Measurement of the delayed neutron fraction to prompt neutron lifetime ratio in the UTR-10 reactor using digital noise analysis

by

David Edward Roth

A Thesis Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Department: Nuclear Engineering
Major: Nuclear Engineering

Iowa State University
Ames, Iowa
1989

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Digital neutron noise analysis has been used to measure the delayed neutron fraction, β, to prompt neutron lifetime, ℓ, ratio in the Iowa State University UTR-10 research reactor. The auto-power spectral densities (APSD) of the currents from an uncompensated ion chamber, a fission chamber, and a BF_3 detector were recorded and analyzed, and the coherence function of two BF_3 detectors was recorded using a Hewlett-Packard 3582A low frequency spectrum analyzer. The APSD contained information about the reactor transfer function, and were therefore affected by the delayed neutron fraction to prompt neutron lifetime ratio. The spectra were transferred from the spectrum analyzer to a microcomputer for normalization and analysis. A non-linear least squares fitting routine called MINSQ was used to fit the parameters of the theoretical APSD and coherence function to the measured spectra, and thereby estimate the $\frac{\beta}{\ell}$ ratio and its uncertainty.

The estimates of the delayed neutron fraction to prompt neutron lifetime ratio made from APSD measurements were in the range of $39.0s^{-1} \pm 3.1s^{-1}(68\%)$ to $48.5s^{-1} \pm 2.3s^{-1}(68\%)$, and depended on the type and location of the neutron detector. The coherence function measurement estimated $\frac{\beta}{\ell}$ to be $41.5s^{-1} \pm 2.3s^{-1}(68\%)$. This compared favorably with the currently accepted value of $43.3s^{-1}$ and with previous investigations using different techniques.

The measurements were performed in response to a mandated conversion of the reactor fuel from high-enrichment uranium to low-enrichment uranium. The conversion will alter the $\frac{\beta}{\ell}$ ratio. During all the measurements, there were no fuel oscillations or fuel vibrations, so the experiment may be used directly on the new core.

CHAPTER 1. INTRODUCTION

Parameters Measured and Their Uses

The goal of these measurements was to find the ratio of the delayed neutron fraction to the prompt neutron lifetime in the Iowa State University UTR-10 research reactor using digital neutron noise analysis.

The *delayed neutron fraction* β is defined as the fraction of neutrons which are not released at the time of fission but are instead released by the decay of fission products. When a fission takes place, neutrons are released into the chain reaction through two main processes: prompt neutrons are released at the instant of fission and delayed neutrons are released through the beta decay of certain fission products. The rate of release of the delayed neutrons is a function of the half-life of the fission products. The delayed release of a small fraction of neutrons (less than 1%) is crucial in reactor physics; if there were no delayed neutrons, reactor control would be very difficult.

The *prompt neutron lifetime* ℓ is the average time a prompt neutron exists in the reactor before being absorbed or leaking out of the system. For the UTR-10, the accepted value of ℓ is 150 microseconds[1].

The delayed neutron fraction is a function primarily of the fuel type. For most

applications there are six main delayed neutron precursor groups produced by fission of uranium-235. The neutrons they release represent 0.650 % of the total number of neutrons produced in thermal fission[2]. The effective delayed neutron fraction β_{eff} is a function of both fuel type and the reactor geometry. It represents the fraction of next generation fissions produced by delayed neutrons when neutron leakage (neutrons escaping from the reactor core) and non-fission neutron absorption are considered. It is generally larger than the delayed neutron fraction of the fissionable isotope because the delayed neutrons produced by precursors have lower energies than the average energy of prompt neutrons. The more energetic prompt neutrons are more likely to escape from the core than are the delayed neutrons, so delayed neutrons make up a larger percent of the neutron population than indicated by the delayed fraction of the fuel. The most recent calculation of β_{eff} for the UTR-10 yielded a value of 0.007675[3]. Throughout the rest of this report, β_{eff} will be simply written as β.

The multiplication factor k is the number of fissions caused by the fission of single fuel atom. If the multiplication factor is less than one, the reactor is subcritical and the reactor power (and neutron population) decreases in the absence of a neutron source. If the multiplication factor is greater than one, the reactor is supercritical and the reactor power increases. If the multiplication factor equals unity, then the reactor is critical and the power is constant in the absence of a neutron source.

The state of the reactor is also described by the amount of reactivity present. Absolute reactivity is defined as

$$\rho = \frac{k-1}{k} \tag{1.1}$$

and may be either positive or negative. Reactivity, as defined above, is dimensionless.

Reactivity is commonly expressed in dollars, given by ρ in Equation 1.1 divided by the delayed neutron fraction β;

$$\rho(\$) = \frac{k-1}{k\beta}. \tag{1.2}$$

The purpose of defining reactivity in dollars is to relate the multiplication factor with the delayed neutron fraction. When one dollar of reactivity is present, the reactor is prompt critical, and the chain reaction is sustained on prompt neutrons alone.

The reactor period T is the amount of time it takes for the reactor power to change by a factor of e. If the reactor is in a prompt critical state the reactor period is given by

$$T = \frac{\ell}{k-1}, \tag{1.3}$$

so power increases very quickly and the reactor is not controllable.

On a day-to-day reactor operations basis, the delayed neutron fraction is used indirectly in calculation of reactivity. The prompt neutron lifetime is used during generation of tables such as reactor reactivity from period. Reactivity values for the UTR-10 reactor are currently calculated assuming a value of 0.0065 for β. For example, if the reactor is supercritical such that $k = 1.004$, the absolute reactivity is .00398, or in dollar units $\rho = \$.613$.

The UTR-10 reactor is operated so that delayed neutrons are necessary for criticality. In the absence of delayed neutrons the reactor would be subcritical and the power level would decrease in the absence of a neutron source. Operating a delayed neutron critical reactor guarantees a slow time scale for any power change in the reactor, thereby allowing simpler control systems.

The measurements described herein were performed in response to a required change in the reactor fuel from 92% to 19.75% uranium enriched in the ^{235}U isotope. The 92% enriched in ^{235}U fuel or high-enrichment uranium (HEU) must by law be replaced with low-enrichment uranium (LEU) with less than 20% ^{235}U. The fuel conversion necessitates calculating new core parameters, including the new effective delayed neutron fraction and the prompt neutron lifetime. As suggested in the fuel conversion proposal [4], the calculated $\frac{\beta}{\ell}$ value for the HEU core must be compared with the measured value to assess the accuracy of the parameters predicted by the core physics codes.

The Reduced Enrichment for Research and Test Reactors (RERTR) group at Argonne National Laboratory, the division investigating the conversion of research and test reactors from HEU fuel to LEU fuel, performed core physics calculations for the UTR-10. In the process, the group produced estimates of β and ℓ. The values the group obtained are shown in Table 1.1. The model used at Argonne was the top half of the UTR-10 with all control rods out. This was done so that the excess reactivity (ρ_{excess}), or available reactivity beyond what is needed for criticality, could be found; the effective multiplication factor in the UTR-10 with all available excess reactivity inserted is about 1.004. The effective multiplication factors (k_{eff}) calculated at Argonne National Lab were less than unity, falsely indicating the reactor is a subcritical assembly. This error was caused by the failure of the computational model to handle the large neutron leakage of the UTR-10.

The $\frac{\beta}{\ell}$ ratio is used in transient analysis. In the case of the UTR-10 analysis, the values of β and ℓ are inputs into the computer code PARET[5]. PARET is used to

Table 1.1: Calculated ISU UTR-10 Core Parameters

Core	k_{eff}	β_{eff}	$\ell, \mu s$	$\frac{\beta}{\ell} s^{-1}$
HEU	0.994766	0.007675	159.56	48.1
LEU	0.995541	0.007631	165.06	46.5

predict the core response to reactivity and power transients. Both the delayed neutron fraction and the prompt neutron lifetime directly affect the response to a reactivity insertion. The reactivity transient responses of both the LEU and the HEU cores were based on a step insertion of all available excess reactivity. The values calculated using PARET will be presented in the results section. PARET uses the input core parameters to numerically solve the coupled neutronics and feedback equations. The neutronics equations describe the change in neutron population in the reactor as a function of time and input reactivity, while the feedback describes the change in reactivity as a function of fuel temperature, moderator density, etc.. The PARET user's guide indicates the power estimates during a transient calculated by PARET are within 25% of powers measured in experiments of the same transients.

Measurement Techniques

The $\frac{\beta}{\ell}$ ratio was estimated based on measurements using digital neutron noise analysis. Neutron noise analysis takes advantage of information carried by the fluctuating neutron population to deduce information about the reactor condition. The noise analysis measurements were done using signals from one or two detectors and the data gathered were analyzed using a digital computer.

The neutron noise was measured by recording the auto power spectral density

(APSD) of the output of a neutron detector. The APSD represents the average power per unit frequency contained in the detector signal and contains magnitude information about the reactor transfer function. The reactor transfer function, which is affected by $\frac{\beta}{\ell}$, is defined as the ratio of the Laplace transform of the output neutron population variation to the Laplace transform of the input reactivity or source disturbance. Therefore, the APSD contains information on $\frac{\beta}{\ell}$; the $\frac{\beta}{\ell}$ ratio may be estimated by measurement of the APSD of the signal from a single neutron detector.

A second method of measuring the neutron noise involved the recording of the coherence function of two detectors. The coherence function describes the correlation between two separate events, so the coherence function differentiates between neutron-induced noise and unrelated line noise produced by the detection system. The coherence function relates two APSDs to each other in such a way that it is influenced by the reactor transfer function, and therefore the $\frac{\beta}{\ell}$ ratio; the $\frac{\beta}{\ell}$ ratio may be estimated by measurement of the coherence function of the output of two neutron detectors.

The measurements took advantage of state-of-the-art digital computer technology for data collection and analysis. The detector current signal was converted to a voltage representative of the neutron population fluctuations (neutron noise). This voltage signal was transformed into a frequency spectrum by a low frequency spectrum analyzer (which directly measured both the APSD of one detector and the coherence function of two detectors) and passed the results to a microcomputer for storage and data manipulation. A non-linear least squares fitting routine was used to analyze the APSD or coherence function and find the $\frac{\beta}{\ell}$ ratio and its uncertainty. The $\frac{\beta}{\ell}$ ratio is

sometimes referred to as the break frequency because a plot of the the APSD versus frequency has a 3 dB drop in the spectral power at the $\frac{\beta}{\ell}$ frequency.

Some of the considerations in the experimental design were the following:

- No external reactivity perturbations were to be present. For example, a reactivity oscillator would not be used.
- The experiment must work for both the existing HEU and new LEU cores. For example, the neutron detector(s) must fit in the same location in both cores.
- The experiment should be relatively easy to set up and operate.

No external reactivity perturbations were allowed because the core physics calculations with which the measured values were to be compared were made using a steady state, perturbation-free model. Any experimental apparatus design must work equally well in both the new and the old cores, or else it will not be possible to measure the effect of the fuel conversion with confidence.

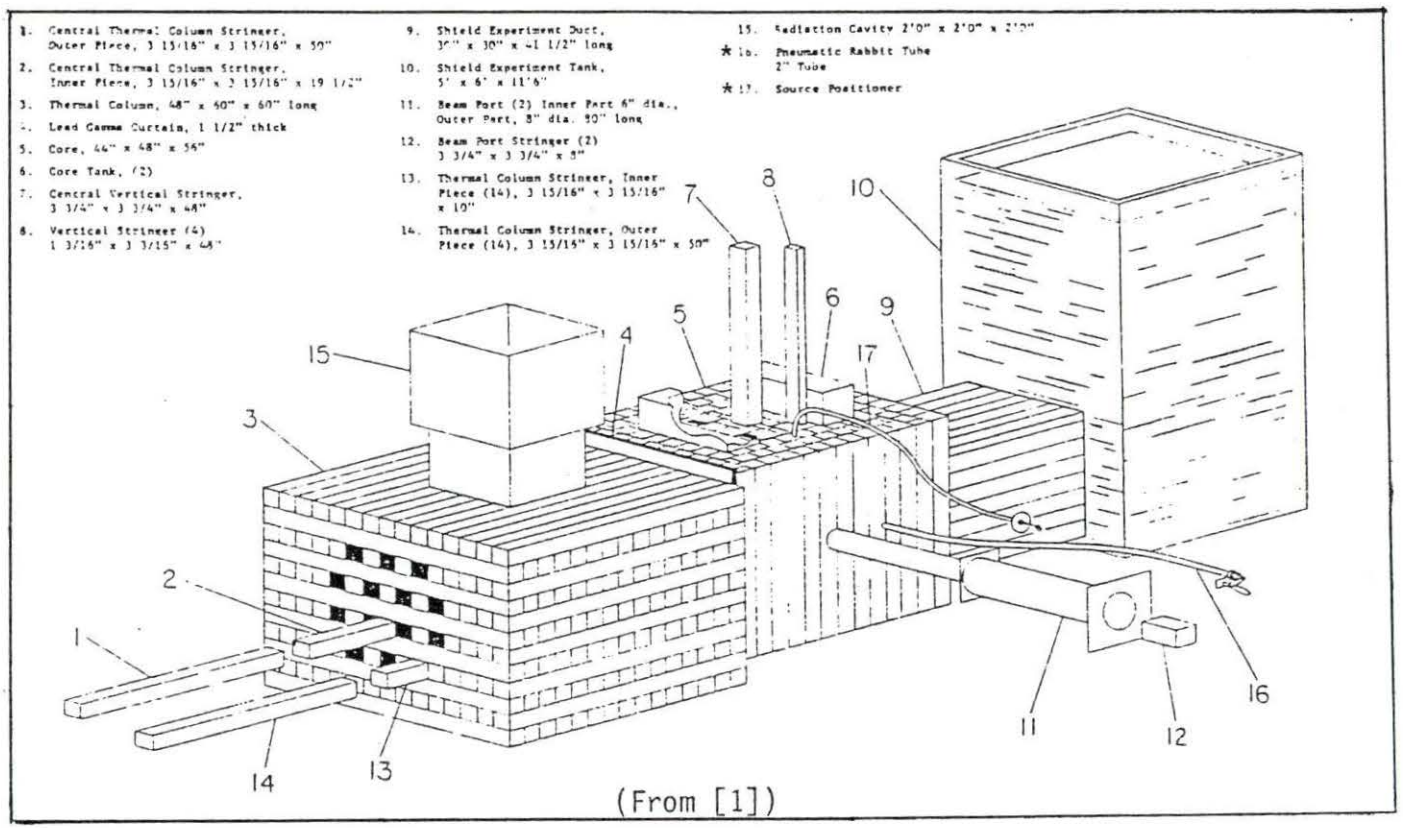
CHAPTER 2. BACKGROUND

Reactor Description

The UTR-10 research reactor is an Argonaut type reactor. It has a coupled core surrounded by a graphite reflector, as shown in Figure 2.1. The core is a 44-in. by 56-in. by 48-in. stack of graphite holding two core tanks. Each core tank is approximately 6-in. by 20-in. by 58-in. and each is divided into six fuel chambers. The HEU fuel assemblies have 12 fuel plates each, while the LEU fuel will have 24 plates per assembly. The term 'core' as used in this report refers to the two core tanks and the graphite reflector; at the center of the core is the internal graphite reflector.

The UTR-10 has six locations available for experiments: the internal reflector, the shield tank, the irradiation cavity, the thermal column, the beam ports, and a rabbit system. The internal reflector, the shield tank, and the thermal column are suitable for neutron detectors.

Reactor power level is controlled by three safety-type control blades and one regulating blade. The control blades are made of boral (a mixture of aluminum and neutron absorbing boron) and clad with aluminum. The reactor control may be manual, in which case the operator manipulates the regulating blade and the shim-safety blade to control power, or automatic. When the reactor is operated in automatic



*Items 16 and 17 should appear on the opposite side of the core.

Figure 2.1: The UTR-10 Reactor

mode, the power is held constant by the regulating blade. The regulating blade is positioned by a feedback mechanism which monitors reactor power and period. The automatic system makes several adjustments to the regulating blade each second to maintain a steady power.

Water flowing between the fuel plates serves as a neutron moderator, slowing fast neutrons to thermal energies through neutron scattering, and as a coolant. The thermal neutrons are much more likely to cause a fission than are fast neutrons. The graphite reflector surrounding the core tanks reflects some neutrons back into the core tanks, where they may cause additional fissions.

Heat produced by fission is removed by the moderating water (primary coolant) flowing up from the bottom of each core tank at a nominal rate of 8.5 gallons per minute. The primary coolant is cooled by City of Ames water (secondary coolant) through a single loop heat exchanger. Since moderator temperature affects reactivity and, therefore, reactor power, the inlet temperature of the primary coolant is regulated. The water is nominally 80^{o}F, but drifts between 78^{o}F and 82^{o}F are possible. The temperature variations result in power drifts unless compensated by insertion or removal of a control blade.

The UTR-10 has a large neutron leakage and several regions of high neutron absorption. The prompt neutron lifetime is shorter for a neutron traveling near an absorber (such as a control blade) or near a non-reflective surface (such as the top of the core) than for a neutron near the center of the core. Absorption and leakage affect the effective delayed neutron fraction and the prompt neutron lifetime. To minimize the influence of leakage and absorption on the $\frac{\beta}{\ell}$ measurement, it is necessary to

position detectors away from any absorbers or boundaries. Ideally, the best place for a detector to measure the neutron lifetime in the UTR-10 is in the core center.

If any neutron absorbing material (poison) is introduced into the core, or if the core leakage rate is increased, ℓ decreases so $\frac{\beta}{\ell}$ increases. Unfortunately, when a detector is present in the core, it introduces poison by increasing parasitic absorption and increases leakage by displacing graphite. Thus, the measured APSD might not produce the clean core delayed neutron fraction to prompt neutron lifetime ratio. It is desirable to use small detectors which displace as little graphite as possible and absorb as few neutrons as possible while still providing a statistically strong signal.

Previous Work at ISU

There are at least two previous investigations of the delayed neutron fraction to prompt neutron lifetime ratio on the UTR-10 reactor. Both were performed in the early 1970s.

Chan [6] used a cross-correlation technique and measured a value of $43.0 s^{-1}$ for $\frac{\beta}{\ell}$. During his measurements, the reactor was perturbed by a small oscillating poison placed between the two core tanks.

Chan's experiment was not repeated because the oscillator in the core was not desirable. Also the analysis of data was performed on an analog as well as a digital computer, and involved manually varying an electronic filter window and repeatedly playing magnetic tape of a signal representative of the neutron noise. The manual operation of the filter introduces potential systematic experimental error.

Nabavian [7] performed measurements of the $\frac{\beta}{\ell}$ ratio using the polarity correla-

tion method. He found the ratio to be 44.4 s^{-1}. His measurement method involved observing the signals from two detectors for one hour, amplifying the signals with an analog computer, and recording the signals on magnetic tape. The tape was then replayed and the signals sent to a polarity correlator. Information from the correlator was used to produce the detector coherence function, and the break frequency was estimated from the coherence function plot by fitting the coherence function equation to the observed data.

Nabavian's experiment also produced valid results. It was not repeated because it required a substantial effort to record the signal and play it back for analysis.

The analysis methods of both Chan and Nabavian were made obsolete by invention of the low frequency spectrum analyzer, which will be described later.

Other Uses of Neutron Noise Analysis

The ratio of the delayed neutron fraction to the prompt neutron lifetime has also been measured at the EWA reactor in Poland [8]. The method used was measurement of power spectral densities using self-powered neutron detectors and fission chambers. The spectrum of the fission chamber contained no frequency dependent information (the spectrum was white) due to the distance from the core to the detector, while the self-powered neutron detector located closer to the core provided an estimate of the delayed neutron fraction to the prompt neutron lifetime of $14.1s^{-1}$, far from the expected $150s^{-1}$. The discrepancy was caused by the detector being surrounded by a beryllium reflector which greatly increased the neutron lifetime. The spectral densities measured showed many small peaks on top of the expected response, but

only one was attributed to the 50 Hz electrical net.

Neutron noise has many uses other than simple measurements of $\frac{\beta}{\ell}$. For example, engineers at Duke Power are using neutron noise in combination with small fluctuations of coolant temperature to measure the moderator temperature coefficient in their Oconee reactors [9]. The results as of this date indicate neutron noise may be used, but the measured temperature coefficient is local instead of global.

Neutron noise may also be used to detect core barrel motion, fuel assembly vibration, control rod vibration, and moderator flow and moderator phase [10][11]. It can also be used to estimate the reactivity present in a subcritical reactor.

CHAPTER 3. THEORY

Neutron noise measurements require the conversion of minor variations of the neutron population into a frequency spectrum, and subsequently into power spectral densities or coherence function spectra. This is done analytically by transforming the point kinetics equations (described below) from the time domain to the frequency domain.

The time dependent neutron population in the UTR-10 may be described by point kinetics equations. The model is simple; it ignores spatial effects and reactivity feedback.

Derivation of the Reactor Transfer Function

A nuclear reactor can be described by a zero dimensional point kinetics model. The model uses coupled differential equations to describe the changes in neutron density, delayed neutron precursor density, and reactivity with respect to time. The point kinetics equations are written

$$\frac{dn}{dt} = (1-\beta)k\frac{n}{\ell} + \sum_{i=1}^{6} \lambda_i c_i + q - \frac{n}{\ell} \tag{3.1}$$

$$\frac{dc_i}{dt} = \frac{\beta_i}{\ell} kn - \lambda_i c_i \tag{3.2}$$

where:

$$\begin{aligned} n &= \textit{neutron density} \\ c_i &= \textit{delayed neutron precursor density of } i^{th} \textit{ group} \\ t &= \textit{time} \\ \beta_i &= \textit{delayed neutron fraction of } i^{th} \textit{ group} \\ \lambda_i &= \textit{decay constant for precursor decay of } i^{th} \textit{ group} \\ q &= \textit{source strength} \\ k &= \textit{multiplication factor} \\ \ell &= \textit{prompt neutron lifetime.} \end{aligned} \tag{3.3}$$

The delayed neutron precursors produced by fission of uranium 235 are usually classified into six delayed groups, as used above, and each group has a different decay constant and delayed neutron fraction. These equations can be simplified by defining the neutron generation time, Λ, and reactivity, ρ, as

$$\Lambda = \frac{\ell}{k} \tag{3.4}$$

$$\rho = \frac{k-1}{k}, \tag{3.5}$$

by using a single delayed neutron precursor group with a weighted average decay constant of

$$\lambda = \left(\frac{1}{\beta} \sum_{i=1}^{6} \frac{\beta_i}{\lambda_i} \right)^{-1}, \tag{3.6}$$

and a single effective delayed neutron fraction of

$$\beta = \sum_{i=1}^{6} \beta_i. \tag{3.7}$$

Now the point kinetics equations are written as:

$$\frac{dn}{dt} = \frac{\rho - \beta}{\Lambda} n + \lambda c + q. \tag{3.8}$$

$$\frac{dc}{dt} = \frac{\beta}{\Lambda} n - \lambda c. \tag{3.9}$$

Equations 3.8 and 3.9 are commonly referred to as the one-group point kinetics equations. They describe changes to the neutron population (or fission rate or power when multiplied by the appropriate constants) with respect to time. When the reactor is in a steady state condition (either subcritical or source-free critical), both Equation 3.8 and Equation 3.9 are equal to zero.

The neutron population is never truly constant. Fission is a statistical process, so the steady state population is always fluctuating about some average value. This fact can be incorporated into the point kinetics equations by assuming the neutron population and the delayed neutron precursor population fluctuate about an average value in response to a minor fluctuation of input reactivity or neutron source. When the input is considered to be a minor variation in reactivity, this is written

$$n = n_O + \delta n \tag{3.10}$$

$$c = c_O + \delta c \tag{3.11}$$

$$\rho = \rho_O + \delta \rho \tag{3.12}$$

$$q = q_O \tag{3.13}$$

where the x_O terms are the steady state values and the δx terms represent small fluctuations in the steady state values.

Substituting these in Equations 3.8 and 3.9 and canceling the steady state terms

yields

$$\frac{d\delta n}{dt} = \frac{\delta\rho n_O}{\Lambda} + \frac{\delta\rho\delta n}{\Lambda} + \frac{\rho_O - \beta}{\Lambda}\delta n + \lambda\delta c \tag{3.14}$$

$$\frac{d\delta c}{dt} = \frac{\beta}{\Lambda}\delta n - \lambda\delta c. \tag{3.15}$$

Since the fluctuations are small, the $\frac{\delta\rho\delta n}{\Lambda}$ term can be neglected, leaving

$$\frac{d\delta n}{dt} = \frac{\delta\rho n_O}{\Lambda} + \frac{\rho_O - \beta}{\Lambda}\delta n + \lambda\delta c \tag{3.16}$$

$$\frac{d\delta c}{dt} = \frac{\beta}{\Lambda}\delta n - \lambda\delta c. \tag{3.17}$$

Equations 3.16 and 3.17 describe changes in small fluctuations of the neutron population and delayed neutron precursor population with respect to time.

The two coupled differential Equations 3.16 and 3.17 may be solved using the Laplace transform. Recall the Laplace transform $F(s)$ of some function $f(t)$ is defined as

$$F(s) = \int_0^\infty f(t)e^{-st}dt \tag{3.18}$$

where

$$s = j\omega \tag{3.19}$$

$$j = \sqrt{-1} \tag{3.20}$$

$$\omega = frequency\ in\ radians\ per\ second. \tag{3.21}$$

The Laplace transforms of Equations 3.16 and 3.17 are

$$s\delta N(s) - \delta n(0) = \frac{n_O}{\Lambda}\delta R(s) + \frac{\rho_O - \beta}{\Lambda}\delta N(s) + \delta C(s)\lambda \tag{3.22}$$

$$s\delta C(s) - \delta c(0) = \frac{\beta}{\Lambda}\delta N(s) - \delta C(s)\lambda \tag{3.23}$$

where

$$\delta R(s) = \int_0^\infty \delta\rho(t)e^{-st}dt. \tag{3.24}$$

The average value of the fluctuations of the neutron population and the delayed neutron precursor population is zero. Both $\delta n(0)$ and $\delta c(0)$ are therefore taken to be zero. Solving for $\delta N(s)$ yields

$$\delta N(s) = \frac{n_O \delta R(s)}{\Lambda s + \beta - \rho_O - \frac{\lambda\beta}{s+\lambda}}. \tag{3.25}$$

Equation 3.25 describes fluctuations in the neutron population caused by fluctuations of reactivity.

A transfer function is the ratio of the Laplace transform of an output signal to the Laplace transform of an input. The reactivity transfer function assumes the reactivity, $\delta R(s)$, to be the input and the neutron fluctuations, $\delta N(s)$, to be the output. Therefore, the reactivity transfer function is given by

$$H_{\delta\rho+\rho_O}(s) = \frac{\delta N(s)}{\delta R(s)} = \frac{n_O}{\Lambda s + \beta - \rho_O - \frac{\lambda\beta}{s+\lambda}} \tag{3.26}$$

which can be re-arranged as

$$H_{\delta\rho+\rho_O}(s) = \frac{n_O(s+\lambda)}{\Lambda s(s + \frac{\beta-\rho_O}{\Lambda} + \lambda - \frac{\rho_O\lambda}{s\Lambda})}. \tag{3.27}$$

When a reactor is critical, $\rho_O = 0$ and $\Lambda = \ell$, so the transfer function becomes

$$H_{\delta\rho}(s) = \frac{n_O(s+\lambda)}{\ell s(s + \frac{\beta}{\ell} + \lambda)}. \tag{3.28}$$

Since $\lambda \ll |s|$ for frequencies of interest[1] and $\lambda \ll \frac{\beta}{\ell}$, the transfer function can be

[1] $\lambda = 0.0767 s^{-1}$ for thermal fission of U-235[2].

approximated as

$$H_{\delta\rho}(s) \approx \frac{n_O}{\ell}\frac{1}{s+\frac{\beta}{\ell}}. \tag{3.29}$$

Ignoring the delayed neutron precursor decay constant is sometimes referred to as neglecting delayed neutron effects.

The magnitude of a Laplace transformed equation is given by

$$|H(s)|^2 = H(s)H^*(s) \tag{3.30}$$

where $H^*(s)$ is the complex conjugate of $H(s)$. The square root of the magnitude is the amplitude. The magnitude contains no imaginary component, so it may be written in the frequency domain as $|H(\omega)|^2$. The magnitude of the reactivity transfer function is therefore

$$|H_{\delta\rho}(\omega)|^2 = (\frac{n_O}{\ell})^2\frac{1}{\omega^2+(\frac{\beta}{\ell})^2}. \tag{3.31}$$

$|H_{\delta\rho}(\omega)|^2$ is commonly called the zero-power reactivity transfer function. Zero-power means there are no reactivity feedback terms in the derivation. Reactivity feedback would appear as an additional input into the system of equations, and would be caused by reactivity variations induced by a large change in the neutron population. The only source of noise in the system is the input reactivity fluctuations.

By considering small changes in the source term instead of the reactivity term, the zero-power source transfer function may be found. It differs from Equation 3.29 by a factor of $\frac{\ell}{n_O}$, and its magnitude may be written as

$$|H_{\delta q}(\omega)|^2 = \frac{1}{\omega^2+(\frac{\beta}{\ell})^2} \tag{3.32}$$

Equation 3.32 describes the magnitude of the frequency response of the reactor to a disturbance in source strength. It is used only for modeling a subcritical assembly,

since the presence of a source in a critical assembly causes an increase in neutron population.

The UTR-10 is a coupled core reactor, meaning there are two core tanks which are subcritical alone but are capable of supercriticality together. It may seem logical that two point reactor models are necessary to describe the UTR-10. Chan[6] showed that the coupled core configuration will not affect the reactor transfer function. The core may be modeled by a single set of point kinetics equations, even though it is made of two physically separated core tanks.

Neutron Noise Analysis

Noise analysis is based on the conversion of signals in the time domain to results in the frequency domain. In the reactor system, the signals of interest are the input source or reactivity fluctuations and the measured detector output. Cohn [12] showed minor variations of reactivity and variations of source strength are independent of frequency and therefore may be considered frequency independent or white noise. When the reactivity input is white noise, then the frequency dependence of output signal is a function of the reactor transfer function, and the detection system and instrumentation transfer functions only.

The common method used to convert a time domain signal to the frequency domain is to use a Fourier transform to convert the signal to a Fourier series approximation. A complete discussion may be found in Churchill and Brown[13] or Conte and de Boor[14]. If a function is converted to the frequency domain using a Fourier transform, the series of squares of the magnitudes of the Fourier coefficients is the

APSD.

The discussion of power spectra which follows will use the notation of Uhrig[15].

The autocorrelation function $\phi_{xx}(\tau)$ describes how a function at time t is related to itself at time $t + \tau$. The APSD $\Phi_{xx}(\omega)$ is the Fourier transform of the autocorrelation function $\phi_{xx}(\tau)$. This is shown as

$$x(t) = \textit{time domain signal} \tag{3.33}$$

$$\phi_{xx}(\tau) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t)x(t+\tau)dt \tag{3.34}$$

$$\Phi_{xx}(\omega) = \int_{-\infty}^{\infty} \phi_{xx}(\tau)e^{-j\omega\tau}d\tau. \tag{3.35}$$

The measured APSD of a system is a function of the input to the system and the system transfer function. Consider a linear system with time domain input $x(t)$, impulse response function $h(t)$ and time domain output $y(t)$. Its input, transfer function, and output may be written in the frequency domain as $\Phi_{xx}(\omega)$, $H(\omega)$, and $\Phi_{yy}(\omega)$, respectively. They are related to each other such that

$$\Phi_{yy}(\omega) = |H(\omega)|^2\Phi_{xx}(\omega). \tag{3.36}$$

The relationship is shown symbolically in Figure 3.1 in both the time and frequency domains.

If the input signal $x(t)$ is white noise, then the input power spectral density $\Phi_{xx}(\omega)$ is independent of frequency and therefore may be written as a constant K. The output power spectral density is then

$$\Phi_{yy}(\omega) = K|H(\omega)|^2. \tag{3.37}$$

For a simple input-output system with white noise input, the APSD is a constant times the magnitude of the system transfer function.

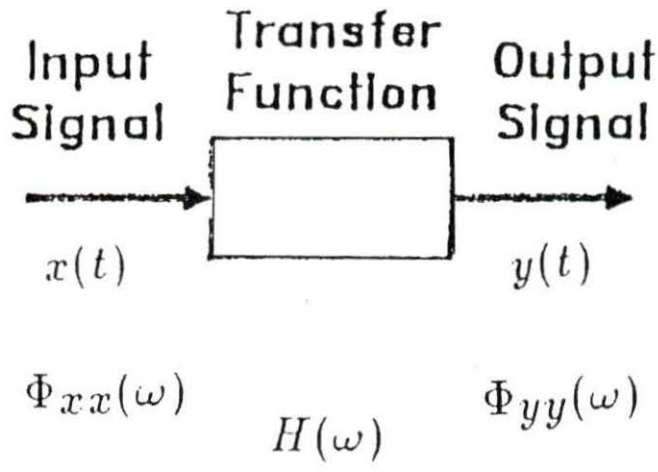


Figure 3.1: Linear Signal Transformation

The $\Phi(\omega)$ are two-sided spectral densities, meaning they are valid for both positive and negative frequencies. Negative frequencies are physically meaningless so one-sided spectral densities $G_{xx}(\omega)$ are used instead. The one-sided power spectral densities are related to two-sided spectral densities by:

$$G_{xx}(\omega) = 2\Phi_{xx}(\omega) \quad (0 < \omega \leq \infty) \tag{3.38}$$

$$G_{xx}(0) = \Phi_{xx}(0) \tag{3.39}$$

$$G_{xx}(\omega) = 0 \qquad (-\infty \leq \omega < 0). \tag{3.40}$$

They may be derived by integrating the autocorrelation function from 0 to ∞ instead of $-\infty$ to ∞.

Reactor Noise Model

A reactor detection system is not a single noise input system. There are many possible sources of noise, such as, the variations of coolant temperature, the automatic power controller variation, and the detection system itself. Only the noise produced by reactivity fluctuations and the steady state noise produced by the statistical nature of the neutron detector were considered in the following development.

Consider a single detector monitoring a reactor, symbolically modeled by Figure 3.2. Both the noise produced by the steady state population and by the fluctuations are shown. The input signal is denoted by $\delta\rho(t)$ with a one-sided APSD of $G_{\rho\rho}(\omega)$ and the output signal from channel A is $i(t)$ with APSD of $G_{ii}(\omega)$. Physically, the input is a small change in reactivity and the output is the measured detector signal for $i(t)$ or the measured power spectrum $G_{ii}(\omega)$. The conversion transfer function

$H_{con}(\omega)$ is described below. $G_{ss}(\omega)$ represents the noise produced by the neutron detector responding to the steady state neutron population n_O, and $H_{inst}(\omega)$ is the transfer function of the instrumentation used to measure the detector signal.

The random input disturbance $G_{\rho\rho}(\omega)$ may also be considered independent of frequency. Cohn [12] showed the small reactivity variation in a reactor is independent of frequency and is given by

$$G_{\rho\rho} = \frac{2\ell}{n_O}\left(\frac{\bar{\nu^2} - \bar{\nu}}{\bar{\nu}}\right), \tag{3.41}$$

where $\bar{\nu}$ is the mean number of neutrons released per fission. This may be simplified using Diven's parameter D_ν[2], which is defined as

$$D_\nu = \frac{\bar{\nu^2} - \bar{\nu}}{\bar{\nu}^2}, \tag{3.42}$$

to write

$$G_{\rho\rho} = \frac{2\ell\bar{\nu}D_\nu}{n_O} \tag{3.43}$$

The neutron population cannot be directly measured. Instead, a neutron detector current proportional to the neutron population is observed. The average current produced by a neutron population of n_O with prompt neutron lifetime ℓ is

$$\bar{i} = n_O\frac{\epsilon Q}{\ell} \tag{3.44}$$

where Q is the charge transferred per neutron absorbed and ϵ is detector efficiency. Efficiency in this context is the fraction of neutrons collected by the detector relative to the total number of neutrons absorbed in (and escaping from) the reactor in any manner[12]. The conversion from a neutron population n to a current i can

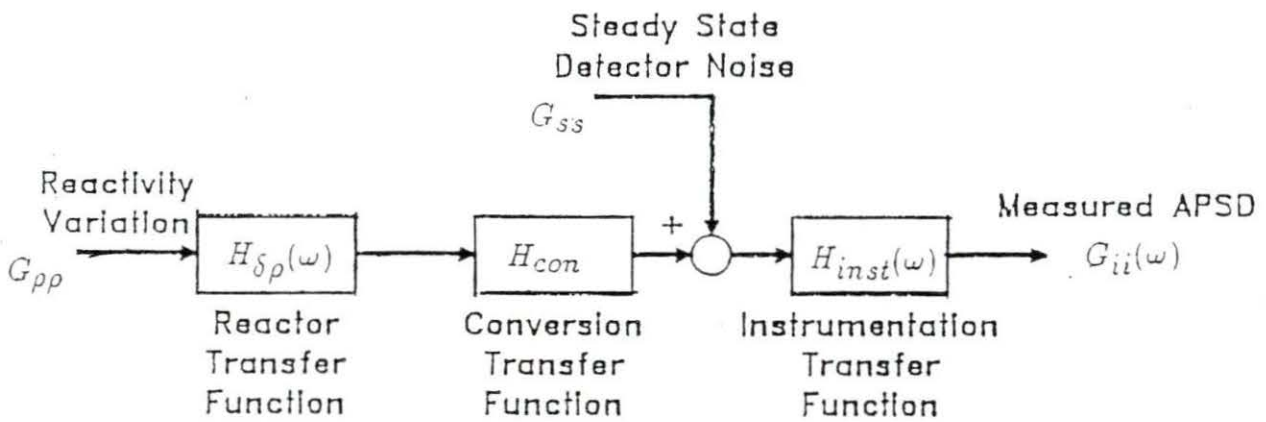


Figure 3.2: Single Detector Reactor Model

be thought of as a frequency independent transfer function H_{con}. The conversion transfer function is

$$H_{con} = \frac{\epsilon Q}{\ell}. \tag{3.45}$$

The conversion of the steady state neutron population will produce a noise component. According to Cohn, steady state current produced by the neutron detector will produce a noise spectrum $G_{ss}(\omega)$ given by

$$G_{ss}(\omega) = 2Q\bar{i}. \tag{3.46}$$

The steady state noise spectrum of the measured current is considered independent of the fluctuation produced (δn) noise and is frequency independent. From now on, the independent variable ω will not be written in the G_{ss} term.

The detection instrumentation is another possible source of frequency dependent noise or frequency dependent transfer functions. The instrumentation transfer function $H_{inst}(\omega)$ represents the effects of current amplifiers, filters, etc., on the signal coming from the detector. It is usually not possible to get a simple formula which explicitly describes the instrumentation transfer function.

The output APSD measured from the neutron detector signal is

$$G_{ii}(\omega) = G_{ss}|H_{inst}(\omega)|^2 + G_{\rho\rho}|H_{inst}(\omega)|^2|H_{con}|^2|H_{\delta\rho}(\omega)|^2. \tag{3.47}$$

The measured APSD can be corrected for the influence of the detection system by writing the corrected APSD $G'_{ii}(\omega)$ as

$$G'_{ii}(\omega) = \frac{G_{ii}(\omega)}{|H_{inst}(\omega)|^2}. \tag{3.48}$$

Applying this to Equation 3.47 produces

$$G'_{ii}(\omega) = G_{ss} + G_{\rho\rho}|H_{con}|^2|H_{\delta\rho}(\omega)|^2. \tag{3.49}$$

The reactivity input, detector noise input, and reactivity transfer function equations may be substituted into Equation 3.49 giving

$$G'_{ii}(\omega) = \frac{2Q^2\epsilon n_O}{\ell} + \frac{n_O Q^2 \epsilon^2}{\ell^3}\frac{\bar{\nu^2} - \bar{\nu}}{\bar{\nu}}\frac{1}{\omega^2 + (\frac{\beta}{\ell})^2} \tag{3.50}$$

which may be simplified to

$$G'_{ii}(\omega) = 2Q\bar{i} + D_\nu \frac{\bar{\nu i}^2}{n_O \ell}\frac{1}{\omega^2 + (\frac{\beta}{\ell})^2}. \tag{3.51}$$

The corrected APSD given by Equation 3.50 varies directly with neutron population n_O. In addition, the second term varies with the square of detector efficiency while the first term is directly proportional to efficiency. If a low efficiency detector is used, the second term, and therefore the $\frac{\beta}{\ell}$, would be obscured by the steady state noise. Equation 3.51 may be written using constants A and B in place of frequency independent terms as

$$G'_{ii}(\omega) = \frac{A}{\omega^2 + (\frac{\beta}{\ell})^2} + B. \tag{3.52}$$

To find the $\frac{\beta}{\ell}$ ratio, Equation 3.52 is fit to measured APSD data. The constants A and B do not provide information on β, and calculation of ℓ based only on A and B would require difficult-to-obtain exact values for detector efficiency, charge transferred per neutron absorbed and neutron population. The delayed neutron fraction cannot be separated from the prompt neutron lifetime without creating an infinite set of solutions for Equation 3.52.

The delayed neutron fraction to prompt neutron lifetime ratio is also called the break frequency. This is because the spectrum produced by Equation 3.52 'breaks off' beyond the $\frac{\beta}{\ell}$ frequency in a manner similar to a simple low pass (RC) filter. On a plot of Equation 3.52, the break frequency is defined as the frequency where the slope of the lines asymptotic to the plot changes from 0 decibels (dB) per decade to -20 dB per decade[16]. One decibel is twenty times the common logarithm of the ratio of two numbers; decibels are always relative units.

Coherence Function

Determination of the corrected APSD requires knowledge of the detection system transfer function. This information is not easy to obtain. As will be shown, the detection system transfer function does not appear in the coherence function of two similar detectors.

The coherence function $\gamma^2(\omega)$ describes how two signals are related to each other. It is the ratio of the square of the cross-power spectral density (CPSD) to the APSD of each measured signal. For two arbitrary signals, $x(t)$ and $y(t)$, the CPSD is given by

$$\Phi_{xy}(\omega) = \int_{-\infty}^{\infty} \phi_{xy}(\tau) e^{(-j\omega\tau)} d\tau \tag{3.53}$$

where the cross-correlation function $\phi_{xy}(\tau)$ is defined as

$$\phi_{xy}(\tau) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t) y(t+\tau) dt. \tag{3.54}$$

The cross-correlation function describes how two functions are related to each other when one function is lagging the other by τ. The APSD $\Phi_{xx}(\omega)$ is given by Equation

3.35. The coherence function[2] is defined by

$$\gamma^2(\omega) = \frac{|\Phi_{xy}(\omega)|^2}{|\Phi_{xx}(\omega)||\Phi_{yy}(\omega)|} \tag{3.55}$$

The CPSD and the coherence function are derived in a similar manner for single-sided $G_{xy}(\omega)$ terms, but are valid only for non-negative frequencies.

Consider the two-detector system shown in Figure 3.3. The reactivity perturbation input into the system $\delta\rho(t)$ or $G_{\rho\rho}$ is transformed by the reactor transfer function $H_{\delta\rho}(\omega)$ to a neutron population fluctuation signal $\delta n(t)$ or $G_{nn}(\omega)$, which is detected by two separate detection systems A and B. The channel A signal is transformed by the conversion transfer function H_{con_a} during conversion from a neutron variation to a detector current variation. The steady state detector noise G_{ss_a} adds white noise to the current. The current is then modified by the instrumentation transfer function $H_{inst_a}(\omega)$. The measured detector current is $i_a(t)$ with an APSD of $G_{ii_a}(\omega)$. Channel B is similar to channel A and has conversion transfer functions H_{con_b}, instrumentation transfer function $H_{inst_b}(\omega)$, steady state noise G_{ss_b}, and output $G_{ii_b}(\omega)$.

The APSD of the measured currents from the detectors are

$$G_{ii_a}(\omega) = |H_{inst_a}(\omega)|^2 \left[G_{\rho\rho}|H_{\delta\rho}(\omega)|^2|H_{con_a}|^2 + G_{ss_a}\right] \tag{3.56}$$

$$G_{ii_b}(\omega) = |H_{inst_b}(\omega)|^2 \left[G_{\rho\rho}|H_{\delta\rho}(\omega)|^2|H_{con_b}|^2 + G_{ss_b}\right]. \tag{3.57}$$

The CPSD of the two detection channels A and B may be derived from Equation 3.53 to be

$$G_{ab}(\omega) = H^*_{inst_a}(\omega)H^*_{con_a}H^*_{\delta\rho}(\omega)H_{inst_b}(\omega)H_{con_b}H_{\delta\rho}(\omega)G_{\rho\rho}, \tag{3.58}$$

[2]Some authors call the square root of γ^2 the coherence function.

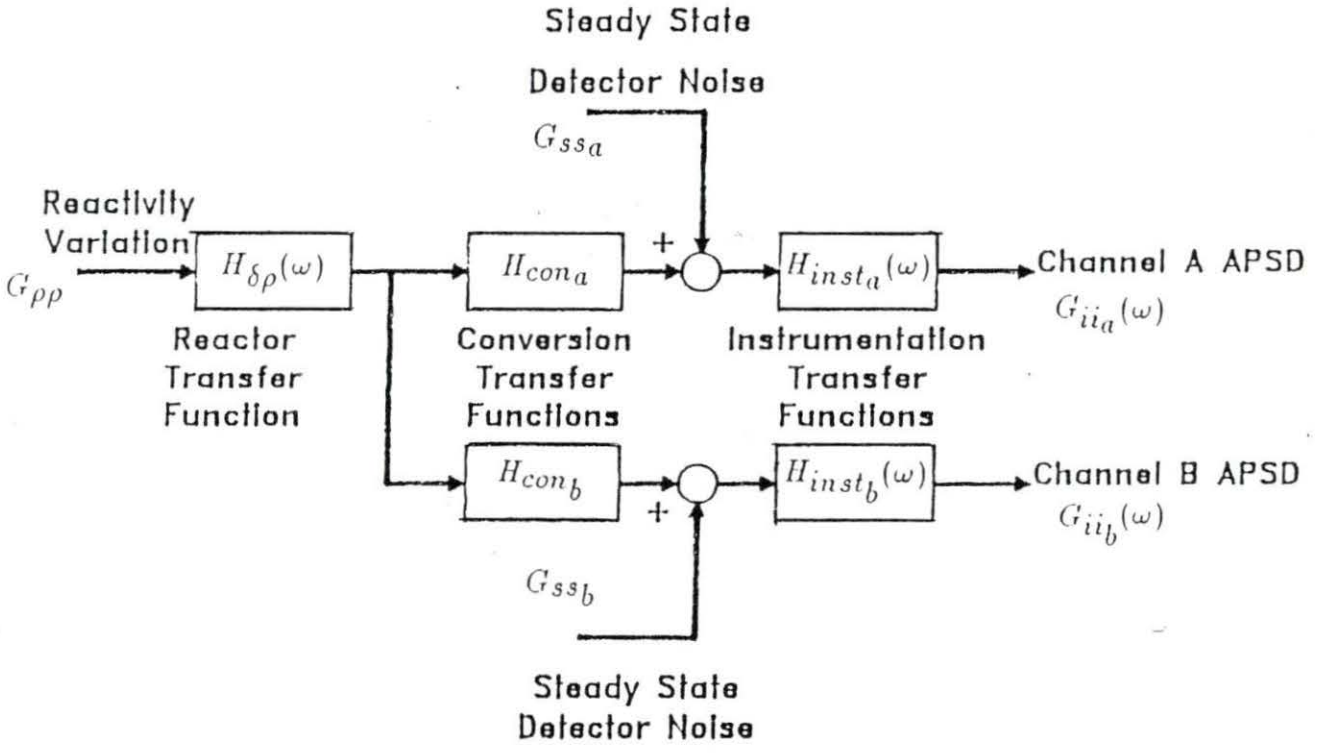


Figure 3.3: Two Detector Reactor System

which can be written as

$$G_{ab}(\omega) = H^*_{inst_a}(\omega) H^*_{con_a} H_{inst_b}(\omega) H_{con_b} |H_{\delta\rho}(\omega)|^2 G_{\rho\rho}. \tag{3.59}$$

The CPSD of two neutron detectors contains no steady state noise terms.

If the detectors and detection systems have conversion transfer function and instrumentation transfer function, the CPSD simplifies to

$$G_{ab}(\omega) = |H_{inst_a}(\omega)|^2 |H_{con_a}|^2 |H_{\delta\rho}(\omega)|^2 G_{\rho\rho}. \tag{3.60}$$

Combining the one-sided version of Equation 3.55 with Equations 3.56, 3.57, and 3.59 yields

$$\begin{aligned} \gamma^2(\omega) &= \frac{|H^*_{inst_a}(\omega) H^*_{con_a} H_{inst_b}(\omega) H_{con_b} |H_{\delta\rho}(\omega)|^2 G_{\rho\rho}|^2}{|H_{inst_a}(\omega)|^2 \left[G_{\rho\rho} |H_{\delta\rho}(\omega)|^2 |H_{con_a}|^2 + G_{ss_a}\right]} \\ &\times \frac{1}{|H_{inst_b}(\omega)|^2 \left[G_{\rho\rho} |H_{\delta\rho}(\omega)|^2 |H_{con_b}|^2 + G_{ss_b}\right]} \end{aligned} \tag{3.61}$$

which may be re-arranged into

$$\gamma^2(\omega) = \left(\left[1 + \frac{G_{ss_a}}{G_{\rho\rho} |H_{\delta\rho}(\omega)|^2 |H_{con_a}|^2} \right] \left[1 + \frac{G_{ss_b}}{G_{\rho\rho} |H_{\delta\rho}(\omega)|^2 |H_{con_b}|^2} \right] \right)^{-1}. \tag{3.62}$$

The instrumentation transfer functions do not appear in the coherence function of two detectors.

If the detector efficiencies and charge produced per neutron are assumed the same for both channel A and channel B, the coherence function becomes

$$\gamma^2(\omega) = \left(1 + \frac{G_{ss}}{G_{\rho\rho} |H_{\delta\rho}(\omega)|^2 |H_{con}|^2} \right)^{-2}. \tag{3.63}$$

Substituting the expressions for the reactivity input, reactivity transfer function, conversion transfer function, and steady state noise equations allow Equation 3.63 to be written as

$$\gamma^2(\omega) = \left(\frac{D_\nu \frac{\bar{\nu}\epsilon}{\ell^2}}{\omega^2 + (\frac{\beta}{\ell})^2 + D_\nu \frac{\bar{\nu}\epsilon}{\ell^2}} \right)^2 . \tag{3.64}$$

Let

$$F = D_\nu \frac{\bar{\nu}\epsilon}{\ell^2}, \tag{3.65}$$

and Equation 3.64 may be written

$$\gamma^2(\omega) = \left(\frac{F}{\omega^2 + (\frac{\beta}{\ell})^2 + F} \right)^2 \tag{3.66}$$

Values of F and $\frac{\beta}{\ell}$ can be estimated by fitting Equation 3.66 to measured coherence function data.

CHAPTER 4. NOISE MEASUREMENTS

Overview of Measurements

The measurements made for this study may be divided into two categories: the one-detector APSD measurements and two-detector coherence function measurements. The one-detector measurements may be further subdivided into categories by both location and type of detector. The current produced in the detectors provided the signals during all measurements.

Neutron detectors were used to measure the neutron flux, and therefore, neutron noise. The detectors measured the flux by collecting ions created by the interaction of a neutron with a fill gas, or the detector wall, rather than directly counting neutrons. The ions created were accelerated by an electric field to the center wire and the walls of the detector where they were collected. The charge difference between the walls and the center wire created a current which was measured. Both an uncompensated ion chamber (UIC) and a fission chamber (FC) were used during collection of APSD data, and two boron-trifluoride (BF_3) detectors were used to measure the coherence function.

The location of the detector for power spectra measurements was chosen based on two criteria: experimental facilities on the UTR-10 large enough for the detectors,

and proximity to the core. The internal reflector, the thermal column, and the shield tank are all large enough for the uncompensated ion chamber. The shield tank is neutronically connected to the core by a duct of graphite 30-in. square by 41-in. long. A neutron which travels along the graphite duct suffers many collisions causing it to lose any information about the fission process. The internal reflector, which has the highest neutron flux, should have neutrons directly affected by the fission process. The internal reflector was therefore used during measurements of $\frac{\beta}{\ell}$, and the shield tank was used to test the frequency response of the equipment to uncorrelated (white) neutron noise. The thermal column was not used.

The uncompensated ion chamber used in this study was held in a water-proof aluminum case suspended under water by the shield tank detector positioner at 100 east/west, 377 up/down, and in contact with the south wall.

The BF_3 detectors were *inside* the 24-in. graphite stringer, which was placed on top of the 20-in. graphite stringer. These locations are all shown on Figure 4.1, and will be referred to by number throughout the rest of this study as follows:

1. On top of a 20-in. graphite stringer in the internal reflector,
2. On top of two stacked 20 and 24-in. graphite stringers in the internal reflector,
3. In the shield tank,
4. Inside the 24-in. stringer on the 20-in. stringer.

The steps in converting a signal from a neutron detector into an estimated $\frac{\beta}{\ell}$ are shown in Figure 4.2, and described below. The input into the reactor was the inherent minor variation in reactivity which perturbed the flux. The perturbation was

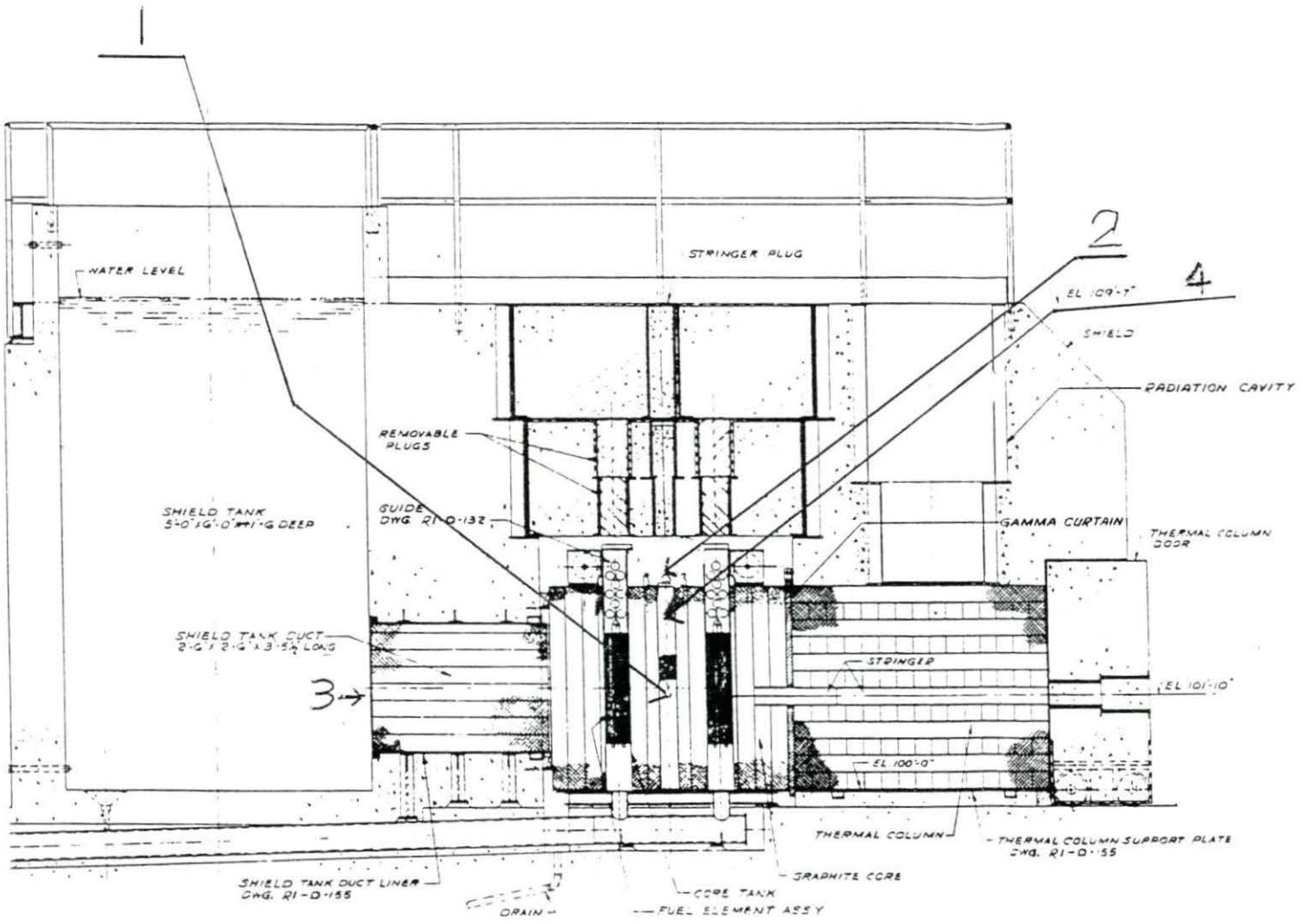


Figure 4.1: Detector Locations During Data Collections

evident in the measured detector current. The current from the detector was sensed by the Keithley 427 current amplifier, which produced a voltage signal proportional the input current. The user selected the gain. The steady state DC voltage was suppressed using a feature of the current amplifier, so the voltage signal represented the current fluctuations times the amplification setting. The DC voltage output from the current amplifier was monitored on a digital multimeter to assure proper suppression, while the AC voltage output was monitored on an oscilloscope to verify proper amplification. The voltage signal was passed through a pair of Krohn-Hite 3321 filters which attenuated frequencies above and below the range of interest. The filtered signal then entered the Hewlett-Packard 3582A low-frequency spectrum analyzer. The spectrum analyzer measured the power spectral density from a single detector, or both the power spectral densities and the coherence function from two detectors. The frequency spectra were read from the spectrum analyzer through its HP-IB interface into a microcomputer (IBM XT) equipped with an HP-IB to MS-DOS interface (HP-82990A). Once stored on the computer, the data were scaled to correct for power drifts and normalized to ease comparisons. Finally, the data were analyzed by a non-linear least squares routine and the $\frac{\beta}{\ell}$ ratio was determined.

For all experiments, the detectors were operated in current mode. The current contained both a steady state component and a 'noise' component. The steady state current was suppressed using the current amplifier and the remaining signal was converted to a voltage signal and amplified above 1 volt peak to peak (amplification ranged from 10^7 to 10^9 volts per ampere, depending on the measured detector current). The correct suppression current was set by monitoring the DC voltage signal

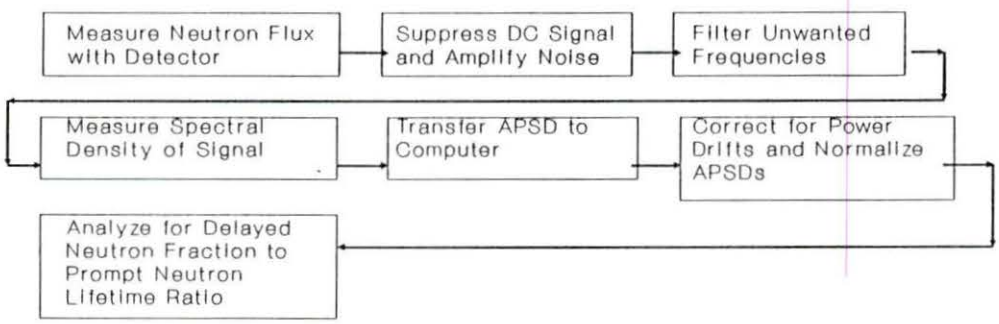


Figure 4.2: Detection Signal Path

with a digital multimeter and adjusting the suppression current to produce 0 volts DC. The remaining neutron noise component was monitored on an oscilloscope to ascertain its magnitude and to make sure it was correctly amplified and not overloading the equipment. Overloads could occur at the input to the current amplifier and at the input to the spectrum analyzer. They were caused by the reactor power drifting and by very large signal fluctuations.

The 'noise' voltage signal from the current amplifier was passed though a low-pass filter and then a high-pass filter. The filters reduced the low-frequency noise including any remaining DC component and reduced the noise with frequencies above the range of interest. The filters were set at the 'cut-off' frequencies 0.1 Hz and 40.0 Hz. If the high-pass filter did not remove the low-frequency noise, there would be a large low-frequency component observed in the resulting spectrum. If the low-pass filters were set above the 60 Hz electrical network, then the 60 Hz noise component limited the sensitivity level on the input into the spectrum analyzer.

In this frequency range the effects of delayed neutrons could be ignored (as was done during the derivation of Equation 3.29) and the effects of the 60 Hz electrical network were not significant. The frequency range used during data analysis was 0.8 Hz to 30.0 Hz ($5.0s^{-1} < \omega < 190s^{-1}$). The range was smaller than the filter settings to reduce the effect of the filtering process on the data fitting process.

The filtered signal was fed into the spectrum analyzer. The analyzer developed the APSD and/or the coherence function using an internal RMS averaging routine on 64 (user-chosen) time records of 'noise' (the RMS averaging will be explained in the next section). The RMS averaged spectrum was then stored on an MS-DOS disk

in the IBM-XT. At this time the reactor power was returned to the starting level, if necessary, by manipulation of the regulating rod or shim-safety rod. The reactor power would change with the slowly varying inlet temperature. The change was about 2%, but after amplification the voltage signal was large enough to overload the current amplifier. Once the power level was re-established, the next RMS collection was started. Except where noted, this was done 10 times per experiment.

Details of Equipment

Detectors

Some of the one-detector APSD measurements were made using a Westinghouse uncompensated ionization chamber (UIC) type 6937A. The argon-nitrogen filled chamber had a sensitive length of approximately 7-in. lined with boron enriched in neutron sensitive ^{10}B. The overall size of the cylindrical ion chamber was 13.75-in. long by 3-in. in diameter. The chamber has a thermal neutron sensitivity of 4.4×10^{-14} amperes per neutron per square centimeter per second[17]. The high-voltage power supply which provided the necessary voltage bias to the chamber was set to +500 VDC.

Ion chambers produce a current directly proportional to neutron flux. The chambers are not very sensitive to the applied voltage; the bias is such that the detector is operated in the ion saturation region. In the ion saturation region, all the ions produced are being collected, so an increase in voltage will not increase the ionization current.

The large size and poisoning effect of the Westinghouse 6937A ion chamber made

it unsuitable for coherence function measurements because two detectors were needed side by side, but it was well suited for one-detector APSD measurements. The APSD using the UIC was recorded while the detector was in Locations 1, 2, and 3. When the UIC was in the internal reflector or above the CVS, it was held in place by a Plexiglas detector mount which electrically isolated the detector from the internal reflector. When it was in the shield tank, it was mounted in a leak-proof aluminum canister, isolated from the water.

The Westinghouse fission chamber (model 6376A) was also used in this study. Its thermal-neutron sensitivity was 1.4×10^{-13} amperes per neutron per square centimeter per second. The neutron sensitive material on the detector wall was U_3O_8 enriched in ^{235}U to greater than 93%. The chamber was 11.5-in. long by 2-in. in diameter, and the sensitive length was 6-in. The walls of the fission chambers are lined with enriched uranium, and the ions are produced by the highly-charged fission products from neutron-induced fission reactions at the walls. There is a constant background of alpha particles from decay of ^{234}U, a small constituent in fission chambers. The alpha-induced pulses were overwhelmed by the charges created by fission fragments during chamber operation in a neutron flux.

The measurements made with the fission chamber were made with the chamber mounted in an aluminum canister shown in Figure 4.3. The chamber was electrically insulated from the canister, so the canister acted as a EM shield, reducing signal contamination. The canister was positioned in the reactor and electrically insulated from the core by a Plexiglas assembly.

The fission chamber was equipped with a female HN type connector. To maintain

the EM shield along the signal cable, it was necessary that leads of the HN connector be protected by an interference shield at the same potential as the canister. This was accomplished by adapting the HN to a twin-axial, three-lead cable. The outside shield of the twin-axial cable was connected to the canister, and the two inside leads were connected to the two leads of the HN connector. The device used is shown in Figure 4.4.

The setup allowed the chamber to be operated in current mode, producing an electric current proportional to neutron flux. Bias voltage for the fission chamber was provided by two 90-volt batteries connected in series. This made an isolated system which operated in current mode, with the current being sent to the current amplifier. The aluminum canister and outside shield of the twin-axial cable provided a shield against external noise sources.

Like the ion chamber, a pair of fission chambers was too large for coherence function measurements, but one was useful for APSD measurements.

The boron-trifluoride counter is a type of neutron detector filled with BF_3 gas enriched in ^{10}B. The BF_3 counters used in this study were manufactured by N. Wood Counter Laboratory (model G-10-5). The sensitive dimensions were approximately 6-in. long by 1-in. in diameter, with an overall length of about 8.5 inches. The thermal neutron sensitivity of the detector operated in current mode was not provided by the manufacturer, but, based on measured currents, it was on the order of 1×10^{-15} amperes per neutron per square centimeter per second.

The small size of the BF_3 detectors made it possible to put two detectors close together in the core and measure the coherence function of their signals without

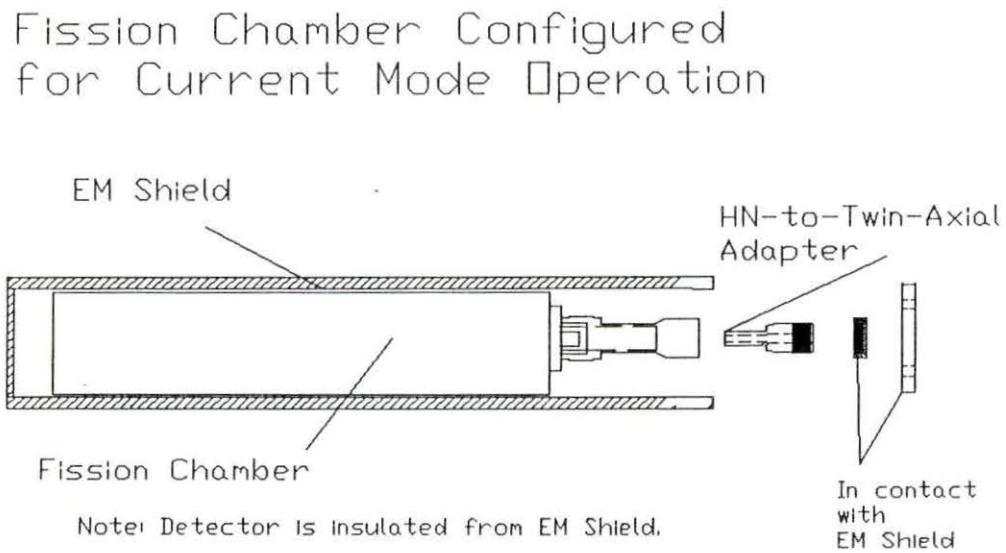


Figure 4.3: Canister for Fission Chamber

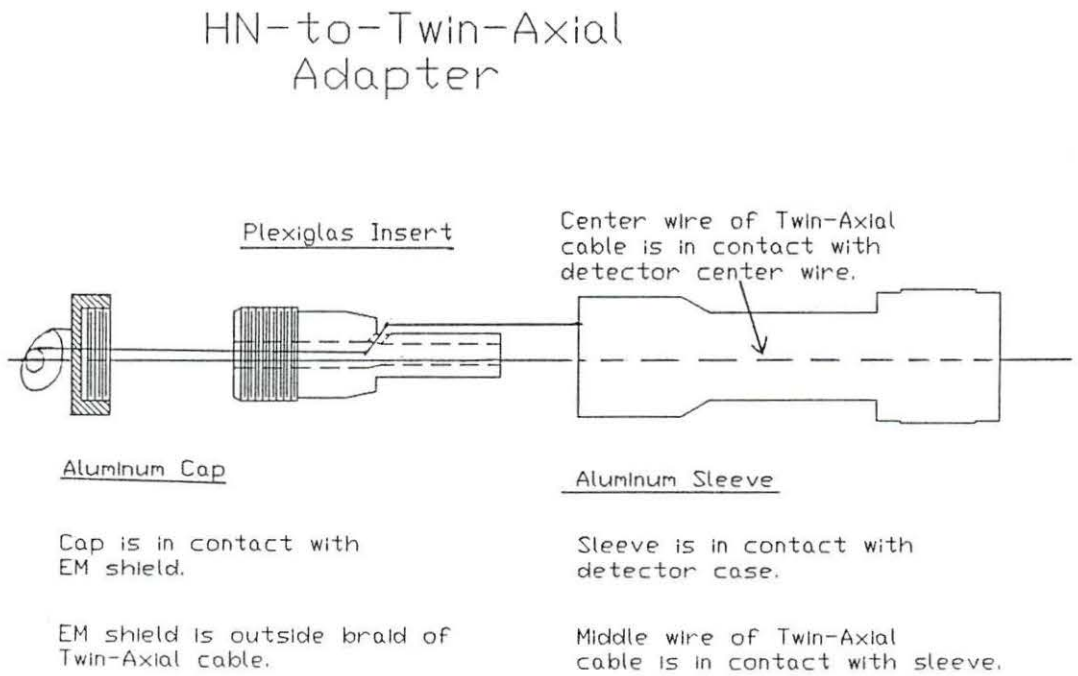


Figure 4.4: HN to Twin-Axial Connector

extreme disturbance of the neutron flux profile and the $\frac{\beta}{\ell}$ ratio.

The BF_3 detectors were mounted in aluminum canisters, which were then placed inside the modified 24-in. graphite stringer. The modified 24-in. stringer had two 1.25 inch diameter by 10.5-in. holes drilled into the top. The graphite stringer held the detectors together for coherence function measurements and avoided significant reactivity reductions so the estimated delayed neutron fraction and prompt neutron lifetime were closer to the clean core condition. The BF_3 detectors were also equipped with HN connectors, and they were converted from pulse to current mode in the same manner as the fission chamber. The only significant difference in the operation of the BF_3 detectors as compared to the fission chamber was that the detector canisters were in electrical contact with the graphite of the reactor core. This did cause some problems, as will be explained in the results section.

The Low Frequency Spectrum Analyzer

The Hewlett-Packard 3582A low frequency spectrum analyzer has many features and functions. Its main function is to convert time domain voltage variations in the input signal into APSD and coherence function spectra (frequency domain information) by performing a Discrete Fast Fourier Transform and other mathematical operations on the input signal. Capabilities include direct measurement of power spectral density, measurement of the transfer function, and measurement of the coherence function. It may have one or two input signals. Output is available by directly reading the screen or by accessing the HP 3582A remotely through a HP-IB interface. For complete details on the operation of the analyzer, consult the operating manual

[18].

The spectrum analyzer is operated by selecting a frequency range of interest called span, a passband shape, signal input (one or two channels and channel sensitivity), and signal output (coherence function, transfer function, signal amplitude). The spectrum analyzer then digitally analyzes the input signals using a fast Fourier transform and stores the results in 128 or 256 frequency bins, for one or two input signals respectively, for both display and internal calculations. Each frequency bin represents frequencies in a narrow bandwidth; if the analyzer is set for a 0 Hz to 100 Hz span with a single input channel then the frequency bin width is 0.4 Hz. The frequency bin width is always a number such as 0.4 because the 0 to 100 Hz span actually covers 0 to 102.4 Hz. The analyzer has a number of averaging routines and several methods of data display.

The time required for data collection decreases with increasing span. The time needed to process a single time record is 2.5 seconds if the span is set for 0 Hz to 100 Hz, while the process time decreases to 0.01 second for the 0 Hz to 25 kHz span.

During all the data collections, the RMS averaging routine was used. The RMS routine averages two or more spectra, smoothing the noise variations. This routine effectively measures the APSD when used with a single input, and it is required for coherence function measurements. The RMS average mode combines each new spectrum with the partial result in each frequency bin f such that, for m spectra collected, the amplitude $A(f)$ is given by

$$A(f) = \left(\frac{1}{m} \sum_{i=1}^{m} \left(A_i(f) \right)^2 \right)^{\frac{1}{2}} . \tag{4.1}$$

The result of this process is a smoothing of the noise variations.

The spectrum analyzer can measure the coherence function of two signals using

$$\gamma^2(\omega) = \frac{|G_{xy}(\omega)|^2}{|G_{xx}(\omega)||G_{yy}(\omega)|}. \tag{4.2}$$

Details such as the confidence limits as a function of the number of RMS averages and experimental examples of coherence function used on the HP 3582A analyzer may be found in reference [19].

The amplitude and coherence function at each frequency of an APSD measurement can be read directly (by the experimenter) using a movable *marker* on the HP 3582A. The analyzer interactively displays the frequency in hertz and the amplitude in volts per square-root hertz at the point selected by the marker. The amplitude, and not the power, of the APSD spectra is displayed by the HP 3582A analyzer in order to increase the range displayable at once. The amplitude squared is the power. The spectrum analyzer can display two traces, or spectra, at the same time.

Reading and recording the marker information by hand was time consuming and subject to error. Therefore, several programs were developed in C language which read and controlled the analyzer remotely. C was chosen over BASIC and PASCAL because it can directly access the spectrum analyzer memory and has superior handling routines for strings and octal numbers.

The marker position can be set remotely and the frequency, amplitude or coherence function pairs could be read through the HP-IB interface using the program *lmk.C*. When *lmk.C* is used to read the display, the program does the following: 1) prompts the user to set the spectrum analyzer marker on the desired trace and to adjust several controls on the spectrum analyzer to correctly read the marker position, 2) reads the information from the display in frequency, amplitude or coherence pairs,

and 3) stores the information in a user-selectable file.

Data read using *lmk.C* can be corrupted by the experimenter if the controls of the spectrum analyzer are improperly adjusted. The user can accidentally select the wrong trace to be read, choose the wrong scale, set the wrong amplitude, or fail to turn the marker on. Not all of these errors can be detected by *lmk.C*, so proper data transfer was verified manually after each collection and storage.

All of the potential user-errors during data transference can be eliminated in the case of the single channel APSD measurements by directly reading the random access memory (RAM) in the spectrum analyzer. In addition, the data transfer was much faster when the RAM was read than when the display was read. There was a price to pay for freedom from user-mistakes and the faster speed: the data stored in RAM are uncorrected by filtering routines internal the the HP 3582A analyzer, and are not scaled to units of volts squared per hertz.

The RMS-averaged data from a single channel are directly read from the RAM of the spectrum analyzer using the program *readrms.C*. The program has five main steps: 1) it reads in the raw bytes from the analyzer and converts them to floating point format, 2) it prompts the user for the number of averages and the span, 3) it uses the user input to select a correction file (already stored on disk), 4) it corrects the data read from the analyzer and converts them into APSD format, and 5) it stores the results in a user-chosen file. The corrections are necessary because reading the memory of the HP 3582A bypasses internal filters; the response of the internal filters is calculated by a separate C program called *maknorm.C*. The internal corrections are performed automatically before data are displayed on the analyzer, but to increase

computational speed the RAM-stored spectra are not internally corrected[20].

The corrections necessary are found using *maknorm.C* in combination with an experimental setup detailed in reference [20].

The data read using *readrms.C* can be corrupted by the user only if the user enters the wrong span or the wrong number of averages when prompted by the program.

Analysis of Data

For a typical single chamber experiment, 10 spectra, consisting of 64 RMS averages each, were collected. The low-pass filter was set at 40 Hz, and the high-pass filter at 0.1 Hz. The flattop passband shape was chosen based on recommendations in the HP 3582A manual.

Collecting ten sets of 64 RMS-averaged spectra had two advantages over a single 640 average collection. If an overload or other anomaly occurred during the collection which rendered the spectrum invalid, only a single spectrum of up to 64 time records was lost. Since the spectrum analyzer does not provide a statistical variance for each frequency, the ten-set spectra were used to calculate the variance for each frequency bin.

The reactor power could not be held exactly constant during data collection. The power variations, due primarily to changes in inlet temperature, changed the magnitude of the spectral densities. The measured power spectrum, as given by Equation 3.51, varies directly with the steady state neutron population n_O. To compensate for the reactor power drifts, the APSD of each individual frequency bin in each spectrum was normalized so that the sum of all the power spectral densities measured across

the entire span were equal to the sum in every other run made under the same conditions; typically, the data were normalized to the first set of 64 RMS-averaged spectra taken that day. The necessary calculations were done using a computer spreadsheet (Symphony[21]), but any other method is acceptable. Note that the high-pass filter reduces the distortion present in the 0 Hz to 0.4 Hz frequency bin by reducing the DC signal, but it does not correct for any other power-drift effects.

Next, each frequency bin in the normalized spectrum was averaged with the corresponding bins in other sets collected during the same reactor run, and the sample standard deviation of the APSD at each frequency bin was estimated. The averages and standard deviations were then scaled such that no average exceeded unity. Scaling to unity made plotting the data simpler and made the use of any fitting routine easier, and in no way affected the break frequency. Again, all necessary calculations and data manipulations were done using a spreadsheet. The output of the spreadsheet consisted of frequency, power, and standard deviation data triplets ready for input into any fitting routine.

The delayed neutron fraction to prompt neutron lifetime ratio was found by fitting the APSD and coherence function equations to the normalized and scaled measured spectra. If the APSD of a single detector was measured, then Equation 3.52 was used in the fitting routine. If the coherence function between two detectors was measured, then the parameters of Equation 3.66 were obtained.

Equation fitting was accomplished using a program called MINSQ[22][23], which was based on a method developed by Powell that did not require derivatives of the function to be fit. The equations were also fit using CURFIT [24], a program based

on the algorithm of Marquardt. A comparison of both programs gave similar results and similar error estimates[25] when used to fit Equation 3.52 to the same measured spectrum. MINSQ was chosen as the program to use for the rest of the study because it produced the variance of the fit parameters and because the driver programs (the subprograms which contain the equation to be fit) were easily modified.

MINSQ reads data triplets of frequency, power (or coherence function), and sample standard deviation of power (or coherence function), and fits the parameters of Equation 3.52 (or Equation 3.66) to the data, thereby predicting the $\frac{\beta}{\ell}$ ratio and its uncertainty. The program is easy to use and quickly converges on a solution. The output of MINSQ, in combination with the driver programs, includs the input data, the parameters and the uncertainties of the fitted function, and the values produced by the fitted function. The driver routine for fitting the APSD data is called FIT1.FOR, and the driver routine for fitting the coherence function data is called FITCOH.FOR. Both driver routines and sample input files used by each are presented the appendices.

Although the constants A, B, and F in Equations 3.52 and 3.66 were found, they are not useful for prediction of any reactor parameters. The constants have unknown terms, in particular the efficiency in units of neutrons counted per neutron present in the core. Also, the amplification, filtering, and scaling all alter results of the estimates of these parameters.

CHAPTER 5. RESULTS AND DISCUSSION

There were five different types of experiments performed which produced estimates of the delayed neutron fraction to prompt neutron lifetime ratio. They were based on:

- Measurement of the APSD of a UIC in Location 1,
- Measurement of the APSD of a fission chamber in Location 1,
- Measurement of the APSD of a UIC in Location 2,
- Measurement of the APSD of a BF_3 chamber in Location 4,
- Measurement of the coherence function of two BF_3 chambers in Location 4.

The reactor was operated as a zero-power system during all measurements. The power levels were 2 watts for measurements with a UIC or fission chamber, and 500 milliwatts for those using the BF_3 chambers. Unless noted, the reactor was in manual control. All instrumentation was set to provide the maximum signal without overloading the current amplifier or the spectrum analyzer.

Table 5.1: Measured Detector Current

Detector	Location	Current
FC	1	4900 nA
UIC	1	1600 nA
UIC	2	83.7 nA
UIC	3	0.275 nA

Single Detector Measurements

The measured DC currents for the UIC and fission chamber in each location are listed in Table 5.1. Location 1 required a 28-in. void in the internal reflector to accomodate the detector. Location 2 required that the base of the detector was 4 in. above the top of the inserted stringers. The detector holder base plate and position of the sensitive area in the detector acted to raise the detector about four inches above the top of the stringers, even with the top of the internal reflector.

By comparing the currents from the uncompensated ion chamber, the relative flux at each location can be determined; the highest neutron flux was observed in the internal reflector. The current measured from the fission chamber is not directly comparable to the ion chamber current in the same location, as the detectors are different sizes and have different efficiencies. The sensitivities of the detectors differ by a factor of 3.2, which agrees with the 3.1 factor difference in the measured currents of the two detectors.

The results of the analyses of the measured spectral densities using a single uncompensated ion chamber are listed in Table 5.2. Two measurements were made on different days at each location to demonstrate the results were repeatable. The break frequency measured in the internal reflector (Location 1) is the more precise

Table 5.2: Estimated $\frac{\beta}{\ell}$ from UIC Measured Spectra

Location	$\frac{\beta}{\ell}, s^{-1}$	$\sigma(\frac{\beta}{\ell}), s^{-1}$	$\rho_{excess}, \$$
1	42.5	1.8	0.06
1	42.2	2.0	0.08
2	39.4	3.0	0.69
2	39.0	3.1	0.69
3	n/a	n/a	0.72
3	n/a	n/a	0.72

estimate of the $\frac{\beta}{\ell}$ based on the standard deviation. The break frequency is higher when the detector is inside the internal reflector. This is due to the reduced prompt neutron lifetime caused by the poisoning effect of the boron-lined ion chamber and the increase in leakage caused by the displaced graphite from the internal reflector. The poisoning is shown by the reduced excess reactivity ρ_{excess} (a measure of how supercritical the reactor could become with control blades withdrawn); a lower excess reactivity indicates a more poisoned reactor.

Break frequencies predicted using a UIC in the shield tank (Location 3) are not applicable (n/a) because the APSD measured there was white due to the distance from the core. The whiteness was caused by the effects of the graphite duct connecting the shield tank to the core. The \$0.72 excess reactivity present when the detector was in the shield tank suggests that the detector was decoupled from the fission process in the core because the excess reactivity of the UTR-10 with no extra detectors is also \$0.72. It is necessary that a decoupled detector not change the excess reactivity, but it is not a guarantee that the neutrons collected by such a detector are free from the effects of the fission process.

The plots of APSD and coherence function were made using a plotting program called *Grapher*[26]. The plotted points represent the normalized and scaled measured APSDs (or coherence function), and the smooth curves are a spline fit to the MINSQ-calculated APSD (or coherence function) data. The spline algorithm is part of to *Grapher*.

Figure 5.1, the spectrum measured in the internal reflector (Location 1) using a UIC, showed the most linearity (on the log scale) from the break frequency to the maximum frequency displayed. It also produced the lowest standard deviation in the break frequency, and had the smallest powers at higher frequencies, indicating its location made it more efficient. This was expected since the entire detector was below the top of the graphite, exposing all the neutron sensitive area to neutrons. A total of 10 runs of 64 time records each were RMS averaged.

The sample standard deviations of selected bins of the APSD recorded during the above measurement is shown in Figure 5.2. The relative deviation was about 10% for each frequency bin. The standard deviations estimated during the other runs were similar, but were not plotted.

The reactor was operated in the automatic mode to investigate the effects of the automatic control blade movements. The measured APSD which resulted is shown in Figure 5.3. The automatic power controller caused an increase in the power at the lower frequencies. To correct for this, the frequencies below 1.2 Hz were not considered during equation fitting. Otherwise, the effects of the automatic controller caused the estimated break frequency to drop to 6.1 Hz. The distortion in the lower frequencies demonstrated the need for manual control during measurements.

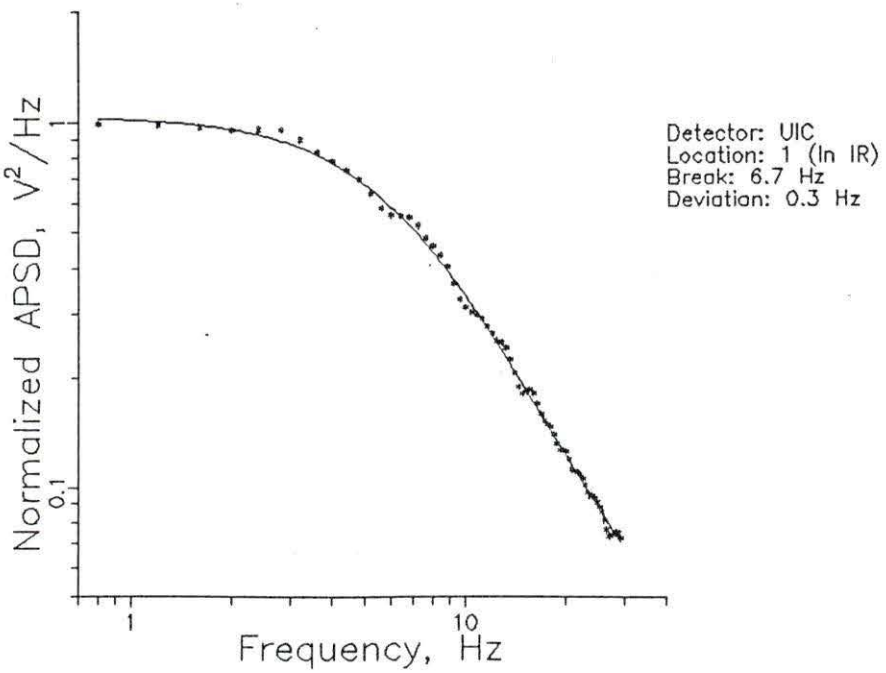


Figure 5.1: APSD from UIC in Location 1

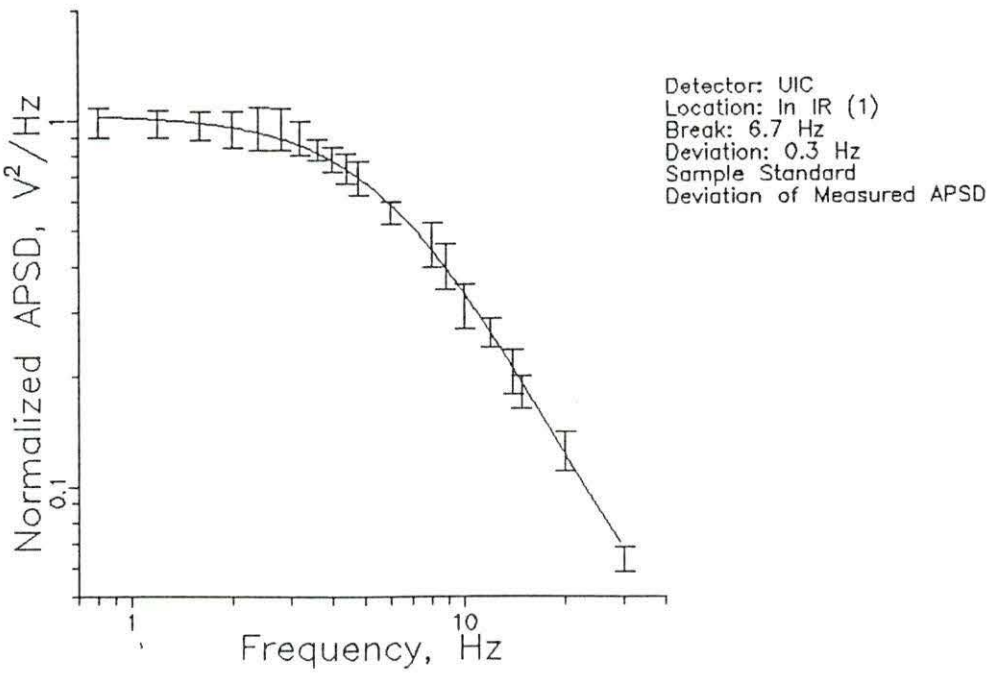


Figure 5.2: Standard Deviation of APSD from UIC in Location 1

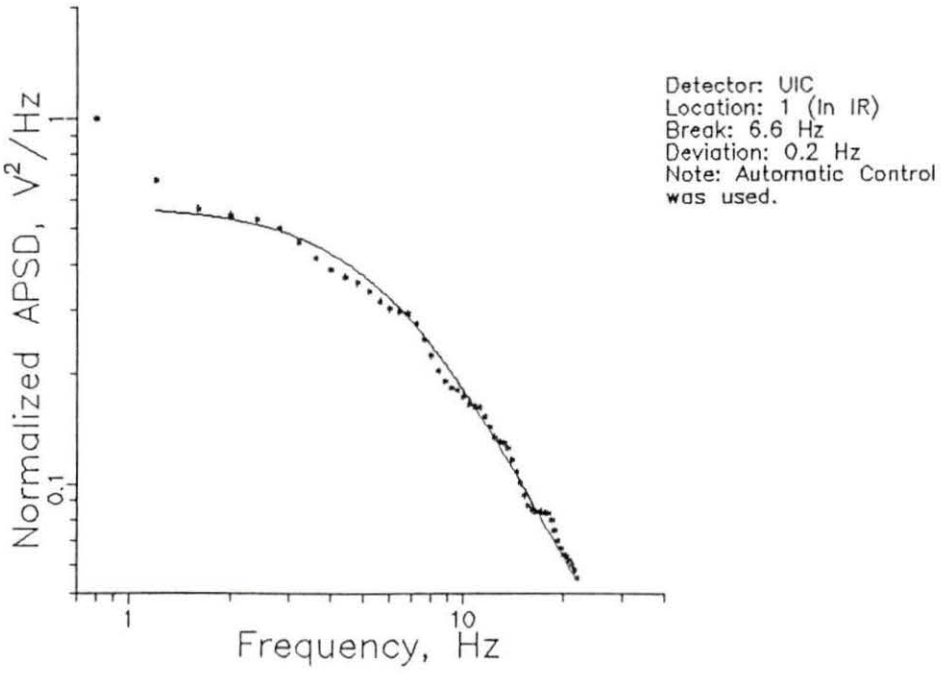


Figure 5.3: APSD from UIC in Location 1 with Auto Control

The result for an uncompensated ion chamber placed at Location 2 is shown in Figure 5.4. When the detector was placed above the internal reflector, the variance was greater than the variance of the measured APSD for a detector in Location 1 in both the measured spectrum and in the break frequency. Since only the bottom four inches of the detector and the detector holder plate could be inserted into the internal reflector when Location 2 was used, the detector was relatively less efficient. The lower efficiency was evidenced by the leveling off of the APSD beyond about 20 Hz. Beyond this frequency, the B term in Equation 3.52 dominated. At the same time, however, the flux profile and the neutronics parameters were less disturbed. The detector did not poison the core to the same extent and the leakage was less due to the presence of more graphite.

A second run with the uncompensated ion chamber at Location 2 produced very similar results. The predicted break frequency was within less than 50% of one standard deviation of the first run. The second run was made using 12 collections instead of 10.

The two APSDs recorded using an uncompensated ion chamber in the shield tank (Location 3) are shown in Figures 5.5 and 5.6. The APSD show no common peaks or valleys in spectral power, which suggests there is no systematic noise contaminate present. The plots have no apparent break frequency; the spectral power does not appear to vary with frequency. Under the assumption that the fission-decoupled neutrons being detected are a white noise source, the frequency independence of the observed APSD suggests the instrumentation transfer function is frequency independent. Therefore $G'_{ii}(\omega)$, the corrected APSD, differs from $G_{ii}(\omega)$, the measured

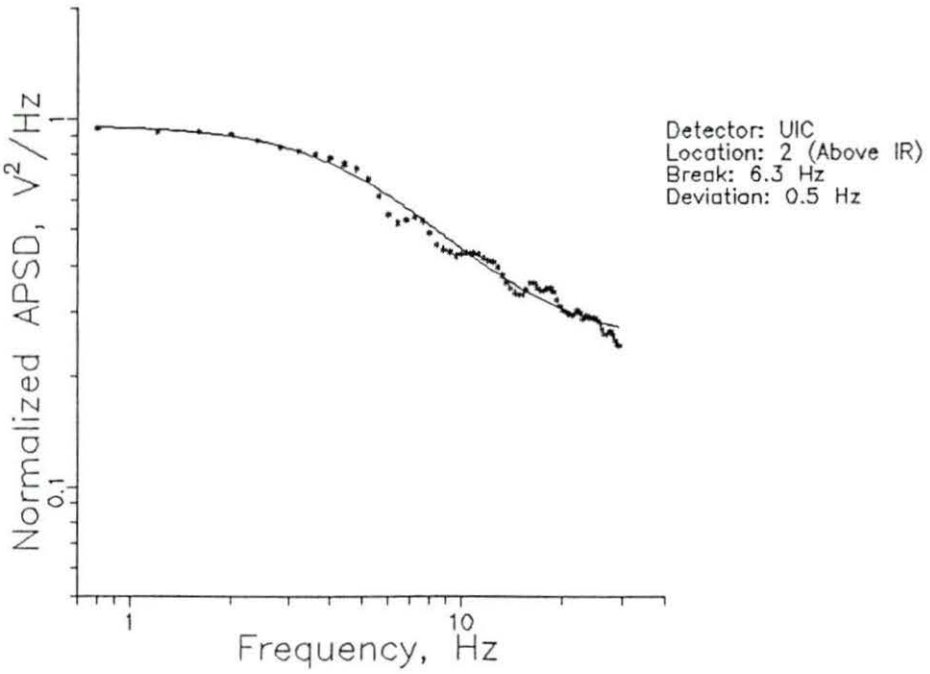


Figure 5.4: APSD from UIC in Location 2

APSD, by a constant, and the instrumentation transfer function H_{inst} is frequency independent in the frequency range of interest. The predicted break frequencies and their deviations were calculated but are meaningless. Only five collections were performed during the first run, but the second resulted from ten 64 RMS averages.

The spectrum at Location 1 was also measured using a fission chamber. The fission chamber was operated in current mode and was powered by two 90-volt batteries connected in series. The break frequency estimated from the data thus obtained $45.0 \pm 2.8s^{-1}$, within one standard deviation of the frequency predicted by the UIC measured spectra. The measured spectrum is shown in Figure 5.7

Since the sensitivity of the fission chamber was three times that of the UIC, the leveling-out beyond 20 Hz should not have appeared. Its presence suggests there was a contaminating noise source present on the signal.

Coherence Function Measurements

The coherence function measurements were performed using two boron-trifluoride detectors operated in current mode and positioned in Location 4. The reactor power was held at 500 milliwatts to reduce damage to the detectors, as will be explained later.

The 24-in. stringer had two 1.25-in. diameter holes drilled 10.5 inches into the top for detector positioning. The stringer was placed in the internal reflector of the core so that the detectors were on the west side. The outside of the detector cases were 1.125-in. apart; the cases were as close together as possible. Each detector was operated in current mode with bias supplied by a single 90-volt battery; the center

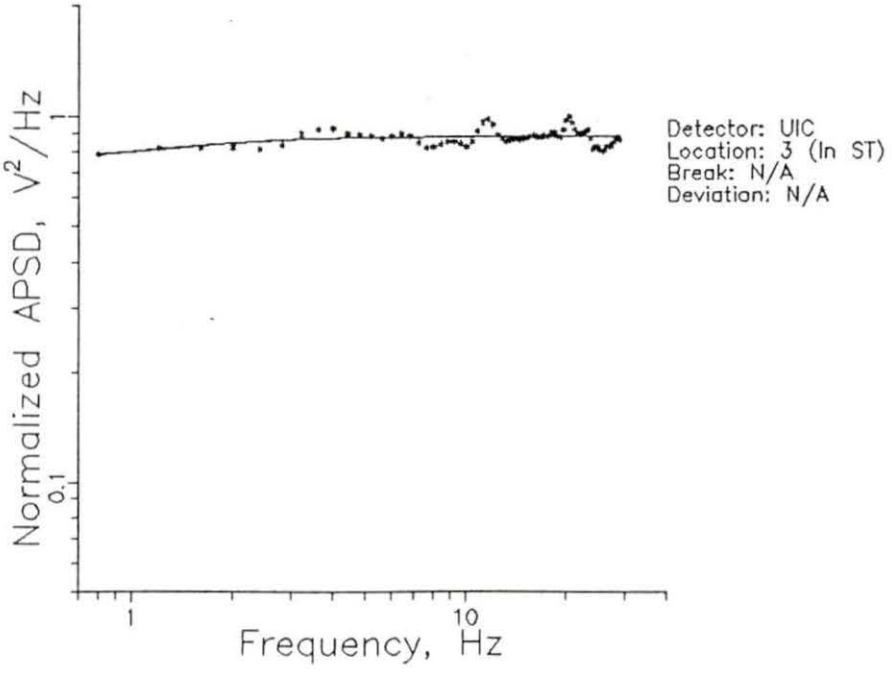


Figure 5.5: APSD from UIC in Location 3

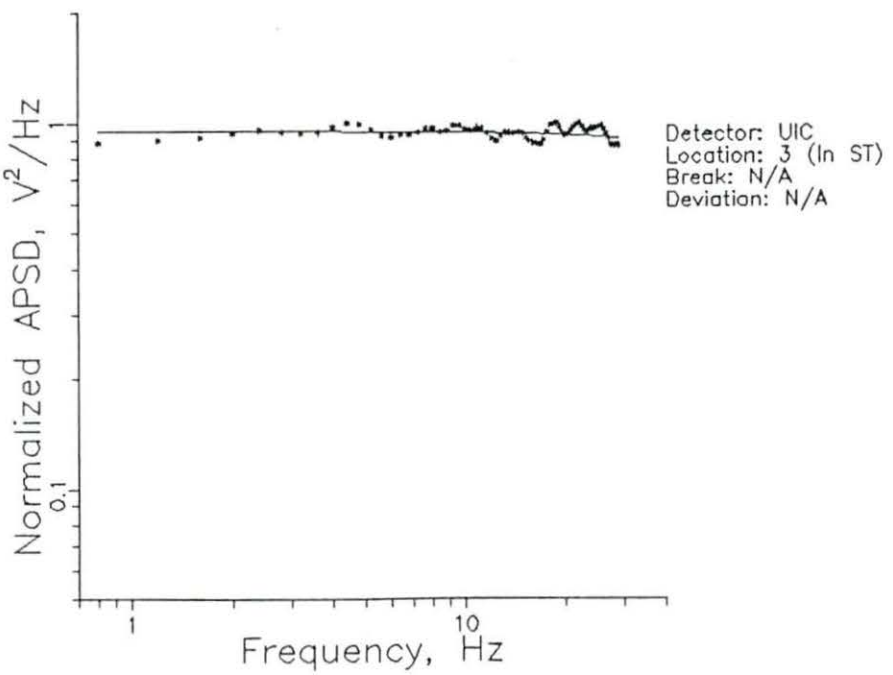


Figure 5.6: APSD from UIC in Location 3: Second Run

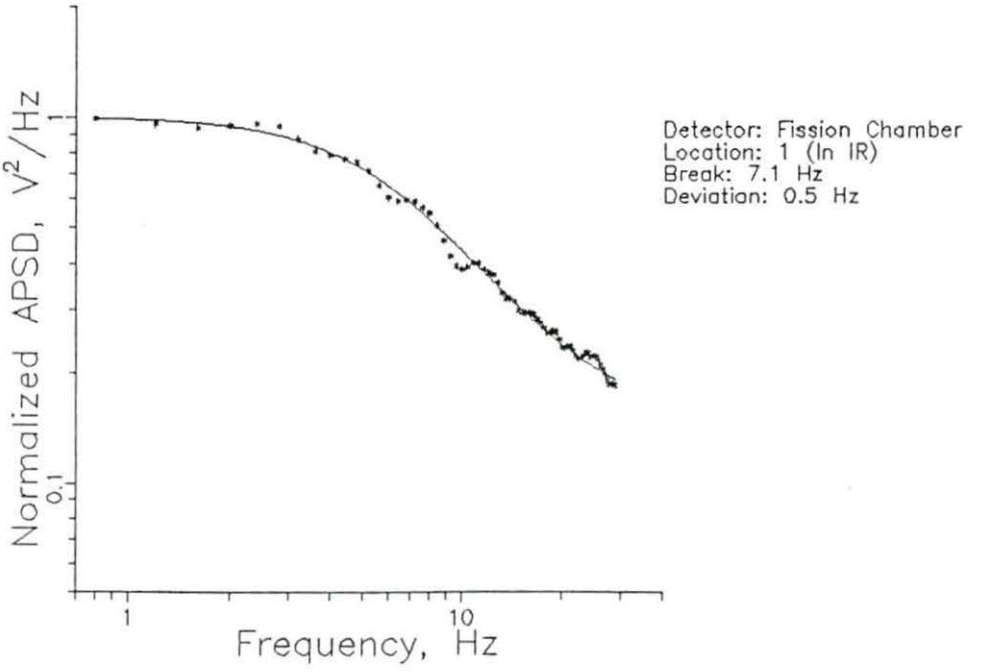


Figure 5.7: APSD from Fission Chamber in Location 1

wire of the detector was attached to the negative pole of the battery.

The signals from each detector were processed just like the signal from a single detector, e.g. amplified, filtered, and sent to the spectrum analyzer. Data collection was performed like the one-detector collections except for the following: each run was 128 time records in order to reduce uncertainty, the span of the spectrum analyzer was set for 0 to 50 Hz to maintain 0.4 Hz-wide frequency bins, and the spectrum analyzer was set for two inputs instead of one. The spectrum analyzer directly calculated the coherence function between the two detectors. Because the coherence function is independent of the neutron population and instrumentation, no correction to the measured coherence function was needed for power drift, although the reactor power was still restored after each collection. The spectrum analyzer directly calculated $\gamma^2(\omega)$ and updated its display after each time record. The measured coherence function was transferred from the spectrum analyzer to the microcomputer using *lmk.C*.

Several problems appeared during the measurement of the coherence function. First, a series of echos of 60 Hz was observed coming from one of the electronic filters. The echos were investigated using the coherence function and the built-in random noise source on the spectrum analyzer, and it was determined that the echos did not have any effect in the frequency range of interest.

There was an intermittent fault-to-ground present in the wiring scheme which was caused by the detector casings being electrically connected to the reactor core, which was itself electrically floating above the electrical ground of the current amplifier. The intermittent grounding problem was cured by connecting a ground wire to the

same ground as the current amplifier input. This reduced the amount of electronic shielding against spurious noise sources, but the additional noise did not significantly degrade the signal.

One detector failed and had to be replaced during the coherence function measurements. The failure was probably caused by the unusual use of the BF_3 in current mode instead of pulse mode. Operation in current mode might have caused a significant buildup of charge in the detector, which eventually caused it to fail. The failure was evident through a reduction in measured current from the detector, and by observation of the coherence function. The coherence function of the operational detector and the failed detector was at a near zero level throughout the frequency range of interest.

To slow any charge buildup in the detectors, the reactor power was reduced from 2 watts to 500 milliwatts. Around 500 milliwatts, the detectors responded linearly with power, and the requirements of a zero-power measurement were still fulfilled.

The measured coherence function of two BF_3 detectors in Location 4 is shown in Figure 5.8. Because the coherence function values given by the spectrum analyzer had a minimum non-zero value of 0.1, which was reached at about 20 Hz, the estimate of the break frequency was made using data pairs in the 0.8 Hz to 15.2 Hz range. The result of fitting Equation 3.66 was $\frac{\beta}{\ell} = 41.5 \pm 2.3 s^{-1} (68\%)$. Shifting to evaluation of smaller range of data did not significantly alter the predicted break frequency; the break frequency estimated by evaluation of the 1.2 Hz to 12.0 Hz range was 42.1 per second.

The spectrum analyzer measures the APSD of each channel during measurement

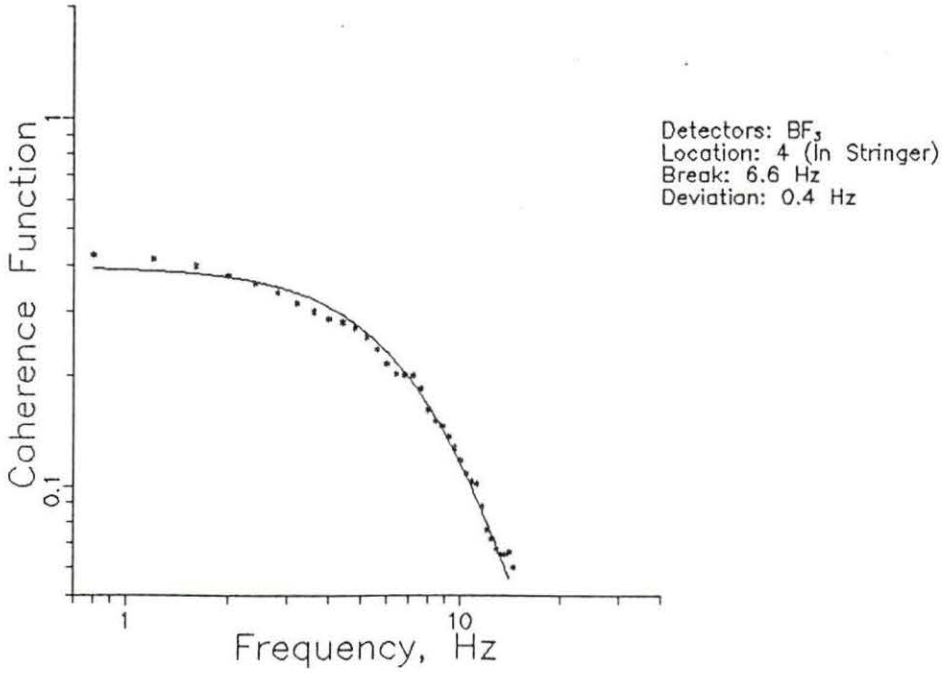


Figure 5.8: Coherence Function from Two BF_3 Detectors in Location 4

of the coherence function. These can be used to calculate the break frequency in the same manner of the single detector measurements. The measured APSD from one BF_3 detector recorded during the same time as the previous coherence function is shown in Figure 5.9. The break frequency estimated using the 0.8 to 30 Hz range of the measured APSD was $48.5 \pm 2.3(68\%)$ per second, higher than the break frequency based on the coherence function. The second BF_3 detector also had a high break frequency. The low efficiency of these small detectors was probably responsible for the difference between these and previous estimates, and for the small observed current of 14.7 nanoamps. It is also possible that the instrumentation had a frequency dependent effect on the measured APSD; the use of the coherence function eliminated any instrumentation effects.

Comparison of Calculated and Measured Results

The measured break frequencies were smaller than the calculated break frequencies, implying the calculated effective delayed neutron fraction was too large, or the calculated prompt neutron lifetime was too short. A comparison of the calculated values of the delayed neutron fraction to prompt neutron lifetime ratio to some of the measured values is shown in Table 5.3. The value calculated at Argonne was higher than any measured value except the values estimated using the APSD from a BF_3.

The calculated values of β and ℓ may be used to separate the measured break frequency into an estimated delayed neutron fraction or a prompt neutron lifetime. Assuming the β calculated at Argonne to be correct and exact, the predicted prompt neutron lifetimes are $181 \pm 8\mu s$ and $185 \pm 10\mu s$ (68%) for the APSD of the UIC and

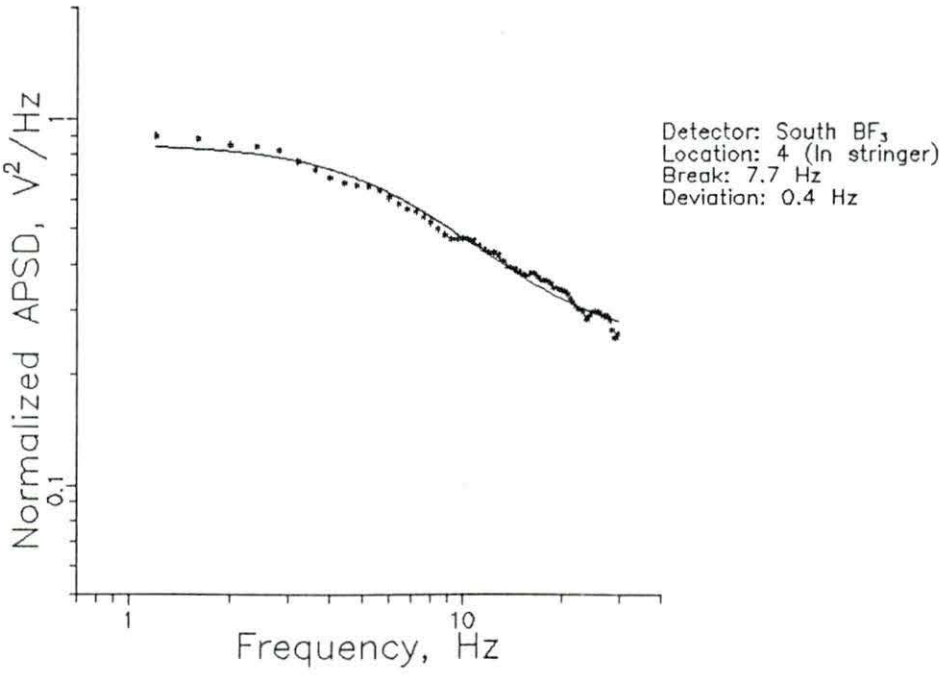


Figure 5.9: APSD from BF_3 Detector in Location 4

Table 5.3: Core Parameters for HEU Core

Method	$\frac{\beta}{\ell}, s^{-1}$	β	$\ell, \mu s$
Calculated at Argonne	48.1	0.0077	160
Currently used at ISU	43.3	0.0065	150
Estimated From APSD of UIC in Location 1	42.5	n/a	n/a
Estimated From Coherence Function	41.5	n/a	n/a
Estimated From APSD of BF_3	48.5	n/a	n/a

coherence function measured spectra, respectively. On the other hand, if the lifetime calculated by Argonne is assumed correct, the effective delayed neutron fractions predicted by the APSD of the UIC and coherence function become 0.00680 ± 0.0003 and 0.00664 ± 0.0004 (68%), respectively.

The discrepancy between the calculated and measured estimates was probably due to the limitations of the computer code used to model the UTR-10. The high neutron leakage rate makes the UTR-10 difficult to model. The error in calculation was already evident from the failure of the model used at Argonne to predict that the reactor could become critical.

With the exception of the estimate made using the APSD from a BF_3 detector, the estimated values of $\frac{\beta}{\ell}$ were all within about 4 Hz. Since three different types of detectors and two different methods were used, this suggests the measured values, and not the calculated value, are more reliable.

Comparison of Reactor Transients

The HEU to LEU fuel conversion project required a study of reactor transients. The transient modeled for the UTR-10 was caused by a sudden insertion of all excess

Table 5.4: Response to Reactivity Transients

β	$\ell, \mu s$	Max power, kW	$\frac{\beta}{\ell}, s^{-1}$
0.00650	166	25.7	39.2
0.00763	166	26.2	46.0
0.00763	141	26.4	54.2

reactivity into a previously critical reactor. The system was then shut down by the reactor safety system through insertion of control blades. The study was to assure that such a transient would not produce a power which damaged the core or was a danger to personnel in any manner.

The program used to model the transient was PARET. The maximum powers and the time to reach those maxima are dependent on the delayed neutron fraction and the prompt neutron lifetime. The effects of varying these parameters on the transient are shown in Table 5.4. In each case, about 0.2 seconds elapsed before maximum power was reached; PARET did not produce a more precise time. The maximum power reached during a transient before the automatic safety system shut down the reactor does not strongly depend on the ratio of the delayed neutron fraction to prompt neutron lifetime. The trend of increasing maximum power with increasing $\frac{\beta}{\ell}$ (decreasing ℓ) is evident in Table 5.4.

CHAPTER 6. CONCLUSIONS AND FUTURE WORK

Conclusions

Digital neutron noise analysis has been used to estimate the delayed neutron fraction to prompt neutron lifetime ratio. The results of the measurements are summarized in Table 6.1.

The range of values estimated using different detectors, different locations, and different methods have overlapping one-standard deviation intervals, suggesting the true value of $\frac{\beta}{\ell}$ is in the estimated range. The method of measurement of neutron noise on a zero-power reactor has been successfully applied to the UTR-10 using state of the art equipment. In particular, the application of the low frequency spectrum analyzer in combination with digital computer analysis has improved the accuracy, speed, and ease of measuring the delayed neutron fraction to prompt neutron lifetime.

Table 6.1: Estimated $\frac{\beta}{\ell}$ from Neutron Noise

Method	$\frac{\beta}{\ell}, s^{-1}$	$\sigma(\frac{\beta}{\ell}), s^{-1}$
APSD from UIC at Location 1	42.5	1.8
APSD from UIC at Location 2	39.4	3.0
APSD from Fission Chamber at Location 1	45.0	2.8
APSD from BF_3 Detector at Location 4	48.5	2.3
Coherence Function from BF_3 pair at Location 4	41.5	2.3

The method may be used directly on the new LEU core without any modification.

Using the automatic power controller distorted the estimation of the delayed neutron fraction to prompt neutron lifetime ratio. The controller altered the reactor from a zero-power system to one with feedback. Estimation of the $\frac{\beta}{\ell}$ ratio should be done using manual control only.

The efficiency of the detector had an affect on the measured APSD, and the more sensitive detectors, such as the UIC and fission chamber, produced more precise estimates of $\frac{\beta}{\ell}$ than the BF_3 detector. The more sensitive detectors also produced the most visually evident break frequency. The break frequency of a low-sensitivity detector can be overwhelmed by the steady state noise produced by the detection process and instrumentation.

Estimation of the $\frac{\beta}{\ell}$ ratio is easier using the coherence function instead of the APSD. Use of the coherence function eliminates dependence on the difficult-to-measure instrumentation transfer function, and it also allows for use of less-efficient detectors. The break frequency estimated using the coherence function of the less-sensitive detectors is not overwhelmed by the steady-state noise.

The time and effort required for estimation of $\frac{\beta}{\ell}$ is reduced compared to previous studies. The estimation may be made from an APSD measurement, including data collection, in a single afternoon. The use of the coherence function reduces the time required by about one hour, so the analysis is further simplified. There are no tapes or filters to manipulate, after the initial data collection and transference. Currently, the necessary data manipulation is done with a spreadsheet, however a simple program could be written to automate the process.

The data transference codes were successfully written using the C language. The process of reading the spectra from the spectrum analyzer and storing the spectra on the computer was easiest using C. The direct reading of the HP-3582A memory eliminated errors due to misread or mis-transcribed numbers.

Use of MINSQ to estimate the delayed neutron fraction to delayed neutron to prompt neutron lifetime ratio eliminated errors produced by visual estimation of the break frequency. MINSQ also produced a variance in the estimate.

Future Work

In the future, these measurements will be repeated on the UTR-10 after the new low enrichment uranium fuel has been loaded. The measured $\frac{\beta}{\ell}$ ratio may then be used in generation of period and reactivity tables for the new core.

Some of the steps which could be improved are: the scaling/normalizing process, which could be performed directly using a specialized computer code, the data collection system which could be programmed to be automatic, and the computational model of the core could be improved so that its predictions of $\frac{\beta}{\ell}$ are closer to the measured values.

The experiments could be repeated using small fission chambers, ones small enough for coherence function measurements. The fission chambers are very sensitive, and may provide more precise results than the BF_3 detectors.

The $\frac{\beta}{\ell}$ ratio could be investigated in the region of the thermal column. The thermal column could also be used to investigate the effects of various lengths of graphite stringers on the APSD and coherence function.

BIBLIOGRAPHY

[1] Richard Hendrickson, Richard Danofsky, Alfred Rohach and Donald Roberts. *Safety Analysis Report for the Training Reactor UTR-10.* ISU-ERI-Ames-82418, 1981.

[2] D. Hetrick. *Dynamics of Nuclear Reactors.* Chicago, Illinois: The University of Chicago Press, 1971.

[3] James Matos. RERTR Division of Argonne National Laboratory. Private communication, 1988.

[4] Robert Williams, Richard Hendrickson and Alfred Rohach. *A Proposal for an Assistance Grant for the Conversion of the Iowa State University UTR-10 Research Reactor to Low Enrichment Uranium Fuel.* Iowa State University Department of Nuclear Engineering, Ames, Iowa, 1987.

[5] C. Obenchain. *PARET-A Program for the Analysis of Reactor Transients.* AEC Research and Development Report, Idaho, 1969.

[6] T. Chan. *Reactor Transfer Function Measurements with the Reactor Oscillator.* Ames, Iowa: Iowa State University Press, 1971.

[7] M. Nabavian. *Reactor Noise Measurements in the UTR-10 Using the Polarity Correlation Method.* Ames, Iowa: Iowa State University Press, 1973.

[8] A. Mikulski. *Some Recent Developments in Reactor Noise Analysis at Swierk.* Progress in Nuclear Energy, 1 701-706 (1977).

[9] J. Eller, Neutronics Division of Duke Power, Charlotte, North Carolina. Personal communication, 1989.

[10] J. Thie. *Power Reactor Noise.* La Grange Park, Illinois: The American Nuclear Society, 1981.

[11] W. Seifritz. *At-power reactor noise induced by fluctuations of the coolant flow.* Reaktorphysik, Atomkernenergie (ATKE) 16 5 (1970).

[12] C. Cohn. *A Simplified Theory of Pile Noise.* Nuclear Science and Engineering, 7 472-475 (1960).

[13] R. Churchill and J. Brown. *Fourier Series and Boundary Value Problems.* New York, New York: McGraw Hill Book Company, 1978.

[14] S. Conte and C. de Boor. *Elementary Numerical Analysis-An Algorithmic Approach.* New York, New York: The McGraw Hill Book Company, 1980.

[15] R. Uhrig. *Random Noise Techniques in Nuclear Reactor Systems.* New York, New York: The Ronald Press Company, 1970.

[16] A. Fitzgerald, D. Higginbotham, and A. Grabel. *Basic Electrical Engineering.* New York, New York: McGraw-Hill Book Company, 1981.

[17] *Westinghouse Detector Information Sheet.* Westinghouse Electric Corporation, Elmira, New York, March 15, 1960.

[18] *Operating Manual for Model 3582A Spectrum Analyzer.* Hewlett-Packard Company, Loveland, Colorado, 1978.

[19] *Measuring the Coherence Function with the HP 3582A Spectrum Analyzer.* AN 245-2. Hewlett-Packard Company, Loveland, Colorado, 1978.

[20] *Accessing the 3582A Memory with HP-IB.* AN 245-4. Hewlett-Packard Company, Loveland, Colorado, 1978.

[21] *Symphony.* Lotus Development Corporation, Cambridge, Massachusetts, 1987.

[22] T. Pomentale *Minimization of a Sum of Squares of Functions.* Cern Computer Centre Program Library. Amended 1978.

[23] M. Powell. *A method for minimizing a sum of squares of non-linear functions without calculating derivatives.* The Computer Journal, 7,303 (1965).

[24] P. Bevington. *Data Reduction and Error Analysis for the Physical Sciences.* New York, New York: McGraw-Hill Book Company, 1969.

[25] U. Colak. *Nonlinear Curve Fitting Programs.* Private communication, Department of Nuclear Engineering, Ames, Iowa, 1988.

[26] *GRAPHER.* Golden Software Inc., Golden, Colorado, 1988.

APPENDIX A. PROCEDURE FOR APSD MEASUREMENTS

The APSD from a single detector may be analyzed and an estimate of $\frac{\beta}{\ell}$ and its variance made.

Briefly, the steps necessary are:

1. Install a detector in the core and set it up for current operation. The detector should be placed in a region where it is exposed to fission-produced neutrons which have not been heavily moderated.

2. Bring the reactor up to a steady-state, zero-power condition. Two watts is sufficient.

3. Suppress the steady-state detector current. This may be done with a current amplifier equipped with a current suppression feature by observing the DC signal on a voltmeter or oscilloscope.

4. Amplify the noise component. The current amplifier should be adjusted to the maximum amplification possible which does not produce overloads.

5. Filter frequencies below and above the range of interest. Typically, the filters would be set at 0.1 Hz and 40.0 Hz.

6. Collect ten sets of 64 RMS averages on the spectrum analyzer, transferring each set to the IBM-XT for analysis. Time limitations may mandate fewer sets of data.

7. Normalize each APSD to correct for power drifts, and scale each to unity. The first data set collected is taken to be the norm by which all other sets are corrected.

8. Calculate the sample standard deviation of each APSD bin. A spreadsheet is a fast and efficient tool for calculation of the averages and standard deviations of each APSD bin.

9. Analyze the APSD using MINSQ. Other programs may be used for the fitting of the APSD equation to the measured data.

APPENDIX B. PROCEDURE FOR COHERENCE FUNCTION MEASUREMENT

The procedure for measuring the coherence function from two detectors and analyzing the data for estimation of $\frac{\beta}{\ell}$ and its variance is similar to that of the APSD measurements.

Briefly, the steps necessary are:

1. Install two detectors in the core and set them up for current operation.
2. Bring the reactor up to a steady-state, zero-power condition. Two watts is sufficient.
3. Suppress the steady-state detector currents.
4. Amplify the noise component.
5. Filter frequencies below and above the range of interest.
6. Collect ten sets of 128 RMS averages on the spectrum analyzer, transferring each coherence function set to the IBM-XT for analysis.
7. Calculate the standard deviation of each coherence function bin.
8. Analyze the coherence function data using MINSQ.

APPENDIX C. DRIVER PROGRAMS USED BY MINSQ

The following two programs are used by MINSQ to estimate the parameters of the coherence function and the corrected APSD. Both programs are written to accept frequency in hertz, therefore the A, B, and F parameters estimated are factors of 2π, 2π, and $(2\pi)^2$ different from the values as shown in Equations 3.52 and 3.66.

Driver for Fitting Coherence Function

FITCOH.FOR

```
**********************************************************
*                                                        *
      Program FITCOH
*                                                        *
**********************************************************
*                                                        *
*     By David Roth                                      *
*     Based on FIT1.FOR by Uner Colak                    *
*                                                        *
**********************************************************
*
*     Declaration
*
      REAL F(150),C(10),E(10),W(1000),YC(150)
      COMMON/B1/YL(150),XL(150),SD(150)
*
*     File Initialization
*
```

```
      OPEN(10,FILE='FITCOHIN.DAT',STATUS='OLD')
      OPEN(11,FILE='COHERTEXT.DAT',STATUS='UNKNOWN')
      OPEN(12,FILE='MEASCOH.DAT',STATUS='OLD')
      OPEN(13,FILE='COHERENCE.DAT',STATUS='UNKNOWN')
*
*     Read FITCOHIN.DAT for limits on iterations,
*     initial guesses, error tolerances, etc.
*
      READ(10,*) M,N,IPRINT,NFUN,NW,COV,XSTEP,IOPT
      READ(10,*) (C(I),E(I),I=1,N)
*
*     Write output text
*
      WRITE(11,400)
      WRITE(11,401) (I,C(I),I,E(I),I=1,N)
*
*     Read measured coherence function and deviation
*
      READ(12,*) (XL(I),YL(I),SD(I),I=1,M)
*
*     Call MINSQ for solution
*
      CALL MINSQ(M,N,F,C,E,IPRINT,NFUN,NW,W,COV,XSTEP)
*
*     Call function maker for calculation of estimated
*     values of coherence function
*
      CALL FM(M,N,C,YC,SUM)
*
*     Write the break frequency and the solution of F
*
      BF=C(2)
      WRITE(11,403) BF
      WRITE(11,500)
      WRITE(11,501) (I,C(I),I=1,N)
      WRITE(11,502) SUM
*
*     If the user does not want input and estimated values,
*     then stop execution, else continue writing output
```

```
*       files
*
        IF(IOPT.EQ.1) GO TO 50
        DO 40 I=1,M
        WRITE(11,*) XL(I),YL(I),YC(I),SD(I)
        WRITE(13,*) XL(I),YL(I),YC(I),SD(I)
 40     CONTINUE
*
*       Text formatting
*
 400    FORMAT(10X,'INITIAL GUESS AND ERROR PARAMETERS',/)
 401    FORMAT(10X,'X(',I2,')=',E12.6,3X,'E(',I2,')=',E12.6)
 403    FORMAT(/,10X,'BREAK FREQUENCY (Hz)=',E12.6)
 500    FORMAT(/,10X,'FITTED COEFFICIENTS:',/)
 501    FORMAT(10X,'X(',I2,')=',1X,E12.6)
 502    FORMAT(5X,'SUM OF SQUARED DIFFERENCES=',E12.6)
*
*       Cease execution of FITCOH.FOR
*
  50    STOP
        END
*                                                              *
****************************************************************
*                                                              *
*       This is the coherence function to be                   *
*       fit.                                                   *
*                                                              *
        SUBROUTINE FCN(M,N,F,C,IFLAG)
        DIMENSION F(M),C(N)
        COMMON/B1/YL(150),XL(150),SD(150)
        DO 10 I=1,M
        XI=XL(I)
        C2=C(2)
        C1=C(1)
*
*       The coherence function, with parameters in
*       HERTZ units.
*
        F(I)=((C1/(XI**2.0+C2**2.0+c1))**2.0-YL(I))/SD(I)
```

```
 10     CONTINUE
        RETURN
        END
*                                                              *
****************************************************************
*                                                              *
*       This calculates estimates of the coherence             *
*       function using the estimated parameters.               *
*                                                              *
        SUBROUTINE FM(M,N,C,YC,SUM)
        DIMENSION YC(M),C(N)
        COMMON/B1/YL(150),XL(150),SD(150)
        SUM=0.
        DO 30 I=1,M
        XI=XL(I)
        C2=C(2)
        YC(I)=(C(1)/(XI**2.0+C2**2.0+c(1)))**2.0
*
*       It also finds the sums of the squares between
*       the estimated and the input coherence function
*
        SUM=SUM+(YC(I)-YL(I))**2.0
 30     CONTINUE
        RETURN
        END
*                                                              *
****************************************************************
```

FITCOHIN.DAT

FITCOHIN.DAT contains information used in the solution of the coherence function, and is accessed by FITCOH.FOR.

```
28      2       0       10000   500
1.      0.5     0
700.    .1
6.50    0.001

This input file is for FITCOH.FOR, a fortran subroutine
linked with MINSQ which solves for two unknowns: F and
the break frequency.  It contains, in order,
M = 28 = the number of data pairs
N = 2 = the number of unknowns
IPRINT = 0 = intermediate print frequency
NFUN = 10000 = maximum iterations
NW = 500 = must be set >= N + M(N+1) + 3N(N+1)/2
COV = 1. = covariance print flag
XSTEP = 0.5 = search start variable <= 0.5
IOPT = 0 = Set to one if no estimates are to be printed
the estimates of values to be fit and tolerances.
C(1) = 700.     E(1) = .1
C(2) = 6.50     E(2) = 0.001
```

MEASCOH.DAT

MEASCOH.DAT contains triplets of (frequency in Hz, coherence function, standard deviation). The coherence function was found by averaging 10 'runs' of 128 RMS averaged time records.

```
 2.40 3.25E-01 6.09E-02
 2.80 3.30E-01 5.68E-02
 3.20 3.23E-01 6.76E-02
 3.60 3.00E-01 6.28E-02
 4.00 2.81E-01 6.83E-02
 4.40 2.70E-01 6.78E-02
 4.80 2.58E-01 3.62E-02
 5.20 2.55E-01 3.59E-02
 5.60 2.49E-01 4.42E-02
 6.00 2.44E-01 3.11E-02
 6.40 2.38E-01 3.73E-02
 6.80 2.15E-01 3.12E-02
 7.20 1.98E-01 3.24E-02
 7.60 1.86E-01 3.89E-02
 8.00 1.65E-01 5.10E-02
 8.40 1.44E-01 6.82E-02
 8.80 1.29E-01 7.94E-02
 9.20 1.13E-01 6.23E-02
 9.60 1.03E-01 3.99E-02
10.00 9.63E-02 2.67E-02
10.40 1.04E-01 2.33E-02
10.80 1.02E-01 3.65E-02
11.20 9.25E-02 3.15E-02
11.60 8.75E-02 3.20E-02
12.00 8.75E-02 2.49E-02
12.40 9.12E-02 3.09E-02
12.80 1.01E-01 4.42E-02
13.20 9.63E-02 4.27E-02
13.60 8.63E-02 3.70E-02
14.00 8.00E-02 3.12E-02
```

COHERTEX.DAT

This file contains the results. Note the variance of the parameters was displayed on the screen, rather than written into the results section. There is a modified version of MINSQ which was used with FIT1.FOR which did write the variance and the values of the difference between predicted and input coherence functions into the output file defined by FORTRAN unit 011, but it was not used to produce the results presented below.

```
      INITIAL GUESS AND ERROR PARAMETERS

      X( 1)=0.700000E+03   E( 1)=0.100000E+00
      X( 2)=0.650000E+01   E( 2)=0.100000E-02

      BREAK FREQUENCY (Hz)=0.711700E+01

      FITTED COEFFICIENTS:

      X( 1)= 0.794553E+02
      X( 2)= 0.711700E+01
 SUM OF SQUARED DIFFERENCES=0.591003E-02
2.40000       0.325000       0.341993       0.609000E-01
2.80000       0.330000       0.331758       0.568000E-01
3.20000       0.323000       0.320508       0.676000E-01
3.60000       0.300000       0.308437       0.628000E-01
4.00000       0.281000       0.295736       0.683000E-01
4.40000       0.270000       0.282589       0.678000E-01
4.80000       0.258000       0.269171       0.362000E-01
5.20000       0.255000       0.255643       0.359000E-01
5.60000       0.249000       0.242146       0.442000E-01
6.00000       0.244000       0.228807       0.311000E-01
6.40000       0.238000       0.215731       0.373000E-01
6.80000       0.215000       0.203006       0.312000E-01
7.20000       0.198000       0.190702       0.324000E-01
7.60000       0.186000       0.178873       0.389000E-01
8.00000       0.165000       0.167557       0.510000E-01
```

8.40000	0.144000	0.156781	0.682000E-01
8.80000	0.129000	0.146559	0.794000E-01
9.20000	0.113000	0.136896	0.623000E-01
9.60000	0.103000	0.127790	0.399000E-01
10.0000	0.963000E-01	0.119230	0.267000E-01
10.4000	0.104000	0.111203	0.233000E-01
10.8000	0.102000	0.103691	0.365000E-01
11.2000	0.925000E-01	0.966727E-01	0.315000E-01
11.6000	0.875000E-01	0.901252E-01	0.320000E-01
12.0000	0.875000E-01	0.840244E-01	0.249000E-01
12.4000	0.912000E-01	0.783458E-01	0.309000E-01
12.8000	0.101000	0.730647E-01	0.442000E-01
13.2000	0.963000E-01	0.681565E-01	0.427000E-01

COHERENC.DAT

COHERENC.DAT, not shown here, contains the (frequency, measured coherence function, fit coherence function, measured standard deviation) data quadruplets in a format useful for graphing.

Driver for Fitting Corrected APSD

The input and output files used with FIT1.FOR, namely IN.DAT, RES.DAT, INR.DAT, and OUT.DAT, correspond with the input and output files used by FIT-COH.FOR, so they will not be presented here.

FIT1.FOR

```
************************************************************
*                                                          *
      Program FIT1
*                                                          *
************************************************************
*                                                          *
*     Written by Uner Colak.   Modified and                *
*     Commented by David Roth.                             *
************************************************************
*
*     Declarations
*
      REAL F(150),C(10),E(10),W(1000),YC(150)
      COMMON/B1/YL(150),XL(150),SD(150)
*
*     File opening
*
      OPEN(10,FILE='IN.DAT',STATUS='OLD')
      OPEN(11,FILE='RES.DAT',STATUS='UNKNOWN')
      OPEN(12,FILE='INR.DAT',STATUS='OLD')
      OPEN(13,FILE='OUT.DAT',STATUS='UNKNOWN')
*
*     Read initial input
*
      READ(10,*) M,N,IPRINT,NFUN,NW,COV,XSTEP,IOPT
      READ(10,*) (C(I),E(I),I=1,N)
*
*     Write results header
*
```

```
      WRITE(11,400)
      WRITE(11,401) (I,C(I),I,E(I),I=1,N)
      READ(12,*) (XL(I),YL(I),SD(I),I=1,M)
*
*     Solve for the parameters
*
      CALL MINSQ(M,N,F,C,E,IPRINT,NFUN,NW,W,COV,XSTEP)
*
*     Calculate the estimated APSD
*
      CALL FM(M,N,C,YC,SUM)
*
*     Write results body
*
      BF=C(2)
      WRITE(11,403) BF
      WRITE(11,500)
      WRITE(11,501) (I,C(I),I=1,N)
      WRITE(11,502) SUM
      IF(IOPT.EQ.1) GO TO 50
      DO 40 I=1,M
      WRITE(11,*) XL(I),YL(I),YC(I),SD(I)
      WRITE(13,*) XL(I),YL(I),YC(I),SD(I)
 40   CONTINUE
 400  FORMAT(10X,'INITIAL GUESS AND ERROR PARAMETERS',/)
 401  FORMAT(10X,'X(',I2,')=',E12.6,3X,'E(',I2,')=',E12.6)
 403  FORMAT(/,10X,'BREAK FREQUENCY (Hz)=',E12.6)
 500  FORMAT(/,10X,'FITTED COEFFICIENTS:',/)
 501  FORMAT(10X,'X(',I2,')=',1X,E12.6)
 502  FORMAT(5X,'SUM OF SQUARED DIFFERENCES=',E12.6)
  50  STOP
      END
*                                                     *
*******************************************************
*                                                     *
*     This subroutine returns the difference between  *
*     the current guess and the actual value and      *
*     gives it a weight of one standard deviation.    *
*                                                     *
```

```
      SUBROUTINE FCN(M,N,F,C,IFLAG)
      DIMENSION F(M),C(N)
      COMMON/B1/YL(150),XL(150),SD(150)
      DO 10 I=1,M
      XI=XL(I)
      C1=C(1)
      C2=C(2)
      C3=C(3)
*
*     The current difference function
*
      F(I)=(C1/(XI**2.+C2**2.)+C3-YL(I))/SD(I)
 10   CONTINUE
      RETURN
      END
*************************************************************
*                                                           *
*     This subroutine calculates values for the APSD        *
*     using the estimated parameters.                       *
*                                                           *
      SUBROUTINE FM(M,N,C,YC,SUM)
      DIMENSION YC(M),C(N)
      COMMON/B1/YL(150),XL(150),SD(150)
      SUM=0.
      DO 30 I=1,M
      XI=XL(I)
      C2=C(2)
      YC(I)=C(1)/(XI**2+C2**2)+C(3)
      SUM=SUM+(YC(I)-YL(I))**2
 30   CONTINUE
      RETURN
      END
*                                                           *
*************************************************************
```

APPENDIX D. SOURCE CODE FOR *LMK.C*

This is the code which read the marker position and amplitude or coherence function. The main steps are all commented in the actual program; the code is executed as shown between the curly braces after the word *main*. All code after this is for the functions called by *main* with the exception of *error_handler*, which is used by the functions themselves.

Full understanding of the operation of this program, and the source codes *readrms.C* and *makenorm.C*, requires knowledge of C, in the use of the HP-IB interface, and in the use of the c_hpib.h include file. In C, the routines to access hardware are listed in *include* files.

```
/* This program will read in the MARKER'S DISPLAYED TRACE */
#define LINT_ARGS       0 /*  Permits limited type checking*/
#include <stdio.h>      /* Include standard i/o handler */
#include <c_hpib.h> /* Include HP-IB access handler */
#include <string.h> /* Include string handler */
#include <conio.h>  /* Include console handler */

/* Define constants for compiler */
#define ISC     7L
#define SPEC 711L
#define TCF 128L   /* TCF is the total number of channels read */

/* Declare Variables */
long    isc, spec;
long    dsp[256];
```

```
short   error, span;
float   freq[TCF];
float   coher[TCF];

/* Beginning of main program */
main()
{
        telluser(); /* Tell user what program does and what to
                                        set on HP-3582A */
        press();    /* Wait for a key press */
        read_em();  /* Read all values of trace */
        local();    /* Return HP-3582A to manual control */
        store_it(); /* Store values read from trace */
}
/* End of main program */

/*********************************************************/
/*      Functions follow.  The purpose of each function is
        given above */
/*********************************************************/

telluser()
{
printf("\f\n\nThis program will read the MARKER'S TRACE.");
printf("\n\nYou must select the TRACE and adjust the AMPLITUDE");
printf("\n\nREFERENCE LEVEL to display everything needed.");
printf("\n\nYou must also select the appropriate SCALE.");
printf("\n\nAfter collecting data:");
printf("\n     1) Set marker to ON and divided by bandwidth");
printf("\n     2) Place marker ");
printf("on the desired trace\n     3) Choose the SCALE");
printf("\n     4) Adjust the AMPLITUDE REFERENCE LEVEL to
display");
printf(" as much of the trace\n        as possible");
printf("\n     5) Press return.");
}

/*********************************************************/
```

```
press()
{
int     ch;
        while(kbhit()== 0);
        ch = getch();
}

/*******************************************************/
/* The following function is used to verify proper HP-IB
   protocol */

error_handler (error, routine)
int             error;
char    *routine;
{
        char    *estring, ch;

        if      (error != NOERR)
                {
                printf("\nError in call to %s \n",routine);
                printf(
               "        Error = %d :%s \n",error, errstr(error));
                printf("Press <RETURN> to continue....\n");
                scanf("%c", &ch);
                }
}
/*******************************************************/
read_em()
{
        extern float freq[TCF];
        extern float coher[TCF];
        float temp[2];
        char buffer[20];
        long length;
        int number;
        short int mp;
        number = 2;

        printf("\n\nNow reading frequency/amplitude pairs....");
```

```
        for (mp = 0; mp<TCF; mp+=1){

/* Position marker and request values */
        length = sprintf(buffer,"run mp%d hlt lmk",mp);
        error = iooutputs(SPEC,buffer,length);
        error_handler(error,"IOOUTPUTS");

/* Fetch the values from the HP-IB bus.  Store them in temp[] */
        error = ioentera(SPEC,temp,&number);
        error_handler(error,"IOENTERA");
        coher[mp] = temp[0];
        freq[mp] = temp[1];
        }
}

/*********************************************************/

local()
{
        int error;
                error = iolocal(SPEC);
                error_handler(error,"IOABORT");
                printf(
                "\n\nLocal control returned to Spec.\n\n");
                error = ioabort(ISC);
                error_handler(error,"IOABORT");
                printf("\nAll interface activity aborted.\n");
        }

/*********************************************************/

store_it()
{
        int w = 0;
        extern float freq[TCF],coher[TCF];
        char    filename[12];
        FILE  *stream;
```

```
/* Ask user for unused filename */
        do {
                printf(
               "\n\nEnter filename to write data pairs to:  ");
                scanf("%s",filename);
                stream = fopen(filename, "r");
                if (stream != NULL) printf(
                "\n\nThat filename is in use.  Try another.");
        } while (stream != NULL);

/* Write values to disk */
        stream = fopen(filename, "w");
        for (w=0; w<TCF; w+=1){
                fprintf(stream,"%f, %f \n",freq[w],coher[w]);
                }
}
/*******************************************************/
```

APPENDIX E. SOURCE CODE FOR *READRMS.C*

This code directly accesses the spectrum analyzer memory to read the values of the APSD.

```
/*      This program will read in the POWER SPECTRAL DENSITY
*/
/*      and store it in a user specified file.
*/

#define LINT_ARGS       0       /*  Permits limited type
checking*/
#define ISC     7L
#define SPEC 711L

#include <stdio.h>    /* Include necessary io, HP-IB, string, */
#include <c_hpib.h>   /* console, and math handlers */
#include <string.h>
#include <conio.h>
#include <math.h>

                          /* Declarations */
long    isc, spec;
short   mantissa[256], phase[256];
short   error, span;
long    exponent[256];
float   mant[256];
float   amp[256];
float   dfq;

main()
```

```
{
        press();    /* Waits for user to press a key  */
        mem_read(); /* Reads memory from HP-3582A */
        make_amps();/* Converts memory to decimal */
        correct();  /* Corrects for internal filters */
        local();    /* Returns HP-3582A to local control */
        store_it(); /* Stores APSD values */
}

/*****************************************************************/
press()
{
int     ch;
        printf("\nPress return when done collecting
data...\n\n");
        while(kbhit()== 0);
        ch = getch();
}
/*****************************************************************/
/* This routine is accessed by other routines for error
checking*/

error_handler (error, routine)

int             error;
char    *routine;

{
        char    *estring, ch;
        if      (error != NOERR)
                {
        printf("\nError in call to %s \n", routine);
        printf("        Error = %d :%s \n", error,
errstr(error));
        printf("Press <RETURN> to continue....\n");
                        scanf("%c", &ch);
                }
}
```

```
/*******************************************************/
mem_read()
{
        char unsigned bytes[130];
        char    *info, buffer[20];
        long    address = 076000, w, x, y, z, length, wmax;
        extern  short mantissa[256], phase[256];
        extern  float mant[256];
        extern  long  exponent[256];
        short   flag = 0, max, rmsb, rlsb, imsb, ilsb;
        char    match = 10;
        float   rr,ii, maxnorm = 0.0;

     /* Set end of transmission flag off */
        error = iomatch(7L, match, flag);

     /* Halt analyzer CPU for faster data x-fer */
        error = iooutputs(711L, "hlt", 3);
        error_handler (error, "IOOUTPUTS");

     /* Read octal memory as string called bytes */
                for (x=0; x<=7; ++x, address += 0100) {
                        y = 0;
                        y = sprintf(buffer,
                              "lfm,%o,128",address);
                        printf("\n\n%s",buffer);

                        length = strlen(buffer);
                        error = iooutputs(711L, buffer, length);
                        error_handler (error, "IOOUTPUTS");

                        max = 128;
                        error = ioenters(711L, bytes, &max);
                        error_handler (error, "IOENTERS");
                        printf("\n Bytes read: %d",max);

                        for (z = 0, w = 32 * x; z <=127; z +=4,
                              ++w) {
                /* Find APSD mantissa and phase */
```

```
                rmsb = bytes[z];
                rlsb = bytes[(z+1)];
                imsb = bytes[(z+2)];
                ilsb = bytes[(z+3)];

                mantissa[w] = ((rmsb << 8) |
                                rlsb);

        /* Phase, though found, is not kept */
              phase[w] = ((imsb << 8) | ilsb);

                rr = mantissa[w];
                ii = phase[w];
                mant[w] = rr;
         }
 }
 printf(
"\n\f\nMantissa read.  Now reading exponent.");

 for (x=0, address = 077000; x<=1; ++x, address +=
          0100) {
        y = 0;
        y = sprintf(buffer,
         "lfm,%o,128",address);
        printf("\n\n%s",buffer);

        length = strlen(buffer);
        error = iooutputs(711L, buffer, length);
        error_handler (error, "IOOUTPUTS");

        max = 128;
        error = ioenters(711L, bytes, &max);
        error_handler (error, "IOENTERS");
        printf("\n Bytes read: %d",max);

        for (z = 0, w = 128*x ; z<=127; z +=1,
                  w+=1) {

                rmsb = bytes[z];
```

```
                                exponent[w] = rmsb - 15L;

                                }
                }
}
/*****************************************************************/
make_amps()  /* calculates the non-corrected APSD */
{

        extern float mant[256], amp[256];
        extern long exponent[256];
        float   a, b, c, d[256];
        short   x, y, z,span;
        double aa,bb,cc;
        double two = 2.0;

        printf("\n\f\nArrived at make_amps()");

        printf(
     "\n\nPlease enter the number of averages completed. ");
        scanf("%d",&z);
        printf("\n You chose %d averages",z);
        cc = z;

        for(x = 0;  x < 256; x+=1)
                {
                        aa = exponent[x];
                        bb = pow(two, aa);
                        amp[x] = mant[x] * bb / cc;
                }
}
/*****************************************************************/
correct()  /* Corrects the APSD for filter effects */
{
        extern float amp[256];
        short x, span;
        FILE *stream;
        char *filename;
        extern  float mant[256];
```

```
        long dsp[256];
        short w;
        extern  float dfq;

        do      {
        printf(
"\n\f\nPlease enter the span of the spectrum analyzer (1-14): ");
        x = scanf("%d",&span);
        if ((span<= 0) || (span >14))
                        {
                printf(
       "\nThe span must be from 1 to 14.  See pullout panel ");
                printf("below analyzer.\n");
     /*  Clear stdin if the read failed*/
                        if (x == 0) scanf("%*s");
                        }
                }
        while ((span<= 0) || (span >14));

        if (span == 1) dfq = 0.004;
        if (span == 2) dfq = 0.01;
        if (span == 3) dfq = 0.02;
        if (span == 4) dfq = 0.04;
        if (span == 5) dfq = 0.1;
        if (span == 6) dfq = 0.2;
        if (span == 7) dfq = 0.4;
        if (span == 8) dfq = 1.0;
        if (span == 9) dfq = 2.0;
        if (span == 10) dfq = 4.0;
        if (span == 11) dfq = 10.0;
        if (span == 12) dfq = 20.0;
        if (span == 13) dfq = 40.0;
        if (span == 14) dfq = 100.0;

/* read back in appropriate norm matrix*/
        filename = "NORM00.DAT";
        sprintf(filename + 4,"%02.2d",span);
        filename[6] = '.';
        filename[7] = 'D';
```

```
        filename[8] = 'A';
        filename[9] = 'T';
        printf("\n\nfilename = %s",filename);

        stream = fopen(filename,"r");

        for (w=0; w<256; w+=1){
                fscanf(stream,"%*d %*1s %f %*1s %ld\n",&mant[w],
&dsp[w]);
                amp[w] /= mant[w];
                                                }

}
/*****************************************************************/
local()
{

        int     error;
        error = iolocal(711L);
        printf("\n\nLocal control returned to Spec.\n\n");
        error = ioabort(7L);
        printf("\nAll interface activity aborted.\n");

}
/*****************************************************************/
store_it()
{
        int w = 0;
        extern float amp[256];
        char    filename[12];
        FILE  *stream;
        extern float dfq;
        float hertz;

        do
                {
                        printf(
                "\n\nEnter filename to write amp to:  ");
                        scanf("%s",filename);
```

```
                        stream = fopen(filename, "r");
                        if (stream != NULL) printf(
                "\n\nThat filename is in use.  Try another.");
                }
        while (stream != NULL);

        stream = fopen(filename, "w");

        for (w=0; w<256; w+=1)
                {
                hertz = w*dfq;
                fprintf(stream,"%f, %f\n",hertz,amp[w]);
                }

}
/*************************************************************/
```

APPENDIX F. SOURCE CODE FOR *MAKENORM.C*

This code was used to calculate the internal filter response.

```
/*      This program will create the norm array and store it in
*/
/*      NORMxx.DAT where xx is the span byte (1-14).
*/

#define LINT_ARGS       0       /*  Permits limited type checking
*/
#include <stdio.h>
#include <c_hpib.h>
#include <string.h>
#include <conio.h>

#define ISC     7L
#define SPEC 711L

long    isc, spec;
short   re[256], im[256];
long    dsp[256];
short   error, span;
float   norm[256];


main()

{
        getspan();  /* Ask user for frequency span to use */
        setspan();  /* Set frequency span */
        press();    /* Wait for key press */
```

```
                /* start a collection */
        error = iooutputs(711L, "ar",2);
        error_handler(error, "IOOUTPUTS");
        press();    /* Wait for key press */
        mem_read(); /* Read memory, calc norm matrix */
        store_it(); /* Store norm matrix */
}

/*****************************************************************/

press()
{
int     ch;
        printf(
"\nPress return when done collecting data...\n\n");
        while(kbhit()== 0);
        ch = getch();
}

/*****************************************************************/

error_handler (error, routine)

int             error;
char    *routine;

{
        char    *estring, ch;

        if      (error != NOERR)
                {
                printf("\nError in call to %s \n",routine);
                printf("        Error = %d :%s \n",error,
errstr(error));
                printf("Press <RETURN> to continue....\n");
                        scanf("%c", &ch);
                }
}
/*****************************************************************/
```

```
getspan()
{
        int x;
        do      {
     printf(
"\n\f\nPlease enter the span of the spectrum analyzer (1-14): ");
        x = scanf("%d",&span);
        if ((span<= 0) || (span >14)) {
        printf(
"\nThe span must be from 1 to 14.  See pullout panel ");
                printf("below analyzer.\n");
                if (x == 0) scanf("%*s");/*  Clear stdin if the
read failed*/
                        }
                }
        while ((span<= 0) || (span >14));
}

/*****************************************************************/

setspan()

{
        extern  short span;
        int radix = 10;
        long length;
        char aspan[20];

                aspan[0] = 's';
                aspan[1] = 'p';
        printf("External variable span = %d",span);
        itoa(span, aspan+2, radix);


        printf("\n\naspan = %s",aspan);

        length = strlen(aspan);
        error = iooutputs(711L, aspan, length);
        error_handler (error, "IOOUTPUTS");
```

```
        error = iooutputs(711L, "ps3 md2 re ar", 13);
        error_handler (error, "IOOUTPUTS");


}
/*****************************************************************/
mem_read()
{
        char unsigned bytes[130];
        char    *info, buffer[20];
        long    address = 076000, w, x, y, z, length, wmax;
        extern  short re[256], im[256];
        extern  float norm[256];
        extern  long dsp[256];
        short   flag = 0, max, rmsb, rlsb, imsb, ilsb;
        char    match = 10;
        float   rr,ii, maxnorm = 0.0;
        short   scum[256];
     /* Set end of read flag off, halt analyzer */
        error = iomatch(7L, match, flag);

        error = iooutputs(711L, "hlt", 3);
        error_handler (error, "IOOUTPUTS");

 /* Read memory into strings */
                for (x=0; x<=7; ++x, address += 0100) {
                y = 0;
                y = sprintf(buffer, "lfm,%o,128",address);
                        printf("\n\n%s",buffer);

                        length = strlen(buffer);
                        error = iooutputs(711L, buffer, length);
                        error_handler (error, "IOOUTPUTS");

                        max = 128;
                        error = ioenters(711L, bytes, &max);
                        error_handler (error, "IOENTERS");
```

```
                    printf("\n Bytes read: %d",max);
                    for (z = 0, w = 32 * x; z <=127; z +=4,
                              ++w) {

                            rmsb = bytes[z];
                            rlsb = bytes[(z+1)];
                            imsb = bytes[(z+2)];
                            ilsb = bytes[(z+3)];

                            re[w] = ((rmsb << 8) | rlsb);
                            im[w] = ((imsb << 8) | ilsb);

                            rr = re[w];
                            ii = im[w];
                            norm[w] = rr * rr + ii * ii;
                            }
            }

            printf(
        "\n\f\nRAM memory read.  Now reading display memory");


/*          Note:  This memory is backward.  Bin 255 is at
74400, while
            bin 0 is at 74777.  There are 512 bytes for 256
extra short
            unsigned numbers.
*/
/* Read displayed screen */

            for (x=0, address = 074400; x<=3; ++x, address +=
0100) {
                    y = 0;
                    y = sprintf(buffer,
"lfm,%o,128",address);
                    printf("\n\n%s",buffer);

                    length = strlen(buffer);
                    error = iooutputs(711L, buffer, length);
```

```
                        error_handler (error, "IOOUTPUTS");

                        max = 128;
                        error = ioenters(711L, bytes, &max);
                        error_handler (error, "IOENTERS");
                        printf("\n Bytes read: %d",max);

                        imsb = 1023;
                        for (z = 0, w = (255 - (64 * x));
                         z <=127; z +=2, --w) {
                                rmsb = bytes[z];
                                rlsb = bytes[(z+1)];
                                dsp[w] = (((rmsb << 8) | rlsb) &
imsb);
     /* Calculated normalizing matrix */
                                norm[w] = norm[w]/dsp[w]/dsp[w];
                                if (norm[w]> maxnorm ) {
                                maxnorm = norm[w];
                                wmax = w;
                                                }
                                }
                }
                for (w = 0; w < 256 ; norm[w++] /= maxnorm);
        printf("\n\n ...and the max occurred at %ld and was %f",
        wmax, norm[wmax]);
}
/*****************************************************************/
store_it()
{
        int w = 0;
        extern float norm[256];
        char    filename[12];
        FILE  *stream;

        printf("\n\nEnter filename to write norm to:  ");
        scanf("%s",filename);
        stream = fopen(filename, "w");
        for (w=0; w<256; w+=1) fprintf(stream,"%d, %f, %ld\n",
                        w,norm[w], dsp[w]);
```

```
}
/*************************************************************/
```

ACKNOWLEDGEMENTS

I wish to express my deepest gratitude to Dr. Robert E. Williams and Dr. Richard A. Hendrickson; this work could not have been done without their guidance, knowledge, and inspiration.

I also wish to thank Dr. Uner Colak for his invaluable assistance in curve-fitting, AutoCad, and data collection. I must also thank David C. Hamm for operating the reactor and helping with the electronics. Thanks to Brad Michael, a friend for many years, for being my personal computer and math consultant

Thanks go to my friends and fellow graduate students – Wm. Mark Nutt, Ramin Mikaili, and Chris Gersey for providing examples (both good and bad) of how to earn a Master's in under ten years.

I would also like to thank Dr. Richard Danofsky and Dr. Herbert David for being on my committee and sharing their wisdom with me.

Thanks to the Power Affiliates program for its financial support.

A hearty thanks to Dr. Bernard I. Spinrad for the occasional motivating talk. Thanks to all the professors in the Department of Nuclear Engineering for teaching me these past years. Also, thanks to Ruth Anderson for her friendly help.

Most important, I want to thank my parents, Ron and Sue Roth, for their financial and emotional support during six years of college. And I must thank my sisters,

Julie and Cheryl, and my grandparents, Ruby and Jack (Nana and Grandpa) Hursh and Bernadine Roth (Grandma B.), and all my aunts, uncles, and cousins for lots of love and encouragement.