A Dynamic Analysis of the Primary System in Experimental Breeder Reactor II

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science in Nuclear Engineering

in

The Department of Nuclear Science

by
George Matthews Marshall V
B.S.E.E., Louisiana State University, 1984
August, 1987

to my father

Acknowledgements

I would like to thank Dr. Mark Williams for teaching me so much of what I know about engineering and Dr. Jack Courtney for his support and for the opportunity to work in the area of nuclear engineering and at the Argonne National Laboratory. Thanks to Dr. Burk Huner for providing me with many great educational experiences and to Dr. Adnan Yucel for serving on my committee.

In the Operations Analysis section of the EBR-II division at Argonne National Laboratory I would like to thank Dr. Wayne Lehto and Eric Dean for their support. Thanks also to Dr. Robert Forrester and Dr. Howard Larson for having time to answer the odd question. I would also like to thank the rest of the Operations Analysis group and all the other people in building T-2 at EBR-II for their incomparable kindness.

On a personal note thanks to John, Pam, Julie and Debbie Gale for their friendship. Last and most thanks to Denise Roshto for her patience during the time that this work was going on.

Table of Contents

List of Tables

List of Figures

Abstract

Dynamic analysis of nuclear reactors and their associated systems is important to insure the safety, stability and expected behavior of these systems. Dynamic analysis may be accomplished using real time or frequency domain methods. In this thesis, the primary system of Experimental Breeder Reactor II at the Argonne National Laboratory is analyzed using numerical Fourier transform techniques on data obtained from the reactor and its systems as a result of a multifrequency rod perturbation. This same kind of analysis is performed on results from a computer simulation using the code DSNP (Dynamic Simulator for Nuclear Power Plants), a real time FORTRAN simulator. Both of these results are compared with Fourier transform functions representing system components or processes derived based on known system parameters.

Chapter One

Introduction

With the advent of larger and faster computers it has become possible to combine equation sets which represent individual components of a given system and solve them creating a total system simulation. Through the use of numerical techniques on the computer, this type of modeling allows the examination of the effect of changing conditions in one part of the system on some other part of the same system.

This type of analysis is now being used on nuclear reactors and their associated systems. Computer simulations of nuclear systems make dynamic reactor analysis, control system evaluation and safety analysis easier.

The Operations Analysis section of the EBR-II (Experimental Breeder Reactor-II) division at Argonne National Laboratory is involved in the development and verification of one of these codes named DSNP (Dynamic Simulator for Nuclear Power Plants) which can simulate nuclear reactors and reactor plant systems. The DSNP program is a FORTRAN based simulation code that models components of reactors. The code is presently being used to predict the behavior of the EBR-II core and balance of plant.

Frequency response testing is one of the means by which the performance of a reactor and associated systems can be measured. Frequency response testing is carried out measuring the system output response arising from sinusoid variation in control rod position, etc, at various driving frequencies. Analytically the frequency response can be determined by calculating the Laplace transforms of the input signal and the output signal and dividing the output by the input. This yields a function of frequency that is termed a transfer function. For a sinusoidal input variation, the steady state output of a linear system will be a sinusoid of the same frequency, but with a different amplitude and phase. The transfer function in this case defines the magnitude and phase shift of the output sinusoid that is associated with an input sinusoid of a certain frequency.

The SHRT (Shutdown Heat Removal Testing) program was a test program implemented at EBR-II to improve understanding and characterization of the EBR-II plant. The purpose of this thesis is to simulate a multifrequency test with DSNP that was implemented as part of the SHRT program, evaluate transfer functions for different parts of the system and compare them with those transfer functions derived analytically based on known system parameters and those transfer functions calculated from collected data. This analysis and comparison will render an evaluation of DSNP's ability to predict dynamic behavior by providing information on the propagation time of signals in the systems modeled by DSNP and on the accuracy of the predicted magnitude of the propagated signals. It will also explore the possibility of constructing an EBR-II system simulator with transfer functions. The results from this thesis should also provide information on the accuracy of known system parameters, the quality of instrumentation at EBR-II and the data acquisition system for those instruments.

Chapter Two

History of EBR-II

EBR-II is an experimental LMFBR (Liquid Metal Fast Breeder Reactor) which is one of the facilities of the Argonne National Laboratory at the Idaho National Engineering Laboratory. Its systems consist of a metal fueled and sodium cooled reactor with a thermal output of 62.5 MWth; an intermediate closed loop of sodium coolant; and a sodium to steam cycle which can produce about 20 MW of electrical power.

EBR-II's original missions were to demonstrate the ability to use an LMFBR to produce power; prove that a breeding ratio greater than one could be obtained in an LMFBR and therefore more fuel produced than consumed; and prove that an integral facility could be designed where spent fuel was reprocessed and returned to the reactor without being removed from the plant site.

In 1961 dry criticality experiments were conducted at EBR-II and in November of 1963 the core was brought to "wet" criticality with the core submerged in sodium. The reactor achieved a power of 45 MWth in March of 1965 and was operated at 62.5 MWth for the first time in November of 1970.

By 1969 EBR-II had demonstrated its design objectives. It is now being used as an irradiation facility supporting the fuels and materials development program for LMFBRs. It is also used to conduct experiments to test special plant response conditions like losing the coolant flow or other accident conditions. A more detailed description of the plant and the history of some of the experiments which have been done that are relevant to this thesis are found in Chapter 5.

Chapter Three

System Response Testing

A system can be defined as any device or combination of devices which produces some response when excited. The input and output of the systems that will be discussed here are functions of the single variable time. The assumption is made that these systems are causal and linear, meaning respectively that there can be no output from these systems until an input has been applied, and that the principal of superposition applies to them. A simple system model is shown in Fig. 1 with an input variation $\delta x(t)$ and on output variation $\delta y(t)$.

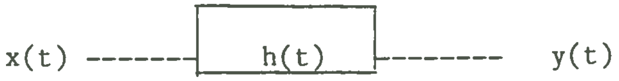

Fig. 1 Simple system model

The time dependent deviations are given by

$$x_0 + \delta x(t) = x(t) \text{ and } y_0 + \delta y(t) = y(t),$$

where x(t) and y(t) are the time dependent input and output responses and x_0 and y_0 are the initial conditions on x and y.

The impulse response of a system is defined as the response of a system to the impulse function, sometimes called the Dirac delta function. The impulse function is shown on the next page in Fig. 2.

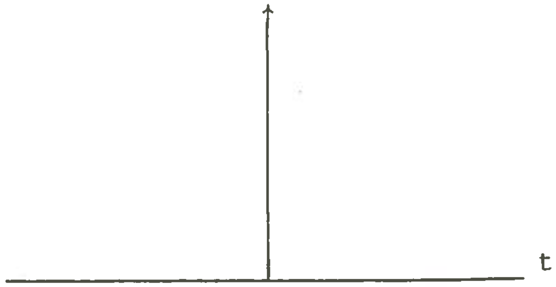

Fig. 2 The impulse function

The main characteristic of the function is that it is nonzero for only an infinitesimal length of time. Its basic properties are

$$f(t) = 0 \qquad \text{for} \qquad t \neq 0 \tag{1}$$

$$\int_{-\infty}^{\infty} f(t)dt = 1 \tag{2}.$$

An example of the impulse response is shown in Fig. 3.

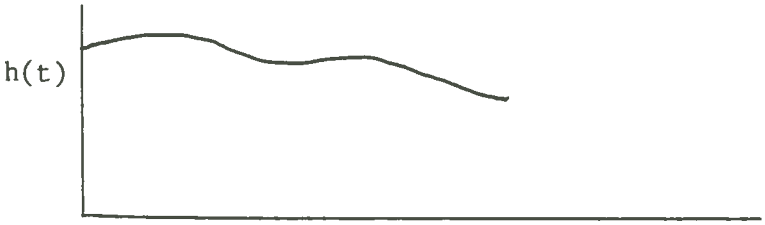

Fig. 3 The impulse response

If one assumes that an input function is composed of a series of Dirac delta functions of the form $x(\tau)d\tau$, then the differential response from one input delta function at some point in time is given by the expression

$$d(\delta y(t)) = \delta x(\tau)d\tau h(t-\tau) \tag{3}$$

and the integration of these responses is

$$\delta y(t) = \int_0^t h(t-\tau)\delta x(\tau)d\tau, \text{ where } \delta x(t) = 0, \text{ for } t < 0 \tag{4}.$$

This is known as the convolution integral. The convolution integral is solved by a computer during a real time simulation. The computer evaluates this integral by discretizing in time to calculate the output of a given system. There will be as many of these integrals for the computer to solve as there are responses. One of the objectives of this thesis is to determine transfer functions which can be used to evaluate the system impulse response functions.

It is possible to expand a piecewise continuous periodic function in a Fourier series

$$f(t) = a_0 + \sum_{n=1}^{\infty} (a_n \cos(nt)+b_n \sin(nt)) \qquad (5)$$

where $a_0 = 1/T \int_T f(t)dt$

$a_n = 2/T \int_T f(t)\cos(n\omega t)dt$ where $\omega = 2\pi/T$

$b_n = 2/T \int_T f(t)\sin(n\omega t)dt$ and T=period of the function.

These are the well known Fourier expansion coefficients which may be represented by line spectra like the ones shown in Fig. 4.

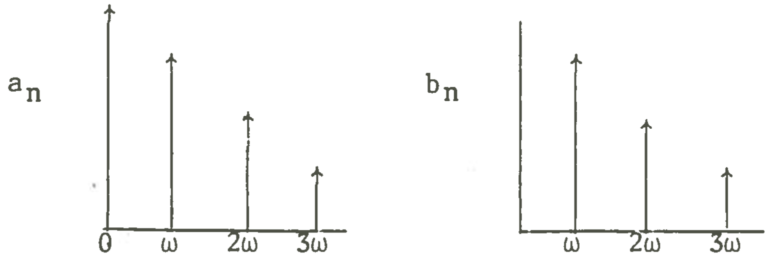

Fig. 4 Fourier expansion coefficients

It is also possible to expand the function in a complex Fourier series

$$f(t) = \sum_{n=-\infty}^{\infty} C_n e^{jn\omega t} \qquad (6)$$

yielding a spectrum which may be real or complex.

If one allows the period of a function to approach infinity, the line spectra converge and one obtains the Fourier transform

$$F(j\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt \qquad (7)$$

and the inverse transform

$$f(t) = 1/2\pi \int_{-\infty}^{\infty} F(j\omega)e^{j\omega t} \, d\omega \qquad (8).$$

These relations indicate that an input function can be viewed as being composed of sinusoidal modes of varying frequencies. Thus we are interested in finding the system output arising from sinusoidal inputs of different frequencies. The system response will be determined by using Laplace transforms. The Laplace transform, F(s), of a time dependent function f(t) is

$$F(s) = \int_{0}^{\infty} f(t)e^{-st} \, dt \quad \text{where s is a complex number} = \sigma + j\omega \quad (9).$$

However, since we are interested in the steady state system output arising from input sinusoids, we assume that the real part of the exponential dies out leaving us the Laplace transform evaluated at s = $j\omega$. In this case the Laplace transform is a continuous function of frequency which can be expressed in the complex polar form

$$F(j\omega) = |F(j\omega)|e^{\theta(j\omega)} \qquad (10)$$

where $|F(j\omega)|$ is the amplitude of the frequency-dependent complex function and

$\theta(j\omega)$ is the phase.

Taking the Laplace transform of the convolution integral one obtains

$$L[\delta y(t)] = L[\int_0^\infty h(t-\tau)\delta x(\tau)d\tau] \qquad (11)$$

then

$$\delta Y(s) = \int_0^\infty e^{-st}[\int_0^\infty \delta x(\tau)h(t-\tau)d\tau]dt \qquad (12)$$

$$\delta Y(s) = \int_0^\infty \delta x(\tau)[\int_0^\infty h(t-\tau)e^{-st}\, dt]d\tau \qquad (13)$$

and with the transformation of variables

$$a = t-\tau \quad da=dt \quad t=a+\tau$$

$$\delta Y(s) = \int_0^\infty \delta x(\tau)[\int_0^\infty h(a)e^{-s(a+\tau)}\, da]d\tau \qquad (14)$$

$$\delta Y(s) = \int_0^\infty \delta x(\tau)e^{-s\tau}\, d\tau\int_0^\infty h(a)e^{-sa}\, da \qquad (15)$$

or

$$\delta Y(s) = \delta X(s)H(s) \qquad (16)$$

$$H(s) = \delta Y(s)/\delta X(s) \qquad (17).$$

The steady state response is obtained by setting $s = j\omega$, so that

$$H(j\omega) = \delta Y(j\omega)/\delta X(j\omega)$$

where $\delta Y(j\omega)$ is the Laplace transform of the output signal, $\delta X(j\omega)$ is the Laplace transform of the input signal and $H(j\omega)$ is a complex variable termed the steady state system transfer function or network transfer function. For all signals of interest, for which δX and δY are zero prior to t=0, the Laplace transform evaluated at $s = j\omega$ will be equivalent to the Fourier transform discussed previously.

By making this Laplace transformation, we change from calculating the response of a system in the time domain by evaluating a convolution integral, to calculating the response of a system in the frequency domain by performing a multiplication. The advantages of the latter approach are obvious when one considers a complicated system of multiple components. For example, suppose that the steady state output of the system in Fig. 5 is desired.

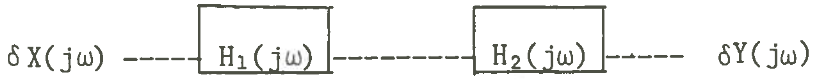

Fig. 5 System example 1

This system consists of two components which have individual transfer functions of H_1 and H_2, respectively. The steady state system output will correspond to

$$\delta Y(j\omega) = H_1(j\omega)H_2(j\omega)\delta X(j\omega) = H_{S1}(j\omega)\delta X(j\omega) \qquad (18)$$

where $H_{s1}$is the system transfer function given by

$$H_{S1} = \delta Y/\delta X = H_1H_2 \qquad (19).$$

As a second example consider the system shown in Fig. 6, which adds a feedback loop to the previous system.

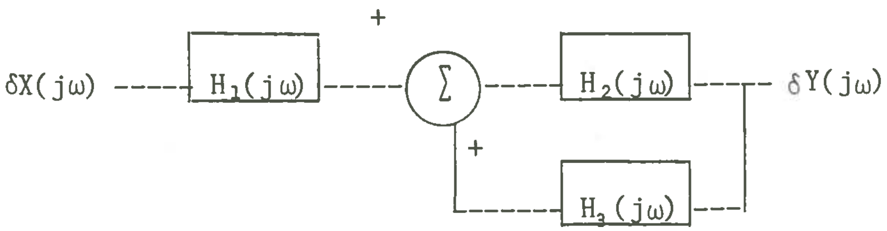

Fig. 6 System example 2

The steady state output of this system can be evaluated using standard block algebra to be

$$\delta Y(j\omega) = H_1(j\omega)\frac{H_2(j\omega)}{1-H_2(j\omega)H_3(j\omega)}\delta X(j\omega) \quad = \quad H_{S2}(j\omega)\delta X(j\omega) \qquad (20)$$

where H_{S2} is the system transfer function. Both of these results are of course still in the frequency domain and would have to be converted into the time domain by taking the inverse Laplace transform.

The transfer function of a system is a valuable thing to know, since the system output can be determined for any input by simply multiplying it by the transfer function. There are two basic ways that one may evaluate the overall transfer function of a system or its components. The first is analytical and the second is experimental.

To analytically evaluate a transfer function, one takes the Laplace transform of the equations representing a system. The equations are manipulated into an expression for the ratio of the output over the input. For nonlinear systems or for time varying systems which have time dependent system parameters, the equations can be linearized and the same process carried out.

Transfer functions are determined experimentaly by applying an appropriate forcing function to a system and recording the output. The Laplace transform of both signals are taken, and the output is divided by the input to yield the transfer function. If the signal and system response are allowed to reach steady state, then the input, I(t), and output, O(t), can be described by the Fourier transform and the transfer function will be of the form,

$$H(j\omega) = \frac{F[O(t)]}{F[I(t)]} \qquad (21)$$

or

$$H(j\omega) = \frac{O(j\omega)}{I(j\omega)} \qquad (22)$$

which is the Fourier integral representation of the transfer function, where

$$O(j\omega) = \int_{-\infty}^{\infty} O(t)e^{-j\omega t}dt \qquad (23)$$

and

$$I(j\omega) = \int_{-\infty}^{\infty} I(t)e^{-j\omega t}dt \qquad (24)$$

If we write the input function as a Fourier series, then the steady state output can be evaluated using superposition to be

$$O(t) = \sum_{k=-\infty}^{\infty} H(j\omega_k)C_k e^{j\omega_k t} \quad \text{, where} \qquad (25)$$

$$I(t) = \sum_{k=-\infty}^{\infty} C_k e^{j\omega_k t} \quad \text{with} \quad \omega = \frac{2\pi k}{T} \qquad (26)$$

by orthogonality, the expansion coefficients are given by

$$C_k = \frac{1}{nT}\int_0^{nT} I(t)e^{-j\omega_k}\,dt \qquad (27)$$

$$C_k H(j\omega_k) = \frac{1}{nT}\int_0^{nT} O(t)e^{-j\omega_k}\,dt \qquad (28).$$

Eq. 28 is divided by Eq. 27 to obtain

$$H(j\omega_k) = \frac{\frac{1}{nT}\int_0^{nT} O(t)e^{-j\omega_k t}dt}{\frac{1}{nT}\int_0^{nT} I(t)e^{-j\omega_k t}dt} \qquad (29)$$

Integrals of the form in Eq. 27 can be expressed in the form of discrete summation

$$\frac{1}{T}\int_0^{T} f(t)e^{-j\omega_k t}dt \cong \frac{1}{T}\sum_{k=0}^{N-1} f(t_k)e^{-j\omega t_k}\Delta t \qquad (30)$$

where $f(t_k)$ is the evaluation of $f(t)$ at the left edge of the time interval Δt and N is the number of time intervals. If we let f_k equal to $f(t_k)$ and since $t=T/N$ we obtain

$$\frac{1}{T}\int_0^{T} f(t)e^{-j\omega_k t}dt \cong \frac{1}{N}\sum_{k=0}^{N-1} f_k e^{-j\omega t_k}$$

This is the DFT (Discrete Fourier Transform) approximation of the Fourier transform. The DFT is the tool used to evaluate transfer functions from experimental data and simulation results in this thesis.

The technique of expanding any signal as a linear combination of orthogonal functions may be written in the general form

$$x(t) = \sum_{k=-\infty}^{\infty} X(k)\phi_k(t) \qquad (31)$$

where $x(t)$ is a function being decomposed into a series of the basis functions $\phi_k(t)$, and the coefficients $X(k)$ are the components of $x(t)$ associated with the respective expansion mode. If we allow this series to describe specific values of $x(t)$ at nT where T is fixed and $n=\ldots,-1,0,1,2,\ldots$ and define $x(n)$ and $\phi_k\ (n)$ as $x(t)$ and $\phi_k\ (t)$ evaluated at $t=nT$ then Eq. 31 becomes

$$x(n) = \sum_{k=-\infty}^{\infty} X(k)\phi_k(n) \qquad (32).$$

If we assume only N coefficients are nonzero then one has

$$x(n) = \sum_{k=0}^{N-1} X(k)\phi_k(n) \qquad (33).$$

This equation can be expressed in matrix notation as

$$\underline{x} = \underline{\underline{\phi}}\,\underline{X} \qquad (34)$$

where $\underline{x}$ and $\underline{X}$ are vectors and $\underline{\underline{\phi}}$ is a matrix. In this equation, $\underline{X}$ is a transform-domain representation of $\underline{x}$. To evaluate $\underline{X}$ one must solve the equation

$$\underline{X} = \underline{\underline{\phi}}^{-1}\,\underline{x} \qquad (35).$$

This would normally involve inverting an NxN matrix. However, there are algorithms which are designed to optimize the solution of this matrix equation for a Fourier basis. These are called the FFT (Fast Fourier Transform) algorithms. Many of them are available, but the one that will be used in this thesis is a "power of two" FFT. It is so called because it requires a number of data points equal to a power of two. These types of FFT algorithms take advantage of the fact that there are symmetries in the $\underline{\underline{\phi}}$ matrix which can be utilized to reduce the number of multiplications and additions to $N\log_2 N$ operations. A copy of the basic code that was used for Fourier analysis in this thesis and a slightly more detailed explanation of how "power of two" FFT algorithms work is in Appendix A. The only requirements for usage are that one uses an integral number of cycles when performing the analysis. If one wishes to use a "power of two" algorithm in order to analyze a signal whose period multiplied by the number of samples per period is not a power of two, then one must "pad" the remaining samples with zeros. There are Fourier analysis codes that do not require a power of two data points. Their disadvantages are that they are more complex and take more computer time to run. A more detailed and complete description of these types of codes can be found in Elliot and Rao (1).

There is one precaution that must be taken when data is being sampled and the FFT calculated. It is not possible to detect frequencies in a sampled signal above 2 samples per cycle or $1/2\Delta t$ cycles/time. This is called the Nyquist frequency. If there are frequencies above the Nyquist folding frequency, analysis will produce aliasing, or show them as lower frequencies instead of the actual higher ones. Precautions must be taken to avoid this situation either in the design of the test by eliminating the higher frequencies, or in the analysis by filtering them out.

Chapter Four

Description and Analytical Derivations of the Primary System in EBR II

As discussed in Chapter 3, one of the ways of obtaining a transfer function is to calculate it analyticaly. This chapter will discuss the primary system of EBR-II and derive analytical models with known system parameters to obtain solutions for the dynamic responses of some parts of it. It should be made clear at this point that it is not the intention of the author to derive all the transfer functions that are to be analyzed in the SHRT tests and DSNP simulation results. Rather the purpose of this section is to familiarize the reader with the types of dynamic systems found in the primary system of EBR-II and the behavior that is likely to be exhibited by some of them. It was originally hoped that the analytically derived transfer functions could be used to obtain more physical insight into the system response.

The schematic of the cooling system and reactor of EBR-II is shown in Fig 7 from Dean (2). The reactor is submerged in a pool of liquid sodium and is cooled by sodium flowing in the lower plenum which is pumped by the primary cooling pumps drawing the sodium from the pool itself. The sodium flows into the lower plenum through the inlet pipe shown in Fig. 8 from Dean (2) and into the fuel subassemblies. The core itself is located 0.4699 meters from the bottom of the fuel subassemblies and approximately 0.356 meters tall.

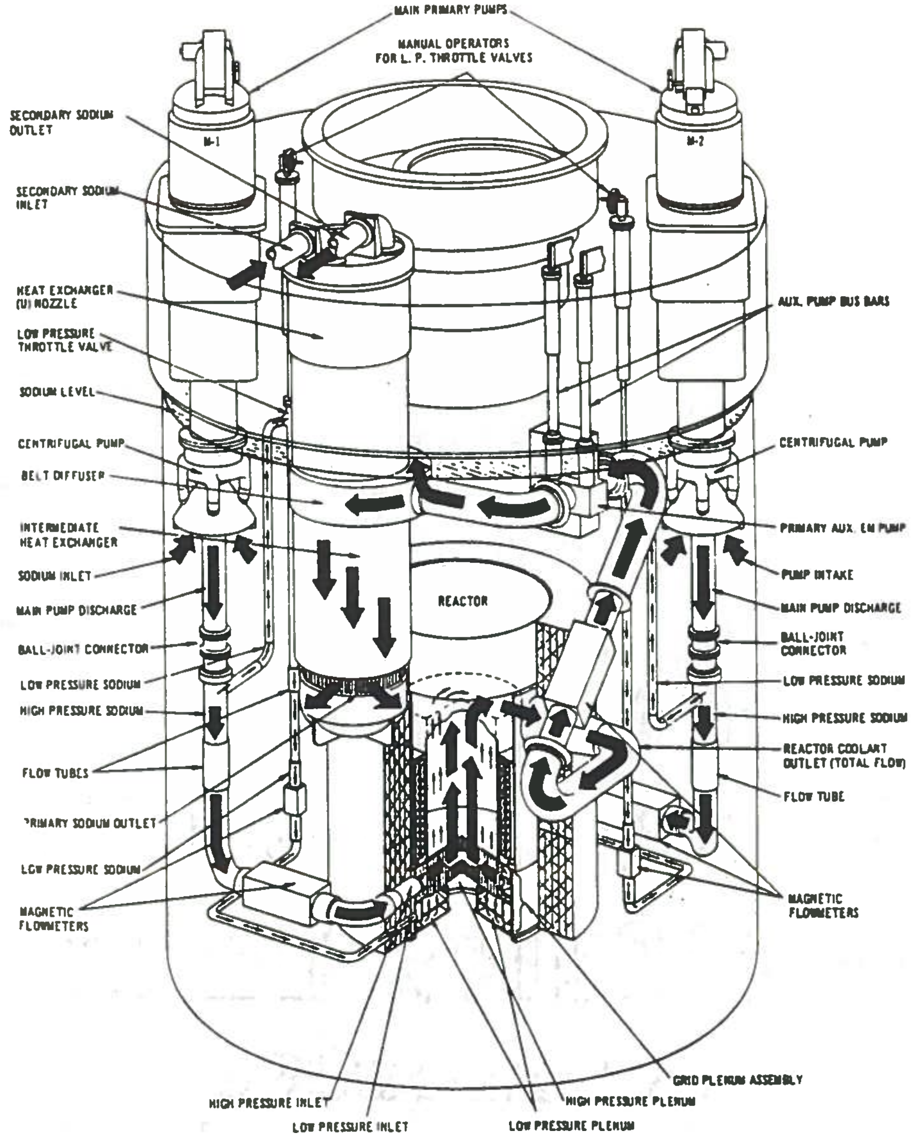

Fig. 7 Primary cooling system

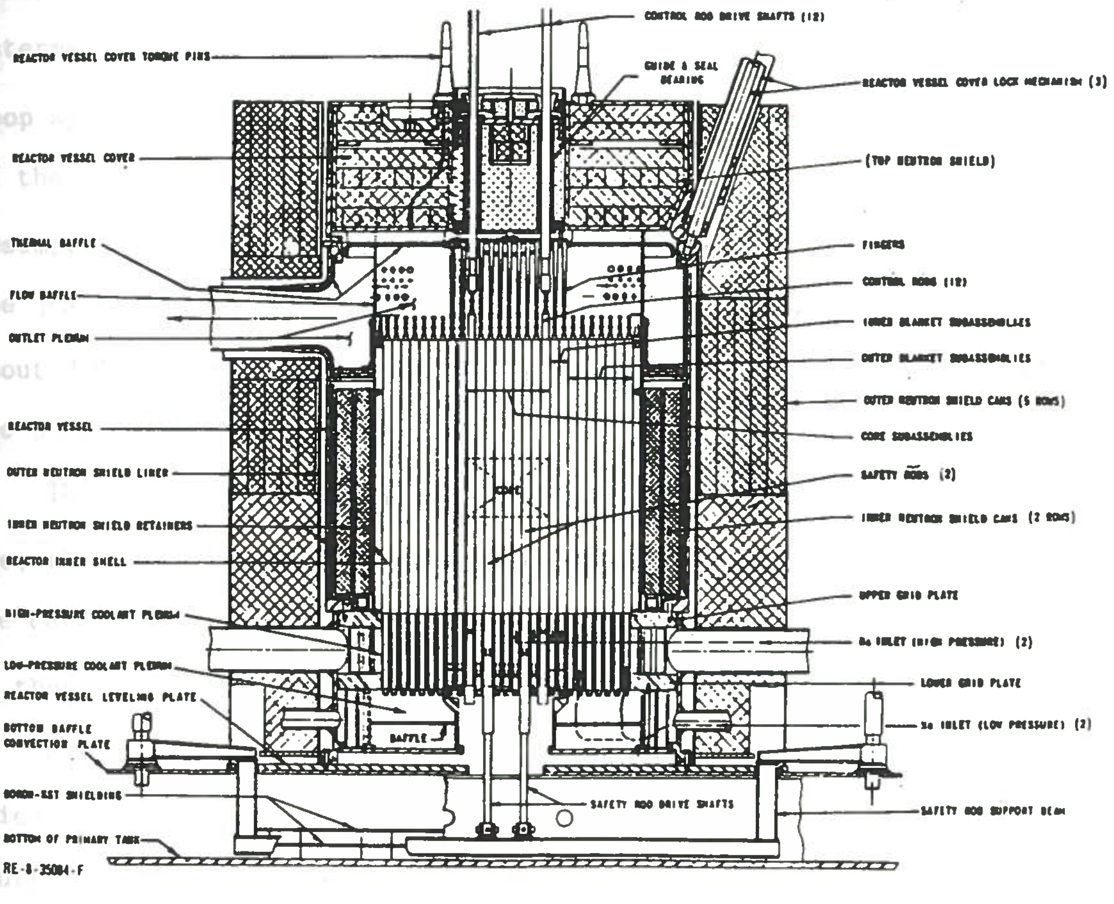
CONTROL ROD DRIVE SHAFTS (12)
REACTOR VESSEL COVER TORQUE PINS
GUIDE & SEAL BEARING
REACTOR VESSEL COVER LOCK MECHANISM (3)
REACTOR VESSEL COVER
(TOP NEUTRON SHIELD)
THERMAL BAFFLE
FINGERS
FLOW BAFFLE
CONTROL RODS (12)
OUTLET PLENUM
INNER BLANKET SUBASSEMBLIES
OUTER BLANKET SUBASSEMBLIES
OUTER NEUTRON SHIELD CANS (5 ROWS)
REACTOR VESSEL
CORE SUBASSEMBLIES
OUTER NEUTRON SHIELD LINER
SAFETY RODS (2)
CORE
INNER NEUTRON SHIELD RETAINERS
INNER NEUTRON SHIELD CANS (2 ROWS)
REACTOR INNER SHELL
UPPER GRID PLATE
HIGH-PRESSURE COOLANT PLENUM
Na INLET (HIGH PRESSURE) (2)
LOW-PRESSURE COOLANT PLENUM
LOWER GRID PLATE
REACTOR VESSEL LEVELING PLATE
BOTTOM BAFFLE CONVECTION PLATE
BAFFLE
Na INLET (LOW PRESSURE) (2)
BORON-SST SHIELDING
SAFETY ROD DRIVE SHAFTS
SAFETY ROD SUPPORT BEAM
BOTTOM OF PRIMARY TANK
RE-8-35084-F

Fig. 8 Reactor vessel and neutron shield assembly

After being heated, the sodium flows out of the top of the subassemblies, into the upper plenum and out into the reactor coolant outlet pipe called the "Z pipe" for its shape. After passing through the Z pipe, the sodium enters the intermediate heat exchanger and yields its energy to the coolant flowing in the secondary system. It then leaves the intermediate heat exchanger and flows back into the pool of sodium. This thesis will be a study of the dynamic behavior of the systems between the point that the coolant flows into the lower plenum and the point which the secondary sodium leaves the intermediate heat exchanger. The primary system is assumed an open loop system, meaning there is no significant feedback from the outlet of the heat exchanger to the inlet of the core. This is a reasonable assumption since there is approximatly 90,000 gallons of sodium in the pool and the volumetric flow rate from the heat exchanger is about 109 gallons per second, or a little more than a thousandth of the pool volume per second.

The primary system will be divided into four subsystems. They are; the active core; the non-active assembly region and plenum above the core; the Z pipe; and the intermediate heat exchanger. The first of these systems considered is the core.

A nuclear reactor is a controlled chain reaction of neutrons which produce power as they fission the fuel nuclei. Most of these neutrons appear in the system immediately following fission and contribute to the chain reaction. These are called prompt neutrons. However, a small fraction of the neutrons appear after the initial fission event. These are called delayed neutrons. They appear when certain fission products, called precursors, are formed by fission and eventually decay, releasing a neutron. The precursors are lumped into six groups whose half lives describe the average times for half the precursors to decay, emitting delayed neutrons in the system. If all the neutrons that appeared in a reactor were prompt neutrons, nuclear reactors would be impossible to control due to the great speed of multiplication. Although only a small fraction of the population, the delayed neutrons allow control of the reactor by effectively increasing the average lifetime of the neutrons in the core. The point kinetics equations are generally used to model the power produced by a nuclear reactor:

$$\frac{dP}{dt} = [(\rho(t) - \beta)/\Lambda]P + \sum_{i=1}^{6} \lambda_i \xi_i \qquad (36)$$

$$\frac{d\xi}{dt} = (\beta_i/\Lambda)P - \lambda_i \xi_i \qquad \text{for } i=1,2,3...6 \qquad (37)$$

where P(t) is the power of the reactor, which is proportional to the neutron population, $\xi_i(t)$ is the number of group i precursor atoms, the fraction of the total fission neutrons that are delayed is β, the fraction which appear in group i is β_i, λ_i is the decay constant of the ith precursor group, Λ is the prompt neutron lifetime and $\rho(t)$ is the total reactivity in the system, which is proportional to the difference between the production and loss rates of neutrons. The system reactivity consists of "external" reactivity which is adjusted by control rods, and "feedback" reactivity arising from various feedback mechanisms. These are called the "point kinetics" equations, because the equations assume the neutron population in the reactor rises and falls without significant changes in the spatial distribution. To derive a transfer function relating the output power of the reactor and the system reactivity, these equations are linearized, and the Laplace transforms are taken.

To linearize the point kinetics equations, we will express the time-dependent variables as the sum of the initial value plus a time dependent deviation:

$$P(t) = P_0 + \delta P(t)$$

$$\xi_i(t) = \xi_{i\theta} + \delta\xi_i(t)$$

$$\rho(t) = \rho_0 + \delta\rho(t)$$

After substituting the above relations into Eq. 36 and neglecting second order terms one obtains

$$\frac{d\delta P(t)}{dt} = [(\rho_0 - \beta)/\Lambda)]\, P(t) + (\delta\rho/\Lambda)P + \sum_{i=1}^{6} \lambda_i\, \delta\xi_i \qquad (38).$$

Turning to Eq. 35, the same process is carried out yielding

$$\frac{d\delta\xi_i(t)}{dt} = (\beta_i / \Lambda)\, P(t) - \lambda_i\, \delta\, \xi_i \qquad \text{for } i=1,2,3...6 \quad (39).$$

Taking the Lapace transform of Eq. 39 and solving for the precursor variation gives

$$\delta\xi_i = [\, \beta_i /(\Lambda(s + \lambda_i))]\delta P \qquad (40)$$

where

$$\delta\xi_i = \delta\xi_i\,(s)$$

$$\delta P = \delta P(s)$$

Then taking the Laplace transform of Eq. 38 and using Eq. 40 one obtains

$$\delta P/\delta\rho = P_0/[s - \rho_0/\Lambda + (1/\Lambda)\sum_{i=1}^{6}((s\,\beta_i)/(s+\lambda_i))] \qquad (41).$$

When the reactor is initially critical ρ_0 is equal to zero yielding

$$\delta P/\delta\rho = P_0/[\Lambda\,s + \sum_{i=1}^{6}(s\,\beta_i)/(s+\lambda_i)] \qquad (42).$$

Eq. 42 is the zero power transfer function, so called because there is no accounting for feedback to the system. It is simply the relationship of the total system reactivity to the output power of the reactor.

Feedback to a nuclear reactor is the result of a number of different mechanisms which are directly related to the temperature of the fuel, coolant or structural material. As the temperature in these materials change, the reactivity is changed by several mechanisms. A cursory discussion of these effects is in order.

The Doppler effect is a negative reactivity insertion caused by the broadening of neutron capture resonances with an increase in fuel temperature. As the temperature in the fuel rises, the kinetic energy of the atoms increases and the relative speed of the neutron and target atom have a wider range in which the capture interaction can take place. This allows more neutrons to be captured in a particular resonance and therefore yields a negative coefficient of reactivity.

An increase in the fuel temperature causes the fuel to expand and its density to drop, increasing the neutron mean free path length. The increase in mean free path length increases neutron leakage and consequently reduces the number of fissions per fissionable atom. At EBR-II the net result is a negative insertion of reactivity with an increase in temperature.

Feedback reactivity due to the expansion of sodium is the result of greater neutron leakage, a harder neutron spectrum and a decrease in the macroscopic capture cross section, with an increase in temperature. The result is a negative reactivity effect at EBR-II. More detail on the way these effects interact and their net effect can be found in Larson (3).

The expansion of structural materials in the core have different effects on the core dynamics depending on their place in the core and respective functions. For example the grid plate at the bottom of EBR-II which holds the subassemblies in place expands when heated and causes a negative insertion of reactivity due to the greater distance between fuel subassemblies. The expansion of stainless steel in the core also causes a change in the absorption cross section and a change in the diffusion constant which adds a negative insertion of reactivity to the core. As the reflector around the core expands, more neutrons leak out also causing a negative insertion of reactivity. Some mechanisms of feedback reactivity which provide a positive coefficient are control rod drive line expansion, safety rod expansion, and the bowing of subassemblies. More detail on the mechanisms that cause feedback at EBR-II can be found in Larson (3). Table 1 gives the compiled coefficients for different feedback effects taken from Dean (2).

Table 1 Coefficients of feedback reactivity in EBR-II

Component	Values
1. Driver fuel expansion	-3.37×10^{-6} Δk/k/°C
2. Core sodium expansion	-7.98×10^{-6} Δk/k/°C
3. Core stainless steel	-1.56×10^{-6} Δk/k/°C
4. Lower axial reflector sodium expansion	-4.3×10^{-6} Δk/k/°C
5. Upper axial reflector sodium expansion	-4.3×10^{-6} Δk/k/°C
6. Radial reflector sodium expansion	-1.49×10^{-6} Δk/k/°C
7. Grid plate expansion	-9.86×10^{-6} Δk/k/°C
8. Control rod drive line expansion	0.1136 Δk/k/m
9. Safety rod expansion	0.0209 Δk/k/m
10. Thermal bowing at $T = 101.67$ °C	2.36×10^{-4}
11. Doppler coefficient	$-3.66 \times 10^{-4}/(T(°C) + 273)$

To calculate the feedback reactivity to the reactor, the next part of the core model, the thermal analysis must be considered. First we will write the simple lumped equations for the core. They are

$$m_f c_f \frac{\partial Tf(t)}{\partial t} = P(t) - hA(Tf(t) - Tc(t)) \qquad (43)$$

for the fuel, and

$$m_T c_T \frac{\partial Tc(t)}{\partial t} = hA(Tf(t) - Tc(t)) - \frac{m_c c_c v_c \; Tc(t)}{\partial z} \qquad (44)$$

for the coolant, where

$m_f c_f$ = mass of the fuel multiplied by the heat capacity of the fuel

$m_c c_c$ = mass of the coolant multiplied by the heat capacity of the coolant

$m_T c_T$ = mass of the coolant plus the structural steel and cladding in the core multiplied by the heat capacity of the coolant plus the structural steel and cladding in the core

Tf = temperature of the fuel

Tc = temperature of the coolant

h = convective heat transfer coefficient between the fuel and coolant

A = area between the fuel and the coolant and clad

v_c = average velocity of the coolant moving through the core

P = thermal power generated by the core

Carrying the same linearization process out on Eqn. 43 and Eqn.

44

with

$$Tf(t) = Tf_0 + \delta Tf(t)$$

and $Tc(t) = Tc_0 + \delta Tc(t)$

and integrating Eqn. 44 over the length of the reactor one obtains

$$m_T c_T \frac{d\delta Tc(t)}{dt} = hA(\delta Tf(t) - \delta Tc(t)) - \frac{m_c c_c v_c (To - Tin)}{L} \qquad (45)$$

where L is the length of the core, Tin is the constant inlet temperature to the core and To is the outlet temperature. Then with the assumption

$$To - Tin = 2(Tc - Tin)$$

where L is the length of the core, we obtain

$$m_f c_f \frac{d\delta Tf(t)}{dt} = P(t) - hA(\delta Tf(t) - \delta Tc(t)) \qquad (46)$$

$$m_T c_T \frac{d\delta Tc(t)}{dt} = hA(\delta Tf(t) - \delta Tc(t)) - \frac{2m_c c_c v_c \delta Tc(t)}{L} \qquad (47).$$

These equations can be rewritten as

$$\frac{d\delta Tf(t)}{dt} = \delta P(t)\phi - \omega_f(\delta Tf(t) - \delta Tc(t)) \qquad (48)$$

$$\frac{d\delta Tc(t)}{dt} = \omega_c (\delta Tf(t) - \delta Tc(t)) - \gamma \delta Tc(t) \qquad (49).$$

where

$$\phi = 1/(m_f c_f) \qquad (50)$$

$$\omega_f = hA/(m_f c_f) \qquad (51)$$

$$\omega_c = hA/(m_T c_T) \qquad (52)$$

$$\gamma = 2m_c c_c v_c/(m_T c_T L) \qquad (53).$$

Then when the Laplace transform of Eqn. 49 is taken we obtain

$$s\delta Tc = \omega_c(\delta Tf - \delta Tc) - \gamma\, Tc$$

which may be rearranged as

$$Tc(s + \omega_c + \gamma\ ) = \omega_c \delta Tf$$

$$\frac{\delta Tc}{\delta Tf} = \frac{\omega_c}{(s + \omega_c + \gamma\ )} \qquad (54)$$

where $\delta Tc = \delta Tc(s)$

and $\delta Tf = \delta Tf(s)$

and s is a complex variable. Eqn. (54) is the transfer function relating the coolant to fuel temperature. When the Laplace tranform of Eqn. 48 is taken one obtains

$$s\delta Tf = \phi\, P - \omega_f(\delta Tf - \delta Tc)$$

which by the use of Eqn. 54 may be written as

$$\delta Tf(s + \omega_f) = \phi\delta P + \frac{\omega_f\, \delta Tf\ \omega_c}{(s + \omega_c + \gamma)}$$

and ultimately as

$$\frac{\delta Tf}{\delta P} = \frac{\phi\ (s + \omega_c + \gamma)}{s^2 + s(\omega_f + \omega_c + \gamma) + \omega_f\, \gamma} \qquad (55)$$

which is the transfer function relating the output power of the reactor to the temperature of the fuel.

The component transfer functions to create a system transfer function relating the external reactivity to the output power have now been derived. The next step is to take transfer functions relating power to fuel and coolant temperature and use them to calculate the feedback reactivity. The system reactivity is defined by the equation

$$\rho = \rho_{ext} + \rho_{f.\beta.}$$

where the external reactivity is that reactivity which is inserted by a control rod and the feedback reactivity is that which is added to the system by a change in temperature of either the fuel or the coolant. The feedback reactivity in EBR-II is comprised of the contributing components in Table 1. Some of these components can not be used in a lumped model; e.g., assembly bowing due to feedback, safety rod expansion and control rod drive line expansion. We have now derived the transfer functions necessary to create a system transfer function relating external reactivity to the output power. The core can be represented by the block diagram in Fig. 9.

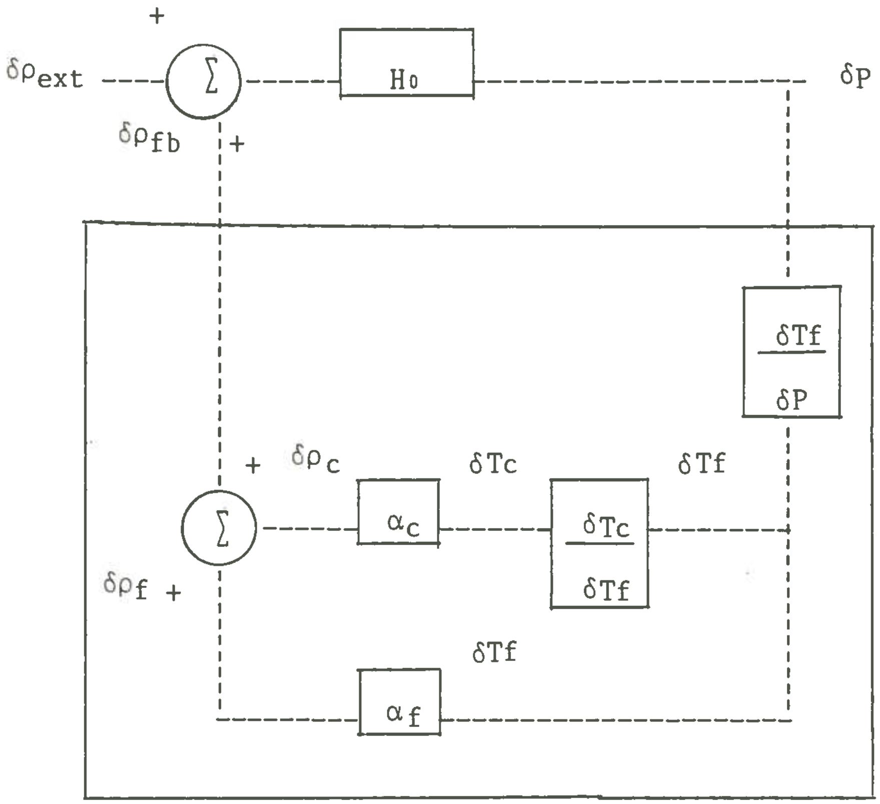

Fig. 9 Core system block diagram

where H_0 is the zero power transfer function, α_f is the total fuel feedback coefficient due to temperature, α_c is the total coolant feedback coefficient due to temperature. The total reactivity to the system, $\delta\rho$, is a result of the external reactivity and the feedback reactivity which are calculated by multiplying the change in the fuel and coolant temperatures, δTf and δTc. Fig. 9 can be simplified to the one in Fig. 10 if G_0 , the feedback transfer function, is defined as

$$G_0 = \frac{\delta\rho_{fb}}{\delta\rho} = (\alpha_c \frac{\delta Tc}{\delta T_f} \frac{\delta Tf}{\delta P} + \alpha_f \frac{\delta Tf}{\delta\rho}) \ \delta P$$

which by use of Eqns. 54 and 55 may be written as

$$G = \frac{s\psi + \chi}{s^2 + s\Omega + \eta} \tag{56}$$

where

$$\psi = \alpha_f \phi \tag{57}$$

$$\chi = \alpha_c \omega_c \phi + \alpha_f \phi \omega_c + \alpha_f \phi \gamma \tag{58}$$

$$\Omega = \omega_f + \omega_c + \gamma \tag{59}$$

$$\eta = \omega_f \gamma \tag{60}.$$

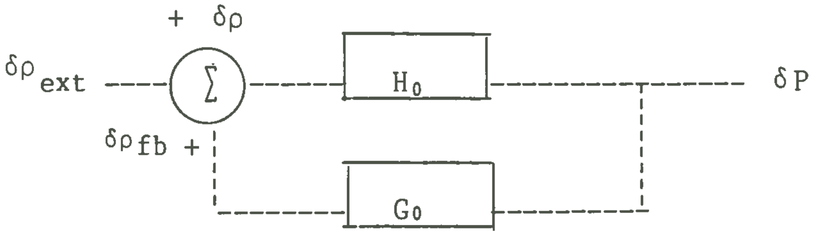

Fig. 10 Simplified core system block diagram

By Eqn. 20 of Chapter 3, the system transfer function is found to be

$$H = \frac{\delta P}{\delta\rho_{ext}} = \frac{H_0}{1 - H_0 G_0} \tag{61}.$$

To obtain a graphical representation of the zero power transfer function it is necessary to have the values of the prompt neutron lifetime, the delayed neutron fraction and precursor half lives for each group. These system parameters are given in Table 2.

Table 2 Transfer function parameters for EBR-II

delay group	β_i	λ_i
1	.000236	.0127
2	.001391	.0317
3	.001262	.115
4	.002743	.311
5	.000961	1.4
6	.000236	3.87

Prompt neutron lifetime = 1.1×10^{-6} second

To numerically calculate and obtain a graphical representation of the system transfer function, the unknowns for the feedback transfer function that have been previously derived must be calculated. The first parameter derived is the heat transfer coefficient in the core. An equivalent fuel pin subchannel for EBR-II is found in Fig. 11 from Dean (2).

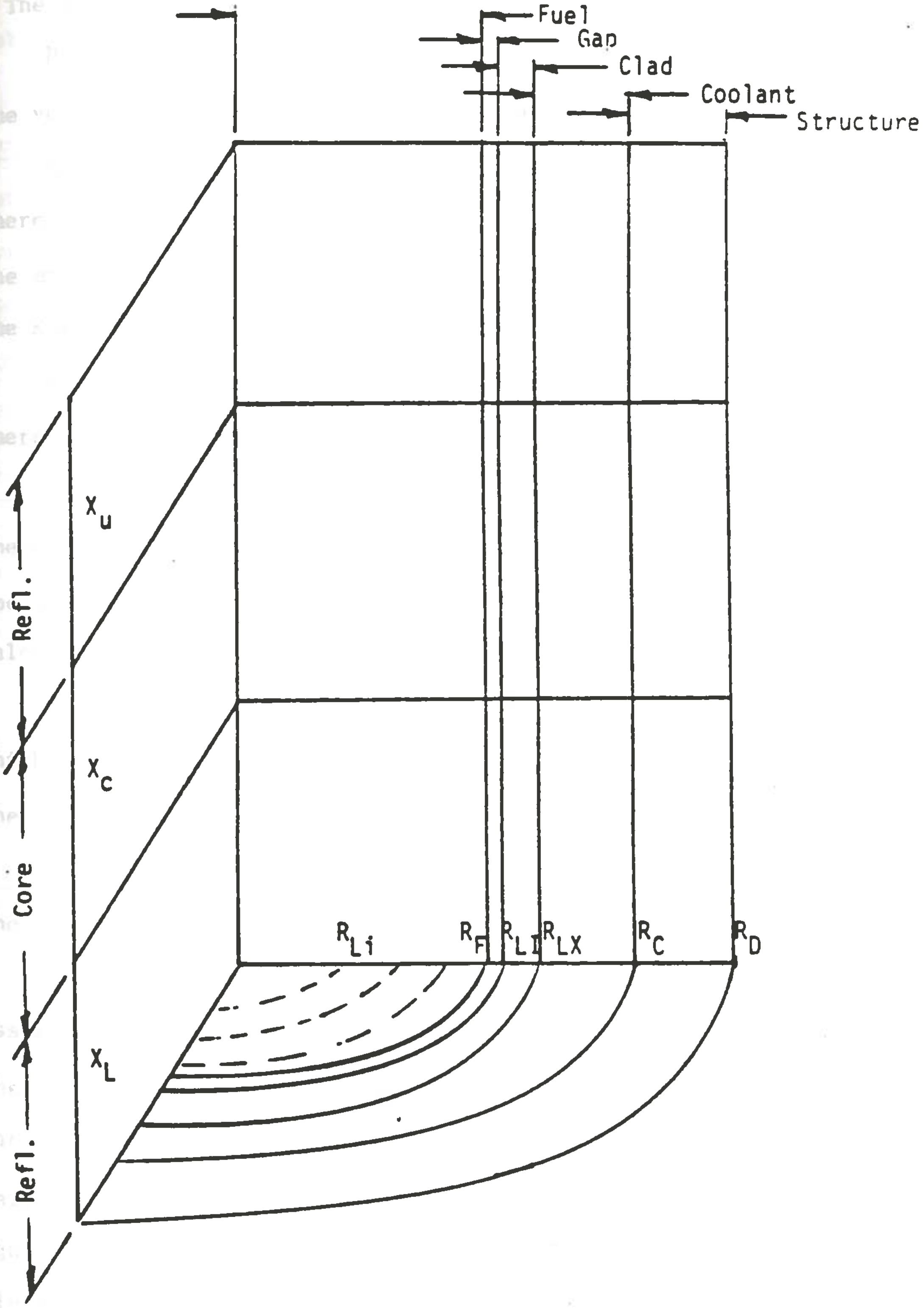

Fig. 11 Eqiuvalent fuel pin subchannel for EBR-II

The equivalent diameter is found by the equation

$$De = 2((R_c) - (R_{LX}))/(R_c + R_{LX}) \qquad (62).$$

The velocity of the coolant in the channel is given by

$$v_c = \dot{m}_c/(\rho_c A)$$

where $\dot{m}_c$ is the mass flow rate for the equivalent suchannel, ρ_c is the average density of the coolant, and A is the area of the clad. The Reynolds number is

$$Re = \rho_c v_c De/\mu$$

where μ is the viscosity of the coolant. The Prandtl number is

$$Pr = \mu c_p/k.$$

where k is the thermal conductivity of the coolant and c is the specific heat of the coolant. The Nusselt number for EBR-II is calculated using the equation

$$Nu = 5.0 + .025(Re\ Pr)^{0.8}$$

which allows the calculation of the heat transfer coefficient with the equation

$$h = Nu \times k/De$$

The area of the fuel pin is given by the equation

A = number of assemblies multiplied by the number of pins per assembly multiplied by the area per pin

There are 79 assemblies and 91 pins per assembly. Using the parameter values for materials in the core found in Table 3 we can calculate the value of the parameters found in Eqns. 50 to 53 and Eqns. 57 to 60 found in Carboneau (4). These calculated values are given in Table 4.

Table 3 Core material parameters of EBR-II

Parameter	Sodium	Fuel	Stainless Steel	Units
Total mass	53.3	408.9	383.4	kg
Average temperature	432	524	443	°C
Specific heat	1270	183.65	460.5	J/(kg-°C)
Conductivity	69.7	17.18	20.2	W/(m-°C)
Density	846.7	19300	7800	kg/m^3
Viscosity	.00026			Pa/s

All temperature dependent quantities are evaluated at the given average temperature.

Table 4 Parameter values defined for the feedback transfer function of EBR-II

Parameter	Value	Units	Parameter	Value	Units
ϕ	13.317×10^{-6}	°C/J	ψ	$-5.100411 \times 10^{-11}$	$\Delta k/k/J$
ω_f	143.795	sec^{-1}	χ	-2.002555×10^{-8}	$\Delta k/k/J\text{-}s$
ω_c	44.217	sec^{-1}	Ω	1143.139	sec^{-1}
γ	7.94978	sec^{-1}	η	195.9618	sec^{-2}

The next subsystems for which transfer functions will be obtained are the assembly section above the core in Fig. 8 and a section of the Z pipe shown in Fig 12. These two flow channels can be modeled by the general differential equation in Eqn. 63 which represents a pipe that can have energy transported through the pipe by the coolant flow and out the sides by the conduction. For the Z pipe:

$$m_c c_c \frac{dT}{dt} = \dot{m}_c c_c (Ti - To) - Ah(T - Tb) \qquad (63)$$

where

T = average coolant temperature inside the pipe

Ti = inlet coolant temperature

To = outlet coolant temperature

Tb = bulk temperature of the fluid outside the channel

$m_c c_c$ = mass of coolant specific heat of coolant

A = area of the flow channel

The average coolant temperature will be approximated as

$$T(t) = (To(t) + Ti(t))/2.$$

Thus Eqn. 63 becomes

$$m_c c_c \frac{dTo(t)}{dt} + m_c c_c \frac{dTi(t)}{dt} =$$

$$2\dot{m}_c c_c (Ti(t) - To(t)) - hA(T(t) - Tb)$$

or

$$\frac{dTo(t)}{dt} + \frac{dTi(t)}{dt} =$$

$$2(\dot{m}_c/m_c)(Ti(t) - To(t)) - (hA/m_c c_c)(To(t) + Ti(t) - 2Tb).$$

Lineraizing this equation and taking the Laplace transform of the result yields

$$s\delta To + s\delta Ti = 2(\dot{m}_c/m_c)(\delta Ti - \delta To) - (hA/m_c c_c)(\delta To + \delta Ti)$$

or $$\delta To/\delta Ti = (-s + 2\dot{m}_c/m_c - hA/\dot{m}_c c_c)/(s + 2\dot{m}_c/m_c + hA/m_c c_c) \qquad (64).$$

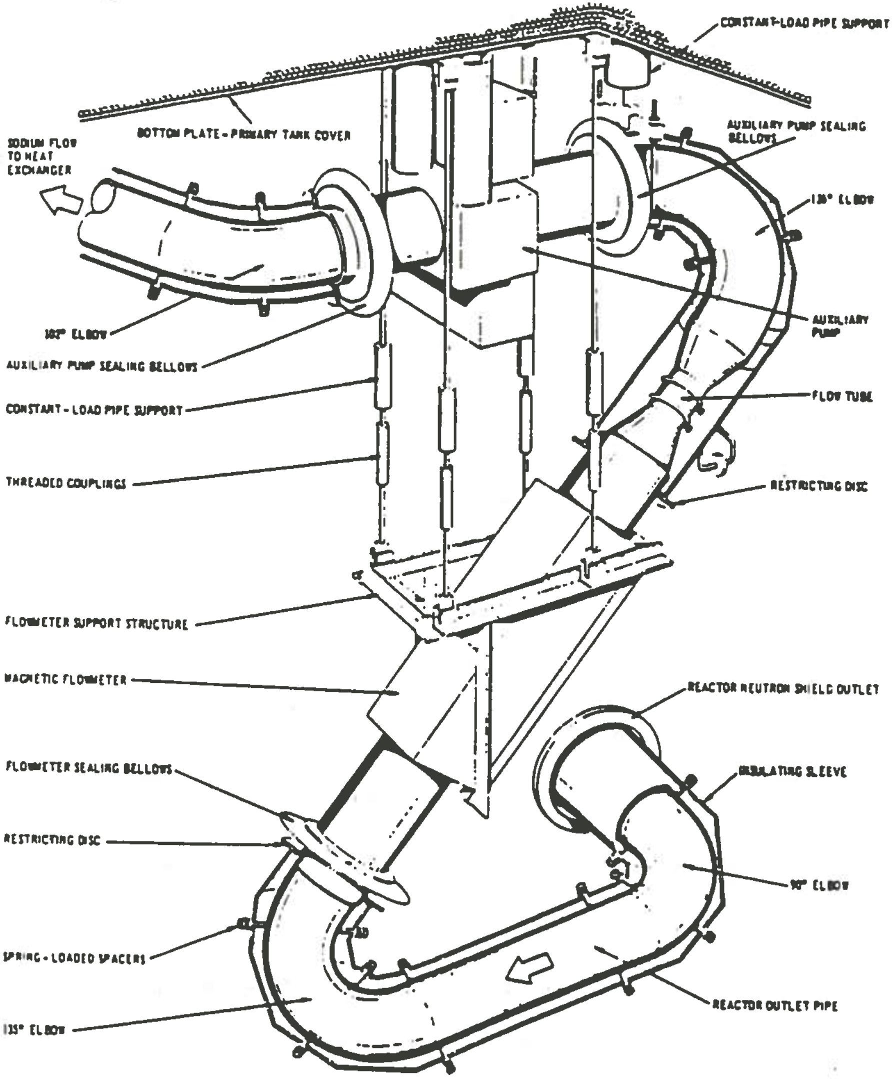
CONSTANT-LOAD PIPE SUPPORT
BOTTOM PLATE - PRIMARY TANK COVER
SODIUM FLOW TO HEAT EXCHANGER
AUXILIARY PUMP SEALING BELLOWS
135° ELBOW
102° ELBOW
AUXILIARY PUMP
AUXILIARY PUMP SEALING BELLOWS
CONSTANT - LOAD PIPE SUPPORT
FLOW TUBE
THREADED COUPLINGS
RESTRICTING DISC
FLOWMETER SUPPORT STRUCTURE
MAGNETIC FLOWMETER
REACTOR NEUTRON SHIELD OUTLET
FLOWMETER SEALING BELLOWS
INSULATING SLEEVE
RESTRICTING DISC
90° ELBOW
SPRING - LOADED SPACERS
REACTOR OUTLET PIPE
135° ELBOW

Fig. 12 Reactor outlet pipe "Z pipe"

For the channel above the core

$m_T c_T$ = mass of coolant multiplied by the specific heat of coolant plus mass of stainless steel multiplied by the specific heat of stainless steel

This yields

$$\delta To/\delta Ti = (-s + 2\dot{m}_c c_c/m_T c_T - hA/m_T c_T)/(s + 2\dot{m}_c c_c/m_T c_T + hA/m_T c_T) \quad (65)$$

These transfer functions have one input and one output and are of the form in Fig. 13.

Fig. 13 Pipe or fluid flow channel transfer function

The last subsystem in the primary system is the intermediate heat exchanger shown in Fig. 14. This unit is a counterflow heat exchanger where the coolant flows into the heat exchanger through the primary sodium inlet, down the support slats and out into the pool. The secondary sodium travels down the central pipe and is heated on the way up in the heat exchanger tubes by the hot primary sodium flowing down. An attempt to derive an analytical transfer function for this complex system will not be made here. An experimentaly determined transfer function, and one that is calculated from the DSNP simulation results will be used for this system.

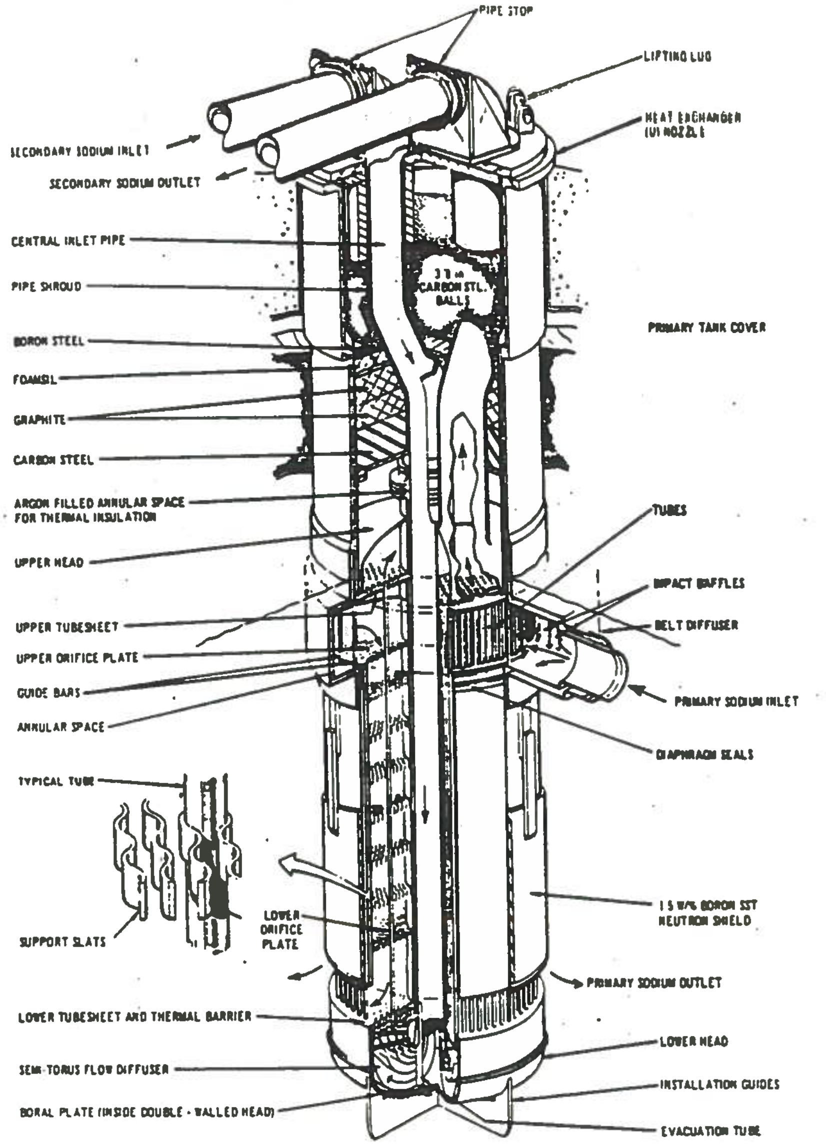

Fig. 14 Intermediate heat exchanger

Chapter Five

Previous System Response Testing Expierience at EBR-II and the SHRT Test Description

The analytical method used in Chapter 4 to calculate transfer functions is based on known system parameters. Experimental system response testing is done by applying the appropriate forcing function, and then recording the output of the system. The Laplace transforms or the Fourier transforms, if the system is at steady state, are taken of the input and output signals. The ratio of the transformed output over the input yields the transfer function of the system of interest. In this chapter, the different types of signals that have been used at EBR-II in the SHRT program to perform dynamic analysis will be examined. The data from the SHRT will be analyzed using the Fourier methods described in Chapter 3.

As stated in Larson (5), reactor-dynamics experiments are performed at EBR-II to investigate and interpret dynamic characteristics of the reactor; to insure that the system transfer function does not undergo major changes; and to identify trends in kinetic behavior. The two methods of measuring transfer functions at EBR-II since its inception have been the rod drop and the rod oscillator tests.

The rod drop test has been used more often for frequency response testing at EBR-II. The rod drop test at EBR-II is done by dropping a stainless steel rod out of the core and then recording the power decrease on the DAS (Data Acquisition System) controlled by the SIGMA V computer. Inverse kinetics is then used to extract the reactivity in the system as described in Larson (5). This test is actually an attempt to measure the step response of the reactor by simulating the Heavyside step function and therefore should excite all frequencies. However, there is only limited power in an input signal, and this test distributes the power over a large range of output frequencies thereby placing insufficient power in some frequencies for an adequate signal to noise ratio. The situation can be improved by multiple rod drops; however this necessitates bringing the reactor back to its original power and letting it "settle out". More detail on this subject can be found in Larson (5).

The rod oscillator test is done at EBR-II with a special Mark IIB oscillator rod driven by the ACRDS (Automatic Control Rod Drive System). The idea of this test is to excite only one frequency of the system at a time. For a linear system at steady state if one drives the system with an input of

$$x(t) = A\sin(\omega t) \tag{66}$$

the steady state output will be of the form

$$y(t) = A|H(j\omega)|\sin(\omega t + \phi) \tag{67}$$

where $|H(j\omega)|$ is the magnitude of the transfer function at frequency and ω the variable ϕ is the phase shift of the transfer function. The total transfer function can be evaluated by performing this test at different frequencies. Compiling a transfer function with this method requires repeating the test at every frequency of interest, and thus takes a great deal of time.

Another type of rod oscillator test closely related to the rod oscillator is the multifrequency test. In this test, the rod is inserted and withdrawn in a pattern which is the summation of several sinusoids as shown in Eqn. 67.

$$I(t) = \sum_{i=1}^{N} A_i \sin(\omega_i t) \qquad (68)$$

The advantage of this test is that it excites more than one frequency at the same time, and therefore saves time. The assumption that EBR-II is a linear system for small perturbations allows the superposition of these sinusoidal signals upon one another. This multifrequency test was run as part of the SHRT program at EBR-II.

As decribed by Feldman and Mohr in (6), the SHRT tests were designed to demonstrate inherent safety mechanisms which should be incorporated into LMFBRs and to provide an extensive data base for use in the development and validation of whole plant simulations. The first of these tests was the multifrequency reactivity perturbation mentioned previously. Its express purpose was to determine the frequency response characteristics of the EBR-II plant. Results from this test are meant to be benchmark analytical models and computer solutions. The test introduces two multifrequency perturbations: a low frequency set and a high frequency set. Both are of the form

$$I(t) = A \sum_{i=1}^{5} \sin\left(\frac{2\pi n_i t}{T} + \phi_i\right) \qquad (69)$$

where A = 1.59 cents of reactivity. The fundamental period for the low frequency test is 512 seconds and for the high frequency test is 512/11 or 46.545 seconds. The five component frequencies for each test and their phase shifts are given in Table 5.

Table 5 Frequencies and phase shifts of SHRT multifrequency signal

			T=512 sec	T=512/11 sec
i	n_i		n_i/T,Hz	n_i/T,Hz
1	1	50.681	0.001953	0.02148
2	3	240.000	0.005859	0.06445
3	5	330.000	0.009766	0.10740
4	11	50.000	0.021480	0.23630
5	25	350.000	0.048830	0.53710

These parameters give a reactivity variation of ±5 cents, which is provided by the ACRDS (Automatic Control Rod Drive System).

In the design of a multifrequency test, the number of frequencies included in the test must be limited so that there is enough power in each of the component frequencies. The entire frequency range of interest is covered by these two sequences. The phases of the sinusoids are chosen to minimize the difference between the maximum and minimum reactivity inserted in the system. There are no even harmonics, since odd harmonics produce an antisymmetric signal which descriminates against nonlinearities. Because the fundamental frequency has a period of 512 seconds and the data recording rate of the DAS is 2 bits per second, the number of data points for one cycle will be 1024. The number of data points is a power of two, which makes use of the Fast Fourier program based on a power of two possible. Notice also that all frequencies considered are harmonics of the fundamental. A test which analyzes an integral number of cycles for the fundamental will always analyze an integral number of cycles of all the harmonics, which is one of the requirements for using the FFT based on a power of two data points.

There are certain limitations on the test that are a function of the data acquisition rate. As was stated earlier, the data at EBR-II is recorded at 2 Hz or 2 samples per second. Therefore the Nyquist frequency at EBR-II is 1 Hz which means that the maximum frequency at which the system may be perturbed is 1 Hz. One must sample at least twice as fast as the highest frequency to avoid the problem of aliasing. It is advisable to sample at 4 times the highest frequency as a rule of thumb. Since the highest frequency in this test is .5371 Hz, this rule has basicaly been followed. Another limitation on the ability to record data is the time constant of a given instrument in the reactor itself. This is beyond the control of the experimentalist in this case since most of the instruments are immersed in liquid sodium, but the constraint must be kept in mind when analyzing the data and when preparing the simulation of the experiments with DSNP.

In the next section, the EBR-II core and plant systems for which data is recorded will be considered. The DAS system records data at 20 different places in the primary system. These are listed in Table 6 on the following page.

Table 6 Data Acquisition System instruments and channel numbers

Instr. #	Description of instrument	
1	Wide range linear power level	(MW)
2	Ion chamber level	(%)
3	Control rod position	(inches)
4	Excess (system) reactivity	(cents)
5	ACRDS rod velocity	(inches/sec)
6	ACRDS rod velocity demand	(inches/sec)
7	ACRDS position demand	(inches)
8	ACRDS power channel 7A	(MW)
9	XX09 TTC-32 coolant temperature at core top	(degrees F)
10	XX09 TTC-33 coolant temperature at core top	(degrees F)
11	XX09 TTC-47 coolant temperature at core top	(degrees F)
12	XX10 14TC-15 coolant temperatue above core	(degrees F)
13	IHX (Intermediate Heat Exchanger)	
	secondary inlet temperature	(degrees F)
14	IHX secondary outlet temperature 546A	(degrees F)
15	IHX secondary outlet temperature 533AA	(degrees F)
16	Subassembly outlet temperature 1A1	(degrees F)
17	High pressure plenum sodium temperature 540AT	(degrees F)
18	Low pressure plenum sodium temperature 540AS	(degrees F)
19	Upper plenum probe temperature 521D	(degrees F)
20	XX09 OTC-02 outlet coolant temperature	(degrees F)

In Chapter 4, we analytically developed some of the transfer functions of physical components in the primary system at EBR-II. The next step is to examine the data that can be collected and decide for which parts of the system it is possible to calculate transfer functions. From the data that we are able to collect, block diagrams can be created with each of the blocks corresponding to a different transfer function. Let us first examine the core system.

In the core, data on the power production is collected on channel number 1. From this information, the SIGMA V computer calculates the system reactivity, which is output on data channel number 4. On channel 3, the rod position is recorded; knowing the worth of the rod, this information may be converted to a reactivity reading. The external reactivity can then be subtracted from the system reactivity to yield the feedback reactivity over time. From this information the transfer function relating external reactivity to power shown in Fig. 15 can be calculated. The feedback reactivity to power transfer function, G_0 , shown in Fig. 15 can also be calculated. In this case external reactivity corresponds to the rod being driven and the feedback reactivity can be calculated by subtracting the external reactivity from the system reactivity. These two transfer functions are ones that the rod oscillator test has been used to calculate in previous runs at EBR-II.

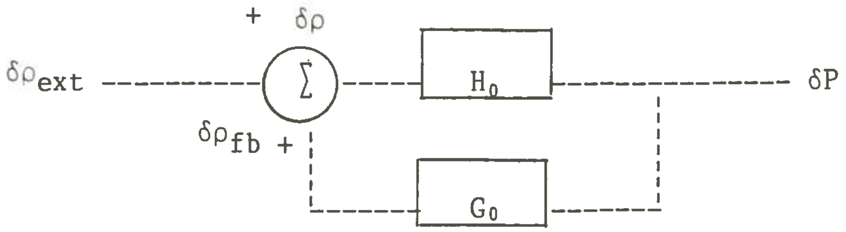

Fig. 15 External reactivity to power transfer function

The next transfer function that will be calculated is the temperature at the top of core to external reactivity transfer function shown in Fig. 10. This is calculated using the temperature at the top of the core measured by instrument number 10 and the external reactivity. This transfer function is also one for which we have derived an analytical model.

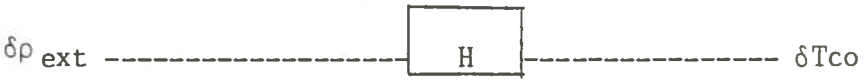

Fig. 16 External reactivity to temperature transfer function

The next two transfer functions that can be calculated are temperature transfer functions. The first is the assembly outlet coolant temperature, measured with instrument number 20, to core top temperature measured with instrument number 10. The second is the secondary outlet temerature of the intermediate heat exchanger, measured using instrument number 14, to outlet coolant temperature measured with instrument number 20. They are simple transfer functions shown in Fig. 17.

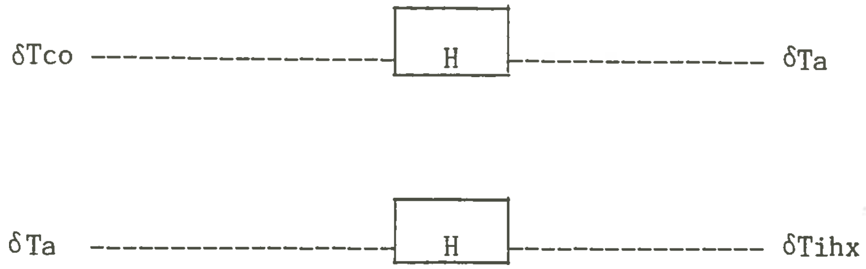

Fig. 17 Temperature to temperature transfer functions

To calculate the transfer functions of interest described in this chapter from the SHRT data, it was neccessary to write a program that reads data collected by the DAS and perform Fourier analysis with the code listed in Appendix A. This code must read an integral number of cycles of data points and skip past one cycle of the harmonic so that transients will die out. This causes the analysis to yield the Fourier transform instead of the Laplace transform.

This chapter completes the description of the SHRT test and the planned analysis of the results. It should be noted that there will be a time constant associated with each of the instruments and thermocouples used to measure temperature. Most of the these time constants are on the order of milliseconds; however the thermocouple on the exit of the secondary side of the intermediate heat exchanger is approximately 8 seconds. Compensation for this fact will have to be made in the DSNP simulation of the experiments. It should be kept in mind that the instruments' time constant fixes the maximum frequency at which it can read data. This is analagous to the low pass characteristics of a resistor and capacitor network. The cutoff frequency of such a network is given by

$$f_c = 1/(2\pi\tau) \tag{70}$$

where τ is the time constant of the instrument. The results of the analysis on the SHRT data will be discussed in Chapter 7.

Chapter Six

DSNP Simulation of EBR-II and the SHRT Test

DSNP is a FORTRAN based real time simulation computer code designed to model nuclear reactor systems specifically and dynamic systems in general. DSNP has its own language which is a set of simple statements that call modules, macros and subprograms and then assemble them into a FORTRAN source code which simulates the selected systems. DSNP has different libraries which contain different levels of complexity of components found in reactor plants. It also provides a library for user constructed modules. In addition to the modules, DSNP has many other basic capabilities such as function generators, connectors and functions capable of calculating properties of materials commonly found in reactor situations. To use DSNP, one has only to draw a block diagram and write a DSNP program with the commands that call the appropriate modules. DSNP has three basic sections: BEGIN, SIMULATE and TERMINATE. The static calculation of variables based on initial conditions in the system is in the BEGIN section of the code. The SIMULATE section is where the dynamic analysis of the system is performed and the TERMINATE section of the code is where print statements and any other functions are placed which are desired after the simulation is over.

The DSNP primary system model of EBR-II was provided by Eric Dean and is shown in Appendix C and documented in great detail by Dean in (2). However a basic description of the input and equations solved by the code is given here.

The DSNP simulation is divided into six basic component types: primary tank, reactor outlet pipe or "Z pipe," intermediate heat exchanger, reactor inlet piping, reactor core model and the reactor outlet plenum with high pressure inlet plenum and low pressure outlet plenum.

The DSNP model of the pool is a hydraulic cavity and mixing plenum combination which is created by using the modules "CAVIT1" and "MPPLE1." The CAVIT1 module when included in the main program calculates the sodium level in the primary tank, pressure at the intermediate heat exchanger outlet, pressure at the two primary pump inlets, and the pressure at two leakage paths which have been built into the module to leak a calculated amount of sodium which is known to leak from the primary cooling system into the tank. The MPPLE1 module simulates the thermodynamics of a mixing plenum using a metal wall and fluid node.

The Z pipe system is simulated with the module "TPIPS1" which models a fluid flowing through a pipe that has two insulating layers of different materials with a layer of stagnant sodium in between. The differential equations for creating this module are solved numerically with a user specified number of nodes.

The intermediate heat exchanger is simulated with a module called "GIHXA1." This module simulates the heat transfer process found in a counterflow heat exchanger. It employs flow dependent weighting factors to the partial differential equations describing the heat transfer process. The two output variables of interest are the primary output and secondary output temperatures. This module also employs a user specified number of nodes to solve the equations in question.

The outlet, high pressure inlet, and low pressure inlet reactor plenums are all solved with the module "MPPLE1" like the sodium pool itself. The reactor inlet piping are modeled as pure time delays without heat loss.

The core model used in DSNP is shown in Fig. 18 by means of a flow chart which is found in Dean (2).

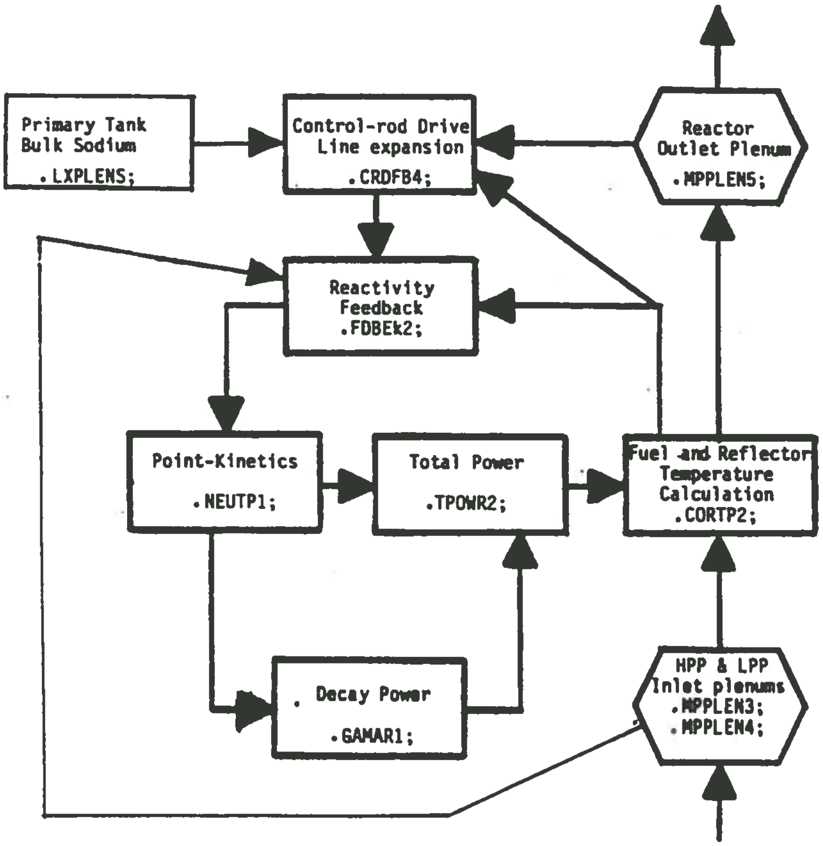

Fig. 18 Block diagram for DSNP core model

The DSNP core model solves the point kinetics equations described in chapter 4 in a module named "NEUTP1," yielding the thermal power of the reactor. The total reactor power distribution is then calculated in "TPOWR2" by adding the decay power of the fission products in the reactor calculated in "GAMAR1." The decay power in "GAMAR1" is calculated based on the degree of burnup in a particular core. The total power from "TPOWR2" is combined with the input plenum information in "CORTP2" to yield the temperature of materials in the core. These temperatures are then used to calculate the feedback reactivity to the reactor from different physical mechanisms. These mechanisms have been detailed in chapter 4 and are listed in Table 1.

The DSNP simulation of EBR-II is modified to produce the SHRT multifrequency test by inserting the reactivity perturbation in the SIMULATE section of DSNP. The variable that is used to do this is called "RKCN" and is set equal to a summation of sinusoids with the proper magnitudes, frequencies and phases given in Table 5. To compare these transfer functions with ones calculated with the SHRT data, the output variables in DSNP must be chosen so that they yield values for variables at the approximate positions of the corressponding instrumentation in EBR-II. The simulation results will also be produced and saved like the SHRT data, at half second intervals. This will allow the calculation of the transfer functions in Figs. 15 through 17 by using the program in Appendix B which has been slightly modified to read DSNP data.

As stated in Chapter 5 it is neccessary to compensate for the large time constant at the exit of the secondary of the intermediate heat exchanger. This can be done easily in DSNP by use of the "LAG" function. The "LAG" function is given by

$$Y = X/(s\tau + 1) \tag{71}$$

where s is a complex variable, X is the input parameter and for the instrument in this case is 8 seconds. The name of this function gives the connotation of time delay; however, τ is a time constant and not a time delay parameter. Notice that this will yield an integrated heat and consequently a higher temperature than normal. The results of the calculation of transfer functions using DSNP generated output, SHRT experimental data and analytical expressions are compared in the next chapter.

Chapter Seven

Results of the Analytical Derivations, Analysis of the SHRT Test and DSNP Simulation of the SHRT Test

This thesis analyzes the dynamic behavior of the primary system of EBR-II by (a) using analytical lumped model calculations (b) analyzing experimantal data collected from the SHRT multifrequency test and (c) simulating the SHRT test numericaly with DSNP. The results of these efforts are presented in this chapter in the order of occurence of subsystem in the primary system.

The first of these is the transfer function relating output power of the reactor to external reactivity. The results of the calculation of this transfer function are in Fig. 19 through Fig. 22, along with the zero power transfer function. The results for this transfer function are excellent. At the lowest frequencies the maximum difference between SHRT and DSNP results in magnitude is approximately 15%. At the highest frequencies this converges to about 10%. Notice that there is a difference between the rod oscillator results and those obtained from the SHRT test. This difference can be as much as 11% at .018 Hz. This is not surprising since different core loadings will cause different amounts of feedback reactivity. However it does put a different light on the DSNP results, indicating that they are really quite accurate when one considers that the experimental measurements can vary this much. The analytical expression also yields surprisingly accurate results. Its greatest divergence from experimental results is about 11% at around .007 Hz. The phase response results for this transfer function are also very good except at low frequencies where the error between DSNP and SHRT experimental measuremants is over 50%. However, once again notice that there is a 40% difference between the phase shifts in the analysis of experimental runs at .006 Hz.

The poles and zeros for the analytical transfer function are shown in Table 7. Notice that they are all real and negative and occur alternately in frequency starting with a pole.

Table 7 Poles and zeros of the analytical system transfer function of EBR-II

Poles		Zeros	
Imaginary	Real	Imaginary	Real
0.0	-.00735	0.0	-.0127
0.0	-.01659	0.0	-.0317
0.0	-.07727	0.0	-.1150
0.0	-.20728	0.0	-.3110
0.0	-1.2455	0.0	-1.400
0.0	-3.7838	0.0	-3.860
0.0	-6.7517	0.0	-6.018
0.0	-189.51	0.0	-189.9
0.0	-62073.		

Having all real and negative roots yields a series of decaying exponentials for the impulse response in the time domain. Notice also that there are nine poles and only eight zeros and that there is a pole on each side of the end point zeros. This configuration insures stability.

The next transfer function evaluated is the feedback transfer function. The results for this analysis are shown in Figs. 23 through 26. Unfortunately, the SHRT experiment data analysis for this transfer function produced meaningless results. It shows feedback reactivity increasing with increasing frequency which one can tell from the analytical derivation and rod oscillator data is not true. The phase angle results are even worse and are not plotted. A likely reason for this behavior is the computer program by which the system reactivity is calculated. It is an inverse kinetics program which estimates the system reactivity by using the measured power. The feedback reactivity is then calculated by subtracting the external reactivity from the system reactivity, which is then used to calculate this transfer function. Eric Dean, in the Operations Analysis section at EBR-II, has suggested that this program may have used too small a step size in time to produce accurate results and apparently was suspect since it has been subsequently replaced. The DSNP calculation compares very well with the rod oscillator data, however. The magnitude is accurate over the whole range of frequencies of interest. The only inaccuracy in the phase is one data point at around .008 Hz. The analytical derivation is about 25% low and breaks a decade later than the DSNP and rod oscillator data.

The error in the break frequency of the analytical solution is also observed in the phase response. The phase shift in the feedback transfer function can be written as

$$\theta(j\omega) = \tan^{-1}(\omega\psi / \chi) - \tan^{-1}[\omega\Omega /(\eta - \omega^2)] \qquad (72).$$

If we assume that the phase shift of the feedback transfer function should be about zero at low frequencies, the negative phase shift shown in Fig. 23 and Fig. 25 occurs because of the second term in Eqn. 72. The expression in this term can be written as

$$\tan^{-1}[\omega(\omega_f + \omega_c + \gamma)/(\omega_f\gamma - \omega^2)] \qquad (73).$$

This shows that the only parameter that can give us more of a phase shift would be ω_c. This parameter is a function of the area of the fuel, the heat transfer coefficient, and the mass of what has been called the coolant. The value of ω_c would be a higher number if the stainless steel was not included in this parameter. It behooves us at this point to examine the feedback transfer function that was derived in Chapter 4 to see what other effects system parameters may have on the feedback transfer function.

The transfer functions in this thesis have been derived using published values of parameters in EBR-II. These values are subject to change from run to run and error. To understand the effect of changing parameters on the feedback transfer function, the sensitivity coefficients relating the magnitude of the feedback transfer function for different changes in system parameters can be found in Table 8. These coefficients not only tell one how important a particular parameter is to a system, they also indicate which parameters to vary to obtain more or less feedback to the system. Some parameters having large sensitivity coefficients are the length of the core, the coolant feedback coefficient, and the coolant mass. Notice that feedback coefficients have a fairly constant effect on the feedback transfer function. They appear in the constant term of the numerator of the expression for the magnitude of the feedback transfer function. The effect of the mass of the coolant, on the other hand, drops off with an increase in frequency. This is because the squared term in the denominator reduces the effect of changing the coolant mass as one increases .

Table 8 Relative sensitivity coefficients for the magnitude of the feedback transfer function at EBR-II

Parameter	.001 Hz	.010 Hz	.100 Hz	1.00 Hz	5.00 Hz	9.00 Hz
Coolant mass	-.878	-.878	-.878	-.874	-.635	-.436
Fuel mass	0.0	0.0	0.0	0.0	-.09	-.17
Stainless steel mass	0.0	0.0	0.0	-.013	-.210	-.355
Core length	+.978	+.978	+.978	+.941	+.439	+.097
Coolant velocity	-.019	-.019	-.019	-.022	-.114	-.180
Fuel area	-.019	-.019	-.019	-.019	-.016	-.012
Heat transfer coefficient	-.019	-.019	-.019	-.022	-.114	-.180
Fuel feedback coefficient	+.135	+.135	+.135	+.135	+.136	+.136
Coolant feedback coefficient	+.867	+.867	+.867	+.867	+.864	+.864

Units are consistent with those found in Chapter 4.

The results of the transfer function relating the temperature of the coolant at the top of the core to the external reactivity are shown in Figs. 27 through 30. They show that the DSNP and SHRT experimental analysis match very well in magnitude though the analytical derivation is about 45% off. This may be lessened again by only treating the sodium and leaving the stainless steel out. It has also been suggested that there is probably less mass in the core than the nominal amount found in Carboneau (4). Since this transfer function magnitude varies as one over the fuel mass, this discrepancy would have a direct effect on the results. The phase response for DSNP and the SHRT analysis can be as much as 30% different in phase. The analytical phase response again breaks at a decade higher than the DSNP and experimental results. Once again the active term in the expression is ω_c. It should be remembered that the experimental phases appear to be able to be off as much as 50% from run to run.

The temperature at the top of the assembly to temperature at the top of core transfer function is shown in Figs. 31 through 34. The results are excellent for the DSNP and SHRT experiment. However, the analytical model results are not very accurate for this channel. In the equation modeling this system, the term with the heat transfer coefficient for the sodium channel was obtained from experimental data since it is very difficult to derive it from basic data because of the complexity of the system. The analytical model magnitude is good at the low frequencies to about 10% but as usual breaks at a later frequency. The transfer function for a section of the Z pipe is shown in Figs. 35 through 39. The analytical model is more accurate for the pipe since the previously mentioned heat transfer term can be derived using the known conductances and thicknesses of the pipe and insulating material.

The transfer function that relates the output temperature of the secondary side of the intermediate heat exchanger to the input temperature of the primary is shown in Figs. 44 through 47. It was created by subtracting the transfer function of the Z pipe as calculated by DSNP from the experimental transfer function using SHRT data that relates the output temperature of the secondary side of the heat exchanger to the temperature at the top of the core (shown in Figs. 40 through 43). Subtracting a subsystem off of a larger system is done by simply dividing the larger system by the smaller system transfer function. This had to be done because there is no way to measure temperature going into the primary side of the heat exchanger. The results show an error of up to 30% error in magnitude as a maximum and 50% error in phase. This error is attributable to the dispersion of the driving function signal. Measuring the transfer function of a system with a small signal that is this far away from the signal itself is pushing the limits of the signal to noise ratio. Notice that at about .06 Hz the experimental data analysis shows that the transfer functions magnitude begins to rise. This behavior is caused by the inability of the instrument to collect data at these higher frequencies. This frequency should be determined by the time constant of the weld and instrument as explained in Chapter 4. The break frequency determined experimentaly may be lower than one would expect. The experimental break frequency is about .0065 Hz. The cutoff frequency of the instrument is .02 Hz. However, the cutoff frequency is not a hard break and one would expect to see some distortion of the results even before this point. There also may be an error due to determining the time constant of the instrument. Notice that as the frequency goes above the instrument's ability to respond to it, the magnitude of the response goes up and not down as would be expected of a low pass filter. This is because the noise of the reactor is now being treated like the output signal and results in a higher magnitude at all frequencies.

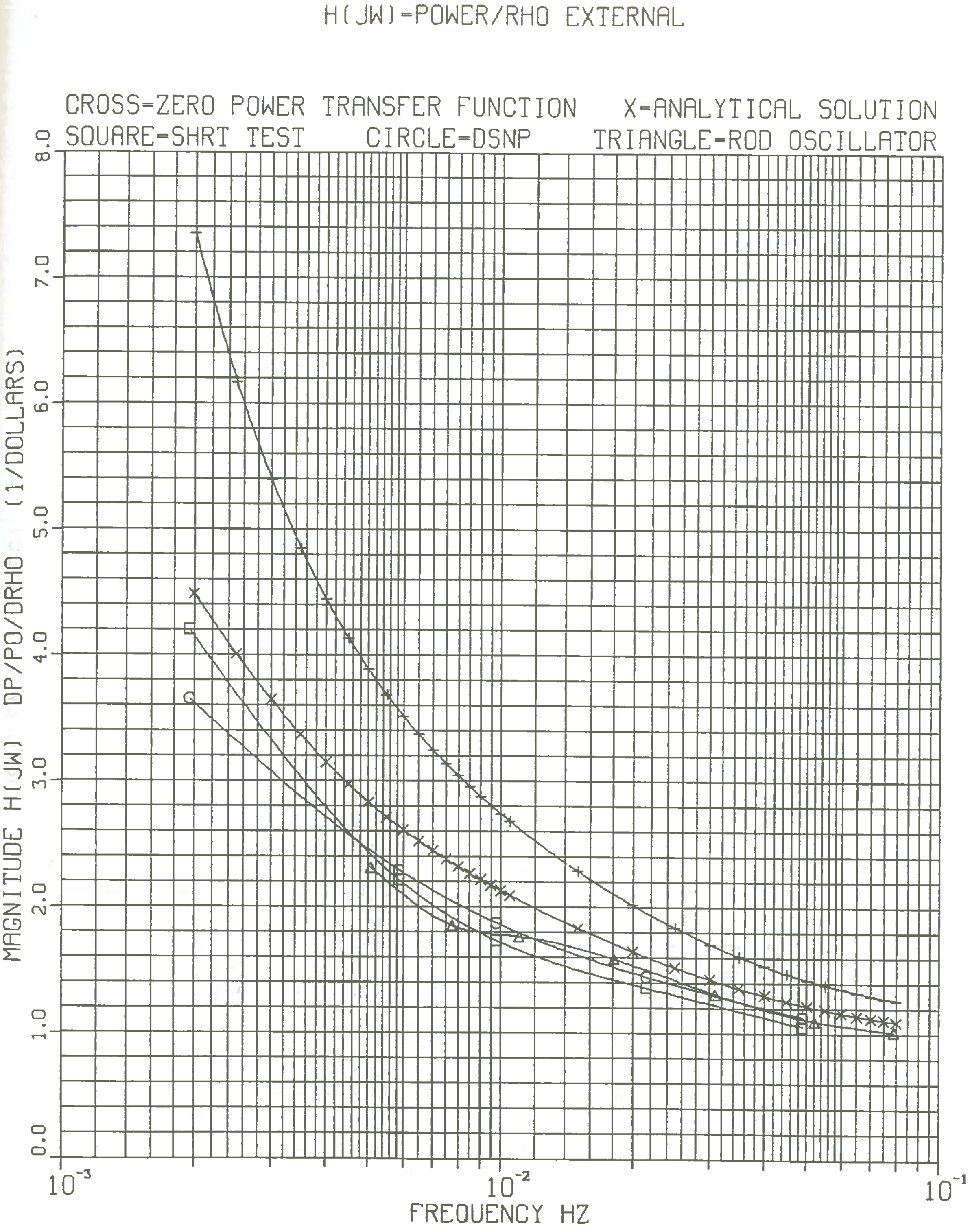

Fig. 19 System transfer function magnitude for low frequencies

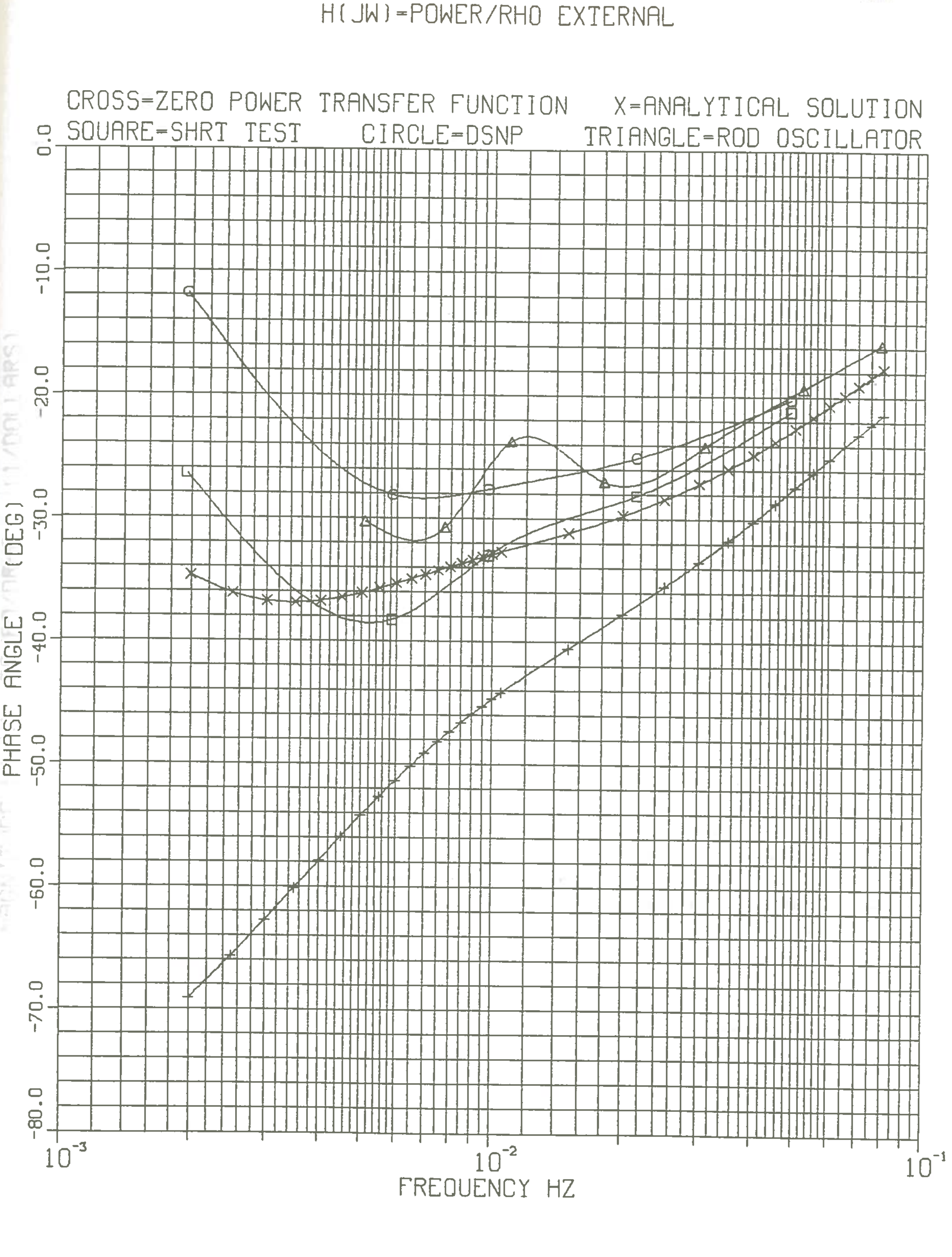

Fig. 20 System transfer function phase for low frequencies

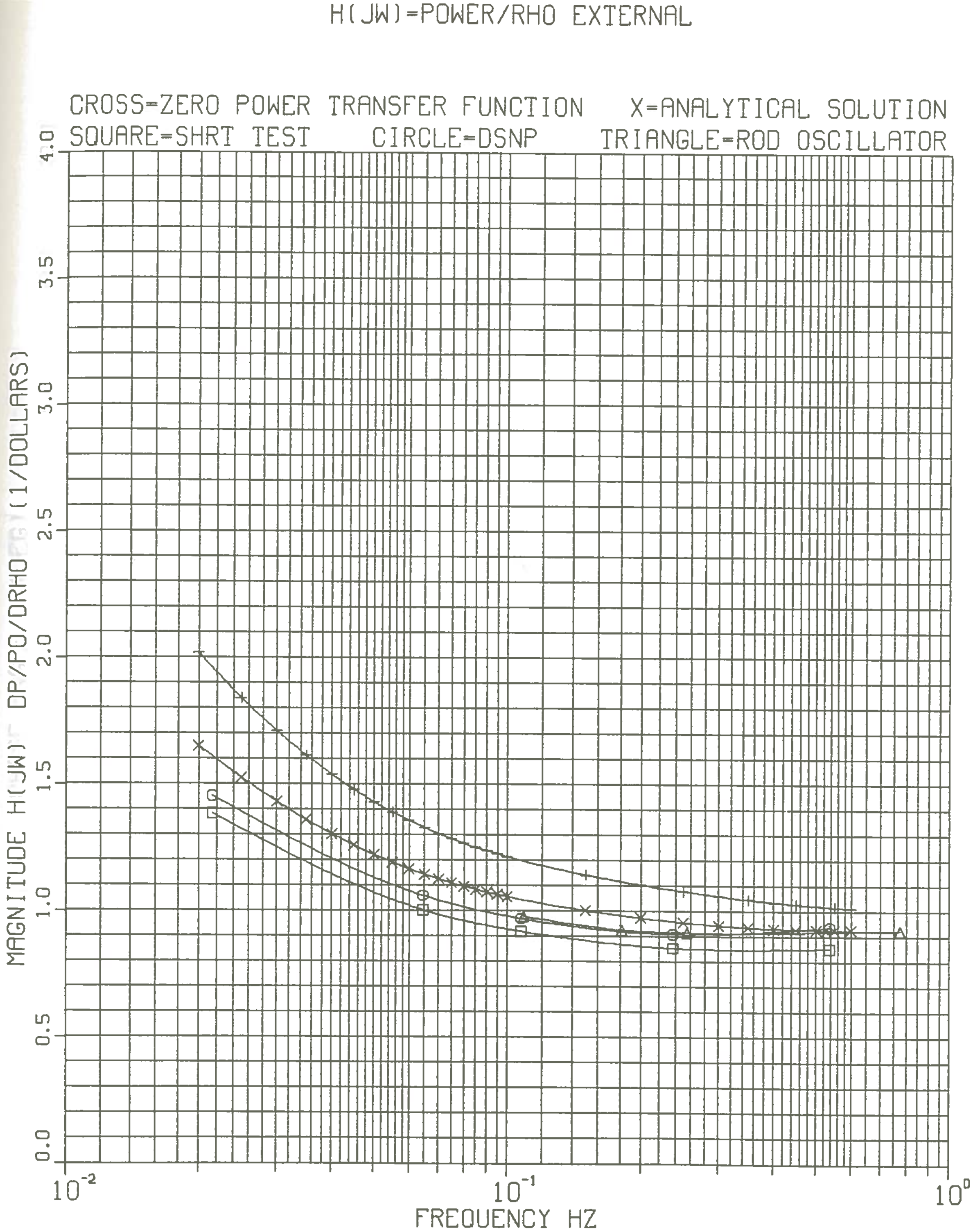

Fig. 21 System transfer function magnitude for high frequencies

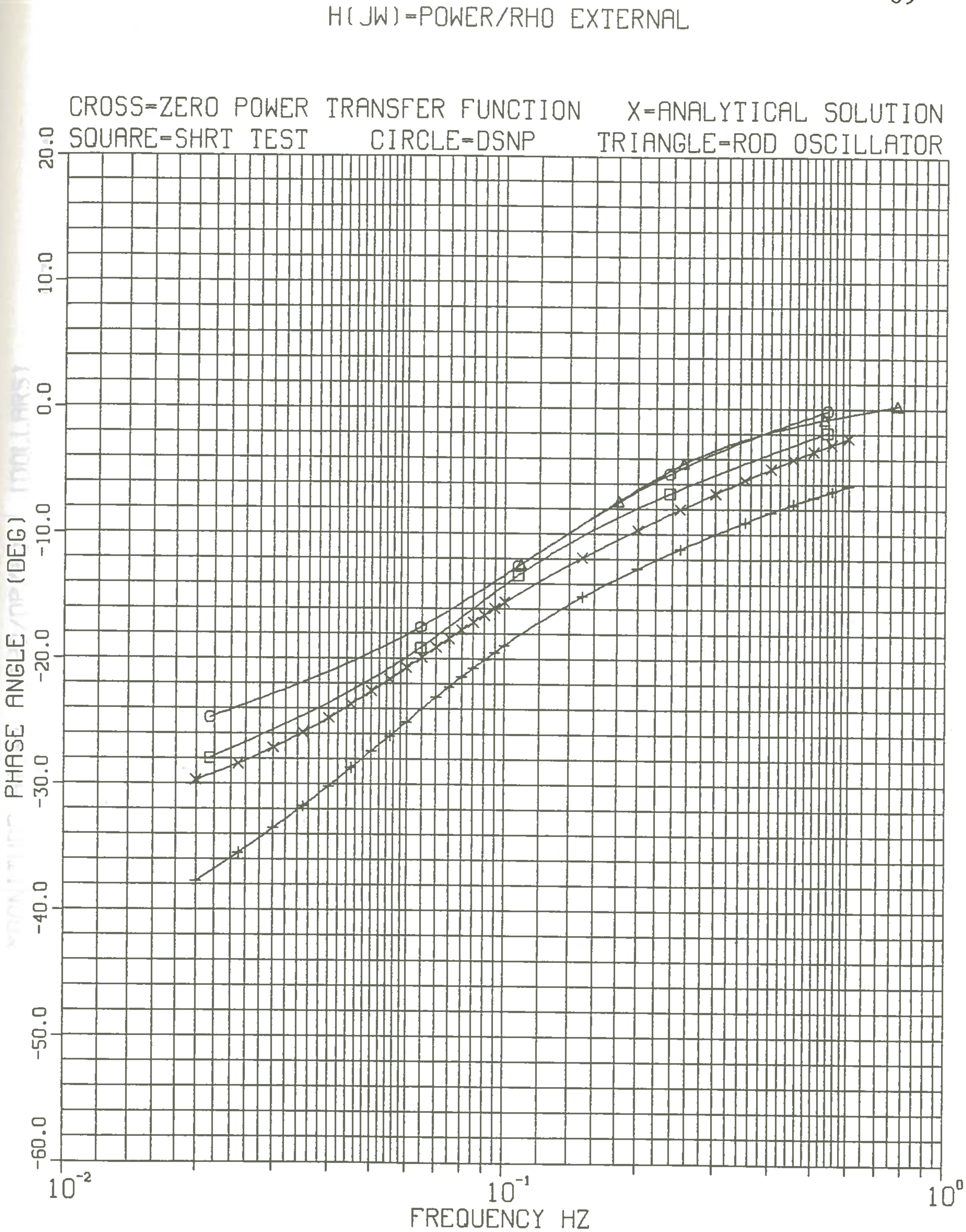

Fig. 22 System transfer function phase for high frequencies

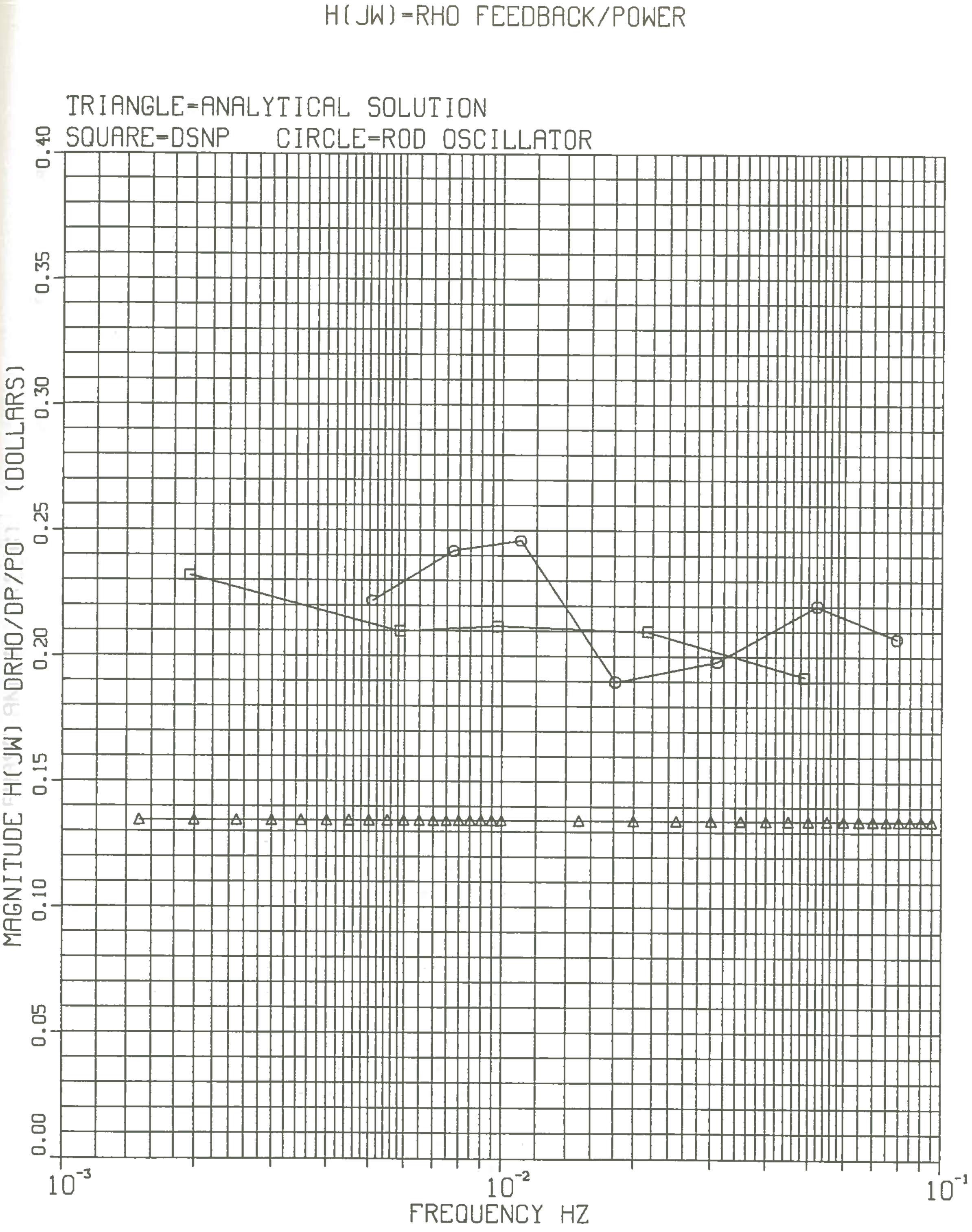

Fig. 23 Feedback transfer function magnitude for low frequencies

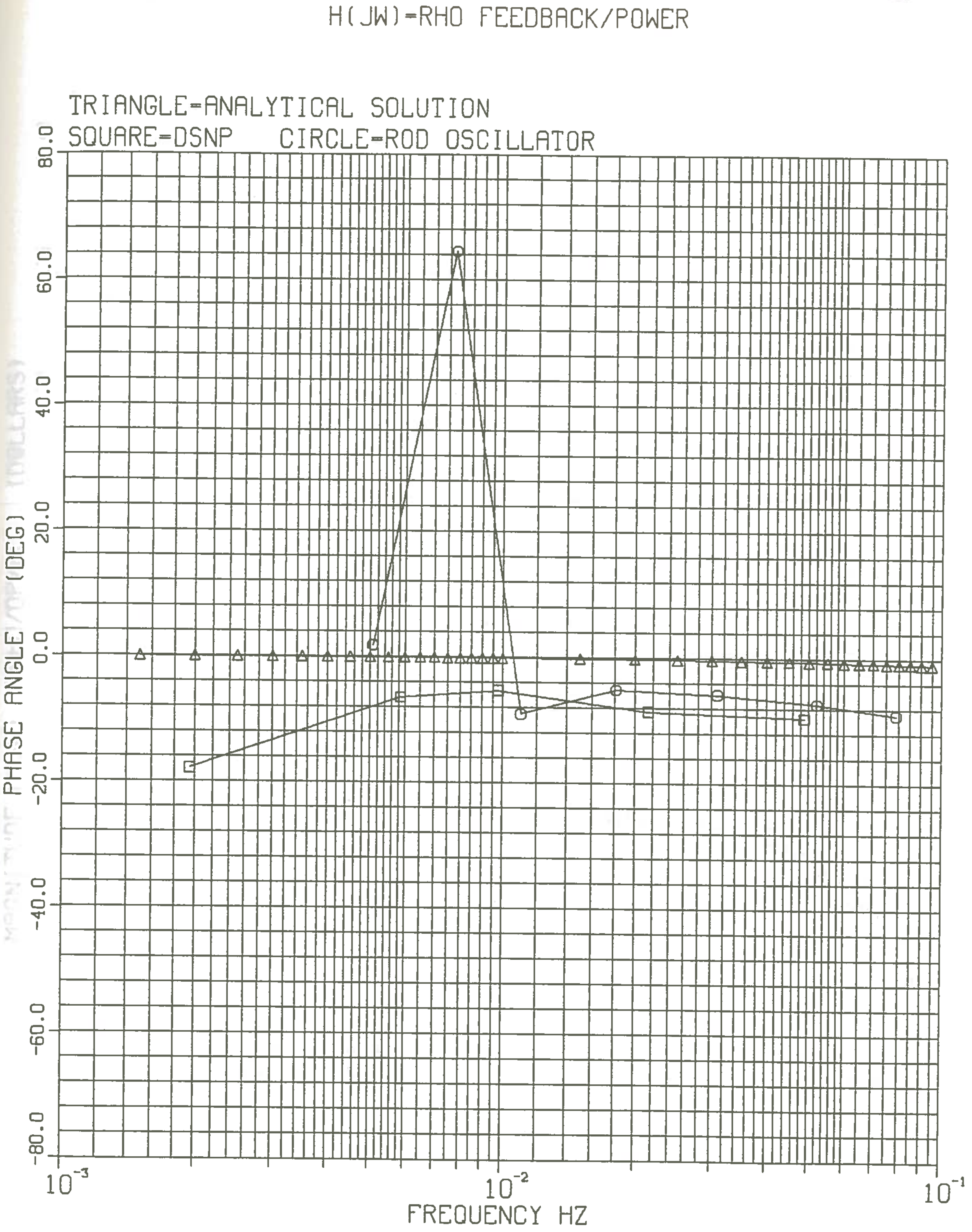

Fig. 24 Feedback transfer function phase for low frequencies

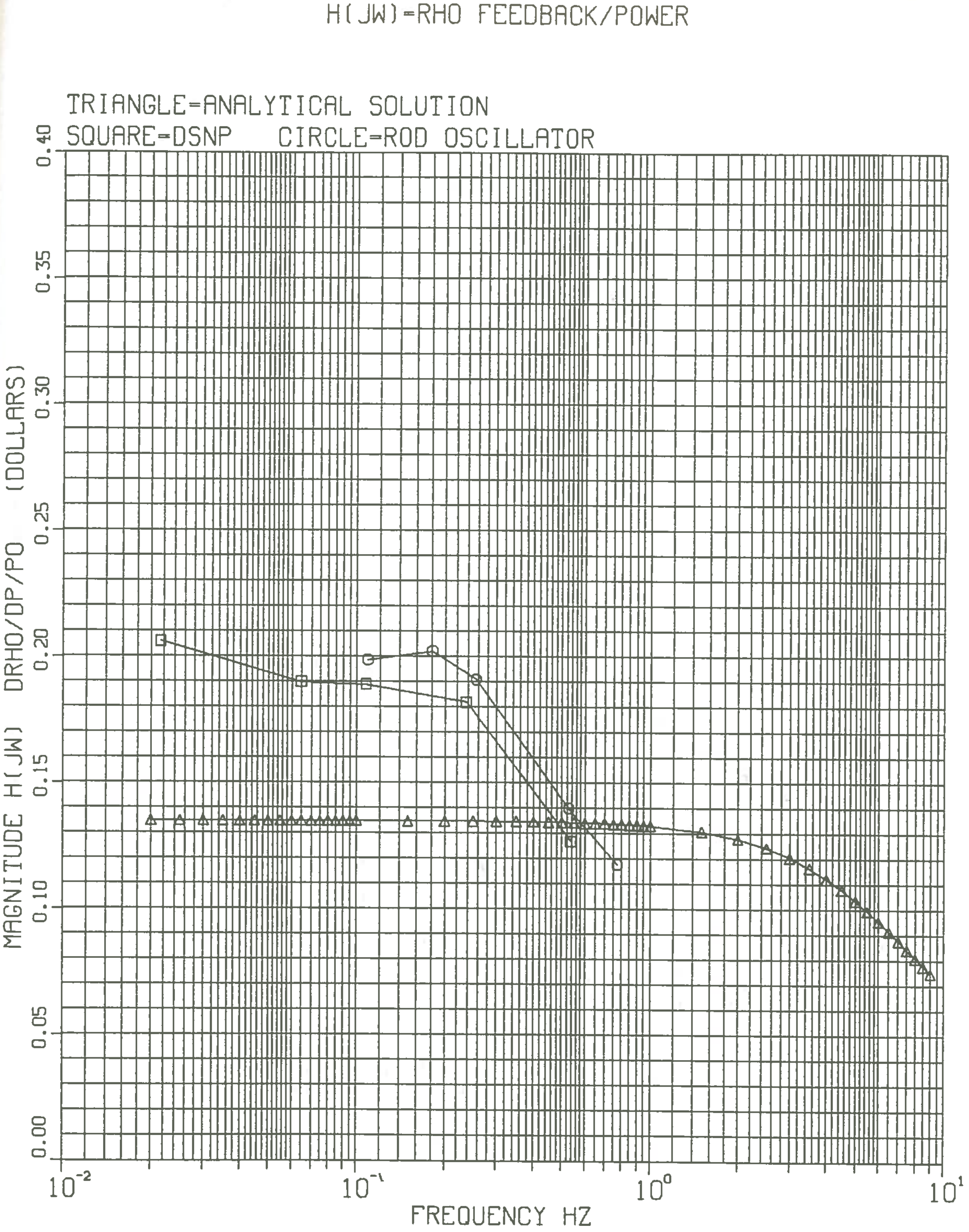

Fig. 25 Feedback transfer function magnitude for high frequencies

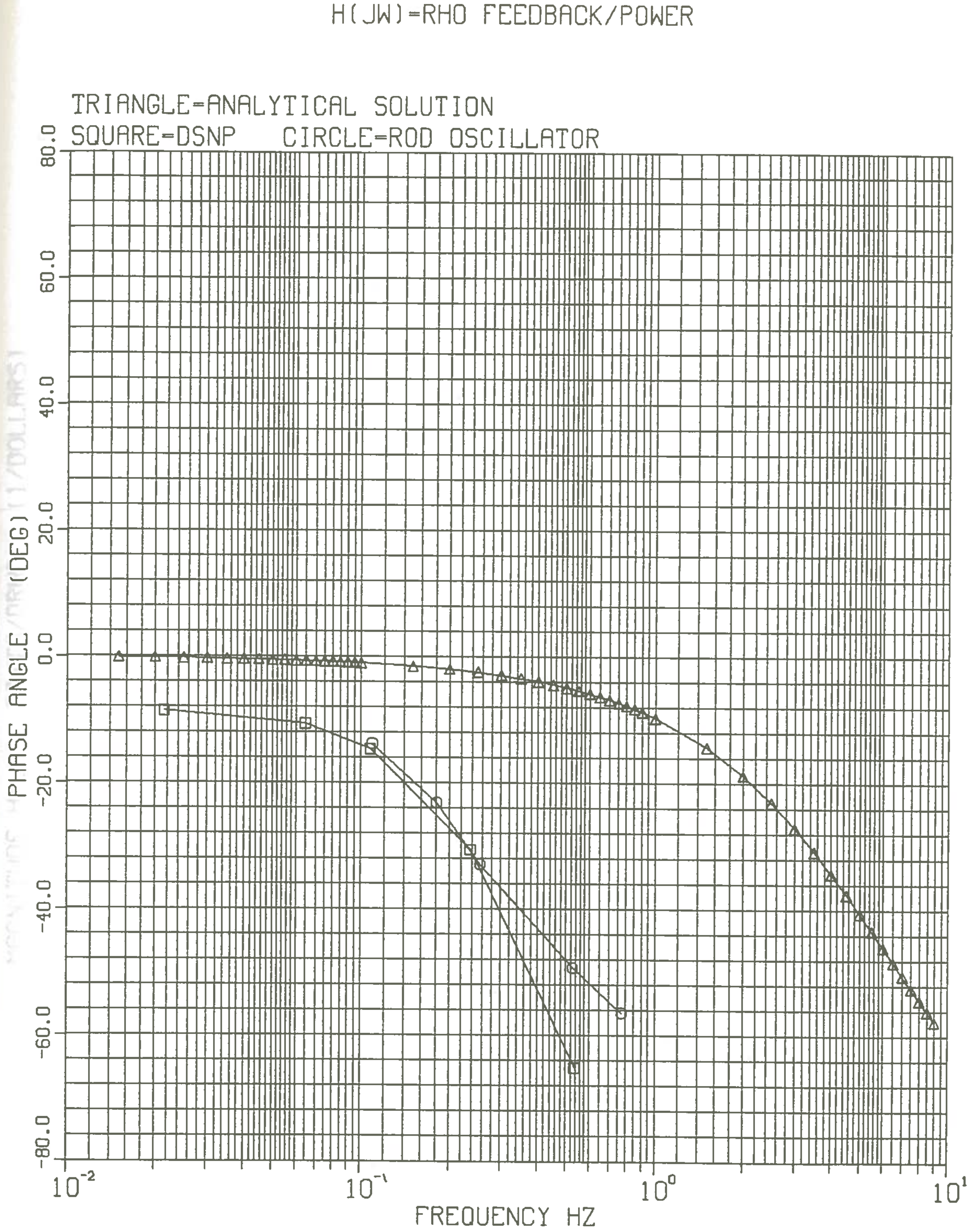

Fig. 26 Feedback transfer function phase for high frequencies

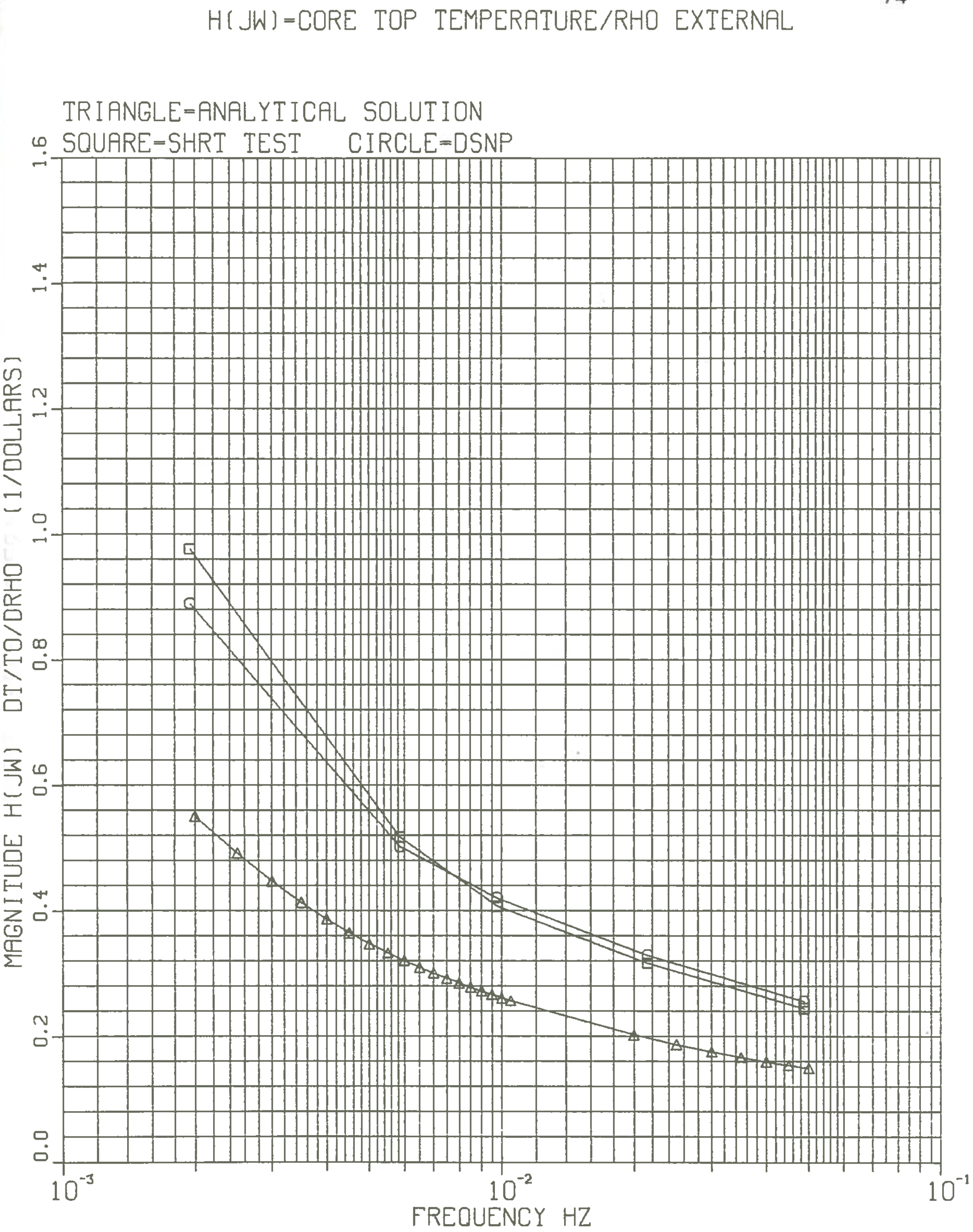

Fig. 27 Core top temperature to external reactivity transfer function magnitude for low frequencies

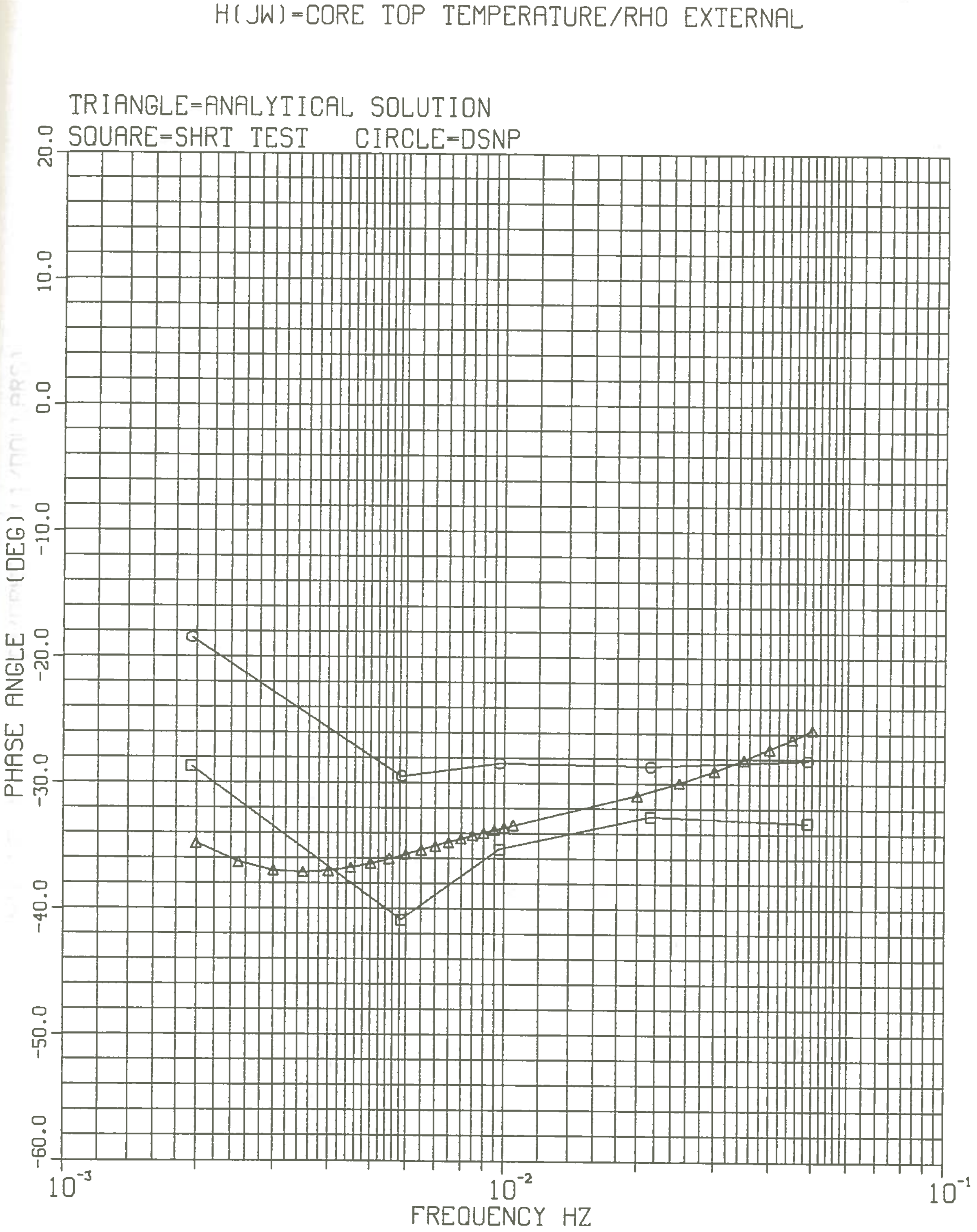

Fig. 28 Core top temperature to external reactivity transfer function phase for low frequencies

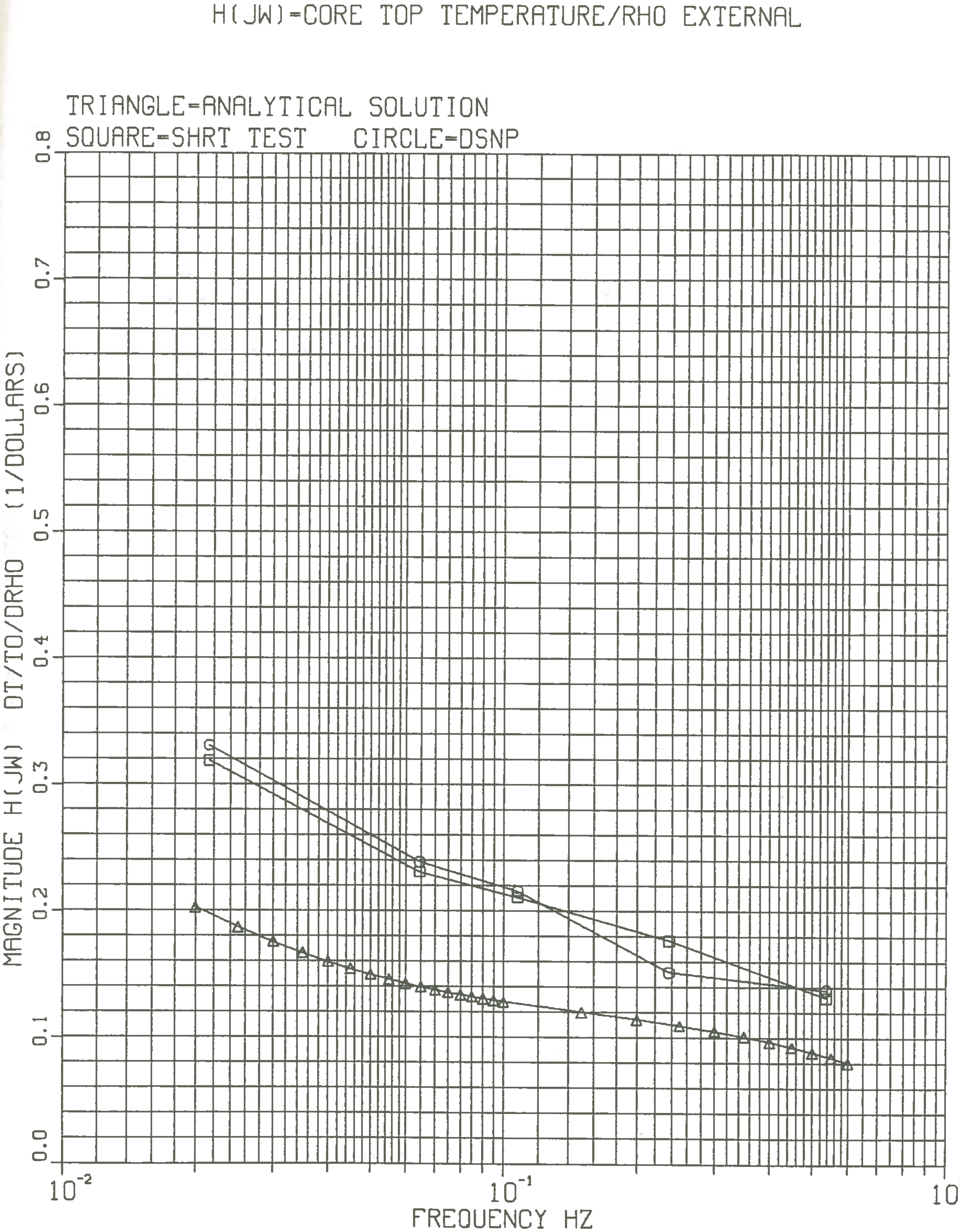

Fig. 29 Core top temperature to external reactivity transfer function magnitude for high frequencies

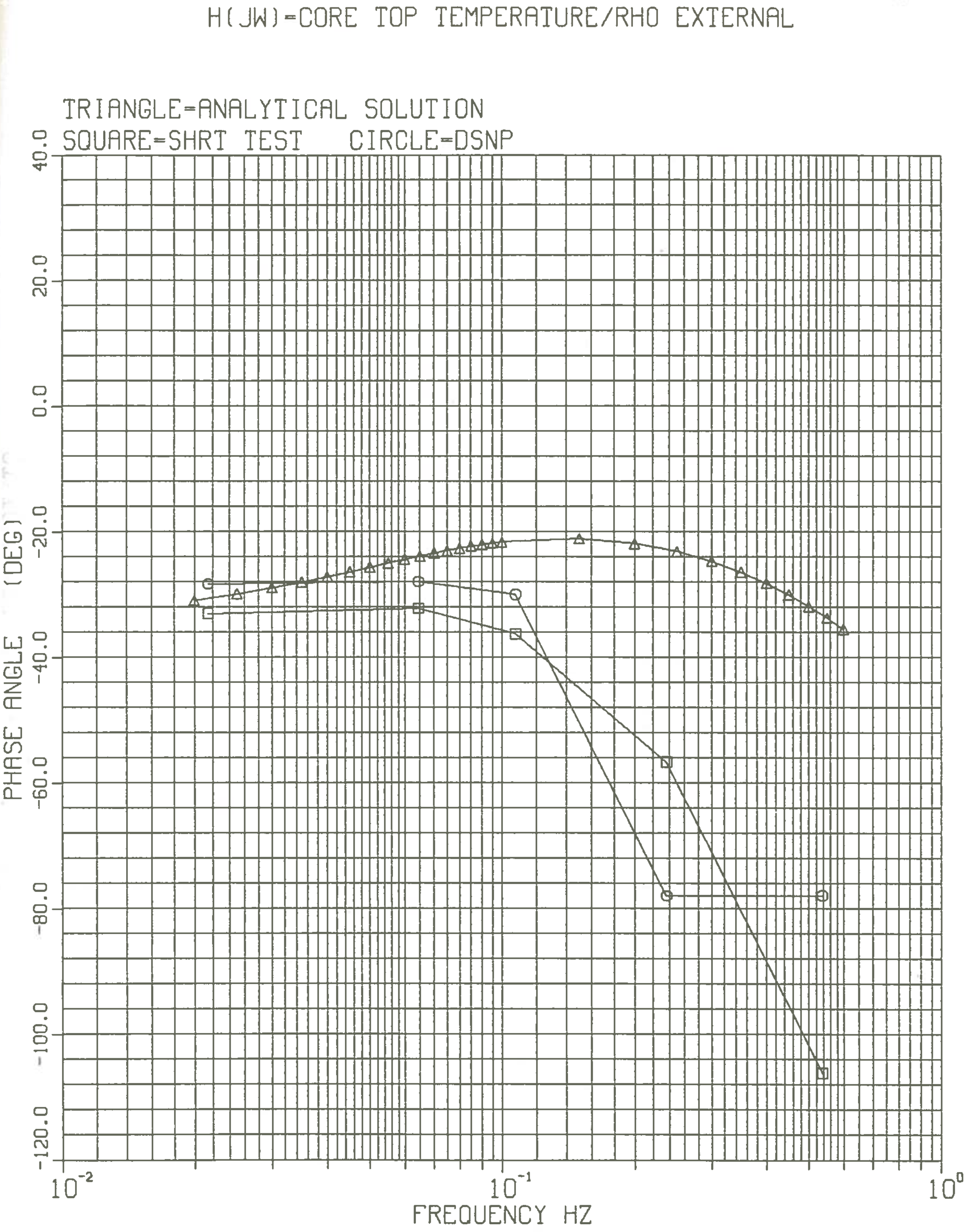

Fig. 30 Core top temperature to external reactivity transfer function phase for high frequencies

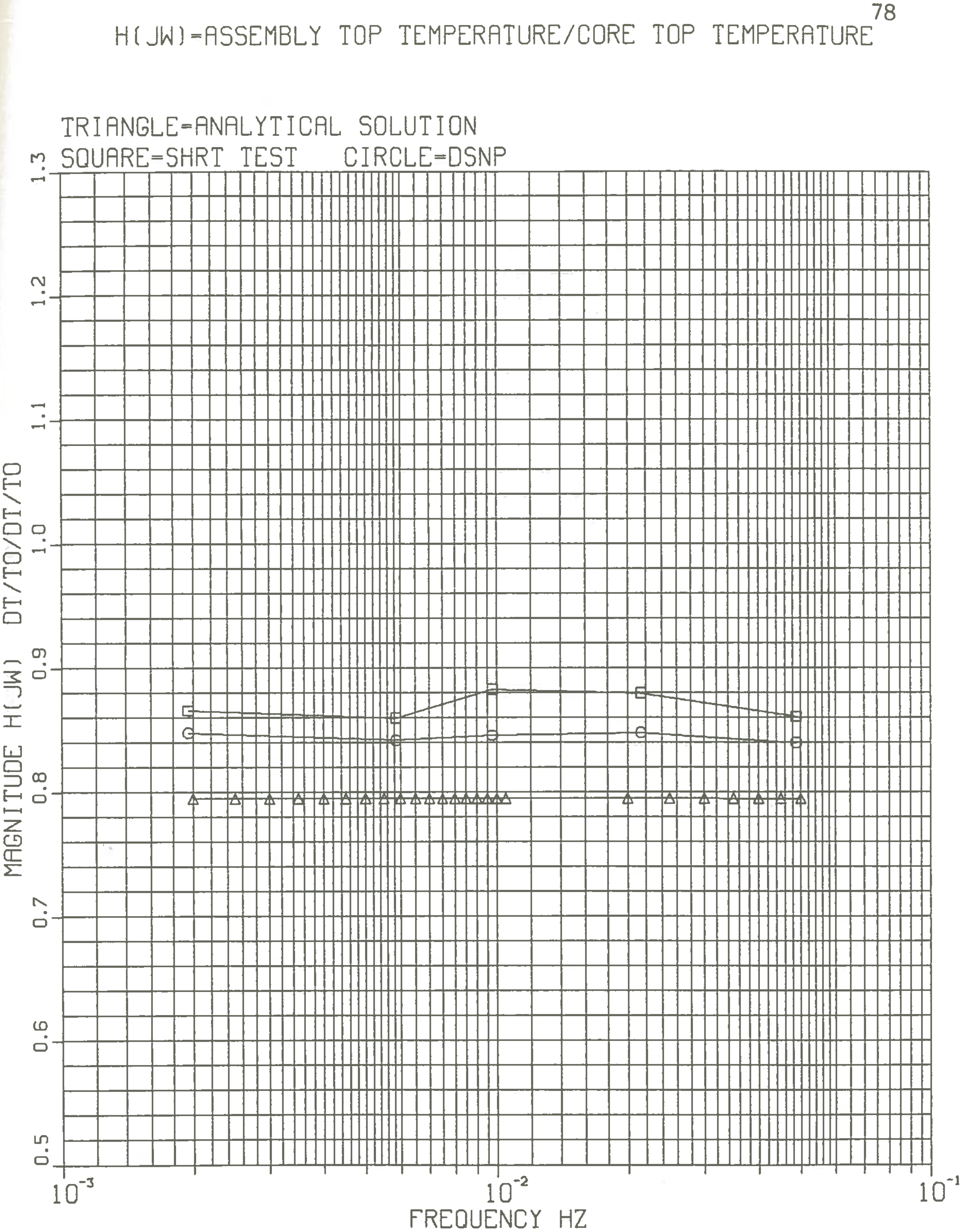

Fig. 31 Assembly top to core top temperature transfer function magnitude for low frequencies

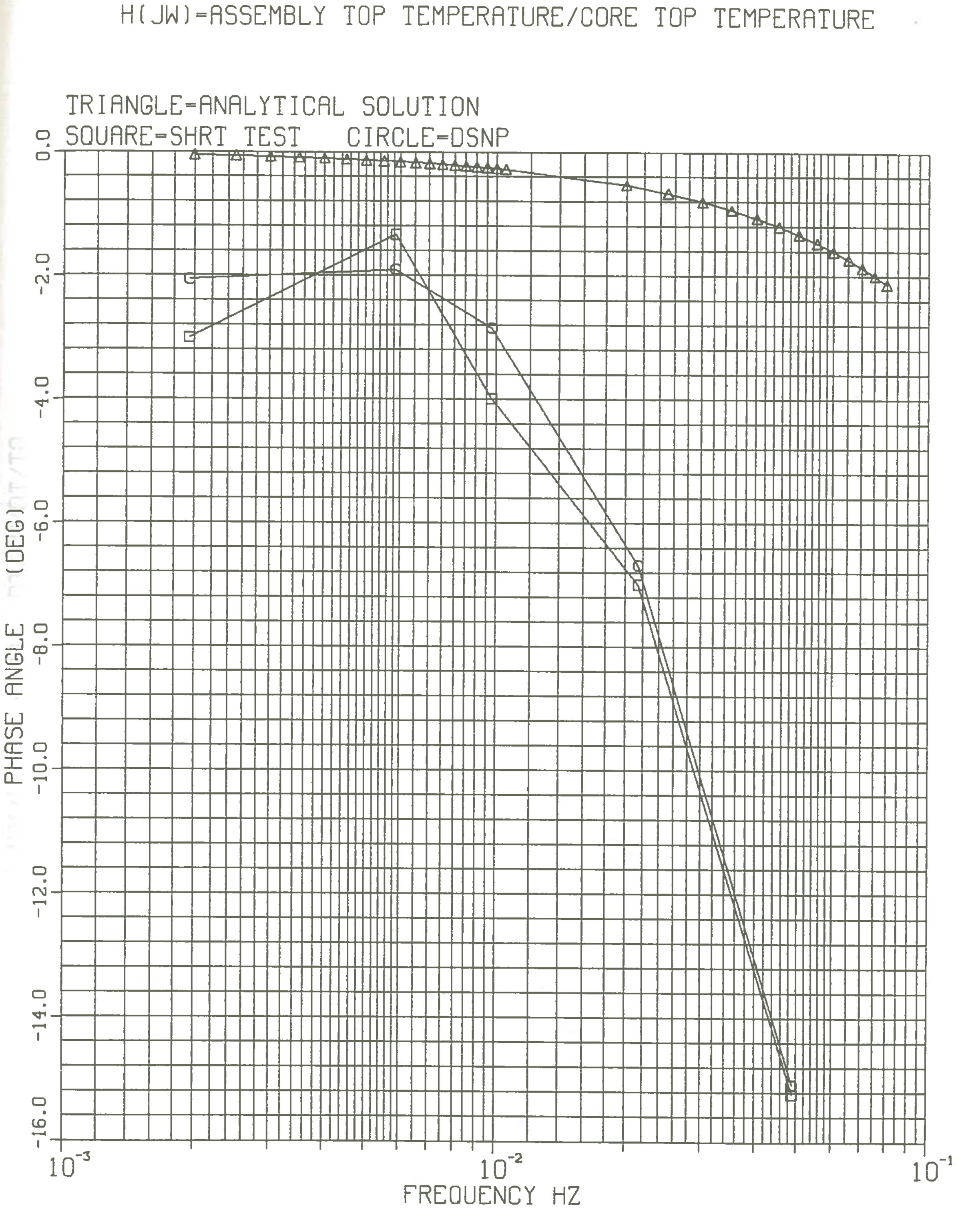

Fig. 32 Assembly top to core top temperature transfer function phase for low frequencies

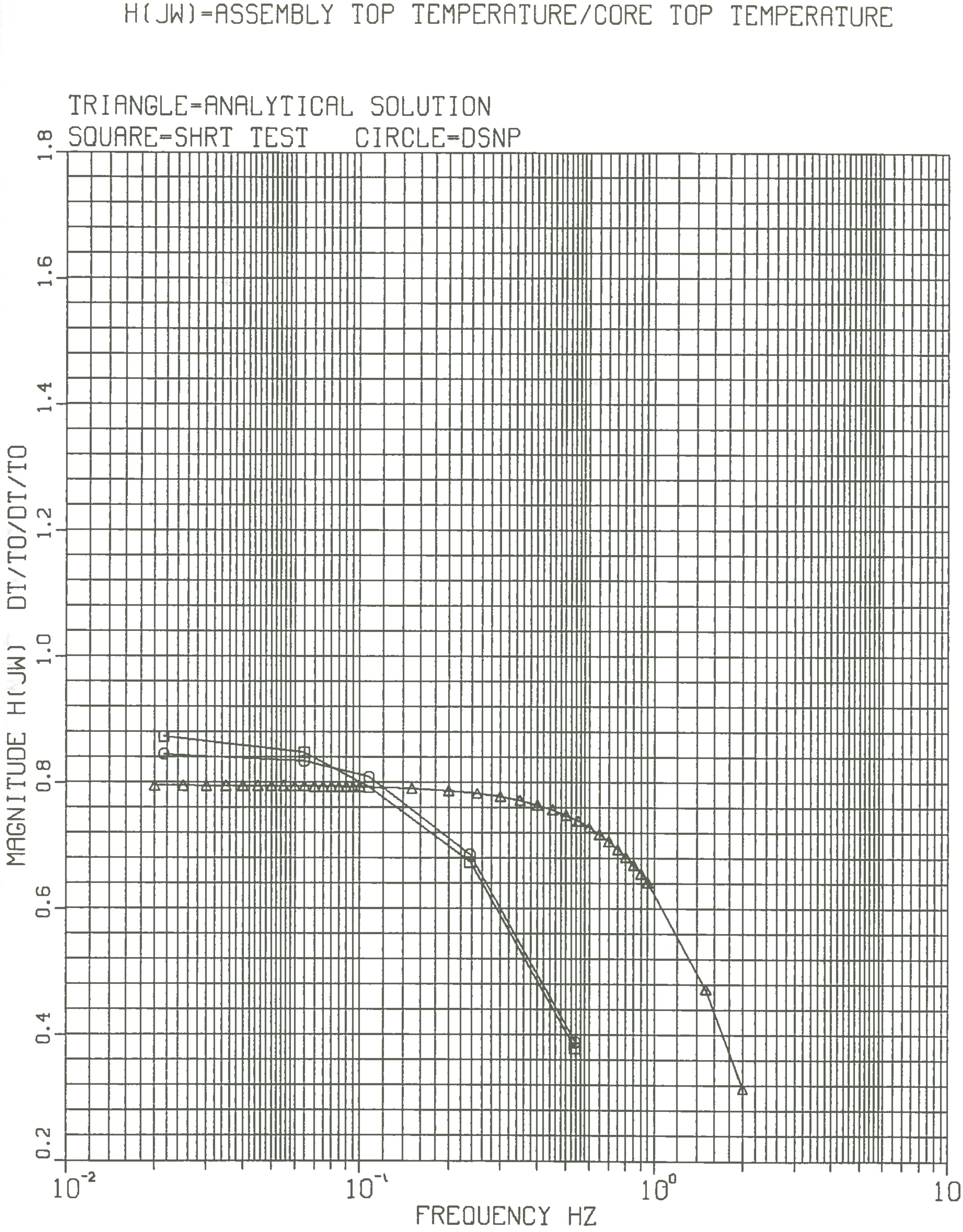

Fig. 33 Assembly top to core top temperature transfer function magnitude for high frequencies

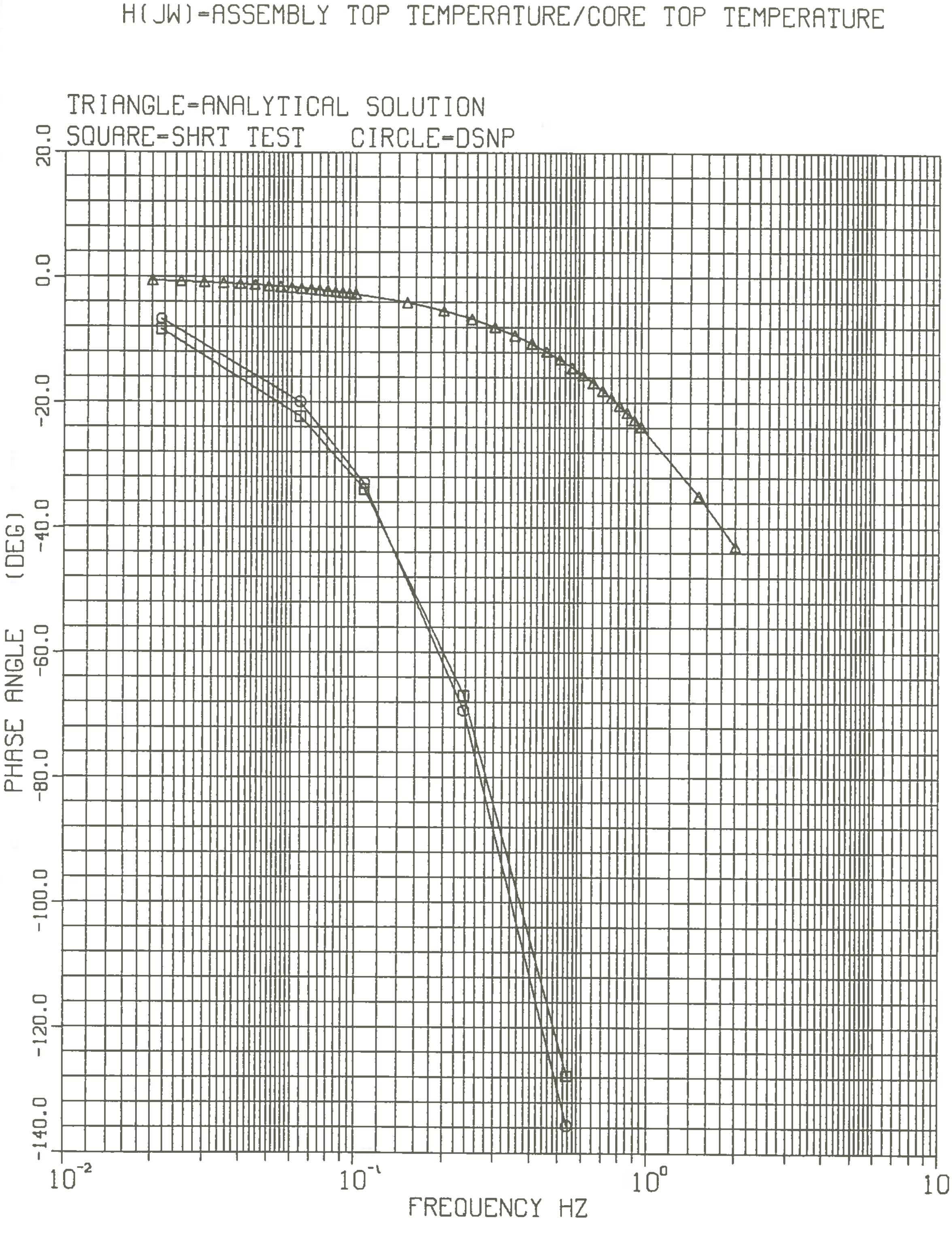

Fig. 34 Assembly top to core top temperature transfer function phase for high frequencies

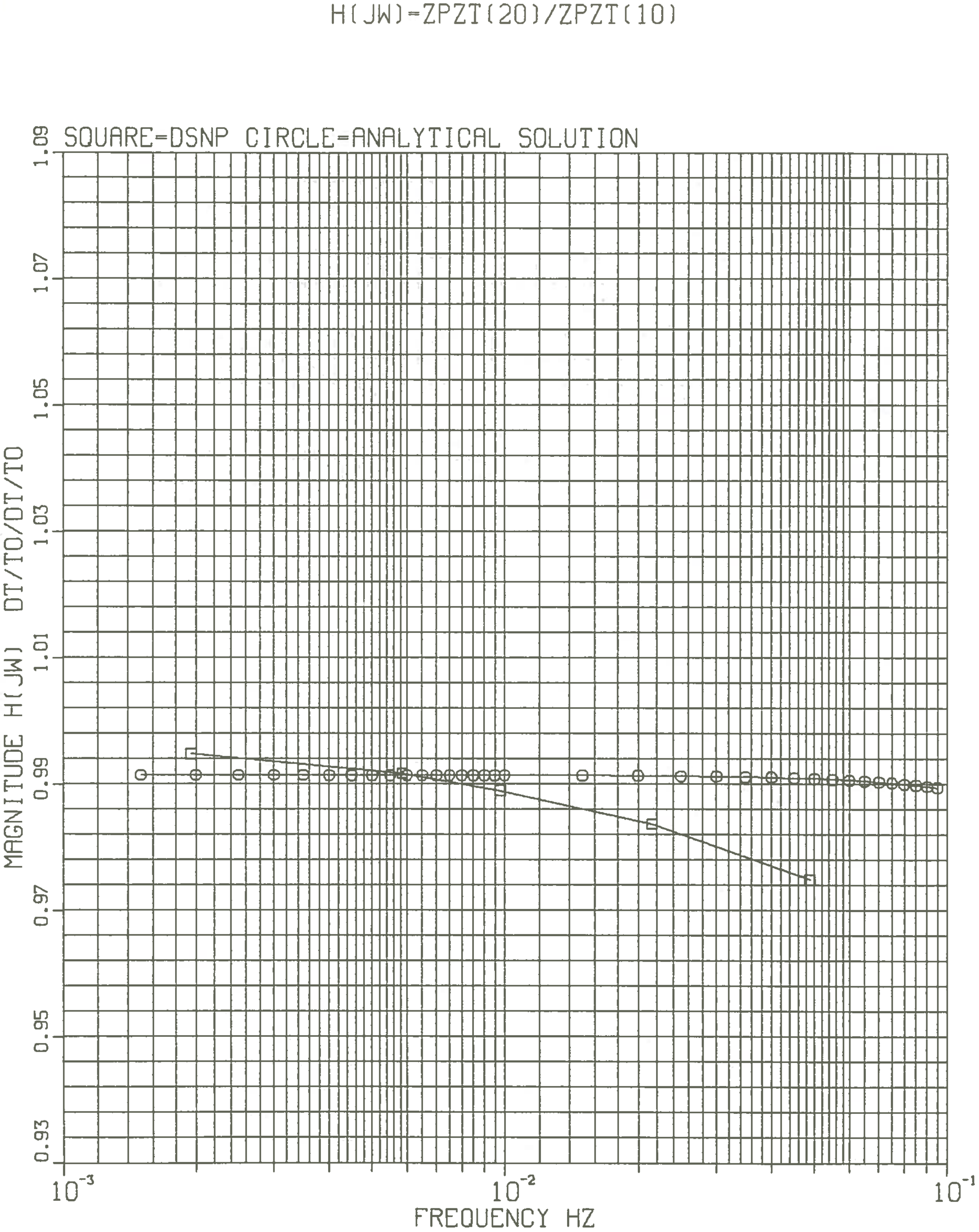

Fig. 35 Z pipe transfer function magnitude for low frequencies

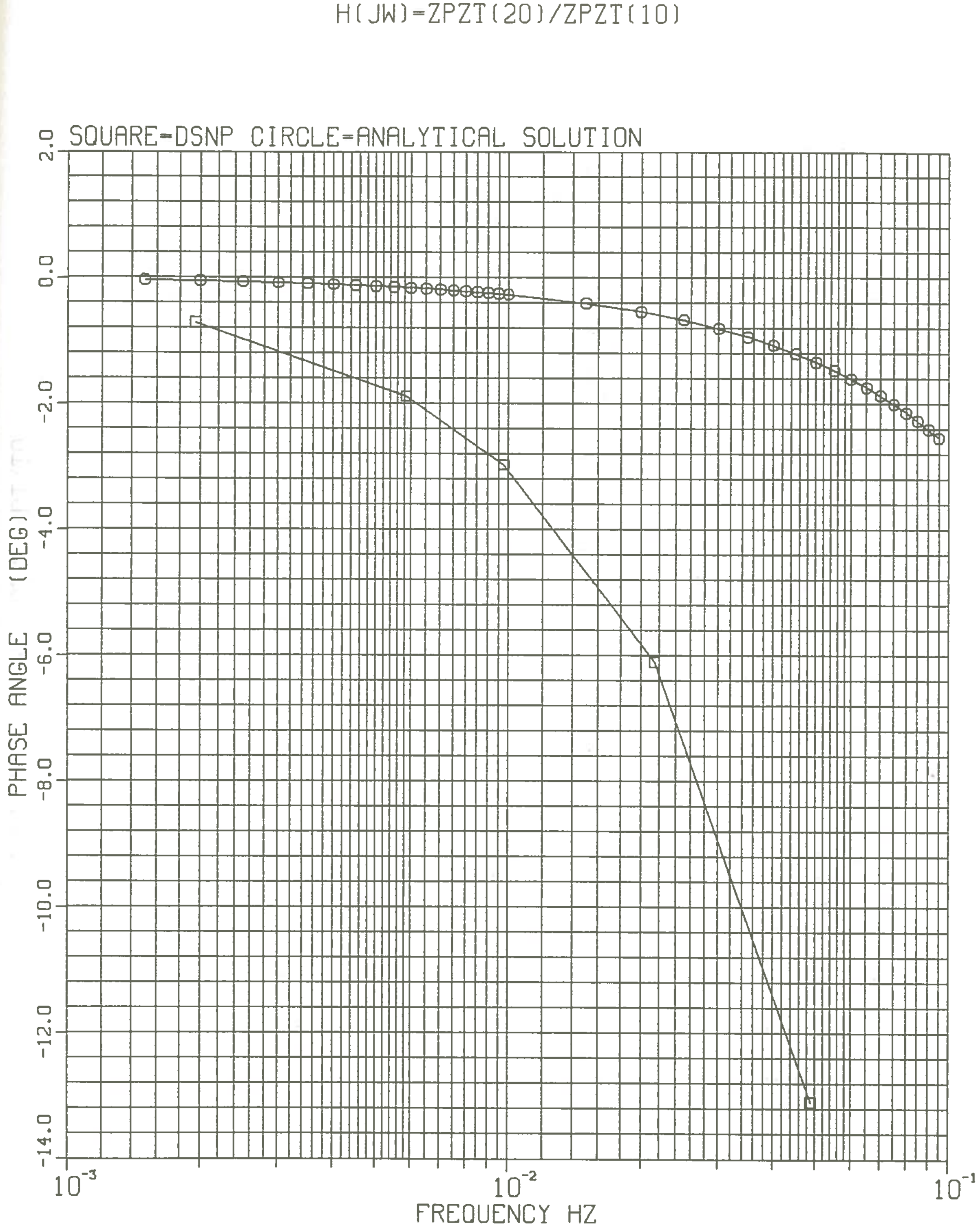

Fig. 36 Z pipe transfer function phase for low frequencies

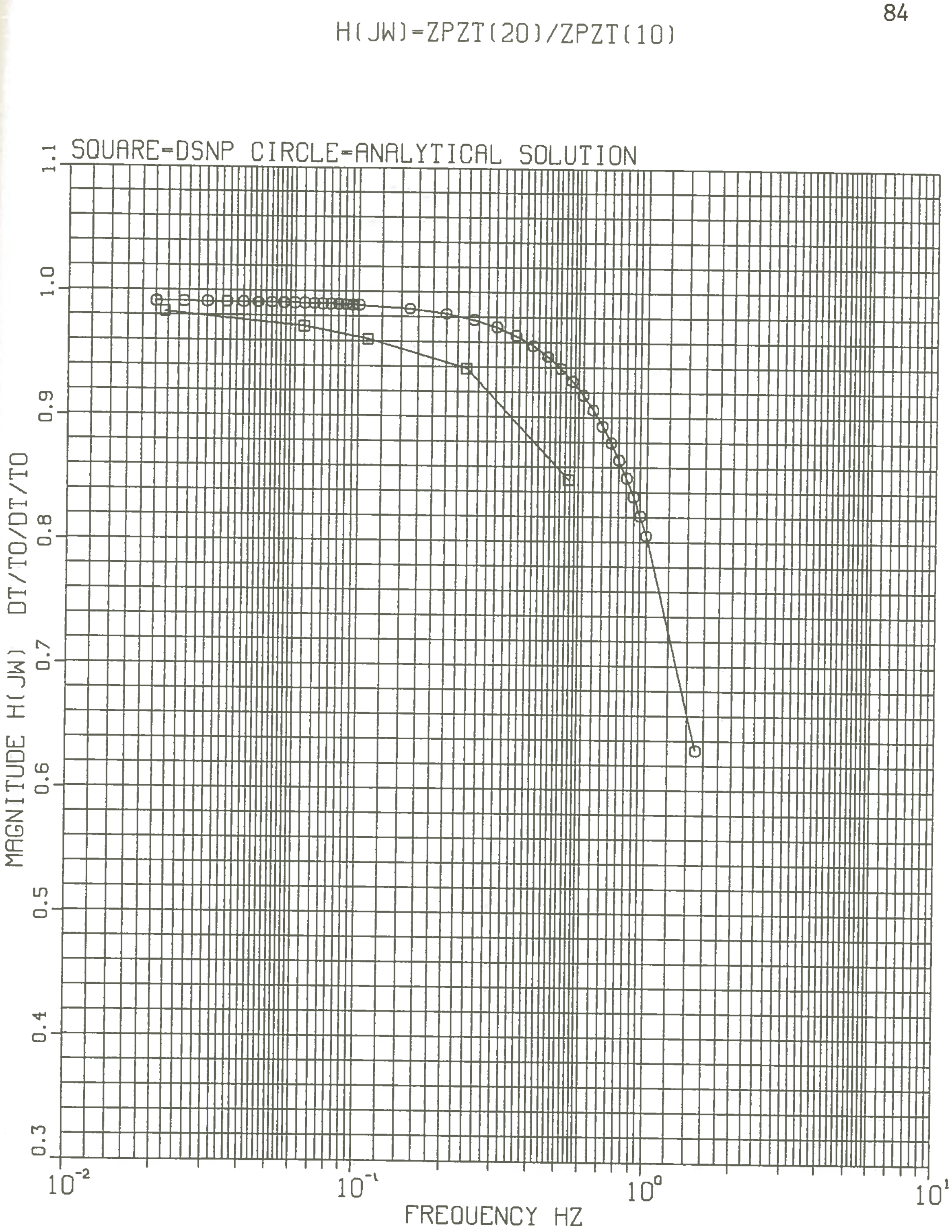

Fig. 37 Z pipe transfer function magnitude for high frequencies

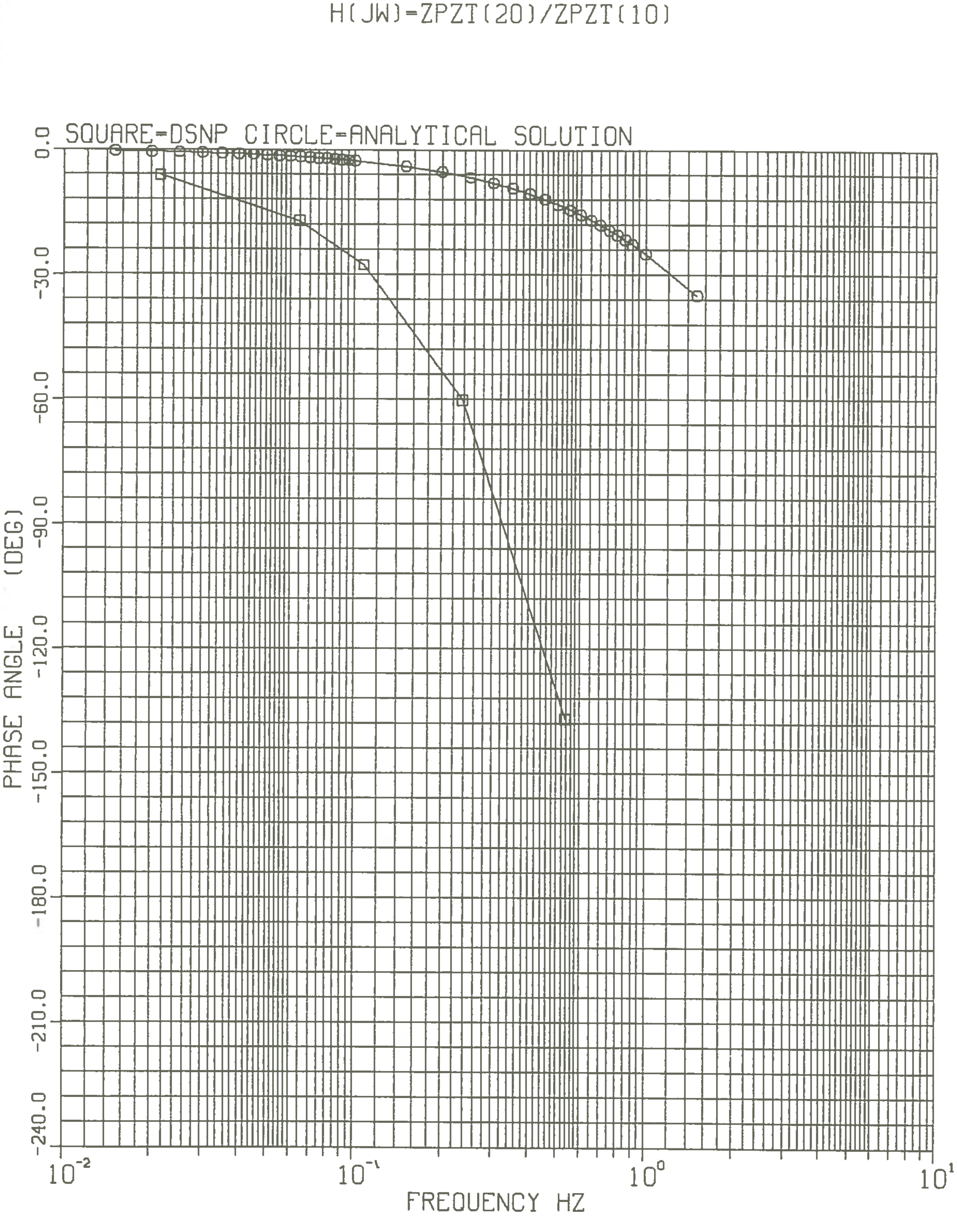

Fig. 38 Z pipe transfer function phase for high frequencies

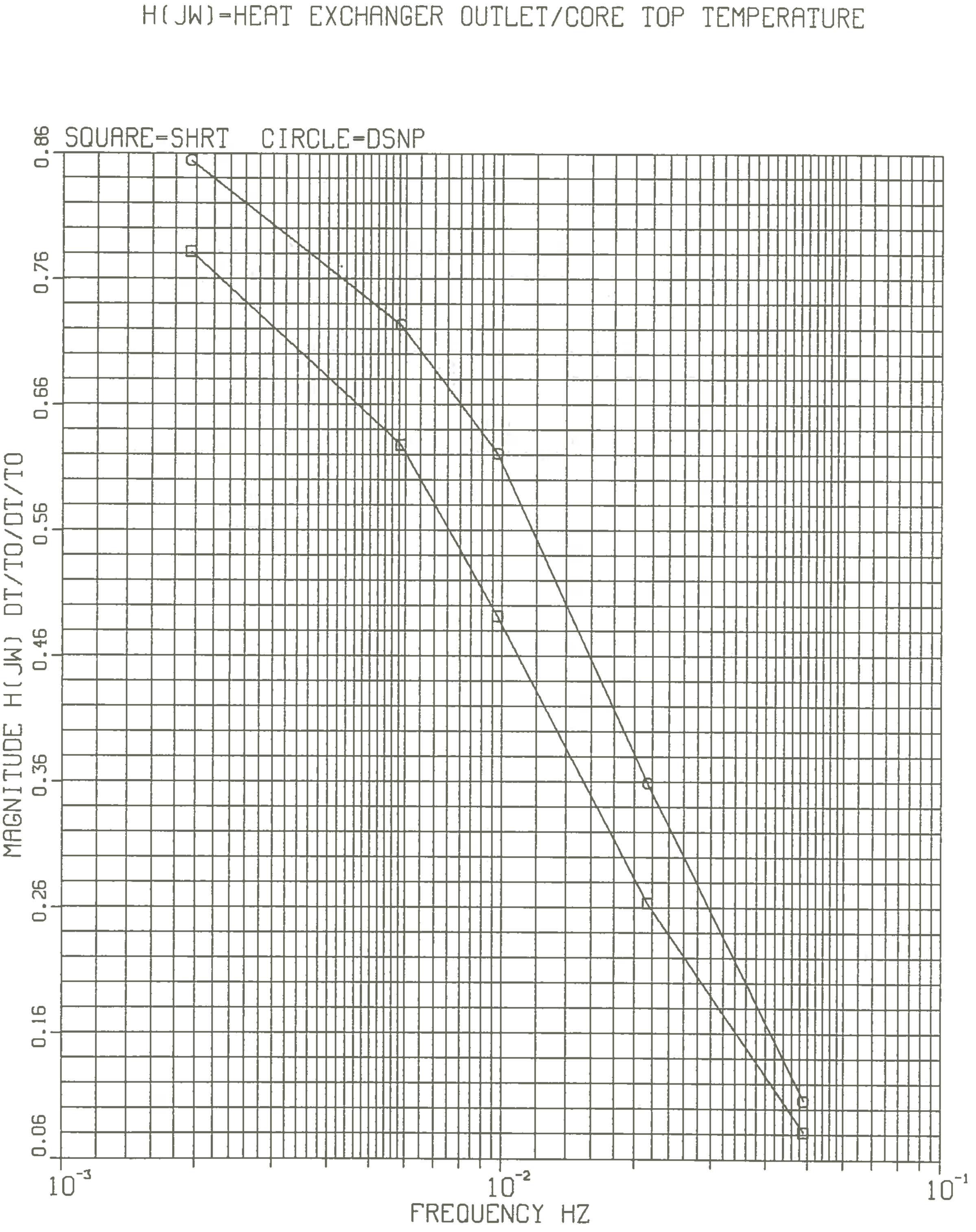

Fig. 39 Heat exchanger secondary outlet to core top temperature transfer function magnitude for low frequencies

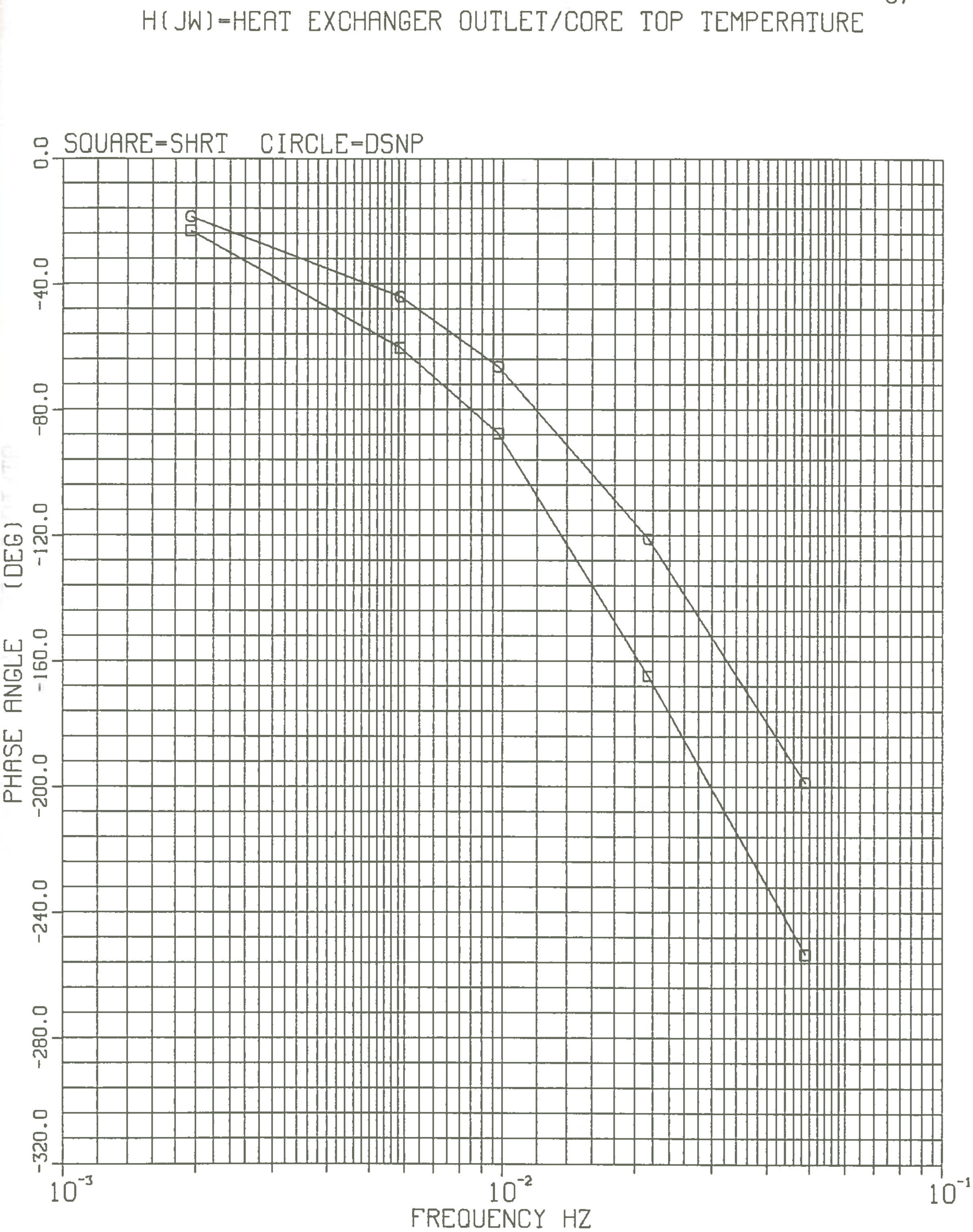

Fig. 40 Heat exchanger secondary outlet to core top temperature transfer function phase for low frequencies

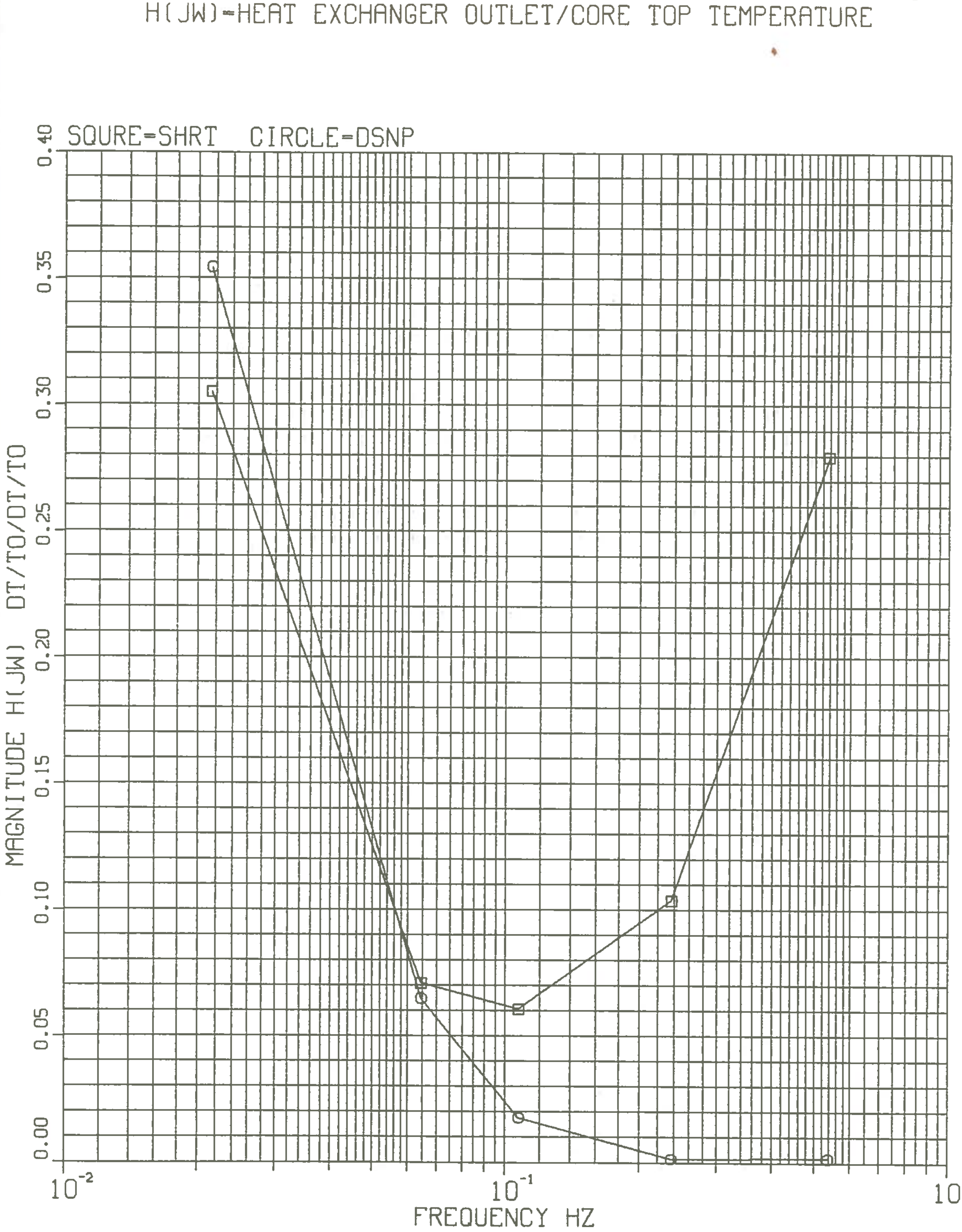

Fig. 41 Heat exchanger secondary outlet to core top temperature transfer function magnitude for high frequencies

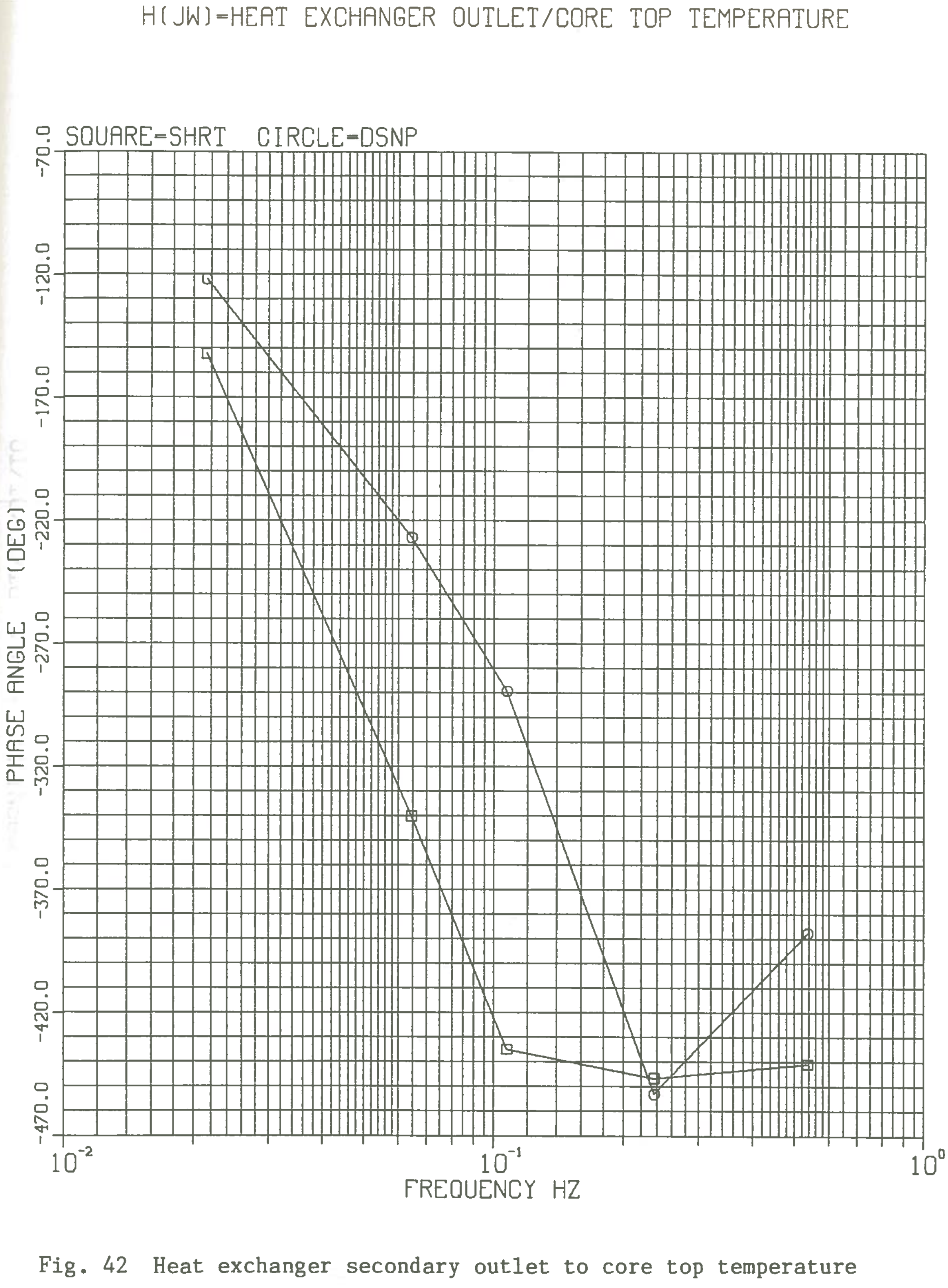

Fig. 42 Heat exchanger secondary outlet to core top temperature transfer function phase for high frequencies

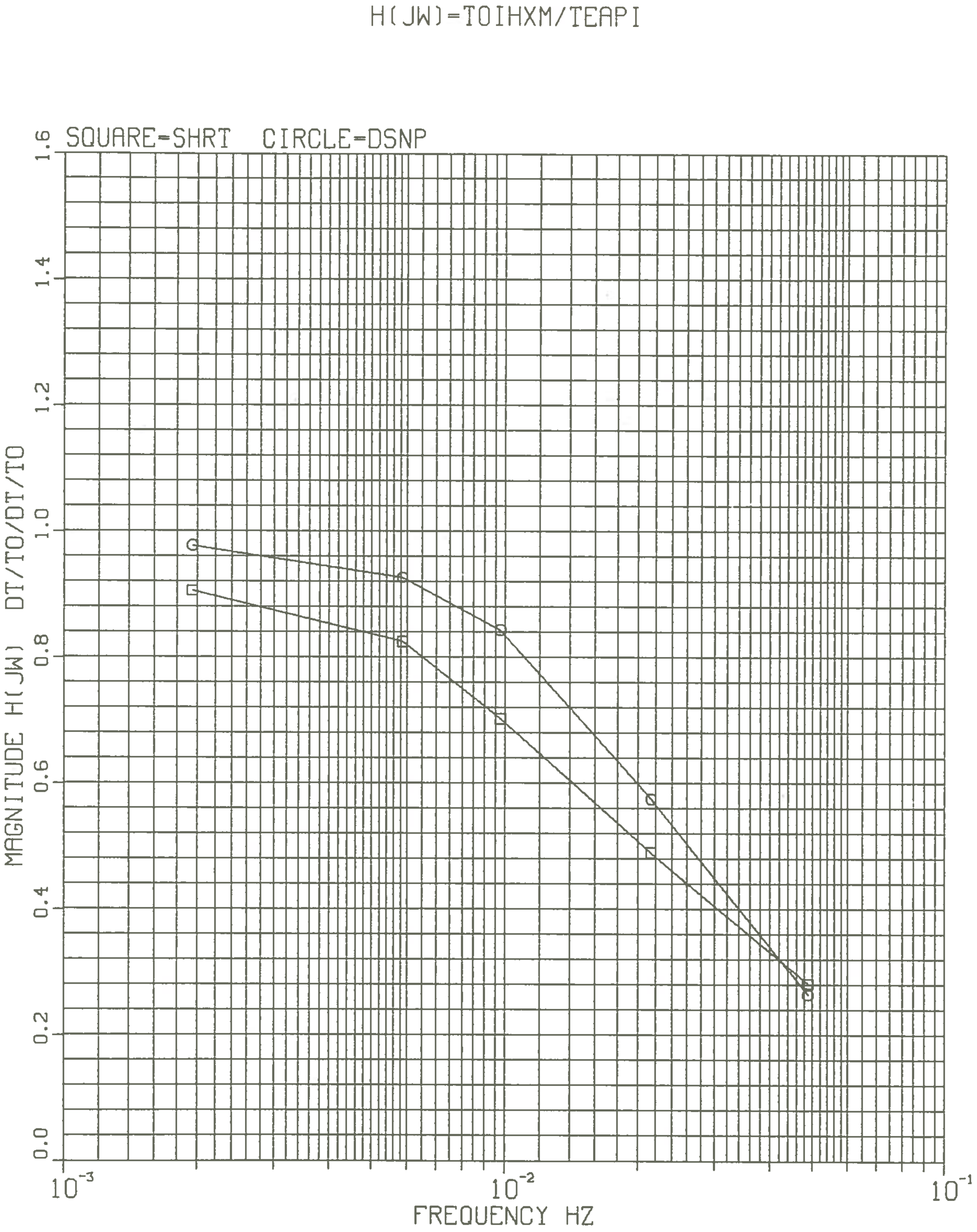

Fig. 43 Heat exchanger secondary outlet to primary inlet temperature transfer function magnitude for low frequencies

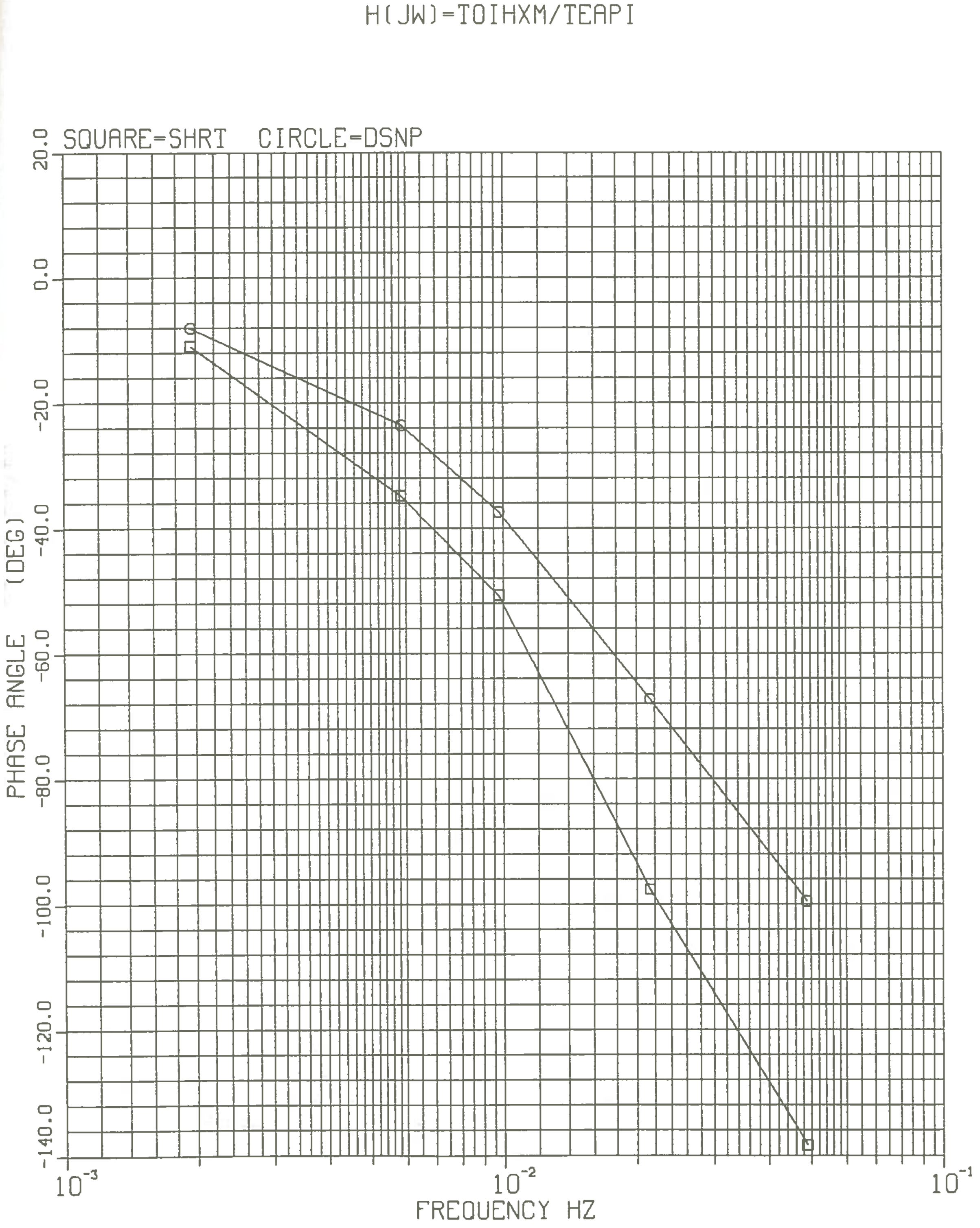

Fig. 44 Heat exchanger secondary outlet to primary inlet temperature transfer function phase for low frequencies

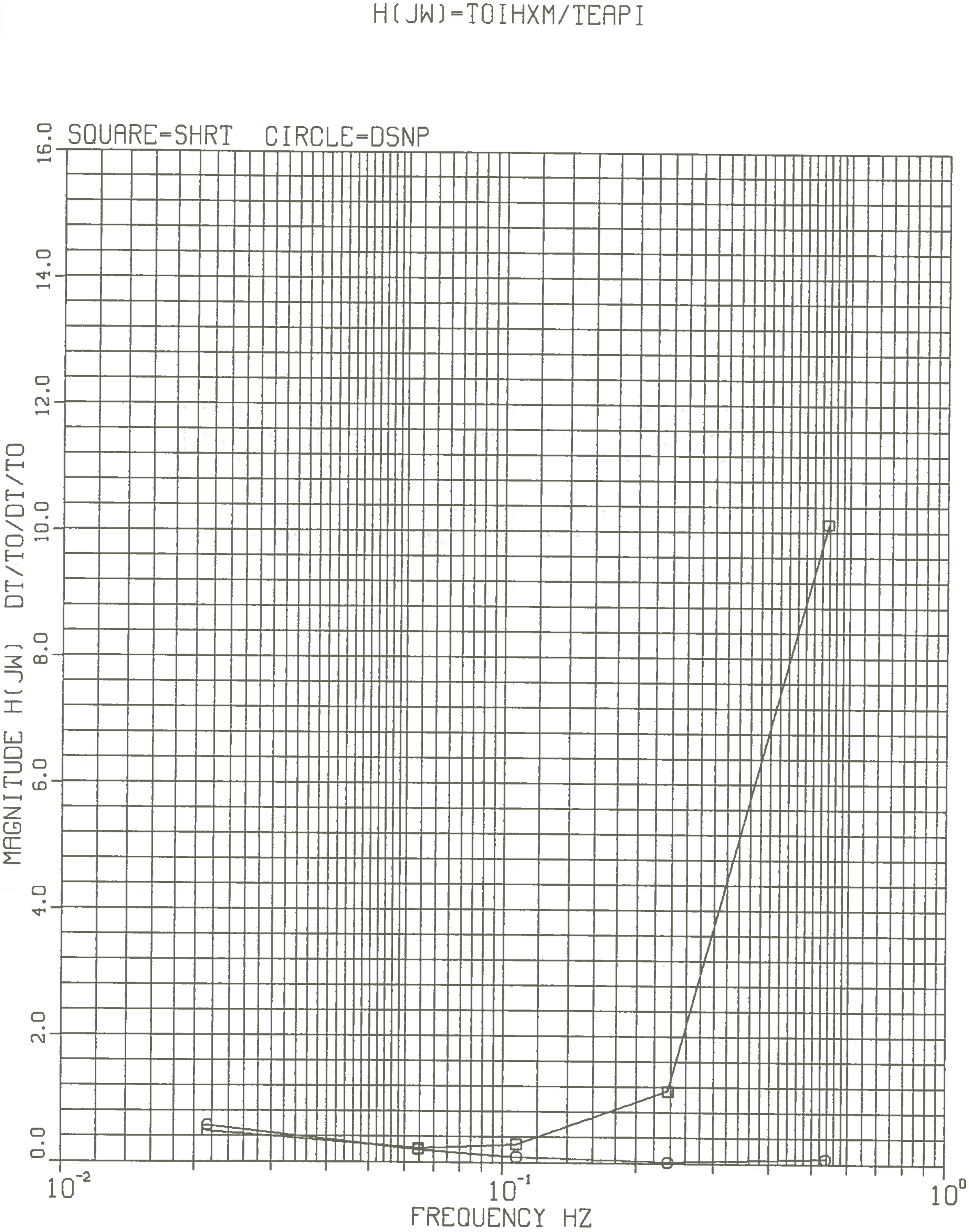

Fig. 45 Heat exchanger secondary outlet to primary inlet temperature transfer function magnitude for high frequencies

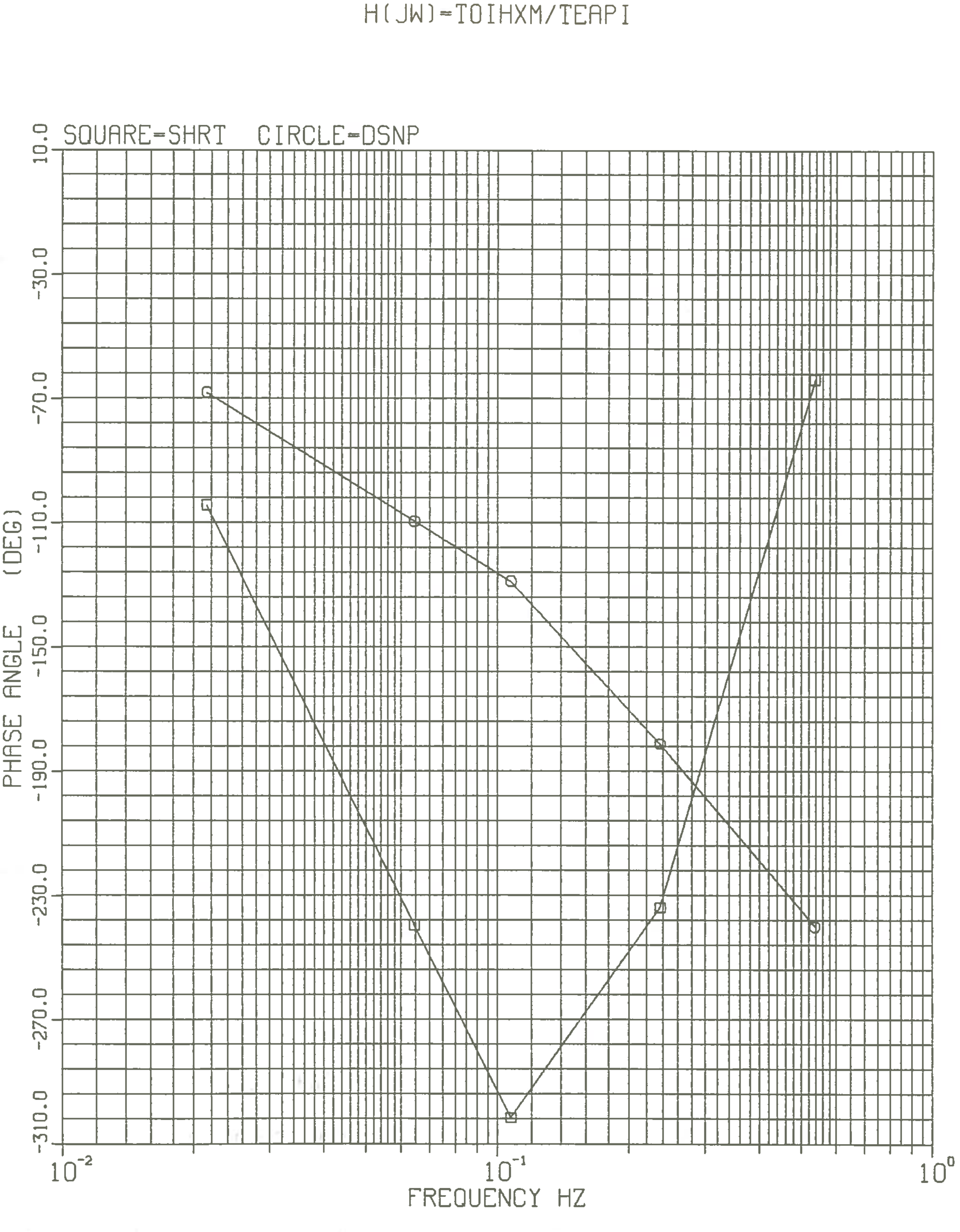

Fig. 46 Heat exchanger secondary outlet to primary inlet temperature transfer function phase for high frequencies

Chapter Eight

Summary and Conclusions

Comparison of the SHRT experiment analysis and DSNP simulation results show that DSNP is very accurate for calculating the magnitude of transfer functions for output responses located reasonably close to the driving function. This can be seen in the results of the core transfer function analysis. Phase angles of these same transfer functions are not as accurately calculated. There may be as much as 30% error in the calculated core transfer functions. However, it should be kept in mind that the experimental measurements themselves vary by as much as 40% from one run to another run for the external reactivity to output power transfer function. So that the inaccuracy of the DSNP runs is difficult to establish. Therefore, even though the phase of a signal may be mispredicted, this discrepancy may represent a small error in predicting the time lag of the signal at a certain frequency of the transfer function. For example, there is a difference of 10 degrees between the phase of the experimental transfer function and that of DSNP at the fundamental frequency. This represents only a 14 second error in predicting the propagation of a 512 second signal. Thus relatively large errors in phase do not neccessarily cause great errors in magnitude. The heat exchanger transfer function is not as accurately calculated. This is probably the result of having to use the DSNP calculated Z pipe transfer functions to calculate it and due to the fact that the heat exchanger is far away from the driving function. The instrumentation at EBR-II is not designed for frequency response testing. Difficulties with it include time constants that are too large and inadequate instrumentation for the measurement of temperatures between the assembly top and the outlet of the secondary of the intermediate heat exchanger. If EBR-II is removed from operation and the main cover removed, additional instrumentation should be added to alleviate this shortcoming.

The analytical transfer functions that have been derived are accurate in the case of the system transfer function and less accurate for other transfer functions that have been considered. They do yield the correct behavior and if one wanted to alter the DSNP source in an attempt to improve results, these transfer functions would indicate where to look and what to work on.

In conclusion, DSNP can be used to predict dynamic analysis of EBR-II with an expected error comparable to the experimental variation for different runs. If one wishes to obtain more accuracy with DSNP, the parameters entered into the present model should be closely examined and reevaluated with particular attention being paid to those parameters that produce the inaccuracies described in this thesis. This study shows that DSNP can be used to construct transfer functions for most components in the EBR-II primary system.

REFERENCES

1. D.F. Elliot and K.R. Rao, Fast Transforms Algorithms, Analyses, Applications, Academic Press, Inc., New York, 1982.

2. E.M. Dean, Description of DSNP Model for the EBR-II Primary System, Operation Analysis Section, EBR-II Division, August 1986.

3. H.A. Larson, LMFBR Characterization Primer, Argonne National Laboratory, April 1975.

4. M.L. Carboneau, et al., A Study of the Temperature Coefficients of Reactivity for EBR-II Run 93, ANL-80-85, Argonne National Laboratory, November 1980.

5. H.A. Larson, Kinetics Research and Testing at EBR-II, ANL-8086, Argonne National Laboratory, April 1974.

6. E.E. Feldman and D. Mohr, ANL memo to W.N. Beck et al., Subject: "SHRT Group VII (Balance of Plant) Testing Program Development", October 11, 1984.

7. M.A. Schultz, Control of Nuclear Reactors and Power Plants, McGraw-Hill, New York, 1961.

8. R.V. Furstenau, The Simulation of EBR-II Frequency Response with Discrete Level Binary Signals, September, 1985.

9. D. Saphier, Dynamic Simulator for Nuclear Power Plants (User Manual), ANL-CT-77-20, September, 1984.

10. D. Saphier, Volume I The Level-One Modules Library for DSNP, ANL-CT-77-22, October, 1984.

11. D. Saphier, Volume II The Level-One Modules Library for DSNP, ANL-CT-77-22, September, 1984.

12. E.E. Feldman and D. Mohr, ANL memo to W.N. Beck et al., Subject: "SHRT Group VII (Balance of Plant) Testing Program Development", October 11, 1984.

13. E.E. Feldman and D. Mohr, ANL memo to W.N. Beck, Subject: "Safety Statements for the SHRT Group VII B200 and B100 Series of Tests, and Procedural Input for the B100 Series", March 21, 1985.

14. R.A. Gabel and R.A. Roberts, Signals and Linear Systems, Wiley and Sons, Inc., New York, 1981.

15. H.A. Larson and E.M. Dean, Summary, Analysis, and Comparison of Data from Experiments with the EBR-II Mark IIB Oscillator Rod, ANL-77-87, April 1978.

16. J.S. Bendat and A.G. Piersol, Randon Data Analysis and Measurement Procedures, Wiley and Sons, Inc., New York, 1986.

17. H.I. McCreery et al., EBR-II Description, March, 1983.

18. G.R. Cooper and C.D. McGillem, Methods of Signal and System Analysis, Holt, Rinehart and Winston, Inc., New York, 1967.

Appendix A

Fast Fourier transform program

In Chapter 3, it was shown that to obtain a transform domain representation of x, a signal which is a function of the single variable time, it will be neccessary to solve the equation

$$\underline{X} = \underline{\underline{\phi}}^{-1}\ \underline{x}$$

where $\underline{X}$ is the transform vector and $\underline{\underline{\phi}}$ is a matrix made up of orthogonal functions.

If we express the DFT coefficients as the series

$$\underline{X}(k) = \frac{1}{N}\sum_{n=o}^{N-1} x(n)e^{-j\frac{2\pi kn}{N}} = \sum_{n=o}^{N-1} X(n)W^{ik}$$

where

$$W = e^{-j2\pi/N}$$

then for N equal to eight the matrix can be constructed

$$\begin{vmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \\ X(6) \\ X(7) \end{vmatrix} = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 2 & 4 & 6 & 8 & 10 & 12 & 14 \\ 0 & 3 & 6 & 9 & 12 & 15 & 18 & 21 \\ 0 & 4 & 8 & 12 & 16 & 20 & 24 & 28 \\ 0 & 5 & 10 & 15 & 20 & 25 & 30 & 35 \\ 0 & 6 & 12 & 18 & 24 & 30 & 36 & 42 \\ 0 & 7 & 14 & 21 & 28 & 35 & 42 & 49 \end{vmatrix} \quad \begin{vmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \\ x(6) \\ x(7) \end{vmatrix}$$

or $\underline{X} = \underline{\underline{W}}\ \underline{x}$

This matrix can be reduced by mod 8 since this will render the same values on the complex unit circle. The columns are then reordered to place the columns with even numbers first yielding the new matrix $\underline{\underline{W}}$ shown on the next page.

$$\underline{\underline{W}} \qquad \left|\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 4 & 6 & 1 & 3 & 5 & 7 \\ 0 & 4 & 0 & 4 & 2 & 6 & 2 & 6 \\ 0 & 6 & 4 & 2 & 3 & 1 & 7 & 5 \\ 0 & 0 & 0 & 0 & 4 & 4 & 4 & 4 \\ 0 & 2 & 4 & 6 & 5 & 7 & 1 & 3 \\ 0 & 4 & 0 & 4 & 6 & 2 & 6 & 2 \\ 0 & 6 & 4 & 2 & 7 & 5 & 3 & 1 \end{array}\right| \underline{\underline{Z}}$$

where $\underline{\underline{Z}}$ is the reordering matrix

$$\underline{\underline{Z}} \qquad \left|\begin{array}{cccccccc} 1 & * & * & * & * & * & * & * \\ * & * & 1 & * & * & * & * & * \\ * & * & * & * & 1 & * & * & * \\ * & * & * & * & * & * & 1 & * \\ * & 1 & * & * & * & * & * & * \\ * & * & * & 1 & * & * & * & * \\ * & * & * & * & * & 1 & * & * \\ * & * & * & * & * & * & * & 1 \end{array}\right|$$

$\underline{\underline{W}}$ can now be expressed as

$$\underline{\underline{W}} = \left| \begin{matrix} A & Ir + A \\ & \\ A & Ir + A \end{matrix} \right| \underline{\underline{Z}}$$

or

$$\underline{\underline{W}} = \left| \begin{matrix} I & Ir \\ & \\ I & Ir \end{matrix} \right| \left| \begin{matrix} A & * \\ & \\ * & A \end{matrix} \right| \underline{\underline{Z}}$$

where asterisks are indicative of no operation.

Reordering the second matrix one obtains

$$\underline{\underline{W}} = \underline{\underline{D}} \left| \begin{matrix} 0 & 0 & 0 & 0 & * & * & * & * \\ 0 & 4 & 2 & 6 & * & * & * & * \\ 0 & 0 & 4 & 4 & * & * & * & * \\ 0 & 4 & 6 & 2 & * & * & * & * \\ * & * & * & * & 0 & 0 & 0 & 0 \\ * & * & * & * & 0 & 4 & 2 & 6 \\ * & * & * & * & 0 & 0 & 4 & 4 \\ * & * & * & * & 0 & 4 & 6 & 2 \end{matrix} \right| \underline{\underline{Z}}\ \underline{\underline{Z}}$$

where r is the row index and I is a four by four unit matrix.

Now $\underline{\underline{W}}$ may be expressed as

$$\underline{\underline{W}} = \underline{\underline{D}}\ \underline{\underline{E}}\ \underline{\underline{F}}\ \underline{\underline{Z}}_1 \underline{\underline{Z}}_2$$

or $\underline{\underline{W}} = \underline{\underline{D}}\ \underline{\underline{E}}\ \underline{\underline{F}}\ \underline{\underline{Z}}$

where $\underline{\underline{D}}$, $\underline{\underline{E}}$, $\underline{\underline{F}}$, and $\underline{\underline{Z}}$ are on the following pages

$$\underline{\underline{D}} = \begin{vmatrix} 0 & * & * & * & 0 & * & * & * \\ * & 0 & * & * & * & 1 & * & * \\ * & * & 0 & * & * & * & 2 & * \\ * & * & * & 0 & * & * & * & 3 \\ 0 & * & * & * & 4 & * & * & * \\ * & 0 & * & * & * & 5 & * & * \\ * & * & 0 & * & * & * & 6 & * \\ * & * & * & 0 & * & * & * & 7 \end{vmatrix}$$

$$\underline{\underline{E}} = \begin{vmatrix} 0 & * & 0 & * & * & * & * & * \\ * & 0 & * & 2 & * & * & * & * \\ 0 & * & 4 & * & * & * & * & * \\ * & 0 & * & 6 & * & * & * & * \\ * & * & * & * & 0 & * & 0 & * \\ * & * & * & * & * & 0 & * & 2 \\ * & * & * & * & 0 & * & 4 & * \\ * & * & * & * & * & 0 & * & 6 \end{vmatrix}$$

and

$$\underline{\underline{F}} = \begin{vmatrix} 0 & 0 & * & * & * & * & * & * \\ 0 & 4 & * & * & * & * & * & * \\ * & * & 0 & 0 & * & * & * & * \\ * & * & 0 & 4 & * & * & * & * \\ * & * & * & * & 0 & 0 & * & * \\ * & * & * & * & 0 & 4 & * & * \\ * & * & * & * & * & * & 0 & 0 \\ * & * & * & * & * & * & 0 & 4 \end{vmatrix}$$

$$\underline{\underline{Z}} = \begin{vmatrix} 1 & * & * & * & * & * & * & * \\ * & * & * & * & 1 & * & * & * \\ * & * & 1 & * & * & * & * & * \\ * & * & * & * & * & * & 1 & * \\ * & 1 & * & * & * & * & * & * \\ * & * & * & * & * & 1 & * & * \\ * & * & * & 1 & * & * & * & * \\ * & * & * & * & * & * & * & 1 \end{vmatrix}$$

It can now be seen that these matrices are full of asterisks requiring no operation. There are $f = \log_2 N$ factor matrices to evaluate. Each row of the matrices has two entries. The maximum number of multiplications required is $N\log_2 N$. The program that does the manipulations described here is on the following page.

```
      SUBROUTINE FFT(X,Y,ASIGN,IPOWR)
C
C*****************************************************************
C*****************************************************************
C******                                                    ******
C******   X = THE REAL PART OF A COMPLEX ARRAY.            ******
C******   Y = THE IMAGINARY PART OF A COMPLEX ARRAY.       ******
C******   ASIGN = THE TYPE OF TRANSFORM TO BE DONE:        ******
C******         = -1.0 FOR THE FORWARD TRANSFORM.          ******
C******         = +1.0 FOR THE BACKWARD TRANSFORM.         ******
C******   IPOWR = THE POWER OF 2 FOR THE TRANSFORM.        ******
C******           2**IPOWR = THE NUMBER OF POINTS.         ******
C******                                                    ******
C*****************************************************************
C*****************************************************************
C
      DIMENSION X(1),Y(1)
      N=2**IPOWR
      AN=N
      IF(ASIGN.LT.0.0)GO TO 5
      DO 4 I=1,N
      X(I)=X(I)/AN
      Y(I)=Y(I)/AN
    4 CONTINUE
    5 P=6.283185/AN
      DO 370 L=1,IPORT
      IPOWR1=2**(IPOWR-L)
      M=0
      MKD=2**(L-1)
      DO 360 I=1,MKD
      K1=M/IPOWR1
      K2=0
      DO 190 K=1,IPOWR
      K3=K1-2*(K1/2)
      K1=K1/2
      IF(K3.EQ.0)GO TO 190
      K2=K2+2**(IPOWR-K)
  190 CONTINUE
      A=P*K2
      Y1=COS(A)
      Y2=ASIGN*SIN(A)
      DO 340 J=1,IPOWR1
      K2=M+1
      K3=K2+IPOWR1
      Y3=X(K3)*Y1-Y(K3)*Y2
      Y4=X(K3)*Y2+Y(K3)*Y1
      X(K3)=X(K2)-Y3
      Y(K3)=Y(K2)-Y4
      X(K2)=X(K2+Y3
      Y(K2)=Y(K2)+Y4
      M=M+1
```

```
340 CONTINUE
    M=M+IPOWR1
360 CONTINUE
370 CONTINUE
    DO 480 I2=1,N
    K1=I2-1
    K2=0
    DO 590 K=1,IPOWR
    K3=K1-2*(K1/2)
    K1=K1/2
    IF(K3.EQ.0)GO TO 590
    K2=K2+2**(IPOWR-K)
590 CONTINUE
    IF(K2.GE.I2)GO TO 480
    K2=K2+1
    AK3=X(I2)
    AK4+Y(I2)
    X(I2)=X(K2)
    Y(I2)=Y(K2)
    X(K2)=AK3
    Y(K2)=AK4
480 CONTINUE
    RETURN
    END
```

Appendix B

Programs to calculate and plot analyticaly derived transfer functions

On the following pages are some of the programs to calculate and plot transfer functions based on known system parameters at EBR-II. The programs that are included here are:

1. System transfer function magnitude for low frequencies
2. System transfer function phase for low frequencies
3. Temperature at top of core to external reactivity transfer function magnitude for high frequencies
4. Temperature at top of core to external reactivity transfer function phase for high frequencies

```
C
C     PROGRAM TO PLOT MAGNITUDE OF TRANSFER FUNCTION
C     H(JW)=POWER/RHOEXT
C     USING SHRT AND DSNP ANALYZED DATA AT LOW FREQUENCIES
C     AND ANALYTICALY DERIVED EXPRESSIONS
C
      DIMENSION SILR(7),SILI(7),SFREQ(7),AMODL(7),ANG(7),FTOD(7),
     *FREQ(5),AMAGS(5),AMAGD(5),GFREQ(35),HOMAG(35),HOANG(35),
     *HSMAG(35),HSANG(35),B(6),EL(6)
      COMPLEX FTOD,ONE,SUM,ZERO,FDBK,HSYS
      DATA SILR/17044.8,13596.98,13793.176,12092.883,10233.473,8867.01,
     18357.96/
      DATA SILI/-9941.695,-8074.68,-6060.2,-6153.2,-4572.5,-3109.42,
     2-2353.93/
      DATA SFREQ/.00507,.00775,.01102,.01812,.03091,.05209,.07897/
      DATA FREQ/.0019531,.0058594,.0097656,.021484,.048828/
      DATA AMAGS/4.205,2.217,1.737,1.361,1.058/
      DATA AMAGD/3.655,2.296,1.877,1.450,1.133/
      DATA B,EL/.000236,.00193,.001264,.002747,.000962,.000226,
     1          .0127,.0317,.115,.311,1.4,3.87/
      DATA DETA,ELIFE/.006703,1.1E-7/
      DATA PSI,CHI,OMEGA,RN/-5.100411E-11,-2.00256E-8,195.9618,1143.139/
      DATA ONE/(1.0,0.0)/, ZERO /(0.0,0.0)/
C
C     CREATE FREQUENCY ARRAY
C
      DO 5 J=1,18
       GFREQ(J)=.0015+0.001/2.*FLOAT(J)
    5  CONTINUE
      DO 7 J=1,14
       JJ=18+J
       GFREQ(JJ)=.01+0.01/2.*FLOAT(J)
    7  CONTINUE
C
C     CALCULATE ZERO POWER TRANSFER FUNCTION
C
      TPI=8.0*ATAN(1.)
      DO 800 J=1,32
       SUM=ZERO
       OM=GFREQ(J)*TPI
       DO 100 I=1,6
        SUM=SUM+CMPLX(B(I)/BETA,0.0)/(CMPLX(0.0,OM)+CMPLX(EL(I),0.0))
  100   CONTINUE
        SUM=SUM+CMPLX(ELIFE,0.0)/BETA
        SUM=ONE/(CMPLX(0.0,OM)*SUM
        RP=REAL(SUM)
        AIP=AIMAG(SUM)
        HOMAG(J)=(RP*RP+AIP*AIP)**0.5
        HOANG(J)=ATAN2(AIP,RP)*360./TPI
        AI1=PSI*OM
        AI2=OMEGA*OM
        R1=CHI
        R2=RN-OM*OM
        FDBK=CMPLX(R1,AI1)/CMPLX(R2,AI2)*51.5E6/BETA
C
C     CALCULATE SYSTEM TRANSFER FUNCTION
C
        HSYS=SUM/(CMPLX(1.,0.)-SUM*FDBK)
        RHS=REAL(HSYS)
        AIHS=AIMAG(HSYS)
        HSMAG(J)=(RHS*RHS+AIHS*AIHS)**0.5
        HSANG(J)=ATAN2(AIHS.RHS)*360./TPI
  800  CONTINUE
C
C     CALCULATE SYSTEM TRANSFER FUNCTION USING ROD OSCILLATOR DATA
C
      ANORM=.0068/58.0
      DO 1000 J=1.7
       SILR(J)=SILR(J)*ANORM
       SILI(J)=SILI(J)*ANORM
       FTOD(J)=CMPLX(SILR(J),SILI(J))
       AMODL(J)=CABS(FTOD(J))
       ANG(J)=ATAN2(SILI(J).SILR(J))*57.295779
 1000  CONTINUE
      NPTS=5
      CALL BENSON
      CALL PAGE(8.5,11.)
      CALL AREA2D(7.0,8.)
      CALL XNAME('FREQUENCY HZ$',100)
      CALL YNAME('MAGNITUDE H(JW)  DP/PO/DRHO(DOLLARS) $',100)
      CALL HEADIN('H(JW)=POWER/RHO EXTERNAL$',
     *100,1.,1)
      CALL MESSAG('CROSS=ZERO POWER TRANSFER FUNCTION  X=ANALYTICAL SOL
     *UTION$',100,0.,8.30)
      CALL MESSAG('SQUARE=SHRT TEST    CIRCLE=DSNP   TRIANGLE=ROD OSCIL
     *LATOR$',100,0.,8.05)
      CALL XLOG(.001,3.5,0.0,1.)
      CALL GRID(5,5)
      CALL SPLINE
      CALL CURVE(FREQ,AMAGS,NPTS,1)
      CALL CURVE(FREQ,AMAGD,NPTS,1)
      CALL CURVE(SFREQ,AMODL,7,1)
      CALL CURVE(GFREQ,HOMAG,32,1)
      CALL CURVE(GFREQ,HSMAG,32,1)
      CALL ENDPL(1)
      CALL DONEPL
      STOP
      END
```

```
C
C     PROGRAM TO PLOT PHASE ANGLE OF TRANSFER FUNCTION
C     H(JW)=POWER/RHOEXT USING ANALYTICALY DERIVED EXPRESSION
C     AND SHRT AND DSNP ANALYZED DATA AT LOW FREQUENCIES
C
      DIMENSION SILR(7),SILI(7),SFREQ(7),AMODL(7),ANG(7),FTOD(7),
     *FREQ(5),ANGS(5),ANGD(5),GFREQ(35),HOMAG(35),HOANG(35),
     *HSMAG(35),HSANG(35),B(6),EL(6)
      COMPLEX FTOD,ONE,SUM,ZERO,FDBK,HSYS
      DATA SILR/17044.8,13596.98,13793.176,12092.883,10233.473,8867.01,
     18357.96/
      DATA SILI/-9941.695,-8074.68,-6060.2,-6153.2,-4572.5,-3109.42,
     2-2353.93/
      DATA SFREQ/.00507,.00775,.01102,.01812,.03091,.05209,.07897/
      DATA FREQ/.0019531,.0058594,.0097656,.021484,.048828/
      DATA ANGS/-26.382,-38.24,-32.974,-28.097,-21.174/
      DATA ANGD/-11.78,-28.089,-27.6,-25.023,-20.222/
      DATA B,EL/.000236,.001393,.001264,.002747,.000962,.000226,
     1          .0127,.0317,.115,.311,1.4,3.87/
      DATA BETA,ELIFE/.006703,1.1E-7/
      DATA PSI,CHI,OMEGA,RN/-5.100411E-11,-2.00556E-8,195.9618,1143.139/
      DATA ONE/(1.0,0.0)/, ZERO /(0.0,0.0)/
C
C     CREATE FREQUENCY ARRAY
C
      DO 5 J=1,18
       GFREQ(J)=.0015+0.001/2*FLOAT(J)
    5  CONTINUE
      DO 7 J=1,14
       JJ=18+J
       GFREQ(JJ)=.01+0.01/2.*FLOAT(J)
    7  CONTINUE
C
C     CALCULATE ZERO POWER TRANSFER FUNCTION
C
      TPI=8.0*ATAN(1.)
      DO 800 J=1,32
       SUM=ZERO
       OM=GFREQ(J)*TPI
       DO 100 I=1,6
        SUM=SUM+CMPLX(B(I)/BETA,0.0/(CMPLX(0.0,OM)+CMPLX(EL(I),0.0))
  100   CONTINUE
        SUM=SUM+CMPLX(ELIFE,0.0)/BETA
        SUM=ONE/(CMPLX(0.0,OM)*SUM)
        RP=REAL(SUM)
        AIP=AIMAG(SUM)
        HOMAG(J)=(RP*RP+AIP*AIP)**0.5
        HOANG(J)=ATAN2(API,RP)*360./TPI
        AI1=PSI*OM
        AI2=OMEGA*OM
        R1=CHI
        R2=RN-OM*OM
        FDBK=CMPLX(R1,AI1)/CMPLX(R2,AI2)*51.5E6/BETA
C
C     CALCULATE SYSTEM TRANSFER FUNCTION
C
        HSYS=SUM/(CMPLX(1.,0.)-SUM*FDBK)
        RHS=REAL(HSYS)
        AIHS=AIMAG(HSYS)
        HSMAG(J)=(RHS*RHS+AIHS*AIHS)**0.5
        HSANG(J)=ATAN2(AIHS,RHS)*360./TPI
  800  CONTINUE
C
C     CALCULATE SYSTEM TRANSFER FUNCTION USING ROD OSCILLATOR DATA
C
      ANORM=.0068/58.0
      DO 1000 J=1,7
       SILR(J)=SILR(J)*ANORM
       SILI(J)=SILI(J)*ANORM
       FTOD(J)=CMPLX(SILR(J),SILI(J))
       AMODL(J)=CABS(FTOD(J))
       ANG(J)=ATAN2(SILI(J),SILR(J))*57.295779
1000   CONTINUE
      NPTS=5
      CALL BENSON
      CALL PAGE(8.5,11.)
      CALL AREA2D(7.0,8.)
      CALL XNAME('FREQUENCY HZ$',100)
      CALL YNAME('PHASE ANGLE DEG$',100)
      CALL HEADIN('H(JW)=POWER/RHO EXTERNSL$',
     *100,1.,1)
      CALL MESSAG('CROSS=ZERO POWER TRANSFER FUNCTION   X=ANALYTICAL SOL
     *UTION$',100,0.,8.30)
      CALL MESSAG('SQUARE=SHRT TEST    CIRCLE=DSNP    TRIANGLE=ROD OSCIL
     *LATOR$',100,0.,8.05
      CALL XLOG(.001,3.5,-80.0,10.)
      CALL GRID(5,5)
      CALL SPLINE
      CALL CURVE(FREQ,ANGS,NPTS,1)
      CALL CURVE(FREQ,ANGD,NPTS,1)
      CALL CURVE(SFREQ,ANG,7,1)
      CALL CURVE(GFREQ,HOANG,32,1)
      CALL CURVE(GFREQ,HSANG,32,1)
      CALL ENDPL(1)
      CALL DONEPL
      STOP
      END
```

```
C
C     PROGRAM TO PLOT MAGNITUDE OF TRANSFER FUNCTION
C     H(JW)=CORE TOP TEMP/RHOEXT
C     USING SHRT AND DSNP ANALYZED DATA AT HIGH FREQUENCIES
C
      DIMENSION SILR(7),SILI(7),SFREQ(7),AMODL(7),ANG(7),FTOD(7),
     *FREQ(5),AMAGS(5),AMAGD(5),GFREQ(35),HOMAG(35),HOANG(35),
     *HSMAG(35),HSANG(35),HTMAG(35),HTANG(35),B(6),EL(6)
      COMPLEX FTOD,ONE,SUM,ZERO,FDBK,HSYS,HT
      DATA FREQ/.021485,.064454,.10742,.23633,.53711/
      DATA AMAGS/0.319,0.231,0.211,0.177,0.132/
      DATA AMAGD/0.311,0.239,0.216,0.152,0.138/
      DATA B,EL/.000236,.001393,.001264,.002747,.000962,.000226,
     1          .0127,.0317,.115,.311,1.4,3.87/
      DATA BETA,ELIFE/.006703,1.1E-7/
      DATA PSI,CHI,OMEGA,RN/-5.100411E-11,-2.0056E-8,195.9618,1143.139/
      DATA ONE/(1.1,0.0)/, ZERO /(0.0,0.0)/
      DO 5 J=1,17
       GFREQ(J)=.015+0.01/2.*FLOAT(J)
    5 CONTINUE
      DO 7 J=1,10
       JJ=17+J
       GFREQ(JJ)=.1+0.1/2.*FLOAT(J)
    7  CONTINUE
C
C     CALCULATE ZERO POWER TRANSFER FUNCTION
C
      TPI=8.0*ATAN(1.)
      R3=2.*51.5E6*13.317E-6*44.22/432.
      DO 800 J=1,27
       SUM=ZERO
       OM=GFREQ(J)*TPI
       DO 100 I=1,6
        SUM=SUM+CMPLX(B(I)/BETA,0.0)/(CMPLX(0.0,OM)+CMPLX(EL(I),0.0))
  100   CONTINUE
        SUM=SUM+CMPLX(ELIFE,0.0)/BETA
        SUM=ONE/(CMPLX(0.0,OM)*SUM)
        RP=REAL(SUM)
        AIP=AIMAG(SUM)
        HOMAG(J)=(RP*RP+AIP*AIP)**0.5
        HOANG(J)=ATAN2(AIP,RP)*360./TPI
        AI1=PSI*OM
        AI2=OMEGA*OM
        R1=CHI
        R2=RN-OM*OM
        FDBK=CMPLX(R1,AI1)/CMPLX(R2,AI2)*51.5E6/BETA
C
C      CALCULATE SYSTEM TRANSFER FUNCTION
C
        HSYS=SUM/(CMPLX(1.,0.)-SUM*FDBK)
```

```
C
C     CALCULATE RHOEXT TO TEMPERATURE AT THE TOP OF THE CORE TRANSFER
C     FUNCTION
C
        TEMP=CMPLX(R3,0.)/CMPLX(R2.AI2)
        HT=HSYS*TEMP
        RHS=REAL(HT)
        AIHS=AIMAG(HT)
        HTMAG(J)=(RHS*RHS+AIHS*AIHS)**0.5
        HTANG(J)=ATAN2(AIHS,RHS)*360./TPI
 800   CONTINUE
      NPTS=5
      CALL BENSON
      CALL PAGE(8.5,11.)
      CALL AREA2D(7.0,8.)
      CALL XNAME('FREQUENCY HZ$',100)
      CALL YNAME('MAGNITUDE H(JW)  DT/TO/DRHO(DOLLARS)$',100)
      CALL HEADIN('H(JW)=CORE TOP TEMPERATURE/RHO EXTERNAL$',100,1.,1)
      CALL MESSAG('TRIANGLE=ANALYTICAL SOLUTION$',
     *100,0.,8.30)
      CALL MESSAG('SQUARE=SHRT TEST  CIRCLE=DSNP$',
     *100,0.,8.05)
      CALL XLOG(.01,3.5,0.0,0.1)
      CALL GRID(5,5)
      CALL CURVE(FREQ,AMAGS,NPTS,1)
      CALL CURVE(FREQ,AMAGD,MPTS,1)
      CALL CURVE(GFREQ,HTMAG,27,1)
      CALL ENDPL(1)
      CALL DONEPL
      STOP
      END
```

```
C
C     PROGRAM TO PLOT PHASE ANGLE OF TRANSFER FUNCTION
C     H(JW)=CORE TOP TEMP/RHOEXT
C     USING SHRT AND DSNP ANALYZED DATA AT HIGH FREQUENCIES
C
      DIMENSION SILR(7),SILI(7),SFREQ(7),AMODL(7),ANG(7),FTOD(7)
     *FREQ(5),ANGS(5),ANGD(5),GFREQ(35),HOMAG(35),HOANG(35),
     *HSMAG(35),HSANG(35),HTMAG(35),HTANG(35),B(6),EL(6)
      COMPLEX FTOD,ONE,SUM,ZERO,FDBK,HSYS,TEMP,HT
      DATA FREQ/.021485,.064454,.10742,.23633,.53711/
      DATA ANGS/-33.000,-32.190,-36.219,-56.653,-106.17/
      DATA ANGD/-28.312,-27.944,-29.953,-77.927,-77.880/
      DATA B,EL/.000236,.001393,.001264,.002747,.000962,.000226,
     1          .0127,.0137,.115,.311,1.4,3.87/
      DATA BETA,ELIFE/.006703,1.1E-7/
      DATA PSI,CHI,OMEGA,RN/-5.100411E-11,-2.00556E-8,195.9618,1143.139/
      DATA ONE/(1.0,0.0)/, ZERO /(0.0,0.0)/
      DO 5 J=1,17
       GFREQ(J)=.015+0.01/2.*FLOAT(J)
    5  CONTINUE
      DO 7 J=1,10
       JJ=17+J
       GFREQ(JJ)=.1+0.1/2.*FLOAT(J)
    7  CONTINUE
C
C     CALCULATE ZERO POWER TRANSFER FUNCTION
C
      TPI=8.0*ATAN(1.)
      R3=2.*51.5E6*44.22*13.317E-6/432.
      DO 800 J=1.27
       SUM=ZERO
       OM=GFREQ(J)*TPI
       DO 100 I=1,6
        SUM=SUM+CMPLX(B(I)/BETA,0.0)/(CMPLX(0.0,OM)+CMPLX(EL(I),0.0))
  100   CONTINUE
        SUM=SUM+CMPLX(ELIFE,0.0)/BETA
        SUM=ONE/(CMPLX(0.0,OM)*SUM)
        RP=REAL(SUM)
        AIP=AIMAG(SUM)
        HOMAG(J)=(RP*RP+AIP*AIP)**0.5
        HOANG(J)=ATAN2(AIP,RP)*360./TPI
        AI1=PSI*OM
        AI2=OMEGA*OM
        R1=CHI
        R2=RN-OM*OM
        R2=RN-OM*OM
        FDBK=CMPLX(R1,AI1)/CMPLX(R2,AI2)*51.5E6/BETA
C
C     CALCULATE SYSTEM TRANSFER FUNCTION
C
```

```
        HSYS=SUM/(CMPLX(1.,0.)-SUM*FDBK)
C
C     CALCULATE RHOEXT TO TEMPERATURE AT TOP OF CORE TRANSFER FUNCTION
C
        TEMP=COMPLX(R3,0.)/CMPLX(R2,AI2)
        HT=HSYS*TEMP
        RHS=REAL(HT)
        AIHS=AIMAG(HT)
        HTMAG(J)=(RHS*RHS+AIHS*AIHS)**0.5
        HTANG(J)=ATAN2(AIHS,RHS)*360./TPI
  800  CONTINUE
      NPTS=5
      CALL BENSON
      CALL PAGE(8.5,11.)
      CALL AREA2D(7.0,8.)
      CALL XNAME('FREQUENCY HZ$',100)
      CALL YNAME('PHASE ANGLE DEG$',100)
      CALL HEADIN('H(JW)=CORE TOP TEMPERATURE/RHO EXTERNAL$',100,1.,1)
      CALL MESSAG('TRINGLE=ANALYTICAL SOLUTION$'
     *100,0.,8.30)
      CALL MESSAG('SQUARE=SHRT TEST  CIRCLE=DSNP$',
     *100,0.,8.05)
      CALL XLOG(.01,3.5,-120.,20.)
      CALL GRID(5,5)
      CALL CURVE(FREQ,ANGS,NPTS,1)
      CALL CURVE(FREQ,ANGD,NPTS,1)
      CALL CURVE(GFREQ,HTANG,27,1)
      CALL ENDPL(1)
      CALL DONEPL
      STOP
      END
```

Appendix C

Program to calculate transfer functions from SHRT data

```
C
C NOTE!!!  YOU MUST DIMENSION ALL THE ARRAYS THAT ARE DIMENSIONED TO
C         3000 TO BE AT LEAST LARGE ENOUGH TO HANDLE THE NUMBER OF
C         DATA POINTS YOU WISH TO ANALYZE
C
      COMMON /PLOTD/X(4,3000,3),Y(4,3000,3),INPTP(1)
      DIMENSION FTIR(3000),RHORL(3000),RHOIM(3000),AMODL(3000),
     1ANG(3000),FTOP(3000),POWRL(3000),POWIM(3000),FREQ(3000),
     2FFT(3000),A(10),IYT(40),IY(4),XTITLE(12),SPITLE(4),IHAR(5)
      COMPLEX FTIR,FTOP,FFT
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCC                                                             CCCCC
CCCCCC    DESCRIPTION OF INPUT DATA                                CCCCC
CCCCCC    NPTS=NUMBER OF DATA POINTS TO BE READ AND ANALYZED       CCCCC
CCCCCC    NPLOTS (ALWAYS=1)                                        CCCCC
CCCCCC    IPLOTN (ALWAYS=1)                                        CCCCC
CCCCCC    NUMH=NUMBER OF FREQUENCIES TO BE PLOTTED                 CCCCC
CCCCCC    IPER=NUMBER OF PERIODS OF DATA TO BE READ                CCCCC
CCCCCC    PERIOD=LENGTH IN SECONDS OF FUNDAMENTAL PERIOD           CCCCC
CCCCCC    RHOPIN=REACTIVITY CONSTANT (DOLLARS/INCH)                CCCCC
CCCCCC    IPOWER=POWER TO WHICH 2 IS RAISED TO EQUAL THE           CCCCC
CCCCCC     NUMBER OF DATA POINTS TO BE ANALYZED                    CCCCC
CCCCCC    RHOIM,POWIM=INITIAL IMAGINARY COMPONENTS TO BE           CCCCC
CCCCCC     PASSED TO FFT SUBROUTINE (ALWAYS=0.0)                   CCCCC
CCCCCC    IJUMP=NMBER OF DATA POINTS READ FROM RECORD BUT          CCCCC
CCCCCC     NOT ANALYZED OR IN OTHER WORDS NUMBER OF DATA           CCCCC
CCCCCC      POINTS SKIPPED                                         CCCCC
CCCCCC    IHAR=THE NUMBER OF NUMH(FREQUENCY TO BE PLOTTED)         CCCCC
CCCCCC                                                             CCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      DATA NPTS,NPLOTS,IPLOTN,NUMH,IPER/1024,1,1,5,11/
      DATA PERIOD,RHOPIN,IPOWER/46.545,0.07,10/
      DATA RHOIM,POWIM,IJUMP,IHAR/3000*0.0,3000*0.0,94,1,3,5,11,25/
      DATA XTITLE/'REAL',' H(J','W) $','    ',
     2            'FREQ','UENC','Y HZ','$   ',
     3            'FREQ','UENC','Y HZ','$   '/
      DATA IYT/'IMAG',' H(J','W)$ ','    ',
     2            'MAGN','ITUD','E H(','JW)$',
     3            'PHAS','E H(','JW) ','DGS$'/
C
C     READ DATA FROM DISK
C
      NPI=NPTS+IJUMP
      DO 90 ND=1,NPI
      READ(20)T,(A(I),I=1,10)
C
C     SAVE INITIAL VALUES OF BOTH DATA PARAMETERS
```

```
C
      IF(ND.EQ.1)PINIT=A(1)
      IF(ND.EQ.1)RINIT=A(3)
      IF(ND.EQ.1)WRITE(6,92)RINIT,PINIT
      IF(ND.EQ.1)WRITE(6,93)
      IF(ND.LE.IJUMP)GO TO 90
C
C     READ DATA IF IT IS AFTER IJUMP DATA POINTS
C
      NN=ND-IJUMP
      POWRL(NN)=(A(1)/PINIT)-1.0
      RHORL(NN)=(A(3)-RINIT)*RHOPIN
C
C     WRITE RAW DATA POINTS
C
      WRITE(6,95)NN,T,RHORL(NN),POWRL(NN)
 90   CONTINUE
 92   FORMAT(//,1X,19HINITIAL REACTIVITY=,F12.8,3X,
     114HINITIAL POWER=,F7.4,//)
 93   FORMAT(//,1X,6HDATAPT,5X,4HTIME,7X,10HREACTIVITY,
     19X,5HPOWER,/)
 95   FORMAT(1X,I4,5X,F7.2,2(5X,F12.8))
C
C         COMPUTE FREQUENCY RESPONSE FUNCTION
C
C     TAKE FOURIER TRANSFORM OF INPUT AND PRINT IT
C
      CALL FFT1(RHORL,RHOIM,-1.0,IPOWER)
      WRITE(6,110)
 110  FORMAT(/5X,41HFOURIER TRANSFORM OF REACTIVITY INPUT
     1//4X,1HK,10X,4HREAL,14X,4HIMAG,14X,3HMOD,16X,3HANG/)
      DO 130 J=1,NPTS
      FTIR(J)=CMPLX(RHORL(J),RHOIM(J))
      AMODL(J)=CABS(FTIR(J))
      ANG(J)=ATAN2(RHOIM(J),RHORL(J))*57.295779
      JJ=J-1
      WRITE(6,120)JJ,FTIR(J),AMODL(J),ANG(J)
 120  FORMAT(1X,I4,4(3X,F15.8))
 130  CONTINUE
C
C     TAKE FOURIER TRANSFORM OF OUTPUT AND PRINT IT
C
      CALL FFT1(POWRL,POWIM,-1.0,IPOWER)
      WRITE(6,140)
 140  FORMAT(/5X,37HFOURIER TRANSFORM OF POWER OUTPUT
     1//4X,1HK,10X,4HREAL,14X,4HIMAG,14X,3HMOD,16X,3HANG/)
C
C     CALCULATE SYSTEM TRANSFER FUNCTION AND PRINT IT
C
      DO 160 J=1,NPTS
      FTOP(J)=CMPLX(POWRL(J),POWIM(J))
      AMODL(J)=CABS(FTOP(J))
      ANG(J)=ATAN2(POWIM(J),POWRL(J))*57.295779
      JJ=J-1
      WRITE(6,150)JJ,FTOP(J),AMODL(J),ANG(J)
 150  FORMAT(1X,I4,4(3X,F15.8))
 160  CONTINUE
      WRITE(6,170)
      DO 180 J=1,NPTS
      FFT(J)=FTOP(J)/FTIR(J)
      FREQ(J)=(J-1)/(PERIOD*IPER)
      AMODL(J)=CABS(FFT(J))
      ANG(J)=ATAN2(AIMAG(FFT(J)),REAL(FFT(J)))*57.295779
      JJ=J-1
      WRITE(6,190)JJ,FREQ(J),FFT(J),AMODL(J),ANG(J)
 190  FORMAT(1X,I4,3X,E11.5,4(4X,F15.8))
 180  CONTINUE
C
C     PLOT TRANSFER FUNCTION AT THE EXCITED FREQUENCIES
C
      CALL STRTPL
      INPTP(1)=NUMH
      II=INPTP(1)
      DO 200 ND=1,II
      IH=(IPER*IHAR(ND))+1
      X(1,ND,1)=REAL(FFT(IH))
      Y(1,ND,1)=AIMAG(FFT(IH))
      X(1,ND,2)=FREQ(IH)
      Y(1,ND,2)=AMODL(IH)
      X(1,ND,3)=FREQ(IH)
      Y(1,ND,3)=ANG(IH)
 200  CONTINUE
      DO 220 I=1,3
      DO 210 J=1,4
      IY(J)=IYT((I-1)*4+J)
      SPITLE(J)=XTITLE((I-1)*4+J)
 210  CONTINUE
      CALL GPLOT4(SPITLE,IPLOTN,IY,NPLOTS,I)
 220  CONTINUE
      CALL DONEPL
      STOP
 170  FORMAT(/20X,29HFREQUENCY RESPONSE OF SYSTEM /4X,1HK
     18X,4HFREQ,15X,4HREAL,14X,4HIMAG,15X,3HMOD,15X,3HANG)
      END
      SUBROUTINE FFT1(X,Y,ASIGN,IPOWR)
C
C******************************************************************
C******************************************************************
C*****   X = THE REAL PART OF A COMPLEX ARRAY.               ******
C*****   Y = THE IMAGINARY PART OF A COMPLEX ARRAY.          ******
C*****   ASIGN = THE TYPE OF TRANSFORM TO BE DONE:           ******
C*****         = -1.0 FOR THE FORWARD TRANSFORM.             ******
```

```
C*****         = +1.0 FOR THE BACKWARD TRANSFORM.            ******
C*****   IPOWR = THE POWER OF 2 FOR THE TRANSFORM,           ******
C*****           2**IPOWR = THE NUMBER OF POINTS.            ******
C*****                                                       ******
C******************************************************************
C******************************************************************
C
      DIMENSION X(1),Y(1)
      N=2**IPOWR
      AN=N
      IF(ASIGN.LT.0.0)GO TO 5
      DO 4 I=1,N
      X(I)=X(I)/AN
      Y(I)=Y(I)/AN
    4 CONTINUE
    5 P=6.283185/AN
      DO 370 L=1,IPOWR
      IPOWR1=2**(IPOWR-L)
      M=0
      MKD=2**(L-1)
      DO 360 I=1,MKD
      K1=M/IPOWR1
      K2=0
      DO 190 K=1,IPOWR
      K3=K1-2*(K1/2)
      K1=K1/2
      IF(K3.EQ.0)GO TO 190
      K2=K2+2**(IPOWR-K)
  190 CONTINUE
      A=P*K2
      Y1=COS(A)
      Y2=ASIGN*SIN(A)
      DO 340 J=1,IPOWR1
      K2=M+1
      K3=K2+IPOWR1
      Y3=X(K3)*Y1-Y(K3)*Y2
      Y4=X(K3)*Y2+Y(K3)*Y1
      X(K3)=X(K2)-Y3
      Y(K3)=Y(K2)-Y4
      X(K2)=X(K2)+Y3
      Y(K2)=Y(K2)+Y4
      M=M+1
  340 CONTINUE
      M=M+IPOWR1
  360 CONTINUE
  370 CONTINUE
      DO 480 I2=1,N
      K1=I2-1
      K2=0
      DO 590 K=1,IPOWR
      K3=K1-2*(K1/2)
      K1=K1/2
      IF(K3.EQ.0)GO TO 590
      K2=K2+2**(IPOWR-K)
  590 CONTINUE
      IF(K2.GE.I2)GO TO 480
      K2=K2+1
      AK3=X(I2)
      AK4=Y(I2)
      X(I2)=X(K2)
      Y(I2)=Y(K2)
      X(K2)=AK3
      Y(K2)=AK4
  480 CONTINUE
      RETURN
      END
C
C***********************************************************************
C***********************************************************************
C***                                                          **********
C***   SUBROUTINE TO CREATE PLOTS OF FREQUENCY RESPONSE       **********
C***                                                          **********
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE GPLOT4(XTIT,IPLOTN,IY,NPLOTS,NPAR)
      COMMON /PLOTD/X(4,3000,3),Y(4,3000,3),INPTP(1)
      DIMENSION  TIT(1),XTIT(1),IPN(1),IY(1) ,
     CXX(3000),YY(3000),W(4,3000)
      CALL BGNPL(IPLOTN)
      CALL PAGE(10.,11.5)
      CALL HEIGHT(0.25)
      CALL NOBRDR
      CALL XINTAX
      CALL PHYSOR(1.,1.)
      NEDL=0
      DO 1 I=1,NPLOTS
      N=INPTP(I)
      DO 1 J=1,N
    1 W(I,J)=Y(I,J,NPAR)
      YMIN=W(1,1)
      YMAX=W(1,1)
      XMIN=X(1,1,NPAR)
      XMAX=X(1,1,NPAR)
      DO 10 I=1,NPLOTS
      NPTP=INPTP(I)
      DO 10 J=2,NPTP
      IF(X(I,J,NPAR).GT.XMAX) XMAX=X(I,J,NPAR)
      IF(X(I,J,NPAR).LT.XMIN) XMIN=X(I,J,NPAR)
      IF(W(I,J).LT.YMIN) YMIN=W(I,J)
      IF(W(I,J).GT.YMAX) YMAX=W(I,J)
      IF(W(I,J).LE.0.0) NEDL=1
```

```
   10 CONTINUE
      IF(NPAR.LT.2)GO TO 20
      IF(NPAR.GT.1)GO TO 15
      IF(NEDL.EQ.1) GO TO 20
C     IF LOG PLOT BETTER
      IF(YMIN.EQ.0.0) GO TO 20
      DEC=(YMAX-YMIN)/YMIN
      IF(DEC.LT.000.) GO TO 20
C     DO A LOG PLOT
   15 CONTINUE
      CALL ALGPLT(XMIN,XMAX,10.0,XORIG,XCYC)
      CALL AXSPLT(YMIN,YMAX,7.5,YORIG,YSTEP,YAXIS)
      CALL MIXALF('L/CSTD')
      CALL TITLE(' ',-1,XTIT,100,IY,100,10.00,7.5)
      CALL XLOG(XORIG,XCYC,YORIG,YSTEP)
      CALL RESET('MIXALF')
      CALL GRID(1,1)
      GO TO 21
C     LINEAR PLOT
   20 CONTINUE
      IF(ABS(YMAX).LT.10000.0) CALL YINTAX
      IF(YMIN.LT.100.) GO TO 30
      DEC=YMAX-YMIN
      IF(DEC.EQ.0.0) GO TO 99
      IF(DEC.LT.10.0) YMAX=YMIN+10.0
   30 IF(YMAX-YMIN) 31,99,31
   31 CONTINUE
      CALL MIXALF('L/CSTD')
      CALL TITLE(' ',-1,XTIT,100,IY,100,10.0,7.5)
      CALL GRAF(XMIN,'SCALE',XMAX,YMIN,'SCALE',YMAX)
      CALL RESET('MIXALF')
      CALL GRID(1,1)
C     PLOT DATA
   21 DO 1000 I=1,NPLOTS
      NPTP=INPTP(I)
      DO 1001 J=1,NPTP
      XX(J)=X(I,J,NPAR)
 1001 YY(J)=W(I,J)
      CALL CURVE(XX,YY,NPTP,20)
 1000 CONTINUE
C     CALL MESSAG(TIT,60,0.0,8.00)
C     CALL MESSAG(IPN,16,8.00,-0.95)
C     CALL LOGO(11.8,8.6,1.4,0.01)
      CALL FRAME
   99 CALL ENDPL(IPLOTN)
      RETURN
      END
```

Appendix D

DSNP program to simulate EBR-II and the SHRT test

```
C
C    PROGRAM THAT CREATES FORTRAN DSNP HI-FREQ SIMULATION FILE
C
C                 EBR-II PRIMARY SYSTEM MODEL
C
C
C     DEFINE USER COMMON BLOCKS TO BE USED
C
.  COMON/PRIMP5/ZPUP1,ZPUP2.ZPLHP,ZPLUP,PPOUT9,
.           ZPUPS1,ZPUPS2,PHSATO,PHSAT1,PHSAT2,PHSAT3,
.           PSIPAS,POUTPL,DPCORE,DPBLNK,TOIHXM;
. COMON/PRIMY5/ZFHPO1,ZFHPO2,ZFHPP1,ZFHPP2,ZFLPP1,ZFLPP2,ZFCOR,ZFBLNK,
.           ZFTOTL,ZFHPP3,ZFUPL,AFCVB,WFHPO,WFHPP,RPM1,RPM2,RADRPM,
.           CTGPM,FT541E,WFTOTO,WFTOTP,WPTRIP,ZFBVB,ZFB11,ZFXX09,
.           FL541E,TEPTS,TEPTB,TEPSI,TEPBI,TEPSO,TEPBO,TECRO,TEBLI,
.           TEPTSI,TEPTBI,TEPBS,TEPBB,TXX09C,ZETKLS,DELTF,TECIOI;
. COMON/CONTR5/XPOSRD;
C
C     DEFINE FUEL PIN INFORMATION
C      FUEL PINS: 1 IS AN AVERAGE DRIVER FUEL PIN
C                 2 IS AN AVERAGE PIN IN THE FIRST FOUR ROWS OF PINS
C                   IN THE INSTRUMENTED SUBASSEMBLY XX09
C                 3 IS A STAINLESS STEEL REFLECTOR SUBASSEMBLY
C
. DFPDIST1(1.150D-4,4.D-3,0.D0,23,4,DISTR(4.8D-5,4.8D-5,4.8D-5,7.204D-4,
.         7.204D-4,9.581D-4,9.581D-4,1.201D-3,1.201D-3,
.         2.017D-1,2.096D-1,2.107D-1,1.951D-1,1.729D-1,
.         8.335D-4,8.335D-4,6.65D-4,6.65D-4,5.0D-4,5.0D-4,3.33D-5
.         3.33D-5,3.33D-5),ZZZ1);
. DFPDIST2(1.39D-4,4.D-3,0.D0,15,4,DISTR(3.33D-4,6.67D-4,1.D-3,
.         1.33D-3,1.667D-3,2.017D-1,2.096D-1,2.107D-1,1.951D-1,
.         1.729D-1,1.667D-3,1.33D-3,1.D-3,6.67D-4,3.3D-4),ZZZ1);
. DFPDIST3(1.832D-4,0.333D0,0.D0,15,3,DISTR(3.33D-4,6.67D-4,1.D-3,
.         1.33D-3,1.667D-3,2.017D-1,2.096D-1,2.107D-1,1.951D-1,
.         1.729D-1,1.667D-3,1.33D-3,1.D-3,6.67D-4,3.3D-4),ZZZ1);
.  DESTROY;
. DFPINS(3,1,23,-7134.0,2,15,-55.0,3,15,-61.0);
. DFFPIN1(23,4,10,14,EB,SS,SS,SS,SO, 4.699D-1,3.56D-1,4.699D-1,
.         1.90D-3,1.905D-3,2.21D-3,2.908D-3,3.454D-3,ZZZ1,ZZZ1,
.         1.282D0,SO,ZFCOR*1.286D-4,46,ZZZ1,ZZZ1,0.9513D0,1,
.         ZZZ1,ZZZ1,TRUE,*);
. DFFPIN2(15,4,6,10,EB,SS,SS,SS,SO, 4.699D-1,3.56D-1,4.699D-1,
.         1.90D-3,1.905D-3,2.21D-3,2.908D-3,3.454D-3,ZZZ1,ZZZ1,
.         1.282D0,SO,ZFXX09/55.0D0,30,0.000D0,0.000D0,0.000D0,1,
.         ZZZ1,ZZZ1,TRUE,PRIMY5);
. DFFPIN3(15,3,6,10,SS,SS,SS,SS,SO,0.584D0,0.356D0,0.584D0,1.9397D-2,
.        2.2397D-2,2.2743D-2,2.9471D-2,3.0233D-2,ZZZ1,ZZZ1,1.3D0,SS,
.        ZFBLNK*2.361D-3,15,0.0D0,0.0D0,0.0D0,1,ZZZ1,ZZZ1,TRUE,*);
. DFRDIST1(-3.66D-4,-1.56D-6,-7.98D-6,-3.37D-6,DDIST(40*0.0D0),
.   FDIST(9*0.0D0,0.160D0,0.219D0,.241D0,0.219D0,0.160D0,26*0.0D0),
.   CDIST(9*0.0D0,0.250D0,0.174D0,0.153D0,0.173D0,0.250D0,26*0.0D0),
.   SDIST(0.0187D0,0.0243D0,0.0388D0,5.27D-2,7.13D-2,0.105D0,0.1214D0,
.   0.2335D0,0.3343D0,0.250D0,0.174D0,0.153D0,0.173D0,0.250D0,0.3343D0,
.   0.2335D0,0.1214D0,0.105D0,7.13D-2,5.27D-2,3.88D-2,2.43D-2,1.87D-2,
.        17*0.0D0),-4.3D-6,-4.3D-6);
. DFRDIST2(0.0D0,0.0D0,0.0D0,0.0D0,DDIST(40*0.0D0),
.   FDIST(5*0.0D0,0.160D0,0.219D0,.241D0,0.219D0,0.160D0,30*0.0D0)
.   CDIST(5*0.0D0,0.250D0,0.174D0,0.153D0,0.173D0,0.250D0,30*0.0D0)
.   SDIST(5*0.0D0,0.250D0,0.174D0,0.153D0,0.173D0,0.250D0,30*0.0D0)
.        0.0D0,0.0D0);
. DFRDIST3(0.0D0,0.0D0,-1.49D-6,0.0D0,DDIST(40*0.0D0),
.   FDIST(5*0.0D0,0.160D0,0.219D0,.241D0,0.219D0,0.160D0,30*0.0D0),
.   CDIST(5*0.0D0,0.250D0,0.174D0,0.153D0,0.173D0,0.250D0,30*0.0D0),
.   SDIST(5*0.0D0,0.250D0,0.174D0,0.153D0,0.173D0,0.250D0,30*0.0D0),
.        0.0D0,0.0D0);
.  DESTROY;
C
C     DEFINE CONTROL-ROD AND SAFETY-ROD REACTIVITY DUE TO
C     THERMAL EXPANSION
C
. DEBRCRF(0.1136D0,TECX,1,23,10,14,TEPTS,(ZTFLS*3.639+ZTSFS*3.956
. )/7.595,TANK(500.0D0),SAFTY(0.0209,50.0));
C
C    DEFINE IHX
C
. DFGIHXMA1(20,3026,1.3D0,SO,SO,SS,0.6067D0,2.926D0,6.41D-3,
           7.94D-3,0.0D0,1.0D-1,5.0D-4,15,15,2,1,0.90D0,0.90D0,
           0.0D0,0.0D0);
.  DESTROY;
C
C     DEFINE EBR-II REACTOR OUTLET PIPE WITH HEAT LOSS TO POOL
C
. THPIPEZ1(TECRO,TEAPI,FPAI,TEPTS,30,SO,SS,
.          SO,SS,SO,11.659D0,0.1778D0,0.1842D0,0.2286D0,
.          0.235D0,0.1D0,1.0D-3,20,0.0D0);
C
C     DEFINE COOLANT TRANSPORT DELAYS FROM POOL TO INLET PLENA
C
. DFPIPEPS(TEPTS,TEPSI,3.0100D0,1.D-3,1.D0,ZFHP01/ZRLES,2,60);
. DFPIPEIS(TEPSI,TEPSO,0.1761D0,1.D-3,1.D0,ZFHP01/ZRLES,2,10);
. DFPIPEHS(TEPSO,TEPTSI,0.4149D0,1.D-3,1.D0,ZFHPP1/ZRLES,2,20);
. DFPIPELS(TEPSO,TEPBS,0.1256D0,1.D-3,1.D0,ZFLPP1/ZRLES,2,20);
. DFPIPEPB(TEPTB,TEPBI,3.0100D0,1.D-3,1.D0,ZFHP02/ZRLES,2,60);
. DFPIPEIB(TEPBI,TEPBO,0.1761D0,1.D-3,1.D0,ZFHP02/ZRLES,2,10);
. DFPIPEHB(TEPBO,TEPTBI,0.4149D0,1.D-3,1.D0,ZFHPP2/ZRLES,2,20);
. DFPIPELB(TEPBO,TEPBB,0.1256D0,1.D-3,1.D0,ZFLPP2/ZRLES,2,20);
.  DESTROY;
C
C     DEFINE EBR-II POOL, ACTIVE VOLUME 24% OF TOTAL VOLUME
C
. DFLXPLENS(SO,SO,95.51D0,935.22D2,13.5D4,INFLO(FPAI,TEAPX,
.           ZFHPP3,ZTSFS,ZFUPL,TECRO),EXFLO(ZFHP01,TEPTS,ZFHP02,TEPTB),
.          A,ZELZP,ZETKLS,1.0D5,16.0D4,CORTP5,PRIMY5,IHXMA5,MAIN05);
.  DESTROY;
C
C
C     DEFINE PRIMARY SYSTEM HYDRAULICS
C
C     DEFINE PIPES WHICH MAKE UP THE SYSTEM AND EQUIVALENT PIPES
C     USED TO DESCRIBE THE SUBASSEMBLIES AND LEAKAGE FLOWS
C
.  DEFHPIP01(ZFHP01,TEPSI,TEPSO,5.779D0,3.368D0,2.411D0,27.80D0,
.            0.305D0,7.306D-2,S0,(ZPUPS1+ZPHNW1(4))/2.0D0,PRIMY5,
.            -1.0D0,0.0D0);
.  DEFHPIP02(ZFHP02,TEPSI,TEPSO,5.779D0,3.368D0,2.411D0,27.80D0,
.            0.305D0,7.306D-2,S0,(ZPUPS2+ZPHNW1(5))/2.0D0.PRIMP5,
.            -1.0D0,0.0D0);
.  DEFHPIP03(ZFHPP1,TEPSO,TEPTSI,3.368D0,1.295D0,5.679D0,110.23D0,
.            0.305D0,7.306D-2,S0,(ZPHNW1(4)+ZPHNW1(1))/2.0D0,*,
.            -1.0D0,0.0D0);
.  DEFHPIP04(ZFHPP2,TEPBO,TEPTBI,3.368D0,1.295D0,5.679D0,110.23D0,
.            0.305D0,7.306D-2,S0,(ZPHNW1(5)+ZPHNW1(1))/2.0D0,*,
.            -1.0D0,0.0D0);
.  DEFHPIP05(ZFLPP1,TEPSO,TEPBS,3.368D0,0.862D0,15.375D0,204.70D0,
.            0.102D0,8.171D-3,S0,(ZPHNW1(4)+ZPHNW1(2))/2.0D0,*,
.            -1.0D0,0.0D0);
.  DEFHPIP06(ZFLPP2,TEPSO,TEPBB,3.368D0,0.862D0,15.375D0,204.70D0,
.            0.102D0,8.171D-3,S0,(ZPHNW1(5)+ZPHNW1(2))/2.0D0,*,
.            -1.0D0,0.0D0);
. DEFHPIP07(ZFBLNK/381.0D0,TECI,TECI,0.862D0,0.872D0,0.010D0,0.42D0,
.            3.710D-3,2.160D-5,S0,(ZPHNW1(2)+ZPHNW1(3))/2.0D0,
.            CORTP5,-1.0D0,0.0D0);
. DEFHPIP08(ZFBLNK/381.0D0,ZATT3,ZATT3,0.872D0,3.341D0,2.469D0,2.47D0,
.            2.230D-2,3.900D-4,S0,(ZPHNW1(2)+ZPHNW1(3))/2.0D0,
.            FUEL35,-1.0D0,0.0D0);
. DEFHPIP09(ZFCOR/101.0D0,TECI,TECI,1.295D0,1.709D0,0.414D0,4.71D1,
.            3.490D-2,8.50D-4,S0,(ZPHNW1(1)+ZPHNW1(3))/2.D0,
.            FUEL15,-1.0D0,0.0D0);
. DEFHPIP10(ZFCOR/101.0D0,ZATL1,ZATL1,1.709D0,2.279D0,0.570D0,4.86D0,
.            3.860D-2,1.17D-3,S0,(ZPHNW1(1)+ZPHNW1(3))/2.D0,
.            *,-1.0D0,0.0D0);
. DEFHPIP11(ZFCOR/101.0D0,ZATC1,ZATC1,2.279D0,2.889D0,0.610D0,
.           0.741D0,2.540D-3,1.22D-3,S0,(ZPHNW1(1)+ZPHNW1(3))/2.D0,
.           *,-1.0D0,0.0D0);
. DEFHPIP12(ZFCOR/101.0D0,ZATU1,ZATU1,2.889D0,3.341D0,0.452D0,
.            5.12D0,4.170D-2,1.37D-3,S0,(ZPHNW1(1)+ZPHNW1(3))/2.D0,
.            *,-1.0D0,0.0D0);
.  DEFHPIP13(ZFTOTL,ZLAPTZ,ZLATPZ,3.341D0,7.099D0,11.659D0,40.400D0,
.            0.343D0,9.24D-2,S0,(ZPHNW1(3)+PPOUT9)/2.0D0,TPIPZ5,
.            -1.0D0,0.0D0);
.  DEFHPIP14(ZFTOTL,ZLATSA,ZLATSA,7.099D0,4.36D0,2.93D0,20.55D0,
.            7.857D-3,0.605D0,(PPOUT9+PPOUT0)/2.0D0,IHXMA5,
.            -1.0D0,0.0D0);
.  DEFHPIP15(ZFHPP3,TECI,TECI,1.295D0,0.562D0,0.457D0,4.9880D-2,
.            2.361D-4,2.411D-3,S0,(ZPHNW1(1)+PHSAT3)/2.0D0,*,
.            -1.0D0,0.0D0);
.  DEFHPIP16(ZFUPL,TECRO,TEPTB,3.341D0,4.36D0,1.019D0,0.34057D0,
.            8.268D-4,2.560D-3,S0,(SPHNW1(3)+PHSAT0)/2.0D0,*,
.            -1.0D0,0.0D0);
.   DEFHPIP17(ZFCVB/127.0D0,ZATT1,ZATT1,1.295D0,3.341D0,2.046D0,
.             38.435D0,3.40D-3,7.727D-5,S0,(ZPHNW1(1)+ZPHNW1(3))/
.             2.0D0,*,-1.0D0,0.0D0);
.   DEFHPIP18(ZFBVB/510.0D0,ZATT3,ZATT3,0.862D0,3.341D0,2.479D0,
.             49.604D0,3.40D-3,7.727D-5,S0,(ZPHNW1(2)=ZPHNW1(3))/
.             2.0D0,*,-1.0D0,0.0D0);
.   DEFHPIP19(ZFB11/49.0D0,TECI,TECI,0.862D0,0.872D0,0.01D0,0.968D0,
.             7.366D-3,4.621D-5,S0,(ZPHNW1(2)+ZPHNW1(3))/2.0D0,*,
.             -1.0D0,0.0D0);
.   DEFHPIP20(ZFB11/49.0D0,ZATT3,ZATT3,0.872D0,3.341D0,2.469D0,2.30D0,
.             1.341D-3,3.76D-4,S0,(ZPHNW1(2)+ZPHNW1(3))/2.0D0,*,
.             -1.0D0,0.0D0);
.   DEFHPIP21(ZFXX09,TECI,TECI,1.295D0,1.709D0,0.414D0,0.906D0,
.             6.55D-3,2.026D-4,S0,(ZPHNW1(1)+ZPHNW1(3))/2.0D0,*,
.             -1.0D0,0.0D0);
.   DEFHPIP22(ZFXX09,ZATL2,ZATL2,1.709D0,2.279D0,0.57D0,5.5D0,
.             3.356D-2,8.844D-4,S0,(ZPHNW1(1)+ZPHNW1(3))/2.0D0,
.             FUEL25,-1.0D0,0.0D0);
.   DEFHPIP23(ZFXX09,ZATC2,ZATC2,2.279D0,2.889D0,0.610D0,
.             0.776D0,2.540D-3,8.488D-4,S0,(ZPHNW1(1)+ZPHNW1(3))
.             /2.0D0,*,-1.0D0,0.0D0);
.   DEFHPIP24(ZFXX09,ZATU2,ZATU2,2.889D0,3.341D0,0.452D0,4.32D0
.             ,3.356D-2,8.844D-4,S0,(ZPHNW1(1)+ZPHNW1(3))/2.0D0,
.             *,-1.0D0,0.0D0);
.     DESTROY;
C
C     DEFINE EBR-II AUX. PUMP
C
. DEFAUXPM1(9000.0D0,1861.58D0,5.2782D0,ZFTOTL,TEAPI);
C
C     DEFINE EBR-II PRIMARY PUMPS
C
. DEFPM1(349.0D0,72.8D0,1075.0D0,2633.3D0,ZFHP01,806.0D0,
.        PHSAT1,TEPTS,32.12D0,S0,-1000.0D0,0.0D0,0.0D0,0.1D0);
. DEFPM2(349.0D0,72.8D0,1033.6D0,2848.2D0,ZFHP02,806.0D0,
.        PHSAT2,TEPTB,32.12D0,S0,-1000.0D0,0.0D0,0.0D0,0.1D0);
.     DESTROY;
C
C     DEFINE FLOW PATHS IN THE PRIMARY SYSTEM
C
.   DFLOW01(ZFHP01)=(241.39D0,ZPUPS1,ZPHNW1(4),20,PIPES(01),
.                    PUMPS(*));
.   DFLOW02(ZFHP02)=(241.39D0,ZPUPS2,ZPHNW1(5),20,PIPES(02),
.                    PUMPS(*));
.   DFLOW03(ZFHPP1)=(206.55D0,ZPHNW1(4),ZPHNW1(1),20,PIPES(03),
.                   PUMPS(*));
.   DFLOW04(ZFHPP2)=(206.55D0,ZPHNW1(5),ZPHNW1(1),20,PIPES(04),
.                   PUMPS(*));
.   DFLOW05(ZFLPP1)=(34.847D0,ZPHNW1(4),ZPHNW1(2),20,PIPES(05),
.                   PUMPS(*));
.   DFLOW06(ZFLPP2)=(34.847D0,ZPHNW1(5),ZPHNW1(2),20,PIPES(06),
.                   PUMPS(*));
.   DFLOW07(ZFBLNK)=(45.867D0,ZPHNW1(2),ZPHNW1(3),20,PIPES(07,08),
.                    PUMPS(*));
.   DFLOW08(ZFCOR)=(378.17D0,ZPHNW1(1),ZPHNW1(3),20,PIPES(09,10,11,12),
.                    PUMPS(*));
.   DFLOW09(ZFTOTL)=(455.83D0,ZPHNW1(3),PHSAT0,20,PIPES(13,14),
.                   AUXPS(1));
.   DFLOW10(ZFHPP3)=(21.074D0,ZPHNW1(1)PHSAT3,20,PIPES(15),
.                   PUMPS(*));
.   DFLOW11(ZFUPL)=(5.8063D0,ZPHNW1(3),PHSAT0,90,PIPES(16),
.                  PUMPS(*));
.   DFLOW12(ZFCVB)=(11.339D0,ZPHNW1(1),ZPHNW1(3),20,PIPES(17),PUMPS(*));
.   DFLOW13(ZFBVB)=(13.503D0,ZPHNW1(2),ZPHNW1(3),40,PIPES(18),PUMPS(*));
.   DFLOW14(ZFB11)=(10.319D0,ZPHNW1(2),ZPHNW1(3),20,PIPES(19,20),
.                   PUMPS(*));
.   DFLOW15(ZFXX09)=(2.603D0,ZPHNW1(1),ZPHNW1(3),20,PIPES(21,22,
.                    23,24),PUMPS(*));
.     DESTROY
. CNCTZF(ZFHP01,ZFLOW(1),ZFHP02,ZFLOW(2),ZFHPP1,ZFLOW(3),
.        ZFHPP2,ZFLOW(4),ZFLPP1,ZFLOW(5),ZFLPP2,ZFLOW(6),
.        ZFBLNK,ZFLOW(7),ZFCOR,ZFLOW(8),ZFTOTL,ZFLOW(9),
.        FPAI,ZFTOTL,ZFHPP3,ZFLOW(10),
.        ZFUPL,ZFLOW(11),ZFCBV,ZFLOW(12),ZFBVB,ZFLOW(13),
.        ZFB11,ZFLOW(14),ZFXX09,ZFLOW(15),
.        ZFLCOR,ZFCOR+ZFCVB+ZFXX09+ZFBLNK+ZFB11+ZFBVB);
C
C     DEFINE EBR-II PRIMARY SYSTEM HYDRAULIC NETWORK TO CALCULATE
C     PRESSURE AT EACH JUNCTION.
C
.     DFLOWNET(1,5);
.     FEFNETW01(INFLO(03,04),EXFLO(08,10,12,15),IOPRS(PHSAT3,10),
.     INJUN(4,3,5,4),EXJUN(3,8,3,12,3,15),NETWK(1));
.     DEFNETW02(INFLO(05,06),EXFLO(07,13,14),IOPRS(*),
.     INJUN(4,5,5,6),EXJUN(3,7,3,13,3,14),NETWK(1));
.     DEFNETW03(INFLO(07,08,12,13,14,15),EXFLO(09,11),IOPRS(PHSAT0,9,
.     PHSAT0,11),INJUN(1,8,1,12,1,15,2,7,2,13,2,14),EXJUN(*),NETWK(1));
.     DEFNETW04(INFLO(01),EXFLO(03,05),IOPRS(ZPUPS1,1),INJUN(*),
.     EXJUN(1,4,2,6),NETWK(1));
. DFLOWNEND(1);
.     DESTROY;
C
C     DEFINE POOL HYDRAULICS
C
.  DFCAVT01(7.595D0,249.0D0,(ZTFLS*3.693+ZTSFS*3.956)/7.595,44.293D0,
.           INFLO(09,4.36D0,PHSAT0,10,0.562D0,PHSAT3,11,4.36D0,
.                PHSAT0),
.          EXFLO(01,5.779D0,PHSAT1,02,5.779D0,PHSAT2),
.          PIPES(01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,
.          17,18,19,20,21,22,23,24),353.819D0,SO);
C
C      END OF PRIMARY HYDRAULIC DEFINITIONS
C
C     DEFINE REACTOR HPP INLET MIXING PLENUM
C
.  DFMPPLEN3(SO,SS,1.964D0,7003.3D0,1.223D4,INFLO(ZFHPP1,TEPTSI,
.           ZFHPP2,TEPTBI),EXFLO(ZFCOR,TECI,ZFCVB,TECI,ZFHPP3,
.           TECI,ZFXX09,TECI),0.2D0,PRIMY5,CORTP5);
C
C     DEFINE REACTOR LPP INLET MIXING PLENUM
C
.  DFMPPLEN4(SO,SS,1.06D0,757.70D0,9.750D3,INFLO(ZFLPP1,TEPBS,
.           ZFLPP2,TEPBB),EXFLO(ZFBLNK,TEBLI,ZFBVB,TEBLI,ZFB11,
.           TEBLI),0.2D0,PRIMY5);
C
C     DEFINE REACTOR OUTLET MIXING PLENUM
C
.  DFMPPLEN5(SO,SS,2.79D0,1491.0D0,1.856D4,INFLO(ZFCOR,TECX,
.           ZFBLNK,ZTC3(16),ZFCVB,TECX,ZFBVB,ZTC3(16),ZFB11,ZTC3(16),
.           ZFXX09,ZTC2(16)),EXFLO(ZFTOTL,TECRO,ZFUPL,TECRO),0.2D0,
.           PRIMY5,CORTP5,FUEL15,FUEL25,FUEL35);
.     DESTROY;
C                REACTOR SHUTDOWN SYSTEM
C  DSCRAM(0.45D0,0.020D0,-2.479D-2,1.0D0,1.0D10,0.0D0);
C  DTRIPF1P(POWER LEVEL,1.15D0,SENSORS(PNJ),DELAY,ELEC(*),
C          GE,0.055D0,NEUTR5);
C  DTRIPF2P(SOT,510.0D0,SENSORS(TECX),THREEP,ELEC(0.5D0,0.178D0,
C          0.06D0),GE,0.125D0,CORTP5);
C  DTRIPF3P(FLOW FT541E,19.0D0,SENSORS(WFTOTP),THREEP,ELEC(1.0D0,
C          0.264D0,0.069D0),LE,0.276D0,PRIMY5);
C
C
C     END OF MODEL DEFINITION STATEMENTS
C
BEGIN AT 0.0D0
       WPTRIP=ZZZ1
       ZPTRIP(1)=ZZZ1
       PSIPAS=6894.757D0
       RADRPM=60.D0/(ZZZ2*ZZZ314)
       TICPU=120.0D0
C
C     INITIALIZE HYDRAULIC PARAMETERS TO THOSE DEFINED ABOVE
C
      CALL HFLOW1(0,ZZZ1,ZZZ1,10,0)
C
C     MODIFY FLOWS AND ABS. ERRORS BY USING ( FL541E ) TO SET DESIRED
C     REACTOR FLOW FOR THE SIMULATION
```

```
C
      DO 9 I=1,15
      ZFLOW(I)=ZFLOW(I)*FL541E
      ZERFLO(I)=ZERFLO(I)*FL541E
   9  CONTINUE
C
C     STEADY STATE SOLUTION, THERMAL PART FIRST
C
.   CNCTZF;
.   NEUTP1;
    1 CONTINUE
.   GAMAR1;
.   TPOWR2;
.   FDBEK2;
.   MPPLEN5;
.   CRDFB4;
      CALL BOWFB(TECX,TECI,TECRO)
.   TPIPZ1;
.          CONVERGR(TECX,0.001,1,25,P);
   25 CONTINUE
.   IHXGA1;
.   SFCTL4;
      CALL SHOOL((TEPSI+TEPBI)/ZZZ2)
.   LXPLENS;
.   PIPEPS; PIPEIS; PIPEHS; PIPELS;
.   PIPEPB; PIPEIB; PIPEHB; PIPELB;
.   MPPLEN3;
.   MPPLEN4;
      CALL GRIDFB(ZTPM3,TECI)
      CALL FLOWST(ZEGIIA,TEAHI,TEAHX,TECI,371.ODO,FHAI)
.     CONVERGA(TECI,0.05,25,50,P);
.    DUMP IHXMA5 TO IO6;
.    DUMP CORTP5 TO IO6;
.    DUMP PRIMY5 TO IO6;
C
C     HYDRAULIC SOLUTION
C
. CNCTZF;
. CAVTO1;
. HNETW1;
      ZPHNW1(3)=PHSATO-ZDPFL(9)
      ZPHNW1(1)=ZPHNW1(3)-ZDPFL(8)
      ZPHNW1(2)=ZPHNW1(3)-ZDPFL(7)
C     CALL EQXLE4(12,ZPHNW1(1),ZPHNW1(3),17,20,0);
. FLOW10; FLOW11; FLOW12; FLOW14; FLOW15;
      ZFLOW(3)=(ZFLOW(8)+ZFLOW(10)+ZFLOW(12)+ZFLOW(15))/ZZZ2
      ZFLOW(4)=ZFLOW(3)
      ZPHNW1(4)=ZPHNW1(3)-ZDPFL(8)-ZDPFL(3)
      ZPHNW1(5)=ZPHNW1(3)-ZDPFL(8)-ZDPFL(4)
. FLOW05; FLOW06;
      ZFLOW(1)=ZFLOW(3)+ZFLOW(5)
      ZFLOW(2)=ZFLOW(4)+ZFLOW(6)
      ZFLOW(13)=ZFLOW(5)+ZFLOW(6)-ZFLOW(7)-ZFLOW(14)
C     CALL EQXLE4(13,ZPHNW1(2),ZPHNW1(3),18,20,0);
      ZPREQ(1)=-DEPF(1,LOOP)-DELPF(3,LOOP)-DELPF(8,LOOP)-DELPF(9,LOOP)
      ZPREQ(1)=ZPREQ(1)+PHSAT0-PHSAT1
      ZPREQ(2)=-DELPF(2,LOOP)-DELPF(4,LOOP)-DELPF(8,LOOP)-DELPF(9,LOOP)
      ZPREQ(2)=ZPREQ(2)+PHSAT0-PHSAT2
      ZPUP1=APUMP0(1,ZFHP01,TEPSI,PHSAT1,LOOP)
      ZPUP2=APUMP0(2,ZFHP02,TEPBI,PHSAT2,LOOP)
      ZPUPS1=ZPUP1+PHSAT1
      ZPUPS2=ZPUP2+PHSAT2
. HNETW1;
. CAVT01;
. FLOW05; FLOW06; FLOW07; FLOW08; FLOW09;
. FLOW10;FLOW11;FLOW12;FLOW13;FLOW14;FLOW15
      ZFLOW(3)=ZFLOW(1)-ZFLOW(5)
      ZFLOW(4)=ZFLOW(2)-ZFLOW(6)
. CNCTZF;
      WFTOT0=ZFTOTL
      WFHP0=ZFHPP1
      WFTOTP=ZFTOTL/WFTOT0*ZZZE2
      WFHPP=ZFHPP1/WFHP0*ZZZE2
      CTGPM=ZZZ1/(ZRPLE5*6.30902D-5)
      FT541E=((ZFLOW(9)*CTGPM*1.0042D-4)**2)*21.86D0
      PPOUT9=ZPHNW1(3)+ZDPH(13)
      CALL SODEN1(TEPTS,RHO,P)
      POUTPL=(ZPHNW1(3)-PHSAT0-1.019D0*ZZZ981*RHO)/PSIPAS
      DPCORE=(-RHO*ZZZ981*2.046D0-ZDPFL(8))/PSIPAS
      DPBLNK=(-RHO*ZZZ981*2.479D0-ZDPFL(7))/PSIPAS
C
C     END OF STEADY-STATE HYDRAULICS
C
.                  CONVERGA(TEAHX,0.05D0,1,10,P);
C
C     OUTER ITERATION LOOP ON ( TEAHX ), WHEN SATISFIED END OF
C     STEADY STATE SOLUTION
C
     TECI0I=TECI
     SECFL0=FHAI
     TXX09C=ZTC2(11)*1.8+32.0
     T0IHXM=TEAHX
. DUMP ALL TO I06;
C
C     DYNAMIC SECTION
C
SIMULATE LOOP01 STIFF1 TDV=2
      TPI=6.2831853
C  INSERTION OF HIGH FREQUENCY REACTIVITY
      IF(TIME.GE.10.0)RKCN=.0159*.007*(DSIN(TPI*.021480*TIME+.88455)+
     1DSIN(TPI*.064450*TIME+4.18876)+DSIN(TPI*.107400*TIME+5.759587)+
     2DSIN(TPI*.23630*TIME+0.872665)+DSIN(TPI*.53710*TIME+6.108652))
```

```
.   NEUTP1;
.   GAMAR1;
.   TPOWR2;
.   CORTP2;
.   FDBEK2;
.   CRDFB4;
      CALL BOWFB(TECX,TECI,TECRO)
      CALLL GRIDFB(ZTPM3,TECI)
.   MPPEN5;
.   TPIPZ1;
.   IHXGA1;
C   SFCTL4;
.   LXPLENS;
.   PIPEPS; PIPEIS; PIPEHS; PIPELS;
.   PIPEPB; PIPEIB; PIPEHB; PIPELB;
.   MPPLEN3;
.   MPPLEN4;
.   LAG(TOIHXM)=(TEAHX,8.0D0);
      CALL SHCOOL((TEPSI+TEPBI)/ZZZ2)
      DELTF=(TECI=TECIOI)*1.8D0
      TXX09C=ZTC2(11)*1.8+32.0
       PPOUT9=ZPHNW1(3)+ZDPH(13)
       CALL SODEN1(TEPTS,RHO,P)
       POUTPL=(ZPHNW1(3)-PHSAT0-1.019D0*ZZZ981*RHO)/PSIPAS
       WFTOTP=ZFTOTL/WFTOT0*ZZZE2
       WFHPP=ZFHPP1/WFHP0*ZZZE2
       CTGPM=ZZZ1/(ZRPLE5*6.30902D-5)
       FT541E=((ZFLOW(9)*CTGPM*1.0042D-4)**2)*21.86D0
       DPCORE=(-RHO*ZZZ981*2.046D0-ZDPFL(8))/PSIPAS
       DPBLNK=(-RHO*ZZZ981*2.479D0-ZDPFL(7))/PSIPAS
TERMINATE AT 522.0D0
.  DUMP ALL TO I06;
.  GRAPH4;
P ZWTHMW,RKCN,RKFB,REAK,TXX09C,ZTC1(23),TECRO,ZTZP(10),ZTZP(20),
P TEAPI,TEAHX,TOIHXM,ZTPF4
P BY 0.5D0 TO 1034.0D0
P 'DSNP BOP TESTS B103 AND B104'
. DATA(NEUTR)=EPSN1/1.D-4/,EPSN2/1.D-5/,EL/2.4D-7/,SOURCE/0.0D/,
.        BETA/2.3142D-4, 1.3669D-3, 1.2429D-3, 2.6955D-3, 9.4398D-4,
.             2.2199D-4/, ELAMDA/1.27D-2, 3.17D-2, 1.15D-1, 3.11D-1,
.             1.4D0, 3.87D0/, PN0/1.D0/;
. DATA(TIMER)=DTMAX/1.0D0/, LETM/2/, DELT/1.D-2/,DTMIN/1.D-6/,
.             IPLOT/15/,ICRT/0/;
. DATA(TPOWR)=PW0/47.94D6/;
. DATA(GAMAR)=IGAM/4/,GAM/1.D-5, 3.D-3, 1.D-2, 4.D-2, 0D0/,
.             BGM/2.7D-1, 2.1D-1, 1.6D-1, 3.6D-1, 0D0/,
.             WGT0/3.060D6/, GA/0.01D0/;
. DATA(GRIDP)=RGRIDC/-9.86D-6/;
. DATA(BOWFB)=ZTDBOW/ 0.0,  3.56,  7.11,  10.67,  14.22,  17.78,
.             21.33,  24.89,  28.44,  32.00,  35.56,  39.11,  42.67,
.             46.22,  49.78,  53.33,  56.89,  60.44,  64.00,  67.56,
.            71.11,  74,67,  78.22,  81.78,  85.33,  88.89,  92.44,
.            96.00,  99.56, 103.11, 106.67, 110.22, 166.67, 333.33, .
.            555.56/,
.            RTBOW/ 0.0DO,     -5.2046D-3,  -9.7208D-3,   -1.2854D-2,
.            -1.4762D-2,  -1.5595D-2,  -1.5495D-2,  -1.4596D-2,
.            -1.3024D-2, -1.0898D-2,  -8.3289D-3,  -5.4184D-3,
.            -2.2616D-3, 1.0546D-3, 4.4511D-3, 7.8569D-3, 1.1209D-2,
.             1.4451D-2, 1.7537D-2, 2.0427D-2, 2.3089D-2, 2.5499D-2,
.             2.7641D-2, 2.9508D-2, 3.1099D-2, 3.2423D-2, 3.3494D-2,
.             3.4336D-2, 3.4981D-2, 3.5468D-2, 3.5844D-2, 3.6165D-2,
.             4.2756D-2, 4.9268D-2, 8.834D-2/,KNBOW,KNRITP/35,3/,
.             ZBOWNF/1.0DO/;
. DATA(CORTP)=UCA/9.D5/, ULA/2.88D6/, CF/188.3DO/,
.            CL/557.DO/, VC/6.15D-2/, QMF/409.DO/,
.            QML/218.2DO/, TECI/371.12DO/, ZFLCOR/483.0DO/,
.            ZEGXC/62.5D6/, TECX/473.DO/, TELA/394.DO/,
.       TECA/394.DO/,TECAO/394.DO/,TEFA/394.DO/,TEFAO/394.DO/,
.       TELAO/394.DO/;
. DATA(IHXMA)=TEAPI/440.DO/,TEAPX/371.1DO/,TEAHI/307.2DO/,TEAHX/436.9DO/
.           ,FPAI/469.3DO/,FHAI/74.66DO/;
.  DATA(LXPLS)=ZTMLS/371.12DO/,ZTPLS/371.12DO/,ZTFLS/371.12DO/;
. DATA(SECFL)=SPRTUP,SPRTDN,SPFSS,SPTAU,SPCHL,SPCLL/7.4D-3,1.1D-2
.           0.173DO,0.25DO,1.0DO,1.0DO/,SMINF/0.10DO/;
.    DATA(PRIMY)=TEPTS,TEPTB,TEPSI,TEPBI,TEPSO,TEPBO,TECRO,TEBLI,
.                TEPTSI,TEPTBI,TEPBS,TEPBB/12*371.1DO/,
.                FL541E/1.000DO/;
.    DATA(AUXC1)=ZCURT1/9000.0DO/;
.    DATA(HFLOW)=ZNESUB(7),ZNESUB(8),ZNESUB(12),ZNESUB(13),ZNESUB(14)/
.                381.0DO,101.0DO,127.0DO,510.0DO,49.0DO/;
.    DATA(TERMN)=KPRLEV/1/;
.    DATA(MPPL3)=ZTPF3/371.12DO/;
.    DATA(MPPL4)=ZTPF4/371.12DO/;
.    DATA(MMPL5)=ZTPF5/371.12DO/;
.    DATA(APUMP)=ZTPUMX/371.12DO,171.12DO/;
C
FUNCTIONS
      SUBROUTINE FRICT1(RE,FF)
C------------------- DSNP REV. 03/4. JUNE 9, 1986 EMD------------------
C**** CALCULATES SMOOTH PIPE FRICTION FACTOR CORRELATION USING KARMAN
C**** NIKURADZE EQUATION. CHANGED BY E.M. DEAN IN THE REGION
C     FOR 2000  RE  30000.
.   TICIL; TASNA('FRICT1');
      IT=0
      IF(RE.LT.ZZZ1) RE=ZZZ1
C     IF(RE.LE.22.D2) FF=64.DO/RE
C     IF(RE.GT.22.D2) FF=0.184DO/RE**0.2
      IF(RE.LE.20.D2) FF=64.DO/RE
      IF((RE.GT.20.D2).AND.(RE.LE.40.D2)) FF=2.1703D-3*RE**0.35402
      IF((RE.GT.40.D2).AND.(RE.LE.3.D4)) FF=0.407352/RE**(0.277132)
      IF(RE.LE.3.D4) GOTO 99
      FF=0.184DO/RE**0.2
```

```
    1 F1=FF
      SF=DSQRT(FF)
      G=0.87D0 * DLOG(RE*SF) - 0.8D0 - ZZZ1/SF
      DG=0.5D0 / FF * (0.87D0 + ZZZ1 / SF)
      FF=F1-G/DG
      IF(DABS((FF=F1)/FF).LE.ZZZER) GOTO 99
      IT=IT+1
      IF(IT.LT.10) GOTO 1
. ER2ROR(9,3,IT,RE);
   99 KSUBST=KSUBST-1
      RETURN
      END
C
      SUBROUTINE SHCOOL(TAVI)
C     SIMPLE ROUTINE TO ACCOUNT FOR HEAT LOSS FROM EBR-II SHUTDOWN
C     COOLERS, AT 375.6 C THE DAMPERS TRIP OPEN.
. COMIN PRIMY5;
      ZETKLS=1.5D5
      IF(TAVI.GT.375.6) ZETKLS=3.25D5
      RETURN
      END
      SUBROUTINE BOWFB(TOUT,TIN,TAVOUT)
C  ROUTINE TO CALCULATE FEEDBACK REACTIVITY THAT IS A FUNCTION
C  OF DELTA TEMPERATURE. BOWING REACTIVITY IS NORMALLY SIMULATED
C  IN THIS WAY.
.  COMON /BOWFB5/RTBOW(35),ZTDBOW(35),RBOWFB,RBOWO,ZBOWNF,KNBOW,KNRITP;
.  TICIL; COMIN FDBEK5,BOWFB5,TIMER5; LOGICAL LFIRST/.FALSE./;
.  TASNA('BOWFB ');
      IF (JTIM.GT.0) GO TO 20
      IF(LFIRST) GO TO 1
      LFIRST=.TRUE.
      DO 2 I=1,KNBOW
      RTBOW(I)=RTBOW(I)*6.828D-3
    2 CONTINUE
    1 CONTINUE
      ZBOWNF=(TAVOUT-TIN)/(TOUT-TIN)
      DTEMP=ZBOWNF*(TOUT-TIN)
      IX=1
      CALL FUNCT1(DTEMP,RBOWO,KNRITP,ZTDBOW,RTBOW,KNBOW,IX)
      PRDBOW=RBOWO
      GO TO 99
   20 CONTINUE
      IF(KNBOW.EQ.0) GO TO 99
      DTEMP=ZBOWNF*(TOUT-TIN)
      IX=1
      CALL FUNCT1(DTEMP,RBOWFB,KNRITP,ZTDBOW,RTBOW,KNBOW,IX)
      RKB=RBOWFB-RBOWO
      RKFB=RKFB+RKB
   99 KSUBST=KSUBST-1
      RETURN
      END
      SUBROUTINE GRIDFB(TMETAL,TREACI)
C  CALCULATES FEEDBACK REACTIVITY FOR GRID PLATE EXPANSION
. COMON /GRIDP5/ZTGRPO,RGRIDC,RKGRP,PRDGRP;
. TICIL; COMIN FDBEK5,TIMER5,GRIDP5; TASNA('GRIDFB');
      RKGRP=ZZZO
      IF(JTIM.GT.0) GO TO 20
      ZTGRPO=TMETAL
      PRDGRP=RGRIDC*(TMETAL-TREACI)
      TPRD=TPRD+PRDGRP
      GO TO 99
   20 CONTINUE
      RKGRP=RGRIDC*(TMETAL-ZTGRPO)
      RKFB=RKFB+RKGRP
   99 KSUBST=KSUBST-1
      RETURN
      END
      SUBROUTINE FLOWST(PO.TIN,TOUT,TCAL,TEMP,FLOW)
C     CALCULATES SECONDARY FLOW REQUIRED TO GET DESIRED REACTOR INLET
C     TEMPERATURE ( TEMP ) WITHIN 0.2 C.
      IMPLICIT REAL *8(A-H,O-Z)
      LOGICAL FLOWSI/.FALSE./
      IF(FLOWSI) GO TO 99
      TD=TEMP-TCAL
      IF(DABS(TD).LE.0.2D0) FLOWSI=.TRUE.
      CALL SODCP1((TIN+TOUT)/2.0D0,CPS,A)
      FLOW=DABS(PO)/((TOUT-TIN+TD)*CPS)
  99  RETURN
      END
C
C
LIBRARY LEVEL-ZERO
C
C  MACRO'S REQUIRED FOR THIS MODEL
C
M <DEBRCRF(#,#,#,#,#,#,#,#,#,TANK(#),SAFTY(#,#))>=%D CRDFB4>
M <CNTRL4>=<%B CALL CNTRL4(LOOP)>
M <SFCTL4>=<%B CALL SFCTL4(FHAI,TECI,TEAPX,LOOP)>
M <DEFAUXPM#(#,#,#,#,#)>=<%D AUXPMO>
M  <DEFPM#(#,#,#,#,#,#,#,#,#,#,#,#,#,#)>=<%D APUMPO >
C
C  COMPONENT MODULES
C
*AUXPMO
C    SIMPLE DC EM PUMP MODEL OF EBR-II PRIMARY AUX. PUMP
M <DEFAUXPM#(#,#,#,#,#)>=<%Z21 %XO %M<AUXP#01>=<AUXPE#01(#05,#06)> %E
       FUNCTION AUXPE#01(FLOW,T); IMPLICIT REAL*8(A-H,O-Z);
       COMON/AUXC#015/ZPAUX#01,ZCURT#01,ZAUXT#01;
       COMIN TERMN5,TIMER5,CONST5,AUXC#015;
       DATA(AUXC#01)=ZAUXT#01/1.0D0/;
       DATA CURTR,PAUXO,AUXC/#02,#03,#04/;
       TASNA('AUXPMO');
       CALL SODEN1(T,RO,P);
       ZPAUX#01=ZAUXT#01*ZCURT#01/CURTR*PAUXO*(ZZZ1-AUXC/RO*FLOW);
       AUXPE#01=ZPAUX#01;
    99 KSUBST=KSUBST-1; RETURN; END;  %IOO>
       END
*APUMPO
C    HOMOLOGOUS PUMP MODEL, SPEED CALCULATED USING REL. SPEED
C    FUNCTIONS
M  <DEFPM#(#,#,#,#,#,#,#,#,#,#,#,#,#,#)>=<%W040+001 %S323
   %M <HOMP#01>=<APUMPO(%V040,#06,#09,#08,LOOP)> %E
   INCLUDE APUMOO;
   DATA(APUMP)=PMRF(%V040),PMRH(%V040),PMRR(%V040),PMRT(%V040),
   PMR(%V040),PMINT(%V040),
   PMMAT(%V040)/#02,#03,#04,#05,#07,#10,'#11'/,
   PMLOCF(%V040),PMLOCR(%V040)/#14,#15/,
   PMRPSP(%V040),PTORQP(%V040)/#12,#13/ %IOO >
      END
*APUMOO
M <APUMOO>=<FUNCTION APUMPO(KZ,PMF,PMT,PMP,LOOP);
   IMPLICIT REAL*8(A-H,O-Z);
   DATA SNAME/'APUMPO'/;
   COMON/APUMP5/ PMRF(%V040),PMRH(%V040),PMRR(%V040),PMRT(%V040),
   PMR(%V040),PMINT(%V040),PMTRIP(%V040),PMTORQ(%V040),PMTORD(%V040),
   PMHEAD(%V040),PMMAT(%V040),PMRHO(%V040),PMDRPM(%V040),PMFTOR(%V040),
   ZCPT1(%V040),ZCPT2(%V040),PMRO(%V040),ZPREQ(%V040),
   PMRPSP(%V040),PTORQP(%V040),PMDECA(%V040),
   PMLOCR(%V040),PMLOCF(%V040),ZTPUMX(%V040);
   COMIN TIMER5,TERMN5,CONST5,DERIV5,APUMP5;
   DIMENSION H(7,3),T(9,3)E(4,3);
   DATA H/
    0.63450,0.13356,-0.20591,-2.04642,4.00471,-2.58014,0.57213,
    -42.00515,94.73749,-84.43033,38.48265,-9.39663,1.16426,-0.05743,
    -723.46647,455.46361,-88.35045,1.47828,1.21901,-0.09357,0.0/;
   DATA T/
    -0.68833,2.04499,0.67489,-9.96885,16.65863,-10.65992,2.38784,0.,0.,
    48.89663,-124.11887,121.44961,-52.19549,5.14022,3.78029,-1.50655,
    0.21971,-0.01176, -964.94245,631.72352,-134.35488,6.41557,
    1.06868,-0.09781,0.0,0.0,0.0/;
   DATA E/
   5.4,0.0,421.3,5.4,5.40,0.0,1205.04,5.4,5.4,23.229,0.000,5.4/;
  KSUBST=KSUBST+1;KSUBST(KSUBST)=SNAME;
  ITER=0;
   CALL SODEN1(PMT,PMRHO(KZ),PMP); CALL SODCP1(PMT,CP,PMP);
  ZFLON=PMF/PMRF(KZ);
  2 ZRPMN=PMR(KZ)/PMRR(KZ);
  T1=ZFLON*ZFLON+ZRPMN*ZRPMN;
  T2=ZZZ314+DATAN2(ZFLON,ZRPMN);
  ZCPT1(KZ)=T1; ZCPT2(KZ)=T2;
  I=1;
  IF(T2.GT.1.499) I=2;
  IF(T2.GT.4.784) I=3;
```

```
 PMHEAD(KZ)=T1*PMRH(KZ)*(H(1,I)+T2*(H(2,I)+T2*(H(3,I)+T2*(H(4,I)+
   T2*(T(5,I)+T2*(T(6,I)+T2*((7,I)))))));
 PMTORQ(KZ)=T1*PMRT(KZ)*(T(1,I)+T2*(T(2,I)+T2*(H(3,I)+T2*(H(4,I)+
   T2*(T(5,I)+T2*(T(6,I)+T2*(T(7,I)+T2*(T(8,I)+T2*T(9,I)))))))));
 I=1;
 A1=ZZZ1; IF(ZRPMN.LT.0.0D0) I=2;
 ABSRPM=DABS(ZRPMN);
 IF (ABSRPM.LT.0.2791D0) I=2;
 IF (A1.LT.0.0D0) I=3;
 PMFTOR(KZ)=(E(4,I)+ABSRPM*(E(2,I)+(E(3,I)*ABSRPM)))*A1;
 CALL MOTRRO(KZ,TORQD,LOOP);
 PMFTOR(KZ)=PMFTOR(KZ)*A1;
 PMDRPM(KZ)=9.5493*(TORQD-PMTORQ(KZ)-PMFTOR(KZ))/PMINT(KZ);
 IF(ABSRPM.GT.ZZZM3) GOTO 10;
 IF(DABS(PMTORQ(KZ)).LE.E(1,I).AND.TORQD.LE.E(1,I)) PMDRPM(KZ)=ZZZ0;
 4 IF(JTIM.GT.0) GOTO 10
 WRITE(6,30)ZFLON,ZRPMN,T1,T2,PMHEAD(KZ),PMTORQ(KZ),ZPREQ(KZ);
 30 FORMAT(/,' FLON=',E12.5,' RPMN=',E12.5,' T1=',E12.5,' T2=',E12.5
        ,' PHEAD=',E12.5,'PMTORQ=',E12.5,'REQ.PRES. =',E12.5);
 IF(LOOP.LT.0) GO TO 10; ITER=ITER+1; AFC=ZZZ981*PMRHO(KZ);
 DPREQ=(ZPREQ(KZ)-PMHEAD(KZ)*AFC)/AFC;
 IF(DABS(DPREQ).LT.ZZZM4) GOTO 10;
 PMDRPM(KZ)=DPREQ*PMRR(KZ)/PMRH(KZ);
 CALL FSOV11(PMR(KZ)*PMDRPM(KZ),ITER,5E1,1E-6,20,0,'AP');
 IF (ITER.LT.20) GOTO 2; PMRO(KZ)=PMR(KZ);
 ZTPUMX(KZ)=PMT+(PMTORQ(KZ)*PMR(KZ)*0.10472D0)/(PMF*CP);
 10 IF(KZ.EQ.1) PMR(KZ)=PMRO(KZ)
  IF(KZ.EQ.2) PMR(KZ)=PMRO(KZ);
   %A >
. APUMOO;
    APUMPO=PMHEAD(KZ)*ZZZ981*PMRHO(KZ)
.  INCLUDE MOTRRO;
. 99 KSUBST=KSUBST-1; RETURN;
    END
*MOTRRO
C   PUMP MOTOR MODULE
C
    SUBROUTINE MOTRRO(KP1,ZTORD,LOOP)
    IMPLICIT REAL*8(A-H,O-Z)
. COMIN TIMER5,TERMN5,APUMP5,CONST5;
. DATA SNAME/'MOTTRO'/;KSUBST=KSUBST+1;ZSUBST(KSUBST)=SNAME;
    IF(JTIM.GT.0) GOTO 10
    PMTRIP(KP1)=ZZZ1
    PMDECA(KP1)=ZZZ1
    ZTORD=PMTORQ(KP1)+PMFTOR(KP1)
    PMTORD(KP1)=ZTORD
    PMRO(KP1)=PMR(KP1)
    GOTO 99
  10 IF(PMTRIP(KP1).GT.0.99999) GOTO 11
    ZTORD=PMTORD(KP1)*PMTRIP(KP1)
    IF(PMR(KP1).GE.PMRPSP(KP1)) GOTO 99
```

```
      ZTORD=PTORQP(KP1)*PMDECA(KP1)
      IF(PMR(KP1).GT.PMLOCR(KP1)) GOTO 99
      IF(PMLOCF(KP1).NE.0.0D0) GOTO 13
      ZTORD=PMTORQ(KP1)+PMFTOR(KP1)
      GO TO 99
   13 PMFTOR(KP1)=PMFTOR(KP1)+PMLOCF(KP1)
      GOTO 99
   11 ZTORD=PMTORD(KP1)*PMTRIP(KP1)
. 99 KSUBST=KSUBST-1; RETURN;
      END
```

Vita

George Matthews Marshall was born in Natchez, Mississippi, on March 2, 1961 to George Matthews and Mary Powell Marshall. He attended Catholic High School in Baton Rouge, Louisiana, and was graduated in 1979.

He entered Louisiana State University in 1979 and began studying electrical engineering. He received his B.S. in 1984 and entered graduate school there in the Department of Nuclear Engineering.