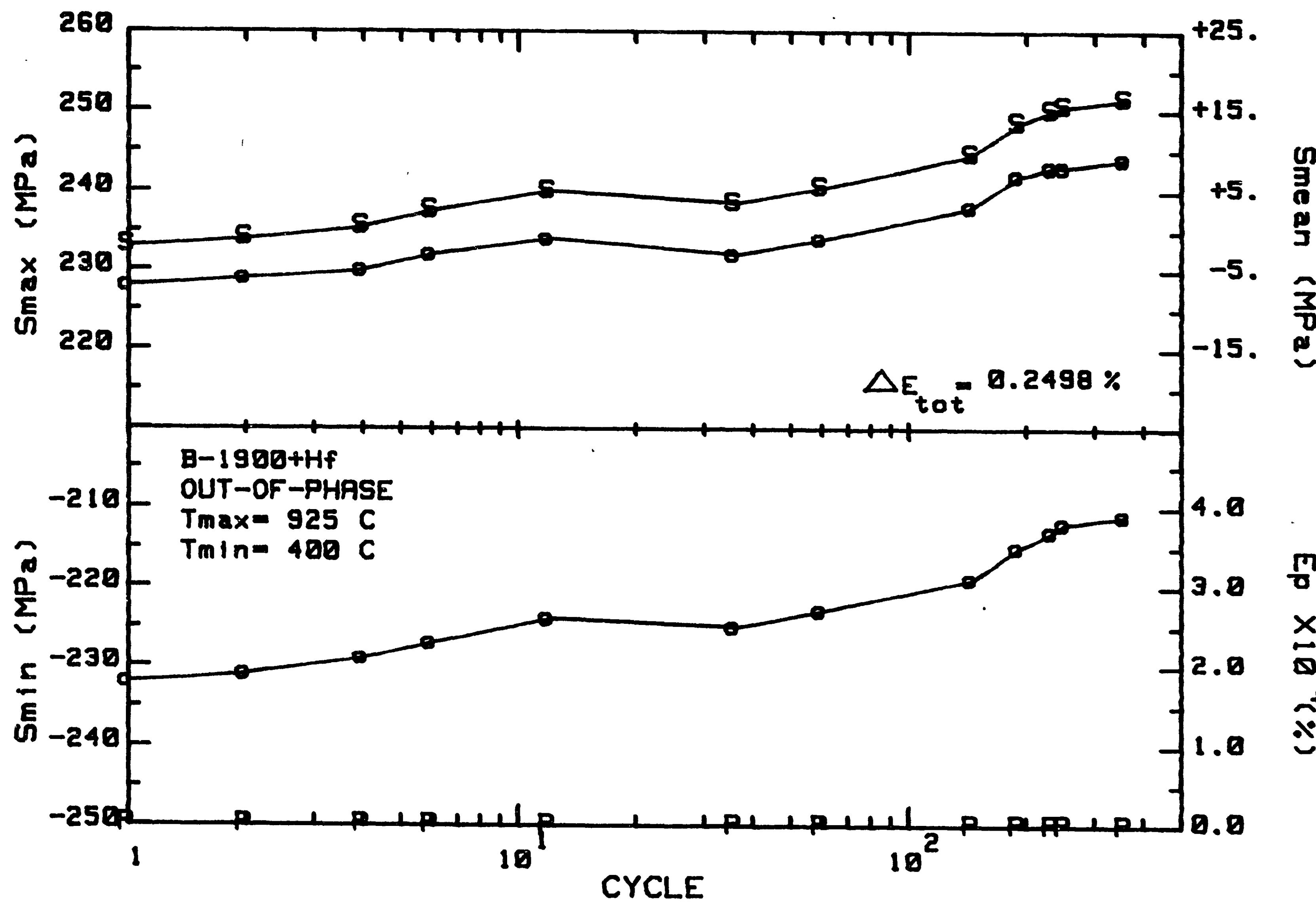
Smax (MPa)
530
520
510
500
490
Smean (MPa)
85.
75.
65.
55.
45.
ΔE tot = 0.5370 %
B-1900+Hf
OUT-OF-PHASE
Tmax= 925 C
Tmin= 400 C
Smin (MPa)
-360
-370
-380
-390
-400
Ep X10^-2 (%)
4.0
3.0
2.0
1.0
0.0
10^1
10^2
CYCLE



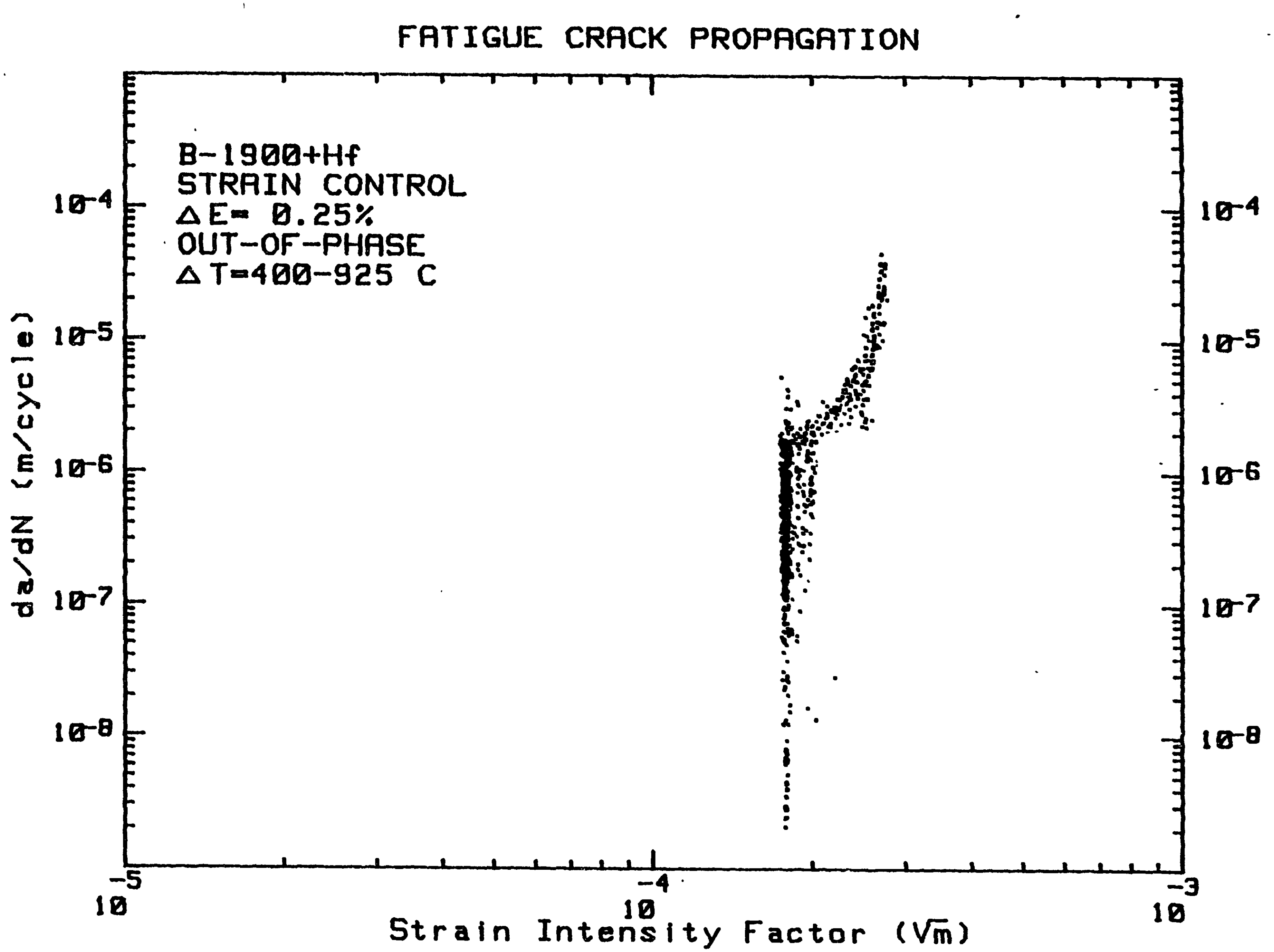
610
590
570
550
530
Smax (MPa)
-380
-400
-420
-440
-460
Smin (MPa)
125.
105.
85.
65.
45.
Smean (MPa)
8.0
6.0
4.0
2.0
0.0
Ep X10^{-2}(%)
ΔE_{tot} = 0.6000 %
B-1900+Hf
OUT-OF-PHASE
Tmax= 925 C
Tmin= 400 C
1
10^1
10^2
CYCLE



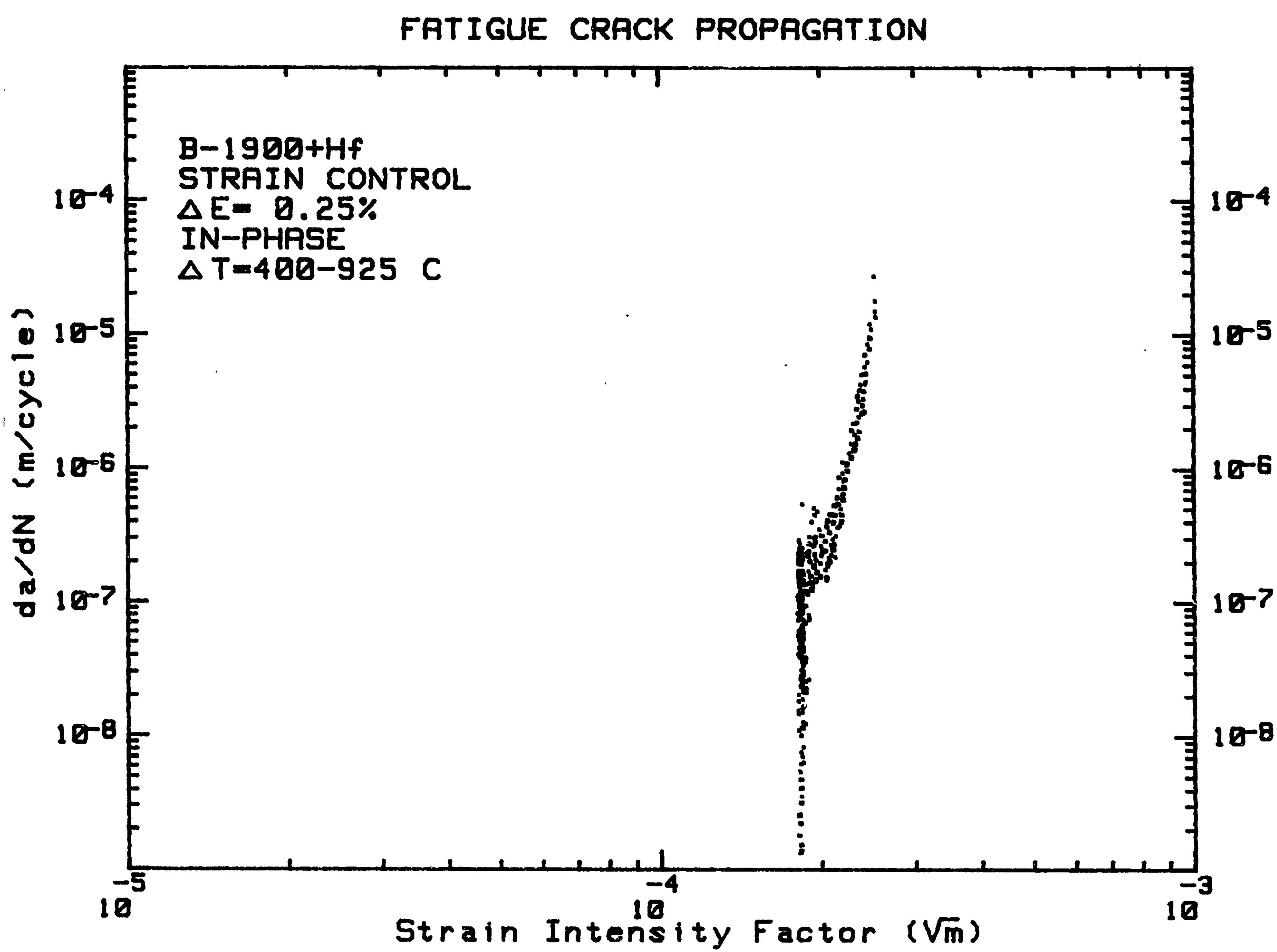
Smax (MPa)
250
240
230
220
210
Smean (MPa)
+50.
+40.
+30.
+20.
+10.
ΔE tot = 0.2153 %
B-1900+Hf
OUT-OF-PHASE
Tmax= 925 C
Tmin= 400 C
Smin (MPa)
-180
-190
-200
-210
-220
Ep X10^-4 (%)
4.0
3.0
2.0
1.0
0.0
1
10^1
10^2
CYCLE

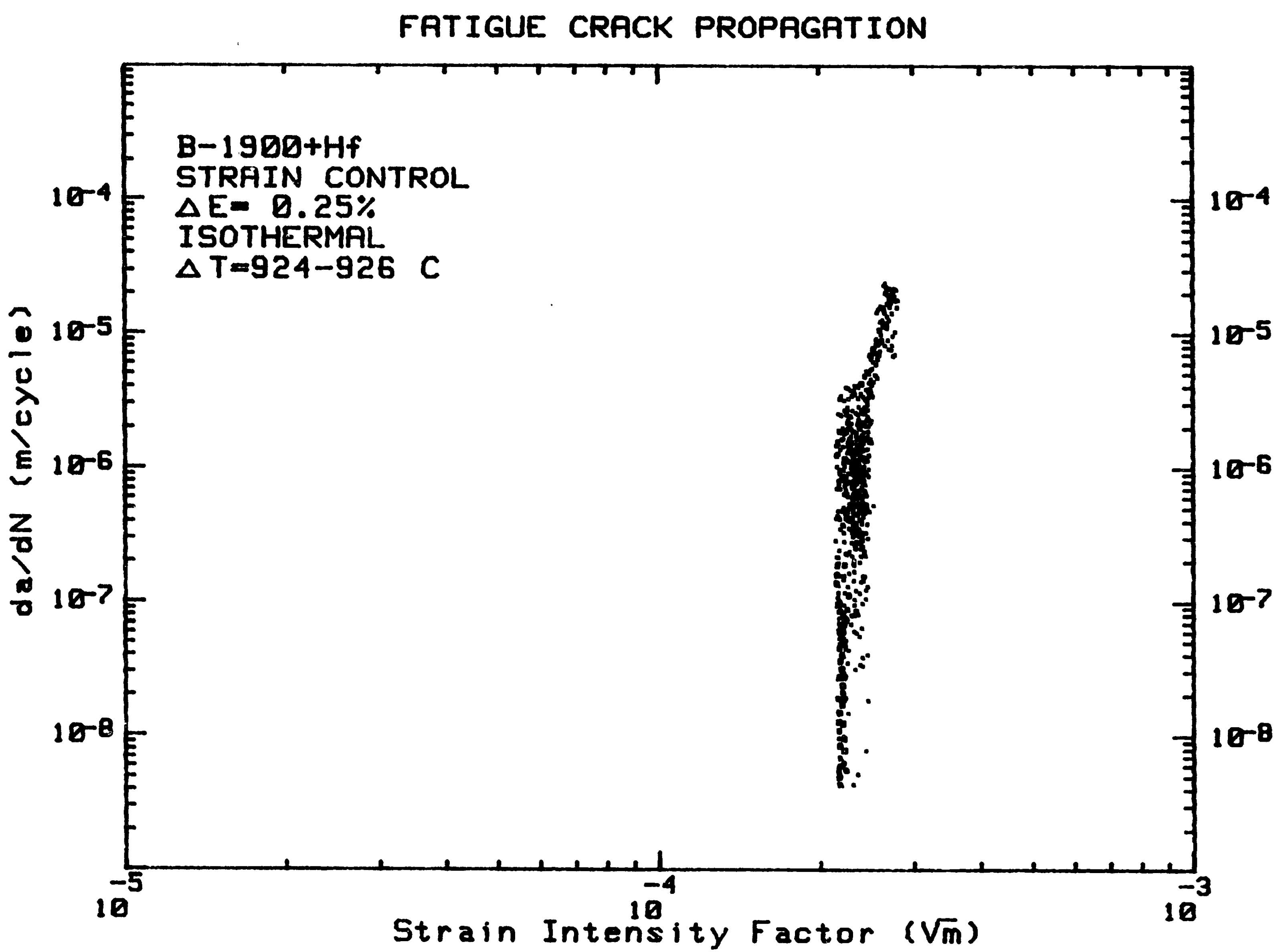


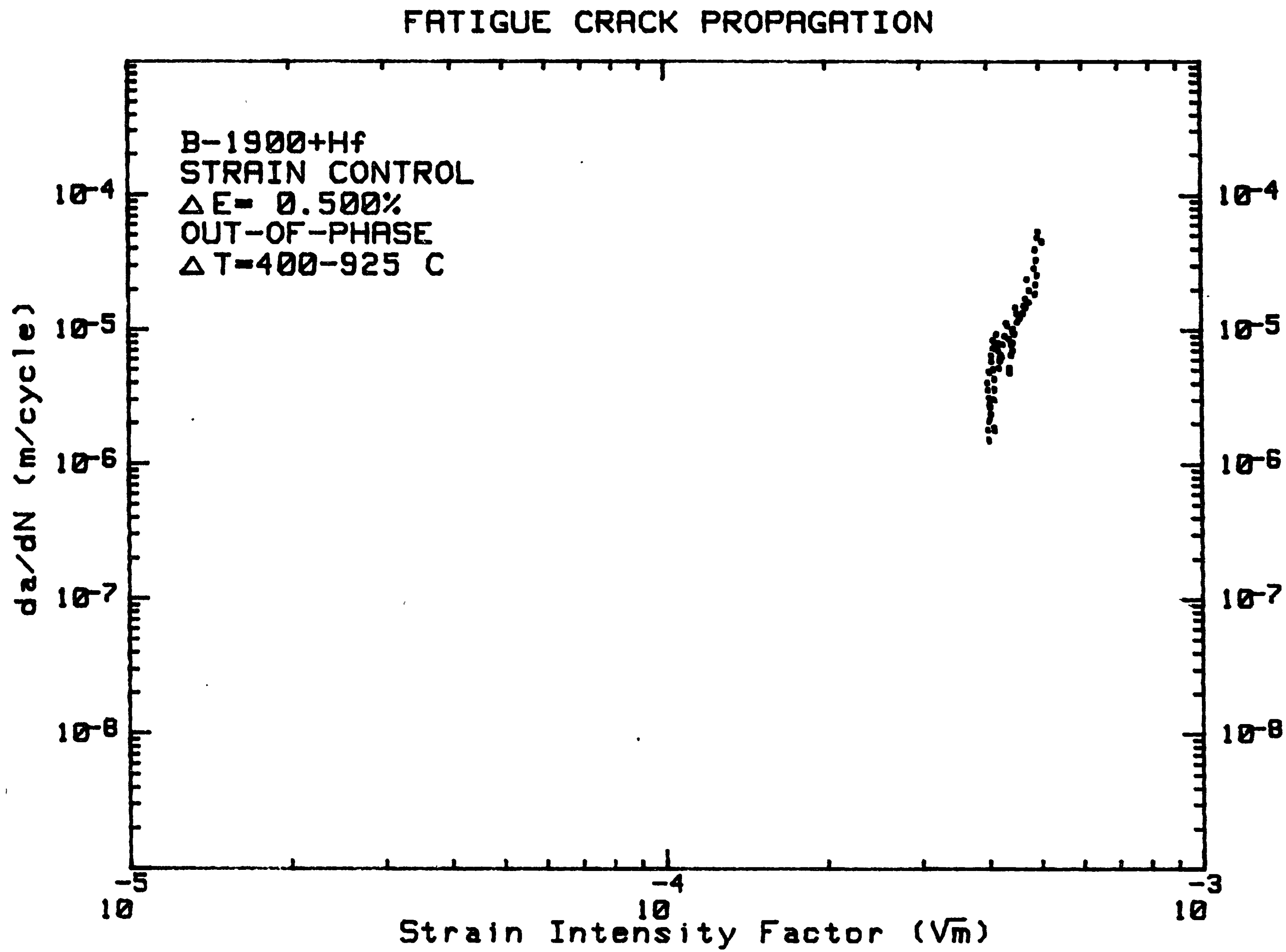
Smax (MPa)
260
250
240
230
220
Smean (MPa)
+25.
+15.
+5.
-5.
-15.
ΔE_{tot} = 0.2498 %
B-1900+Hf
OUT-OF-PHASE
Tmax= 925 C
Tmin= 400 C
Smin (MPa)
-210
-220
-230
-240
-250
Ep X10^{-4}(%)
4.0
3.0
2.0
1.0
0.0
1
10^1
10^2
CYCLE



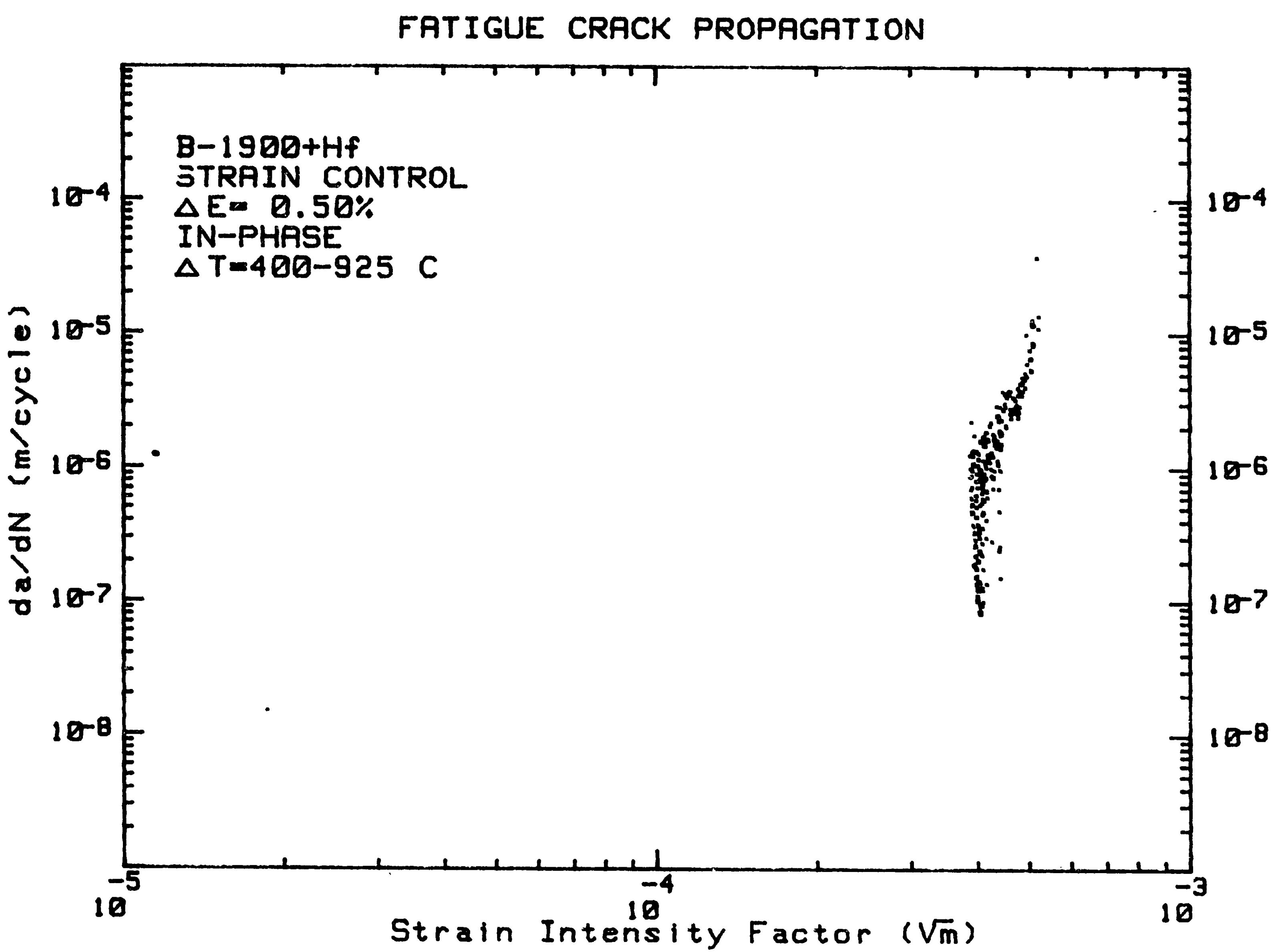
FATIGUE CRACK PROPAGATION
B-1900+Hf
STRAIN CONTROL
ΔE= 0.25%
OUT-OF-PHASE
ΔT=400-925 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10-5
10-4
10-3
Strain Intensity Factor (√m)


FATIGUE CRACK PROPAGATION
B-1900+Hf
STRAIN CONTROL
ΔE= 0.25%
IN-PHASE
ΔT=400-925 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
Strain Intensity Factor (√m)
10-5
10-4
10-3

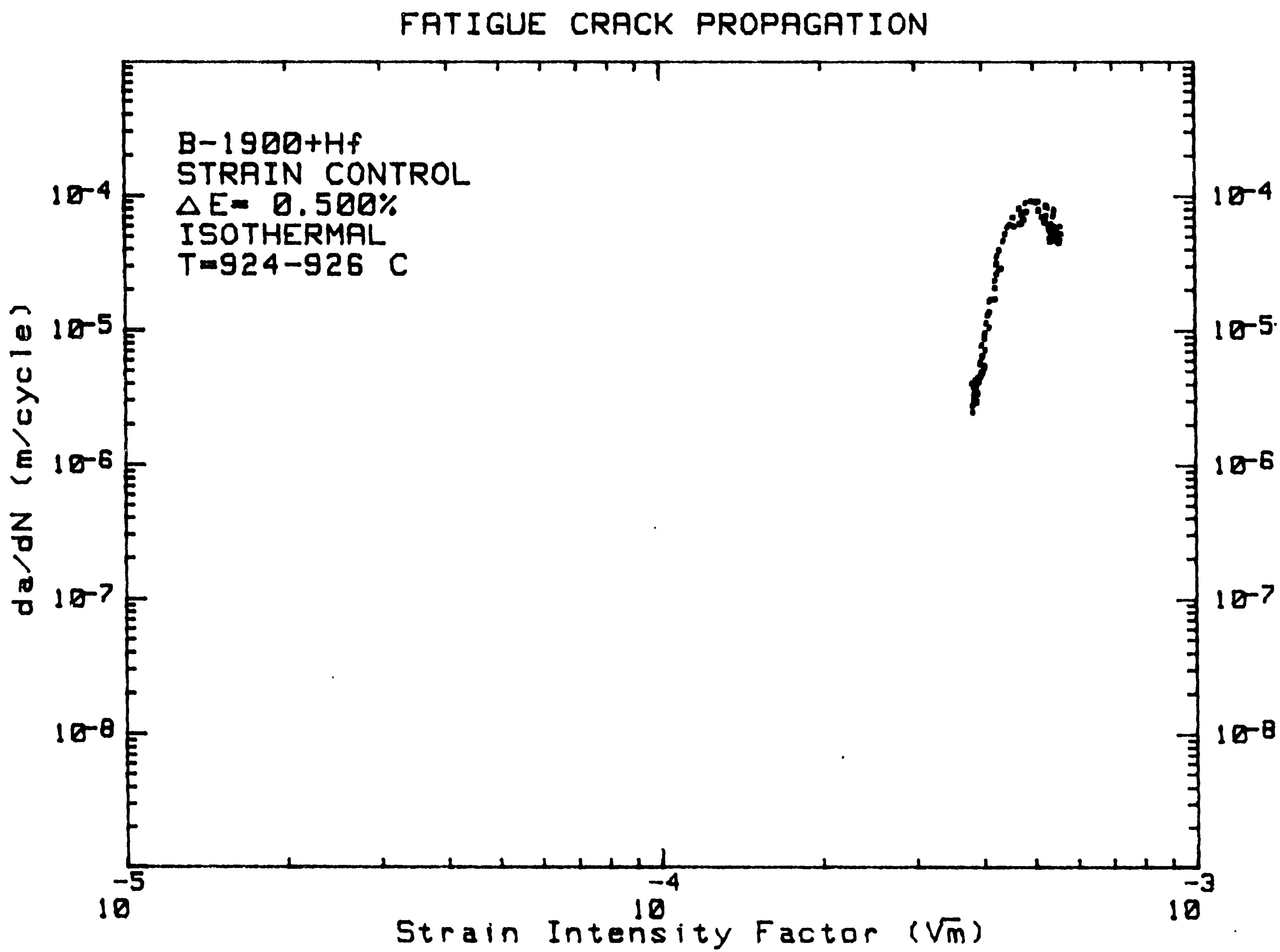

FATIGUE CRACK PROPAGATION
B-1900+Hf
STRAIN CONTROL
ΔE= 0.25%
ISOTHERMAL
ΔT=924-926 C
da/dN (m/cycle)
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-5}
10^{-4}
10^{-3}
Strain Intensity Factor (√m)



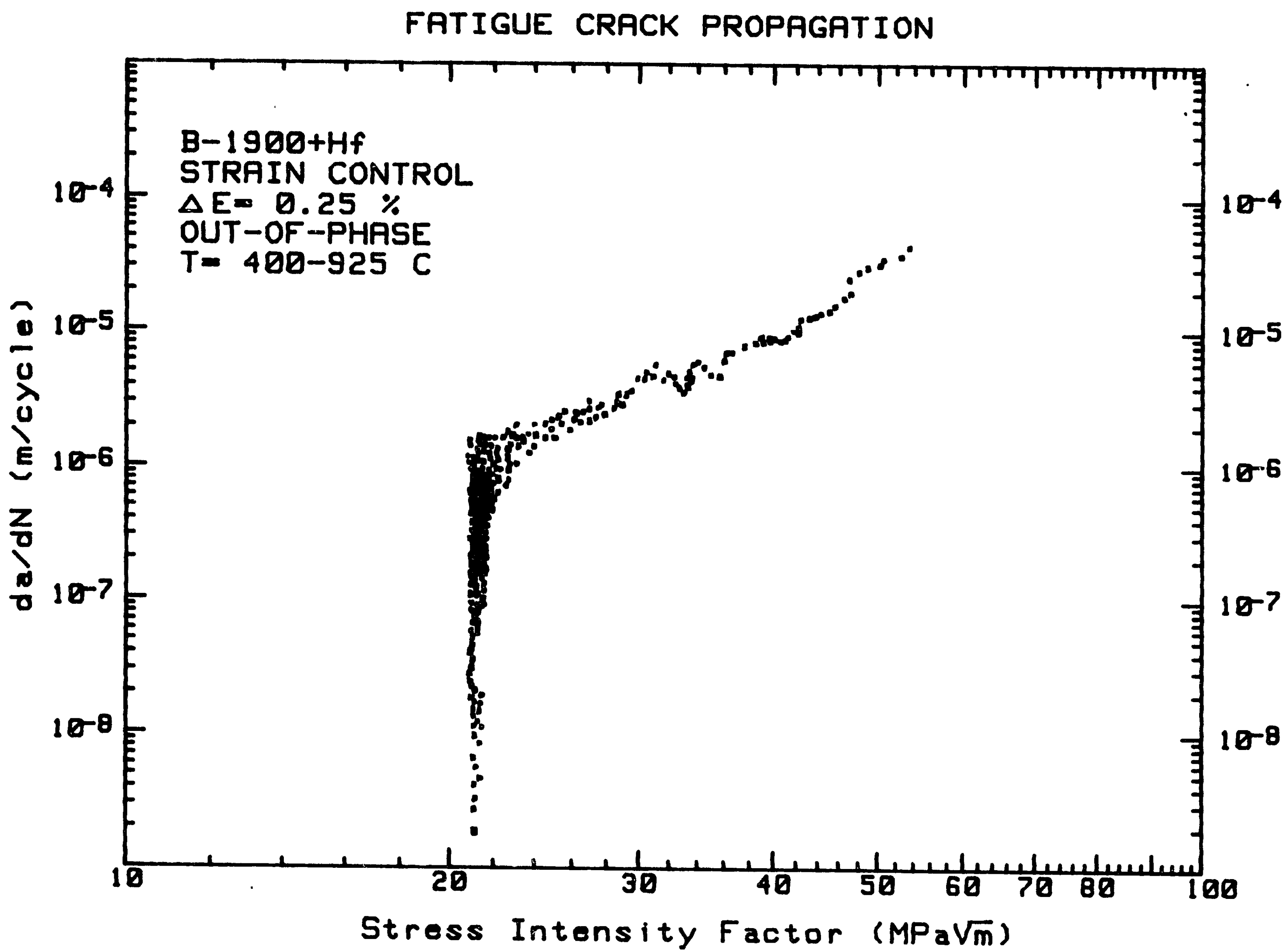
FATIGUE CRACK PROPAGATION
B-1900+Hf
STRAIN CONTROL
ΔE= 0.500%
OUT-OF-PHASE
ΔT=400-925 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10-5
10-4
10-3
Strain Intensity Factor (√m)



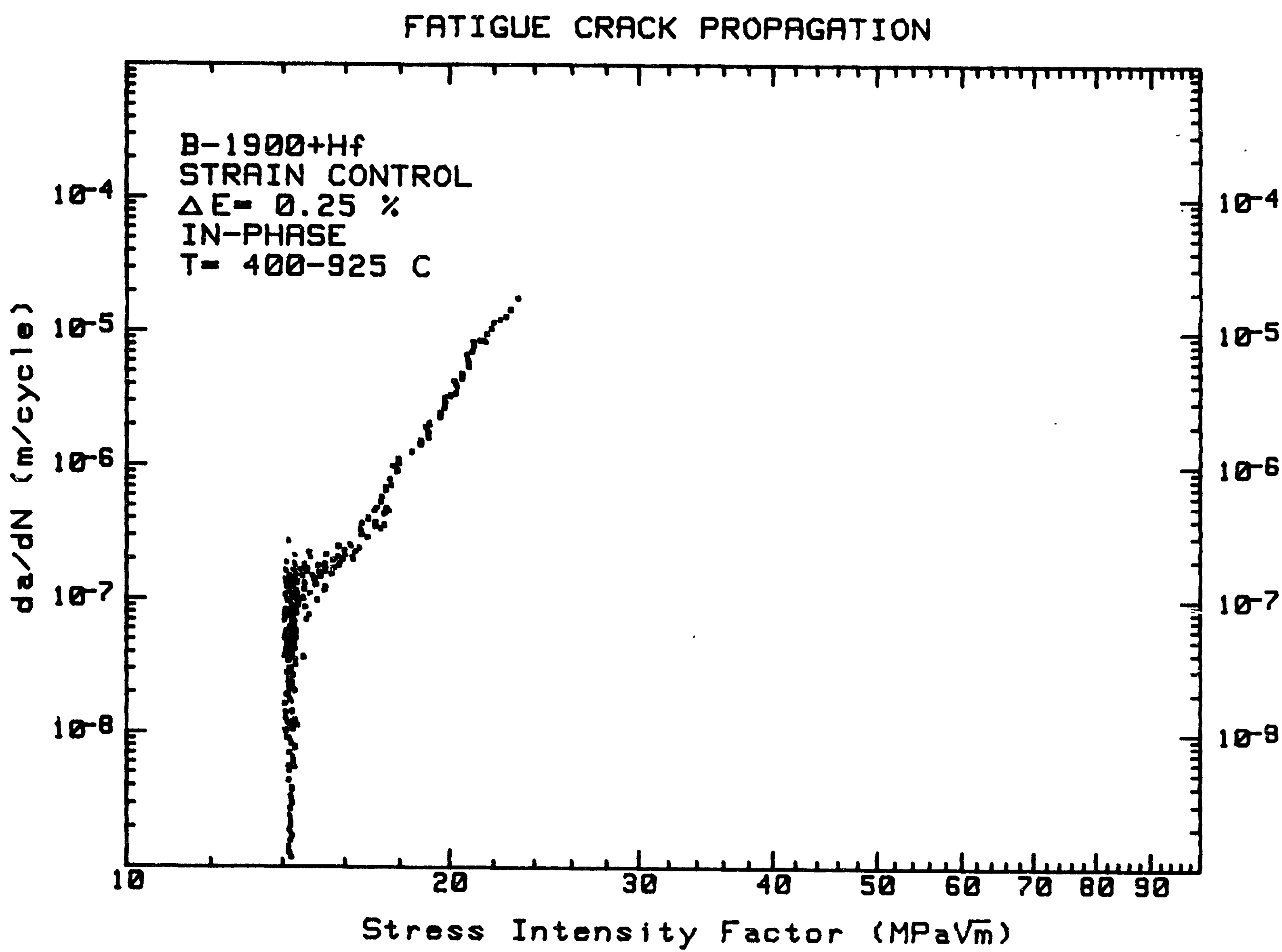
FATIGUE CRACK PROPAGATION
B-1900+Hf
STRAIN CONTROL
ΔE= 0.50%
IN-PHASE
ΔT=400-925 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10-5
10-4
10-3
Strain Intensity Factor (√m)

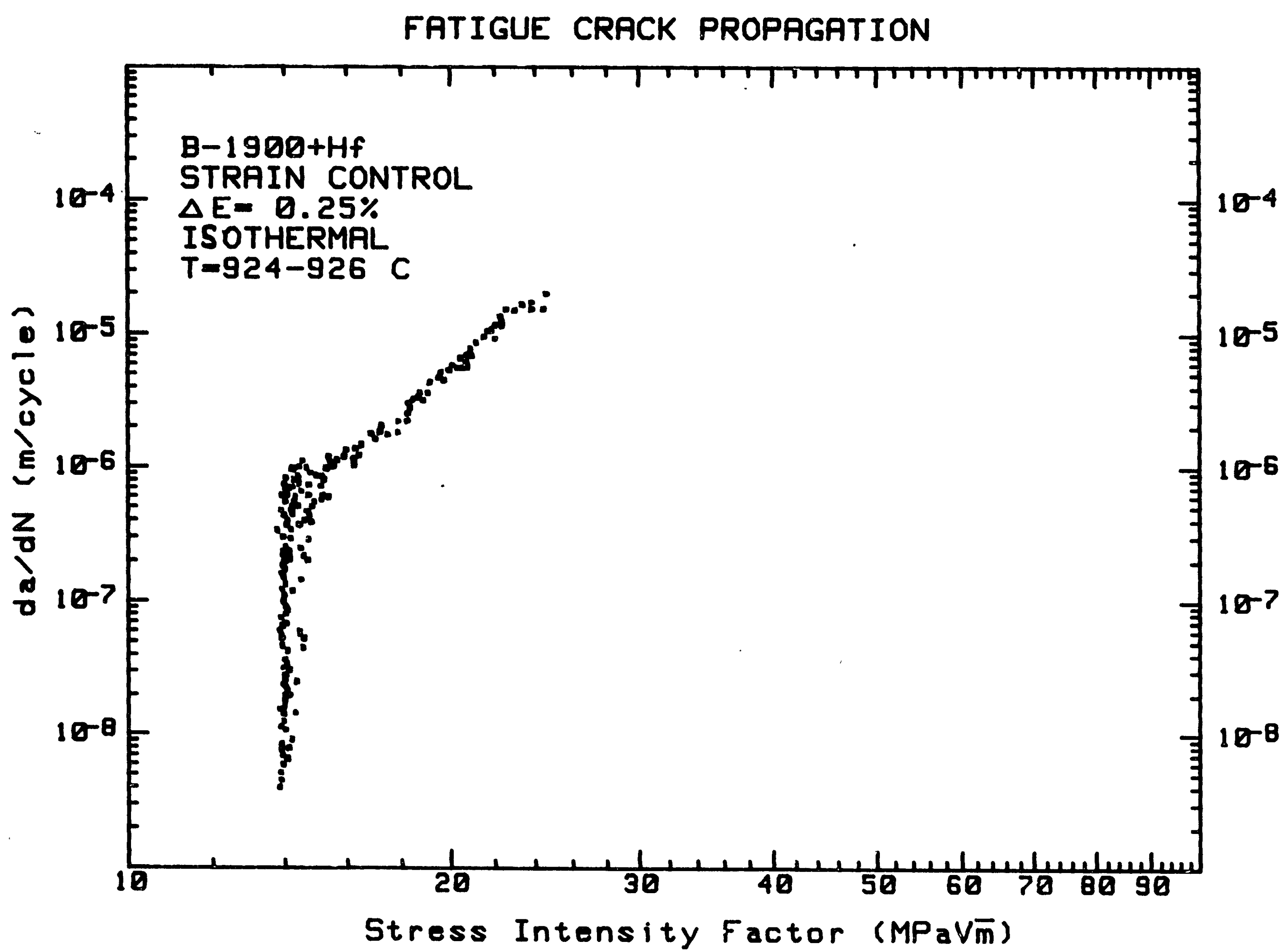


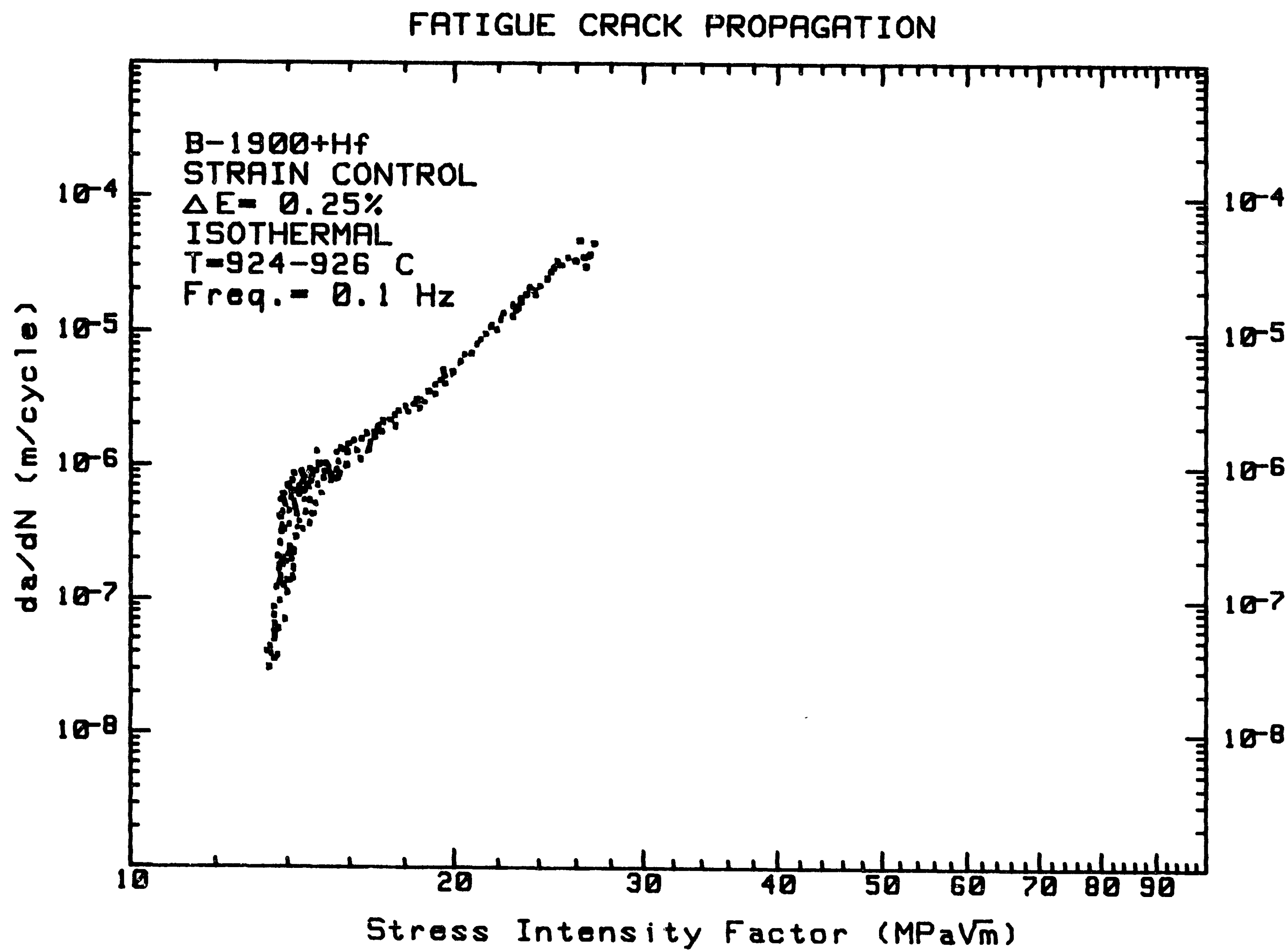
FATIGUE CRACK PROPAGATION
B-1900+Hf
STRAIN CONTROL
ΔE= 0.500%
ISOTHERMAL
T=924-926 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10-5
10-4
10-3
Strain Intensity Factor (√m)

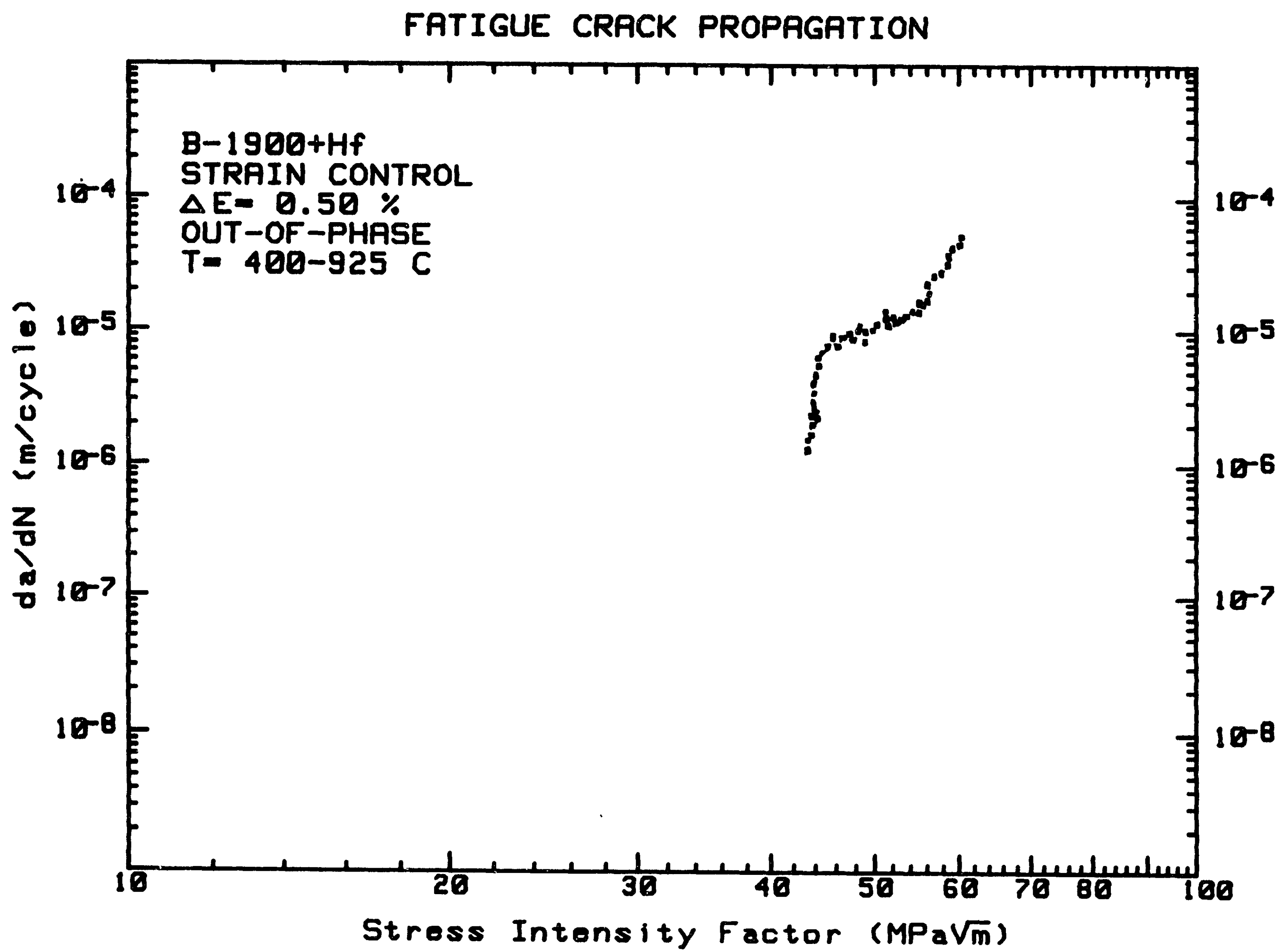


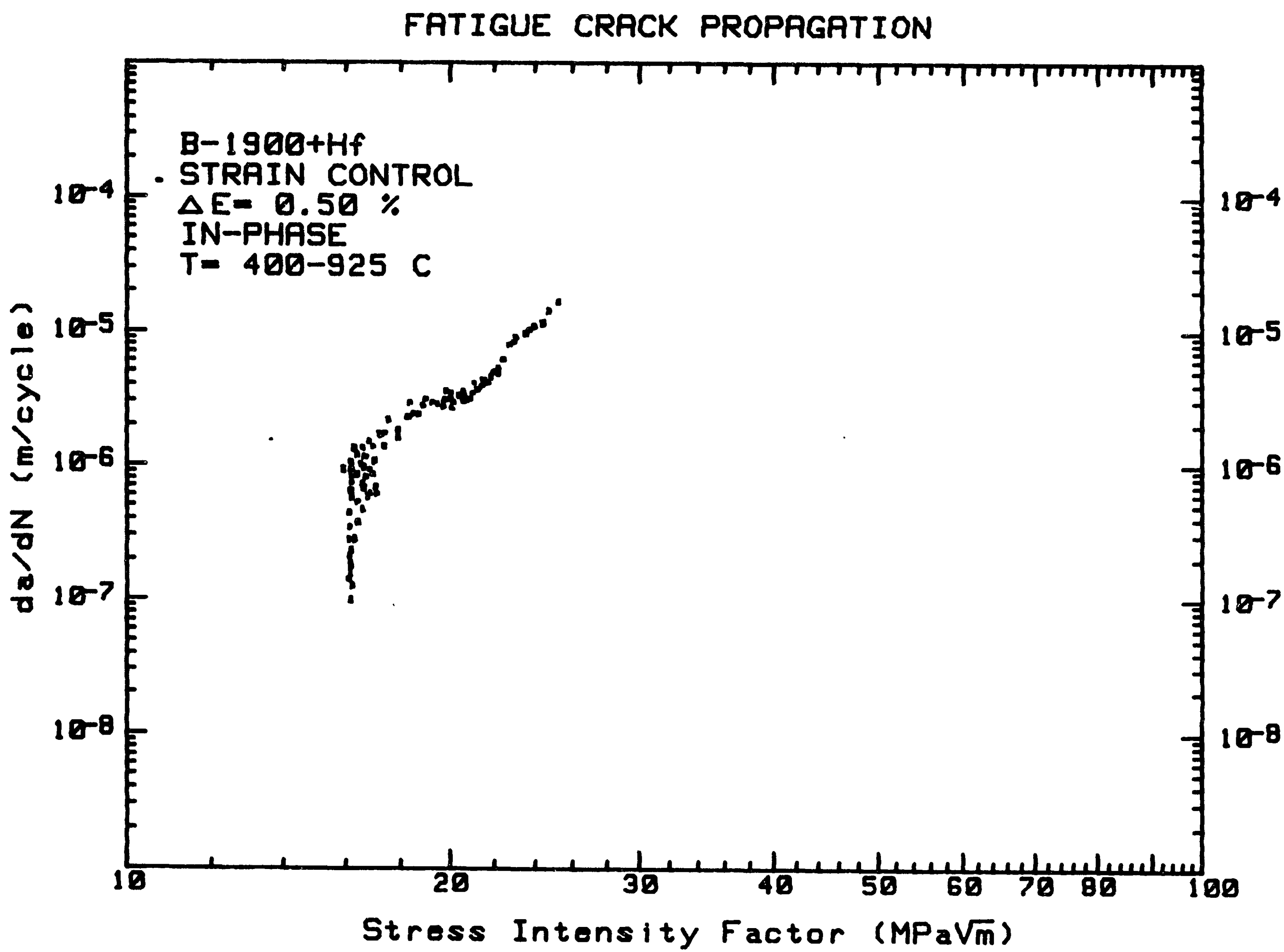
FATIGUE CRACK PROPAGATION
B-1900+Hf
STRAIN CONTROL
ΔE= 0.25 %
OUT-OF-PHASE
T= 400-925 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10
20
30
40
50
60
70
80
100
Stress Intensity Factor (MPa√m)


FATIGUE CRACK PROPAGATION
B-1900+Hf
STRAIN CONTROL
ΔE= 0.25 %
IN-PHASE
T= 400-925 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10
20
30
40
50
60
70
80
90
Stress Intensity Factor (MPa√m)

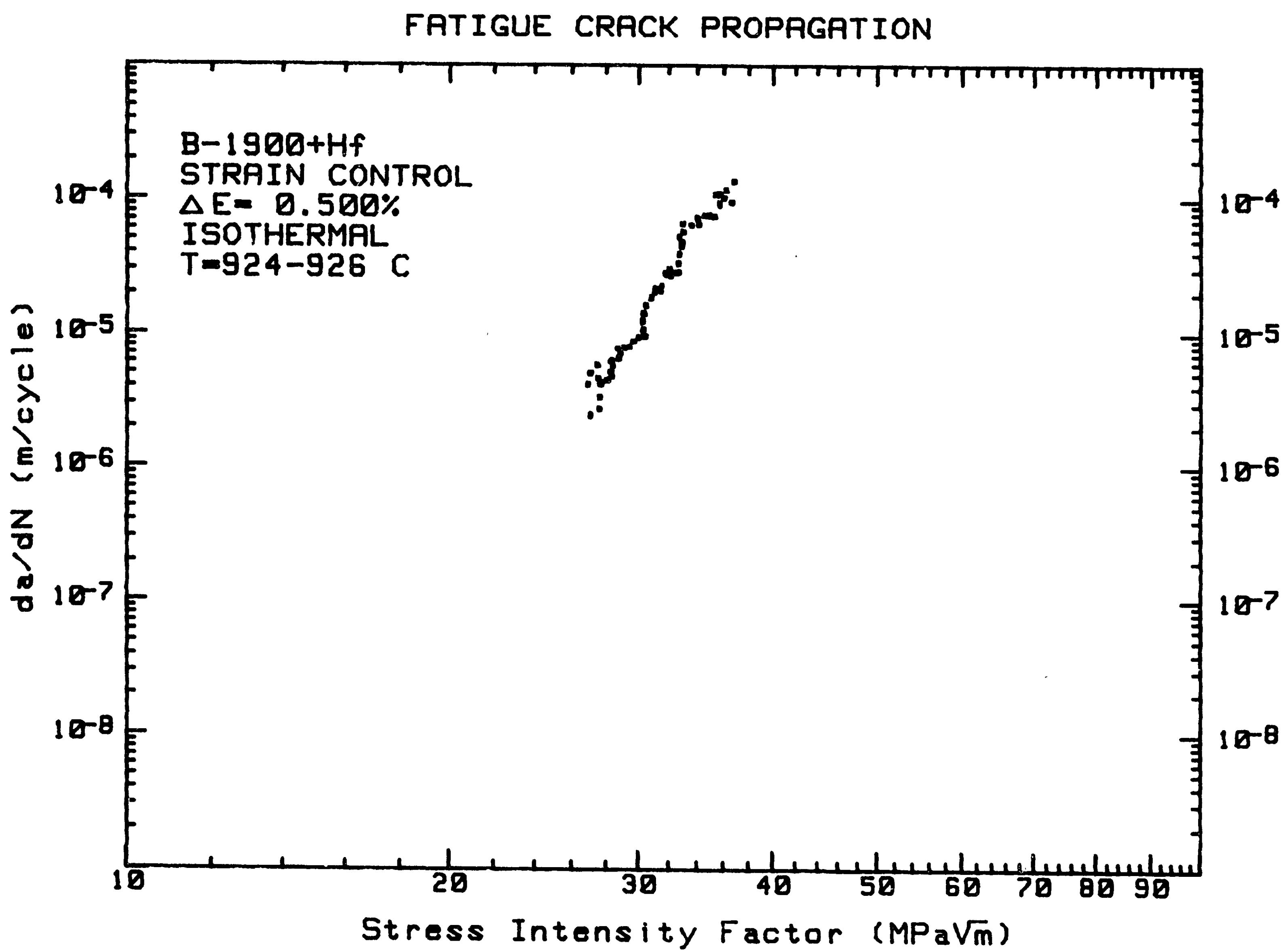

FATIGUE CRACK PROPAGATION
B-1900+Hf
STRAIN CONTROL
ΔE= 0.25%
ISOTHERMAL
T=924-926 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10
20
30
40
50
60
70
80
90
Stress Intensity Factor (MPa√m)

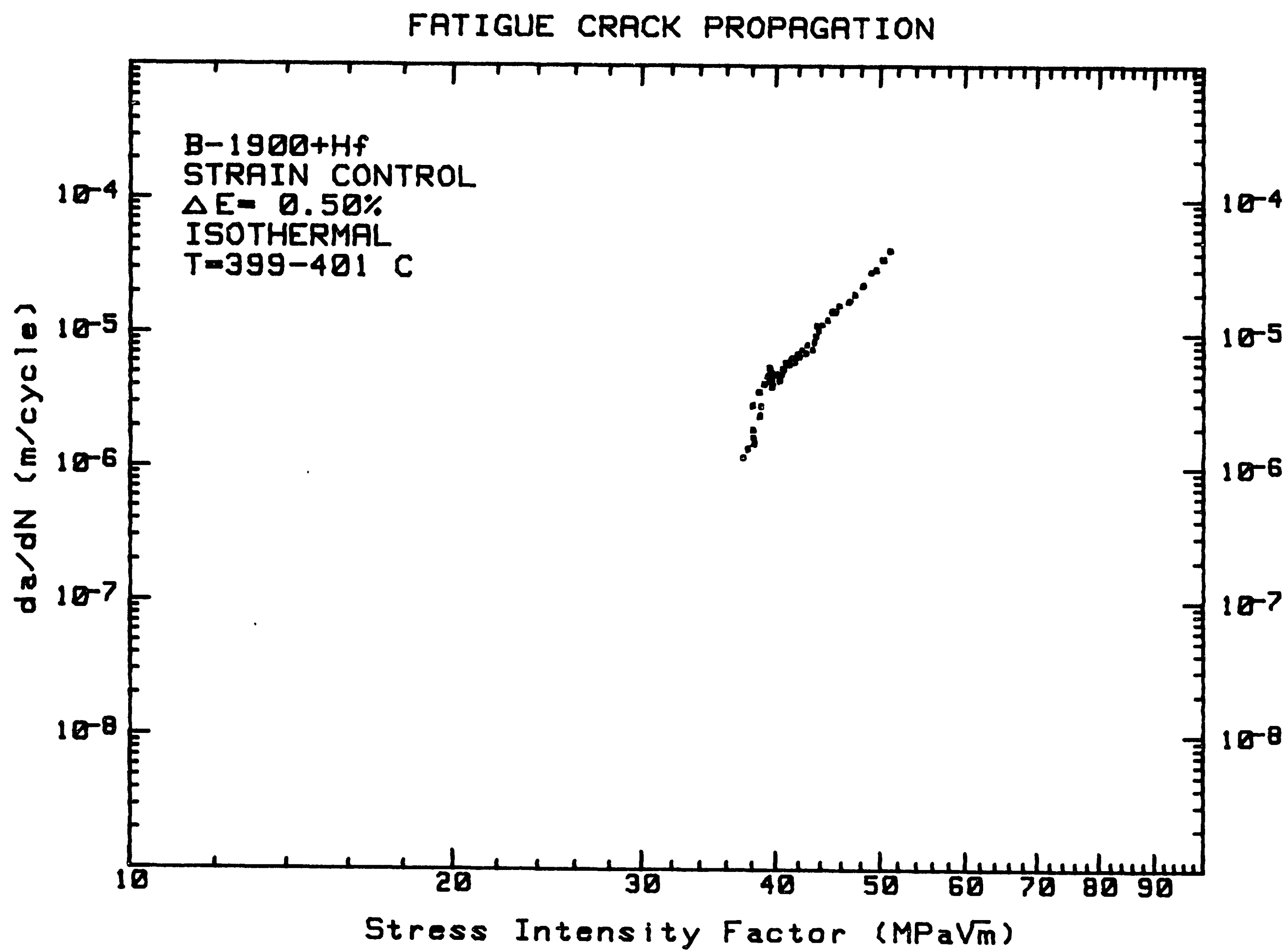

FATIGUE CRACK PROPAGATION
B-1900+Hf
STRAIN CONTROL
ΔE= 0.25%
ISOTHERMAL
T=924-926 C
Freq.= 0.1 Hz
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10
20
30
40
50
60
70
80
90
Stress Intensity Factor (MPa√m)

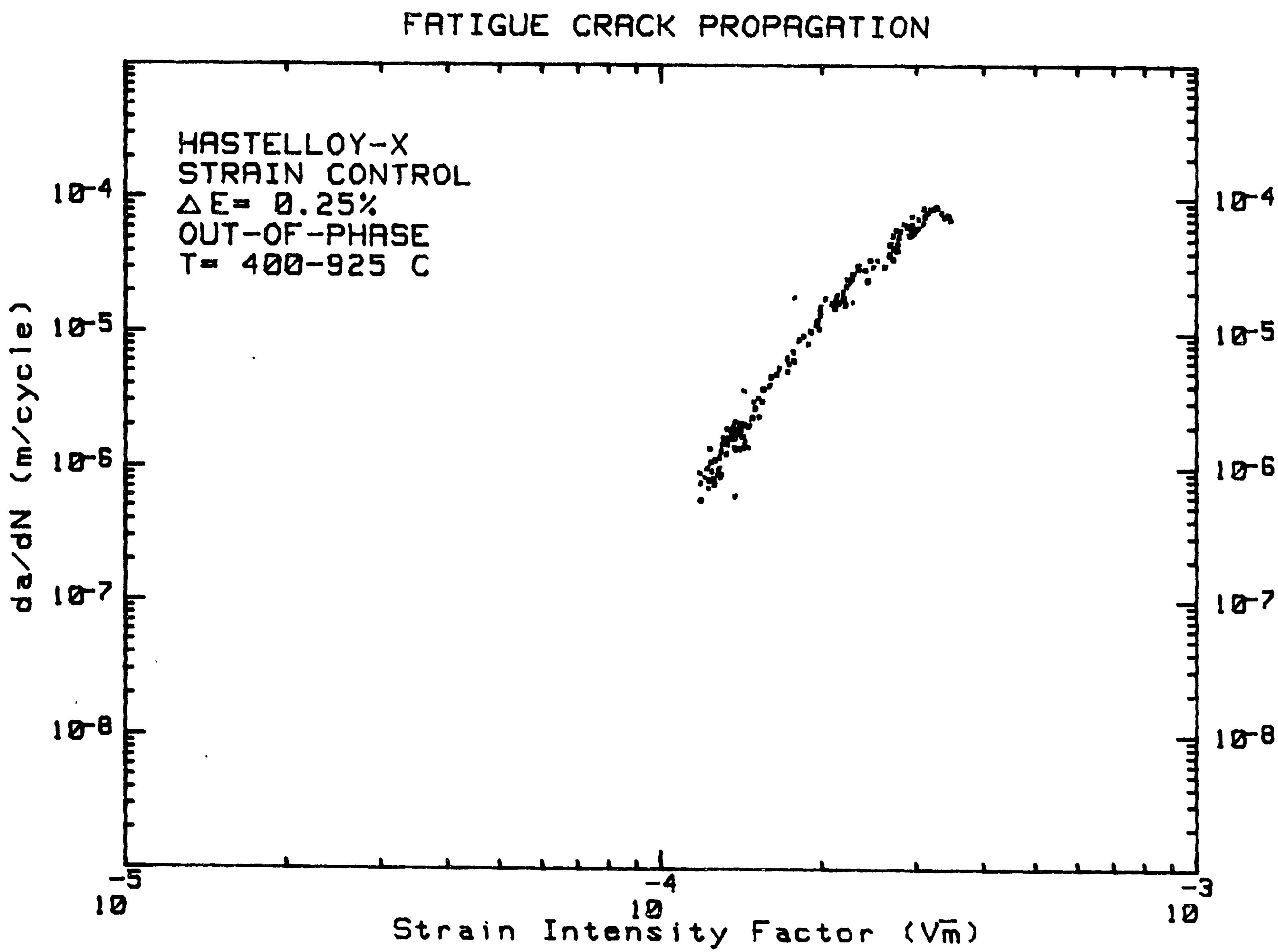

FATIGUE CRACK PROPAGATION
B-1900+Hf
STRAIN CONTROL
ΔE= 0.50 %
OUT-OF-PHASE
T= 400-925 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10
20
30
40
50
60
70
80
100
Stress Intensity Factor (MPa√m)

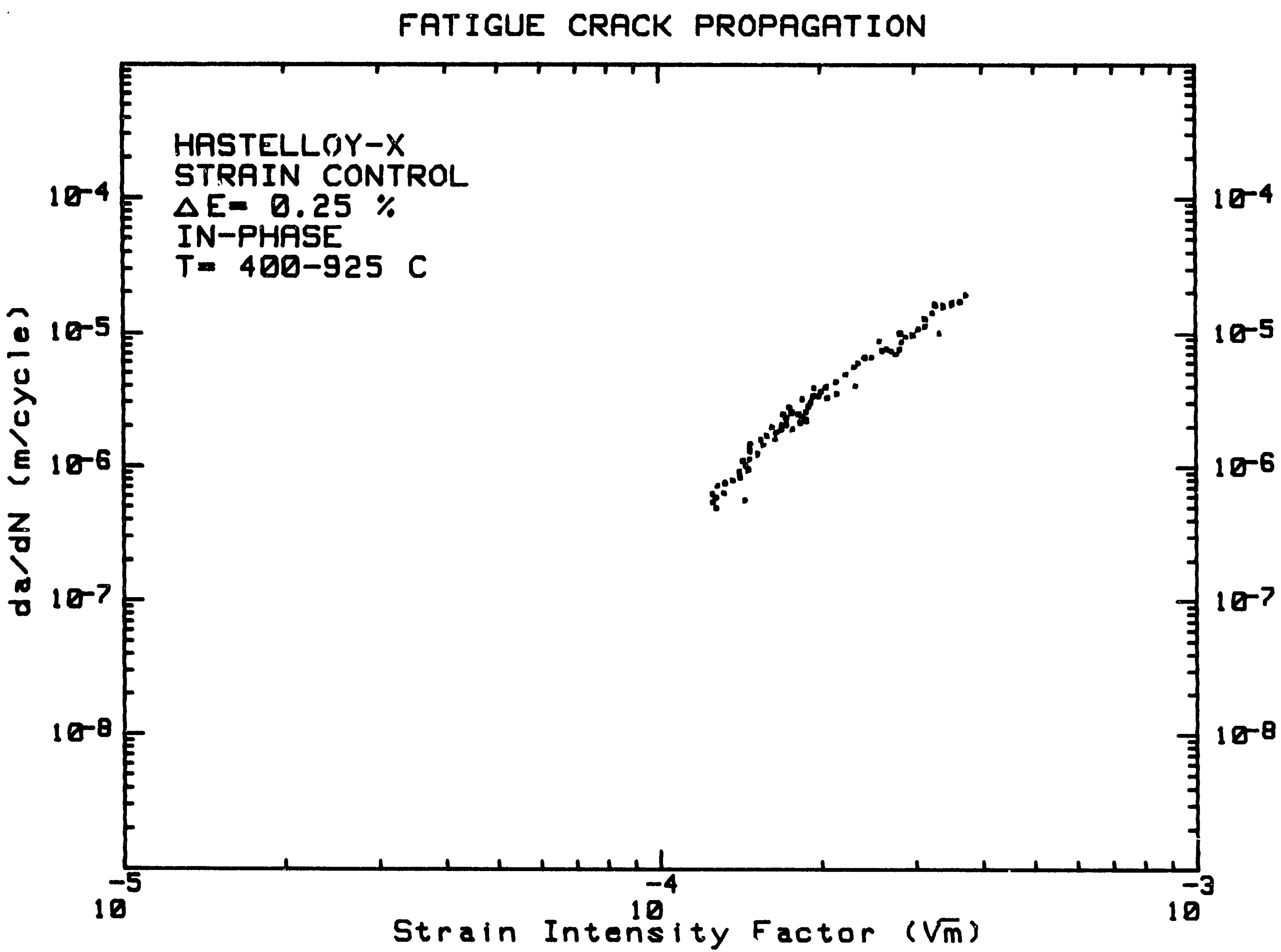


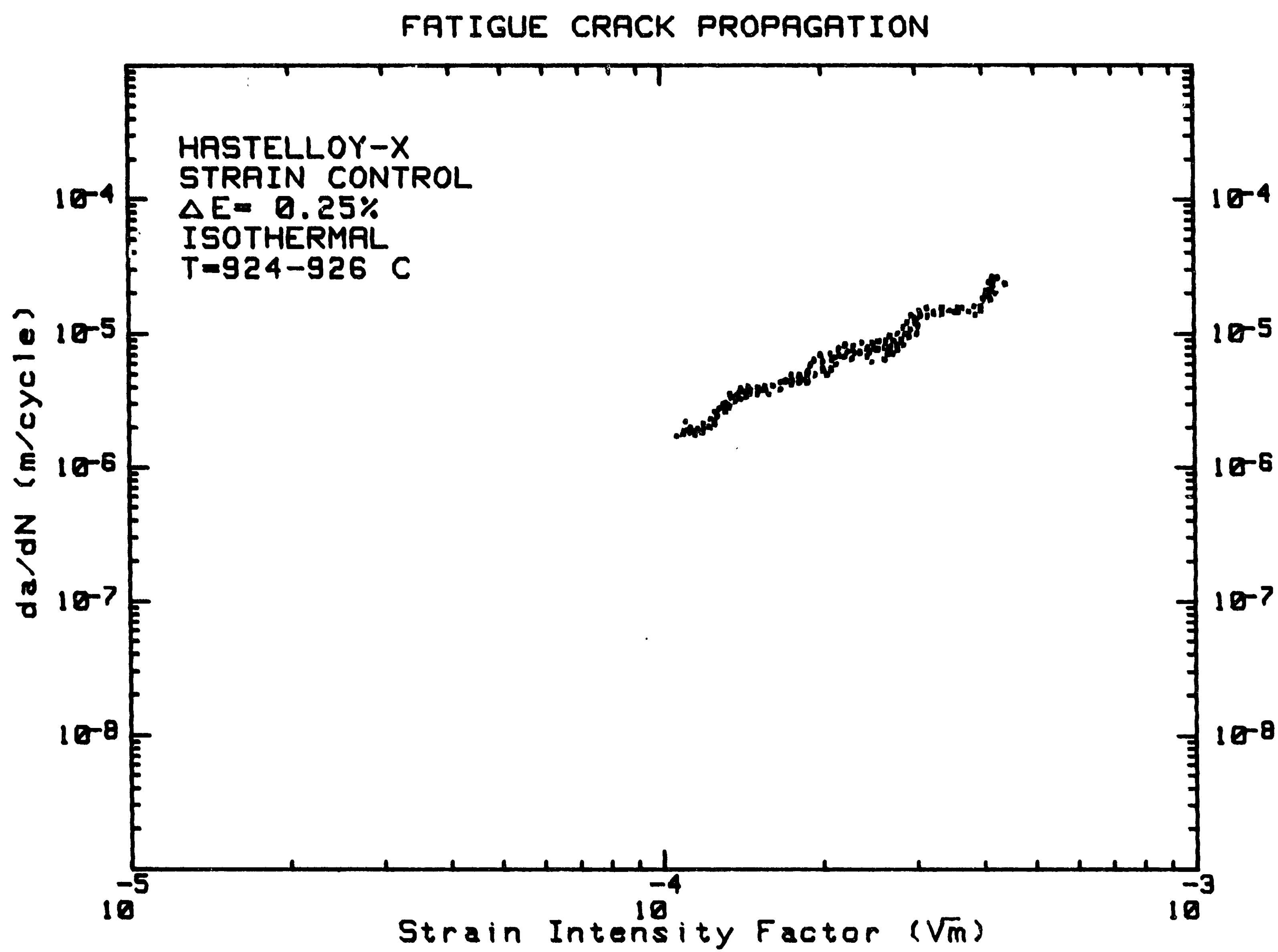
FATIGUE CRACK PROPAGATION
B-1900+Hf
STRAIN CONTROL
ΔE= 0.50 %
IN-PHASE
T= 400-925 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10
20
30
40
50
60
70
80
100
Stress Intensity Factor (MPa√m)

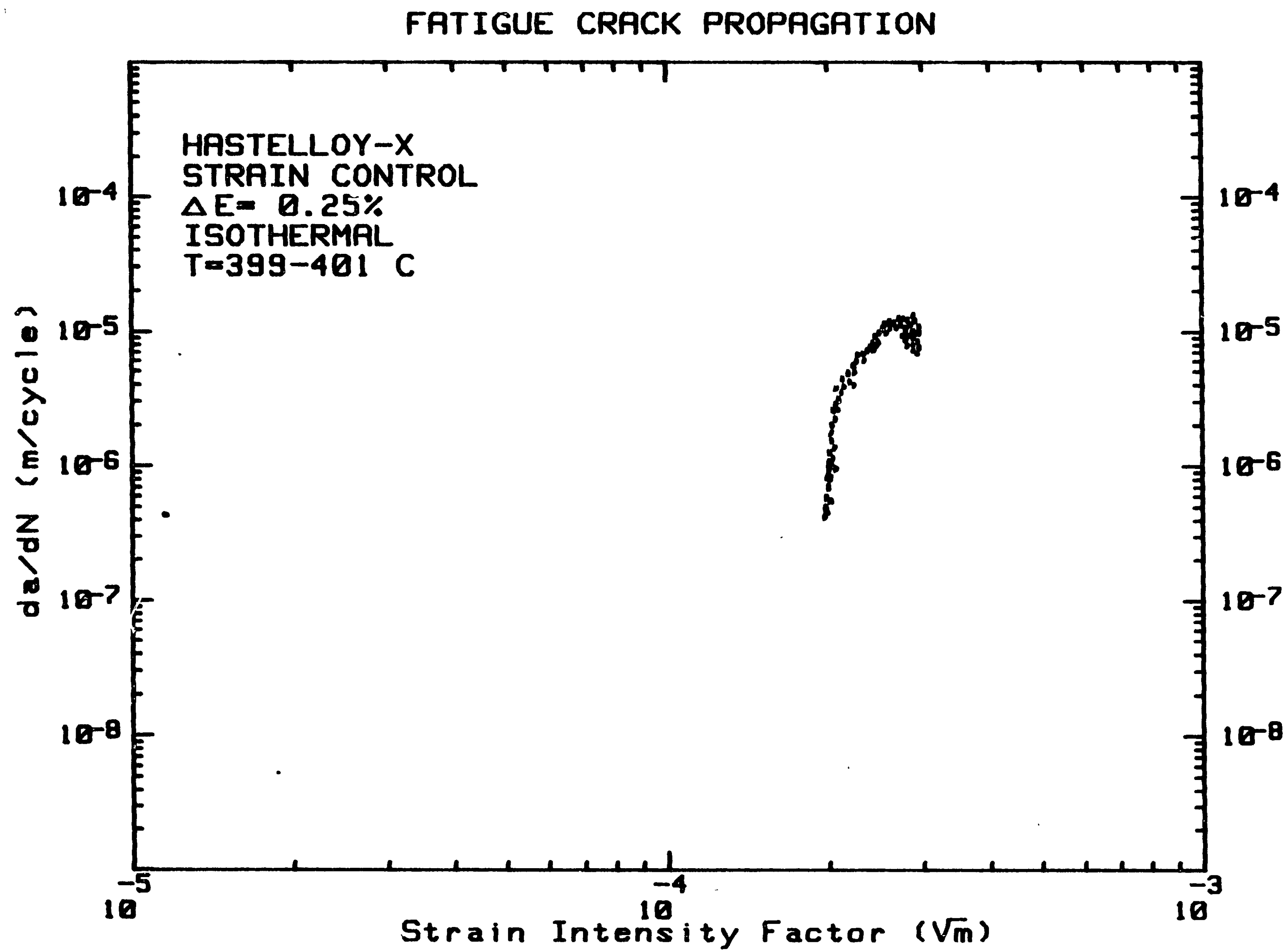

FATIGUE CRACK PROPAGATION
B-1900+Hf
STRAIN CONTROL
ΔE= 0.500%
ISOTHERMAL
T=924-926 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10
20
30
40
50
60
70
80
90
Stress Intensity Factor (MPa√m)


FATIGUE CRACK PROPAGATION
B-1900+Hf
STRAIN CONTROL
ΔE= 0.50%
ISOTHERMAL
T=399-401 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10
20
30
40
50
60
70
80
90
Stress Intensity Factor (MPa√m)

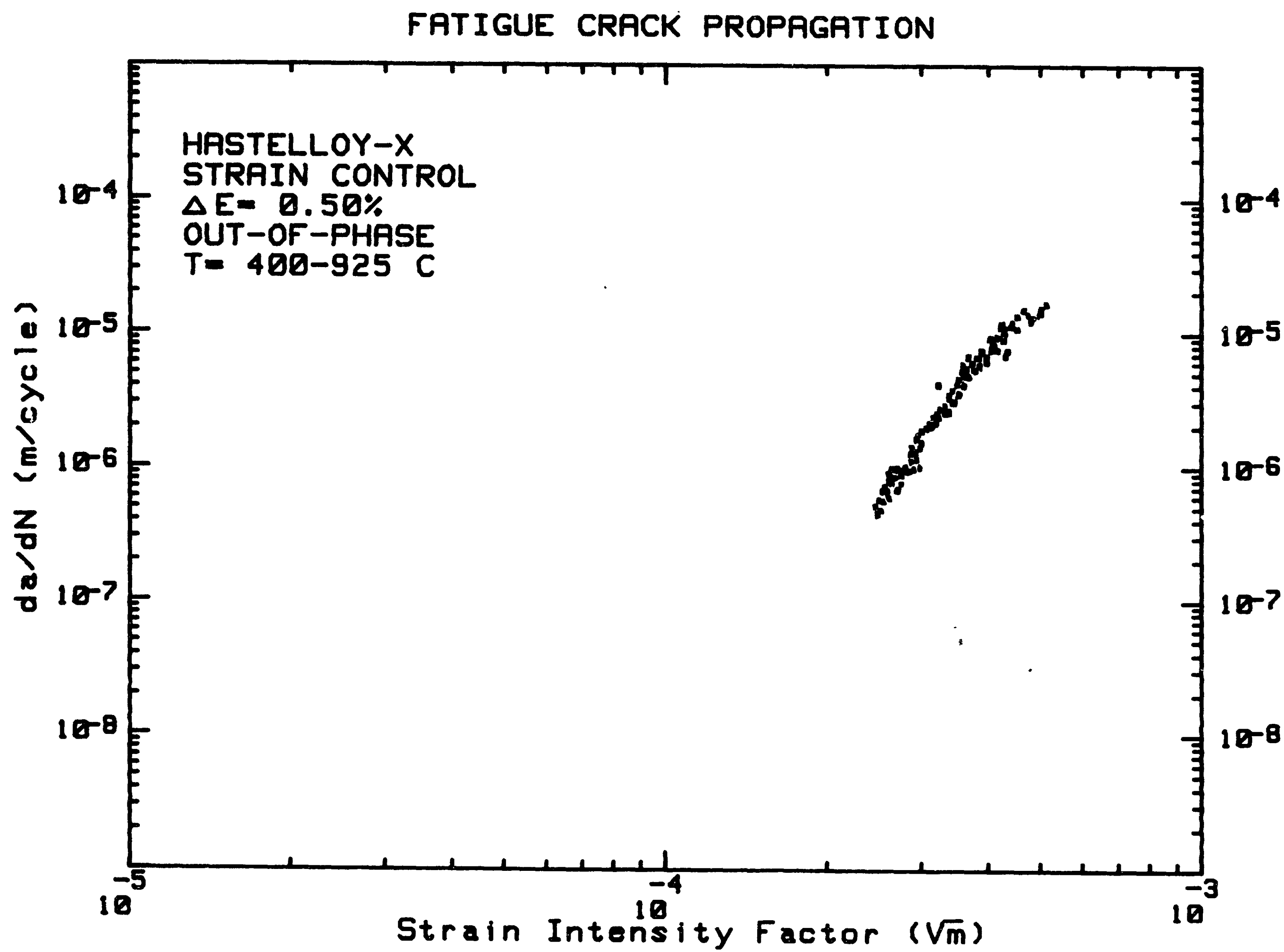

FATIGUE CRACK PROPAGATION
HASTELLOY-X
STRAIN CONTROL
ΔE= 0.25%
OUT-OF-PHASE
T= 400-925 C
da/dN (m/cycle)
Strain Intensity Factor (V̄m)
10-4
10-5
10-6
10-7
10-8
10-5
10-4
10-3

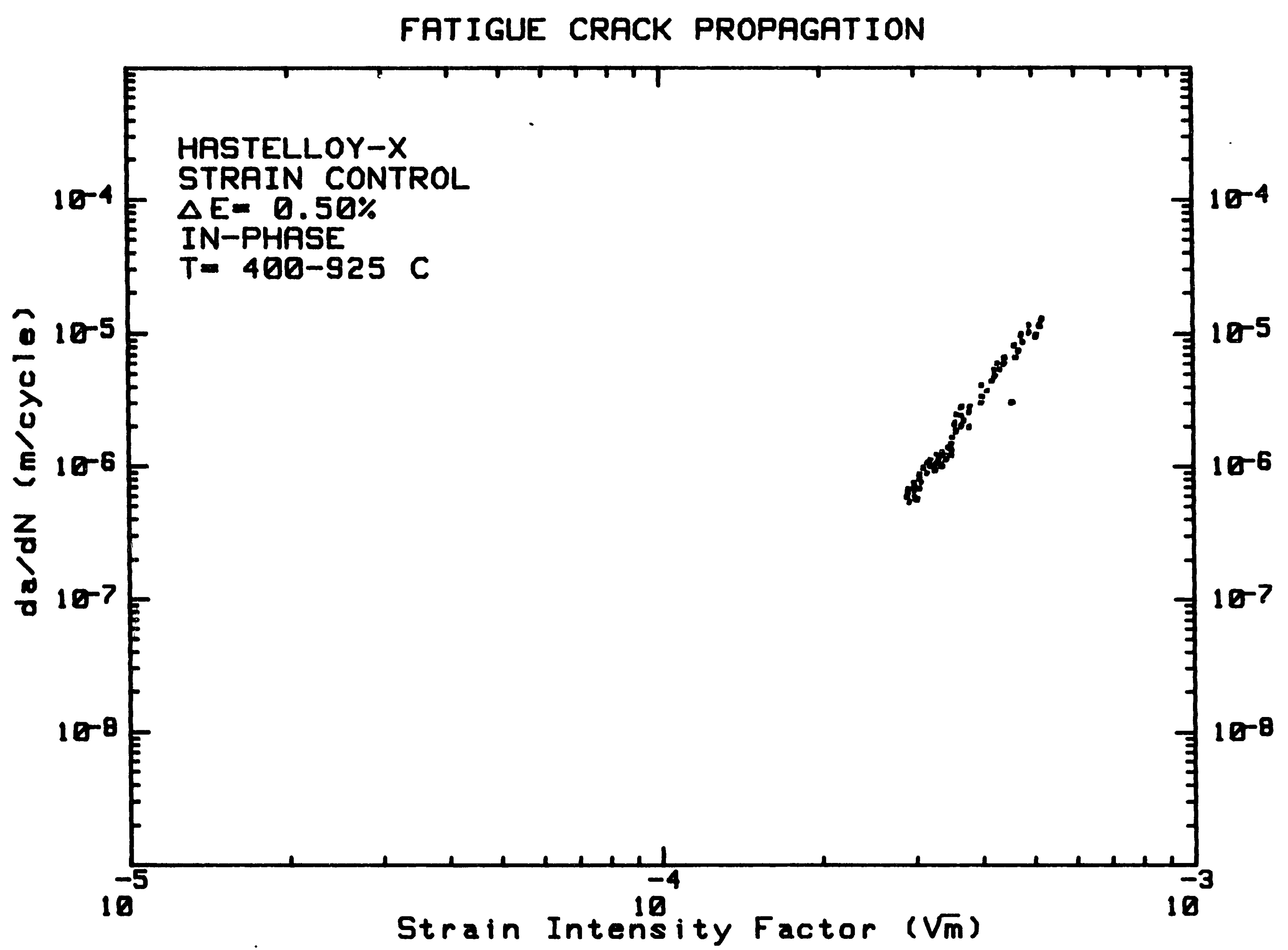

FATIGUE CRACK PROPAGATION
HASTELLOY-X
STRAIN CONTROL
ΔE= 0.25 %
IN-PHASE
T= 400-925 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
Strain Intensity Factor (√m)
10-5
10-4
10-3

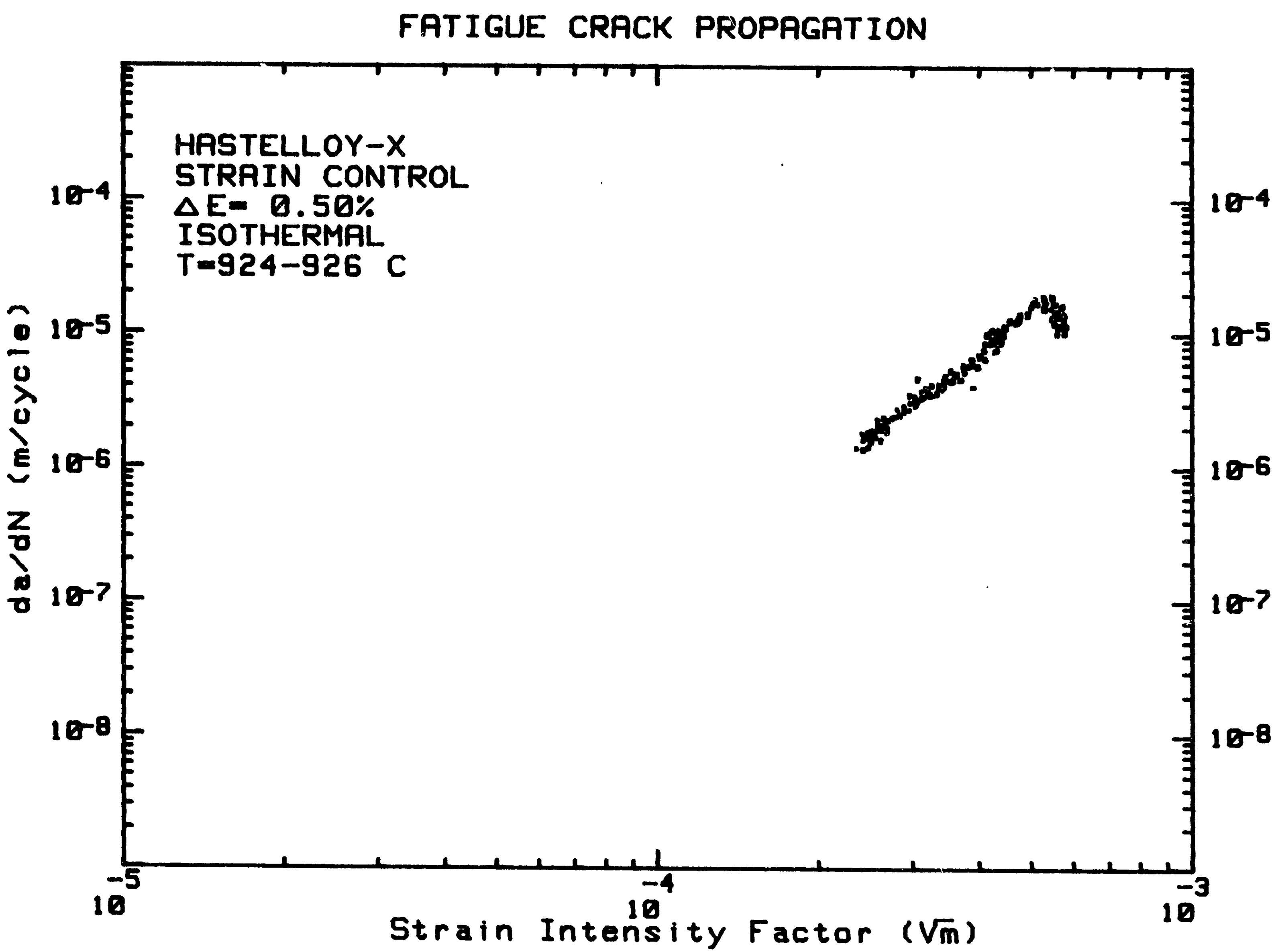

FATIGUE CRACK PROPAGATION
HASTELLOY-X
STRAIN CONTROL
ΔE= 0.25%
ISOTHERMAL
T=924-926 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10-5
10-4
10-3
Strain Intensity Factor (√m)



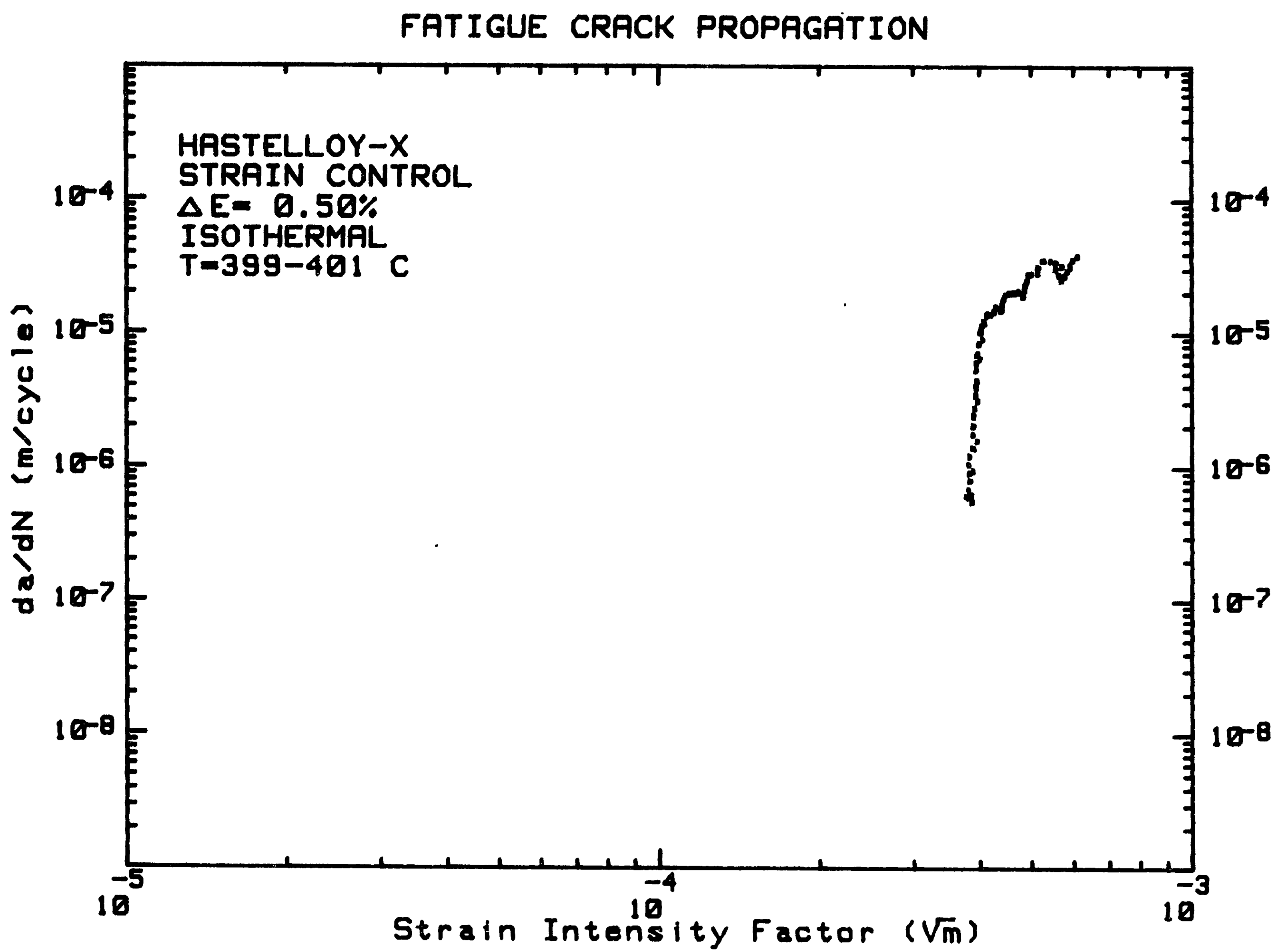
FATIGUE CRACK PROPAGATION
HASTELLOY-X
STRAIN CONTROL
ΔE= 0.25%
ISOTHERMAL
T=399-401 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10-5
10-4
10-3
Strain Intensity Factor (√m)


FATIGUE CRACK PROPAGATION
HASTELLOY-X
STRAIN CONTROL
ΔE= 0.50%
OUT-OF-PHASE
T= 400-925 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10-5
10-4
10-3
Strain Intensity Factor (√m)

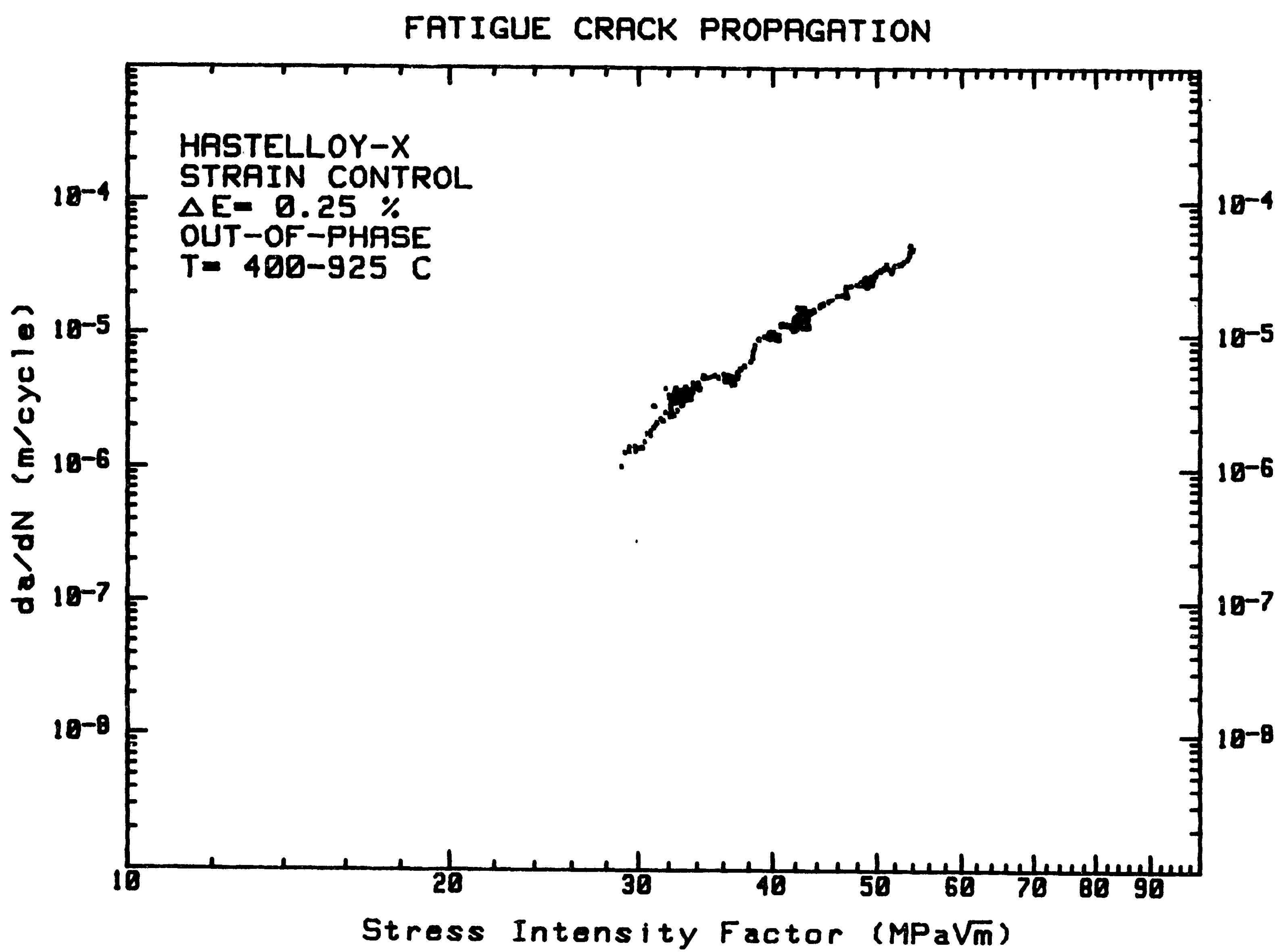

FATIGUE CRACK PROPAGATION
HASTELLOY-X
STRAIN CONTROL
ΔE= 0.50%
IN-PHASE
T= 400-925 C
da/dN (m/cycle)
10^-4
10^-5
10^-6
10^-7
10^-8
10^-5
10^-4
10^-3
Strain Intensity Factor (√m)

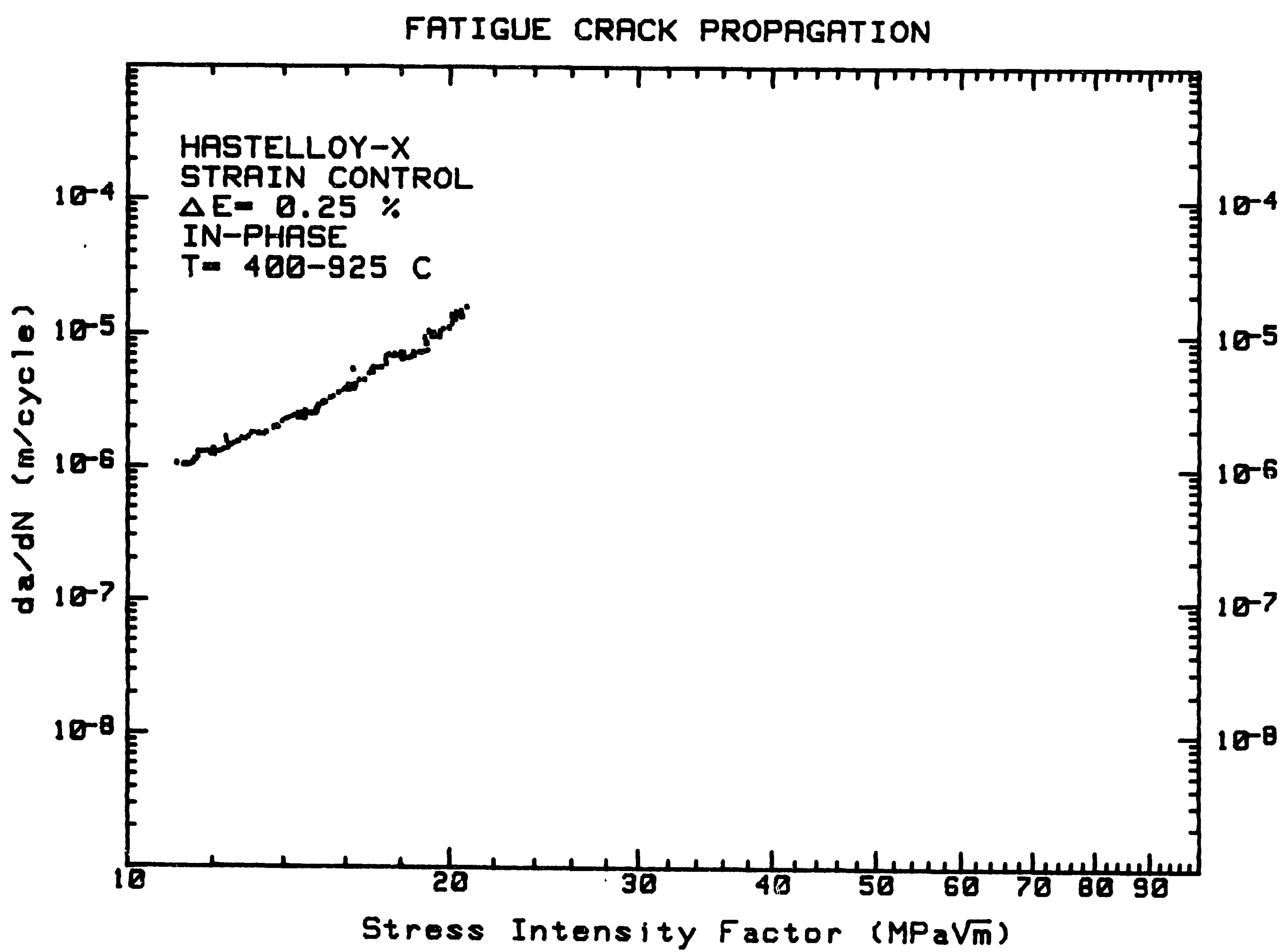


FATIGUE CRACK PROPAGATION
HASTELLOY-X
STRAIN CONTROL
ΔE= 0.50%
ISOTHERMAL
T=924-926 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10-5
10-4
10-3
Strain Intensity Factor (√m)

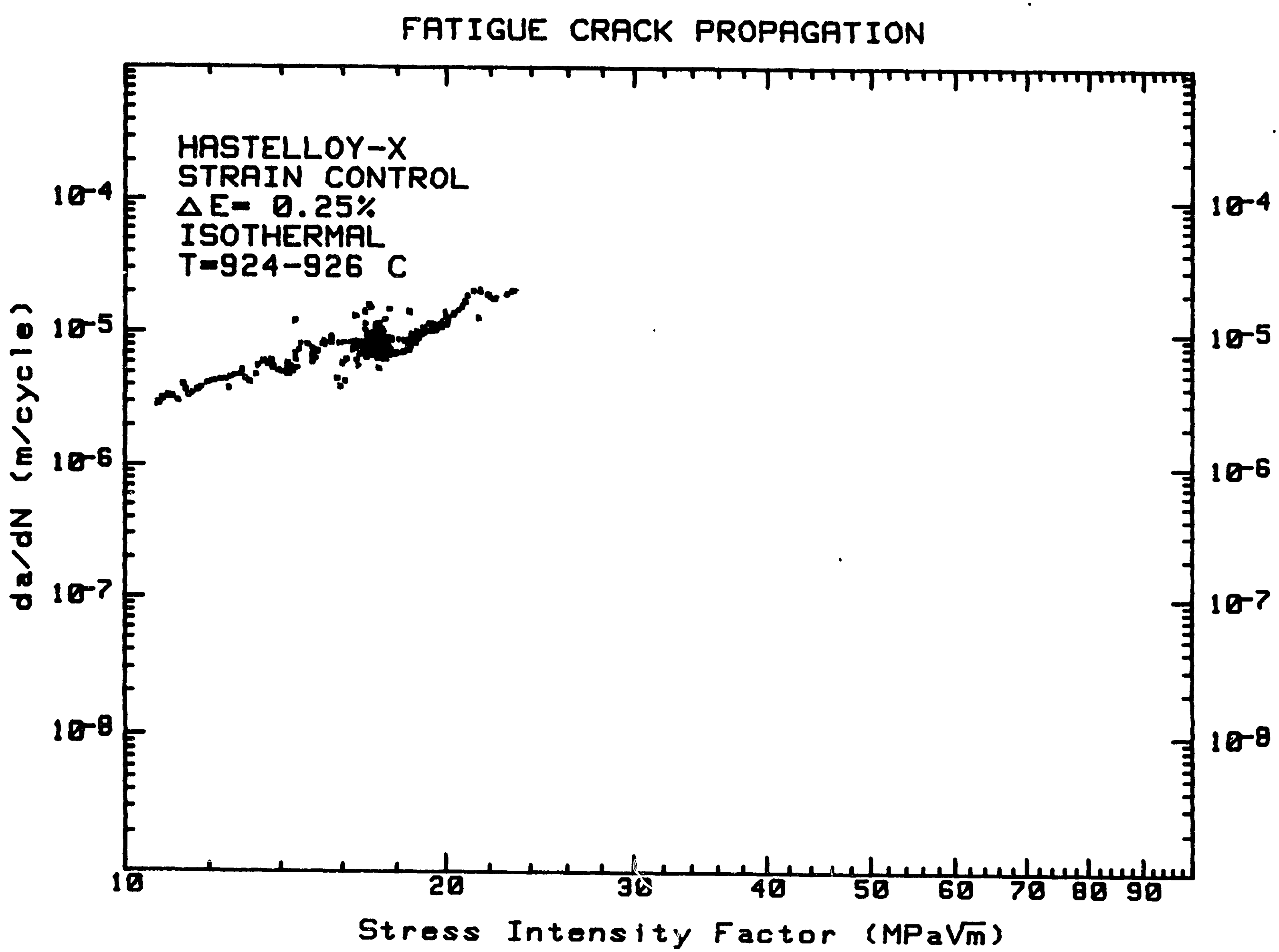


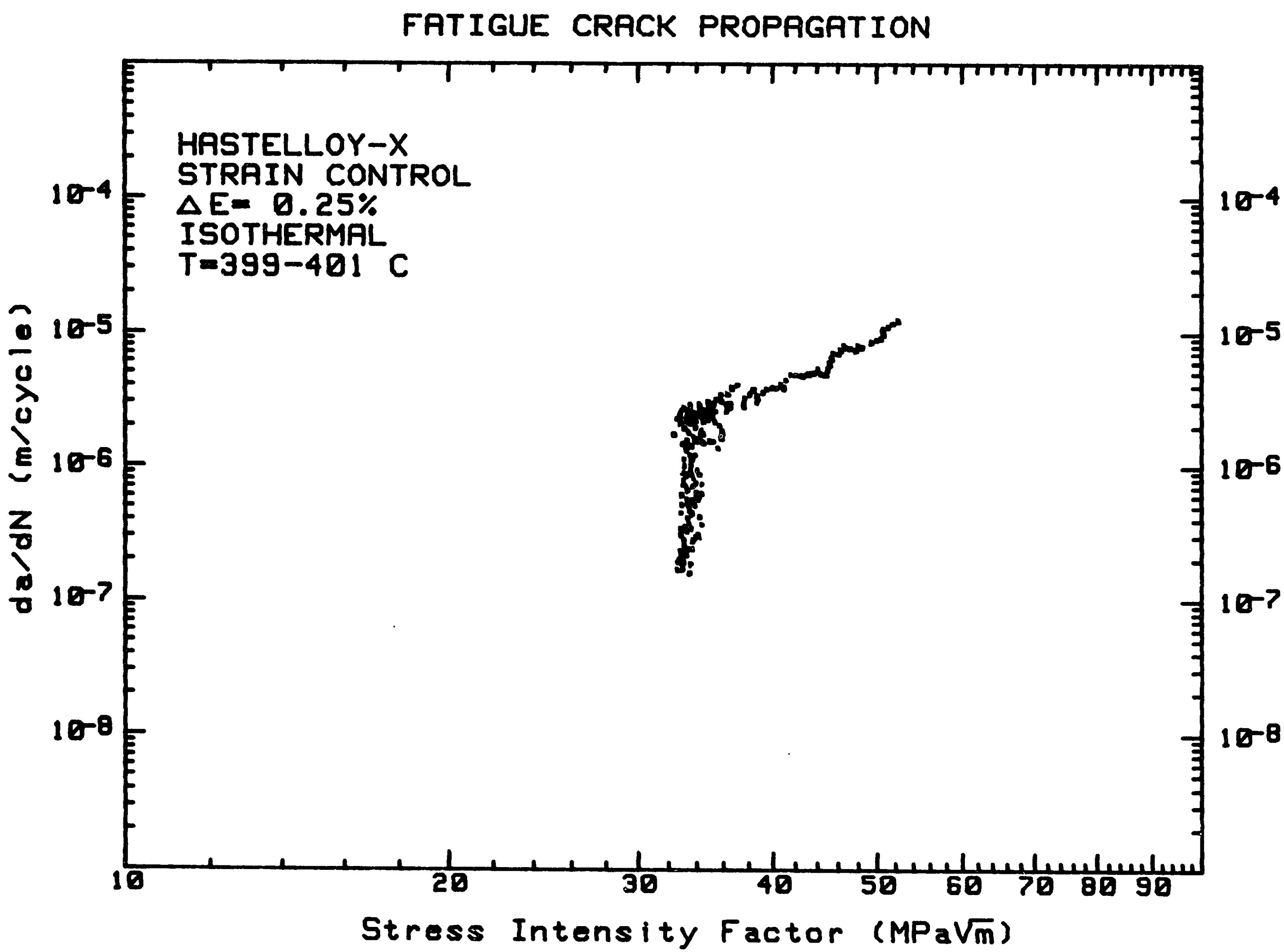
FATIGUE CRACK PROPAGATION
HASTELLOY-X
STRAIN CONTROL
ΔE= 0.50%
ISOTHERMAL
T=399-401 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10-5
10-4
10-3
Strain Intensity Factor (√m)

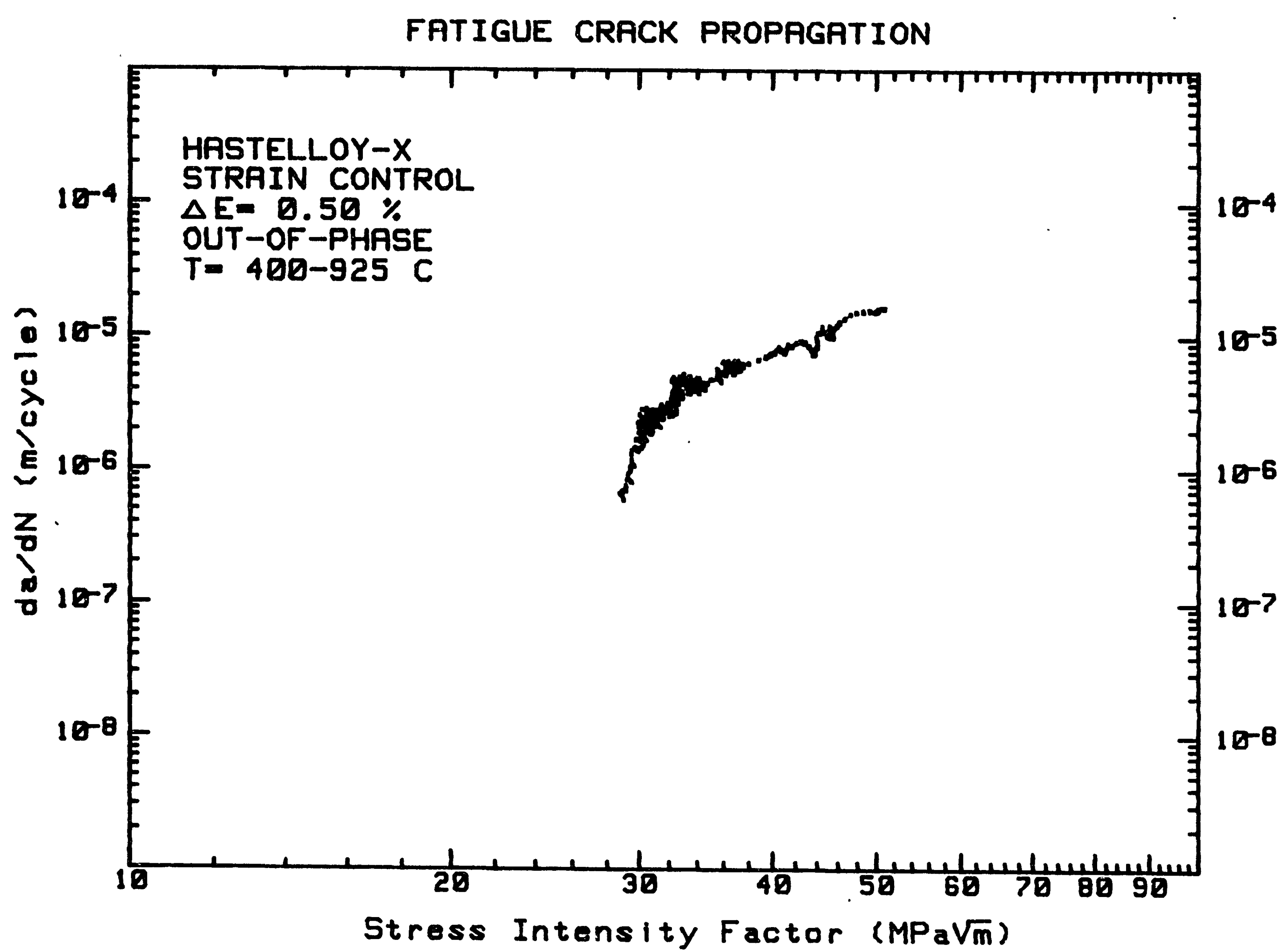

FATIGUE CRACK PROPAGATION
HASTELLOY-X
STRAIN CONTROL
ΔE= 0.25 %
OUT-OF-PHASE
T= 400-925 C
10-4
10-5
10-6
10-7
10-8
da/dN (m/cycle)
10
20
30
40
50
60
70
80
90
Stress Intensity Factor (MPa√m)

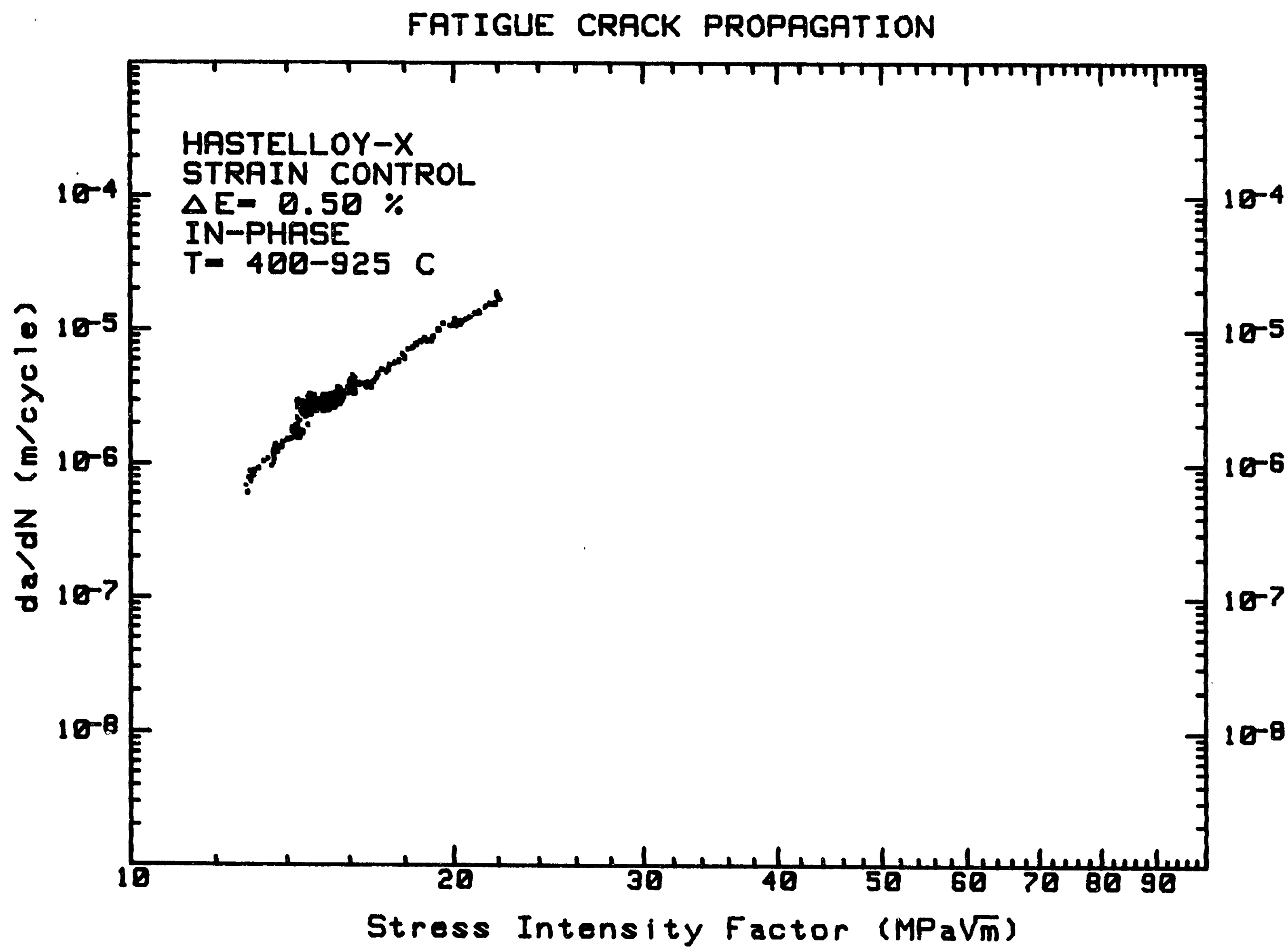


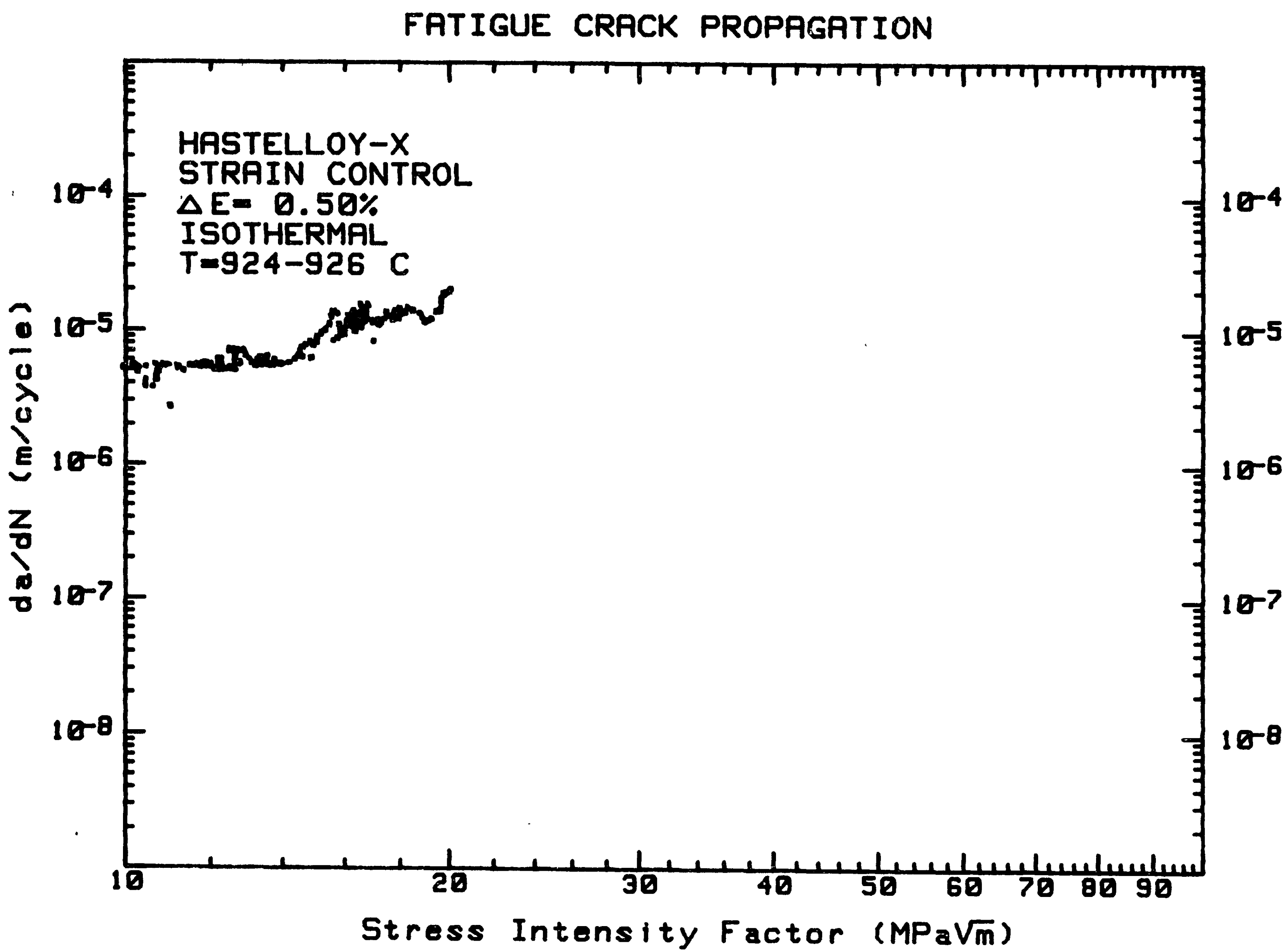
FATIGUE CRACK PROPAGATION
HASTELLOY-X
STRAIN CONTROL
ΔE= 0.25 %
IN-PHASE
T= 400-925 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10
20
30
40
50
60
70
80
90
Stress Intensity Factor (MPa√m)

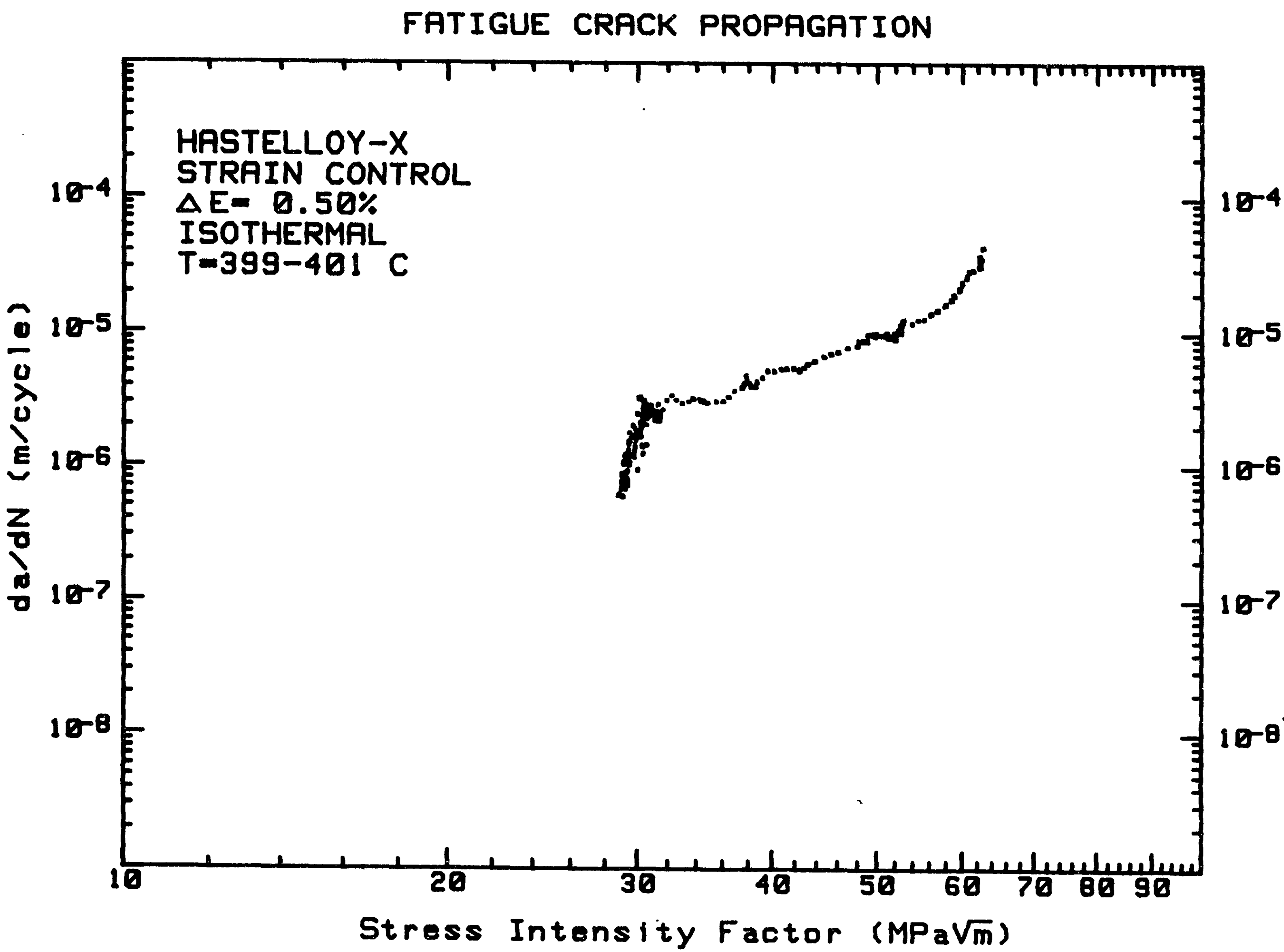

FATIGUE CRACK PROPAGATION
HASTELLOY-X
STRAIN CONTROL
ΔE= 0.25%
ISOTHERMAL
T=924-926 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10
20
30
40
50
60
70
80
90
Stress Intensity Factor (MPa√m)


FATIGUE CRACK PROPAGATION
HASTELLOY-X
STRAIN CONTROL
ΔE= 0.25%
ISOTHERMAL
T=399-401 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10
20
30
40
50
60
70
80
90
Stress Intensity Factor (MPa√m)


FATIGUE CRACK PROPAGATION
HASTELLOY-X
STRAIN CONTROL
ΔE= 0.50 %
OUT-OF-PHASE
T= 400-925 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10
20
30
40
50
60
70
80
90
Stress Intensity Factor (MPa√m)


FATIGUE CRACK PROPAGATION
HASTELLOY-X
STRAIN CONTROL
ΔE= 0.50 %
IN-PHASE
T= 400-925 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10
20
30
40
50
60
70
80
90
Stress Intensity Factor (MPa√m)


FATIGUE CRACK PROPAGATION
HASTELLOY-X
STRAIN CONTROL
ΔE= 0.50%
ISOTHERMAL
T=924-926 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10
20
30
40
50
60
70
80
90
Stress Intensity Factor (MPa√m)


FATIGUE CRACK PROPAGATION
HASTELLOY-X
STRAIN CONTROL
ΔE= 0.50%
ISOTHERMAL
T=399-401 C
da/dN (m/cycle)
10-4
10-5
10-6
10-7
10-8
10
20
30
40
50
60
70
80
90
Stress Intensity Factor (MPa√m)

Appendix V

K_I - Solutions for Single Edge Notch Specimens Under Fixed End Displacements

INTRODUCTION

The single-edge-notch (SEN) geometry with end constraints has received considerable attention in analytical fracture mechanics because of its frequent use as a test specimen. The problem of a SEN specimen with fixed-end displacements has been studied by several authors [1-3]. In particular, the formulation of Harris [2] has received much attention because of its ease of application. However, we felt that the closing bending moments associated with zero far-field relative rotation were perhaps more significant than those implied by Harris' solution, and this premise motivated the present analysis.

ANALYSIS

The presence of a crack in a rectangular sheet of thickness B loaded by uniform imposed end displacements cause the line of action of the applied load to shift relative to the specimen centerline. This shift produces a bending moment that tends to close the crack. For the SEN geometries, the only known boundary conditions are the imposed displacements, and the applied stresses are not known *a priori* and must be determined.

We consider an isotropic linear elastic sub-specimen of length L that is "sufficiently long" compared to crack length (a) and specimen width (W). The specimen is of thickness B and is subject to opposing forces N and moments M as shown schematically in Fig. 1. The line of action of the force N is taken as the mid-specimen. The following considerations are restricted to mode I behavior although the argument can be

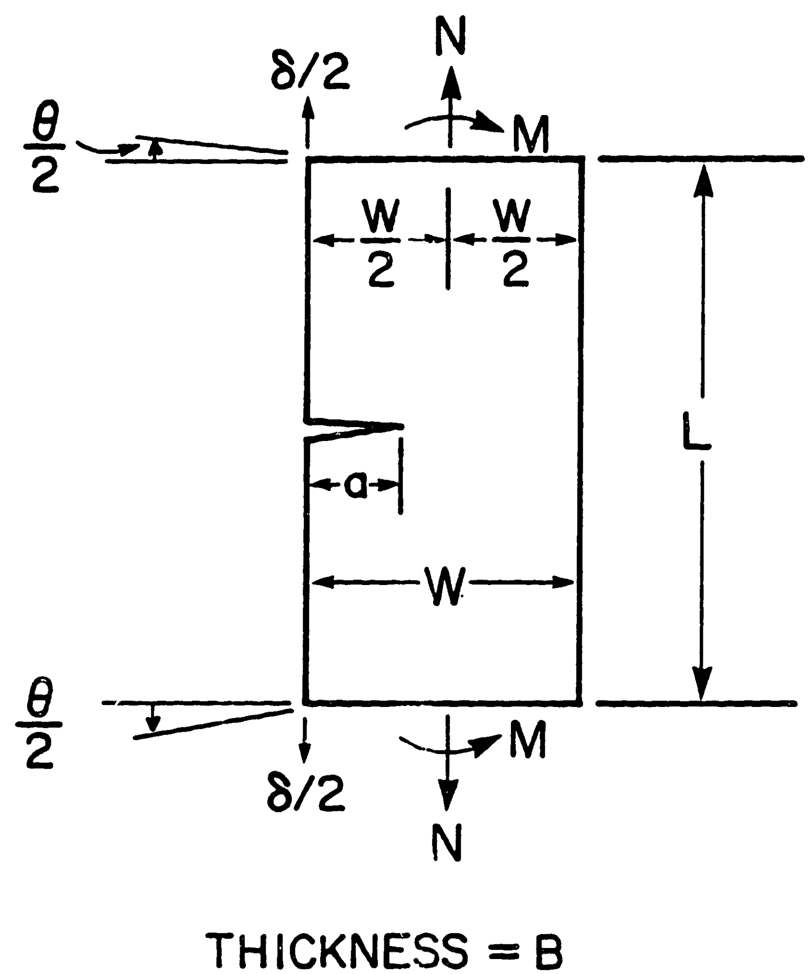


Figure 1. Resultant force N and moment M transmitted by the grips to a SEN specimen.

generalized to include other modes.

The relative displacement δ and rotation θ of the specimen ends can be taken as the sum of a "crack" and "no-crack" parts:

$$\begin{bmatrix} \delta_{tot} \\ \theta_{tot} \end{bmatrix} = \begin{bmatrix} \delta_{c} \\ \theta_{c} \end{bmatrix} + \begin{bmatrix} \delta_{nc} \\ \theta_{nc} \end{bmatrix} . \tag{1}$$

The compliance of the "no-crack" beam gives its extension δ_{nc} and rotation θ_{nc} in terms of the tensile force N acting throughout its center and the corresponding moment M[4]:

$$\begin{bmatrix} \delta_{nc} \\ \theta_{nc} \end{bmatrix} = \begin{bmatrix} L/E'A & 0 \\ 0 & L/E'I \end{bmatrix} \begin{bmatrix} N \\ M \end{bmatrix} , \tag{2}$$

where $E' = E$(Young's modulus) for plane stress and $E/(1-\nu^2)$ for plane strain, ν being the Poisson's ratio. The section moment of inertia $I = BW^3/12$ and the cross section area $A = BW$ can be substituted into Eq. 2 to yield:

$$\begin{bmatrix} \delta_{nc} \\ \theta_{nc} \end{bmatrix} = \begin{bmatrix} L/E'BW & 0 \\ 0 & 12L/E'BW^3 \end{bmatrix} \begin{bmatrix} N \\ M \end{bmatrix} \tag{3}$$

The expression for the "cracked" terms are obtained by considering the complementary energy of the specimen U in terms of N and M,

$$U(N,M) = \frac{1}{2}\,[N\ M]\left[\begin{bmatrix}\delta_{nc}\\ \theta_{nc}\end{bmatrix} + \begin{bmatrix}\delta_c\\ \theta_c\end{bmatrix}\right] , \qquad (4)$$

where the matrix scalar product is indicated. Since the "no-crack" terms given in Eq. 3 are independent of crack length, the energy release rate is

$$\frac{\partial U}{\partial a} = \frac{1}{2}\,[N\ M] \cdot \begin{bmatrix}\partial\delta_c/\partial a\\ \partial\theta_c/\partial a\end{bmatrix} . \qquad (5)$$

With the standard relation between G (fracture mechanics energy release rate) and K_I, the change in complementary energy with respect to crack length a is equal to [5]:

$$\frac{\partial U}{\partial a} = B\,\frac{K_I^2}{E'} . \qquad (6)$$

The stress intensity factor for combined tension and bending can be obtained by superposition of the stress intensity factors applicable to tension and bending. Therefore, in matrix form

$$K_I = \begin{bmatrix}\frac{\partial K}{\partial N} & \frac{\partial K}{\partial M}\end{bmatrix}\begin{bmatrix}N\\ M\end{bmatrix} . \qquad (7)$$

Highly accurate functional forms of $\partial K/\partial N$ and $\partial K/\partial M$ can be obtained from the compilations of Tada et. al. [5] where, for example

$$\frac{\partial K}{\partial N} = \frac{(\pi a)^{1/2}}{BW} \sqrt{\frac{2W}{\pi a} \tan(\frac{\pi a}{2W})} \left[\frac{0.752+2.02(\frac{a}{W}) + 0.37(1-\sin(\frac{\pi a}{2W}))^3}{\cos(\frac{\pi a}{2W})} \right] \tag{8.1a}$$

$$\equiv (\pi a)^{1/2} \cdot F_1(\frac{a}{W})/BW , \tag{8.1b}$$

$$\frac{\partial K}{\partial M} = \frac{6(\pi a)^{1/2}}{BW^2} \sqrt{\frac{2W}{\pi a} \tan(\frac{\pi a}{2W})} \left[\frac{0.923 + 0.199 \ (1-\sin(\frac{\pi a}{2W}))^4}{\cos(\frac{\pi a}{2W})} \right] \tag{8.2a}$$

$$\equiv (\pi a)^{1/2} \cdot F_2(a/W) \cdot \frac{6}{BW^2} \quad . \tag{8.2b}$$

Note that in Eq. (8.1b, 8.2b), alternative non-dimensional function F_1, F_2 are introduced in such a manner as to facilitate K-calibrations in terms of a nominal stress "σ" as $K_I = F \cdot \sigma \cdot (\pi a)^{1/2}$. For tension, $\sigma = N/BW$, while for bending $\sigma = 6M/BW^2$. Now equating Eq. 5 and Eq. 6 and inserting Eq. 7 into Eq. 6, one obtains

$$\frac{1}{2} \begin{bmatrix} \frac{\partial \delta_c}{\partial a} \\ \\ \frac{\partial \theta_c}{\partial a} \end{bmatrix} = \frac{B}{E'} \begin{bmatrix} (\frac{\partial K}{\partial N})^2 & (\frac{\partial K}{\partial N})(\frac{\partial K}{\partial M}) \\ \\ (\frac{\partial K}{\partial N})(\frac{\partial K}{\partial M}) & (\frac{\partial K}{\partial M})^2 \end{bmatrix} \begin{bmatrix} N \\ M \end{bmatrix} . \tag{9}$$

Integrating Eq. 9 with respect to "a" provides the "crack" compliance matrix:

$$\begin{bmatrix} \delta_c \\ \theta_c \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} N \\ M \end{bmatrix} \quad , \tag{10}$$

where

$$C_{11} = \frac{2B}{E'} \int_0^a \left[\left(\frac{\partial K(a')}{\partial N} \right) \right]^2 da' = \frac{\partial \delta_c}{\partial N} \tag{11.1}$$

$$C_{12} = C_{21} = \frac{2B}{E'} \int_0^a \left(\frac{\partial K(a')}{\partial N} \right) \left(\frac{\partial K(a')}{\partial M} \right) da' = \frac{\partial \delta_c}{\partial M} = \frac{\partial \theta_c}{\partial N} \tag{11.2}$$

$$C_{22} = \frac{2B}{E'} \int_0^a \left[\left(\frac{\partial K(a')}{\partial M} \right) \right]^2 da' = \frac{\partial \theta_c}{\partial M} \; . \tag{11.3}$$

The generalized forces N, M can now be evaluated in terms of imposed displacements δ_{tot}, θ_{tot} by using Eqs. 1, 3 and 10, providing

$$\theta_{tot} = \frac{12L}{E'BW^3} M + C_{22}M + C_{21}N \tag{12.1}$$

$$\delta_{tot} = \frac{L}{E'BW} N + C_{11}N + C_{21}M \quad , \tag{12.2}$$

or, on multiplying by E'B;

$$E'B\theta_{tot} = \frac{12L}{W^3} M + E'BC_{22}M + E'BC_{21}N \tag{13.1}$$

$$E'B\delta_{tot} = \frac{L}{W} N + E'BC_{11}N + E'BC_{21}M. \tag{13.2}$$

For fixed-end displacement with no shear force $\theta_{tot} = 0$ and $\delta_{tot} = \delta$. Then from Eq. 13.1 we get

$$\frac{M}{W} = - \left[\frac{E'BWC_{12}}{12(L/W)+EBW^2C_{22}} \right] N \ , \tag{14}$$

while Eq. 13.2 can be re-written

$$E'B\delta = [\frac{L}{W} + E'BC_{11}]\ N + E'BWC_{21}\ (\frac{M}{W})\ . \qquad (15)$$

On defining the dimensionless cracked compliance $\hat{C}_{ij}$ as

$$\hat{C}_{11} = E'BC_{11} \qquad (16.1)$$

$$\hat{C}_{22} = E'BW^2C_{22} \qquad (16.2)$$

$$\hat{C}_{12} = \hat{C}_{21} = E'BWC_{12} = E'BWC_{21}\ , \qquad (16.3)$$

Eq. 14 simplifies to

$$\frac{M}{W} = \frac{-\hat{C}_{12}}{12(\frac{L}{W}) + \hat{C}_{22}} \cdot N \qquad (17)$$

while Eq. 15 simplifies to

$$E'B\delta = (\frac{L}{W} + \hat{C}_{11})\ N + \hat{C}_{12}\ \frac{M}{W}\ . \qquad (18)$$

When Eq. 17 is inserted into Eq. 18, the axial force N can be obtained as

$$N = E'BW \frac{\delta}{L} \cdot \frac{1}{\{1 + \frac{W}{L}[\hat{C}_{11} - \frac{(\hat{C}_{12})^2}{(12(\frac{L}{W}) + \hat{C}_{22})}]\}} \quad . \tag{19}$$

The substitution of Eq. 19 into Eq. 17 provides

$$\frac{M}{W} = \frac{E'BW\delta}{L} \cdot \frac{\left\{ \frac{-\hat{C}_{12}}{(12(L/W) + \hat{C}_{22})} \right\}}{\left\{ 1 + \frac{W}{L}\left[\hat{C}_{11} - \frac{(\hat{C}_{12})^2}{(12(L/W)+\hat{C}_{22})} \right] \right\}} \quad . \tag{20}$$

Eq. 19 and 20, respectively provide the force N and bending moment M applied to the crack in terms of the imposed relative displacement δ under conditions of zero relative rotation. Note from Eq. 20 that, since the $\hat{C}_{ij}$ are inherently non-negative, the sign of M is indeed negative, tending to close the crack.

The total stress intensity factor for this specimen is, by superposition, the sum of that due to tension and that due to bending. Due to linearity,

$$K_I = \frac{\partial K}{\partial N} \cdot N + \frac{\partial K}{\partial M} \cdot M \quad , \tag{21}$$

and on combining Eqs. (8.1b, 8.2b, 19, 20) we obtain

$$K_I = \frac{E'\delta}{L} \cdot (\pi a)^{1/2} \cdot \frac{F_1(\xi)\left\{1-6 \cdot \dfrac{F_2(\xi)}{F_1(\xi)} \cdot \dfrac{\hat{C}_{12}(\xi)}{12\left(\frac{L}{W}\right) + \hat{C}_{22}(\xi)}\right\}}{\left\{1 + \frac{W}{L}\left[\hat{C}_{11}(\xi) - \dfrac{\hat{C}_{12}^{2}(\xi)}{12\left(\frac{L}{W}\right) + \hat{C}_{22}(\xi)}\right]\right\}} , \tag{22}$$

where $\xi = a/W$. As shown in [6], the terms $\hat{C}_{ij}(\xi)$ are normalized dimensionless cracked compliances which need be calculated only once. In the present application, the integrals of Eq. 11 were numerically evaluated in increments of dimensionless crack length $\Delta\xi = \Delta a/W$ of size 0.01 using Simpson's rule. If we now substitute the nominal stress value $\sigma \equiv E'\delta/L$ into Eq. 22, we recover the familiar form

$$K_I = \sigma\sqrt{\pi a} \cdot G(\xi,\eta) \ , \tag{23}$$

where

$$G(\xi,\eta) = \frac{F_1(\xi) \cdot \left\{1 - \dfrac{6F_2(\xi)\cdot\hat{C}_{12}(\xi)}{F_1(\xi)[12\eta + \hat{C}_{22}(\xi)]}\right\}}{\left\{1 + \dfrac{1}{\eta}\left[\hat{C}_{11}(\xi) - \dfrac{\hat{C}_{12}(\xi)^2}{(12\eta + \hat{C}_{22}(\xi))}\right]\right\}} \ , \tag{24}$$

and $\xi = a/W$, $\eta = L/W$.

In Fig. 2, plots of $G(\xi, \eta)$ versus a/W are given for different values of L/W. The dimensionless functions F_1 and F_2 of Tada et. al. [5] (see Eqs. 8.1b, 8.2b) for pure tension and pure bending, respectively, are also shown for reference purposes. Finally, Fig. 2 also shows the approximate stress intensity factor calibration of the SEN specimen due to Harris [2], who gives

$$K_I = \sigma\sqrt{\pi a}\ F_H(a/W) \tag{25a}$$

with

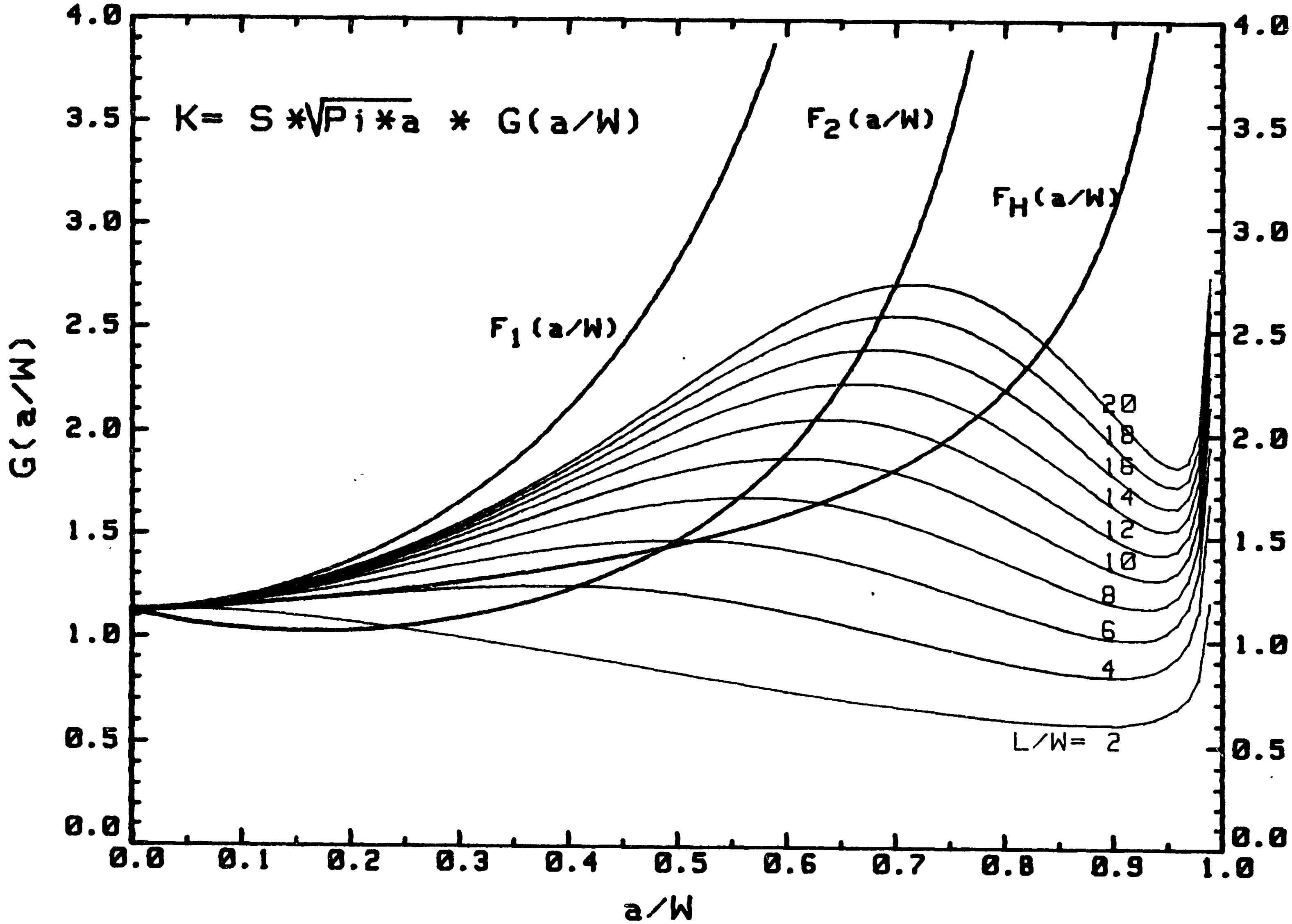


Figure 2. Geometrical factors F_1, F_2, F_H and Eq. 24 as a function of the dimensionless crack lenght.

$$F_H(a/W) = \left\{\frac{25}{20-13(a/W)-7(a/W)^2}\right\}^{1/2} . \qquad (25b)$$

In the interpretation of Eq. 25a, the nominal stress "σ" is understood to be N/BW, the nominal far-field stress, which in the present application must be determined (analysis) or measured (experiment) in terms of the imposed loading parameter δ.

As can be seen, our solution for G is strongly dependent on both relative crack depth and specimen length-to-width ratio and shows a local maximum with respect to a/W at some intermediate value of a/W for all (L/W) ratios larger than 2. For smaller ratios, the geometric correction factor G shows a monotonic decrease with respect to a/W until very deep cracks, a/W > .9, are considered. Tada's and Harris' geometry correction factors which were derived for tension, F_1 and F_H, under conditions of constant remote uniform tensile stress show a monotone increase with increasing a/W. Tada's bending geometry correction factor F_2 shows a slight decrease with increasing a/W, reaching a minimum value near a/W = 0.15, followed by a monotonic increase.

A basic assumption of the preceding analysis is that the specimen length (L) at which displacement boundary conditions are being imposed is sufficiently long in comparison to the appropriate St. Venant decay distance for the local stress disturbance introduced by the presence of the crack. For short cracks, $a/W << 1$, the characteristic decay distance is "a", so that providing $L/W > 1$, the analysis should be valid. For very deep cracks, the decay length is specimen width W, so L must in that case exceed some multiple of W. This analysis cannot precisely quantify a requisite minimum value of L/W for any particular maximum value of a/W. However, good agreement has been obtained with Bowie and Freese's solution [3] for low values of $L/W = 2$, the agreement between the two solutions was within one percent for all values of a/W covered by the analysis.

EXPERIMENTAL VERIFICATION

An experimental check of the solution was provided by performing fatigue crack growth tests on SEN specimens subject to imposed displacement loading. The dimensions of the specimen are shown in Fig. 3. The notch, about 1 mm deep, was cut by electro-discharge machining. The test material was B-1900+Hf, a high-strength/low ductility superalloy with good creep and oxidation resistance at elevated temperature. The tests were run under fully reversed strain control condition in the elastic regime ($\Delta\varepsilon = 0.25\%$) in laboratory air. This nominal axial strain was controlled over a gauge length of 12.7 mm which included the crack. The temperature and frequency of the tests were 925° C and 0.1 Hz. Before starting the tests, the specimens were precracked at 10 Hz and room temperature under load-controlled conditions up to a ΔK of about 20 $MPa\sqrt{m}$ (Harris' solution) which corresponds to an a/W ratio of 0.125. Details of the experimental set-up and crack growth measurement technique are given elsewhere [7,8].

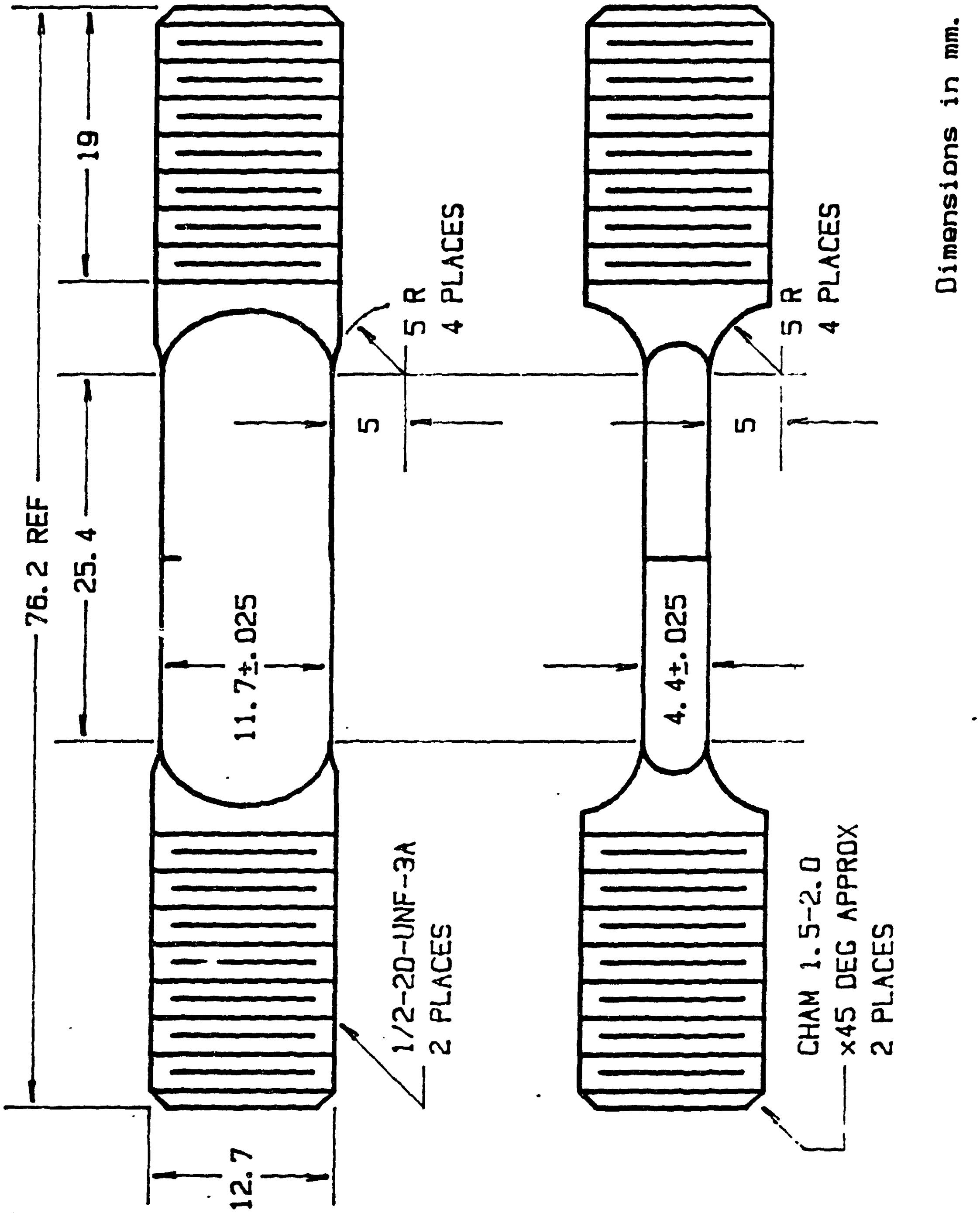


Figure 3. Single edge notched (SEN) specimen used for the testing (notch: 1 mm deep x 0.025 mm wide).

Fig. 4 shows the crack growth rates as a function of the strain intensity factor ΔK_ε. The ΔK_ε is used for convenience because fatigue at high temperature is a strain-controlled process. It is defined as follows [9,10]:

$$\Delta K_\varepsilon = \Delta\varepsilon \cdot \sqrt{\pi a} \cdot G(a/W) \ . \qquad (26)$$

In the above expression, G (a/W) is the same geometric correction term derived in connection with the stress intensity factor. That is, in Fig. 4, the strain intensity is calculated as $\Delta\varepsilon \cdot \sqrt{\pi a} \cdot F_H(a/W)$ with $\Delta\varepsilon = .25\%$. In spite of the fact that ΔK_ε, so-defined, lacks a rigorous mechanics interpretation, it has been used for correlating crack growth data under strain-controlled conditions [9-12]. Clearly, the non-monotone correlation of fatigue crack growth rate with strain intensity factor range is not to be expected in cracks of macroscopic dimension. Thus, the non-monotonic nature of this correlation strongly suggests that there are deficiencies in this analysis of the strain intensity factor.

To provide a more rigorous understanding of the result, the ΔK_ε were re-defined as

$$\Delta K_\varepsilon = \Delta K/E' \qquad (27)$$

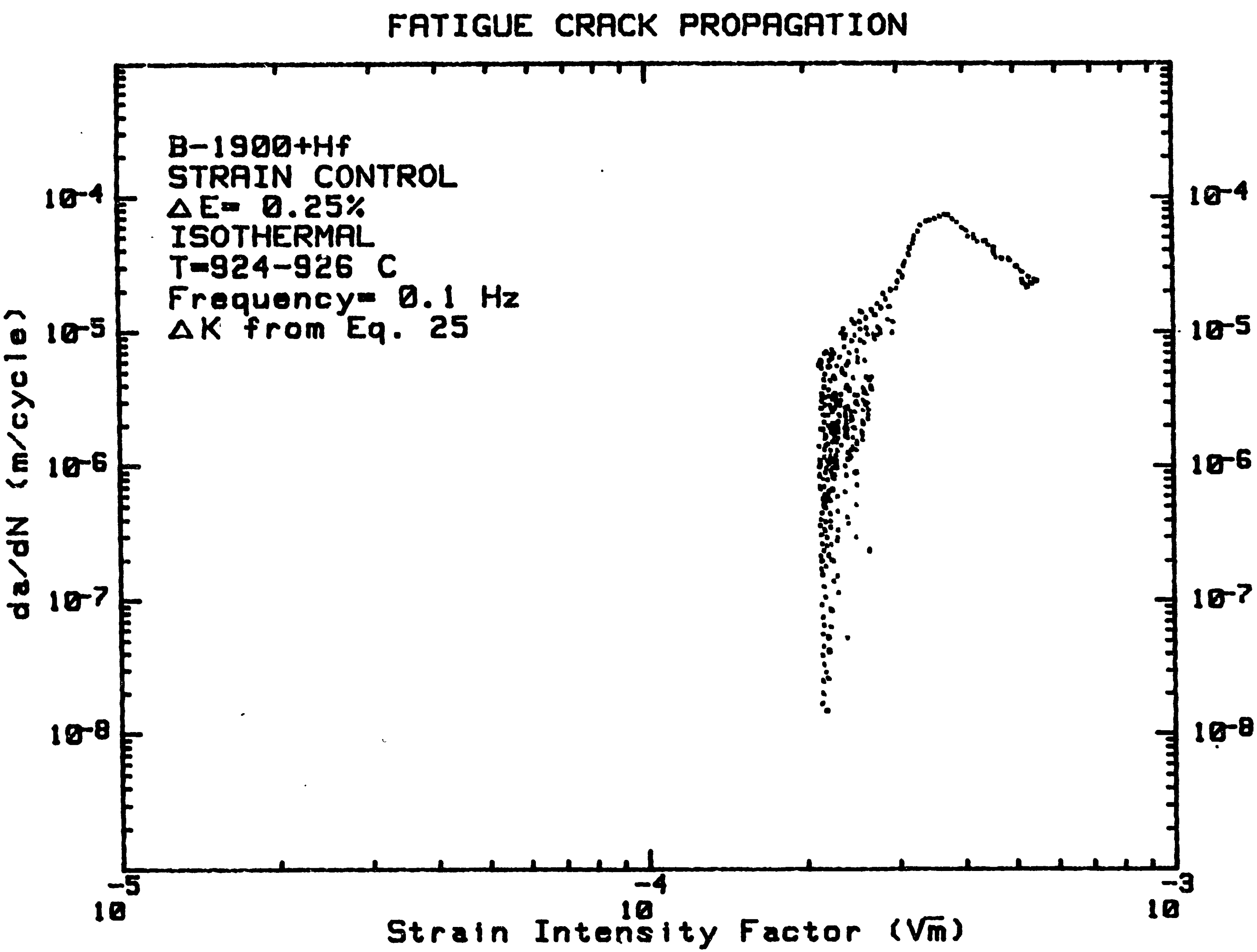


Figure 4. FCP rates as a function of the strain intensity factor calculated using Eq. 22.

where the ΔK is the stress intensity factor derived for stress-controlled (Eq. 25) or displacement-controlled (Eq. 23) conditions. In cases of predominantly elastic behavior, this definition of strain intensity factor is consistent with standard fracture mechanics. Fig.5 re-plots the fatigue crack growth rate versus the cyclic strain intensity factor deduced from Eq. 25 and from the current analysis (Eq. 23) using the value of $L/W = 2.17$ corresponding to the length of the uniform reduced gauge section of the specimen. In the application of Harris' formulation, the nominal stress σ was calculated using the tensile force N measured by the load cell. It should be noted that the load cell output is insensitive to the shift in load line which is associated with longer cracks. Again, the curve indicates that the correlation between measured fatigue crack growth rate and inferred strain intensity factor is non-monotone. Note, however, that the correlation is somewhat better than that shown in Fig. 4 because the inferred range of ΔK_ε occurring during the tests is reduced. The lower inferred range in the present case is due to the use of the measured load value, which decreases under fixed displacement conditions, due to the increasing crack compliance. This load-shedding is not reflected in the previous definition of ΔK_ε as $\Delta\varepsilon \cdot \sqrt{\pi a} \cdot F_H(a/W)$. The importance of load-shedding in the determination of the actual driving force has been recognized by Leis et. al. [13-14] which also used the measured load in their K calculations.

At this point, it is useful to consider more carefully the mathematical model of an imposed displacement with no rotation over a length "L"

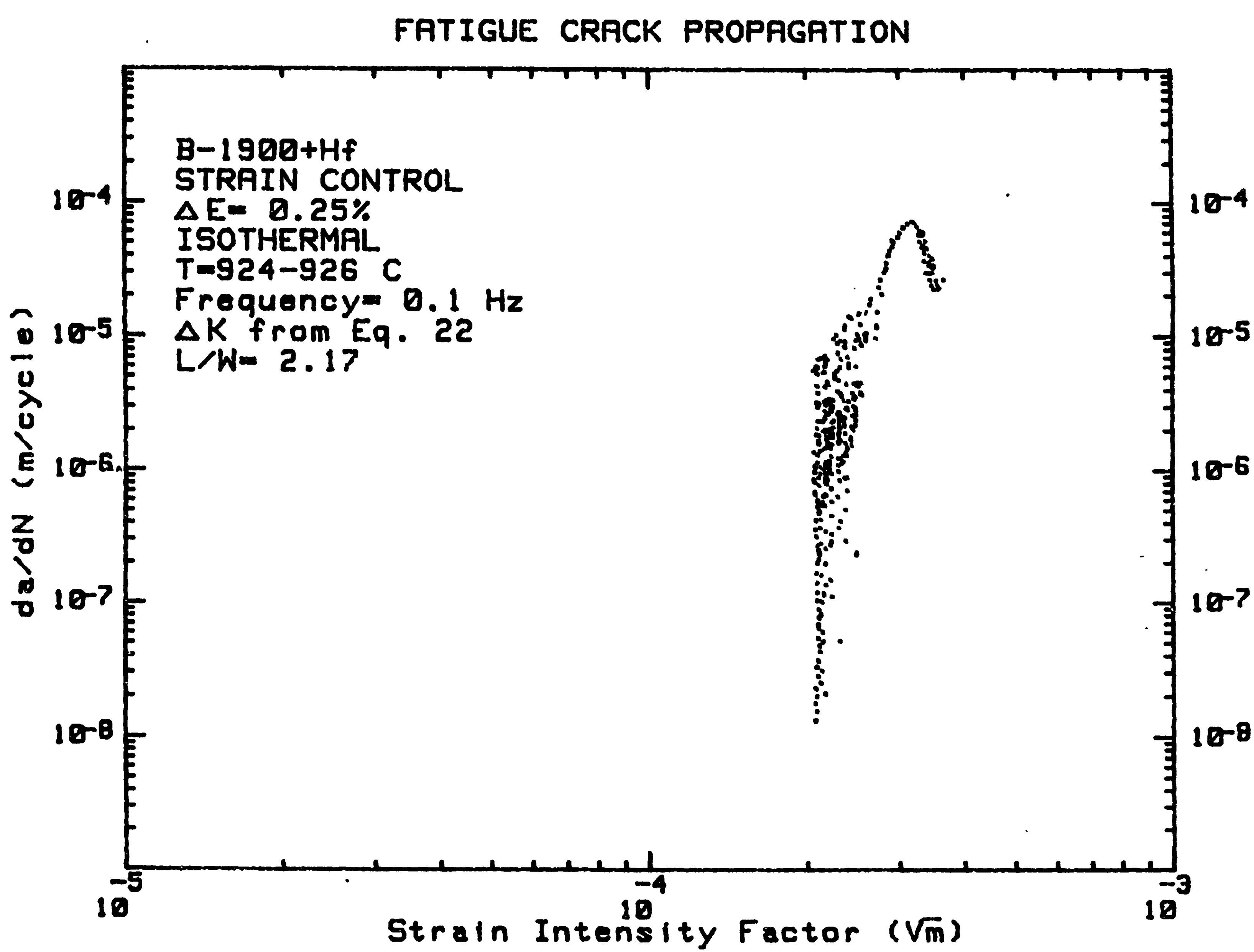


Figure 5. FCP rates as a function of the strain intensity factor calculated using Eq. 25.

as it relates to an actual specimen, such as that in Fig. 3, and its grips. Two points need to be reviewed. First, the applicability of the zero rotation conditions and secondly, the implication of measuring and controlling the imposed displacement at the back face of the specimen.

In the present case, the zero rotation condition is effectively enforced at the base of the threaded ends of the specimen. Thus, the effective value of L/W for the specimen might be expected to be somewhat greater than that based on the length of uniform reduced gage section. The additional effective length would correspond to the no-crack bending compliance of the tapered shoulders connecting gage section and threaded ends. The tensile part of the additional compliance is not required, since the mean extension δ is measured over the gauge length of $L_{gauge} = 2.17 \cdot W = 25.4$ mm. If we let L_b denote the augmented effective bending length of the specimen, and $\eta_b = L_b/W$, then the modified versions of Eq. 22 is (see Appendix V.a).

$$K_I = \frac{E'\delta}{L} \cdot (\pi a)^{1/2} \cdot G(\xi\ ,\ \eta,\ \eta_b) \tag{28}$$

where

$$G(\xi\ ,\ \eta\ \eta_b) = \frac{F_1(\xi)\left\{1 - \dfrac{6F_2(\xi)}{F_1(\xi)} \cdot \dfrac{\hat{C}_{12}(\xi)}{(12\eta_b + C_{22}(\xi))}\right\}}{\left\{1 + \dfrac{1}{\eta}\left[\hat{C}_{11}(\xi) - \dfrac{\hat{C}^2_{12}(\xi)}{(12\eta_b + \hat{C}_{22}(\xi))}\right]\right\}} \tag{29}$$

and, again, $\eta = L/W$.

The other point to consider is the consequence of controlling the back face displacement, δ_{BF} (see Fig. 3) instead of at the centerline of the specimen. Because the displacement δ is controlled at L instead of L_b where no-rotation is effectively enforced, the displacement at the back face is equal to the displacement at the centerline (δ_{cl}) minus the displacement induced by rotation, θ_L, at L, i.e.,

$$\delta_{BF} = \delta_{cl}(L) - \frac{W}{2} \cdot \theta_L. \tag{30}$$

Now, using Eq. 30 with $\delta_{cl}(L)$ given by Eq. 12.2, the final corrected version K_I which takes into account both the effective length (L_b) and back face displacement is given by (See Appendix V.a).

$$K_I = \frac{E'\delta}{L} \cdot (\pi a)^{1/2} \cdot G_c\,(\xi\,,\,\eta,\,\eta_b) \tag{31}$$

where

$$G_c\,(\xi,\eta,\eta_b) = \frac{F_1(\xi)\left\{1 - \dfrac{6F_2(\xi)}{F_1(\xi)} \cdot \dfrac{\hat{C}_{12}(\xi)}{(12\eta_b + \hat{C}_{22}(\xi))}\right\}}{\left\{1 + \dfrac{1}{\eta}\left[\hat{C}_{11}(\xi) - \dfrac{\hat{C}_{12}^2(\xi)}{12\eta_b + \hat{C}_{22}(\xi)} + \dfrac{6\,\hat{C}_{12}(\xi)(\eta-\eta_b)}{12\eta_b + \hat{C}_{22}(\xi)}\right]\right\}} \tag{32}$$

with η, η_b and ξ defined as before. The additional term appearing in the denominator of Eq. 32, as compared to Eq. 29, is of litte consequence for shorter cracks, and small values of $\eta_b-\eta$. Thus, the fact that back face, as opposed to centerline displacement is monitored is generally of minor significance.

Based on the assumption that $(da/dN)_{max}$ coincides with a maximum in the ΔK_ε (i.e., the correlation is monotone), plots of da/dN versus a/W were compared with plots of $G'_c(a/W)$ versus a/W where

$$G'_c(a/W) \equiv (\pi\xi)^{1/2} \cdot G_c(\xi, \eta, \eta_b) \tag{33}$$

for various values of η_b. Note from Eq. 33 that for prescribed δ, ΔK_ε is linearly proportional to $G'_c(\xi)$. The results are plotted in Fig. 6 versus relative crack depth, a/W. The maximum growth rate, occurring at a/W = 0.52, coincides with the location of maximum G'_c (ξ,η,η_b) for the case of η_b = 2.5. A plot of da/dN versus ΔK_ε inferred from the present analysis with L_b/W = 2.5 is shown in Fig. 7. A very tight correlation between the crack growth rates and the ΔK_ε's is observed with a monotone correlation, within reasonable limits of experimental scatter. One concludes that our K-solutions (Eq. 28, 31) correlate the data providing the proper "effective" bending length is used. It should be pointed out, that an L_b/W ratio of 2.5 (which corresponds to a length of 29 mm) is the distance between the mid-shoulders of the specimen.

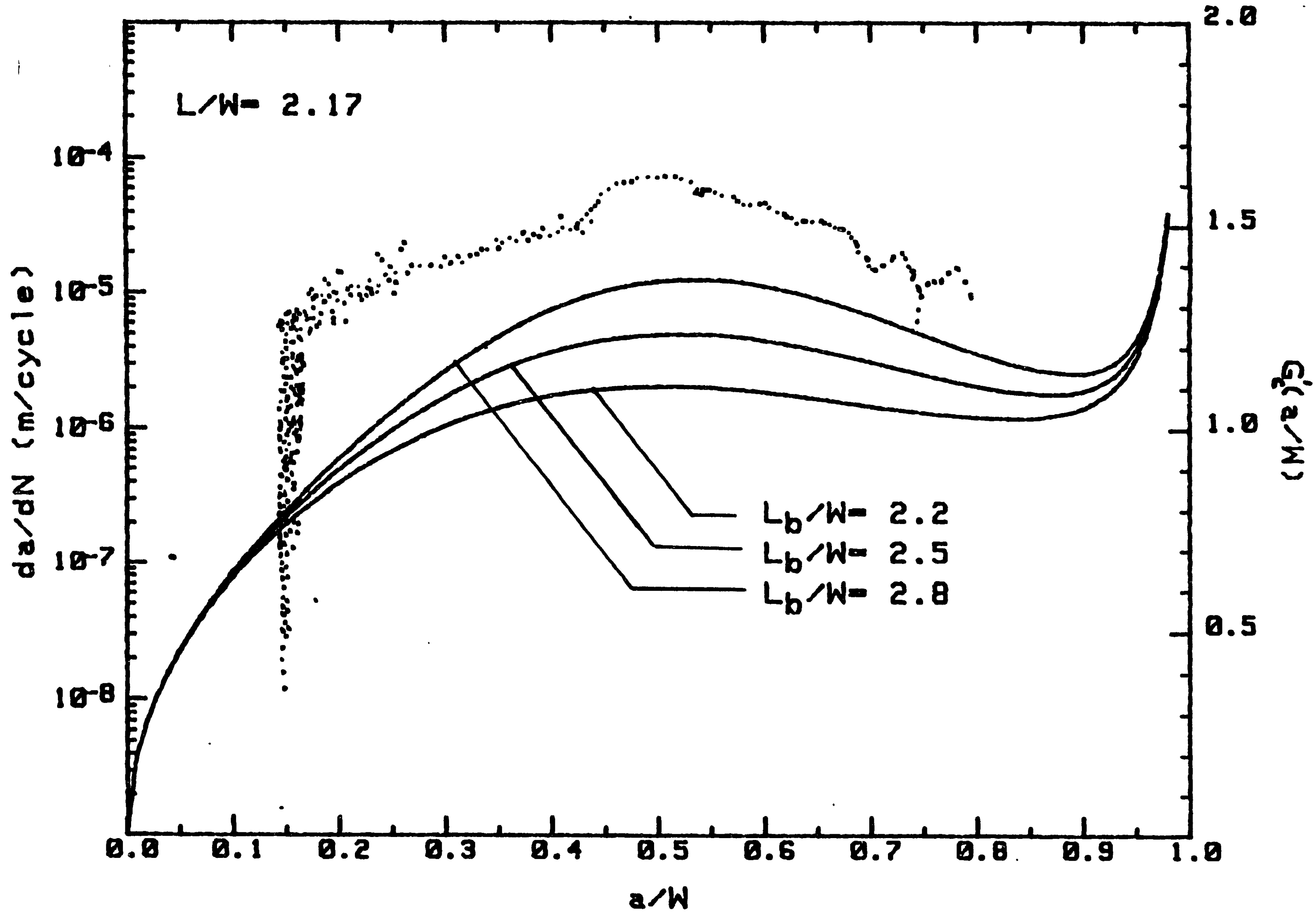


Figure 6. FCP rates as a function of a/W and Eq. 33 as a function oa a/W for various η_b ratios.

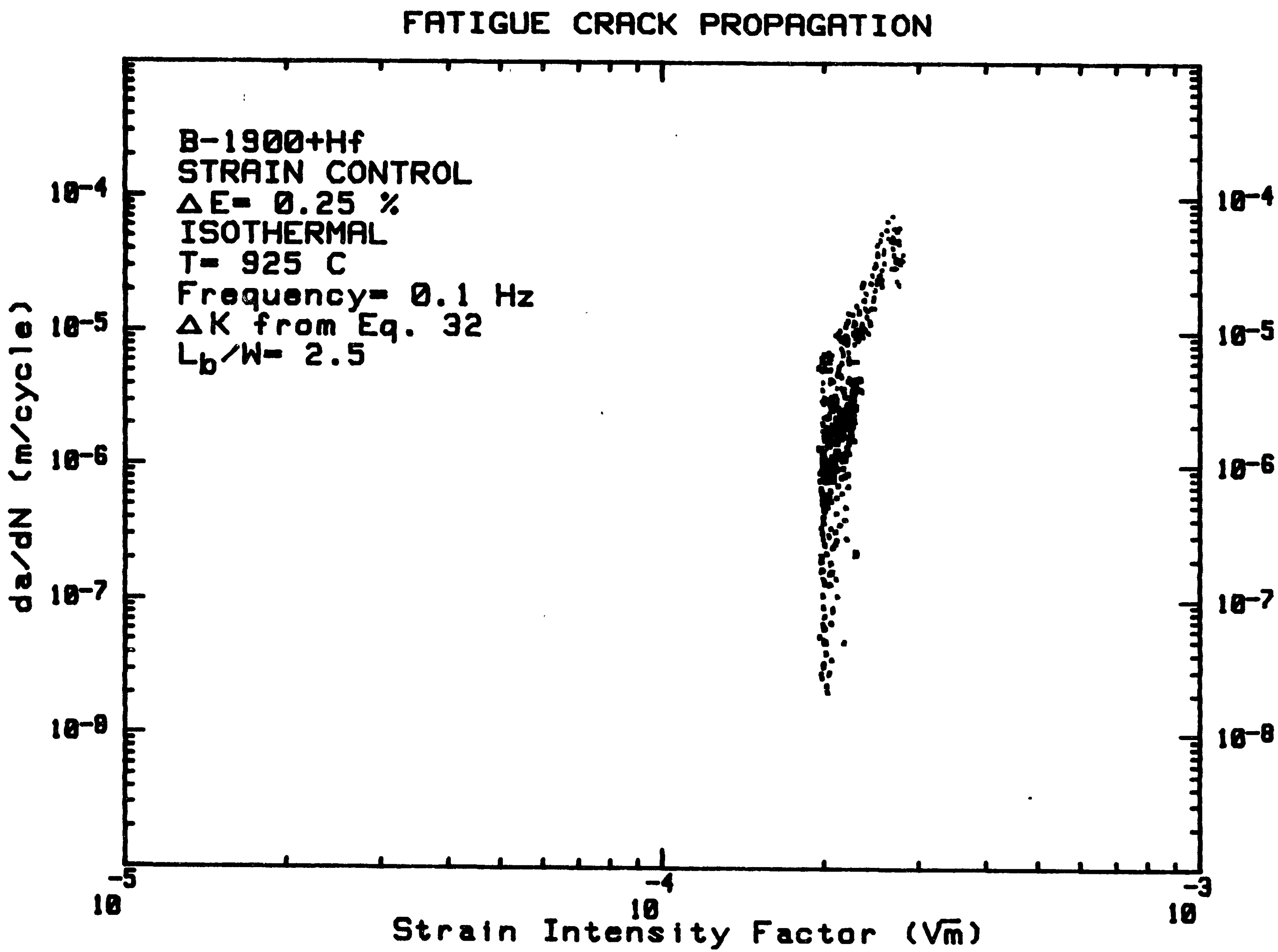


Figure 7. FCP rates as a function of the strain intensity factor calculated using Eq. 32.

CONCLUSION

For a given geometry and applied end-displacements, Figs. 4 to 6, show that the uncritical use of conventional stress-derived K_I solutions can substantially overestimate the K_I values. While such an eventuality is conservative in application, it is non-conservative when testing to obtain basic materials behavior.

In the present case, the use of the strain intensity factor as represented by Eq. 26 seems to have two important limitations, both of which grow in importance at higher a/W levels. First, this definition of strain intensity factor as crack growth driving parameter does not account for the substantial load-shedding associated with the increasing crack compliance. This limitation can, in part, be mitigated (as in Fig. 5), by monitoring the decreasing load amplitude as the crack extends, and substituting this crack-length-dependent load into a tensile loading intensity factor calibration. The second major limitation of the traditional strain intensity factor defnition is that it fails to account for the development of a closing bending moment, associated with the prescribed zero-rotation boundary condition. This closing bending moment further reduces the effective crack driving force. A complete analysis of tension and bending moment in the specimen is required. When the K-solutions are derived from a line-spring analysis and combined with an effective specimen length, good correlations are found between K and the experimental fatigue crack growth data.

REFERENCES

1. J.M. Bloom, International Journal of Fracture, Vol. 3 (1967), pp. 235-242.

2. D.O. Harris, Trans. of ASME, Journal of Basic Engineering, Vol. 89 (1969), pp. 49-54.

3. O.L. Bowie and C.E. Freese, Engineering Fracture Mechanics, Vol. 14 (1981), pp. 519-526.

4. E.P. Popov, Introduction to Mechanics of Solids, Prentice Hall, Englewood Cliffs (1968).

5. H. Tada, P.C. Paris, and G.R. Irwin, The Stress Analysis of Cracks Handbook, Del Research Corporation, Hellentown (1973).

6. F.A. McClintock, D.M. Parks, J.W. Holmes, and K.W. Bain, Engineering Fracture Mechanics, Vol. 20 (1985), pp. 159-167.

7. N. Marchand and R.M. Pelloux, "A Computerized Test System for Thermal Mechanical Fatigue Crack Growth", submitted for publication.

8. N. Marchand and R.M. Pelloux, in Time-Dependent Fracture, Martinus Nijhoff (1985), pp. 167-179.

9. R.C. Boettner, C. Laird, and A.J. McEvily, Trans. of Metallurgical Society of AIME, Vol. 233 (1965), pp. 379-387.

10. A.J. McEvily, "Fatigue Crack Growth and the Strain Intensity Factor", in Fatigue and Fracture of Aircraft Structures and Materials, AFFDL-TR-70-144 (1970).

11. A.E. Gemma, F.X. Ashland, and R.M. Masci, ASTM Journal of Testing and Evaluation, Vol. 9 (1981), pp. 209-213.

12. T. Koizumi and M. Okazaki, Fatigue of Engineering Materials and Structure, Vol. 1 (1979), pp. 509-520.

13. B.N. Leis and T.F. Forte, ASTM STP 743 (1981), pp. 100-124.

14. B.N. Leis, Engineering Fracture Mechanics, Vol. 22 (1985), pp. 279-293.

APPENDIX V.a

Consider a specimen of gage length L with its associated shoulders. Let's define L_b the effective length at which the zero rotation is enforced ($\theta|_{L_b}=0$). Obviously $L_b > L$ and, as a consequence the back face displacement δ_{BF} measured at L will be equal to the centerline displacement minus the angular displacement at L, i.e., $\frac{W}{2}\bullet\theta_L$. Therefore, the known quantities are δ_{BF} (measured at L) and θ_{tot} across L_b. Equations 12.1 and 12.2 can now be written as

$$0 = \theta_{tot}(L_b) = \theta(L_b) = \theta_{nc}(L_b) + \theta_c \tag{A.1}$$

$$\delta_{BF} = \delta_{tot}(L) = \delta_{nc}(L) + \delta_c(L) - \frac{W}{2}\ \theta(L) \tag{A.2}$$

Substitute Eq. 10 into A.1 and re-arranging, yielding

$$\frac{M}{W} = \frac{-C_{12}\ E'BW}{12\ \eta_b + C_{22}\ E'BW} \cdot N \tag{A.3}$$

On using the dimensionless crack compliance defined by Eqns. 16.1 to 16.3, Eq. A.3 simplifies to

$$\frac{M}{W} = \frac{-\hat{C}_{12}}{12\ \eta_b + \hat{C}_{22}} \cdot N \tag{A.4}$$

where $\eta_b = L_b/W$. Noting that $M(L_b) = M(L)$, the displacement δ_{BF} (Eq. A.2) is, using Eq. 10 and A.1,

$$\delta_{BF} = [\frac{NL}{BWE'} + C_{11}N + C_{12}M\ ] - \frac{W}{2}\ [\frac{12ML}{E'BW^3} + C_{22}M + C_{12}N] \tag{A.5}$$

which, upon re-arranging and using the dimensionless crack compliance (Eqns. 16.1-16.3) yields

$$N = E'BW\,\frac{\delta_{BF}}{L} \cdot \frac{1}{\left\{1 + \frac{1}{\eta}\left[\hat{C}_{11}(\xi) - \frac{\hat{C}^2_{12}(\xi)}{(12\eta_b + \hat{C}_{22}(\xi))} + \frac{6\hat{C}_{12}(\xi)(\eta-\eta_b)}{(12\eta_b + \hat{C}_{22}(\xi))}\right]\right\}} \tag{A.6}$$

Finally, recalling that

$$K_I = \frac{N}{BW}\sqrt{\pi a} \cdot F_1(\xi) + \frac{6M}{BW^2} \cdot \sqrt{\pi a} \cdot F_2(\xi) \tag{A.7}$$

and upon substitution of Eqns. A.4 and A.6 into A.7, we get

$$K_I = \frac{E'\delta_{BF}}{L} \cdot \sqrt{\pi a} \cdot \frac{F_1(\xi)\left\{1 - \frac{6F_2(\xi)}{F_1(\xi)} \cdot \frac{\hat{C}_{12}}{(12\eta_b + \hat{C}_{22})}\right\}}{\left\{1 + \frac{1}{\eta}\left[\hat{C}_{11}(\xi) - \frac{\hat{C}^2_{12}}{12\eta_b + \hat{C}_{22}(\xi)} + \frac{6\hat{C}_{12}(\xi)(\eta-\eta_b)}{12\eta_b + \hat{C}_{22}}\right]\right\}} \tag{A.8}$$

Equation A.8 corrects for the fact that actual displacements are measured at back face instead of the centerline, and for the fact that the zero-rotation condition is imposed not at L, but at some distance L_b close to the actual grips.