Measurement of the Parity Violating Gamma Ray Asymmetry in Polarized Neutron Capture on ^{35}Cl, ^{113}Cd, and ^{139}La

by

Christopher S. Blessinger

Submitted to the faculty of the University Graduate School
in partial fulfillment of the requirements
for the degree
Doctor of Philosophy
in the Department of Physics
Indiana University
October 2001

Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

October 2001

Snow

Baxter

Bowman

Nann

© 2001
Christopher S. Blessinger
All Rights Reserved

ACKNOWLEDGEMENTS

First I would like to thank my Professor W. M. Snow for his guidance and support as my advisor. In addition, I would like to thank Hermann Nann, David Baxter, and David Bowman for serving on my committee.

I would also like to thank the low energy neutron team in P-23 at LANL: Greg Mitchell, Seppo Penttila, Todd Smith, and Scott Wilburn for their help and advice while I worked at LANSCE.

This work was supported by the United States Department of Energy and the National Science Foundation.

ABSTRACT

Measurement of the Parity Violating Gamma Ray Asymmetry in Polarized Neutron Capture on ^{35}Cl, ^{113}Cd, and ^{139}La

by
Christopher S. Blessinger

Of the four fundamental forces, only the weak force fails to conserve parity. When polarized neutrons are captured by a nucleus, the momentum of the emitted gamma rays is correlated with the spin of the neutron. This asymmetry offers an opportunity to directly measure the strength of the weak interaction within the nucleus, which otherwise is obscured by the much larger strong force. The gamma ray distribution is of the form $1+A\cos\theta$, where A is the coefficient of interest, θ is the angle between the spin of the neutron and the momentum of the emitted gamma ray, and $A\cos\theta$ together is the parity violating part. The value of A is different for each nucleus and is dependent on the nuclear structure as well as the weak coupling. This thesis presents the results of the measurement of A for ^{35}Cl, ^{113}Cd, and ^{139}La. The relationship to theory will be discussed.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . . . **iv**

LIST OF FIGURES . . . . . **xi**

LIST OF TABLES . . . . . **xix**

LIST OF APPENDICES . . . . . **xxii**

CHAPTER

I. Theory and Motivation 1

1.1 Introduction 1
1.1.1 Definition of Parity 1
1.1.2 The Discovery of Parity Non-conservation in the Weak Interaction 3
1.1.3 Weak Interaction at the Lepton/Quark Level 4
1.1.4 The Weak Contribution to the N-N Interaction . . . 5
1.1.5 The Weak Interaction in Medium and Heavy Nuclei 6
1.1.6 Current Status of the Theoretical Predictions for N-N Weak Couplings 7
1.2 The NPDGamma Experiment 9
1.2.1 The Fundamental Importance of NPDGamma . . . 9
1.2.2 Apparatus of the NPDGamma experiment 12

1.3 The Parity Violating Gamma Ray Asymmetries in Polarized Neutron Capture on ^{35}Cl, ^{113}Cd, and ^{139}La 12
1.3.1 Motivation 12
1.3.2 Apparatus 14
1.3.3 NPDGamma Detector Alignment 14

II. Experimental Overview 16

2.1 The Neutron Beam 17
2.2 The Magnetic Holding Field 20
2.2.1 Theory of the Neutron Spin Motion in a Magnetic Field 20
2.2.2 Construction of the Magnetic Field 21
2.3 The ^{3}He Spin Filter 22
2.3.1 The Effect on the Neutron Beam 22
2.3.2 Polarizing ^{3}He through Spin Exchange with Optically Pumped Rb 23
2.3.3 Measurement of the ^{3}He Polarization by Adiabatic Fast Passage Nuclear Magnetic Resonance 26
2.4 The Radio Frequency Spin Flipper 28
2.4.1 Theory of the NMR Spin Flip 29
2.4.2 Construction and Tuning of the RFSF 31
2.4.3 Efficiency of the RFSF 33
2.5 The Targets 36
2.5.1 The Chlorine Target 36
2.5.2 The Lanthanum Target 36
2.5.3 The Cadmium Target 40
2.6 The Detector Assembly 40

2.6.1 The ^{6}Li Plastic Shield 40
2.6.2 The CsI Array 41
2.6.3 The Photodiodes and Electronics 45
2.7 The Supermirror Polarization Analyzer 48
2.8 The Neutron Backmonitor 51
2.8.1 Requirements 51
2.8.2 Gas Mixture Design 51
2.8.3 Mechanical Design 53
2.8.4 The Time of Flight Neutron Spectrum and the Aluminum Bragg Edges 58
2.8.5 γ Insensitivity 61
2.8.6 The High Voltage Plate Bias 63
2.8.7 ^{3}He Attenuation 63
2.8.8 Noise and Linearity 65
2.9 The Data Acquisition 67
2.9.1 The System Requirements 67
2.9.2 The Data File Format 69
2.9.3 The Triggering System 70
2.9.4 The Computers 71
2.9.5 The VME Hardware 74
2.9.6 The Software Components 75
2.9.7 The Signal List 78

III. Analysis 81

3.1 The Asymmetry, A_γ 81
3.2 The Detector Signals 82
3.2.1 Balancing the Detector Gains 83

3.2.2 Eliminating the Detector Offsets 83
3.3 Establishing cuts on the Data 87
3.3.1 The Cut on the Eight-step Sequence 87
3.3.2 The Cut on the Magnitude of the Detector Sum . . 87
3.3.3 The Cut on the Change in Detector Sum within an Eight-step Sequence 89
3.4 Calculation of the Correction Factors 94
3.4.1 Polarization, $P(E_n)$ 94
3.4.2 Spin Flipper Efficiency, $\epsilon_R(E_n)$ 94
3.4.3 Detector Energy Weighted Cosine, G 95
3.4.4 Calculation of the Depolarization in the Target, T . 101
3.5 Calculation of the Asymmetry 102
3.6 Summary 104

IV. Control of Systematic Errors 105

4.1 Nature of Systematic Errors 105
4.2 The Eight-step Sequence 105
4.3 The Supermirror Catastrophe 106
4.4 Cross-correlation of the Asymmetries 108
4.5 Measurement of Systematics 110
4.5.1 False Asymmetries 111
4.5.2 Counting Statistics 111
4.6 Summary 116

V. Conclusions 117

5.1 Science Results 117

5.2 Connecting Experiment to Theory 121
5.2.1 Origin of the Parity Violation 121
5.2.2 The Specific Nature of the Target Nuclei 123
5.2.3 Mean-square Estimate of the Magnitude of the Asymmetry 124
5.2.4 Detailed Analysis of ^{35}Cl 128
5.2.5 Agreement of Experimental and Theoretical Results 128
5.3 The future of the NPDGamma experiment 129
5.3.1 The Neutron Beam and Flight Path 130
5.3.2 The Polarized ^{3}He Filter 130
5.3.3 The Magnetic Guide Field 130
5.3.4 Radio Frequency Spin Flipper 130
5.3.5 The Target 131
5.3.6 The Detector Array 132
5.3.7 The Back Neutron Monitor 132
5.3.8 The Data Acquisition System 134

APPENDIX **135**

BIBLIOGRAPHY **194**

LIST OF FIGURES

Figure

1.1 Muon decay, a leptonic weak interaction 4

1.2 Beta decay, a semi-leptonic weak interaction 5

1.3 Hadronic weak interaction 6

1.4 Weak coupling constants H^1_π and H^0_ρ extracted from ^{18}F (light band), ^{205}Tl (medium band), and ^{133}Cs (dark band) experiments. Also shown are the DDH "best" values (square) and "reasonable range" (box) 8

1.5 The reaction studied in this thesis. 10

1.6 Experimental setup for the NPDGamma experiment 13

1.7 Experimental setup, beam travels left to right 15

2.1 Top down view of the experimental apparatus. The gray area is the experimental cave. 16

2.2 An ariel photograph of LANSCE 17

2.3 Schematic diagram of the accelerator, the spallation target is at the Lujan center,in the lower left of the diagram 18

2.4 Floorplan of flight paths at the Lujan center, we used flight path 11A for these measurements 18

2.5 Neutron spectrum 19

2.6 Neutron transmission of the ^{3}He cell and the polarization of the filtered neutron beam 24

2.7 Neutron beam polarization, dots represent measured polarization, the line is the theoretical prediction from ^{3}He transmission, error on this measurement is $\approx 1\%$ of full scale. 24

2.8 This is the double ^{3}He cell. The optical pumping cell is in front. The box surrounding it is the oven; laser light enters through the window in the top. The filter cell is on the far side; small pickup coils are located on either side. The NMR coils surrounding the assembly are partially visible. 25

2.9 Rb energy levels and transitions 25

2.10 ^{3}He NMR signal over the experiment, 40meV is proportional to a polarization of 26.5% 28

2.11 Photo of the Radio Frequency Spin Flipper 31

2.12 Schematic of the Radio Frequency Spin Flipper, it consist of a RF coil surrounded by an Al can 32

2.13 Transmission of the spin flipper through the supermirror analyzer . 33

2.14 Spin Flipper Efficiency 35

2.15 Schematic of the CCl_4 target 38

2.16 Photo of the CCl_4 target 39

2.17 Photo of the La target 40

2.18 CsI detector array schematic 42

2.19 Time response of the CsI crystal, measured with a sealed ^{137}Cs gamma source and a photomultiplier 43

2.20 Mechanism of scintillation in inorganic crystals 44

2.21 Schematic of the scintillator/photodiode/amplifier assembly 45

2.22 Hamamatsu ZC875-R2046PT photodiode which was used in this experiment 46

2.23 Left: the entire detector assembly, right: the CsI-Tl scintillator . . . 47

2.24 Bench measurement of the photodiode preamp noise 47

2.25 Sample photo-diode pre-amp signal for the Cl target, this is the gamma intensity converted to voltage as a function of TOF 48

2.26 The beam travels from right to left, the supermirror analyzer is in the center, the H/^{3}He ion chamber is to the left, the end of the holding field coils is on the right, the data acquisition system is visible in the alcove behind the supermirror 50

2.27 Closeup of the front of the supermirror showing its arrangement of stacked, magnetized glass plates, the data acquisition can be seen in the background 50

2.28 The beam path runs from left to right, the back monitor is on the right, the supermirror is on the left 52

2.29 Backmonitor assembly 54

2.30 Collection plate 55

2.31 High voltage electrode 55

2.32 Half voltage guard ring 56

2.33 Preamp 56

2.34 Current mode signal from all 10 plates in the ion chamber as a function of neutron TOF 58

2.35 Al Bragg edges 60

2.36 Signal from the first plate when a ^{6}Li absorber intersects the beam . 62

2.37 Peak signal in Plate 1 as a function of bias voltage 64

2.38 Evolution of the spectrum from plates 2 to 5. The red lines are signals from the ion chamber. The black circles are the attenuation calculated from equation 2.32. 66

2.39 DAQ hardware diagram: Gray boxes represent hardware; Those with red borders are NIM modules; blue borders are VME; green borders are PCs. Brown boxes represent software; those with blue borders are Perl code; red borders are C code. Black connections are single pin BNC/lemo; yellow are two-pin lemo; green is ethernet; the blue hashed line is the VME bus. 68

2.40 Run file format 70

2.41 DAQ trigger flowchart 72

2.42 Schematic diagram of the spin sequencer 73

2.43 DAQ timing diagram 73

2.44 Data acquisition code flowchart: Brown is the daqgui_II program; blue is the b_II.c program; green is npdg.c. Black arrows represent progression within a program; blue arrows are communications through socket connections; red arrows are communications through the Unix shell. 76

2.45 DAQ socket diagram 77

3.1 Sample detector signals 84

3.2 Total cross-section of natural Cl, courtesy of ENDF 85

3.3 Total cross-section of natural La, courtesy of ENDF 85

3.4 Total cross-section of natural Cd, courtesy of ENDF 85

3.5 Cd detector sum cut 91

3.6 Cl detector sum cut 92

3.7 La detector sum cut 93

3.8 neutron beam polarization, dots represent measured polarization, the line is the theoretical prediction from ^{3}He transmission 95

3.9 Spin Flipper Efficiency 96

3.10 CsI efficiency, taken from NIST Physical Reference Data, $\frac{\mu}{\rho} = x^{-1} ln(\frac{I_o}{I})$ 99

3.11 Value of the geometry coefficient, G, for an ideal detector of fixed angular width and a point source of γ-rays 100

4.1 Attenuation of the beam stop before the supermirror 108

4.2 Scatter plot of the false asymmetries 113

4.3 Comparison of neutron counting statistics and the electronic noise. . 115

5.1 Comparison of the raw asymmetry values with the measured false asymmetry of the electronic noise. The points represent two dimensional gaussian curves, showing the parity violating up/down asymmetry and the parity conserving left/right simultaneously. The error bars represent one standard deviation of the mean. (This graph does include the sign correction for the polarization only) 120

5.2 Amplitudes used in the theoretical model: (a) A neutron is captured in a s resonance and decays or a direct capture occurs. (b) A neutron is captured in a p resonance and decays or a direct capture occurs. (c) A neutron is captured in a s resonance and the weak force mixes the s and p allowing a decay from the p state. (d) A neutron is captured in a p resonance and the weak force mixes the s and p allowing a decay from the s state. 122

5.3 Liquid Hydrogen Target for NPDGamma, the neutron beam enters on the right 133

5.4 Detector array for NPDGamma 133

LIST OF TABLES

Table

1.1 Vector operators under parity inversion 3

1.2 Estimated weak meson-nucleon couplings from reference [6] 9

1.3 Summary of past measurements of A_γ in the nuclei of interest . . . 14

2.1 NMR signal for each target and run block($\star$ Not every run in a data block is a good asymmetry run) 29

2.2 Table of selected isotopes, isotopes of interest are boldfaced, the m superscript designates isomers 37

2.3 Target material densities 38

2.4 Triggering system components 72

2.5 Daq VME boards 74

2.6 Daq software 78

2.7 Signal list 80

3.1 Detector offsets 88

3.2 Monte Carlo values of geometry correction factor, G, for each target 97

3.3 Cross-sections[22] and Depolarization, T, for each target 102

3.4 These are scale factors which shift the value of the asymmetry. All apply before the scaling. 104

3.5 These are errors which change the size of the asymmetry. "Scale" is the value by which the raw asymmetry is multiplied. The error is given as percentage of the final asymmetry. 104

4.1 Asymmetry cross-correlation 110

4.2 Data by run number 112

4.3 List of false asymmetry measurements 112

4.4 These are scale factors which shift the value of the asymmetry. All apply before the scaling. 116

4.5 These are errors which change the size of the asymmetry. "Scale" is the value by which the raw asymmetry is multiplied. The error is given as percentage of the final asymmetry. 116

5.1 These are scale factors which shift the value of the asymmetry. All apply before the scaling. 117

5.2 These are errors which change the size of the asymmetry. "Scale" is the value by which the raw asymmetry is multiplied. The error is given as percentage of the final asymmetry. 118

5.3 raw asymmetries: these are differences between the gamma detectors before any corrections are made 118

5.4 The final results 119

5.5 Summary of past measurements of A_γ in the nuclei of interest . . . 119

5.6 $\langle A_9 \rangle$ is the mean square estimate of the fraction of the neutron capture cross-section that is due to the parity violating gamma asymmetry[9]. The second entry is the fraction of energy contained by the average first gamma ray in the de-excitation cascade. A_{UD} corresponds to the experimental measurements of this thesis. 127

5.7 A comparison of the experimental and theoretical results for the parity violating asymmetry. 129

A.1 Number density and thermal neutron cross-section of isotopes within target materials 138

A.2 Cross-sections[22] and Depolarization, T, for each target 141

LIST OF APPENDICES

Appendix

A. Target Materials 136

A.1 Signal Dilution 136
A.2 Depolarization within the Target 139

B. Derivations 142

B.1 Asymmetry Calculation 142
B.2 Detector Offset Correction 145

C. Code 148

C.1 Data Acquisition Code 148
C.1.1 daqgui_II 148
C.1.2 npdg.c 153
C.1.3 b_II.c 168
C.1.4 auto_to_scp 180
C.2 Analysis Code 181
C.2.1 trw.c 181
C.2.2 MCNP input files 190

CHAPTER I

Theory and Motivation

1.1 Introduction

1.1.1 Definition of Parity

The parity of a system depends on its behavior under a spatial inversion. A symmetry in the Hamiltonian of a system implies a conservation law. Invariance under a spatial translation implies Conservation of Momentum. Invariance under time translation implies Conservation of Energy. Invariance under spatial inversion implies Conservation of Parity. This is sometimes called reflection symmetry.

The operation of the parity operator on a wave function is defined as:

$$P(\psi(r)) \equiv \psi(-r) \tag{1.1}$$

which means that

$$P^2(\psi(r)) = \psi(r) \tag{1.2}$$

or

$$P^2 = 1 \tag{1.3}$$

Therefore, the parity operator, P, has eigenvalues of $+1$ or -1. For example, the wave functions of a spinless particle in a spherical potential are:

$$\psi(\vec{r}) = \chi(r) Y_m^l(\theta, \phi) \tag{1.4}$$

and parity inversion yields:

$$P\psi(\vec{r}) = \chi(r)(-1)^l Y_m^l(\theta, \phi) = (-1)^l \psi(\vec{r}) \tag{1.5}$$

Therefore, an even value of l results in an eigenvalue of 1 and is called 'even' parity; An odd value of l results in a eigenvalue of -1 and is called 'odd' parity.

Each particle has an intrinsic parity as well. Baryons have positive intrinsic parity. Mesons have negative intrinsic parity. Spatial inversion is a discrete symmetry which means that parity is a multiplicative quantum number. The parity of the entire system is the product of the intrinsic parity of its component particles and the parity of the wave function.

Parity is a conserved quantum number if the parity operator commutes with the Hamiltonian.

$$[H, P] = (HP - PH) = 0 \tag{1.6}$$

This condition of 'Conservation of Parity' restricts the form the Hamiltonian can take to exclude any term which switches parity of the wave function. All terms of H must be even under the parity transformation.

Since the vector products of operators which compose the terms of the Hamiltonian must be even to meet the parity conservation condition, it is important to understand how vector operators behave under a parity transformation. Polar vectors(e.g. a momentum) switch sign. Axial vectors(e.g. a spin) do not. A short list of vector operators and their behavior under parity inversion is shown in table 1.1.

	$\vec{r}$	$\vec{k_n}$	$\vec{s_n}$	$\vec{J}$	$\vec{L}$	$\vec{s_n}\cdot\vec{k_\gamma}$	$\vec{s_n}\cdot(\vec{k_n}\times\vec{k_\gamma})$
P	-1	-1	$+1$	$+1$	$+1$	-1	$+1$

Table 1.1: Vector operators under parity inversion

This thesis is about the measurement of an asymmetry in the distribution of the γ-rays generated by neutron capture on nuclei. This asymmetry is dependent on the vector product, $\vec{s_n}\cdot\vec{k_\gamma}$, and therefore violates the parity conservation condition. The rest of this chapter is devoted to the origin of parity violating effects and how they relate to the nucleon-nucleon interaction. It begins with a short overview of how the property of parity violation can be used to distinguish the strong and weak force in nuclei and then continues with a summary of the current knowledge of the weak force in nuclear reactions. And it closes with a description of the neutron capture γ-ray asymmetry measurements on ^{35}Cl, ^{113}Cd, and ^{139}La and the planned NPDGamma experiment.

1.1.2 The Discovery of Parity Non-conservation in the Weak Interaction

It used to be believed that parity conservation was universal. But in the early 50's, it was noticed that a K could decay into either 2π or 3π. Since mesons have a intrinsic negative parity, the final states of these decays in the same partial wave are opposite in parity. This caused a conflict with the parity conservation assumption.

It was noticed by Lee and Yang in 1956 that all previous checks of parity conservation had been made only on the strong and electromagnetic forces and that parity violation in the weak force could exist.

This theory was tested by Wu, Ambler, Hoppes and Hayward in a polarized ^{60}Co β-decay experiment. The distribution of the electrons emitted is given in equation 1.7.

$$W(\theta) = 1 + A(\vec{J}\cdot\vec{k_e}) \tag{1.7}$$

The second term of equation 1.7 is dependent on $\vec{J}\cdot\vec{k_e}$ where $\vec{J}$ is the spin of the ^{60}Co nucleus and $\vec{k_e}$ is the momentum of the electron. Because $\vec{J}$ is even under parity inversion and $\vec{k_e}$ is odd under parity inversion, the entire term $A(\vec{J}\cdot\vec{k_e})$ is odd under parity inversion and therefore represents a parity violating effect. The existence of this asymmetry proved that the weak force did not conserve parity. The existence of parity violation in the weak force has led both to an interest in testing other fundamental symmetries/conservation laws and to the experimental technique of using parity as a filter to separate the weak interaction from other effects.

1.1.3 Weak Interaction at the Lepton/Quark Level

The weak interaction differs from the strong and the electromagnetic in that parity and quark flavor are not conserved.

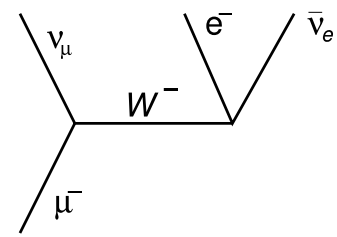

Figure 1.1: Muon decay, a leptonic weak interaction

It is modeled as a W$^\pm$ or Z^0 boson connecting either two lepton vertices(leptonic), two quark vertices(hadronic),or a one quark vertex and one lepton vertex(semi-leptonic). A couple of examples of the weak interaction are shown in figure 1.1 and figure 1.2. The "standard model" of the electroweak interaction is in complete agreement with experiment at the moment with the possible exception of the neutrino sector.

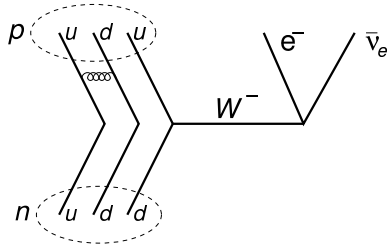

Figure 1.2: Beta decay, a semi-leptonic weak interaction

1.1.4 The Weak Contribution to the N-N Interaction

Because the range of the $W^{\pm}$ and Z^0, about .02 fm, is much smaller than the distance between nucleons($\approx$1 fm, where the strong force becomes highly repulsive.), it is unlikely for them to be exchanged directly between nucleons. Instead, a weak gauge boson is emitted by a quark, travels a short distance, and decays into a light meson which couples strongly to another nucleon.

At low energy, the strong interaction between nucleons is modeled by two strong vertices mediated by meson exchange. The weak nucleon-nucleon interaction is envisioned similarly but one strong vertex and one weak vertex connected by a weak boson and a meson, as in figure 1.3.

π, ω, ρ mesons can all act as the mediating particles but the π-meson coupling is special due to the fact that it has the longest range (lowest mass).

Measuring the weak interaction within a nucleus is difficult because it is obscured within a much larger strong background. An estimate of the relative size of the two can be made on dimensional grounds.

$$4\pi G m_\pi^2 \approx 10^{-7} g_{\pi NN} \tag{1.8}$$

G is the Fermi constant, m_π is the mass of the pion, and $g_{\pi NN}$ is the strong

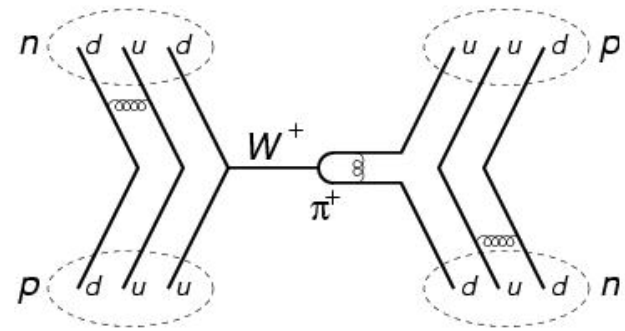

Figure 1.3: Hadronic weak interaction

coupling constant for the N-N interaction.

Since the weak interaction violates parity and the strong interaction conserves parity, parity violation can be used as a filter to pick out the weak force within the enormous background of the strong force.

1.1.5 The Weak Interaction in Medium and Heavy Nuclei

The weak interaction in medium and heavy nuclei is complicated by the nuclear structure.

The nuclear structure can serve to amplify PNC effects by two means. Since the interactions between nucleons are dominated by the strong force, the effect of the weak force on energy levels and wave functions is usually modeled as a perturbation on the strong interaction Hamiltonian. In heavy nuclei, it is possible for two strong interaction eigenstates of opposite parity to be very close together in energy. If the level spacing is small compared to the weak potential, the perturbation of the wave functions is large and the opposite-parity admixture of one state into the other is

amplified. This is called *dynamnic enhancement*.

Kinematic enhancement is proportional to the square root of the neutron widths. The neutron widths are related to the lifetime of the compound nuclei. If states of opposite parity which are close together have a large difference in their relative lifetime, the probability of the long lifetime state mixing into the short lifetime state and then decaying is amplified. In low energy neutron reactions in nuclei this happens near p-wave (l=1) resonances. The effect of both *dynamic* and *kinematic enhancement* on a parity violating observable is given in the following proportionality.

$$A \propto \frac{2\langle\psi_s|H_w|\psi_p\rangle}{E_s - E_p}\sqrt{\frac{\Gamma_s^n(Ep)}{\Gamma_p^n(Ep)}} \tag{1.9}$$

E_s and E_p are the energy of the neutron resonances and Γ_s^n and Γ_p^n are the gamma widths of those resonances.

The effect can also be suppressed in the integral experiments(such as this one) due to cancellations. When a heavy compound nucleus is formed, it decays to the ground state through several states which produces a γ-ray cascade. The intermediate states do not retain the same level of polarization as the initial state. The sign of the parity violation may not be the same either. Both of these effects can cause the parity violating asymmetries to be attenuated if the measurement technique sums over all of the resulting γ-rays. In these measurements a theoretical description of this effect is needed to compare theory and experiment.

1.1.6 Current Status of the Theoretical Predictions for N-N Weak Couplings

A combination of experimental and theoretical work has placed bounds on the values of the two longest range weak N-N coupling constants as shown in figure 1.4. These values come from measurements of the circular polarization of a $\Delta I = 1$ parity forbidden gamma transition in ^{18}F and from anapole moment measurements on ^{133}Cs

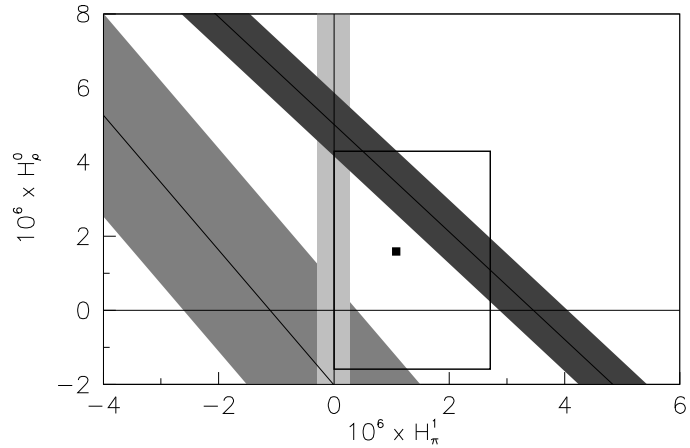

Figure 1.4: Weak coupling constants H_π^1 and H_ρ^0 extracted from ^{18}F (light band), ^{205}Tl (medium band), and ^{133}Cs (dark band) experiments. Also shown are the DDH "best" values (square) and "reasonable range" (box)

and ^{205}Tl.

The calculated values of the coupling constants disagree. But the calculations involve approximations of nuclear systems that are not completely understood. Experiments that directly measure the values of the weak coupling constants are necessary to resolve the conflict. The NPDGamma experiment is a direct measurement of H_π^1.

Weak Coupling Constants

Exchanged Meson	Coupling	ΔI	"Best Value" (10^{-6})	"Reasonable Range" (10^{-6})
π	H^1_π	1	1.08	0.0 - 2.71
ρ	H^0_ρ	0	1.59	-1.59 - 4.29
ρ	H^1_ρ	1	0.03	0.0 -.053
ρ	H^2_ρ	2	1.33	-1.06 - 1.54
ρ	H'^1_ρ	1	0.00	none
ω	H^0_ω	0	0.80	-2.39 - 4.29
ω	H^1_ω	1	0.48	0.32 - 0.80

Table 1.2: Estimated weak meson-nucleon couplings from reference [6]

1.2 The NPDGamma Experiment

1.2.1 The Fundamental Importance of NPDGamma

The goal of the NPDGamma experiment is to measure the fraction of γ-rays whose momentum is correlated with the spin of a neutron which is captured by a proton in the reaction:

$$\vec{n} + p \rightarrow d + \gamma + 2.23\text{MeV} \tag{1.10}$$

The observable is of the form $\vec{s_n} \cdot \vec{k_\gamma}$, which is odd under a parity transformation.

$$P(\vec{s_n} \cdot \vec{k_\gamma}) = -1 \tag{1.11}$$

The coefficient which multiplies this dot product is often written as a constant, called the asymmetry coefficient, A. The distribution of the γ-rays is given by equation 1.12 [1].

[1] It is of the same form as the electron distribution in the ^{60}Co β-decay experiment.

$$\frac{\partial\omega}{\partial\Omega} = \frac{1}{4\pi}(1 + A\frac{\vec{s_n} \cdot \vec{k_\gamma}}{|\vec{k_\gamma}|}) = \frac{1}{4\pi}(1 + A\cos(\theta)) \tag{1.12}$$

Figure 1.5 shows the form of the γ-ray distribution.

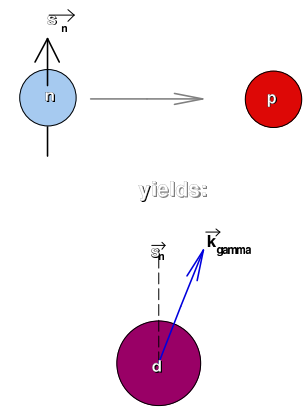

Figure 1.5: The reaction studied in this thesis.

Because hydrogen is a lone proton, no nuclear structure enhancement of the asymmetry exists[2]. The expected value is only 5*10^{-8}. Although measuring an asymmetry of this size presents many technical difficulties, it offers a clear look at the weak nucleon-nucleon coupling constants that have been obscured by uncertainty in the nuclear structure in heavier elements.

According to a calculation by Desplanques[7], the relation between NPDGamma asymmetry coefficient and the weak coupling constants is:

$$A_\gamma = -0.045(H^1_\pi - 0.02H^1_\rho + 0.02H^1_\omega + 0.04H^{1'}_\pi) \tag{1.13}$$

[2] see section 1.1.5

The superscripts denote a change of isospin. The subscripts denote the meson involved.

The NPDGamma measurement is primarily one of H_π^1. Weak couplings in which $\Delta I = 1$ are believed to be dominated by Z^0 exchange, which does not change the flavor of the quarks involved. $W^\pm$ exchange does change the flavor of the quarks. A $(u \to d) \times (d \to u)$ transition from a symmetric product of the two charged weak currents in the usual low energy current-current Hamiltonian results in $\Delta I = 0 + 2$ and is therefore not included in H_π^1. A $(u \to s) \times (s \to u)$ transition (which does give $\Delta I = 1$) is possible but is greatly suppressed by the small quark mixing angle involved ($V_{us}^2 \approx 0.04$). NPDGamma, therefore, is primarily sensitive to Z^0 exchange.

In the long term, NPDGamma can be seen as one of a series of experiments which will determine all of the weak coupling constants.

1. pp scattering at 221MeV, performed recently by TRIUMF[12]:

$$A_z = .028(H_\rho^0 + H_\rho^2/\sqrt{6}) = (.84 \pm .29(\text{stat}) \pm .17(\text{syst})) * 10^{-7} \tag{1.14}$$

2. pp scattering at 45MeV, performed by PSI[13]:

$$A_z = -.053(H_\rho^0 + H_\rho^2/\sqrt{6}) - .016(H_\omega^0 + H_\omega^1) = (-1.50 \pm 0.22) * 10^{-7} \tag{1.15}$$

3. Deutron photo-disintegration by polarized γ-rays:

$$P_\gamma = .022H_\rho^0 + .043H_\rho^2/\sqrt{6} - .002H_\omega^0 \tag{1.16}$$

4. Parity violating spin rotation in liquid para-hydrogen:

$$\Phi_{PNC} = -1.31H_\pi^1 - .23H_\rho^1 - .25H_\rho^2 - .23H_\omega^0 \tag{1.17}$$

Together with NPDGamma, there are 5 equations and 5 unknown coupling constants[3]. Experiments 1 and 2 have been performed accurately enough to get nonzero results and together they determine linear combinations of $(H_\rho^0 + H_\rho^2/\sqrt{6})$ and $(H_\omega^0 + H_\omega^1)$.

1.2.2 Apparatus of the NPDGamma experiment

A new flight path and beam line are being constructed for the NPDGamma at the spallation source at LANSCE. A 20 Hz pulsed beam of 800 MeV protons strikes a tungsten target producing pulses of neutrons by spallation. A liquid hydrogen moderator slows the neutrons and a neutron guide transports them to the experimental area. The beam is first filtered by transmission through a glass cell of polarized ^{3}He. The selective capture property of polarized ^{3}He filters one spin state of the neutrons leaving a polarized neutron beam. The polarized neutrons then travel through a radio frequency spin flipper which flips the spin of the neutrons on a pulse by pulse basis. The polarized neutrons then enter the target where the capture reaction takes place. The angular distribution of the resulting γ-rays is then measured by an array of CsI scintillation detectors. The front and back monitors measure the neutron beam current. A diagram of the NPDG experiment is shown in figure 1.6.

1.3 The Parity Violating Gamma Ray Asymmetries in Polarized Neutron Capture on ^{35}Cl, ^{113}Cd, and ^{139}La

1.3.1 Motivation

The goal of the experiment of this thesis was to measure parity violating $\vec{s_n} \cdot \vec{k_\gamma}$ asymmetry of the nuclei ^{35}Cl, ^{139}La, and ^{113}Cd.

[3] $H_\rho^{'1}$ is believed to be very small.

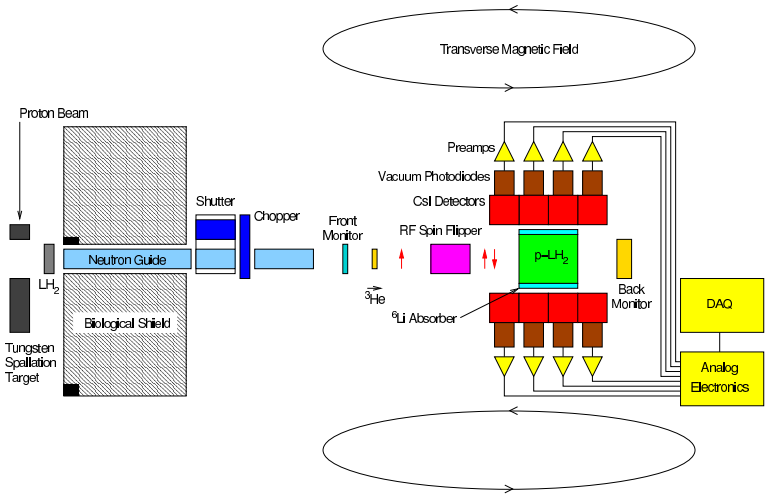

Figure 1.6: Experimental setup for the NPDGamma experiment

The principal purpose of making these measurements was to test the method and apparatus of the NPDGamma experiment. Because of the complicated nuclear structure of these isotopes is not fully understood, it is impossible to calculate the asymmetry coefficients or directly relate them to the weak coupling constants with high accuracy. But these asymmetries are approximately 10^3 times larger than the asymmetry of NPDGamma. This means that a factor of 10^6 less events are necessary to resolve the effect.

These expectations come from two past experiments involving $s_n \cdot k_\gamma$ measurements of these isotopes. One was performed at the B. P. Konstantinov Lenningrad Nuclear Physics Institute [1] and measured all three isotopes. The other was at ILL [3] and measured ^{35}Cl only. Both used reactor based beams, polarizing guides,

$A_\gamma * 10^6$

	^{35}Cl	^{113}Cd	^{139}La
BPKLNP[1]	-27.8 ± 4.9	-1.3 ± 1.4	-17.8 ± 2.2
ILL[3]	-21.2 ± 1.7	-	-

Table 1.3: Summary of past measurements of A_γ in the nuclei of interest

adiabatic flippers, and NaI/ photomuliplier detectors. Their results are summarized in table 1.3.

From the results of past experiments, both ^{35}Cl and ^{139}La were expected to have an asymmetry on the order of one part in 10^5. The particular nuclear structure of these isotopes results in large dynamic and kinematic enhancement effects which make these asymmetries large compared to the size of the weak N-N couplings. Neutron capture on ^{113}Cd was expected to have no asymmetry[4] from thermal neutron capture and was chosen to serve as a null check.

1.3.2 Apparatus

Since this experiment is a test of the method for NPDGamma, the apparatus is essentially a scale model of the NPDGamma apparatus. A diagram is shown in figure 1.7. Each of the components is described in detail in chapter 2.

1.3.3 NPDGamma Detector Alignment

We also hoped to find a parity conserving asymmetry that could be used to align the detector array. Measuring the parity violating γ-ray asymmetry of neutrons capturing on protons has a potential problem; there is a known parity conserving asymmetry,$(\vec{s_n} \cdot (\vec{k_n} \times \vec{k_\gamma})) = .23 * 10^{-8}$ at $E_n = 1$meV[14], which is on the same order as

[4]The prediction for the asymmetry of ^{113}Cd is not zero but it small compared to the 10^{-6} statistical error achievable in this experiment.

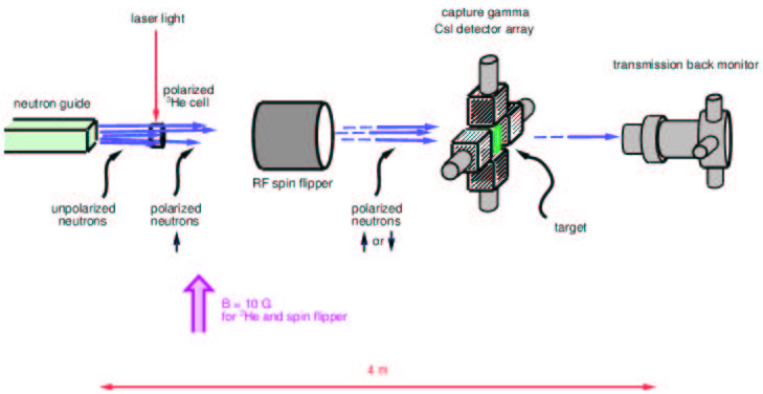

Figure 1.7: Experimental setup, beam travels left to right

the estimated value of the parity violating asymmetry. Therefore, the detector must be aligned to sufficient precision to ensure that this signal, which is orthogonal to the $\vec{s_n} \cdot \vec{k_\gamma}$, does not leak into it. If we could discover a nucleus in which $\vec{s_n} \cdot (\vec{k_n} \times \vec{k_\gamma})$ was large, we could use it to perform this alignment. ^{35}Cl and ^{139}La are known to have large $\vec{s_n} \cdot \vec{k_\gamma}$ asymmetries. The nuclear enhancement effects which make these asymmetries large are caused by mixing of s and p levels in the compound nucleus. The large p-wave resonance that is necessary for a large $\vec{s_n} \cdot \vec{k_\gamma}$ asymmetry is also a condition for a $\vec{s_n} \cdot (\vec{k_n} \times \vec{k_\gamma})$ asymmetry. Therefore, ^{35}Cl and ^{139}La were potential candidate nuclei to find such an asymmetry that could be used to align the NPDGamma detector array.

CHAPTER II

Experimental Overview

The experimental apparatus, shown in figure 2.1, consisted of a number separate systems each of which will be described in detail in the following sections.

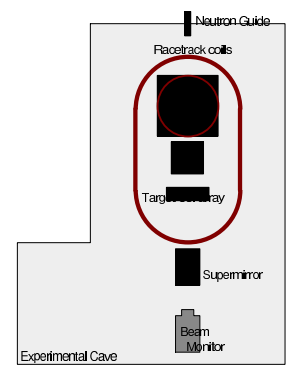

Figure 2.1: Top down view of the experimental apparatus. The gray area is the experimental cave.

2.1 The Neutron Beam

Neutrons for this experiment were supplied by the Lujan Center at LANSCE(shown in fig 2.2). The accelerator(figure 2.3), an 800 MeV, 100μA, 20Hz pulsed proton beam is directed to the Lujan center where it strikes a tungsten target. The impact of a 800MeV proton on a tungsten atom produces neutrons by spallation.

The proton beam strikes the tungsten target from above and the spallation neutrons spill out in all directions. Flight paths extend radially from the spallation target (figure 2.4). Each flight path has a moderator and/or a neutron guide tailored for the particular instrument.

Figure 2.2: An ariel photograph of LANSCE

Flight path 11A was used for this experiment. This flight path has a 13 cm square, partially-coupled[1] liquid hydrogen moderator at 20K. In the moderator, most

[1] A coupled moderator is one in which neutrons which would escape through the sides are reflected back into the moderator; flight path 11A is "partially-coupled" because some of the neighboring

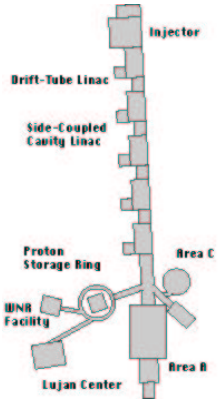

Figure 2.3: Schematic diagram of the accelerator, the spallation target is at the Lujan center,in the lower left of the diagram

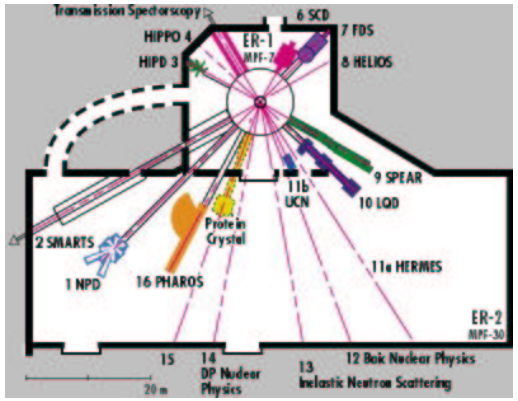

Figure 2.4: Floorplan of flight paths at the Lujan center, we used flight path 11A for these measurements

of the neutrons are slowed to thermal energies. One meter down the flight path, the neutrons enter a neutron guide.

The neutron guide is an evacuated tube of ^{58}Ni coated glass. It exploits the optical properties of thermal neutrons in matter to guide the neutrons by total external reflection from the metal coatings on the glass walls. Neutrons whose velocity component perpendicular to the walls is less than 7.5 $\frac{m}{s}$ will be reflected. Faster neutrons are transmitted through the walls and will escape.

The neutron guide is 19 m long, 6 cm square, and exits into the experimental cave, a steel and polyethylene shielded enclosure within ER-2, the high-bay of the Lujan center.

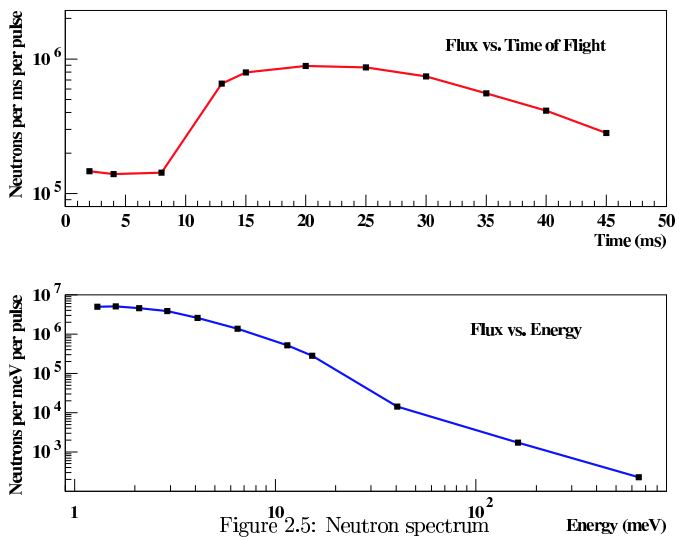

Figure 2.5: Neutron spectrum

moderators are coupled and their reflectors inadvertently effect the moderator of 11A.

The pulsed nature of the proton source means that the spallation neutron pulse occurred at the same 20 Hz interval. Since the source of neutrons is pulsed, the energy of the neutrons can be determined by their "time of flight"(TOF) between the moderator and a detector. The time of flight spectrum and the energy spectrum for this source are shown in figure 2.5.

2.2 The Magnetic Holding Field

A uniform vertical magnetic field is necessary to maintain the polarization of the neutrons and the ^{3}He by defining a static field that is large compared to any stray fields.

2.2.1 Theory of the Neutron Spin Motion in a Magnetic Field

The motion of the spin of a particle in a magnetic field is governed by the following equation:

$$\frac{d\vec{s}}{dt} = \frac{2\mu}{\hbar}\vec{s} \times \vec{B} \tag{2.1}$$

where μ is the magnetic moment of the particle. This will result in a Larmor precession; the spin will precess about the direction of $\vec{B}$ with a frequency of:

$$\omega_L = \frac{2\mu}{\hbar}B = \gamma B \tag{2.2}$$

γ is called the gyromagnetic ratio. An expanded derivation of this can be found in reference [28] or any text book on NMR. In this case, the functional effect of the Larmor precession is that once a neutron has been aligned parallel or anti-parallel to the uniform field, it stays aligned with that field. As long as the time rate of change in the direction of the magnetic field is small compared to the Larmor precession frequency, then the spin of the neutrons will follow slow changes in the the direction

of this guiding field. In practice, the time rate of change of DC field seen by a neutron is determined by the gradient of the field in the direction of the neutron travel times the velocity.

$$\omega \gg \frac{\frac{\partial \vec{B}}{\partial t}}{|\vec{B}|} = \frac{\frac{\partial \vec{B}}{\partial z_n} \cdot \frac{\partial z_n}{\partial t}}{|\vec{B}|} \tag{2.3}$$

It is important that the direction of the field is constant in the beam path in order to guide the spin of the neutrons between the elements of the beam line and that the change in the magnitude of the field is small.

2.2.2 Construction of the Magnetic Field

A vertical[2] 10 gauss field was created by a set of four horizontal racetrack coils which encompass the whole beam line from the spin filter to the target (see 2.1). The center pair of coils are Helmholz coils with 1 m diameter circular sections, 1.5 m straight "racetrack" sections, and a 1 m vertical separation. They are centered on the beam line.

The outer pair of coils are trim coils and were placed 0.5 m above and below the Helmholz coils and had half as many turns.

Because of the asymmetric steel walls of the experimental cave [3], the exact position and drive current of these coils were tuned in place in order to minimize the field gradients at the the position of the ^{3}He cell and the RF spin flipper, the two most sensitive devices. The coils were connected in series and needed a current of 3 Amps to deliver the 10 gauss field. At the positions of the ^{3}He spin filter and the RF spin

[2] pointing toward the floor

[3] The Flight Path 11A had walls and a ceiling of 6" thick steel, but the floor was concrete. The beam guide was not exactly centered nor was the cave square. And there was an alcove on one side of the back of the cave. The structure had originally been designed solely to shield ER2 from radiation and no interest was given to the magnetic symmetry within the cave

flipper, the field gradients were measured to be on the order of a few milligauss per cm, which was well within their operational limits.

The holding field coils were supported by an aluminum Unistrut frame which was both bolted to anchors in the concrete floor of ER2 and to steel Unistrut welded to the roof of the flight path. This frame then provided the physical foundation upon which the rest of the experiment was anchored.

2.3 The ^{3}He Spin Filter

2.3.1 The Effect on the Neutron Beam

The polarized neutron beam for the experiment is produced by filtering the unpolarized beam through a glass cell(figure 2.8) filled with polarized ^{3}He. When the spin of a ^{3}He nucleus is opposite to that of a neutron, the thermal capture cross-section is very large, $10.64 * 10^3$ barns at 25 meV. When the spins of the neutron and the ^{3}He are aligned the total cross-section is just a few barns. This results in a large fraction of one polarization state of neutrons being captured while the other polarization state is only slightly attenuated as it passes through. The calculation of this effect follows.

$$\sigma_a = (\frac{2200\frac{m}{s}}{v_n})(5320 \text{ barns}) \tag{2.4}$$

$$T_+ = e^{-n_3 l \sigma_a}(1 - P_3) \tag{2.5}$$

$$T_- = e^{-n_3 l \sigma_a}(1 + P_3) \tag{2.6}$$

$$P_n = \frac{T_+ - T_-}{T_+ + T_-} = \tanh(n_3 l \sigma_a P_3) \tag{2.7}$$

$$T_n = \frac{T_+ + T_-}{2} = e^{-n_3 l \sigma_a} \cosh(n_3 l \sigma_a P_3) \tag{2.8}$$

The transmission, T_n, which is easily measured with the neutron monitor, gives us the product, $n_3 l \sigma_a P_3$, through equation 2.8; which in turn gives us the polarization of the neutron beam, P_n, through equation 2.7.

Figure 2.6 is a plot of P_n and T_n over the relevant energy range. Figure 2.7 is a plot of the calculated versus measured P_n versus time of flight for one accelerator pulse. The ^{3}He polarization, P_3, was $.265 \pm .003$. The thickness of the cell, $n_3 l$, was 6.0 atm-cm.

2.3.2 Polarizing ^{3}He through Spin Exchange with Optically Pumped Rb

The ^{3}He cell also contains small amounts of Rb and N_2. The Rb is the target of the optical pumping. The N_2 gas quenches the excited Rb atoms, absorbing the energy and angular momentum in the rotation of the N_2 molecules. This prevents reradiation of the light by the Rb, which would be isotropic and therefore depolarizing.

The Zeeman effect created by the magnetic field, splits the two degenerate $m_j = \pm\frac{1}{2}$ electron states into two distinct energy levels. The splitting of the levels allows us to optically pump the Rb into a state of preferred angular momentum by illuminating it with circularly polarized light from a laser diode array, see figure 2.9. Because the light carries one unit of angular momentum, only the $5s_{\frac{1}{2},m_j=-\frac{1}{2}} \rightarrow 5p_{\frac{1}{2},m_j=\frac{1}{2}}$ excitation is allowed. Then it is the hyperfine interactions between the Rb electrons and the ^{3}He nuclei that transfer the angular momentum to the nuclei.

The cell for this experiment was a dumbbell shaped double cell. One end was located within a hot air oven in order to vaporize the Rb so that it could be optically pumped. The other end extended out of the oven and was used to filter the neutron beam. The polarized ^{3}He gas diffused between the two cells. The major advantage of this setup is that it has a minimum of unwanted material in the path of the neutron beam, i.e. the oven is kept out of the way. By choosing to polarize the neutrons transversely to the neutron beam direction, the laser optics are kept out of the beam path as well.

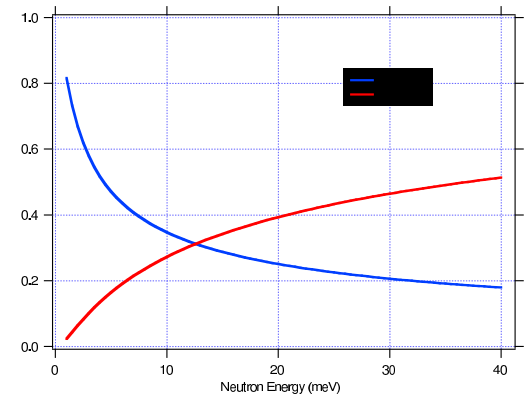

Figure 2.6: Neutron transmission of the ^{3}He cell and the polarization of the filtered neutron beam

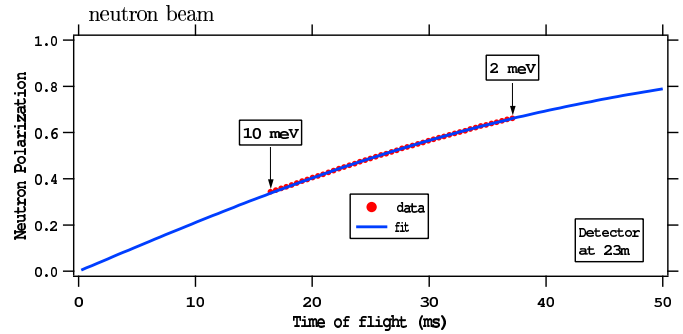

Figure 2.7: Neutron beam polarization, dots represent measured polarization, the line is the theoretical prediction from ^{3}He transmission, error on this measurement is $\approx 1\%$ of full scale.

Figure 2.8: This is the double ^{3}He cell. The optical pumping cell is in front. The box surrounding it is the oven; laser light enters through the window in the top. The filter cell is on the far side; small pickup coils are located on either side. The NMR coils surrounding the assembly are partially visible.

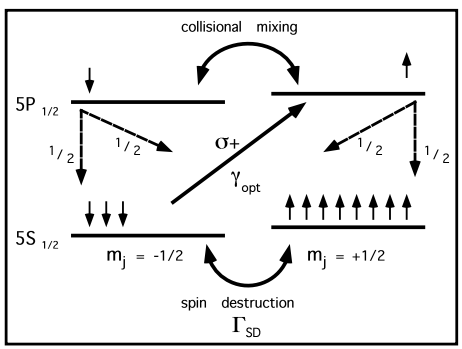

Figure 2.9: Rb energy levels and transitions

The final polarization achieved by the cell is a complicated function of many variables. It depends on pressure and temperature, spin exchange rates, efficiency of the optical pumping, depolarization rates, the choice of cell glass, etc. This is discussed in detail in the references [15] and [16].

2.3.3 Measurement of the ^{3}He Polarization by Adiabatic Fast Passage Nuclear Magnetic Resonance

It is necessary to know the polarization of the neutrons during the experiment because the measured asymmetry is dependent on it. But during the experiment, it wasn't practical to make regular measurements of the neutron beam polarization because of the additional apparatus that was placed in the beam path between the ^{3}He cell and the transmission monitor.

We measured the ^{3}He polarization by adiabatic fast passage nuclear magnetic resonance(AFP-NMR) and then calculate the neutron transmission. The AFP-NMR is a method that involves two sets of coils oriented perpendicular to each other and to the holding field coils. The first set creates a radio frequency(RF) magnetic field :

$$\vec{B}_{DC} + \vec{B}_{rf} = B_0\hat{z} + B_1(t)\cos(\omega t)\hat{y} \tag{2.9}$$

The DC field, $B_0\hat{z}$, is the field produced the magnetic field coils. The RF field, $B_1(t)$ is produced by the one meter RF coils. ω and γ are the Larmor frequency and the gyromagnetic ratio of the ^{3}He atom, respectively. $\hat{z}$ is vertical. $\hat{y}$ is horizontal in the direction of neutron momentum.

Traditionally, the next step is to transform the equation into a frame of reference that rotates with the ^{3}He atom's Larmor precession about the $\hat{z}$-axis with a frequency of ω.

$$\vec{B}_{eff} = \left(B_0 - \frac{\omega}{2\pi\gamma}\right)\hat{z} + \frac{B_1(t)}{2}\hat{y} \tag{2.10}$$

In this frame of reference, the RF field, which is tuned to alternate with the same frequency as the Larmor precession frequency, will appear to be constant. For $B_0 \gg B_1(t)$ and ω off-resonance, $\vec{B}_{eff}$ is mainly in $\hat{z}$. But when ω is on resonance, $\vec{B}_{eff} = \frac{B_1(t)}{2}\hat{y}$. Therefore, the ^{3}He atom's spin will also precess about the RF. The strategy of the AFP-NMR is to rotate the spins of the ^{3}He atoms by exactly π. Two things control the angle of rotation of the spin: the gyromagnetic ratio and the integral of strength times duration of the RF magnetic field. The first is a natural constant. The second is tuned to provide the 180 degree flip.

$$\pi = \gamma \int B_1(t)dt \tag{2.11}$$

As the ^{3}He atoms flip together, the magnetic field composed of their combined magnetic moments rotates as well. By placing coils perpendicular to the RF coils, we can measure a changing magnetic flux created by the combined magnetic moments of the flipping ^{3}He atoms.

$$emf = -\frac{\partial \Phi}{\partial t} \tag{2.12}$$

The exact flux through the coil generated by a rotating ^{3}He atom within the cell is difficult to calculate due to uncertainty in the geometry. But as long as the setup remains undisturbed, the voltage produced will be some unknown geometric constant times the number of polarized ^{3}He atoms.

The transmission measurement discussed in section 2.3.1 was used to calibrate the NMR voltage signal to the polarization of the ^{3}He. Then AFP-NMR measurements were made throughout the experiment to give a relative value of the ^{3}He polarization. The results of these measurements are shown in table 2.1 and plotted in figure 2.10.

The loss of polarization during this process is negligible. The cause of the loss is that the lasers can pump in the reversed state (without adjusting the optics). While the lasers are turned off, the ^{3}He polarization decays exponentially to zero. This

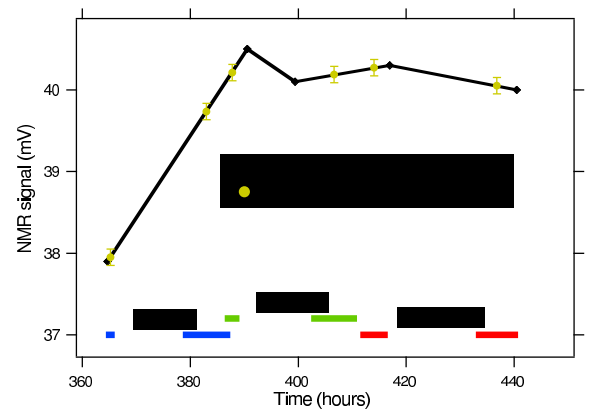

Figure 2.10: ^{3}He NMR signal over the experiment, 40meV is proportional to a polarization of 26.5%

polarization loss is negligible because the time to make the AFP-NMR measurement is small (a few seconds) compared to the polarization lifetime of the cell (38 hours).

The ^{3}He polarization varied by less than 1%, leading to a variation of the neutron polarization of less than 2%.

2.4 The Radio Frequency Spin Flipper

Flipping the polarization of the neutron beam on a pulse by pulse basis allows the cancellation of systematic errors which are uncorrelated with the neutron spin. The traditional solution to this problem is a gradient spin flipper[4], but that was

[4] A gradient spin flipper works by exploiting the adiabatic neutron transport condition given in equation 2.3. It consists of a gradually changing DC magnetic field which changes the Larmour precession frequency to bring neutrons into resonance with a static RF field.

Target	Runs ⋆	NMR signal
Chlorine	3216 – 3220	37.95mV
	3293 – 3382	39.73mV
	3216 – 3382	39.64mV
Cadmium	3386 – 3403	40.21mV
	3494 – 3576	40.18mV
	3386 – 3576	40.19mV
Lanthanum	3589 – 3640	40.27mV
	3666 – 3739	40.05mV
	3589 – 3739	40.13mV

Table 2.1: NMR signal for each target and run block(⋆ Not every run in a data block is a good asymmetry run)

impractical because of the low magnetic field gradient requirement of the ^{3}He system and, in the case of the NPDGamma experiment, because of Stern-Gerlach steering of the neutron beam[5]. Our solution was to flip the neutron spin by AFP-NMR as was done in the ^{3}He polarization measurements.

2.4.1 Theory of the NMR Spin Flip

The theoretical discussion of the RFSF is similar to that of the spin transport field and the AFP-NMR of the ^{3}He.

$$\vec{B}_{DC} + \vec{B}_{rf} = B_0\hat{z} + B_1(t)\cos(\omega t)\hat{y} \tag{2.13}$$

Again the DC field, B_0 is the field produced the holding field coils. The RF field,

[5]Spin dependent steering of the neutron beam combined with a gain difference in the detectors would create a systematic error

$B_1(t)$, is produced by the solenoid of RFSF itself. ω and γ are now the neutron Larmor frequency and the neutron gyromagnetic ratio, respectively. $\hat{z}$ is vertical. $\hat{y}$ is horizontal in the direction of neutron momentum. Again we switch to a rotating frame of reference.

$$\vec{B}_{eff} = \left(B_0 - \frac{\omega}{2\pi\gamma}\right)\hat{z} + \frac{B_1(t)}{2}\hat{y} \tag{2.14}$$

In this reference frame, the neutron spin will also precess about the RF. And again, we want to rotate the spin by exactly π. The primary[6] difference between the RFSF and the ^{3}He AFP-NMR is that the time the neutron is in the RF field is no longer an arbitrary duration. Because the neutrons in the beam are moving through the apparatus, the time a neutron spends in the RFSF, T, is dependent on the velocity of the neutrons and the length of the chamber. Due to the nature of the spallation source, the velocity of a neutron is set by the difference in time between the proton pulse striking the target and the neutron entering the spin flipper, t, and the distance between these two points, d. The length of the RFSF, L, was set at its construction.

$$T = \frac{L}{v(t)} = \frac{Lt}{d} \tag{2.15}$$

So the RF magnetic field amplitude must be maintained in inverse proportionality with time of flight so that the angle of precession about the rotating $\hat{y}$-axis will always be exactly π for all neutron velocities as they arrive at the device.

$$B_1(t) = \frac{d}{\gamma L t} \tag{2.16}$$

Since a magnetic field can't be created without fringe effects, it is really the

[6]besides the change in frequency with the change in particle

integral of $B_1(y,t)$ over the length of the field that must vary inversely with time.

$$\int B_1(y,t)dy = \frac{d}{\gamma L t} \tag{2.17}$$

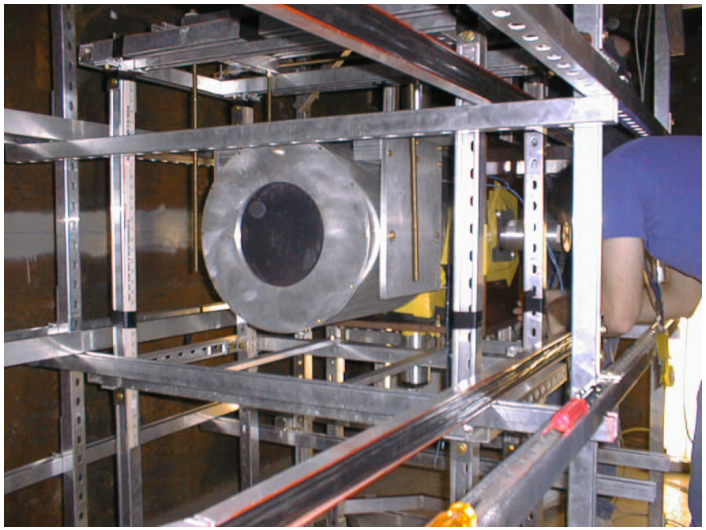

Figure 2.11: Photo of the Radio Frequency Spin Flipper

2.4.2 Construction and Tuning of the RFSF

A photo of the spin flipper in place is shown in figure 2.11 and a schematic diagram is shown in figure 2.12. The DC component of the field is just the 10 gauss provided by the magnetic holding field, which is also visible in the photo. The RF field transverse to the neutron spin is created by a solenoid. The 40 cm long, 25 cm inner diameter solenoid was made with 130 turns of 14 gauge wire. It had an inductance of 7.7 mH and was nominally driven at 29.6 kHz.

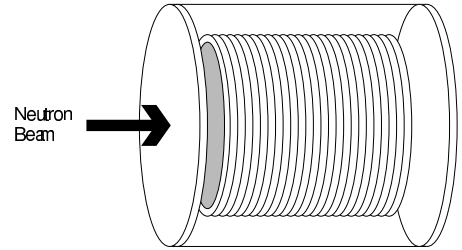

Figure 2.12: Schematic of the Radio Frequency Spin Flipper, it consist of a RF coil surrounded by an Al can

The solenoid is enclosed in an aluminum can to shield the electronics of the rest of the experiment from the radiated RF noise. This is essential for the suppression of spin correlated noise in the electronics, which would create a false asymmetry. The 2 mm thick Al windows on either end of the solenoid allow the beam to pass with only slight attenuation($< 1\%$) while still being 4 times the .5 mm skin depth of 30KHz RF.

The RF spin flipper could only be activated for a fraction of the macro pulse for two reasons. For short time of flight(i.e. fast neutrons), the magnetic field, $\vec{B}_1(t)$, would need to go to infinity in order to meet the $\frac{1}{t}$ form required to flip the spin. Second, at long time of flight, there is a problem with neutron pulses overlapping. The accelerator runs at approximately 20 Hz and the triggering of the spin flipper as well as the rest of the experiment is synchronized to the accelerator clock. If we tried to use the entire 50 ms of a macro pulse, then the spin flipper wouldn't always finish before the next pulse arrived because the accelerator clock occasionally ran faster than quoted. It is more important for the spin flipper not to overlap pulses than it is to capture the last few milliseconds of the macro pulse which have too few neutrons to be useful anyway. As a result, the spin flipper was run from 5 to 45 ms

after T_0, the accelerator synchronization trigger signal.

2.4.3 Efficiency of the RFSF

In order to measure the efficiency of the RFSF, ϵ_R, it was necessary to compare the polarization of the neutron beam before and after the RFSF.

$$\epsilon_R \equiv \frac{\text{beam polarization after RFSF spin flip}}{\text{beam polarization before RFSF spin flip}} \tag{2.18}$$

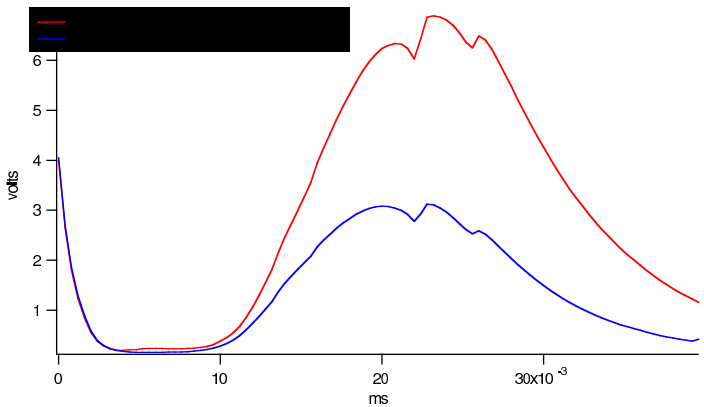

Figure 2.13: Transmission of the spin flipper through the supermirror analyzer

By this definition, the efficiency is 100% when the spin flipper is turned off. To determine ϵ_R when the spin flipper is on, the polarization was determined down stream by a neutron supermirror transmission analyzer. The supermirror has an energy dependent analyzing power of its own[7]. To take this into account, the ^{3}He spin filter was polarized to pass the same spin state as the supermirror and the

[7]The supermirror is describe in more detail in section 2.7.

neutron time-of-flight spectrum was measured by the $^3He/H_2$ ion chamber. Then the ^{3}He spin filter was polarized into the opposite state and the RFSF was used to flip the neutron spin back. A sample of both spectrums is shown in figure 2.13.

A comparison of these two spectrums yielded the efficiency. In order to measure this efficiency as a function of the beam position in the RFSF, a one inch collimater was used to narrow the beam and an adjustable mount moved the spin flipper, repeating the measurement at each position. The results are shown in figure 2.14. Although the performance falls off on the edges, it is very efficient, $98 \pm 2\%$, for the two inch diameter beam that we used for these measurements[8].

Although the RFSF is axially symmetric about the beam line, the DC magnetic field which produces the precession is not. The flux return of the magnetic field lines at the ends of the solenoid produces field gradients. The closer the neutrons are to the edge of the solenoid the larger these gradients are. Vertical gradients are aligned with the vertical DC magnetic field and do not change the orientation of the magnetic field. Horizontal gradients do change the orientation of the magnetic field. This causes the efficiency as a function of the vertical offset and the efficiency as function of horizontal offset to be different.

[8]The performance of the RFSF as a function of position is important to the NPDGamma experiment, which will use a different flight path with a much wider beam than was used in this measurement.

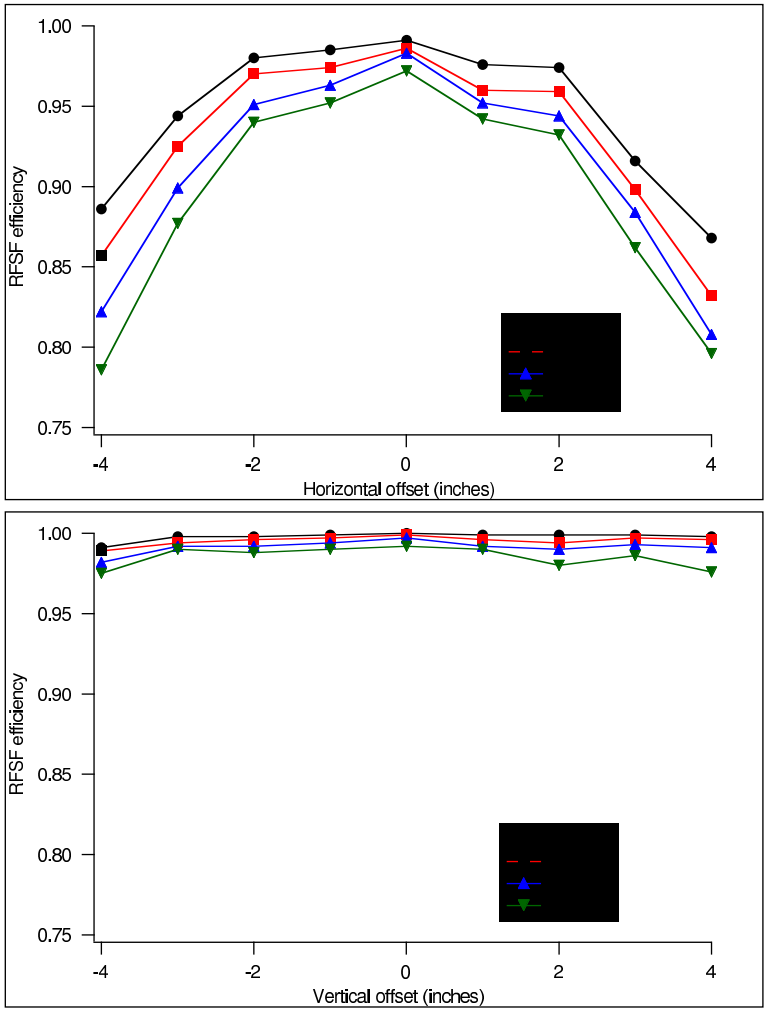

Figure 2.14: Spin Flipper Efficiency

2.5 The Targets

Three targets were needed for the experiment: one for ^{35}Cl, ^{113}Cd, and ^{139}La.

2.5.1 The Chlorine Target

Although natural chlorine is composed of both ^{35}Cl and ^{37}Cl, the ^{37}Cl has a comparatively small neutron capture cross-section and makes only a small contribution to the background in the γ-detector. (Not nearly enough to justify to using expensive isotopically pure ^{35}Cl.)

Using pure elemental chlorine was obviously impractical as well due to cost, health risks, and the technical difficulty of liquifying the gas. So the choice of which chlorine compound to use had to be made. Two easily available and relatively safe sources of chlorine are NaCl and CCl_4. Even though it is mechanically more difficult to manage, we chose CCl_4 because of the lower background generated within the materials. The relative merits of CCl_4 and NaCl are compared in appendix A. NaCl was used in the previous measurements[1] and [3].

The carbon tetrachloride target was contained within an Al can with two fill ports and dimensioned in the schematic, shown in figure 2.15. The bend in one of the fill lines served as a place to hold air bubbles out of the beam. A photo is shown in figure 2.16.

2.5.2 The Lanthanum Target

Natural lanthanum is composed of 99.91% ^{139}La, which is the nucleus of interest. This target was a pure La metal cylinder, 4 inches and diameter and 1 inch thick, enclosed in an evacuated #304 stainless steel can with .040" thick walls. It had been

Table of selected isotopes

Element /Isotope	% of natural abundance	thermal neutron capture cross-section(barns)	lifetime $t_{\frac{1}{2}}$	AWU
Carbon				12.000
^{12}C	98.89	.0037		
^{13}C	1.11	.0009		
^{14}C	0.00	-	5770 years	
Sodium				22.999
^{23}Na	100.00	.53		
^{24}Na	0.00	-	15 hours	
Chlorine				35.453
^{35}Cl	75.53	45		
^{36}Cl	0.00	90	$3 * 10^5$ years	
^{37}Cl	24.47	.005		
^{38m}Cl	0.00	-	1 s	
^{38}Cl	0.00	-	37.3 minutes	
Cadmium				112.400
^{106}Cd	1.22	-		
^{107}Cd	0.00	-	6.7 hours	
^{108}Cd	0.88	-		
^{109m}Cd	0.00	-	12μs	
^{109}Cd	0.00	-	470 days	
^{110}Cd	12.39	.20		
^{111m}Cd	0.00	-	49 minutes	
^{111}Cd	12.75	-		
^{112}Cd	24.07	.03		
^{113m}Cd	0.00	-	14 years	
^{113}Cd	12.26	27000		
^{114}Cd	28.86	.14		
^{115m}Cd	0.00	-	43 days	
^{115}Cd	0.00	-	2.3 days	
^{116}Cd	7.58	1.4		
^{117m}Cd	0.00	-	3.2 hours	
^{117}Cd	0.00	-	2.8 hours	
Lathanum				138.91
^{138}La	.089	-	$1.1 * 10^{11}$ years	
^{139}La	99.911	8.9		
^{140}La	0.000	-	40.2 hours	

Table 2.2: Table of selected isotopes, isotopes of interest are boldfaced, the m superscript designates isomers

Target Material Densities

Density	NaCl	CCl_4	La	Cd
$\rho(\frac{g}{cc})$	2.165	1.587	6.085	8.650

Table 2.3: Target material densities

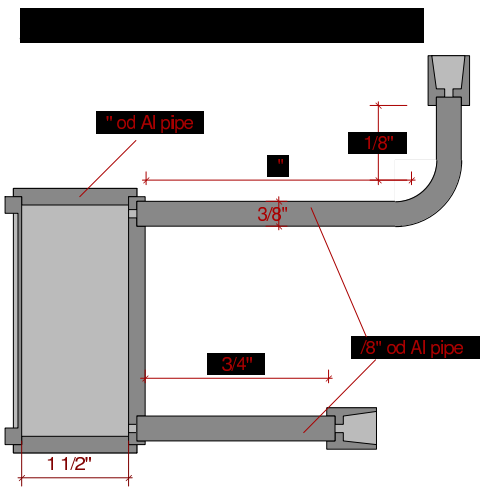

Figure 2.15: Schematic of the CCl_4 target

Figure 2.16: Photo of the CCl_4 target

Figure 2.17: Photo of the La target

built for and used in the TRIPLE experiment; a photo is shown in figure 2.17.

2.5.3 The Cadmium Target

Natural cadmium consists of a multitude of isotopes, but the neutron capture in ^{113}Cd dominates the γ-ray spectrum, having a neutron capture cross-section which is three to four orders of magnitude greater than that of the other isotopes (see table 2.2).

The cadmium target was a $\frac{1}{32}$ inch thick piece of regular Cd neutron shielding foil shaped into a four inch diameter circle.

2.6 The Detector Assembly

2.6.1 The ^{6}Li Plastic Shield

A tube of ^{6}Li doped plastic served two purposes. First it was used to mechanically support the targets within the detector array. It had a 4 inch inner diameter in which

the circular targets exactly fit and a six inch outer diameter which fits exactly within the detector array.

Secondly, it was used to shield the scintillators from neutrons which scattered out of the targets before absorption. If neutrons were captured in the CsI, it would create background problems during the experiment and storage problems after the experiment when the crystals became activated. ^{6}Li is a good choice for the shielding because it has a very high thermal neutron cross-section and captures neutrons without creating γ-rays which would interfere with the measurement of the asymmetry. The reaction is:

$$\mathrm{n} + {}^6\mathrm{Li} \rightarrow {}^4\mathrm{He} + {}^3\mathrm{H} + 4.78\mathrm{MeV} \tag{2.19}$$

2.6.2 The CsI Array

An array of four 15 cm cubes of thallium doped CsI(Tl) crystals were used to detect the γ-rays. These scintillators were placed above, below, to the left and to the right of the beam and target as shown in fig 2.18.

The visible line through the center of the CsI cubes is a seam where two smaller crystals were combined to form the cube[9]. This seam is coated with optical grease and measurements by the supplier indicated no significant light transmission losses at the junction. The two pin LEMO sockets in diagonally opposite corners are connections to LEDs that can be used test the photodiode/amplifier.

The CsI cube is then enclosed in a thin aluminum can in order to protect the slightly hygroscopic crystals from absorbing water and becoming clouded and to protect people from a potentially hazardous material. The inside of the aluminum

[9]The two smaller crystals were cheaper than one large crystal; our supplier was geared to production of crystals that were 15 cm $\times$ 15 cm $\times$ 7.5 cm.

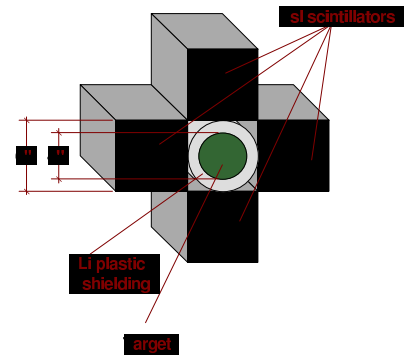

Figure 2.18: CsI detector array schematic

can is lined with a thin layer of white teflon to diffusely reflect the scintillation light back into the crystal and eventually out the window and into the photodiode.

As mentioned before, the CsI cubes were shielded from stray neutrons. This is necessary to prevent neutron activation in the detectors due to capture on ^{137}Cs, which gives a β-active nucleus with a long halflife. The primary shielding was the ^{6}Li loaded plastic tube supporting the target. The other five sides of the CsI cubes were shielded from stray neutrons by boron doped plastic. Although neutron capture on ^{10}B yields a γ-ray, the energy is low(.5 MeV), the number of neutrons which strike the sides of the detectors is small, and the signal will be uncorrelated to the spin of the neutrons [10].

[10]The only measurement of the parity odd gamma asymmetry in polarized neutron capture on ^{10}B resulted in an asymmetry of $A_\gamma = (6.8 \pm 3.7) * 10^{-7}$[17]. This factor of 100 combined with

This detector counts photons in current mode, meaning that the pulses from individual γ-rays overlap. Because there is no need to distinguish the shape of the pulses, it is only important that the time response of the CsI crystal be fast compared to the DAQ sampling rate so that digitization error is small compared to $\sqrt{N}$ counting statistics. Figure 2.19 shows the time response of the crystal, which is much faster that the DAQ sampling rate of 2.5 kHz[11].

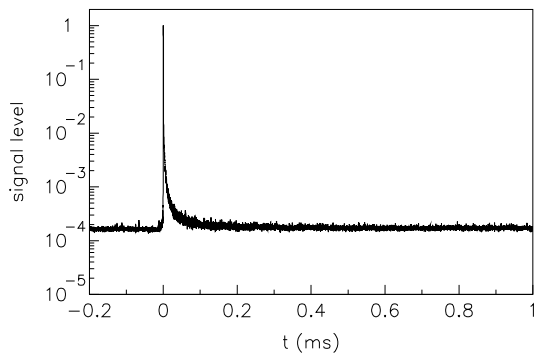

Figure 2.19: Time response of the CsI crystal, measured with a sealed ^{137}Cs gamma source and a photomultiplier

The process of scintillation is described in figure 2.20. The γ-ray excites electrons from the valence band to the conduction band as it scatters through the crystal. The energies of the conduction and valence band are dependent on the interatomic

the reduced flux and polarization from the multiple scattering required to reach the ^{10}B shielding causes this effect to be an insignificant error.

[11] The DAQ system is described in detail in section 2.9.

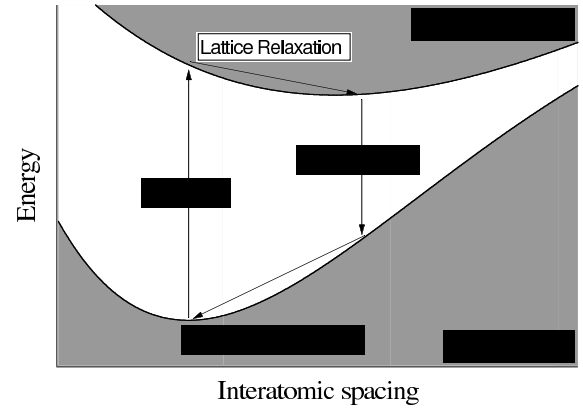

Figure 2.20: Mechanism of scintillation in inorganic crystals

spacing. In equilibrium, the crystal spacing is determined by the energy minimum of the band in which the electron is residing. Because the lattice relaxation time (10^{-12} s) is very small compared to the decay transition time (10^{-6} s), the crystal is expanded locally by an electron in the conduction band. When luminescence occurs, the band gap is smaller. The photons produced do not have enough energy to excite electrons into the conduction band in the bulk of the crystal (which has the lattice spacing determined by the minimum in the valence band). There is no mechanism to reabsorb the light so the crystal is transparent to its own luminescence. This is a necessary condition for light to be detected by the photodiode. A full description of this process is given in reference[26].

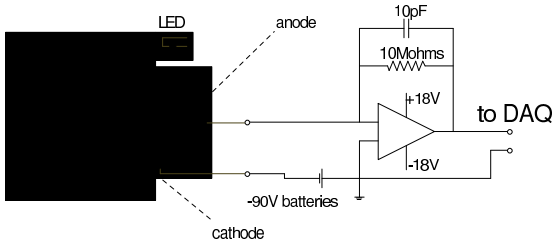

Figure 2.21: Schematic of the scintillator/photodiode/amplifier assembly

2.6.3 The Photodiodes and Electronics

The gains of photomultiplier tubes are very sensitive to magnetic fields and therefore could potentially create a systematic error in this experiment; pickup from the spin flipper could create a spin dependent gain for the photomultiplier and appear as a false asymmetry.

To avoid this problem, a photodiode and a low noise analog electronic I→V amplifier were chosen to convert the light into a voltage signal. Although the photodiode can still be affected by magnetic fields the size of the effect is many orders of magnitude smaller(a fractional change of $< 10^{-4}$/Gauss according to measurements [18]). The CsI and photodiode combination produced between 300 and 1200 photoelectrons per MeV.

The electrical current from the photodiode was transformed into a voltage by an analog I→V amplifier. The voltage was then read directly by the analog-to-digital converters of the data acquisition system. The feedback resistor for the amplifier is a 10MΩ. The feedback capacitor is 10pF. The op amp is an AD745A with a noise figure of 20 $\frac{\text{fA}}{\sqrt{\text{Hz}}}$.

A schematic diagram of the scintillator/photodiode/electronic amplifier assembly is shown in figure 2.21. A photos are shown in figure 2.22 and figure 2.23 .

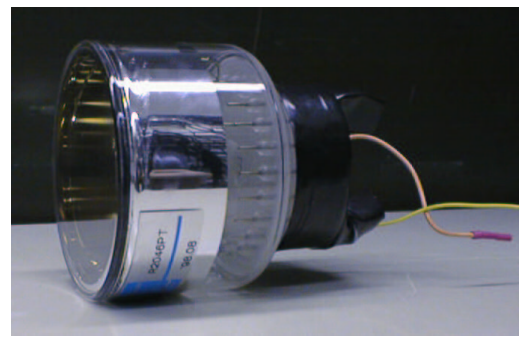

Figure 2.22: Hamamatsu ZC875-R2046PT photodiode which was used in this experiment

It is desirable that the neutron counting statistics be the dominant noise source in order to minimize the error on the final asymmetry measurement. The noise design of this amplifier was done for the NPDGamma experiment which will be collecting data at approximately a rate 100 times our collection rate and is therefore built well beyond the needs of the present experiment. A plot of the bench measurement of the noise figure for this amplifier is shown in figure 2.24. A comparison of the relative

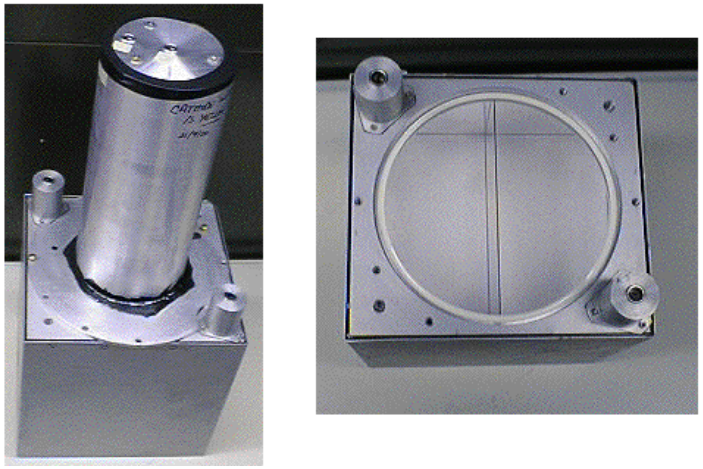

Figure 2.23: Left: the entire detector assembly, right: the CsI-Tl scintillator

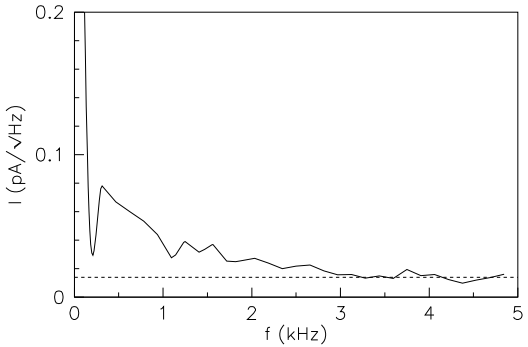

Figure 2.24: Bench measurement of the photodiode preamp noise

contributions of electronic noise and neutron counting statistics is made in chapter 4.

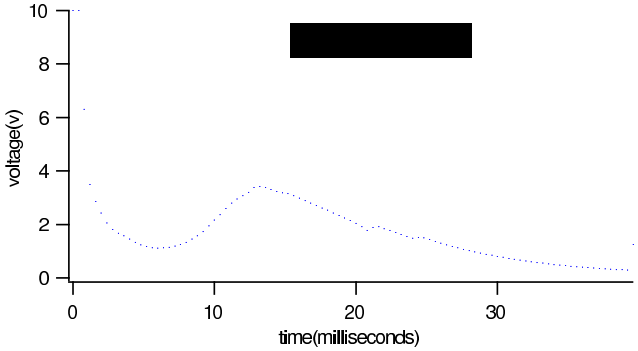

Figure 2.25: Sample photo-diode pre-amp signal for the Cl target, this is the gamma intensity converted to voltage as a function of TOF

A sample of the signal from these γ-detectors is shown in figure 2.25.

2.7 The Supermirror Polarization Analyzer

A supermirror analyzer, borrowed from NIST, was used to measure the polarization of the neutron beam.

This device consists of a large permanent magnet and a series of glass plates on which layers of cobalt and gadolinium metal was sputtered. Its operation is best understood when viewed from the reference frame of the neutron. Neutrons whose deBroglie wavelengths are large compared to the spacing of the cobalt atoms, see the entire cobalt layer as a uniform barrier(an optical potential) of about 100 neV as opposed to a distribution of individual scattering centers. The neutrons also see the

magnetization in the Co layer as a potential, $\pm\vec{\mu}\cdot\vec{B}$. So for one spin state of neutrons and the correct value of $\vec{B}$, the magnetic and nuclear potentials cancel. Neutrons of this spin state pass through the cobalt and are absorbed in the gadolinium layer behind. The other spin state sees a potential that is twice that of the nuclear potential alone and is reflected. The glass plates are stacked at a small angle to the neutron beam such that there is no clear shot through the supermirror. Only neutrons which scatter once through a small angle are able to pass through.

A supermirror has high total transmission but is only effective in a limited energy range[12], 2meV to 10meV (The high energy cutoff is due to the long wavelength approximation becoming invalid. The low energy cutoff is due to the $\frac{1}{v}$ cross-section becoming large enough to dominate).

More information about supermirrors can be found in reference [19].

[12]They function well as spin analyzers and polarizers for narrow energy bands and the ease of use is an advantage over the more complex ^{3}He setup.

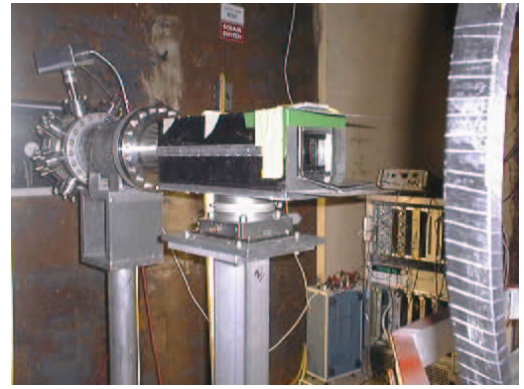

Figure 2.26: The beam travels from right to left, the supermirror analyzer is in the center, the H/^{3}He ion chamber is to the left, the end of the holding field coils is on the right, the data acquisition system is visible in the alcove behind the supermirror

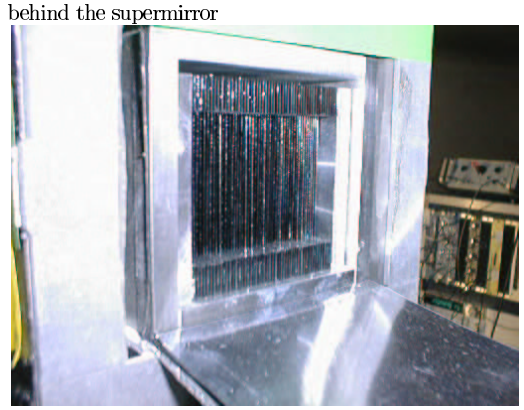

Figure 2.27: Closeup of the front of the supermirror showing its arrangement of stacked, magnetized glass plates, the data acquisition can be seen in the background

2.8 The Neutron Backmonitor

2.8.1 Requirements

The last element of this experiment in the beam is the ^{3}He/H ion chamber used to monitor the transmitted neutron beam (figure 2.28). This device was used to measure the neutron energy spectrum in current mode on a pulse by pulse basis. This device was used in the experiment for a number of purposes, including (1) a measurement of the neutron pulse intensities, (2) a transmission detector for the ^{3}He spin filter[13] and supermirror analyzer[14], and (3) a measurement of the time of flight energy scale of the neutrons.

The following requirements led to the choice of a ^{3}He-based ion chamber: (1) a well-understood detector efficiency as a function of neutron energy over a large energy range(meV to 1 eV), (2) low sensitivity to gammas, (3) and no γ-ray production within the device itself.

2.8.2 Gas Mixture Design

Neutrons are detected through the reaction:

$$n +^3 \mathrm{He} \rightarrow^3 \mathrm{H} + p + 0.78\mathrm{MeV} \tag{2.20}$$

The kinetic energies of the triton and proton ionize the gas as they slow down. The charge carriers from the ionization process are then accelerated by an electric

[13]Monitoring transmission through the ^{3}He spin filter was a measurement of the neutron beam polarization.

[14]Monitoring transmission through the supermirror was used as a measurement of the RF spin flipper efficiency.

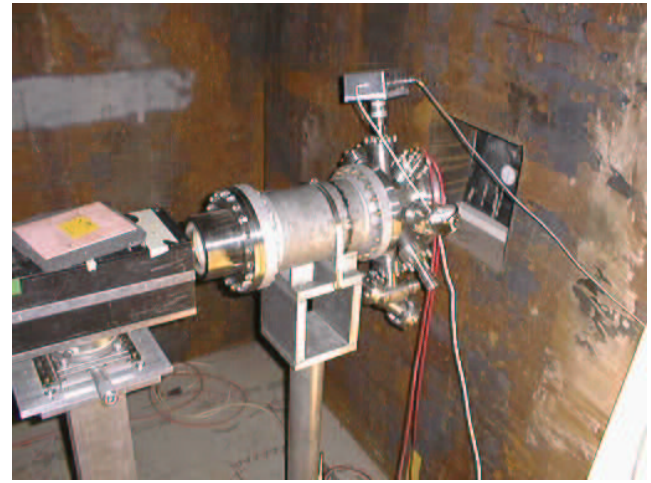

Figure 2.28: The beam path runs from left to right, the back monitor is on the right, the supermirror is on the left

potential to electrodes located at several positions along the beam and the current on the electrodes is measured. There are several concerns which determine the gas mixture and the geometry of the device. All of the neutrons over a wide range(meV to eV) of energies need to be captured. This sets a lower limit on the total amount of ^{3}He intercepting the beam. The maximum density of ^{3}He is set by acceptable size of the 'wall effect' on the front vacuum window (the range of the proton and triton within the chosen gas mixture needs to be small compared to the reaction volume or much of the energy will be lost in wall effects as the proton and triton collide with the electrodes). The mobility of the ions needs to be large so that all of the ions

produced are quickly collected. Higher electric potentials will accelerate ions faster but will limit the gas mixture to one with a high breakdown voltage.

The gas mix was chosen to be 0.5 atm ^{3}He and 3 atm H_2. Hydrogen has low Z so the γ sensitivity is low and the mobility of its ions within itself is 13.0 $\frac{cm^2}{sec*V*atm}$[15]. 3 atm of hydrogen still stops the proton and triton within a few cm, while it has a high breakdown voltage of $62\frac{kV}{cm}$.

The neutron absorption cross-section on ^{3}He is so high that the large scattering cross-section on H does not significantly effect the operation of the device.

2.8.3 Mechanical Design

The ion chamber consists of ten 3 inch diameter independent longitudinal segments. Each segment has a collection electrode sandwiched between high voltage electrodes and guard rings. A side view of the assembled monitor is shown in figure 2.29. Expanded views of the separate plates are shown in figures 2.30 through 2.32.

The collection electrodes are thin(about .030"thick), 3" diameter Al sheets supported by insulating ceramic rings. The electrical connections are made mechanically by crimping wires under the screws which secure the electrodes to the rings.

The high voltage electrodes are Al sheets identical to the collection electrodes. The HV electrodes are clamped to copper rings. Each copper ring is brazed[16] to a copper wire which makes the electrical connection to the high voltage input.

The half voltage guard rings are the same as the full voltage plate except they lack the electrode. Located halfway between the HV plates and the collection plates

[15] This is an order of magnitude larger than for other gases.

[16] A solder connection would melt during the high temperature bake of the ultra high vacuum chamber

and biased at half the voltage of a HV plate, they serve to create a more uniform electric field.

This geometry of alternating collection and HV plates causes the voltage along the axis of the chamber to form a sawtooth potential that varies from 0 V to the high voltage setting(usually 3 kV).

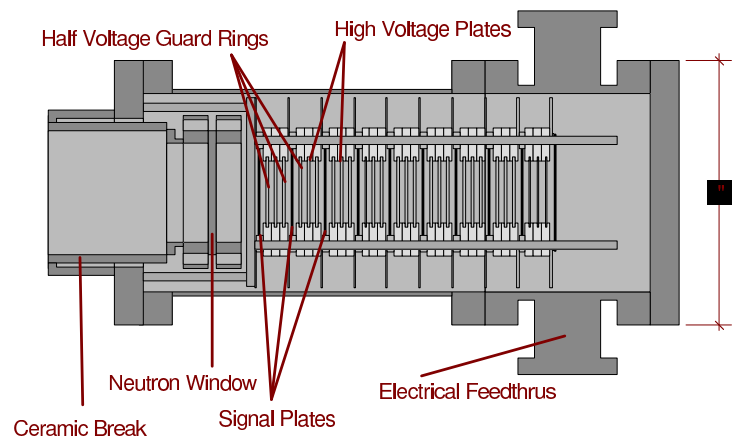

Figure 2.29: Backmonitor assembly

The electrodes are separated by ceramic spacers suspended on rods and the whole assembly is enclosed in an Al UHV(ultra high vacuum) chamber to protect the purity

[redacted]

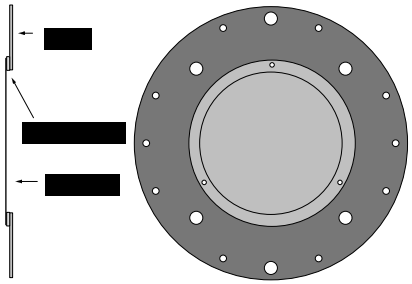

Figure 2.30: Collection plate

[redacted]

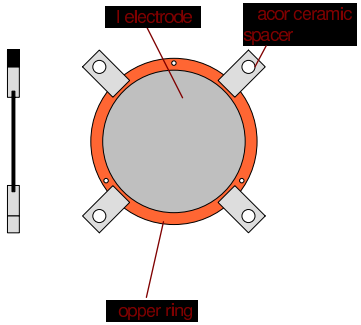

Figure 2.31: High voltage electrode

[redacted]

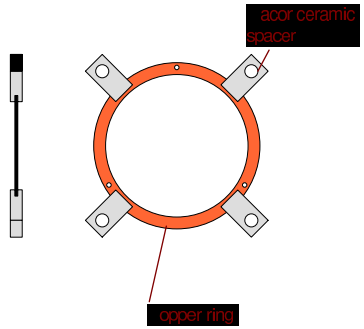

Figure 2.32: Half voltage guard ring

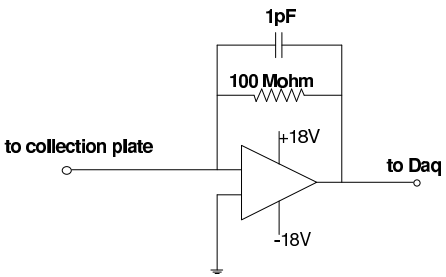

Figure 2.33: Preamp

of the gas mixture. The chamber was baked to 150 °C and evacuated to 10^{-8} torr, before filling it with .5 atm of 99.95% pure ^{3}He and 3 atm of 99.9999% pure H_2. The front window of the UHV chamber is itself a high voltage plate in order to utilize the ions produced in the front of the chamber. The rear of the chamber has UHV connections for the signal outputs, the high voltage inputs (the full and half voltages for the plates and guard rings are fed in separately), a pressure gauge, and the pumpout port.

Due to the mechanical difficulty of making the high vacuum/high voltage window the space between the front window and the first collection plate does not conform exactly to the other plate spacings, nor does it have a guard ring. Similarly, mechanical constraints on the electrical connections prevented placing a high voltage plate and half voltage guard ring after the last collection plate. These "end effects" make it difficult to include plates one and ten in quantitative comparisons between the performance of the other plates.

The thin Al of the electrodes allows the neutrons to pass through mostly unattentuated, while the high cross section of the ^{3}He absorbs the neutrons between the plates. The resulting ions are accelerated by the sawtooth electric potential of the high voltage plates to the nearest collection plate. These ions produce current signals in each collection plate. The current signals are then each an input of an I→V amplifier (figure 2.33) similar to those used for the photodiodes in the gamma detectors. The collection plate signals exit the chamber through two ports(plates 1 to 5 through one, plates 6 through 10 through the other). The amplifiers are built on circuit boards that are enclosed in metal housings mounted directly to the electrical feedthrough ports of the chamber. The voltage signal outputs from these amplifiers are then read by ADCs of the data acquisition system.

2.8.4 The Time of Flight Neutron Spectrum and the Aluminum Bragg Edges

Figure 2.34 shows the signal from all ten of the ion chamber plates. The feedback resistor on the preamps for this data was 100 MΩ. The largest signal is from the front plate and the signal size shrinks with each following plate.

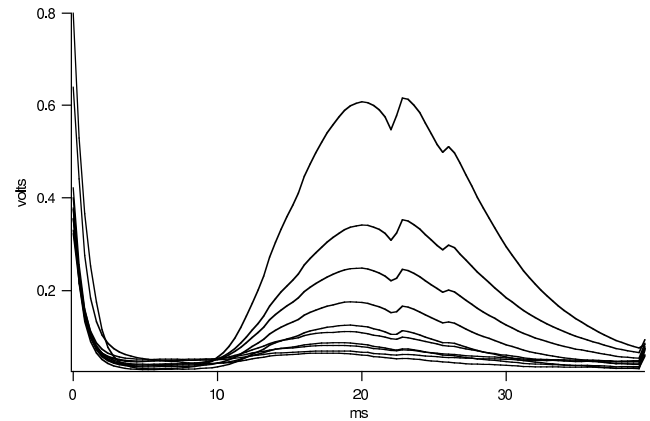

Figure 2.34: Current mode signal from all 10 plates in the ion chamber as a function of neutron TOF

The notches in the spectrum are due to neutron diffraction from Al windows in the beam. Due to Bragg scattering, the total neutron cross-section of polycrystalline materials possess discontinuities as a function of neutron energy. When the deBroglie wavelength of the neutrons is equal to the spacing between the planes of the crystal, interference patterns are formed. Peaks in the scattering cross-section leave notches in the spectrum of the transmitted neutrons. These are Bragg edges. Knowledge of

the neutron energies at which Bragg edges occur allow us to accurately relate the TOF spectrum to the energy spectrum.

By increasing the sampling rate in the DAQ, a data run (run00001919) was taken with the time resolution increased by a factor of ten (to .04 ms per sample) so that the position of the Bragg edges in the spectrum could be measured accurately. Figure 2.35 shows the data taken in this run. The notches centered at 25.76 ms and 22.32 ms are respectively the 200 and 111 Al Bragg edges generated by the vacuum and spin flipper windows in the beam. Time of flight is proportional to wavelength of the neutron (λ is proportional to $\frac{1}{v}$ which is in turn proportional to $\frac{t}{L}$ and L, the length of the flight path, is fixed). So the time of flight of these two edges are separated by a factor equal to the plane separation in the crystal or $\frac{\sqrt{3}}{2}$ (within the .04 ms time resolution). The length of the flight path can be calculated by the following method. The largest source of uncertainty in this method is in the crystal spacing of Al.:

$$2d\sin(\theta) = n\lambda \tag{2.21}$$

$$\lambda = \frac{h}{p} = \frac{ht}{mL} \tag{2.22}$$

$$\theta = \frac{\pi}{2}; d = .46 \pm .01\text{nm}; n = 2 \tag{2.23}$$

Therefore,

$$L = 22.21 \pm .49\text{m} \tag{2.24}$$

The Bragg edges are discrete changes in the neutron cross-section. Two effects create the observed slope in the Bragg edge. One is the ion chamber's response time ($\approx 125\mu$s). The other is the distribution of neutron energies emitted in a given TOF window due to the moderation process. Because the identical slope appears in the γ-detector signals(which have a rise time on the order of nanoseconds) it

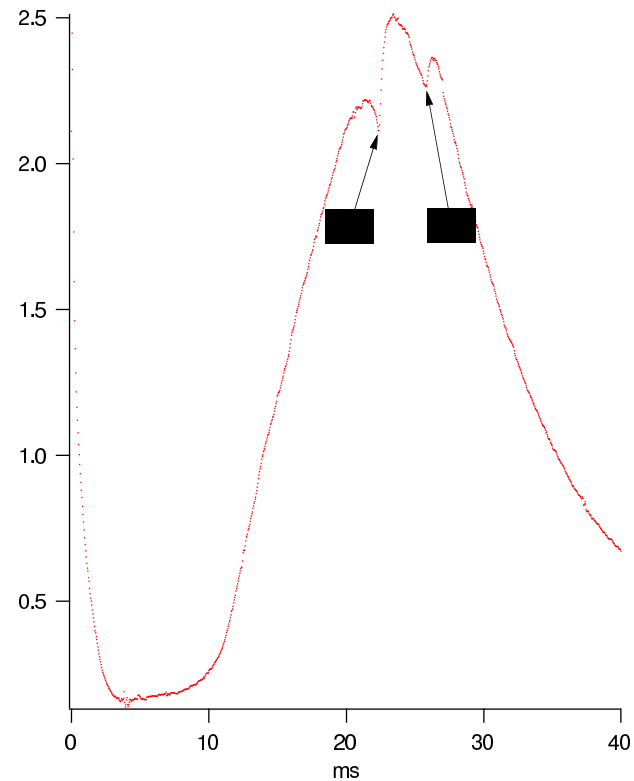

Figure 2.35: Al Bragg edges

can be determined that the primary cause of the slope is the energy width of the neutron spectrum. All that can be said experimentally about the response time of the chamber is that it is fast compared to the 400μs sampling of the γ-detector signals.

2.8.5 γ Insensitivity

To show that the detector signal is due to neutron capture only , we performed two experiments.

First, we placed a 2.4 mm thick sheet of ^{6}Li plastic (17.1% Li by weight) intersecting the neutron beam in order to stop the neutrons without creating gammas[17]. Figure 2.36 shows the response of the front plate to the shielded beam in red; the blue trace is an example of the unattenuated neutron spectrum for comparison.

The peak at low time of flight is due to the fast neutrons penetrating the ^{6}Li absorber. The black circles which overlap the red line, are a calculation of the ^{6}Li absorption effect using the following method:

$$\frac{\text{Transmitted}}{\text{Incident}} = e^{-nl\sigma_a} \tag{2.25}$$

where

$$n = 6.5 * 10^{21} atom/cc \leftarrow {}^6\text{Li number density} \tag{2.26}$$

$$l = 2.4mm \leftarrow \text{absorber thickness} \tag{2.27}$$

$$\sigma_a = \sigma_o \frac{v_o}{v} \tag{2.28}$$

[17]Neutron absorption in ^{6}Li produces ^{3}H and ^{4}He, the gamma branch is smaller by 10^3.

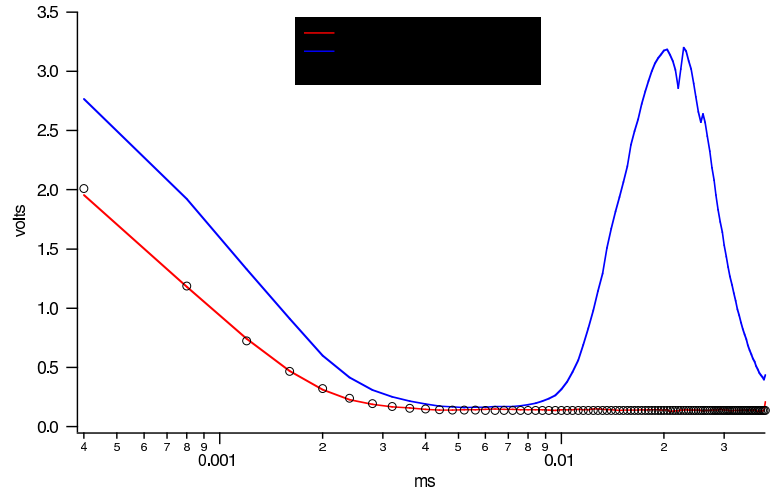

Figure 2.36: Signal from the first plate when a ^{6}Li absorber intersects the beam

$$\sigma_o = 938 \text{ barns} \leftarrow \text{n } {}^6\text{Li absorption cross section} \tag{2.29}$$

$$v_o = 2200 \frac{m}{s} \tag{2.30}$$

$$v = \frac{L}{t} = \frac{22.21m}{t} \tag{2.31}$$

Based on this agreement we conclude that the response of the ion chamber to neutrons is understood over 4 orders of magnitude in neutron energy.

Secondly, we attempted to measure the gamma sensitivity of the chamber by placing a 1 cm thick B_4C absorber (boron carbide), essentially converting every neutron in the beam into a 0.5 MeV gamma ray. However, we found that the

response of the chamber to these gammas was so weak that it was buried below the electronic noise. We can place a bound on the size of the signal that a 0.5 MeV γ-ray produces in the ion chamber to be less than 0.3% of the signal of a thermal neutron. But the actual γ-ray sensitivity is probably much lower.

2.8.6 The High Voltage Plate Bias

It is also important to know if all of the ions produced are collected on the electrodes without gas amplification (In other words, that the chamber is operating in the ionization region). As the bias is increased more and more ions reach the collection plates until all the ions produced are collected. If too much bias is applied, first amplification and then arcing will occur. Figure 2.37 shows the maximum signal voltage plotted against the bias. With our high voltage supply we were unable to overbias the chamber. We generally operated the chamber at 3kV which is far into the flat region.

2.8.7 ^{3}He Attenuation

It is also important to know how the neutron TOF spectrum changes as the beam passes through the chamber as an internal check on the operation of the device. Not only is it reduced in magnitude, but the shape of the spectrum changes as well. This is because of the $\frac{1}{v}$ cross section of the ^{3}He. Only a small fraction of the very fast neutrons are captured in each plate. Therefore, the number of fast neutrons in the beam is only slightly attenuated as the beam progresses, which means that the small fraction of fast neutrons captured changes slightly with depth in the chamber. On the other hand, the slow neutrons are heavily absorbed in the ^{3}He which leaves a smaller and smaller number of slow neutrons for each succeeding plate to measure.

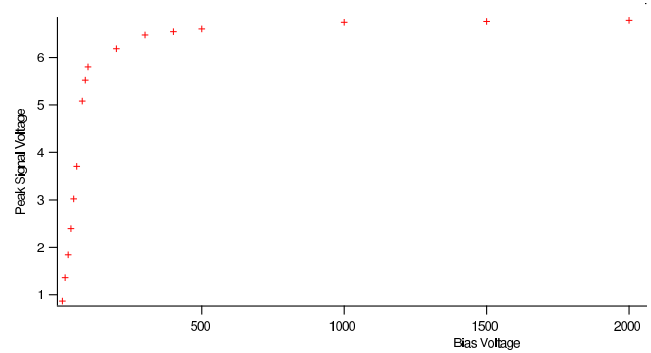

Figure 2.37: Peak signal in Plate 1 as a function of bias voltage

The calculation for the change in the beam is similar to the preceding ^{6}Li shielding calculation and is shown below:

$$s_x(t) = e^{(-.095-n(\Delta L)\sigma_a)(x-2)} s_2(t) \tag{2.32}$$

$s_x(t)$ is the spectrum of plate x. The number density of the ^{3}He is $n = 1.34 * 10^{19}$ atom/cc. ΔL is the plate spacing, 2.5 cm. The constant. 0.095, is an energy independent attenuation due to the neutrons scattering out of the active region of ion chamber from the Al signal plates and the H_2 gas. σ_a is determined from the equation 2.33 and equation 2.34.

$$\sigma_a = \sigma_o \frac{v_o}{v} = \sigma_o \frac{2200\frac{m}{s}}{v} \tag{2.33}$$

$$v = \frac{L + x(\Delta L)}{t} = \frac{22.21 + x(.025)m}{t} \tag{2.34}$$

The velocity needs to be scaled because the energy correspondence to the TOF spectrum is slightly different for each plate.

Figure 2.38 shows the comparison between the measured signals and the calculated estimate. These calculations have been done for plates number 2 through 5 and are normalized to plate 2. The black crosses are data; the red lines are calculation. In the calculation above, it was assumed that the plate spacing is identical. It was normalized to plate 2 in order to avoid a correction for the complex geometry of plate 1. This calculation takes into account the attenuation in the Al windows, the "wall effect", or neutron scattering in the H_2 only indirectly in the constant term in the attenuation model.

2.8.8 Noise and Linearity

In general, noise in the ion chamber signals was caused by one of three sources: microphonic noise, pre-amp noise, or EM pick-up. Of the three, the pick-up was by far the worst because many electrical signals, particularly the RFSF, were correlated with the beam T_0 and were not averaged out by the pulse summing of the DAQ.

The best assessment of the linearity of the ion chamber is actually the analysis of the ^{3}He transmission data in "NPDGamma Polarized ^{3}He Spin Filter Technical Note" which was taken using this device. These analysis results of the ^{3}He were presented earlier in this chapter. The ^{3}He transmission varies over a large range and differs from theory by less than 1% over the range of 2 meV to 10 meV.

At its peak, the neutron beam intensity was $1.12*10^6 \frac{neutrons}{pulse\ second}$. This compares to the first plate's voltage of about 0.6 V and the tenth plate's voltage of about 0.06 V. The feedback resistor on the preamp was 100 MΩ, so one 1 uA of input yields 1

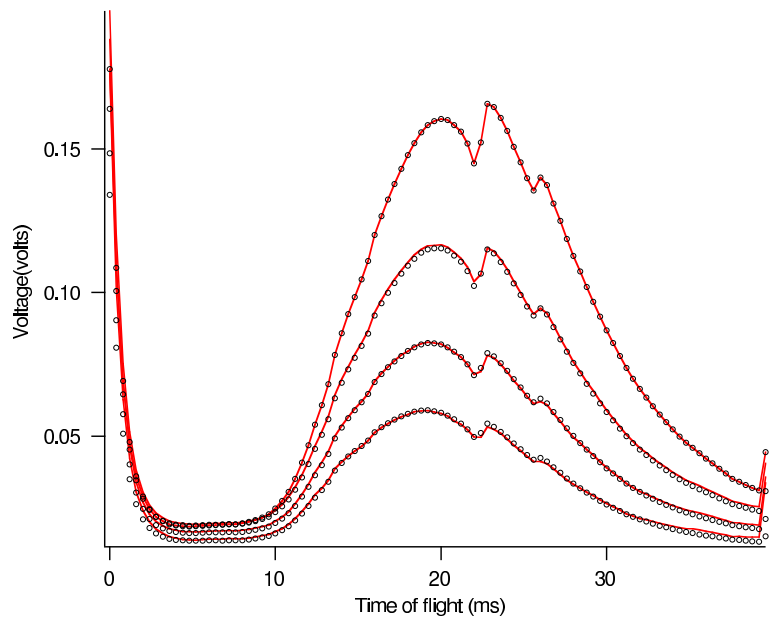

Figure 2.38: Evolution of the spectrum from plates 2 to 5. The red lines are signals from the ion chamber. The black circles are the attenuation calculated from equation 2.32.

V of output. Therefore, the sensitivity of the ion chamber is on the order of 50 to 500 fC per neutron depending on the plate and the neutron energy or 50 to 500 nV per neutron/s

2.9 The Data Acquisition

The data acquisition system is an assembly of electronics and computers which recorded the output of the various instruments (photodiodes, ion chamber, etc) and organized it into a readable form. Data recording was performed by a MVME167 VME crate controller using an array of peripheral boards (ADCs, scalers,TTL inputs, etc) it controlled through a VME bus. The distribution of trigger signals for these inputs was done in hardware by NIM modules and a HP function generator. A Linux PC controlled the system and sorted and stored the data. A functional diagram of the data acquisition system is shown in figure 2.39. Each of the components are explained in detail in the sections below.

2.9.1 The System Requirements

There were essentially two sampling rates in this experiment. The first category was data which only needed to be sampled once per 20 Hz accelerator macro-pulse. This included TTL digital signals like the spin flipper on/off state and the charge integrating ADC which sampled the proton pulse signal (A scaler was also connected and run in this group, but wasn't needed.) The second category was the analog data sampled at 2.5 KHz for 40 ms per macro-pulse synchronized with the T_0 signal from the accelerator. This sampling rate has a 400 μs resolution. We sampled some runs at a 40 μs resolution (see figure 2.35) which is faster than the 100+ μs energy resolution

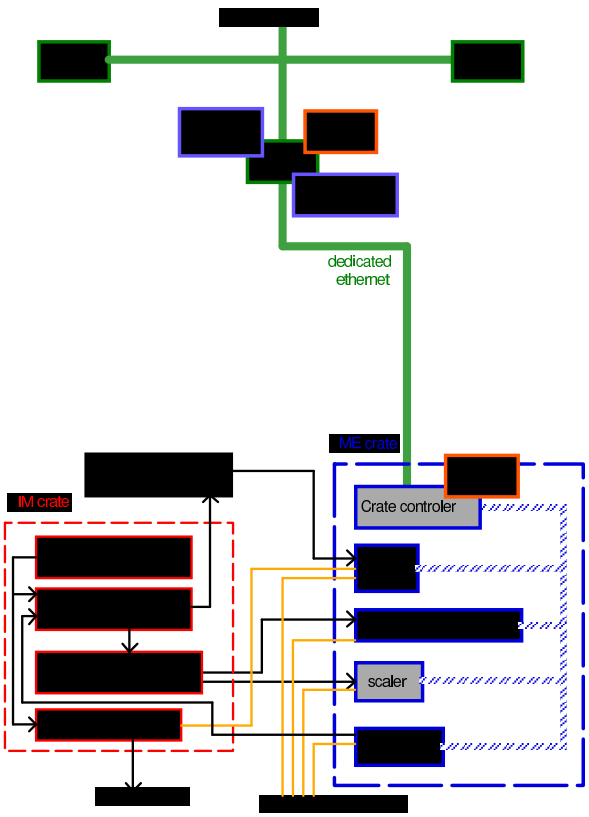

Figure 2.39: DAQ hardware diagram: Gray boxes represent hardware; Those with red borders are NIM modules; blue borders are VME; green borders are PCs. Brown boxes represent software; those with blue borders are Perl code; red borders are C code. Black connections are single pin BNC/lemo; yellow are two-pin lemo; green is ethernet; the blue hashed line is the VME bus.

of the neutron moderator. There is no information gained in sampling faster than the moderator resolution and we saw no features faster than the sampling resolution of 400 μs. The slightly lower sampling rate was chosen in order to reduce the bulk of the data files; twenty-four four byte channels, one hundred times per macropulse is 19.2 $\frac{kilobytes}{pulse}$ or about 384 $\frac{kilobytes}{second}$. To acquire the necessary statistical error, runs had to be on the order of eight hours. This means we needed provision for storing a few tens of gigabytes over the course of the experiment.

2.9.2 The Data File Format

A depiction of the file format is shown in figure 2.40. Each macro-pulse begins with a header which includes time and date. Then the once per pulse data followed which included one 2 byte charge integrating ADC, 50 fC per bit, one 4 byte scaler, and 4 bytes of digital I/O, each bit a 0 to 5 volt TTL signal, and finally the 24 ADC channels for each of one hundred time bins. The ADCs were sampled a thousand times per accelerator pulse, but they were averaged in groups of ten, creating the one hundred time bins. Each channel and bin consisted of a two byte identifier and two bytes of ADC data. The ADC measured -10V to +10V full scale and stored the data as two byte integers. The data acquisition system limited the length of each file to six thousand Tzeros, or about 300 seconds(five minutes) at 20 Hz. Data runs were divided like this in order to keep the files to a manageable size and to prevent the loss of large amounts of data in the event that any single file was corrupted. Each of these "runfiles" was given a distinct identifying number reflected in the format of the file name, "run00000000". An archive of all of the experimental data is kept on CD at LANL.

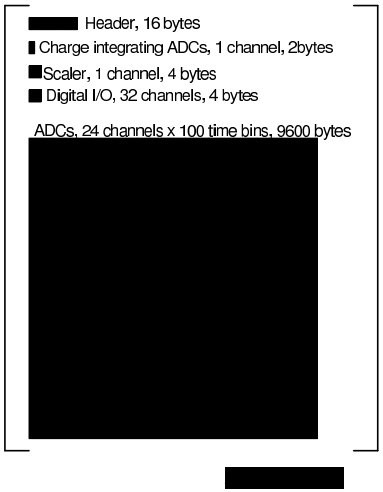

Figure 2.40: Run file format

2.9.3 The Triggering System

Figure 2.41 is a flowchart depicting the system. The T_0 signal is a signal from the accelerator synchronized to the proton pulse striking the spallation target. It is a 20 Hz TTL pulse and therefore was easily simulated for taking background noise data when the accelerator wasn't running.

The "GO" from the PC is a signal provided by the data acquisition system itself. It is a TTL signal which switched high when data taking began and switched low

again at the end of data taking. By requiring that the "GO" and T_0 signal be in coincidence the beginning of the data taking always coincided with the beginning of a macro-pulse, never in the middle.

This coincidence signal was then branched four ways. The first went to a HP function generator. With each trigger by the coincidence signal, the function generator generated a burst of one thousand 25 kHz TTL pulses. This signal was used to trigger the transient digitizer boards (the fast ADCs) one thousand times between 0 and 40 ms immediately following the spallation production of the neutrons. If the accelerator was running steadily at 20 Hz, which it usually did, this meant that data were taken for 40 ms followed by 10 ms of dead time.

The second and third branches went to gate generators for the charge integrating ADC and the scaler. These devices were read only once per macro-pulse and the gates were tailored to the signals they read.

The fourth branch went to the spin state generator, a schematic of which is shown in figure 2.42. This was a custom built system of logic gates which generated the control signals for the radio frequency spin flipper. Each trigger from the coincidence signal advanced a four bit clock through one of sixteen states. The spin state generator could be configured by sixteen on board jumpers to turn the spin flipper on or off in each state of the clock. This is how the eight step sequence, ↑↓↓↑↓↑↑↓, could be generated.

A timing diagram for the triggering system is shown in figure 2.43.

2.9.4 The Computers

The DAQ employs several PCs for different purposes. NPDGPC is a linux box which was used to communicate with the VME crate and as a switch yard for the

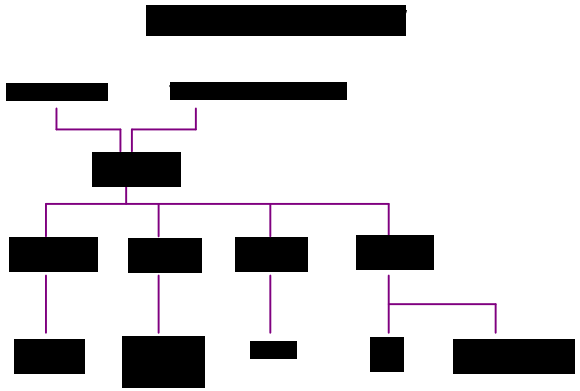

Figure 2.41: DAQ trigger flowchart

Data Acquisition NIM modules

manufacturer	model	Description	Number
Lecroy	688AL	NIM-TTL level adapter	1
Lecroy	364AL	4-fold logic gate	1
PS	794	quad gate delay	1
Lecroy	222	dual gate generator	1
—	—	RFSF spin state generator	1

Table 2.4: Triggering system components

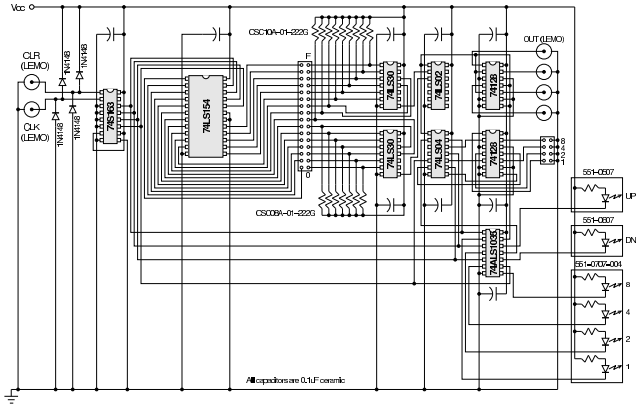

Figure 2.42: Schematic diagram of the spin sequencer

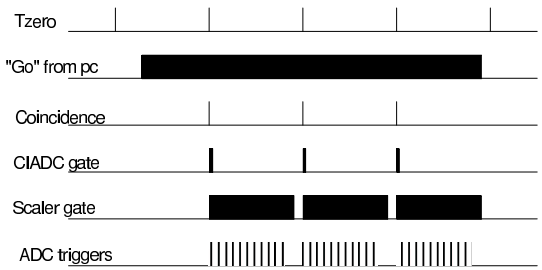

Figure 2.43: DAQ timing diagram

Data Acquisition VME boards

Manufacturer	Model	Description	Number
Motorola	MVME167	Crate Controller	1
Alphi	ADM31	ADC Controller	1
Alphi	ADM-24D	ADC boards	2
Acromag	AVME945X	TTL I/O	1
Lecroy	1182ADC	Charge integrating ADC	1
Joerger	VSC16	Scaler	1
WEISCZ	VDIS	Bus spy	1

Table 2.5: Daq VME boards

data. In order to prevent outside traffic from interfering with the operation of the data acquisition system, a dedicated ethernet network connected the NPDGPC and the VME crate . A second network connection allow NPDGPC to communicate to the internet.

LENEUT was another linux box which served as a temporary storehouse for the data. Other machines, such as the LENMAC1 which is a Macintosh that was used for much of the online analysis, could then connect to LENEUT and view the data without disturbing NPDGPC while the DAQ was running.

2.9.5 The VME Hardware

VME is a bus standard. A VME system consists of a crate controller and possibly one or more peripheral boards. The MVME167 is the VME crate controller used in this system. It serves the functions of the motherboard, RAM and ethernet card in a PC. Each of the peripheral boards appears as an address space on the VME bus.

The particular address is usually set by jumpers on the peripheral boards themselves. The crate controller runs a program, in this case written in C, which controls and monitors the peripheral boards by writing to and reading from the VME address space. This program is loaded with the operating system, PSOS, as a single object when the machine boots via TFTP from the the linux box, NPDGPC.

With a single exception, our peripheral boards are simply electronics or memory attached to the VME bus. The exception is the ADM31. It is the fast 48 channel, 16 bit analog-to-digital converter used to measure the photodiode signals and anything else sampled at 25kHz. Each of these boards has its own CPU and memory and runs a firmware code that it downloads on the VME bus from the MVME167 during the initialization. The advantage is that ADM31 can take control of the VMEbus and load its data into the memory of the MVME167 without interrupting the crate controller's code. Since the bottleneck in the entire operation is the bandwidth of the ethernet(the VMEbus is about eight times as fast), this setup allows the MVME167 to continuously send out data without having to stop and read the ADCs.

2.9.6 The Software Components

Three separate processes were running on NPDGPC and one on the MVME167. A flowchart of the three primary codes and their interactions is shown in figure 2.44.

The user seated in front of NPDGPC runs a Perl program called *daqgui_II*. This program creates a graphical user interface which controls from one to three other programs. It must run a C program called *b_II.c*. This program receives data from the VME and writes both the raw data and a condensed summary into files. The other process was a Perl script, *auto_to_scp* , set to run at low priority and to scp data to temporary storage on LENEUT. A C program, *npdg.c*, was downloaded by

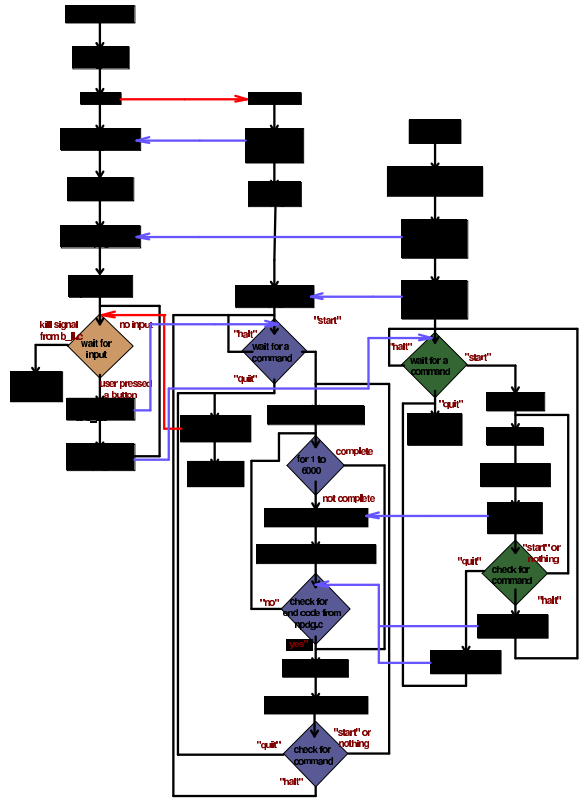

Figure 2.44: Data acquisition code flowchart: Brown is the daqgui_II program; blue is the b_II.c program; green is npdg.c. Black arrows represent progression within a program; blue arrows are communications through socket connections; red arrows are communications through the Unix shell.

the MVME167 VME crate controller when it was booted.

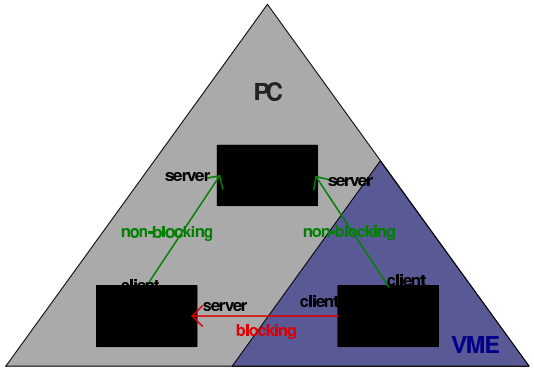

Figure 2.45: DAQ socket diagram

Ethernet connections were managed by a socket interface(see figure 2.45). The graphical buttons of the *daqgui_II* program were linked to functions which send commands via single characters. The programs, *b_II.c* and *npdg.c*, checked their connections with *daqgui_II* once per macro-pulse to receive their instructions. The program, *npdg.c*, determined the length of the macro-pulse through the aforementioned hardware triggering. The program, *b_II.c*, determined the length of a macro-pulse by the length of the data transferred by *npdg.c*

The *auto_to_scp* script determined which files to scp and tape by how *b_II.c* named the files.

Data Acquisition Software

Program name	Machine	Language	Parent Process
daqgui_II	npdgpc	Perl	none
b_II.c	npdgpc	C	daqgui_II
auto_to_scp	npdgpc	Perl	daqgui_II
npdg.c	MVME167	C	none

Table 2.6: Daq software

2.9.7 The Signal List

In the final setup for the asymmetry measurements, eighteen signals were recorded. Several empty channels were recorded as well because it was convenient to be able to keep the same file format even if signals were added in the middle of the experiment. A complete list of recorded signals and their location is provided in table 2.7.

We used the charge integrator to measure the proton current from a signal produced by a toroidal pickup coil. It is a signal provided by LANSCE and was advertised as being proportional to the neutron beam current. We found this to be true to 10% accuracy (we never found the cause of this discrepancy and LANSCE did not give us permission to investigate). There were other easier methods to normalize the beam flux which were pursued (the γ-detector sum cut discussed in chapter 3).

The scaler was originally intended to be used with a fission chamber beam monitor, but it wasn't used in this experiment.

Two channels of the digital IO module were used. One was to record the spin state. The second recorded (and created) the "GO" triggering signal.

15 channels of the ADCs were used. 4 were for the γ-ray detectors. 10 for the 10 plates of the backmonitor. And one was used as an auxiliary spin state measurement.

This was necessary because a timing problem was discovered in the testing phase of the data acquisition[18]. The fear of finding another such problem and the uncertainty in the amount of beam time we had to correct it if we did led us to also record the TTL of the spin state generator in the ADCs as well as the digital IO. Because all of the ADCs are sampled at the same time, driven by the same hardware trigger, we knew that an undiscovered timing problem wouldn't make the data irrecoverable. There was a concern this could create a systematic error through cross talk in the ADC channels. We felt this was unlikely because the signal was slow (TTL) and small(we attenuated by a factor of 10) and was connected to the ADCs in the VME crate via a separate board than the detector signals. In any case, an upper bound of the false asymmetry created by all parts of the system had to measured in order to make any valid asymmetry measurement; this procedure is discussed in chapter 4.

[18]The data recorded once (charge integrator and digital IO) was being linked one pulse out of phase with the ADC data. It was discovered when the spin flipper efficiency was tested with the eight-step sequence for the first time, having been overlooked with the help of a sign error in the two-step pre-run testing. Although, we didn't find any further bugs, the whole interrupt handling routine will need to be rewritten anyway. The crate controller for NPDGamma will run Linux instead of PSOS as the operating system. Linux has advantages of cost and compatibility, while the advantages of PSOS, realtime control and very small size, aren't necessary in our experiment. PSOS was used originally because LANSCE had a site license and Linux wasn't available in VME crate controllers at that time.

Signal List

charge integrator		
channel number	signal name	purpose
0	proton current toroid	a measurement of the proton beam flux on the spallation target
digital IO		
channel number	signal name	purpose
1	spin state	spin state
9	"GO" from PC	trigger coincidence signal
ADC		
channel number	signal name	purpose
0	up detector	These are the photodiode signals
1	right detector	
2	down detector	
3	left detector	
12	1st plate	These are the ion chamber siganls
13	2nd plate	
14	3rd plate	
15	4th plate	
16	5th plate	
17	6th plate	
18	7th plate	
19	8th plate	
20	9th plate	
21	10th plate	
22	spin state	Extra spin state recording

Table 2.7: Signal list

CHAPTER III

Analysis

3.1 The Asymmetry, A_γ

The γ-ray energy weighted angular distribution, $\Gamma(\theta)$, takes the form:

$$\Gamma(\theta, E_\gamma, E_n) = \alpha(1 + P_n(E_n)A_\gamma(E_\gamma, E_n)\cos\theta) \tag{3.1}$$

where θ is the angle between the neutron spin and the γ-ray momentum, P_n is the polarization of the neutron beam, and E_γ and E_n are the energies of the γ-ray and neutron, respectively.

Collecting data in current mode has the advantage of collecting statistics much faster than pulse counting. But it also means that the measurement of $\Gamma(\theta)$ is an integral of A_γ over E_γ and E_n. In addition, an integral of $\cos\theta$ over the finite solid angle is effectively performed by the detector.

The goal of this experiment was to measure the "integral asymmetry",

$$A \equiv \frac{1}{\langle E_\gamma \rangle \Phi_T} \int \Phi(E_n) E_\gamma A_\gamma(E_\gamma, E_n) \delta E_\gamma \delta E_n \tag{3.2}$$

or,

$$\Gamma(\theta) = \alpha(1 + P_n(E_n)A\cos\theta) \tag{3.3}$$

This chapter describes how A was extracted from the detector signal data.

3.2 The Detector Signals

Throughout this chapter the four detectors will be referred to by their respective positions in the array. The designations, up detector and down detector, refer to the detectors above and below the beam, respectively. Left and right detectors refer to the detectors to the left and right looking in the direction that the neutron beam travels. Recall that the neutron polarization is vertical.

The asymmetry data were taken with the analog-to-digital converters mentioned in the data acquisition system description of chapter 2. Each voltage signal of a detector/photodiode/preamp assembly was converted into a 16 bit integer with a quantization error of $\frac{(10\text{V})-(-10\text{V})}{2^{16}}$. Throughout the analysis calculations, this arbitrary unit of "one bit of the ADC" was used. By not scaling before the calculation, one step and one round off error is saved. Still the voltage of some things (amplifier offsets for example) are included so that a concept of the signal scales can be maintained.

The digitization in time was .04 ms, one thousand evenly spaced samples taken during the 40 ms immediately following the proton pulse on the spallation target. Inside the VME crate, the data was compressed by averaging the samples in groups of ten. Therefore, the analyzed data is one hundred 0.4 ms samples per macro-pulse.

Samples of the up detector signals for each of the three targets are shown in

figure 3.1. For both chlorine and lanthanum the cross-section of the target materials is proportional to $\frac{1}{v}$. Cadmium, on the other hand, contains a low energy resonance[1]. Plots of the ENDF data for each of these elements over the relevant energy range are shown in figure 3.2 through figure 3.4.

In the discussion of this chapter, I describe the up/down asymmetry only. The calculations for the left/right asymmetry are identical with the replacement of up and down with left and right, respectively, for the detectors only.

3.2.1 Balancing the Detector Gains

The geometric mean method for calculating the asymmetry from the U and D detector signals, described by equation 3.4, can be derived from equation 3.3. The principal advantage of using this form is that the detector gains cancel directly. The derivation of proportionality will be discussed after the derivation of its components.

$$A \propto \frac{\sqrt{\frac{U_\uparrow D_\downarrow}{U_\downarrow D_\uparrow}} - 1}{\sqrt{\frac{U_\uparrow D_\downarrow}{U_\downarrow D_\uparrow}} + 1} = \frac{\sqrt{U_\uparrow D_\downarrow} - \sqrt{U_\downarrow D_\uparrow}}{\sqrt{U_\uparrow D_\downarrow} + \sqrt{U_\downarrow D_\uparrow}} \tag{3.4}$$

where the U and D denote the signals in particular detectors and the $\uparrow$ and $\downarrow$ denote the spin state of the neutron.

3.2.2 Eliminating the Detector Offsets

Offset in the preamp and dark current in the photodiode cause DC offsets in the detector signals. These offsets do not create a false asymmetry but they do change the scale of the asymmetry measured. We measured the DC offsets by taking data

[1] The neutron capture in Cd is dominated by this s-wave resonance at 177.6 meV.

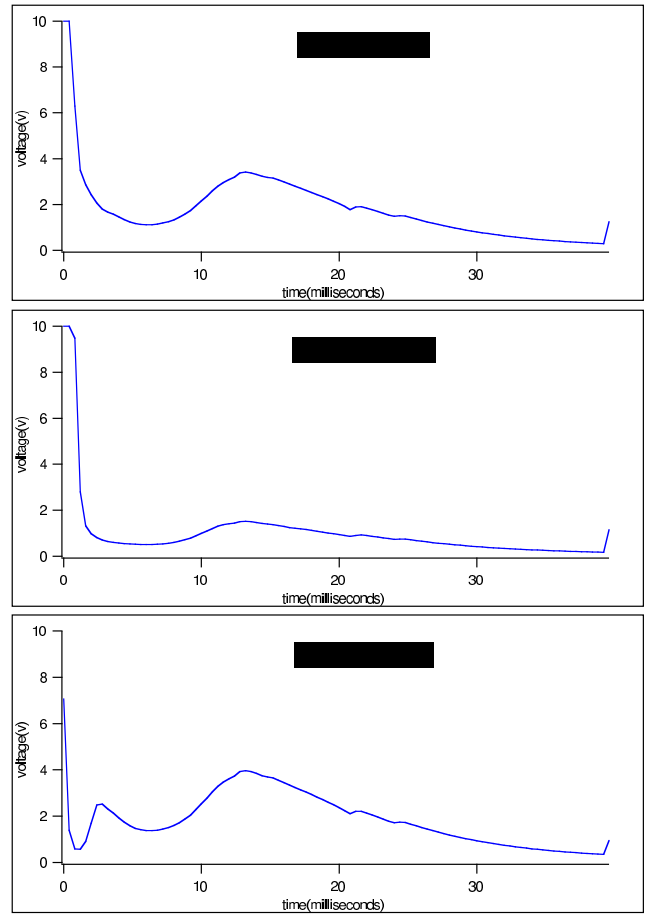

Figure 3.1: Sample detector signals

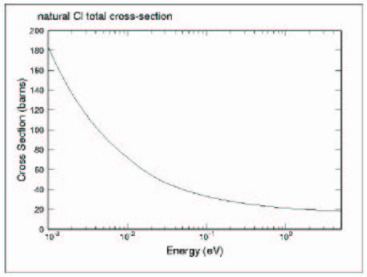

Figure 3.2: Total cross-section of natural Cl, courtesy of ENDF

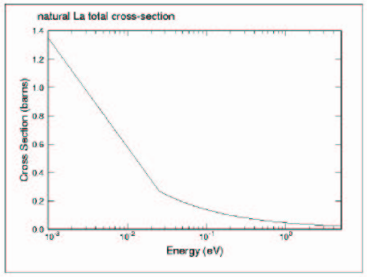

Figure 3.3: Total cross-section of natural La, courtesy of ENDF

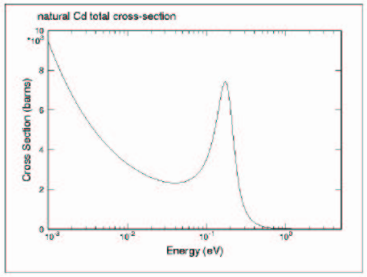

Figure 3.4: Total cross-section of natural Cd, courtesy of ENDF

with the beam shutter closed. The average offset value for each detector and run is shown in table 3.1. The corresponding average offset was subtracted from each channel before calculating the asymmetry.

The problem is not entirely eliminated because the offsets drifted over time, probably due to temperature change in the experimental cave, and we only have information on the offsets during breaks in the data taking. But the change is small, on the order of 0.1% to 1% of the signal size. And we can place a bound on this systematic error that is much smaller than the statistical error.

The effect of uncorrected offsets is described by equation 3.5. A is the physics asymmetry. $\tilde{A}$ is the asymmetry before correcting for the detector offsets. u_o and d_o are the offset signals in the up and down detectors respectively. $U_\downarrow$ and $D_\uparrow$ are the γ-ray signals in the detectors. A full derivation of this equation can be found in appendix B.

$$A = \tilde{A}\frac{1}{\left[\frac{1}{2}\left(\frac{U_\downarrow}{U_\downarrow+u_o} + \frac{D_\uparrow}{D_\uparrow+d_o}\right)\right]} \tag{3.5}$$

Essentially, the effect of having DC offsets in the detectors is to create a false γ-ray signal with no asymmetry. Since the asymmetry coefficient is a ratio between $\cos\theta$ distributed γ-rays over the isotropic γ-rays, the DC offsets change the size of the denominator. As long as the offsets are small compared to the "real" denominator, the change in the asymmetry coefficient will be small.

The offset effect can be canceled in large part by subtracting the measured offsets before the asymmetry calculation. But because of the drifts it can't be canceled entirely. If we take the largest total drift from beginning to end for the worst detector, it results in a error of 1.3% in the size of the asymmetry which is small compared to

the statistical error.

3.3 Establishing cuts on the Data

3.3.1 The Cut on the Eight-step Sequence

The data was collected with the neutron polarization flipped or not at 20Hz according to the eight-step sequence[2], ↑↓↓↑↓↑↑↓. Obviously, if the wrong number of up pulses and down pulses are summed together, then the calculation of the asymmetry will be wrong. So the first cut on the data was on the spin sequence.

The spin flipper state from eight pulses are compared with the spin sequence: ↑↓↓↑↓↑↑↓. If the spin states all agree, or agree by shifting the whole sequence to the left or right, then the eight-step is accepted. All of the pulses of a whole eight-step sequence are either accepted or rejected together. Then the next eight pulses are compared. This results in less than 1% of the data being cut.

3.3.2 The Cut on the Magnitude of the Detector Sum

We had originally planned to use the neutron back monitor to establish the cuts based on beam current. This was not possible because the fringe DC field produced by the supermirror's permanent magnet was large enough to effect the tuning of the RF spin flipper. The RF spin flipper could not be tuned without the supermirror because neutron polarization analysis was required for this task and the supermirror was the only polarization analyzer that was available. However, we couldn't make the asymmetry measurements with the supermirror in the beam path because it

[2] This sequence was chosen to eliminate linear and quadratic time-dependent drifts in the detector efficiencies. It is discussed more thoroughly in section 4.2.

Detector Offsets

cadmium runs		
detector	average ADC bits offset	average voltage offset(mV)
Up	164	50.0
Right	79	24.1
Down	75	22.9
Left	73	22.3
chlorine runs		
detector	average ADC bits offset	average voltage offset(mV)
Up	179	54.6
Right	86	26.2
Down	80	24.4
Left	83	25.3
Lathanum runs		
detector	average ADC bits offset	average voltage offset(mV)
Up	207	63.2
Right	94	28.7
Down	77	23.5
Left	132	40.3

Table 3.1: Detector offsets

produced a spin dependent γ-ray background[3]. The solution to this problem was to place a $^6Li/B_4C$ beam block in front of the supermirror. Unfortunately, this was also in front of the back monitor as well. Because we don't have information from the backmonitor during the actual asymmetry measurements, the sum of the CsI detector signals is the best measurement of the beam current and is therefore used in determining cuts. The resolution of this situation is discussed in more detail in chapter 4.

As the offset discussion of the previous section has shown, the effect of the offsets can only be ignored if it is small compared to the signal size. If the beam current is low, the detector signal will be small, but the offsets won't change. It is therefore necessary to cut on the beam current/detector signal size.

3.3.3 The Cut on the Change in Detector Sum within an Eight-step Sequence

When the accelerator starts, it does not reach full current in the first pulse. The proton current ramps up slowly which means that the neutron flux ramps up slowly as well. The geometric mean scheme of calculating the data is used to cancel the difference in the gains of opposite detectors. It does this well with the assumption that the neutron flux is not spin dependent. But during the accelerator start up, the neutron flux is increasing from pulse to pulse. Because the spin state generator starts at the same place in the sequence, the changing beam current creates a spin dependent gain. In order to eliminate this potential systematic error, a cut on the

[3]the supermirror analyzes the polarization of the beam by absorbing one spin state of the neutrons in gadolinium which produces γ-rays(i.e. the supermirror produces spin dependent γ-rays).

sum of the detector signals is necessary.

The solution is as follows. The sum of all four detector signals are integrated over one eight-step sequence. An average of the the integrated detector sums is determined. An entire eight-step sequence is then cut if one or more of the detector sum integrals differs by more than 1% from the average for that eight-step sequence.

This cut is less than 0.1% of the total data. Figures 3.5 to 3.7 show comparisons of the detector sum, the UD asymmetry, and the LR asymmetry between the uncut data and the cut data for each of the targets.

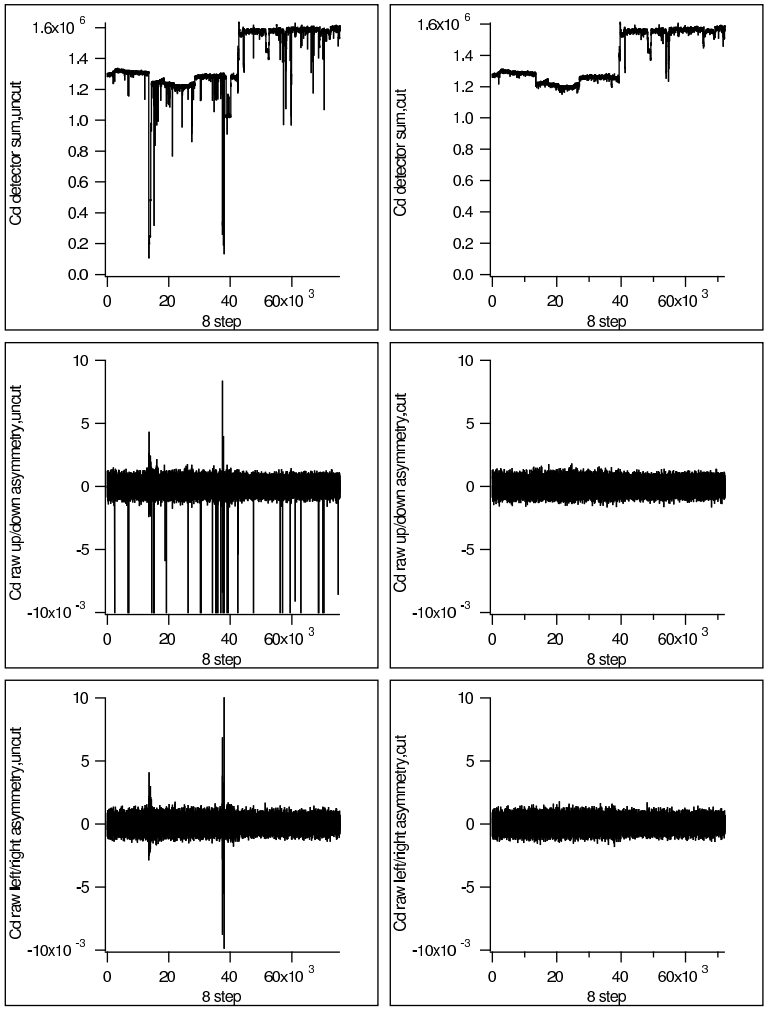

Figure 3.5: Cd detector sum cut

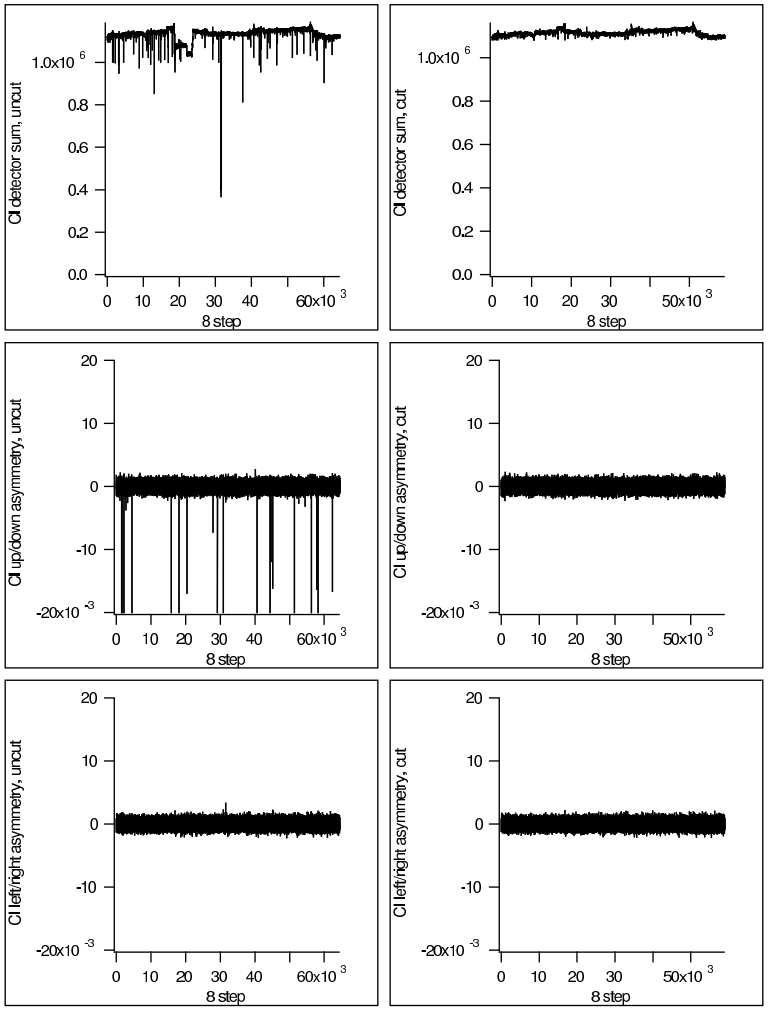

Figure 3.6: Cl detector sum cut

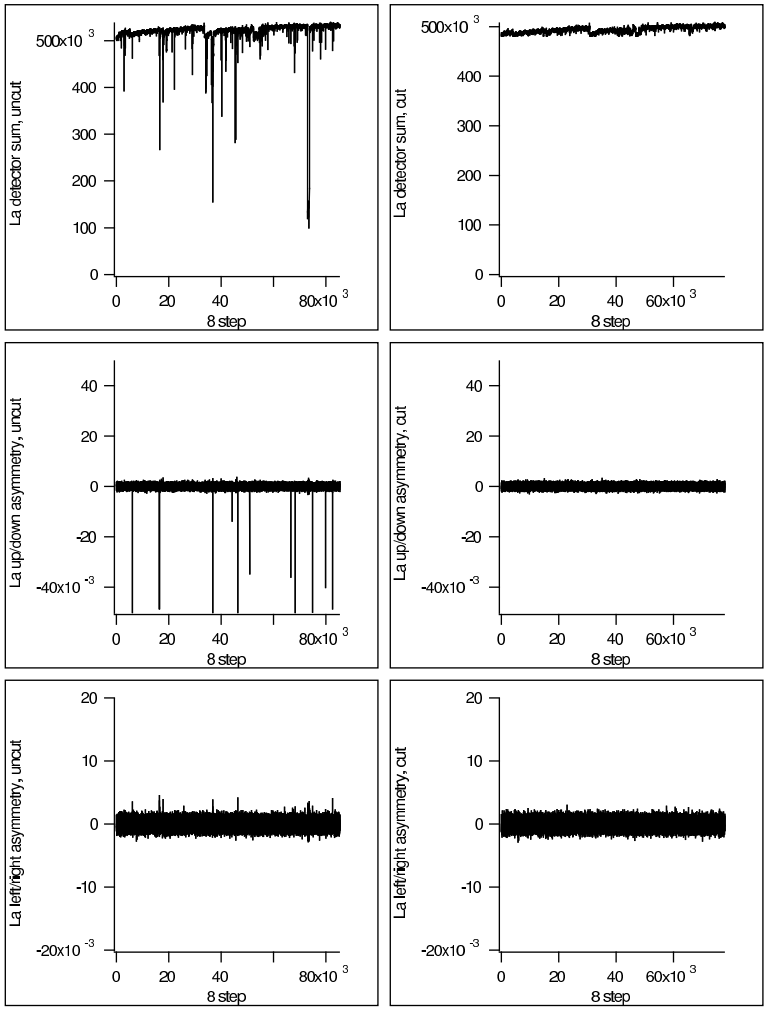

Figure 3.7: La detector sum cut

3.4 Calculation of the Correction Factors

These are factors which scale the size of the asymmetry after the measurement is made.

$$\Gamma(\theta) = \alpha(1 + P_n(E_n)A\cos\theta) \tag{3.6}$$

The polarization of the neutron beam, P_n, is composed of four factors. The polarization, P, generated by filtering the beam through the ^{3}He. The efficiency, ϵ_R, with which the RFSF maintains the polarization of the neutrons through the spin flip. The finite solid angle of the detectors is accounted for by the factor, G. And T adjusts for depolarization in the target.

$$P_n = P(E_n)\epsilon_R(E_n)GT \tag{3.7}$$

3.4.1 Polarization, $P(E_n)$

P is the polarization of the neutron beam after being filtered by the polarized ^{3}He cell. It is a function of time-of-flight and the equations which determine it are discussed in chapter two. The polarization is shown again in figure 3.8. The uncertainty of the polarization contributes a $< 2\%$ uncertainty to the scale of the asymmetry.

3.4.2 Spin Flipper Efficiency, $\epsilon_R(E_n)$

ϵ_R is the fraction of the polarization retained by the beam being flipped by the RFSF. Because the RFSF doesn't alter the beam when it is off, one spin state has the polarization of the neutrons as they leave the ^{3}He, or just P.

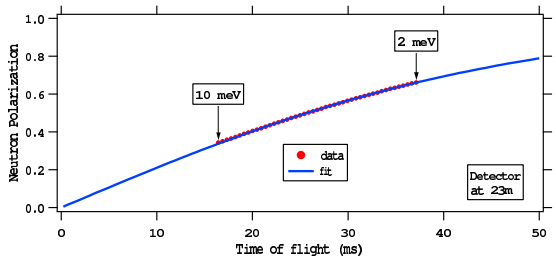

Figure 3.8: neutron beam polarization, dots represent measured polarization, the line is the theoretical prediction from ^{3}He transmission

Because the field of the RFSF isn't ideal, the spin state is not completely reversed. The efficiency, ϵ_R, is repeated from chapter 2 in figure 3.9. This means that the polarization is $\epsilon_R P$ when the spin flipper is on.

This supplies a factor of $\frac{1+\epsilon_R}{2}$ to the asymmetry calculation. Since $\epsilon_R = .98 \pm .02$ over the beam area of this experiment this yields a factor of .99±.01 in the asymmetry.

3.4.3 Detector Energy Weighted Cosine, G

Because the detector is both finite in size and has an efficiency which is dependent on angle, there is a correction factor to the asymmetry which is based upon its angular distribution. This correction factor is denoted G.

$$G = \frac{\int_0^{2\pi} \delta\phi \int_0^{\pi} \delta\theta \epsilon_d(\theta,\phi)\cos(\theta)\sin(\theta)}{\int_0^{2\pi} \delta\phi \int_0^{\pi} \delta\theta \sin(\theta)} \tag{3.8}$$

θ is the angle between the neutron spin and the momentum of the γ-ray. ϕ is the radial coordinate. The $\sin(\theta)$ is from the calculation of the solid angle. The $\cos(\theta)$

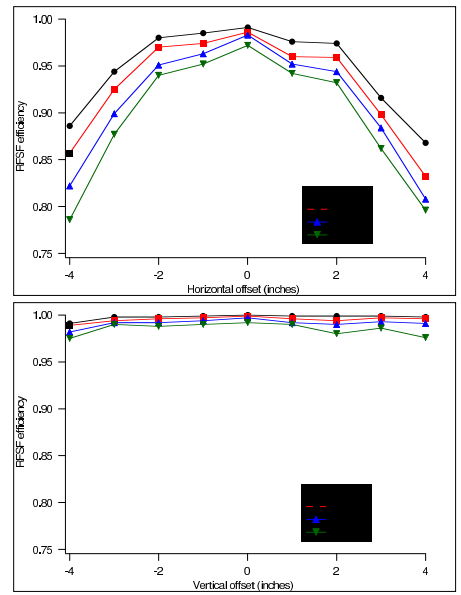

Figure 3.9: Spin Flipper Efficiency

G for each target

	^{35}Cl	^{113}Cd	^{139}La
G	.856	.861	.867

Table 3.2: Monte Carlo values of geometry correction factor, G, for each target

is a weighting due to the angular distribution of the asymmetry. The trouble in this calculation is that the angular efficiency of the γ-detector, $\epsilon_d(\theta, \phi)$, is not known very well. In a more exact calculation, $\epsilon_d(\theta, \phi)$ would also be a function of the coordinates of neutron capture as well.

In order to deal with all of these complications, a Monte Carlo approach was taken. G was calculated through a software package called MCNP[4].

This Monte Carlo program simulates the production of γ-rays by neutron capture in the target and their absorption in the CsI scintillator. The calculation determines the cosine of the relative angle between the γ-ray momentum and the neutron spin. Then an average is formed weighting each cosine value by the energy deposited in the crystal by the respective γ-ray.

G is slightly different for each target. It is the distribution of neutron captures in the target relative to the geometry of the detector that primarily determines the value of the factor G.

The Monte Carlo results are shown in table 3.2. The errors generated by MCNP were on the part per thousand level, but these are only the errors associated with the statistics of the Monte Carlo algorithm. The real error for this calculation is primarily determined by what was left out of the simulation.

There are two principal flaws in this Monte Carlo simulation. One is the assump-

[4]more specifically, MCNP4C-C00700.ALLCP02

tion that the scintillation light is collected by the photodiode with equal efficiency regardless of where it is produced in the crystal. Since the this is almost certainly dependent on θ and ϕ, it creates an systematic error in G.

The second flaw is that the ENDF[5] for ^{139}La was unavailable. The solution was to use the ENDF file for ^{35}Cl and to modify the number density of the chlorine so that the cross-section per unit length was equivalent to lanthanum. This causes the correct distribution of the neutron captures within the target but the resulting gamma spectrum is incorrect. Because the mass attenuation coefficient of CsI is fairly uniform over the energy range including the ^{35}Cl and ^{139}La spectra(see figure 3.10), the effect of substituting one spectra for the other is small. The energy weighted average absorption length of ^{35}Cl is 5.78 cm compared to 5.64 cm for ^{139}La, a difference of about 2.5%. This combined with the fact that the detector is $\approx 3\sigma$ thick for both spectra, reduces this effect to a small correction.

To help justify the plausibility of the results of the Monte Carlo, the scale of the G can be approximated by some crude hand calculations.

Assuming an ideal, radially symmetric detector(one in which $\epsilon_d(\theta) = 1$ when $\theta < \theta_{max}$, 0 otherwise) and a point source of γ-rays at the center of the target, equation 3.8 reduces to:

$$G = \frac{\int_0^{\theta_{max}} \cos(\theta)\sin(\theta)\delta\theta}{\int_0^{\theta_{max}} \sin(\theta)\delta\theta} \tag{3.9}$$

Integration yields:

[5]ENDF files are the source from which MCNP draws the physical data about the isotopes to perform the Monte Carlo calculation

Table of Contents | back to table 4

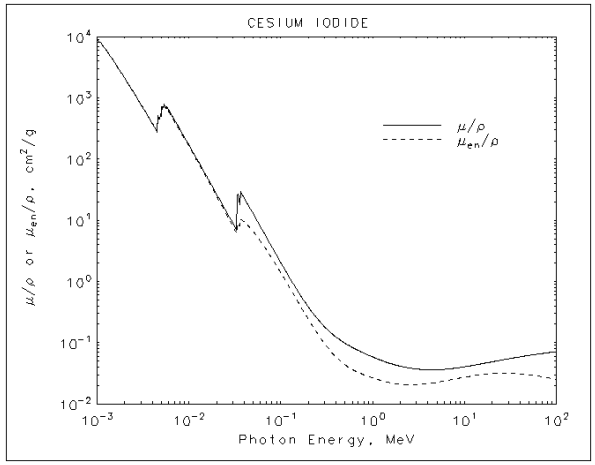

Figure 3.10: CsI efficiency, taken from NIST Physical Reference Data, $\frac{\mu}{\rho} = x^{-1} ln(\frac{I_0}{I})$

$$G = \frac{1}{2} \frac{\sin^2(\theta_{max})}{(1 + \cos(\theta_{max}))} \tag{3.10}$$

Figure 3.11 is a plot of G given by equation 3.10

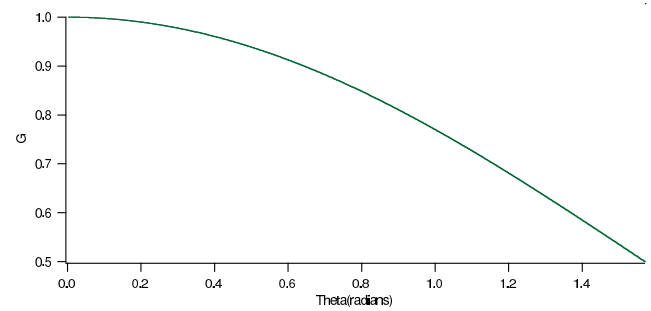

Figure 3.11: Value of the geometry coefficient, G, for an ideal detector of fixed angular width and a point source of γ-rays

As the detector narrows to an ideal point detector(i.e. only photons with $\theta = 0$ are integrated), the value of G approaches 1, its maximum value for any geometry.

As the solid angle approaches 2π (i.e. all photons less with $\theta < \frac{\pi}{2}$ are integrated or the up and down detectors are two hemispheres), the value of G approaches $\frac{1}{2}$, its minimum value.

I am quoting $\pm 5\%$ as a conservative guess on the error on this calculation. For a value of $G = .860 \pm 5\%$, this is equivalent to an ideal detector with a radial acceptance, $\theta_{max} = 36$ to 51 degrees. This seems to be reasonable and is still small compared to the statistical error of $\pm 20\%$.

3.4.4 Calculation of the Depolarization in the Target, T

The observed asymmetry is also reduced by a depolarization of the neutron beam in the target. If a neutron scatters before it captures there is a chance that the neutron will depolarize (angular momentum will be transferred between the scattering nucleus and the neutron). In general, this effect is small in the target nuclei because their absorption cross-sections are large compared their scattering cross-sections.

Table 3.3 shows the absorption[6], incoherent scattering and coherent scattering cross-sections for each target from reference[22].

If we approximate the scattering cross-section which flips the neutron spin as:

$$\sigma_{\text{flip}} = \frac{2}{3}\sigma_I \tag{3.11}$$

And the scattering cross-section which does no flip the neutron spin as:

$$\sigma_{\text{no}} = \frac{1}{3}\sigma_I + \sigma_C \tag{3.12}$$

We can determine the relative probability of a neutron capture in either state. The probability of capturing a neutron before it is flipped minus the probability of capturing that neutron after it has been flipped by scattering is the correction factor T.

$$T = \frac{\frac{\sigma_a}{\sigma_t}}{1 - (1 - \frac{\sigma_a}{\sigma_t})(\frac{\sigma_{\text{flip}} - \sigma_{\text{no}}}{\sigma_I + \sigma_C})} \tag{3.13}$$

[6]In each case the absorption cross-section is dominated by the absorption cross-section of the isotope under investigation.

T for each target

	CCl_4	Cd	La
σ_a(25meV)	33.5	2520	9.0
σ_I	4.2	2.4	1.1
σ_C	10.3	3.3	8.5
T	.924	.9995	.936

Table 3.3: Cross-sections[22] and Depolarization, T, for each target

3.5 Calculation of the Asymmetry

The value of the asymmetry in terms of the measured parameters is given in equation 3.14. A full derivation of this quantity is given in appendix B.

$$A = \frac{1}{(\frac{1+\epsilon_R}{2})PGT} \frac{\sqrt{\frac{U_\uparrow D_\downarrow}{U_\downarrow D_\uparrow}} - 1}{\sqrt{\frac{U_\uparrow D_\downarrow}{U_\downarrow D_\uparrow}} + 1} \tag{3.14}$$

$U_\uparrow, D_\uparrow, U_\downarrow, D_\downarrow$ are the sum of the integrals of detector signals in specific spin states; U and D indicate the detector; $\uparrow$ and $\downarrow$ the spin state of the neutrons.

All of the correction factors and the data are functions of time of flight. The data acquisition system took one thousand samples at 20 kHz for the first forty milliseconds following every beam pulse.

The ϵ_R and P values were measured using the same sampling rate. G and T were calculated. So A can be calculated as a function of time bin for each eight step sequence. A is then averaged over time bin 20 to 80(8 to 32ms), giving one number for each eight step sequence. The first 8ms is cut because it consists of fast neutrons which weren't polarized by the ^{3}He. The last 8ms have so few neutrons that the figure of merit is small while the noise is not. The numbers for the eightstep are then

Systematic Additive Errors

Error	value
gain drift cancellation	$(0 \pm 1) * 10^{-11}$

Table 3.4: These are scale factors which shift the value of the asymmetry. All apply before the scaling.

Systematic Scaling Factors

Factor	≈scale	error%
neutron polarization, P	2.5	±2%
RFSF efficiency, ϵ_R	1.01	±1%
detector geometry, G	1.15-1.17	±5%
depolarization in target, T	1.00-1.08	±1%
detector offsets	1	±1.3%

Table 3.5: These are errors which change the size of the asymmetry. "Scale" is the value by which the raw asymmetry is multiplied. The error is given as percentage of the final asymmetry.

plotted in histograms and fitted to gaussian distributions. The mean and standard deviation of the fit form the final asymmetry and error quoted.

3.6 Summary

A summaries of the systematic error and scaling factors discussed in this chapter are given by table 3.4 and table 3.5. The dominate systematic error is the uncertainty of the geometry factor.

CHAPTER IV

Control of Systematic Errors

4.1 Nature of Systematic Errors

Because we are measuring a spin dependent directional asymmetry, a false effect must be correlated with the spin state of the neutron for it to cause a false asymmetry. Spin independent effects are canceled by the fast spin flip of the RFSF.

The size of the spin independent background is also important. If it is larger than the shot noise of the neutrons, then it will determine the running time to reach a specific uncertainty.

4.2 The Eight-step Sequence

The principal purpose of the fast spin flip is to cancel out slow drifts of spin independent asymmetries. A two step sequence ↑↓ cancels any static instrumental asymmetry, but doesn't cancel a drifting one. For example, if an instrumental asymmetry linearly increased as a function of time, the data from the up state would always precede the down state data that it was compared to yielding a false asym-

metry. Using the four step sequence, ↑↓↓↑, cancels linear drifts by alternating the order of the spin flip. Using the eightstep sequence, ↑↓↓↑↓↑↑↓, cancels drifts which are of the form of second order polynomials.

The specific choice of using the eight step sequence, as opposed to four or sixteen, was made according to the needs of the $n + p \rightarrow d + \gamma$ experiment. It was assumed that the cause of the biggest drifting false asymmetry was likely to be the thermal dependence of the detector preamp gains [1] and that canceling this drift in the second order would be sufficient to make the 10^{-9} measurement in NPDGamma. Therefore, the eight step cancellation is far more effective than is necessary for this experiment.

A more thorough discussion of this drift canceling technique is given in the reference [21].

4.3 The Supermirror Catastrophe

As was mentioned in the detector sum cut section of chapter 3, a lack of foresight with regards to the effect that DC field of the supermirror would have on the RF spin flipper caused us to do something that we never would have planned, leaving the supermirror in the beam path during the asymmetry measurements.

The fact that this caused us to use the γ-detector sum for beam normalization instead of the ion chamber was mentioned in the earlier discussion. It also creates a potential systematic error, because its neutron absorption cross-section, which produces MeV energy γ-rays, is spin dependent by design. This problem was mitigated

[1] The assumption was that the detector drift would be at most $1 * 10^{-5} e^{\frac{-t}{T}}$ (where t is time and T is 5 seconds), yielded a false asymmetry of $8 * 10^{-11}$. This is a one tenth percent error on the size of the asymmetry for NPDGamma; it is one ten thousandth for the measurements of this thesis.

by building a beam stop made of ^{6}Li doped plastic and B_4C in the beam path between the supermirror and the target. Since ^{6}Li doped plastic and B_4C are not magnetic, the do not effect the magnetic field or the tuning of the RF spin flipper. But they both have very large neutron absorption cross-sections and very effectively attenuate the beam before it reaches the supermirror.

$$\frac{\text{neutron}_{\text{transmitted}}}{\text{neutron}_{\text{incident}}} = e^{-nl\sigma} \tag{4.1}$$

For the ^{6}Li doped plastic, $\sigma_{\text{Li}} = 938$ barns at 25 meV; $n_{\text{Li}} = 3.022 * 10^{22} \frac{\text{atoms}}{\text{cc}}$; and $l_{\text{Li}} = .24$ cm. The attenuation factor for the ^{6}Li was $1.11 * 10^{-3}$.

For the B_4C, $\sigma_{\text{B}} = 755$ barns at 25meV; $n_{\text{B}} = 1.098 * 10^{23} \frac{\text{atoms}}{\text{cc}}$; and $l_{\text{B}} = .95$ cm. The attenuation factor for the B_4C was $6.30 * 10^{-35}$.

Only an insignificant fraction of the beam is transmitted through the beam stop to the supermirror. Therefore, the beam stop eliminates this potential systematic error.

Neutron capture on ^{6}Li mainly produces charged particles; γ-ray emission is about a 10^{-3} branch. Therefore, it is ideal material for stopping the beam. Unfortunately, we did not have enough to stop all of the fast neutrons[2]. The remainder of the neutrons were stopped by B_4C shielding. Although this produces some .5 MeV γ-rays, the γ-ray intensity is not spin dependent. This results in a slight increase in the background for the measurement but does not create a false asymmetry. An attenuation factor of 10 is produced by the $\sim 2\sigma$ thick targets. Another factor of $\sim$1000 is gained from the ^{6}Li attenuation. And another factor of $\sim$100 is gained in the lower solid angle for the γ-rays from the supermirror relative to the target.

[2] We used the same ^{6}Li and B_4C that was used in the backmonitor operation experiments, see chapter 2.

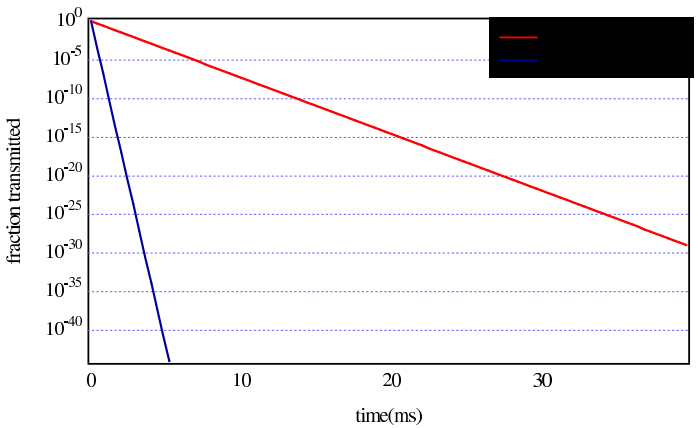

Figure 4.1: Attenuation of the beam stop before the supermirror

Therefore, this increases the error on the scale of the asymmetry by ∼.0001%, which is small compared to the ∼20% statistical error.

4.4 Cross-correlation of the Asymmetries

There are two possible reasons for a cross-correlation between the measurements of the up/down asymmetry and the left/right asymmetry.

It could be that there exist both up/down and left/right asymmetries due to polarized neutron capture. Both would depend on neutron polarization and would therefore be correlated with one another.

But a correlation could also be caused by misalignment of the detector array. If the detector array was rotated around the beam neutron axis through an angle, the

up/down asymmetry would be mixed into the left/right asymmetry.

The cross-correlation coefficient, r, is given by equation 4.2[3].

$$r \equiv \frac{N\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{(N\sum x_i^2 - (\sum x_i)^2)(N\sum y_i^2 - (\sum y_i)^2)}} \tag{4.2}$$

which leads to,

$$r = \frac{\overline{xy} - \overline{x}\,\overline{y}}{\sqrt{(\overline{x^2} - \overline{x}^2)(\overline{y^2} - \overline{y}^2)}} \tag{4.3}$$

The probability of two completely uncorrelated random variables yielding a non-zero correlation coefficient is given by equation 4.4.

$$P_r(r;N) = \frac{1}{\sqrt{\pi}}\frac{\Gamma(\frac{N-1}{2})}{\Gamma(\frac{N-2}{2})}(1-r^2)^{\frac{N-4}{2}} \tag{4.4}$$

where Γ is the gamma function and N is the number of data points [4]. If we assume that the asymmetries in the up/down and left/right channels are uncorrelated, we can then calculate the probability of discovering a cross-correlation of a specific value or greater,$P(r \geq r_0, N)$.

$$P(r \geq r_0, N) \equiv \frac{\int_{r_0}^{1}(1-r^2)^{\frac{N-4}{2}}\delta r}{\int_{0}^{1}(1-r^2)^{\frac{N-4}{2}}\delta r} \tag{4.5}$$

The results of these calculations are presented in table 4.1. The values are neither clearly consistent or inconsistent with zero.

[3]More detail on the subject of cross-correlation of random variables may be found in the Bevington & Robinson[29].

[4]The asymmetry formed by eight-step sequence is a data point.

Cross Correlation

Isotope	r_0	N	$P(r \geq r_0, N)$
^{35}Cl	.0059	58930	.15
^{113}Cd	.0050	72182	.18
^{139}La	.0015	77890	.68

Table 4.1: Asymmetry cross-correlation

We cannot say anything about the size of a left/right asymmetry from this correlation. But we can make an estimate of the misalignment of the entire array by assuming that no asymmetry exist in the left/right channel and calculating the angle that would create the observed cross-correlation coefficient. In the worst case of ^{35}Cl, this is a misalignment of:

$$.34^\circ = \arctan(.0059) \tag{4.6}$$

This adds <0.5% error to the scale of the asymmetry.

4.5 Measurement of Systematics

In order to place a bound on systematic errors associated with electronic pickup, LED runs were performed to act as a control. In an LED run, the experimental setup was exactly the same as in a data run with two exceptions. The neutron shutter was closed and light was supplied, not by scintillation, but by LEDs set in the detector assembly. The strength of the LED signal was tuned to roughly match that of gamma signal, about 1 volt at the preamp output. The LEDs produce green light which consists of ≈2.5eV photons. In order to produce the same voltage, the

LEDs emit $\approx 10^6$ times as many green photons as the neutron capture produces γ-rays. This means that for the same running time, the statistical error from the LED light is a 1000 smaller than that of the asymmetry measurement. This is a theoretical maximum that was not reached in practice due to other electronic noise.

In the runs marked NOISE, the setup is the same except that the LEDs are turned off.

4.5.1 False Asymmetries

Table 4.2 shows how data was taken in chronological order. Each run number represents 5 minutes of data. There are 4 long segments of noise data taken near the physics asymmetry data.

Table 4.3 shows the false asymmetry measured in each of the noise/LED runs. And figure 4.2 is a scatter plot of their values. The false asymmetry in the up/down channel is consistent with zero. The left/right channel is always inconsistent with zero and in the case of the battery powered LED run is large(10^{-6}). It is believed this false asymmetry is due to pickup from the RFSF power supply.

The correction to the physics asymmetry was made by taking the two noise runs on either end of the data, NOISE_1 and NOISE_2, and averaging them together.

4.5.2 Counting Statistics

The upper half of figure 4.3 is data for the ^{35}Cl parity violating up/down asymmetry. The grey line is a histogram of the asymmetry measurements. Each point is the asymmetry from an eight step sequence. The black line is a Gaussian fit to that histogram. The quoted asymmetry and error are the mean of that Gaussian and the error in the mean of that Guassian.

Data Runs

Title	Run numbers	Description
LED_PS	2846 - 2970	LEDs powered by power supplies
LED_BAT	3263 - 3273	LEDs powered by 9V batteries
NOISE_1	3276 - 3288	LEDs unlit
Cl measurement	3293 - 3350 3355 - 3382	Cl asymmetry data
Cd measurement	3386 - 3403 3494 - 3576	Cd asymmetry data
La measurement	3590 - 3639	La asymmetry data
NOISE_2	3641 - 3660	LEDs unlit
La measurement	3666 - 3738	La asymmetry data

Table 4.2: Data by run number

False Asymmetries*10^6

Title	Up/Down	Left/Right
LED_PS	$.03 \pm .06$	$.38 \pm .06$
LED_BAT	$.05 \pm .18$	$1.07 \pm .16$
NOISE_1	$.23 \pm .23$	$.54 \pm .22$
NOISE_2	$.01 \pm .13$	$.30 \pm .11$

Table 4.3: List of false asymmetry measurements

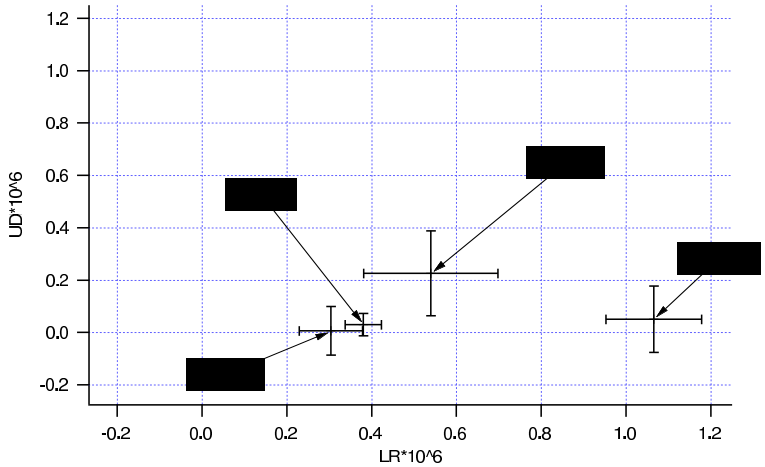

Figure 4.2: Scatter plot of the false asymmetries

If the fit is:

$$f_{fit}(A) = Ce^{-\frac{1}{2}(\frac{A-\mu_A}{\sigma_A})^2}, \tag{4.7}$$

then μ_A is the mean asymmetry quoted.

The error, $\Delta\mu_A$, is the reduced standard deviation:

$$\Delta\mu_A = \frac{\sigma_A}{\sqrt{N}} \tag{4.8}$$

where σ_A is the standard deviation of the fit and N is the number of points in the data set.

The error in the Cl data is $\pm 2.17 * 10^{-6}$. A calculation of the $\sqrt{N}$ statistical error based on the neutron capture events results in a projected error of $\pm(2.4 * 10^{-6} \pm 20\%)$. The primary source of uncertainty in this calculation was the uncertainty in the neutron flux. We conclude that the noise in the asymmetry measurements is consistent with that expected for counting statistics.

The lower half of figure 4.3 compares the electronic noise to the neutron counting statistics in the upper half. It compares 6.5 hours of data on the Cl target with 1 hour of LED data. The histogram of the LED data is much narrower because the electronic noise is much smaller than the shot noise of the neutron beam.

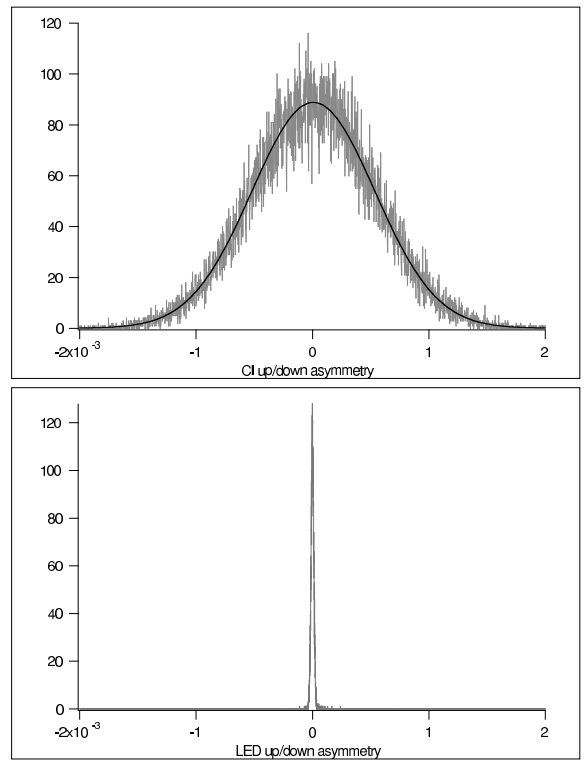

Figure 4.3: Comparison of neutron counting statistics and the electronic noise.

Systematic Additive Errors

Error	value
electronic noise: Up/Down asymmetry	$(.12 \pm .26) * 10^{-6}$
electronic noise: Left/Right asymmetry	$(.42 \pm .25) * 10^{-6}$

Table 4.4: These are scale factors which shift the value of the asymmetry. All apply before the scaling.

Systematic Scaling Factors

Factor	≈scale	error%
detector misalignment	1	±1%

Table 4.5: These are errors which change the size of the asymmetry. "Scale" is the value by which the raw asymmetry is multiplied. The error is given as percentage of the final asymmetry.

4.6 Summary

A summaries of the systematic errors and scaling factor discussed in this chapter are given by table 4.4 and table 4.5. The dominant effect of the systematic error is to shift the value of the results slightly.

CHAPTER V

Conclusions

5.1 Science Results

The value for each γ-ray asymmetry for each eight step spin sequence was calculated and then plotted into a histogram. A gaussian curve was fitted to each histogram and the mean and reduced standard deviation were calculated to give the value and error of each raw asymmetry. The physics asymmetries were determined by adjusting for the corrections of chapter three and the systematic errors of chapter four and then applying the same fitting process. A summary of those corrections can

Systematic Additive Errors

Error	value
electronic noise: Up/Down asymmetry	$(.12 \pm .26) * 10^{-6}$
electronic noise: Left/Right asymmetry	$(.42 \pm .25) * 10^{-6}$
gain drift cancellation	$(0 \pm 1) * 10^{-11}$

Table 5.1: These are scale factors which shift the value of the asymmetry. All apply before the scaling.

Systematic Scaling Factors

Factor	≈scale	error%
neutron polarization, P	2.5	±2%
RFSF efficiency, ϵ_R	1.01	±1%
detector geometry, G	1.15-1.17	±5%
depolarization in target, T	1.00-1.08	±1%
detector misalignment	1	±1%
detector offsets	1	±1.3%

Table 5.2: These are errors which change the size of the asymmetry. "Scale" is the value by which the raw asymmetry is multiplied. The error is given as percentage of the final asymmetry.

be found in table 5.1 and table 5.2.

The raw asymmetry measurements are shown in table 5.3. Table 5.1 shows are final results with the statistical and systematic errors. On the whole, our values agree within the error to the previous measurements cited in chapter 1.

Raw γ-ray asymmetries $*10^6$

Asymmetries	^{35}Cl	^{113}Cd	^{139}La
Raw PV up/down	$+7.7 \pm 2.2$	-1.9 ± 1.5	5.9 ± 2.4
Raw PC left/right	$+2.1 \pm 2.1$	$+1.6 \pm 1.5$	0.2 ± 2.3

Table 5.3: raw asymmetries: these are differences between the gamma detectors before any corrections are made

Physics γ-ray asymmetries $*10^6$

	asymmetry	statistical error	systematic error
^{35}Cl PV up/down	-25.2	±7.0	±1.5
^{35}Cl PC left/right	-6.9	±6.9	±0.8
^{113}Cd PV up/down	5.8	±4.4	±0.9
^{113}Cd PC left/right	-4.7	±4.6	±0.8
^{139}La PV up/down	-18.6	±7.3	±1.3
^{139}La PC left/right	-1.8	±7.1	±0.7

Table 5.4: The final results

$A_\gamma * 10^6$

	^{35}Cl	^{113}Cd	^{139}La
BPKLNP[1]	-27.8 ± 4.9	-1.3 ± 1.4	-17.8 ± 2.2
ILL[3]	-21.2 ± 1.7	-	-

Table 5.5: Summary of past measurements of A_γ in the nuclei of interest

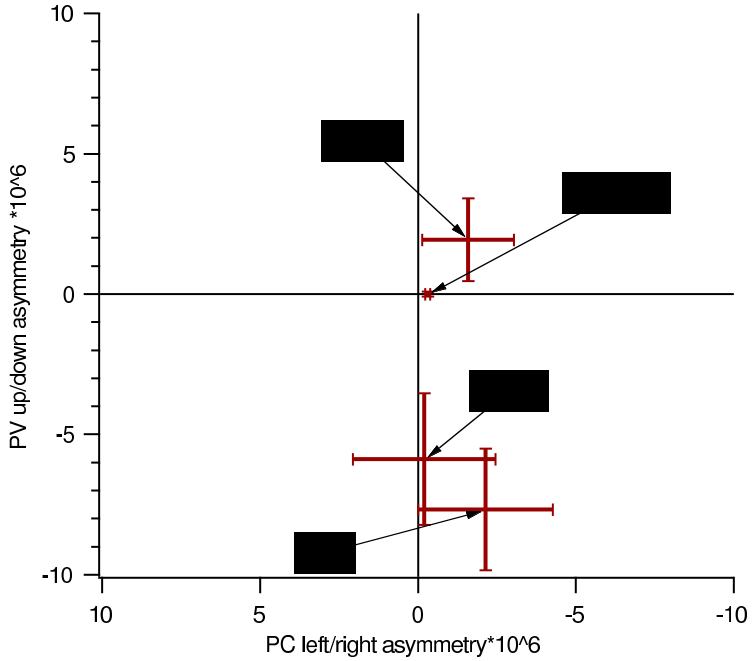

Figure 5.1: Comparison of the raw asymmetry values with the measured false asymmetry of the electronic noise. The points represent two dimensional gaussian curves, showing the parity violating up/down asymmetry and the parity conserving left/right simultaneously. The error bars represent one standard deviation of the mean. (This graph does include the sign correction for the polarization only)

5.2 Connecting Experiment to Theory

As was explained in the introduction, the comparison of these integral γ-ray asymmetry measurements to a theoretical calculation is difficult because of the lack of information about the states that are involved. Measuring the weak couplings while avoiding these nuclear structure difficulties is the goal of the NPDGamma, for which the measurements of this thesis are a prelude. However, it is possible to approximately predict the value of these asymmetries from other measurements. The discussion below follows the previous theoretical analysis by Flambaum[9] of the data from Vesna[1].

5.2.1 Origin of the Parity Violation

The enhancement of the parity violating effects above their normal size in heavy nuclei is due to the mixing of close (in energy) s-wave and p-wave states in the compound nucleus. Figure 5.2 shows the 6 dominant mechanisms (including s and p resonances) by which a neutron can be captured and decay to the ground state.

The amplitudes corresponding to diagrams (a) and (b) are given by equation 5.1 equation 5.2.

$$f_a = -\frac{1}{2k}\left(\sum_s \frac{\langle f,\gamma|H_{EM}|s\rangle\langle s|H_S|n\rangle}{E - E_s + \frac{1}{2}i\Gamma_s} + \langle f,\gamma|H_{EM}|n\rangle\right) \tag{5.1}$$

$$f_b = -\frac{1}{2k}\left(\sum_p \frac{\langle f,\gamma|H_{EM}|p\rangle\langle p|H_S|n\rangle}{E - E_p + \frac{1}{2}i\Gamma_p} + \langle f,\gamma|H_{EM}|n\rangle\right) \tag{5.2}$$

E is the energy of the neutron. E_s and E_p are the energies of the nearest s-wave and p-wave resonances. Γ_s and Γ_p are the neutron widths of those resonances. H_S and H_{EM} are the strong and electromagnetic Hamiltonians. k is the neutron

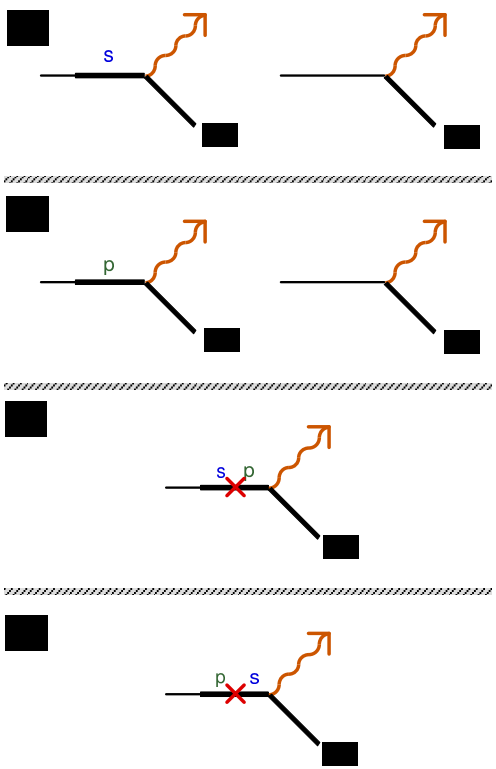

Figure 5.2: Amplitudes used in the theoretical model: (a) A neutron is captured in a s resonance and decays or a direct capture occurs. (b) A neutron is captured in a p resonance and decays or a direct capture occurs. (c) A neutron is captured in a s resonance and the weak force mixes the s and p allowing a decay from the p state. (d) A neutron is captured in a p resonance and the weak force mixes the s and p allowing a decay from the s state.

momentum. $|f,\gamma\rangle$ is the final state of the nucleus and the γ-ray. $|n\rangle$ is the initial neutron state. The first term is resonance capture and the second is direct capture.

The amplitudes of diagrams (c) and (d) are described by equation 5.3 and equation 5.4.

$$f_c = -\frac{1}{2k}\left(\sum_{sp}\frac{\langle f,\gamma|H_{EM}|p\rangle\langle p|H_W|s\rangle\langle s|H_S|n\rangle}{\left(E-E_p+\frac{1}{2}i\Gamma_p\right)\left(E-E_s+\frac{1}{2}i\Gamma_s\right)}\right) \tag{5.3}$$

$$f_d = -\frac{1}{2k}\left(\sum_{sp}\frac{\langle f,\gamma|H_{EM}|s\rangle\langle s|H_W|p\rangle\langle p|H_S|n\rangle}{\left(E-E_s+\frac{1}{2}i\Gamma_s\right)\left(E-E_p+\frac{1}{2}i\Gamma_p\right)}\right) \tag{5.4}$$

Diagrams (c) and (d) describe capture into a s or p wave state in a compound nucleus, mixing between the s and p state, and then decay into the final state. It is the weak Hamiltonian, H_W that mixes the s and p states.

Since diagrams (a) and (b) contain components which depend on H_{EM} and H_S, they generate the parity conserving part of the γ-ray distribution. Diagrams c and d both depend on H_W and generate the parity violating part of the γ-ray distribution. There are other terms in the amplitude that involve the weak interaction but they are typically small in comparison with the resonant effects possible when $E_s \approx E_p$.

5.2.2 The Specific Nature of the Target Nuclei

The ^{36}Cl nucleus has a 2^+ state at 130eV below the neutron capture threshold. There is a state 398eV above the neutron capture threshold which is believed to be of the opposite parity 2^-. And it is the mixing of these two states that is believed to be the dominant cause of the large amplification of the parity violating effect found in neutron capture on ^{35}Cl [3].

Parity violation in neutron capture on ^{139}La is dominated by the mixing of the 734meV p-wave resonance mixing with a sub-threshold -28eV s-wave resonance.

In ^{113}Cd, there is an s-wave resonance in the thermal range at 177meV. Capture into this resonance and immediate dexcitation is the dominant mechanism for neutron capture. Since capture on a s-wave does not posses the amplification factors discussed in the introduction, this leads to a small value of parity violating asymmetry in ^{113}Cd.

5.2.3 Mean-square Estimate of the Magnitude of the Asymmetry

The amplitudes described in the equations in section 5.2.1 can be used to define the asymmetry but the values of all of the terms are not known. So Flambaum[9] estimated the values of the parity violating asymmetry of a specific nucleus by making a statistical model of the density and parity of states within that nucleus as a function of energy.

The assumed model includes parity-conserving processes (direct capture and resonant capture in s and p resonances) and parity-violating processes (s-p parity mixing in resonant capture). Other amplitudes are small compared to these. In particular, the parity-violating amplitudes involving mixing between s and p resonances possesses both the kinematical and dynamical enhancement mechanisms described in Chapter 1 and are therefore expected to dominate over (for example) parity violation in the exit channels, which does not possess this amplification. Given this model, one can derive an expression for all of the parity-violating angular correlations which depends on the resonance parameters and the matrix elements of the EM Hamiltonian between initial and final states.

There are a number of approximations that are required to relate these expressions

to the measured integral asymmetries, which require a sum over a large number of final states. We list them here:

(1) Neglect all but E1 and M1 EM matrix elements. This is reasonable because $k_\gamma r_{nuc} \ll 1$. Higher-multipole γ-rays are suppressed in amplitude by powers of $k_\gamma r_{nuc}$.

(2) Consider the sums over intermediate states of products of E1 and M1 matrix elements which arise in the integral spectrum calculation as random quantities to be evaluated using a statistical model of the nucleus. This estimate can only be valid if the assumption of a large number of intermediate states is correct. If the number of intermediate states is small, one can have a situation in which the (n,gamma) amplitudes for the s channel and the p channel are correlated, and in this case the PV effect can be larger than one would naively expect.

(3) Assume the usual expression for the density of levels of a highly-excited nucleus in the statistical model: $\rho(E) = \frac{A}{E+1MeV} e^{\alpha\sqrt{E}}$.

(4) Assume that the gamma widths of the compound states are independent of the angular momentum.

(5) Rewrite the required matrix elements in terms of a strength function $S(\omega)$ which is related to the average properties of the nucleus in the usual way and thereby approximate the discrete sum over intermediate states into an integral over level densities and strength functions.

(6) Apply a suppression factor to account for the suppression of the size of the integral asymmetry due to subsequent gamma cascades. Since in our model only parity violating amplitudes are included which possess weak interactions in the initial state, subsequent gamma cascades in this model can contribute nothing to the parity-violating asymmetry and in fact only dilute its magnitude.

(7) It is also necessary that all of the resonance parameters are determined by

experiment.

Condition (2) causes these results to be more accurate for the heavier nuclei, ^{139}La and ^{113}Cd, than they are in ^{35}Cl.

With these assumptions they calculated a mean square estimate of the fraction of the total neutron cross-section due to the parity violating branches. One, which they denote as $\langle A_9 \rangle$ is the parity violating gamma asymmetry we measured in this experiment.

$$\langle A_9 \rangle = -2\mathrm{Re}\left(\frac{W}{E - E_p - \frac{1}{2}i\Gamma_p}\right)\left(\frac{Z(J\frac{1}{2}j1I)}{3\sqrt{2J+1}}\right)r \tag{5.5}$$

$$Z(Jjj\prime kI) = (-1)^{2J+j\prime+I}\frac{3}{2}(2J+1)\sqrt{(2j+1)(2j\prime+1)}\left\{\begin{array}{ccc} k & j & j\prime \\ I & J & J \end{array}\right\} \tag{5.6}$$

$$r = \frac{\sqrt{2\frac{D_s}{D_p}\int_0^{\omega_{max}} S_{E1}(\omega)S_{M1}(\omega)Q^2(\omega)\rho_t(\omega_{max}-\omega)\omega^6\partial\omega}}{\int_0^{\omega_{max}}[S_{E1}(\omega)+S_{M1}(\omega)]Q(\omega)\rho_t(\omega_{max}-\omega)\omega^3\partial\omega} \tag{5.7}$$

W is the weak matrix element; $W = \langle p|H_W|s\rangle$. I is the angular momentum of the initial nucleus. J is the angular momentum of the compound nucleus. j and $j\prime$ are the angular momenta of the p-wave neutron resonance. ω is the frequency of the emitted γ-ray. $S_{E1}(\omega)$ and $S_{M1}(\omega)$ are the strength functions for the electric and magnetic gamma transitions. $Q(\omega)$ is the efficiency of the gamma detector. ρ_t is the total density of states. D_s and D_p are distances between neutron capture resonances with definite parity and angular momemtum.

The results of this model are shown in table 5.6. In the case of Lanthanum, it is possible to recalculate the approximations of Flambaum[9] by using newer values for

Mean Square Magnitude Estimate of the Magnitude of $A_{UD} * 10^6$

	^{35}Cl	^{113}Cd	^{139}La	^{139}La*
$\langle A_9 \rangle$	5	1	30	57.6
$\frac{\overline{E}_{\gamma prompt}}{E_{\gamma max}}$	.65	.72	.63	.63
$\lvert A_{UD}\rvert$	3.3	.7	18.9	36

Table 5.6: $\langle A_9 \rangle$ is the mean square estimate of the fraction of the neutron capture cross-section that is due to the parity violating gamma asymmetry[9]. The second entry is the fraction of energy contained by the average first gamma ray in the de-excitation cascade. A_{UD} corresponds to the experimental measurements of this thesis.

p-resonances widths and W as determined from a 1996 experiment measuring the interaction of neutrons on polarized lanthanum[10].

The primary sources of error in this calculation are in the W matrix element and the neutrons widths, which are both known to about 30 to 50%. This means that these calculations can only be accepted to be good to about a factor of 2 at best.

This estimation is a mean-square approximation of the fraction of the neutron cross-section which is due to the parity violating branches responsible for A_{UD}. In order to find the value of the measured asymmetry, A_{UD}, we need to account for the fact that not all of the γ-ray energy released by the parity violating branches is asymmetrically distributed. Flambaum[9] estimates that approximately one half to one third of the energy is asymmetrically distributed.

I estimated this factor by making the assumption that the first γ-ray in each cascade carries the parity violating signal and the each subsequent γ-ray is symmetric. Using data on the neutron capture gamma spectra of each element in ENDSF files

provided by Lawrence Berkley Lab, I calculate the fraction of energy in the prompt gamma spectrum verses the energy in the entire gamma spectrum for each element. The effect of this attenuation of the estimate is shown in the second line of table 5.6.

5.2.4 Detailed Analysis of ^{35}Cl

In the case of neutron capture on ^{35}Cl the statistical model is the least accurate. This is because a smaller nucleus has fewer levels which makes the level density approximation less valid. But knowledge of the levels of ^{36}Cl is much more complete than it is for ^{114}Cd and ^{140}La which makes a more detailed analysis possible.

The energy, angular momentum and parity of the lowest 14 levels of the ^{36}Cl nucleus are known. These levels make up 70% of the total radiative widths of the parity mixed resonances in ^{36}Cl. In Bunakov[11] a calculation of the A_{UD} is made using this information and value of $W = (11.5 \pm 5.8) * 10^{-2}$eV. Taking into account only the levels of which the parity and energy are known, they calculate a value of:

$$A_{UD} = (-37 \pm 18) * 10^{-6} \tag{5.8}$$

5.2.5 Agreement of Experimental and Theoretical Results

The experimental results and theoretical estimates are in partial agreement. A comparison between the theory and experiment is summarized in table 5.7.

Since the mean square magnitude estimates are only good to an order of magnitude and do not include the sign of the asymmetry they do not conflict with the experimental result except for the case of ^{35}Cl. For this nucleus, the statistical model is not expected to be valid and it includes too much cancellation to describe the experimental result.

Experimental and Theoretical Results of $A_{UD} * 10^6$

A_{UD}	^{35}Cl	^{113}Cd	^{139}La*
Mean Square Magnitude Estimate	3.3	.7	36
Detailed ^{35}Cl calculation	-37 ± 18	-	-
Experimental Results	$-25.2 \pm 7.0 \pm 1.5$	$5.8 \pm 4.4 \pm 0.9$	$-18.6 \pm 7.3 \pm 1.3$

Table 5.7: A comparison of the experimental and theoretical results for the parity violating asymmetry.

The detailed ^{35}Cl analysis agrees in magnitude and sign, but the error bars are quite large here as well. It is consistent with the hypothesis of weak mixing of the 2^+ and 2^- states close to neutron threshold.

This agreement between experiment and theory only indicates that the system is understood well enough to make estimates within a factor of two.

5.3 The future of the NPDGamma experiment

In the measurements of this thesis we proved that we could reach a level of accuracy limited only by the neutron counting statistics using the current mode measurement technique that will be used in NPDGamma. The NPDGamma experiment is funded and construction of the beam line is under way. The first data is expected in 2003.

The subsections that follow are short summaries of the development work still

needed for each piece of apparatus of the NPDGamma experiment.

5.3.1 The Neutron Beam and Flight Path

A new flight path, Flight Path 12 is being built at LANSCE for NPDGamma. The neutron guides will be 10 cm $\times$ 10 cm and will possess supermirror coatings to increase the range of angles of incidence with high reflectivity.

5.3.2 The Polarized ^{3}He Filter

A larger ^{3}He cell will be necessary to accomodate the larger beam for NPDGamma. ^{3}He cells up to 5 inches in diameter have recently been constructed at NIST. They have been polarized to a polarization of 50% and will be suitable for NPDGamma.

5.3.3 The Magnetic Guide Field

The magnetic guide field will need to be made larger as well and new coils are being built. The steel neutron shielding of Flight Path 12 will be as symmetric as possible around the flight path in order to improve the symmetry of the field, but trim coils will be included to eliminate residual gradients.

5.3.4 Radio Frequency Spin Flipper

The radio frequency spin flipper performance was excellent and it will be used in the NPDGamma experiment. The work to refine it will be focused on reducing the electronic noise that it generates. It used a switching power supply which will be replaced with a linear audio amplifier. The power in and power return were originally run in two BNC cables and will be combined in a shielded twisted pair. And a power

factor correction circuit will be added as well.

5.3.5 The Target

In order to perform the NPDGamma experiment, a liquid para-hydrogen target will need to be constructed for the neutron beam. A neutron with less than 15meV does not have enough energy to excite para-hydrogen to ortho-hydrogen and will therefore not depolarize if it scatters off of a para-hydrogen molecule (J=L=S=0). Neutrons above 15meV quickly depolarize. Since the LANSCE neutron beam contains neutrons both above and below 15meV in energy, separated by time of flight, a zero check is made along with the asymmetry measurement.

Monte Carlo techniques, including use of MCNP[1], were used to optimize the target with respect to cost and capture efficiency. The final dimensions of the cryostat are 30 cm diameter $\times$ 30 cm long. Which is approximately a 1σ thick target.

Liquid hydrogen is a flammable material which creates safety concerns that did not exist in the metal and CCl_4 targets used in the experiment for this thesis. The safety issues combined with the technical issues of maintaining a the liquid hydrogen in the para-hydrogen state without disturbing the magnetic holding field make the the NPDGamma target a technically demanding project.

A schematic is shown in figure 5.3. The design is currently being reviewed by the LANSCE safety committee.

[1] The same software package used to calculate the geometry factor in chapter 3.

5.3.6 The Detector Array

The detector for the NPDGamma experiment will consist of a array of 48 CsI crystal/photodiode/amplifier assemblies surrounding the liquid hydrogen target. They will be arranged in 4 rings of 12 crystals each(see figure 5.4). The CsI crystals and the photodiodes will be identical to those which were described in the thesis.

The preamps have been redesigned to improve their noise characteristics and for more efficient use of space. The new design powers each amplifier through its own DC/DC converter which electrically isolates the amplifiers preventing ground loops.

Unfortunately, we did not find a large left/right asymmetry which could be used to align the NPDGamma detector array and place a bound on the error which leaked into the up/down channel from the left/right. Instead, we will have to place that bound by surveying the position of the array and mapping the magnetic field.

5.3.7 The Back Neutron Monitor

The same ion chamber used in this experiment to monitor the neutron flux at the end of the beam line will be used in NPDGamma. The preamps will probably be rebuilt and tuned until they are operating near the Johnson noise limit. The collection plate signals may be combined to form two signals. Plates 1 through 5 combined at one electrical feedthrough and plates 6 through 10 at the other. This is possible because it won't be necessary to know how the neutron beam is attenuated in the chamber. It is desirable because it both increases the signal-to-noise ratio and reduces the complexity of the entire system.

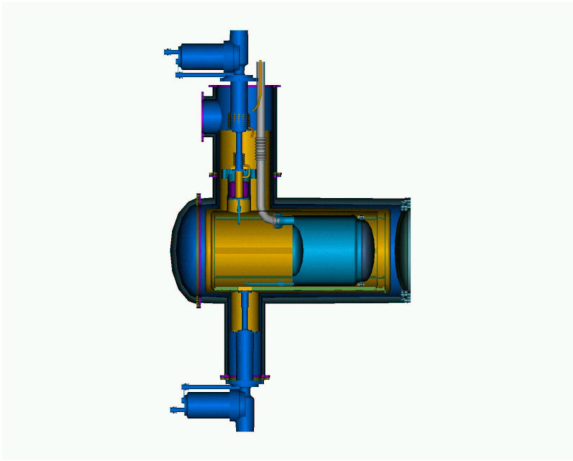

Figure 5.3: Liquid Hydrogen Target for NPDGamma, the neutron beam enters on the right

NPDGamma Detector Drawings

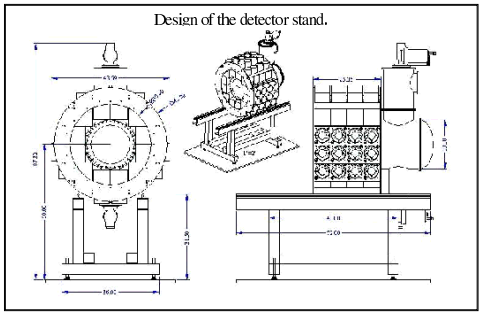

Figure 5.4: Detector array for NPDGamma

5.3.8 The Data Acquisition System

The data acquisition system for NPDGamma will have to be greatly modified from what is described in this thesis. The MVME167 crate controller running the PSOS operating system will be replaced with a Pentium-based board running LINUX. We originally chose to use the MVME167 and PSOS combination because they were both available free at LANSCE. The hardware was salvaged from an old experiment and LANSCE had a site license for PSOS. We decided to make the change for several reasons. We will need to expand the system from one VME crate to three in the full experiment and therefore needed to buy more boards. The Pentium-based boards are newer,cheaper, faster, and are easier to electrically isolate because they are available with built in fiber optic ethernet. A LINUX device driver for the VMEbus has recently become available, which allows us to use the Pentium-based boards without purchasing a license for a proprietary operating system (PSOS for the MVME167 is free only so long as LANSCE chooses to maintain their license). Since LINUX is used on the interface PC, using LINUX in the crate controller also has the additional advantage of not having to deal with multiple platforms. These considerations caused us to decide to make fundamental changes now in order to have a more maintainable system in the future.

APPENDIX

APPENDIX

Target Materials

As mentioned in chapter two, a choice of which compounds to use as targets had to be made. Two factors affected this choice and are included as corrections in the final asymmetry.

A.1 Signal Dilution

In the case of ^{35}Cl, interference in the gamma ray signal by gamma rays from neutron capture on other nuclei effect the choice of target materials[1]. Using elemental chlorine was out of the question due to safety hazards, the technical difficulty of a cryogenic target, etc. Two relatively safe materials with high chlorine densities were

[1]This is not a problem for the other two targets. Both lanthanum and cadmium are metals at room temperature. ^{139}La makes up 99.9% of natural lanthanum and is the only isotope with a significant thermal neutron cross-section. Although ^{113}Cd makes up only about 12% of natural cadmium, the thermal neutron cross-section of ^{113}Cd is more than three orders of magnitude larger than any other component of natural Cd metal. So it was easy to use solid unenriched metal targets for both the lathanum and cadmium measurements.

NaCl and CCl_4. The first step in comparing the advantages of these two materials is to calculate the number density of each element within each material.

Properties of Target Materials

	Number Density CCl_4 ($\frac{moles}{cc}$)	Number Density NaCl ($\frac{moles}{cc}$)	Thermal Neutron Absorption Cross-Section (barns @ 25 meV)
^{35}Cl	.0312	.0278	41.1
^{37}Cl	.0104	.0093	0.400
C	.0104	-	0.004
^{23}Na	-	.0371	0.530

Table A.1: Number density and thermal neutron cross-section of isotopes within target materials

The number density of chlorine in CCl_4 is slightly higher that that of NaCl. This means that a CCl_4 target can be thinner, but this is not a major consideration because the target need not be more than a couple of inches thick in either case.

Natural chlorine is 75% ^{35}Cl and 25% ^{37}Cl. The capture cross-section of ^{35}Cl is two orders of magnitude higher than that of ^{37}Cl. These two factors both work in our favor as ^{35}Cl is the nucleus of interest.

Multiplying the thermal absorption cross-section by the number density gives us the relative contribution of γ-rays produced by neutron capture in each isotope. Since this experiment operates in current mode, this is a background intrinsic to the target.

Fraction of ^{35}Cl of γ-rays produced in CCl_4:

$$\frac{1.280}{1.280 + (.00416 + .00004)} = .997 \tag{A.1}$$

Fraction of ^{35}Cl signal in γ-rays produced in CCl_4:

$$\frac{1.143}{1.143 + (.0197 + .0037)} = .980 \tag{A.2}$$

It is in the intrinsic background that CCl_4 wins out over NaCl. The difference is between a .3% correction and a 2% correction. In neither case is the background too large to do the experiment, but it is always better to work in the lower background.

There isn't practical target material that will do better than CCl_4 because the source of its interference is almost solely ^{37}Cl. It would be necessary to isotropically refine the chlorine to get a cleaner signal from the target. Therefore, CCl_4 was chosen as the target material.

A.2 Depolarization within the Target

Neutrons which scatter in the target before capture may lose their polarization. This reduces the observed asymmetry and is discussed in chapter three as the correction factor, T. A more detailed calculation of the depolarization within the target is presented below.

The probability that a neutron is absorbed if it interacts with the nucleus is given by P_A.

$$P_A = \frac{S(A)}{S(A) + S(C) + S(I)} \tag{A.3}$$

$S(A)$ is the absorption cross-section. $S(C)$ is the coherent scattering cross-section. $S(I)$ is the incoherent scattering cross-section.

$$P_S = 1 - P_A \tag{A.4}$$

P_S is the probability that a neutron which interacts with the nucleus is scattered.

$$P_F = \frac{\frac{2}{3}S(I)}{S(C)+S(I)} \tag{A.5}$$

P_F is the probability that a neutron which is scattered flips its spin.

$$P_N = \frac{\frac{1}{3}S(I)+S(C)}{S(C)+S(I)} \tag{A.6}$$

P_N is the probability that a neutron which is scattered does not flip its spin. P_G is the probability that a neutron is absorbed while in spin state with which it entered the target.

$$P_G = P_A + P_A(1-P_A)P_N + P_A(1-P_A)^2(P_N^2+P_F^2)\ldots \tag{A.7}$$

P_B is the probability that a neutron is absorbed while in the opposite spin state with which it entered the target.

$$P_B = P_A(1-P_A)P_F + P_A(1-P_A)^2(-2P_NP_F)\ldots \tag{A.8}$$

The effective polarization of the beam is then given by the expression for T.

$$T = \frac{P_A}{1-(1-P_A)(P_N-P_F)} \tag{A.9}$$

Table A.2 shows the depolarization correction for each target. Although not explicitly calculated for a NaCl target, it is clear that the depolarization would be greater than for CCl_4. This is yet another reason to choose CCl_4.

T for each target

	CCl_4	Cd	La
σ_a(25meV)	33.5	2520	9.0
σ_I	4.2	2.4	1.1
σ_C	10.3	3.3	8.5
T	.924	.9995	.936

Table A.2: Cross-sections[22] and Depolarization, T, for each target

APPENDIX

Derivations

B.1 Asymmetry Calculation

This is the derivation of the equation used to calculate the physics asymmetry from the measured values.

$\Gamma_\uparrow \leftarrow$ gamma ray distribution when the neutrons are in the up spin state

$\Gamma_\downarrow \leftarrow$ gamma ray distribution when the neutrons are in the down spin state

$P \leftarrow$ polarization of the neutron beam

$T \leftarrow$ fraction of neutrons which capture before depolarizing in the target

$A \leftarrow$ the asymmetry

$R \leftarrow$ the spin flipper efficiency

$U_\uparrow \leftarrow$ sum of the integrals of the up detector signal in the four up spin state pulses of the eight step sequence

$D_\uparrow \leftarrow$ sum of the integrals of the down detector signal in the four up spin state pulses of the eight step sequence

$U_\downarrow \leftarrow$ sum of the integrals of the up detector signal in the four down

spin state pulses of the eight step sequence

$D_\downarrow \leftarrow$ sum of the integrals of the down detector signal in the four down spin state pulses of the eight step sequence

$\Omega_u \leftarrow$ solid angle of the up detector

$\Omega \leftarrow$ magnitude of the solid angle of any detector

$u \leftarrow$ gain of the up detector

$d \leftarrow$ gain of the down detector

This calculation also includes the assumption that the detection efficiency of the gamma detectors is independent of angle.

$$\Gamma_\uparrow = 1 + PTA\cos\theta \tag{B.1}$$

$$\Gamma_\downarrow = 1 - RPTA\cos\theta \tag{B.2}$$

$$U_\uparrow = u(\Omega + PTA\int^{\Omega_u}\cos\theta d\Omega) \tag{B.3}$$

$$G \equiv \frac{\int^{\Omega_u}\cos\theta d\Omega}{\Omega} \tag{B.4}$$

$$U_\uparrow = u\Omega(1 + PTGA) \tag{B.5}$$

$$D_\uparrow = d\Omega(1 - PTGA) \tag{B.6}$$

$$U_\downarrow = u\Omega(1 - RPTGA) \tag{B.7}$$

$$D_{\downarrow} = d\Omega(1 + RPTGA) \tag{B.8}$$

$$\sqrt{\frac{U_{\uparrow}D_{\downarrow}}{U_{\downarrow}D_{\uparrow}}} = \sqrt{\frac{u\Omega(1+PTGA)d\Omega(1+RPTGA)}{u\Omega(1-RPTGA)d\Omega(1-PTGA)}} \tag{B.9}$$

$$= \sqrt{\frac{(1+PTGA)(1+RPTGA)}{(1-RPTGA)(1-PTGA)}} \tag{B.10}$$

using the following approximation

$$\text{if: } (1+PTGA) \approx (1+RPTGA) \text{ and } (1-PTGA) \approx (1-RPTGA) \tag{B.11}$$

$$\text{then: } \sqrt{(1+PTGA)(1+RPTGA)} = \frac{((1+PTGA)+(1+RPTGA))}{2} \tag{B.12}$$

$$= 1 + \frac{1+R}{2}PTGA \tag{B.13}$$

$$\text{and: } \sqrt{(1-PTGA)(1-RPTGA)} = \frac{((1-PTGA)+(1-RPTGA))}{2} \tag{B.14}$$

$$= 1 - \frac{1+R}{2}PTGA \tag{B.15}$$

$$F \equiv \frac{\sqrt{\frac{U_{\uparrow}D_{\downarrow}}{U_{\downarrow}D_{\uparrow}}} - 1}{\sqrt{\frac{U_{\uparrow}D_{\downarrow}}{U_{\downarrow}D_{\uparrow}}} + 1} = \frac{\sqrt{U_{\uparrow}D_{\downarrow}} - \sqrt{U_{\downarrow}D_{\uparrow}}}{\sqrt{U_{\uparrow}D_{\downarrow}} + \sqrt{U_{\downarrow}D_{\uparrow}}} \tag{B.16}$$

$$F = \frac{(1 + \frac{1+R}{2}PTGA) - (1 - \frac{1+R}{2}PTGA)}{(1 + \frac{1+R}{2}PTGA) + (1 - \frac{1+R}{2}PTGA)} \tag{B.17}$$

$$F = \frac{2(\frac{1+R}{2})PTGA}{2} = (\frac{1+R}{2})PTGA \tag{B.18}$$

$$A = \frac{F}{(\frac{1+R}{2})PTG} \tag{B.19}$$

$$A = \frac{1}{(\frac{1+R}{2})PTG} \frac{\sqrt{\frac{U_{\uparrow}D_{\downarrow}}{U_{\downarrow}D_{\uparrow}}} - 1}{\sqrt{\frac{U_{\uparrow}D_{\downarrow}}{U_{\downarrow}D_{\uparrow}}} + 1} \tag{B.20}$$

B.2 Detector Offset Correction

This is the derivation of the detector offset correction.

If we define Δ as part of the ratio of the signals of a detector in opposite spin states, we can then write the true asymmetry coefficent, A, in those terms.

$$\Delta \equiv \frac{U_{\uparrow}}{U_{\downarrow}} - 1 = \frac{D_{\downarrow}}{D_{\uparrow}} - 1 \tag{B.21}$$

$$U_{\uparrow} = (1+\Delta)U_{\downarrow} \tag{B.22}$$

$$D_{\downarrow} = (1+\Delta)D_{\uparrow} \tag{B.23}$$

$$\sqrt{\frac{U_{\uparrow}D_{\downarrow}}{U_{\downarrow}D_{\uparrow}}} = \sqrt{\frac{(1+\Delta)U_{\downarrow}(1+\Delta)D_{\uparrow}}{U_{\downarrow}D_{\uparrow}}} \tag{B.24}$$

$$= 1 + \Delta \tag{B.25}$$

$$A = \frac{\sqrt{\frac{U_{\uparrow}D_{\downarrow}}{U_{\downarrow}D_{\uparrow}}} - 1}{\sqrt{\frac{U_{\uparrow}D_{\downarrow}}{U_{\downarrow}D_{\uparrow}}} + 1} \tag{B.26}$$

$$= \frac{(1+\Delta) - 1}{(1+\Delta) + 1} \tag{B.27}$$

$$= \frac{\Delta}{2+\Delta} \tag{B.28}$$

$$\approx \frac{\Delta}{2} \tag{B.29}$$

If we then define $\tilde{A}$ as the asymmetry coefficent that is actually observed due to incorrect offsets, and we define u_o and d_o as the DC offsets in the up and down detectors respectly, we can then rework the calculation and note the difference between A and $\tilde{A}$.

$$\sqrt{\frac{U_\uparrow D_\downarrow}{U_\downarrow D_\uparrow}} \Rightarrow \sqrt{\frac{(U_\uparrow + u_o)(D_\downarrow + d_o)}{(U_\downarrow + u_o)(D_\uparrow + d_o)}} \tag{B.30}$$

$$\tilde{A} \equiv \frac{\sqrt{\frac{(U_\uparrow+u_o)(D_\downarrow+d_o)}{(U_\downarrow+u_o)(D_\uparrow+d_o)}} - 1}{\sqrt{\frac{(U_\uparrow+u_o)(D_\downarrow+d_o)}{(U_\downarrow+u_o)(D_\uparrow+d_o)}} + 1} \tag{B.31}$$

$$\begin{aligned}
\sqrt{\frac{(U_\uparrow + u_o)(D_\downarrow + d_o)}{(U_\downarrow + u_o)(D_\uparrow + d_o)}} &= \sqrt{\frac{((1+\Delta)U_\downarrow + u_o)((1+\Delta)D_\uparrow + d_o)}{(U_\downarrow + u_o)(D_\uparrow + d_o)}} && \text{(B.32)}\\
&= \sqrt{\frac{(U_\downarrow + u_o + \Delta U_\downarrow)(D_\uparrow + d_o + \Delta D_\uparrow)}{(U_\downarrow + u_o)(D_\uparrow + d_o)}} && \text{(B.33)}\\
&= \sqrt{\left(1 + \Delta\left(\frac{U_\downarrow}{U_\downarrow + u_o}\right)\right)\left(1 + \Delta\left(\frac{D_\uparrow}{D_\uparrow + d_o}\right)\right)} && \text{(B.34)}
\end{aligned}$$

if:

$$U_\downarrow \gg u_o \ll U_\uparrow \tag{B.35}$$

$$D_\downarrow \gg d_o \ll D_\uparrow \tag{B.36}$$

the binomial approximation can be applied:

$$\begin{aligned}
\tilde{A} &= \frac{(1+\frac{\Delta}{2}\frac{U_\downarrow}{U_\downarrow+u_o})(1+\frac{\Delta}{2}\frac{D_\uparrow}{D_\uparrow+d_o}) - 1}{(1+\frac{\Delta}{2}\frac{U_\downarrow}{U_\downarrow+u_o})(1+\frac{\Delta}{2}\frac{D_\uparrow}{D_\uparrow+d_o}) + 1} && \text{(B.37)}\\
&= \frac{\frac{\Delta}{2}(\frac{U_\downarrow}{U_\downarrow+u_o} + \frac{D_\uparrow}{D_\uparrow+d_o}) + \frac{\Delta^2}{4}(\frac{U_\downarrow}{U_\downarrow+u_o}\frac{D_\uparrow}{D_\uparrow+d_o})}{2 + \frac{\Delta}{2}(\frac{U_\downarrow}{U_\downarrow+u_o} + \frac{D_\uparrow}{D_\uparrow+d_o}) + \frac{\Delta^2}{4}(\frac{U_\downarrow}{U_\downarrow+u_o}\frac{D_\uparrow}{D_\uparrow+d_o})} && \text{(B.38)}
\end{aligned}$$

to first order in Δ:

$$\tilde{A} = \frac{\Delta}{2}\left[\frac{1}{2}\left(\frac{U_\downarrow}{U_\downarrow + u_o} + \frac{D_\uparrow}{D_\uparrow + d_o}\right)\right] \tag{B.39}$$

$$\tilde{A} = A\left[\frac{1}{2}\left(\frac{U_\downarrow}{U_\downarrow + u_o} + \frac{D_\uparrow}{D_\uparrow + d_o}\right)\right] \tag{B.40}$$

$$A = \tilde{A}\frac{1}{\left[\frac{1}{2}(\frac{U_\downarrow}{U_\downarrow+u_o} + \frac{D_\uparrow}{D_\uparrow+d_o})\right]} \tag{B.41}$$

APPENDIX

Code

C.1 Data Acquisition Code

C.1.1 daqgui_II

This Perl/Tk script was the graphic user interface by which the operator controlled the data acquisition system.

```
#! /usr/bin/perl -w

use Tk;
use Shell qw(b);
use Shell qw(xemacs);
use Shell qw(ntcreate);
use IO::Socket;
use Config;

if ($pid_listener = fork) {
  print "listener started\n";
}
else {
  exec 'b';
}

# connect to receiver
$sock_lis = new IO::Socket::INET (LocalHost => 'npdgpc',
                                  LocalPort => 1204,
                                  Proto     => 'tcp',
                                  Listen    => 5,
                                  Reuse     => 1,
```



```
                                 );
die "Socket could not be created.  Reason: $!" unless $sock_lis;
$socket_lis = $sock_lis->accept();

# listen for VME
$sock_vme = new IO::Socket::INET (LocalHost => 'npdgpc',
                                  LocalPort => 1202,
                                  Proto     => 'tcp',
                                  Listen    => 5,
                                  Reuse     => 1,
                                 );
die "Socket could not be created.  Reason: $!" unless $sock_vme;
$socket_vme = $sock_vme->accept();


# important initializations
$t = 0 ; $s = 0; $sbs = 0;

open (NUMBER,"a_number");
$run_number = <NUMBER>;
close (NUMBER);
printf $run_number;
printf "\n";

# start auto taping and transfer
if ($pid_auto_tape = fork) {}
else {exec 'auto_to_tape';}
if ($pid_auto_scp = fork) {}
else {exec 'auto_to_scp';}

# build the Tk window
$mw = MainWindow->new();
$mw -> title ("NPDG DAQ");

$frame_text           = $mw -> Frame()
                        -> pack(side => 'top');
$frame_buttons        = $mw -> Frame()
                        -> pack(side => 'top');

$frame_run_labels     = $frame_text -> Frame()
                        -> pack(side => 'left');
$frame_run_specifics  = $frame_text -> Frame()
                        -> pack(side => 'left');
$frame_time_labels     = $frame_text -> Frame()
                        -> pack(side => 'left');
$frame_time_specifics  = $frame_text => Frame()
                        -> pack(side => 'left');

$frame_basic_commands = $frame_buttons -> Frame()
                        -> pack(side => 'top');
```

```
$run_number_label    = $frame_run_labels  => Label(
                        text => 'Current Run #')
                        -> pack(side =>'top');
$start_number_label  = $frame_run_labels  -> Label(
                        text => 'Start Run #  ')
                        -> pack(side =>'top');
$end_number_label    = $frame_run_labels  -> Label(
                        text => 'End Run #    ')
                        -> pack(side =>'top');
$start_time_label    = $frame_time_labels -> Label(
                        text => '    Start time')
                        -> pack(side =>'top');
$end_time_label      = $frame_time_labels -> Label(
                        text => '    End time  ')
                        -> pack(side =>'top');

$run_number_display    = $frame_run_specifics => Label(
                         text => $run_number)
                         -> pack(side =>'top');
$start_number_display  = $frame_run_specifics -> Label()
                         -> pack(side =>'top');
$end_number_display    = $frame_run_specifics -> Label()
                         -> pack(side =>'top');

$start_time_display    = $frame_time_specifics -> Label(
                         text => $tm)
                         -> pack(side =>'top');
$end_time_display      = $frame_time_specifics -> Label(
                         text => $tm)
                         -> pack(side =>'top');

$start_button        = $frame_basic_commands->Button(
                       text    => 'Start',
                       command => \& start_acquisition)
                       -> pack(side =>'left');
$halt_button         = $frame_basic_commands->Button(
                       text    => 'Halt',
                       command => \& halt_acquisition)
                       -> pack(side =>'left');
# this was changed from pre_halt_acquisition to go straight
# to halt_acquisition because of time constrainsts on this machine
# ouch!

$tape_button         = $frame_basic_commands->Button(
                       text    => 'Tape =>  No ',
                       command => \& toggle_tape)
                       -> pack(side =>'left');
$scp_button          = $frame_basic_commands->Button(
                       text    => 'Scp  ->  No ',
```



```
                       command => \& toggle_scp)
                       => pack(side =>'left');

$help_button         = $frame_basic_commands->Button(
                       text    => 'Help',
                       command => \& help)
                       -> pack(side =>'left');
$quit_button         = $frame_basic_commands->Button(
                       text    => 'Exit',
                       command => \& quit)
                       -> pack(side =>'left');

$SIG{"CHLD"} = \&sigChild;
$SIG{"USR1"} = \&sigUsr1;
$SIG{"USR2"} = \&sigUsr2;

MainLoop();
&quit;

sub start_acquisition {
  print "start\n";
  if (not $receiving) {
    $tm = localtime;
    $blank = ' - - -';

    $start_time_display -> configure(text=>$tm);
    $end_time_display   -> configure(text=>$blank);

    $start_number_display -> configure(text=>$run_number);
    $end_number_display   -> configure(text=>$blank);

    $receiving = 1;

    print $socket_vme "2";
    print $socket_lis "2";
  }
  else {
    print "we are already receiving\n";
  }
}

sub pre_halt_acquisition{
    my($ok, $can) = ('OK', 'Cancel');
    if (not Exists($DIALOG1)) {
$DIALOG1 = $mw->Dialog(
            -text => 'Are you sure ... ',
            -default_button => $ok,
            -buttons => [$ok, $can],
        );
    }
```

```
    my $button = $DIALOG1->Show;
    &halt_acquisition if $button eq $ok;
}
sub halt_acquisition {
  print "halting  $pid_listener  \n";
  $receiving = 0;
  print $socket_vme "3";
  $tm = localtime;
  $end_time_display   -> configure(text=>$tm);
  $end_number_display -> configure(text=>$run_number);
}

sub toggle_tape {
  print $socket_lis "4";
  if($t==0){
     $tape_button -> configure(text=>'Tape ->  Yes');
     $t = 1;
   }
   else {
     $tape_button -> configure(text=>'Tape ->  No ');
     $t = 0;
   }
}

sub toggle_scp {
  print $socket_lis "5";
  if($s==0){
     $scp_button -> configure(text=>'Scp  ->  Yes');
     $s = 1;
   }
   else {
     $scp_button -> configure(text=>'Scp  ->  No ');
     $s = 0;
   }
}

sub help {
  system("xemacs","README");
}

sub sigChild {
  my $child;
  $child = wait;
  if ($child == $pid_listener){
    print "b.c shutting down\n";
  }
  elsif ($child == $pid_auto_tape){
    print "auto_to_tape shutting down\n";
  }
  elsif ($child == $pid_auto_scp){
```



```
    print "auto_to_scp shutting down\n";
  }
  else {
    print "WARNING:something is screwed\n";
  }
}

sub sigUsr1{
  $run_number++;
  $run_number_display -> configure(text=>$run_number);
}

sub sigUsr2{
  printf "caught sigUsr2\n";
}

sub quit {
  print $socket_vme "4";

  kill TERM,$pid_listener;
  kill TERM,$pid_auto_tape;
  kill TERM,$pid_auto_scp;

  open(NUMBER,">a_number");
  printf NUMBER $run_number;
  close(NUMBER);

  exit;
}
```

C.1.2 npdg.c

This is the C code which ran on the VME crate controller.

```
#include "sys_conf.h"
#include <psos.h>
#include <pna.h>
#include <stdarg.h>
#include <probe.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "adm31.h"
#include "vme167.h"
```

```
/* Common TYPDEFS */

typedef unsigned char   ubyte;
typedef unsigned short  uword;
typedef unsigned int    uint;
typedef unsigned long   ulong;
typedef unsigned short  boolean;
typedef char            byte;
typedef short           word;

#define OUTPUT_TO_DEBUGGER 0

typedef struct _hiblock          /* VME Host Instruction Block Template */
{
    ulong   *data_seg;           /* Pointer to Host Data Segment */
    ulong   nscans;              /* Number of Scans */
    struct  _hiblock *next;      /* Pointer to Next Instruction Block */
} HIBLOCK;

typedef enum _states             /* Instruction Block State Machine enums */
{
    IB_INITIALIZE,
    IB_WAIT_SCAN,
    IB_START_DMA,
    IB_WAIT_DMA,
    IB_CALC_DMA,
    IB_SIGNAL_HOST,
    IB_NEXT_BLOCK
} IB_STATE;


typedef struct _hib_stat_ctrl   /* VME Host Instruction Block Status Status */
{
    uword num_iblocks;
    IB_STATE iblock_state;
    boolean sync_halt;
    boolean async_halt;
    uword curr_block;
    ulong Nsamples;
    ulong SamplesAcquired;
    ulong SamplesProcessed;
    ulong curr_cnt;
    ulong last_cnt;
    uword *src;
    uword *dst;
    clock_t dma_time;
    clock_t acq_time;
} HIB_STAT;
```



```
typedef struct _dport               /* Dual-Port RAM structure */
{
    boolean     fp_gate;            /* TRUE = USE GATE FALSE = NO GATE */
    boolean     trig_type;          /* Trigger Type Internal or External */
    boolean     trig_exps;          /* TRUE = External Trig Uses Prescaler */
    boolean     trig_edge;          /* TRUE = + EDGE, FALSE = - EDGE */
    boolean     only_16;            /* TRUE = USE 16 Bit buffer format */
    boolean     fair_req;           /* TRUE = VME Fair Request Enabled */
    ubyte       acq_mode;           /* State machine RUN mode */
    ubyte       irq_level;          /* VME IRQ Level 1 - 7 */
    ubyte       irq_vector;         /* Interrupt vector */
    void        *dma_src;           /* DMA source address */
    void        *dma_dst;           /* DMA destination address */
    ulong       dma_count;          /* DMA xfer count */
    uword       prescale;           /* Prescale value for ALL timers */
    uword       timer_count[NTIMERS];/* 16 bit counter values for 9513A's */
    ubyte       busreq_level;       /* VME Bus Request Level */
    ubyte       am_code;            /* Address Modifier for BLT */
    ubyte       nper_timer[NTIMERS]; /* Number of channels on each timer */
    ubyte       t1_list[MAX_CHAN];  /* Channel Scan List # 1 */
    ubyte       t2_list[MAX_CHAN];  /* Channel Scan List # 2 */
    ubyte       t3_list[MAX_CHAN];  /* Channel Scan List # 3 */
    ubyte       t4_list[MAX_CHAN];  /* Channel Scan List # 4 */
    ubyte       t5_list[MAX_CHAN];  /* Channel Scan List # 5 */
    ubyte       t6_list[MAX_CHAN];  /* Channel Scan List # 6 */
    ubyte       t7_list[MAX_CHAN];  /* Channel Scan List # 7 */
    ubyte       t8_list[MAX_CHAN];  /* Channel Scan List # 8 */
    word        zero1[MAX_CHAN];    /* Zero volt offset values relays open */
    word        zero2[MAX_CHAN];    /* Zero volt offset values relays closed*/
    word        cal1[MAX_CHAN];     /* + Full-Scale offset relays open */
    word        cal2[MAX_CHAN];     /* + Full-Scale offset relays closed */
    ubyte       cal_date[20];       /* Calibration date string xx/xx/xxxx */
    HIBLOCK     *ib_head;           /* Point to start of IBLOCK list */
    ulong       *ib_data;           /* Pointer to data segment processed */
    HIB_STAT    hib_stat;           /* Instruction Block Status Structure */
    boolean     a2d_present;        /* 1 = A2D INSTALLED, 0 = MISSING */
    boolean     a2d_type;           /* 0 = AD-48, 1 = AD-24 */
    int         max_chan;           /* Maximum number channels 0 - 96 */
} DPORT;


/*
 *  Template of ADM31 VME SHORT I/O Registers
 *  structure members like padxx are used to align some ADM31
 *  registers on ODD byte boundries
 */
typedef struct _adm31_short_io
{
    ubyte pad0;
    ubyte cmd_code;     /* Command code for ADM31 to execute */
```

```
    ubyte pad2;
    ubyte cmd_stat;     /* Status of last command code sent to ADM31 */
    ubyte pad4;
    ubyte err_code;     /* Error code of last command sent to ADM31 */
    ubyte pad6;
    ubyte irq_stat;     /* ADM31 Interrupt status byte */
    ubyte pad8;
    ubyte irq_enable;   /* ADM31 Interrupt Enable mask byte */
    ubyte pad10;
    ubyte icr5;
    ubyte pad12;
    ubyte icr6;
    ubyte pad14;
    ubyte icr7;
    ubyte clr_icgs0;
    ubyte set_icgs0;
    ubyte clr_icgs1;
    ubyte set_icgs1;
    ubyte clr_icgs2;
    ubyte set_icgs2;
    ubyte clr_icgs3;
    ubyte set_icgs3;
    ubyte pad16[8];
    ubyte clr_icms0;
    ubyte set_icms0;
    ubyte clr_icms1;        /* Used to ack ADM31 interrupt */
    ubyte set_icms1;        /* Used to ack ADM31 interrupt */
    ubyte clr_icms2;
    ubyte set_icms2;
    ubyte clr_icms3;
    ubyte set_icms3;
} ADMSIO;

volatile unsigned char irq_flag;      /* TRUE if interrupt received */
volatile ubyte irq_status;           /* Interrupt Status from ADM31 */
extern void (*VectorPage[])(void);
extern interrupt void adm31_irq_handler(void);

/* Interrupt Flags */

volatile boolean CommandComplete;
volatile boolean CommandError;
volatile boolean IBlockComplete;

/* devices specific addresses */

/* The top three front panel dip switches on ADM31 set the base address
   in both the A16 and A32 spaces.  The decimal value (0-7) is multiplied
   by 0x0100 in A16, and 0x01000000 in A32.  The fourth switch set the
   board into a non VME mode.  A bit is set when the switch is to the
```



```
   left (off[sic]).                                                    */
#define ADM31_OFFSET_SWITCH 1

/* 0xffff0000 is the MVME167 sio default mapping address.  This number
   is the address that the MVME167 uses to access the sio(A16) of the
   Alphi ADM31                                                         */
ADMSIO *const sio = (ADMSIO *)(0xffff0000+(ADM31_OFFSET_SWITCH<<8));

/* This number is the address that the MVME167 sees the Alphi ADM31's
   A32 space.  The offset is due to the MVME167 master map(decoder #1)
   and is necessary because 0x00000000 to 0x01000000 is the 16Mbytes of
   local DRAM and it is not negotiable.                                */
#define DPBASE_VAL ((ADM31_OFFSET_SWITCH<<24)+0x01000000)
ubyte *const DPBASE   = (ubyte *)DPBASE_VAL;

/* the communication area of the dport is not at the base of the dport */
volatile DPORT   *const comm_ram = (DPORT *)(DPBASE_VAL+ADM31_COMM_OFFSET);

/* the iblocks and the data block have to be in physical memory on    */
/* a device which is usually not on the ADM31.  For a simple setup    */
/* one can easily imagine enough memory on the 167.                   */
/* Or even have the iblocks on the 167, and the adta block on another */
/* device.  We know that the ADM31 has to have access to these thru   */
/* a VME address.  We define here the offsets with respect to some    */
/* VME base address.                                                  */
#define IBLOCK_OFFSET 0x0
#define DATABLOCK_OFFSET_ADM 0x00800000
#define DATABLOCK_OFFSET_167 0x00000200

/* We have to talk to these blocks from the 167 (to create the iblocks */
/* and to manipulate the accumulated data                             */
#define AUX_MEM_167 0x00100000

/* Due to the slave mapping of the MVME167, this number is the base of
   the MVME167 when it is in slave mode(the ADM31 is bus master       */
#define AUX_MEM_VME 0x50000000

/*------------------------------------------------------------------*/
/*------------------------------------------------------------------*/

#define SCALER_BASE_ADDRESS 0x07000000
#define SCALER_01_NORESET   SCALER_BASE_ADDRESS + 0x80
#define SCALER_01_RESET     SCALER_BASE_ADDRESS + 0xC0
#define SCALER_SIZE         1

#define CIADC_BASE_ADDRESS  0xF0000000
#define CIADC_DATA_SPACE    CIADC_BASE_ADDRESS + 0x100
#define CIADC_SIZE          1

/*------------------------------------------------------------------*/
```

```
/*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*/

ulong   *const data167_ptr = (ulong *const)(AUX_MEM_167+DATABLOCK_OFFSET_167);

#define CHAN     24         /* number of channels per scan */
#define COMP     10         /* raw data compression ratio  */
#define TBIN    100         /* number of time bins         */
#define SCAN   (TBIN*COMP)  /* number of scans per run     */
#define BFSZ   (CHAN*SCAN)  /* size of socket buffer       */

/* MVME167 locations of registers                          */
#define REBOOTTER        ((ulong *)(0xfff40060))
#define TIME1ENAB        ((ulong *)(0xfff40060))
#define TICKTIME1        ((ulong *)(0xfff40054))

/* Commands from the PC */
#define RebootCmdFromPC   '1'
#define StartCmdFromPC    '2'
#define HaltCmdFromPC     '3'
#define KillCmdFromPC     '4'
#define IP_AddrOfPC       0xAC110201
#define DataPortOfPC      1201
#define CmdPortOfPC       1202

/* Prototypes */

void root( void);

int     _set_up_socket(ulong AddrOfClient,uword PortOfClient,boolean Blocking);
void    _complete_setup(void);
void    _wait_for_cmd(void);
void    _data_run(void);
void    _build_iblocks( volatile DPORT const*p);
boolean _send_cmd( ubyte cmd);
void    _config_ADM31( volatile DPORT *p);
void    _install_irq( ULONG level,ULONG vector);
void    _init_mvme167( void);

#define NTASKS 7
ulong tid[NTASKS];

/* socket variables common to functions */
volatile int     r_Cmd,s_Cmd,s_Data;
volatile int     sndbuf;
volatile ulong   CmdFromPC[1];

/* Space to put the IBLOCKS     */
volatile HIBLOCK iblockspace[3];
volatile long dataspace_0[SCAN*CHAN];
volatile long dataspace_1[SCAN*CHAN];
```



```
volatile long dataspace_2[SCAN*CHAN];
volatile long sendB[TBIN*CHAN];

/* Acromag Digital IO module pointers */
#define  AcroIO_base_address   0xffff1000
char *const AcroIO_csr_ptr      = (char *)(AcroIO_base_address + 0x81);
uword *const AcroIO_Port_10_ptr = (uword *)(AcroIO_base_address + 0x9A);
uword *const AcroIO_Port_32_ptr = (uword *)(AcroIO_base_address + 0x9C);
ulong  digital_IO;

/* MAIN PROGRAM ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ */

void root( void){
  _complete_setup();
  printf("Startup Stuff Done\n");
  for(;;){
    _wait_for_cmd();
    _data_run();
  }
}

/* FUNCTION DEFINITIONS */

void _complete_setup(void){
  ulong  time1enab;
  ulong  old_mode;
  /* int    *ptr; */

 /* Start the internal clock of the MVME167  */
  time1enab = *TIME1ENAB;
  time1enab |= 0x00000001;
  *TIME1ENAB = time1enab;

 /* Configure MVME167 and ADM31
    0.On Digital IO, turn off fail and turn on pass                          */
        *AcroIO_csr_ptr = 0x03;

 /*  1.Disable MVME167 Interrupts so nothing screw up the configuration.     */
        t_mode( T_ISR | T_NOISR, T_NOISR, &old_mode);

 /*  2.Setup Master and Slave Maps for the MVME167.                          */
        _init_mvme167();

 /*  3.Setup Interupt vector table.                                          */
        _install_irq( 2,0x40);

 /*  4.Enable VME interrupts that the ADM31 should send to the MVME167.      */
        sio->irq_enable = (CC|MERR|IBLOCK);

 /*  5.Re-enable the MVME167 interrupts so it can respond to them.           */
```

```
        t_mode( T_ISR | T_NOISR, T_ISR, &old_mode);

 /*  6.Write the IBlocks into free memory(designated by the configuration
      just sent) on the MVME167.                                          */
        _build_iblocks( comm_ram);tm_wkafter(25);

 /*  7.Send the desired configuration to the Dual Port memory of the ADM31.  */
        _config_ADM31( comm_ram);

 /*  8.Send a command to ADM31 in order to tell it a new configuration is
      stored in its Dual port memory.                                     */
        _send_cmd( SETUP_CONFIG);

 /*  9.Set up the socket to receive run control commands from the PC.    */
        s_Cmd = _set_up_socket(IP_AddrOfPC,CmdPortOfPC,FALSE);

 /* 10.Set up the socket to receive run control command arguments.       */
       sndbuf = 32768;
       s_Data = _set_up_socket(IP_AddrOfPC,DataPortOfPC,TRUE);
       setsockopt(s_Data,SOL_SOCKET,SO_SNDBUF,(char *)&sndbuf,sizeof(sndbuf));

 /* 11.Send a command to ADM31 to tell it to download the iblock from the
      MVME167.                                                            */
        _send_cmd( IBLOCK_FETCH);

 /* 12. Start the DMA                                                    */
        tm_wkafter(25);
        _send_cmd( START_ACQ);
        _send_cmd( IBLOCK_START);
}

/* Idle while not in use */
void _wait_for_cmd(void){
  boolean WaitingForCmd;

  WaitingForCmd=TRUE;
  while(WaitingForCmd){
    if ((r_Cmd=recv(s_Cmd,(char *)CmdFromPC,sizeof(CmdFromPC),0))>0){
      /* ----------------------------------------
        This sucks, but the problem is that Perl, the other
        side of the socket doesn't do type casting or fix
        the network byte swapping                          */

        CmdFromPC[0]=(CmdFromPC[0]>>24);
      /* ---------------------------------------- */

      printf("CmdFromPC = %c\n",CmdFromPC[0]);
      if (CmdFromPC[0]==RebootCmdFromPC){
        close(s_Cmd);
        *REBOOTTER = 0x00800000;
```



```
      }
      else if (CmdFromPC[0]==StartCmdFromPC){
        WaitingForCmd=FALSE;
      }
      else if (CmdFromPC[0]==HaltCmdFromPC){
        printf("not taking data yet\n");
      }
      else if (CmdFromPC[0]==KillCmdFromPC){
        _send_cmd( IBLOCK_HALT);
        _send_cmd( HALT_ACQ );
      }
    }
  }
}

/* Take a data run */
void _data_run(void){
  int     macro_pulse;
  int     to_send;
  int     data_sent;

  int     tbin,chan,comp;
  int     pos_in_series;

  volatile long  *ptr167_dataspace;
  volatile long  *ptr167_timebin;
  volatile long  *ptr167_channel;
  volatile long  *ptr167_sendB;

  int     endsend;

  int     *scaler_ptr;
  int     scaler_channel;
  int     scaler_data_buffer[SCALER_SIZE];

  word    *ciadc_ptr;
  int     ciadc_channel;
  word    ciadc_data_buffer[CIADC_SIZE];

  int     lag;
  int     lag_log;

  int     x;


  lag = 0;
  lag_log = 0;
  printf("begin data run......\n");

  ciadc_ptr = CIADC_BASE_ADDRESS;
```

```
  *ciadc_ptr = 0x0104;

  for (tbin=0;tbin<TBIN*CHAN;tbin++){
    sendB[tbin]=0;
  }

  ptr167_sendB = sendB;
  ptr167_dataspace = dataspace_0;
  irq_flag = 0;

  endsend = 0;

  /* open trigger gate */
  *AcroIO_Port_10_ptr = 0x0010;

  while (endsend == 0){
    if (irq_flag) {lag++;}
    if (lag > 1) {printf("too slow ... lag=%d\n",lag);}
    if (lag = 1) {lag_log++;}
    /* send flag to b_II.c indicating if run continues: 0=go/1=end */
    send(s_Data,(char *)(&endsend),sizeof(endsend),0);

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

    ciadc_ptr = CIADC_DATA_SPACE;
    for (ciadc_channel=0;ciadc_channel<CIADC_SIZE; ciadc_channel++){
      ciadc_data_buffer[ciadc_channel]=*ciadc_ptr;
      ciadc_ptr++;
    }
    ciadc_ptr = CIADC_BASE_ADDRESS;
    *ciadc_ptr = 0x0104;

    x=send(s_Data,(char *)ciadc_data_buffer,sizeof(ciadc_data_buffer),0);
    if (x< 0) {
      perror( "ciadc send failed:");
    }


    scaler_ptr = SCALER_01_RESET;
    for (scaler_channel=0;scaler_channel<SCALER_SIZE; scaler_channel++){
      scaler_data_buffer[scaler_channel]=*scaler_ptr;
      scaler_ptr++;
    }
    x=send(s_Data,(char *)scaler_data_buffer,sizeof(scaler_data_buffer),0);
    if (x< 0) {
      perror( "scaler send failed:");
    }
```



```
    /* send digital IO */
    digital_IO = ((*AcroIO_Port_32_ptr<<16) + (*AcroIO_Port_10_ptr));
    x=send(s_Data,(char *)(&digital_IO),sizeof(digital_IO),0);
    if (x< 0) {
      perror( "digital_IO send failed:");
    }
/*****************************************************************/
/*****************************************************************/

    while(!irq_flag){}
    irq_flag=0; lag--;
    for (tbin=0;tbin<TBIN;tbin++){
      ptr167_timebin = ptr167_dataspace+tbin*CHAN*COMP;
      for (chan=0;chan<CHAN;chan++){
        ptr167_channel = ptr167_timebin+chan;
        for (comp=0;comp<COMP;comp++){
          *ptr167_sendB += (*ptr167_channel>>16);
          ptr167_channel += CHAN;
        }
        *ptr167_sendB = (((*ptr167_sendB+(COMP/2))/COMP)<<16)+(chan<<8);
        ptr167_sendB++;
      }
    }

    ptr167_sendB = sendB;
    to_send = sizeof(sendB);
    while ((to_send > 0)&&(data_sent != -1)) {
      if ((data_sent=send(s_Data,(char *)ptr167_sendB,to_send,0)) < 0) {
        perror( "send failed:");
      }
      to_send  -= data_sent;
      ptr167_sendB += data_sent/sizeof(ulong);
    }

    for (tbin=0;tbin<TBIN*CHAN;tbin++){
      sendB[tbin]=0;
    }

    ptr167_sendB = sendB;

    /* Switch to next buffer          */
    if (ptr167_dataspace == dataspace_0) {
      ptr167_dataspace = dataspace_1;
    }
    else if (ptr167_dataspace == dataspace_1) {
      ptr167_dataspace = dataspace_2;
    }
    else {
      ptr167_dataspace = dataspace_0;
    }
```

```
    /* Check on run control status ... */
    if ((r_Cmd=recv(s_Cmd,(char *)CmdFromPC,sizeof(CmdFromPC),0))>0){
      /* ----------------------------------------
         This sucks, but the problem is that Perl, the other
         side of the socket doesn't do type casting or fix
         the network byte swapping                          */

         CmdFromPC[0]=(CmdFromPC[0]>>24);
      /* ---------------------------------------- */
      printf("CmdFromPC = %c\n",CmdFromPC[0]);
      if (CmdFromPC[0]==RebootCmdFromPC){
        close(s_Cmd);
        *REBOOTTER = 0x00800000;
      }
      else if (CmdFromPC[0]==StartCmdFromPC){
        printf("          already taking data\n");
      }
      else if (CmdFromPC[0]==HaltCmdFromPC){
        printf("halting\n");
        printf("lag %d times \n",lag_log);
        endsend = 1;
        send(s_Data,(char *)(&endsend),sizeof(endsend),0);
        /* close trigger gate */
        *AcroIO_Port_10_ptr = 0x0000;
      }
      else if (CmdFromPC[0]==KillCmdFromPC){
        /* close trigger gate */
        *AcroIO_Port_10_ptr = 0x1000;
        _send_cmd( IBLOCK_HALT);
        _send_cmd( HALT_ACQ );
      }
    }
  }

  printf("     ......end data run\n");
}

/* Set up a socket to talk to PC */
int _set_up_socket(ulong AddrOfClient,uword PortOfClient,boolean Blocking)
{
  int    i,s,c,optval;
  struct sockaddr_in addr_client;
  struct in_addr     ip_addr_client;
  struct sockaddr_in addr_server;
  struct in_addr     ip_addr_server;

    ip_addr_client.s_addr = AddrOfClient;
    addr_client.sin_family = AF_INET;
    addr_client.sin_port   = htons(PortOfClient);
```



```
    addr_client.sin_addr   = ip_addr_client;
    for(i=0;i<8;i++){
       addr_client.sin_zero[i]='0';
    }

    /* Add randomness to the port used by the server,Mvme167 */
    ip_addr_server.s_addr = 0;
    addr_server.sin_family = AF_INET;
    srand(*TICKTIME1);
    addr_server.sin_port = htons(1023+rand()/10);
    printf( "ephemeral port %d\n", addr_server.sin_port);
    addr_server.sin_addr   = ip_addr_server;
     for(i=0;i<8;i++){
       addr_server.sin_zero[i]='0';
    }

  if ((s=socket(AF_INET,SOCK_STREAM,TCP)) < 0) {
    perror( "socket failed:");
  }
  optval = 1;
  setsockopt(s,SOL_SOCKET,SO_REUSEADDR,(char *)&optval,sizeof(optval));
  if ((c=bind(s,&addr_server,16)) < 0) {
    perror( "bind failed:");
  }
  if ((c=connect(s,&addr_client,16)) < 0) {
    perror( "connect failed:");
  }
  if (!Blocking) {ioctl(s,FIONBIO,0);}

  return(s);
}


/* Construct a linked list of Two Instruction Blocks    */
/*    iblocks tell the ADM31 where the data buffers are */
void _build_iblocks( DPORT *theDport)
{
    ulong ptr167_iblockspace;

    theDport->ib_head = (HIBLOCK *)((ulong)iblockspace+AUX_MEM_VME);
    printf("ib_head = %x\n",(HIBLOCK *)((ulong)iblockspace+AUX_MEM_VME));

    iblockspace->data_seg = (ulong *)((ulong)dataspace_0+AUX_MEM_VME);
    iblockspace->nscans  = SCAN;
    iblockspace->next = (HIBLOCK *)((ulong)(iblockspace+1)+AUX_MEM_VME);

    (iblockspace+1)->data_seg = (ulong *)((ulong)dataspace_1+AUX_MEM_VME);
    (iblockspace+1)->nscans = SCAN;
    (iblockspace+1)->next = (HIBLOCK *)((ulong)(iblockspace+2)+AUX_MEM_VME);
```

```
    (iblockspace+2)->data_seg = (ulong *)((ulong)dataspace_2+AUX_MEM_VME);
    (iblockspace+2)->nscans = SCAN;
    (iblockspace+2)->next = (HIBLOCK *)((ulong)iblockspace+AUX_MEM_VME);
}

/* Send a command to the ADM31 and wait for result */
boolean _send_cmd( ubyte cmd )
{
    boolean result;
    ulong timeout;

    /* Clear Interrupt Flag */
    irq_flag = FALSE;

    sio->cmd_code = cmd;

    timeout = TIMEOUT;

    /* Loop until we get the interrupt or we time-out */
    while( !irq_flag)
    {
            timeout--;
            if ( ! timeout ) break;   /* Uh-Oh */
    }

    fflush((FILE *)1);
    result = irq_flag;
    return (result);
}

void _config_ADM31( DPORT *p )
{
    int i;
    int nchan=1;

    p->fp_gate      = FALSE;
    p->trig_type    = TRUE;     /* TRUE for ext trig, FALSE int trig */
    p->trig_exps    = FALSE;
    p->trig_edge    = FALSE;
    p->only_16      = FALSE;
    p->fair_req     = TRUE;
    p->busreq_level = 2;
    p->am_code      = 0x0f;

    /* What the hell is wrong with this? */
    /* modes 2 and 3 interfere with the operation of START_ACQ */
    p->acq_mode     = 1;

    p->irq_level    = 2;
    p->irq_vector   = 0x40;
```



```
    /* use channel number 2 only right now */
    for (i=0; i<NTIMERS; i++) p->nper_timer[i] = 0;
    p->nper_timer[1] = CHAN;
    for (i=0; i<CHAN; i++) p->t2_list[i] = i;

    /* 3MHz/prescale = sampling frequency */
    p->prescale = 300;

    for (i=0; i<NTIMERS; i++) p->timer_count[i] = 1;
}

/* Setup interrupt vector table to point to our functions */

void _install_irq( ulong level, ulong vector  )
{
    uword sr_value;
    ulong vbr_value;

    interrupt void (**vect)( void );

    /* Install handler for ADM31 interrupts */
    VectorPage[vector] = adm31_irq_handler;

    switch (level)
    {
        case 1: *IRQ_LEVEL = (1L <<  0); break;
        case 2: *IRQ_LEVEL = (1L <<  4); break;
        case 3: *IRQ_LEVEL = (1L <<  8); break;
        case 4: *IRQ_LEVEL = (1L << 12); break;
        case 5: *IRQ_LEVEL = (1L << 16); break;
        case 6: *IRQ_LEVEL = (1L << 20); break;
        case 7: *IRQ_LEVEL = (1L << 24); break;
        default: break;
    }

    if (level) *IRQ_ENABLE |= (1 << (level - 1));
    printf("IRQ installed\n");
}

/* Initialize MVME167 hardware */

void _init_mvme167(void)
{
    /* 1. MASTR_I1EN is the short IO, enabled for both the Alphi ADM31
                                                        and the ACROMAG AVME 945
       2. MASTR_I2EN is the A24/D16, enabled for LeCroy 1182
       3. MASTR_EN1 is the A32/D32, enabled for the the Alphi ADM31
       4. Nothing else is used                                          */
    *MASTR_CNTL = *MASTR_CNTL|(MASTR_I1EN|MASTR_I2EN|MASTR_EN1);
```

```
    *MASTR_CNTL = *MASTR_CNTL&(~(MASTR_EN2|MASTR_EN3|MASTR_EN4));

   /* set the address modifer code for 1 maps to ext sup data access 32 */
    *MASTR_AM    = 0x00000004;
    *MASTR_ADDR1 = 0x08000100;

    /* Choose Bus Req Lvl 0, Release on Req, and Fair Arbitration, page 2-51 */
    *ARBITRATER = 0x00000900;

    /* The Alphi ADM31 takes control of the bus at start so the MVME167 must
       be configured to act as a bus slave.  (see 167 manual, page 2-33) */
    *SLAVE_AM    = 0x000000a5;
    /* MVME167's address on A32 VME */
    *SLAVE_ADDR1 = 0x51005000;
    /* maps to the MVME167 0x00000000 to 0x01000000, 16 Mbytes */
    *SLAVE_OFFST1 = 0x0000ff00;
}
```

C.1.3 b_II.c

This was the C code that receive data from the VME crate and created summaries for online analysis.

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <errno.h>
#include <sys/time.h>
#include <unistd.h>
#include <sys/ioctl.h>
#include <signal.h>
#include <math.h>

typedef unsigned short  boolean;
#define TRUE   1
#define FALSE  0

#define SCALER_SIZE          1
#define CIADC_SIZE           1


#define MAX_BUFFER        6000   /* size of run 1200 is about 1 minute */
#define CHANNELS            24   /* number of channels per scan          */
#define TIME_BINS          100   /* number of time bins per macro pulse  */
```



```
#define MPULSES_per_BUFFER    1   /* number of macro-pulses per buffer    */
                          /* this is screwed because of the addition of the spin
                             sorter, must be set to 1 (or the sorter must be
                             turned off                                    */

#define SUBHEADER_SIZE          4                           /* ulong */
#define RUNHEADER_SIZE         38                           /* char  */
#define DATA_SIZE         (MPULSES_per_BUFFER*TIME_BINS*CHANNELS)/* ulong */
#define SUMMARY_HALF_SIZE      (TIME_BINS*(CHANNELS+2))     /* ulong */
#define SUMMARY_SIZE           2*SUMMARY_HALF_SIZE          /* ulong */
#define ASYM_BUF_HALF_SIZE     6*TIME_BINS
#define ASYM_BUF_SIZE          2*ASYM_BUF_HALF_SIZE

#define GAIN   100  /* gain of sum/diff stage */
#define RATIO  1  /* ratio of number of 'volts at ADC' to '# of gamma rays' */

/* Reference time for a data taking, Jan 1, 00:00:00, 1999 */
#define NPDG_epoch  915174000

#define RebootCmdFromPC     '1'
#define StartCmdFromPC      '2'
#define HaltCmdFromPC       '3'
#define ToggleTapeCmdFromPC '4'
#define ToggleScpCmdFromPC  '5'


int    run_number;              /* run number                          */
int    current_buffer;          /* current buffer number in the run    */
int    max_buffer;              /* run length in receive buffers       */
int    create_tape_file;        /* create binary file for auto-taping  */
int    create_summary_file;     /* create summary file for analysis    */

struct timeval  tv;             /* time and date variables             */
struct timezone tz;
long   npdg_epoch;

int    fd_archive;              /* archive file descriptor             */
int    fd_summary;              /* summary file descriptor             */
int    nw;                      /* write return value                  */

char   run_filename_ptr[14];    /* arrays for filenames                */
char   temp_run_filename_ptr[14];
char   summary_filename_ptr[14];

extern int errno;               /* error code for debugging            */

int    s_gui;                   /* socket function return values       */
int    s_vme;

int    r_gui;
```

```
int    r_vme;

char   CmdFromPC[1];

int    parent;
char   number_filename_ptr[14];
char   run_number_str[14];
int    fd_number;

/* Prototypes   */
void _wait_for_cmd(void);
void _data_run(void);

int  _setup_gui_socket(void);
int  _setup_vme_socket(void);

main(){
    sleep(1);

/* Open the socket to talk to the GUI */
   printf(" connecting to GUI \n");
   s_gui = _setup_gui_socket();

/* Open the socket to talk to the MVME167 */
   printf(" setting up server for VME \n");
   s_vme = _setup_vme_socket();

/* read the run number */
   sprintf(number_filename_ptr,"a_number");
   fd_number = open(number_filename_ptr,0);
   read(fd_number,run_number_str,14);
   run_number = atoi(run_number_str);
   printf("b_II.c initial run_number is %d\n",run_number);
   close(fd_number);

/* initialize a bunch of stuff            */
  create_tape_file    =    0;
  create_summary_file =    0;

  sprintf(temp_run_filename_ptr,"temp_run_file");

  printf("t=%d    s=%d\n",create_tape_file,create_summary_file);

  parent     = getppid();

  /* main loop */
  for(;;){
    _wait_for_cmd();
    _data_run();
  }
```



```
}

int  _setup_gui_socket(void){
  struct sockaddr_in addr_server;   /* setup gui socket                  */
  struct sockaddr_in addr_client;   /*                                   */
  struct in_addr     ip_addr_server;/*                                   */
  struct in_addr     ip_addr_client;/*                                   */
  int    addrlen;                   /*                                   */
  int    optval;                    /*                                   */
  int    rcvbuf;                    /*                                   */
  int    s,b,c,a,r,i;               /* socket function return values     */
  int    block;                      /* trying anything ??? */
  int    *ptr_to_block;

    ip_addr_server.s_addr  = htonl(0x7f000001);
    addr_server.sin_family = AF_INET;
    addr_server.sin_port   = htons(1204);
    addr_server.sin_addr   = ip_addr_server;
    for(i=0;i<8;i++){addr_server.sin_zero[i]='0';}

    ip_addr_client.s_addr  = 0;
    addr_client.sin_family = AF_INET;
    addr_client.sin_port   = htons(1203);
    addr_client.sin_addr   = ip_addr_client;
    for(i=0;i<8;i++){addr_client.sin_zero[i]='0';}

    addrlen = 16;
    optval = 1;
    rcvbuf = 32768;

    block = 1;
    ptr_to_block = &block;

  s=socket(AF_INET,SOCK_STREAM,IPPROTO_TCP);
  setsockopt(s,SOL_SOCKET,SO_REUSEADDR,(char *)&optval,sizeof(optval));

  b=bind(s,&addr_client,addrlen);
  if (b<0) {
    perror( "bind failed:");
  }
  c=connect(s,&addr_server,addrlen);
  if (c<0) {
    perror( "connect failed:");
  }
  if (ioctl(s,FIONBIO,ptr_to_block)) {
    perror( "ioctl failed:");
  }

  printf("socket ...%3d\n",s);
  printf("bind   ...%3d\n",b);
```

```
  printf("connect ...%3d\n",c);

  return(s);
}

int  _setup_vme_socket(void){
  struct sockaddr_in addr;          /* setup vme socket                  */
  struct in_addr     ip_addr;       /*                                   */
  int    addrlen;                   /*                                   */
  int    optval;                    /*                                   */
  int    rcvbuf;                    /*                                   */
  int    s,b,l,a,r,i;               /* socket function return values     */

    ip_addr.s_addr = INADDR_ANY;
    addr.sin_family = AF_INET;
    addr.sin_port   = htons(1201);
    addr.sin_addr   = ip_addr;
    for(i=0;i<8;i++){addr.sin_zero[i]='0';}
    addrlen = 16;
    optval = 1;
    rcvbuf = 32768;

  s=socket(AF_INET,SOCK_STREAM,IPPROTO_TCP);
    setsockopt(s,SOL_SOCKET,SO_REUSEADDR,(char *)&optval,sizeof(optval));
    setsockopt(s,SOL_SOCKET,SO_RCVBUF,(char *)&rcvbuf,sizeof(rcvbuf));

  b=bind(s,&addr,addrlen);
  if (b<0) {
    perror( "bind failed:");
  }
  l=listen(s,5);

  printf("socket ...%3d\n",s);
  printf("bind   ...%3d\n",b);
  printf("listen ...%3d\n",l);

  a=accept(s,&addr,&addrlen);
  printf("accept ...%3d\n",a);

  return(a);
}

void _wait_for_cmd(void){
  boolean WaitingForCmd;

  WaitingForCmd=TRUE;
  while(WaitingForCmd){
    usleep(10000);
    if ((r_gui=recv(s_gui,CmdFromPC,sizeof(CmdFromPC),0))>0){
      if (CmdFromPC[0]==RebootCmdFromPC){
```



```
      }
      else if (CmdFromPC[0]==StartCmdFromPC){
        WaitingForCmd=FALSE;
      }
      else if (CmdFromPC[0]==HaltCmdFromPC){
        printf("not taking data yet\n");
      }
      else if (CmdFromPC[0]==ToggleTapeCmdFromPC) {
        if (create_tape_file == 0) {
          create_tape_file = 1;
        }
        else {
          create_tape_file = 0;
        }
        printf("create_tape_file = %d\n",create_tape_file);
      }
      else if (CmdFromPC[0]==ToggleScpCmdFromPC) {
        if (create_summary_file == 0) {
          create_summary_file = 1;
        }
        else {
          create_summary_file = 0;
        }
        printf("create_summary_file = %d\n",create_summary_file);
      }
    }
  }
}


void _data_run(void){
  boolean end_the_run;
  char   runheader[RUNHEADER_SIZE]; /* array to store the run header        */
  ulong  subheader[SUBHEADER_SIZE]; /* array storing the macro-pulse header */
  ulong  data_buffer[DATA_SIZE];    /* array storing one data buffer        */
  ulong  *data_buffer_ptr;
  int    summary_buffer[SUMMARY_SIZE]; /* array storing the data summary   */
  int    *summary_buffer_ptr;
  double asymmetry_buffer[ASYM_BUF_SIZE];
  double *asymmetry_buffer_ptr;

  /* the detectors differences are ch00,ch01,ch02,ch03 for U,L,D,R and
       the sum is ch04 */
  double    U,L,D,R,S;
  double    number[2],*number_ptr;
  double    sum[4*TIME_BINS],*sum_ptr;
  double    sum_of_squares[4*TIME_BINS],*sum_of_squares_ptr;

  int    i,j,k;                     /* data sorting FOR iteration variables */
  int    to_get;                    /* number of bytes left to receive in a
```

```
            particular buffer                    */
int    endsend;
int    *endsend_ptr;

ulong  digital_IO;
ulong  *digital_IO_ptr;

int       scaler_data_buffer[SCALER_SIZE];
int       *scaler_data_buffer_ptr;
short int ciadc_data_buffer[CIADC_SIZE];
short int *ciadc_data_buffer_ptr;

short int channel23;
ulong     x;


      /* ------------------- */
/*  int       scaler_CH1_temp[MAX_BUFFER];
 short int ciadc_CH0_temp[MAX_BUFFER];
 int       *scaler_CH1_temp_ptr;
 short int *ciadc_CH0_temp_ptr;

 int       fd_scaler;
 int       fd_ciadc;   */
      /* ------------------- */
ulong  spin_up;


      /* ------------------- */
/*  scaler_CH1_temp_ptr = scaler_CH1_temp;
 ciadc_CH0_temp_ptr  = ciadc_CH0_temp;
 fd_scaler = creat("data_scaler",0644);
 fd_ciadc  = creat("data_ciadc",0644); */
      /* ------------------- */
printf("b.c::begin data run\n");
end_the_run = FALSE;

for (;;run_number++){
  /* Clear summary buffers */
  for (i=0;i<SUMMARY_SIZE;i++){
    summary_buffer[i]=0;
  }
  for (i=0;i<ASYM_BUF_SIZE;i++){
    asymmetry_buffer[i]=0;
  }


  /* Assign appropriate file names */
  if (create_tape_file){
    sprintf(run_filename_ptr,"run%#08d",run_number);
```



```
}
else {
  sprintf(run_filename_ptr,"na_run%#08d",run_number);
}

if (create_summary_file){
  sprintf(summary_filename_ptr,"sum%#08d",run_number);
}
else {
  sprintf(summary_filename_ptr,"na_sum%#08d",run_number);
}

/* Build the run header */
gettimeofday(&tv,&tz);
sprintf(runheader,"Run %8d     %s",run_number,ctime(&tv.tv_sec));
printf("%s\n", runheader);


/* Create archive file */
fd_archive = creat(temp_run_filename_ptr,0644);
if (fd_archive < 0) {
  perror( "create archive failed:");
}


/* Receive and Sort the Data */
for (current_buffer=0;current_buffer<MAX_BUFFER;current_buffer++){
  /* printf("current_buffer = %d\n",current_buffer); */
  gettimeofday(&tv,&tz);

  /* Build Subheader and write to Archive */
  subheader[0]=0xabcdabcd;
  subheader[1]=tv.tv_sec-MPDG_epoch;
  subheader[2]=tv.tv_usec;
  subheader[3]=0xabcd0000+current_buffer;

  nw=write(fd_archive,subheader,sizeof(subheader));
  if (nw < 0) {
    perror("write subheader to archive failed:");
  }

  endsend_ptr = &endsend;
  to_get = sizeof(endsend);
  while (to_get>0){
    r_vme=recv(s_vme,(char *)endsend_ptr,to_get,0);
    to_get -= r_vme;
    data_buffer_ptr += r_vme/sizeof(int);
  }

  if (endsend){
```

```
printf("endsend = %d\n",endsend);
        end_the_run = TRUE;
        break;
      }
      else {
        /* Receive charge integrating adc data */
        ciadc_data_buffer_ptr = ciadc_data_buffer;
        to_get = sizeof(ciadc_data_buffer);
        while (to_get>0){
          r_vme=recv(s_vme,(char *)ciadc_data_buffer,to_get,0);
          to_get -= r_vme;
          data_buffer_ptr += r_vme/sizeof(short int);
        }

        /* Receive scaler data */
        scaler_data_buffer_ptr = scaler_data_buffer;
        to_get = sizeof(scaler_data_buffer);
        while (to_get>0){
          r_vme=recv(s_vme,(char *)scaler_data_buffer,to_get,0);
          to_get -= r_vme;
          data_buffer_ptr += r_vme/sizeof(int);
        }

        /* Receive digital IO */
        digital_IO_ptr = &digital_IO;
        to_get = sizeof(digital_IO);
        while (to_get>0){
          r_vme=recv(s_vme,(char *)digital_IO_ptr,to_get,0);
          to_get -= r_vme;
          data_buffer_ptr += r_vme/sizeof(ulong);
        }
        digital_IO = ntohl(digital_IO);

/* Receive a ADC data until buffer is full*/
        data_buffer_ptr = data_buffer;
        to_get = sizeof(data_buffer);
        while (to_get>0){
          r_vme=recv(s_vme,data_buffer_ptr,to_get,0);
          to_get    -= r_vme;
          data_buffer_ptr += r_vme/sizeof(ulong);
        }
        /* ------------------- */
/*  *scaler_CH1_temp_ptr = scaler_data_buffer[0];
         *ciadc_CH0_temp_ptr  = ntohs(ciadc_data_buffer[0]);
         scaler_CH1_temp_ptr++;
         ciadc_CH0_temp_ptr++;   */
        /* ------------------- */

        /* The timing is screwed spin_up = digital_IO & 0x0001; */
        /* channel23 = (ntohl(data_buffer[1222])>>16); */
```



```
        x = ntohl(data_buffer[1222]);
        channel23 = (short)(x>>16);
        if (channel23 > 5000) {spin_up = 1;}
        else {spin_up = 0;}
        /* printf(" channel23 = %d     spin = %d \n",channel23,spin_up);*/

        data_buffer_ptr = data_buffer;

        if (spin_up) {
          summary_buffer_ptr = summary_buffer;
          asymmetry_buffer_ptr = asymmetry_buffer;
          number_ptr         = number;
          sum_ptr            = sum;
          sum_of_squares_ptr = sum_of_squares;
        }
        else {
          summary_buffer_ptr = summary_buffer+SUMMARY_HALF_SIZE;
          asymmetry_buffer_ptr = asymmetry_buffer+ASYM_BUF_HALF_SIZE;
          number_ptr         = number+1;
          sum_ptr            = sum+(2*TIME_BINS);
          sum_of_squares_ptr = sum_of_squares+(2*TIME_BINS);
        }

        *number_ptr       += 1;

        for (j=0;j<TIME_BINS;j++){
          for (k=0;k<CHANNELS;k++){
            *data_buffer_ptr = ntohl(*data_buffer_ptr);
            /* this line clips off the tbin and channel ID, casts
               the 2 byte data as a 4 byte data and adds it to
               the sum */
            *summary_buffer_ptr += (int)((short)(*data_buffer_ptr>>16));

            if (k==0) U = (double)((short)(*data_buffer_ptr>>16));
            if (k==1) L = (double)((short)(*data_buffer_ptr>>16));
            if (k==2) D = (double)((short)(*data_buffer_ptr>>16));
            if (k==3) R = (double)((short)(*data_buffer_ptr>>16));
            if (k==4) S = (double)((short)(*data_buffer_ptr>>16));

            data_buffer_ptr++;
            summary_buffer_ptr++;
          }
          /* tack digialIO and # of macro-pulses on the end*/
          *summary_buffer_ptr += 1;
          summary_buffer_ptr++;
          *summary_buffer_ptr = digital_IO;
          summary_buffer_ptr++;

/* calculate and store up/down asymmetry and error */
           *asymmetry_buffer_ptr+=(U-D)/(U+D+2*GAIN*S);
```

```
        *sum_ptr += *asymmetry_buffer_ptr;
        *sum_of_squares_ptr += pow(*asymmetry_buffer_ptr,2);
        asymmetry_buffer_ptr++;

        *asymmetry_buffer_ptr = (*number_ptr)*sqrt(
                    ((*number_ptr)*(*sum_of_squares_ptr)-pow(*sum_ptr,2))
                    /((*number_ptr)*(*number_ptr-1)));
        asymmetry_buffer_ptr++;

        *asymmetry_buffer_ptr+=2*sqrt((RATIO*(U+GAIN*S)*(D+GAIN*S))/
                          pow((U+D+2*GAIN*S),3.0));
        asymmetry_buffer_ptr++;

/* move sum pointers from UD to LR */
        sum_ptr++;
        sum_of_squares_ptr++;

       /* calculate and store left/right asymmetry and error */
        *asymmetry_buffer_ptr+=(L-R)/(L+R+2*GAIN*S);
        *sum_ptr += *asymmetry_buffer_ptr;
        *sum_of_squares_ptr += pow(*asymmetry_buffer_ptr,2);
        asymmetry_buffer_ptr++;

        *asymmetry_buffer_ptr = (*number_ptr)*sqrt(
                    ((*number_ptr)*(*sum_of_squares_ptr)-pow(*sum_ptr,2))
                    /((*number_ptr)*(*number_ptr-1)));
        asymmetry_buffer_ptr++;

        *asymmetry_buffer_ptr+=2*sqrt((RATIO*(L+GAIN*S)*(R+GAIN*S))/
                          pow((L+R+2*GAIN*S),3.0));
        asymmetry_buffer_ptr++;

/* move sum pointers from LR to UD of next time bin */
        sum_ptr++;
        sum_of_squares_ptr++;

      }

      /* Write to the Archive */
      nw=write(fd_archive,ciadc_data_buffer,sizeof(ciadc_data_buffer));
      if (nw < 0) {
        perror( "write ciadc to archive failed:");
      }
      nw=write(fd_archive,scaler_data_buffer,sizeof(scaler_data_buffer));
      if (nw < 0) {
        perror( "write scaler to archive failed:");
      }
      nw=write(fd_archive,&digital_IO,sizeof(digital_IO));
      if (nw < 0) {
        perror( "write digital_IO to archive failed:");
```



```
      }
      nw=write(fd_archive,data_buffer,sizeof(data_buffer));
      if (nw < 0) {
        perror( "write data to archive failed:");
      }

      /* Replace the Summary */
      fd_summary = creat(summary_filename_ptr,0644);
      nw=write(fd_summary,summary_buffer,sizeof(summary_buffer));
      nw=write(fd_summary,asymmetry_buffer,sizeof(asymmetry_buffer));
      if (nw < 0) {
        perror( "write to summary failed:");
      }
      close(fd_summary);
    }

    if ((r_gui=recv(s_gui,CmdFromPC,sizeof(CmdFromPC),0))>0){
      printf("r_gui=%d\n",r_gui);
      if (CmdFromPC[0]==RebootCmdFromPC){
      }
      else if (CmdFromPC[0]==StartCmdFromPC){
        printf("b.c :: already taking data\n");
      }
      else if (CmdFromPC[0]==HaltCmdFromPC){
        printf("b.c :: halting\n");
        end_the_run = TRUE;
        break;
      }
      else if (CmdFromPC[0]==ToggleTapeCmdFromPC) {
        if (create_tape_file == 0) {
          create_tape_file = 1;
        }
        else {
          create_tape_file = 0;
        }
      }

      else if (CmdFromPC[0]==ToggleScpCmdFromPC) {
        if (create_summary_file == 0) {
          create_summary_file = 1;
        }
        else {
          create_summary_file = 0;
        }
      }
    }
  }
      /* ---------------- */
/*    write(fd_scaler,scaler_CH1_temp,sizeof(scaler_CH1_temp));
   write(fd_ciadc,ciadc_CH0_temp,sizeof(ciadc_CH0_temp));
```

```
     close(fd_scaler);
     close(fd_ciadc); */
        /* ------------------- */
    rename(temp_run_filename_ptr,run_filename_ptr);
    close(fd_archive);
    if (end_the_run) {
      printf("kicking out of the loop\n");
      run_number++;
      kill(parent,SIGUSR1);
      kill(parent,SIGUSR2);
      break;
    }

    printf("incrementing run_number\n");
    kill(parent,SIGUSR1);
  }
}
```

C.1.4 auto_to_scp

This Perl script scp data files to the LENEUT

```
#! /usr/bin/perl -w
use File::Copy;

$flag = 1;
$SIG{"TERM"} = \&sigTERM;

#$destination = "csb\@leneut:data/TVOSFD";
$destination = "csb\@leneut:data/CVOCM";

$tape_position = 0;

while($flag){
  sleep 3;
  @run_files = <run*>;
  foreach $run_file (@run_files){
    system("scp $run_file $destination");
    system("rm $run_file");
  }

  @sum_files = <sum*>;
  foreach $sum_file (@sum_files){
    system("scp $sum_file $destination");
    system("rm $sum_file");
  }
```



```
  # if ($tape_position < 20) {
  #   system("scp $run_file $destination");
  # }
  #$tape_position++;


}

exit;

sub sigTERM {
  $flag = 0;
}
```

C.2 Analysis Code

C.2.1 trw.c

This C code sorts raw data files and produces a list of asymmetries, one per eight step spin sequence.

```
/* this version compiles the asym weighted by timebin, and will include
   a cut if the sum of the detectors differs by more than a set percentage
   within an eightstep.


   Same as two but will at an output for each detector so that the gains
   may be normalized.


   this version compiles the asym weighted by timebin and includes three
   cuts:

   eight step

   total detector sum

   relative detector sum

   It also places all of the output into one file with column headers
```

```
*/

#include <stdio.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <errno.h>
#include <sys/time.h>
#include <unistd.h>
#include <sys/ioctl.h>
#include <signal.h>
#include <math.h>


/* data file format constants */
#define SCALER_SIZE          1
#define CIADC_SIZE           1
#define MAX_BUFFER        6000   /* size of run 1200 is about 1 minute   */
#define SEQ                750
#define STEP                 8
#define CHANNELS            24   /* number of channels per scan         */
#define TIME_BINS          100   /* number of time bins per macro pulse  */
#define SUBHEADER_SIZE       4                                /* ulong */
#define DATA_SIZE        (TIME_BINS*CHANNELS)                 /* ulong */

/* default integration limits */
#define START_BIN           20
#define STOP_BIN            80

/* correct detector gains */
#define UtoD_ratio  3.8
#define LtoR_ratio  2.4

/* 3He NMR signal, determines neutron polarization */
#define NMR_PEAK          .040     /* in volts */

/* detector offsets */

  /* NO OFFSET */

#define cut         0
#define offset_u    0
#define offset_r    0
#define offset_d    0
#define offset_l    0


  /* Cl OFFSET */
/*
```



```
#define cut           1.08*1000000
#define offset_u   179
#define offset_r    86
#define offset_d    80
#define offset_l    83
*/

  /* Cd OFFSET */
/*
#define cut           1.15*1000000
#define offset_u   164
#define offset_r    79
#define offset_d    75
#define offset_l    73
*/

  /* La OFFSET */
/*
#define cut           0.48*1000000
#define offset_u   207
#define offset_r    94
#define offset_d    77
#define offset_l   132
*/

/* limits of acceptable deviation from the average detector sum of one eight step */
#define DETECTOR_SUM_LB 0.99
#define DETECTOR_SUM_UB 1.01


main(int argc,char *argv[]){
  /* input file structure variables */
  ulong     subheader[SUBHEADER_SIZE];       /* array for header           */
  short int ciadc_data_buffer[CIADC_SIZE];   /* "       " charge integrator*/
  int       scaler_data_buffer[SCALER_SIZE]; /* "       " scaler           */
  ulong     digital_IO;                      /* variable for TTL IO        */
  ulong     data_buffer[DATA_SIZE];          /* array for ADCs             */
  ulong     *data_buffer_ptr;

  /* time bin integration limits */
  int       start_bin,stop_bin;

  /* asymmetry correction variables */
  double   Polarization[TIME_BINS];
  double   RFSFeff[TIME_BINS];

  /* raw input channels */
  double   CH00;       /* U detector signal */
  double   CH01;       /* R detector signal */
  double   CH02;       /* D detector signal */
```

```
double   CH03;        /* L detector signal */

/* detector signal variables */
double   U[STEP][TIME_BINS];
double   R[STEP][TIME_BINS];
double   D[STEP][TIME_BINS];
double   L[STEP][TIME_BINS];

/* mean weighting variables */
double   NORM[TIME_BINS];
double   INTEGRAL;

/* input file variables */
int       nr;         /* "read()" return value       */
int       fd_arg;     /* input file from command line */

/* spin state variables */
int       time_bin;
int       seq;           /* # of 8 step sequence         */
int       step;          /* step within the sequence     */
short int channel23;     /* channel in which the spin state is stored */
int       spin_state[STEP];

/* spin sequence cut variables */
int       match;
int       good_spin_sequence;
int       number_good_spin_sequence;  /* # of sequences not ejected   */
int       the_spin_sequence[15];
int       shift;

/* detector sum cut variables */
double   int_sum[STEP];
double   int_sum_LB,int_sum_UB;
double   int_sum_avg;
int      good_detector_sum;
int      number_good_detector_sum;

/* individual detector signals */
double   U_sum_avg;
double   R_sum_avg;
double   D_sum_avg;
double   L_sum_avg;

/* an eightstep is good if it passes all cuts */
int      good;
int      number_good;

/* asymmetry calculation variables */
double    Uu[TIME_BINS];/* Capital-> detector , Small-> spin state */
double    Ru[TIME_BINS];
```



```
double    Du[TIME_BINS];
double    Lu[TIME_BINS];
double    Ud[TIME_BINS];
double    Rd[TIME_BINS];
double    Dd[TIME_BINS];
double    Ld[TIME_BINS];

double    frac;
double    asymUD[SEQ][TIME_BINS];    /* asym over time */
double    asymLR[SEQ][TIME_BINS];
double    rfsfUD[SEQ][TIME_BINS];    /* asym correction for RFSF over time */
double    rfsfLR[SEQ][TIME_BINS];
double    asymUDint[SEQ];            /* asym integrated over sequence */
double    asymLRint[SEQ];


/* output file pointers */
FILE      *fp_SUM;



/* ------------------------------------------------------------*/
for (time_bin=0;time_bin<TIME_BINS;time_bin++){
  Polarization[time_bin]=1;
  /* Polarization[time_bin] = tanh(.02276*(NMR_PEAK/.040)*.4*time_bin); */
  RFSFeff[time_bin]=1;
}
/* ------------------------------------------------------------*/


/* ------------------------------------------------------------*/
/* the defined eight step sequence for comparison to measured */
the_spin_sequence[ 0] = 1;
the_spin_sequence[ 1] = 0;
the_spin_sequence[ 2] = 0;
the_spin_sequence[ 3] = 1;
the_spin_sequence[ 4] = 0;
the_spin_sequence[ 5] = 1;
the_spin_sequence[ 6] = 1;
the_spin_sequence[ 7] = 0;
/* repeat the sequence -1 */
the_spin_sequence[ 8] = 1;
the_spin_sequence[ 9] = 0;
the_spin_sequence[10] = 0;
the_spin_sequence[11] = 1;
the_spin_sequence[12] = 0;
the_spin_sequence[13] = 1;
the_spin_sequence[14] = 1;
/* ------------------------------------------------------------*/
```

```
/* ==========================================================*/
/* assign intregration limits, use the default if there are no arguments */
start_bin=START_BIN; stop_bin=STOP_BIN;
if (argc>=3){
  start_bin=atoi(argv[2]); stop_bin=atoi(argv[3]);
}
/* ----------------------------------------------------------*/


/* ----------------------------------------------------------*/
/* open all files */
fd_arg = open(argv[1],0,0);
fp_SUM = fopen("c_sum","a");

/* print the header */
/*
fprintf(fp_SUM,"   ud    lr    u_avg    r_avg    d_avg    l_avg    sum \n");
*/
/*
fprintf(fp_SUM,"Cl_ud Cl_lr Cl_u_avg Cl_r_avg Cl_d_avg Cl_l_avg Cl_sum \n");
*/
/*
fprintf(fp_SUM,"Cd_ud Cd_lr Cd_u_avg Cd_r_avg Cd_d_avg Cd_l_avg Cd_sum \n");
*/
/*
fprintf(fp_SUM,"La_ud La_lr La_u_avg La_r_avg La_d_avg La_l_avg La_sum \n");
*/
/* ----------------------------------------------------------*/

number_good_spin_sequence = 0;


for (seq=0;seq<SEQ;seq++){

  U_sum_avg = 0;
  R_sum_avg = 0;
  D_sum_avg = 0;
  L_sum_avg = 0;

  for (step = 0; step<STEP; step++){
    /* load the data for one macro pulse from the file */
    nr = read(fd_arg,subheader,sizeof(subheader));
    nr = read(fd_arg,ciadc_data_buffer,sizeof(ciadc_data_buffer));
    nr = read(fd_arg,scaler_data_buffer,sizeof(scaler_data_buffer));
    nr = read(fd_arg,&digital_IO,sizeof(digital_IO));
    nr = read(fd_arg,data_buffer,sizeof(data_buffer));

    int_sum[step] = 0;
```



```
    /* get the spin */
    channel23 = (data_buffer[1222])>>16;
    if (channel23 > 5000) {spin_state[step] = 1;}
    else {spin_state[step] = 0;}

    for (time_bin=start_bin;time_bin<stop_bin;time_bin++){
      /* remove channel ID's from the data array */
      data_buffer_ptr = data_buffer+time_bin*CHANNELS;
      CH00 = ((double)((short)(*data_buffer_ptr>>16)));
      data_buffer_ptr++;
      CH01 = ((double)((short)(*data_buffer_ptr>>16)));
      data_buffer_ptr++;
      CH02 = ((double)((short)(*data_buffer_ptr>>16)));
      data_buffer_ptr++;
      CH03 = ((double)((short)(*data_buffer_ptr>>16)));
      /* place data from file into detector arrays */
      U[step][time_bin] = CH00-offset_u;
      R[step][time_bin] = CH01-offset_r;
      D[step][time_bin] = CH02-offset_d;
      /*R[step][time_bin] = LtoR_ratio*CH01-offset_r;
        D[step][time_bin] = UtoD_ratio*CH02-offset_d;*/
      L[step][time_bin] = CH03-offset_l;
      /* calculate the integral of the sum of all the detectors for each step of the sequence */
      int_sum[step] += U[step][time_bin] + R[step][time_bin] + D[step][time_bin] + L[step][time_bin];
      /* calculate the intregral of the individual detectors summed for all eight steps */
      U_sum_avg += U[step][time_bin];
      R_sum_avg += R[step][time_bin];
      D_sum_avg += D[step][time_bin];
      L_sum_avg += L[step][time_bin];
    }
  }

  /* normalize the averages so that they are average voltage integrated over one pulse */
  U_sum_avg *= 1/8.0;
  R_sum_avg *= 1/8.0;
  D_sum_avg *= 1/8.0;
  L_sum_avg *= 1/8.0;

  /* ==========================================================*/
  /* DETERMINE THE DETECTOR SUM CUT */
    int_sum_avg = 0;
    for(step = 0; step<STEP; step++){
      int_sum_avg += int_sum[step];
    }

    int_sum_avg *= 1/8.0;

    int_sum_LB = DETECTOR_SUM_LB*int_sum_avg;
    int_sum_UB = DETECTOR_SUM_UB*int_sum_avg;
```

```
      good_detector_sum = 1;

/* This is commented cut in order to do a no cut comparison
      for(step = 0; step<STEP; step++){
*/
        /* relatitve detector sum cut */
/*        if (int_sum_LB > int_sum[step]){
          good_detector_sum = 0;
        }
        if (int_sum_UB < int_sum[step]){
          good_detector_sum = 0;
        }
*/
        /* absolute detector sum cut */
/*        if (int_sum[step] < cut ){
          good_detector_sum = 0;
        }
      }
*/

      if (good_detector_sum) {
        number_good_detector_sum++;
      }


   /* ----------------------------------------------------------*/

   /* ----------------------------------------------------------*/
   /* DETERMINE THE SPIN SEQUENCE CUT */
   good_spin_sequence  = 0;
   for (shift=0;shift<STEP;shift++){
     match = 0;
     for (step=0;step<STEP;step++){
       if(spin_state[step] == the_spin_sequence[step+shift]){
         match += 1;
       }
     }
     if (match == 8) {
      good_spin_sequence = 1;
      number_good_spin_sequence++;
     }
   }
   /* ----------------------------------------------------------*/

   /* ----------------------------------------------------------*/
   /* combine all the cuts */
   good = 0;
   if (good_detector_sum & good_spin_sequence){
     good = 1;
     number_good++;
```



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



   if (good){
     for (time_bin=start_bin;time_bin<stop_bin;time_bin++){
       Uu[time_bin] = 0;
       Ru[time_bin] = 0;
       Du[time_bin] = 0;
       Lu[time_bin] = 0;
       Ud[time_bin] = 0;
       Rd[time_bin] = 0;
       Dd[time_bin] = 0;
       Ld[time_bin] = 0;
     }

     for (time_bin=start_bin;time_bin<stop_bin;time_bin++){
       for (step=0;step<STEP;step++){
         NORM[time_bin] += U[step][time_bin]+D[step][time_bin];
         INTEGRAL       += U[step][time_bin]+D[step][time_bin];
       }
     }

     for (time_bin=start_bin;time_bin<stop_bin;time_bin++){
       for (step=0;step<STEP;step++){
         if (spin_state[step]) {
           Uu[time_bin] += U[step][time_bin];
           Ru[time_bin] += R[step][time_bin];
           Du[time_bin] += D[step][time_bin];
           Lu[time_bin] += L[step][time_bin];
         }
         else {
           Ud[time_bin] += U[step][time_bin];
           Rd[time_bin] += R[step][time_bin];
           Dd[time_bin] += D[step][time_bin];
           Ld[time_bin] += L[step][time_bin];
         }
      }

       frac = sqrt((Uu[time_bin]*Dd[time_bin])/(Ud[time_bin]*Du[time_bin]));
      asymUD[seq][time_bin] = (frac-1)/(frac+1);
       frac = sqrt((Lu[time_bin]*Rd[time_bin])/(Ld[time_bin]*Ru[time_bin]));
      asymLR[seq][time_bin] = (frac-1)/(frac+1);

      frac = NORM[time_bin]/(((1+RFSFeff[time_bin])/2)*Polarization[time_bin]);
      asymUDint[seq] += frac*asymUD[seq][time_bin];
      asymLRint[seq] += frac*asymLR[seq][time_bin];

     }
```

```
      asymUDint[seq] *= 1/INTEGRAL;
      asymLRint[seq] *= 1/INTEGRAL;

     /* write asymmetries to the output files */
     if (asymUDint[seq]>=0) {fprintf(fp_SUM," %1.15e ",asymUDint[seq]);}
       else {fprintf(fp_SUM,"%1.15e ",asymUDint[seq]);}
     if (asymLRint[seq]>=0) {fprintf(fp_SUM," %1.15e ",asymLRint[seq]);}
       else {fprintf(fp_SUM,"%1.15e ",asymLRint[seq]);}

     fprintf(fp_SUM,"%1.15e   %1.15e   %1.15e   %1.15e   %1.15e\n",
            U_sum_avg,R_sum_avg,D_sum_avg,L_sum_avg,int_sum_avg);
    }
  }

  printf("good_spin_sequence=%d  ||  good_detector_sum=%d  ||  good=%d\n",
         number_good_spin_sequence, number_good_detector_sum, number_good);


  fclose(fp_SUM);

  close(fd_arg);
}
```

C.2.2 MCNP input files

These files were the input to MCNP for the purpose of making the solid angle calculation.

```
geometry factor
c this program determines the cos*energy of the flux into the detector
c and the total flux into the detector and the total energy deposited
c in the detector
c cell cards
  1 0   1                                                    $ the endless void
  2 0   (-10:18:-20:28:-30:32) (-41:42:43) (-51:52:-43:53) -1 $ a bounded void
 11 1 -4.51   10 -18 20 -28 31 -32          $ the up detector
 21 2 -8.65   41 -42 -43                    $ the target
 31 3 -.301   51 -52  43 -53                $ 6Li neutron shield
 70 0         10 -18 20 -28 30 -31    -70
 71 0         10 -18 20 -28 30 -31 70 -71
 72 0         10 -18 20 -28 30 -31 71 -72
 73 0         10 -18 20 -28 30 -31 72 -73
 74 0         10 -18 20 -28 30 -31 73 -74
 75 0         10 -18 20 -28 30 -31 74 -75
 76 0         10 -18 20 -28 30 -31 75 -76
```



```
 77 0         10 -18 20 -28 30 -31 76 -77
 78 0         10 -18 20 -28 30 -31 77 -78
 79 0         10 -18 20 -28 30 -31 78 -79
 80 0         10 -18 20 -28 30 -31 79 -80
 81 0         10 -18 20 -28 30 -31 80 -81
 82 0         10 -18 20 -28 30 -31 81 -82
 83 0         10 -18 20 -28 30 -31 82 -83
 84 0         10 -18 20 -28 30 -31 83 -84
 85 0         10 -18 20 -28 30 -31 84 -85
 86 0         10 -18 20 -28 30 -31 85 -86
 87 0         10 -18 20 -28 30 -31 86 -87
 88 0         10 -18 20 -28 30 -31 87 -88
 89 0         10 -18 20 -28 30 -31 88

c surface cards
 1 so 10000      $ edge of the universe
10 px   -7.5     $ front of the detector array
18 px    7.5     $ back of the detector array
20 py   -7.5     $ vertical detector planes
28 py    7.5
30 pz    7.499
31 pz    7.5     $ horizontal detector planes
32 pz   22.5
41 px   -0.04    $ front of target
42 px    0.04    $ back
43 cx    6.0     $ cylindrical side
51 px   -8.0     $ front of neutron shield
52 px    8.0     $ back
53 cx    7.4     $ cylindrical side
60 px  -10.0     $ source surface
70 cz    0.5     $ tally surface segmenters
71 cz    1.0
72 cz    1.5
73 cz    2.0
74 cz    2.5
75 cz    3.0
76 cz    3.5
77 cz    4.0
78 cz    4.5
79 cz    5.0
80 cz    5.5
81 cz    6.0
82 cz    6.5
83 cz    7.0
84 cz    7.5
85 cz    8.0
86 cz    8.5
87 cz    9.0
88 cz    9.5
```

```
c other stuff
MODE N P
IMP:N 0 1 23r     $ neutrons are not important in the endless void
IMP:P 0 1 23r     $ photons are not important in the endless void
SDEF POS=-10 0 0 SUR 60 DIR 1 RAD D1 ERG 2.280535e-09
 SI1 H 0  0.254 0.508 0.762 1.016  1.270 1.524 1.778 2.032 2.286  2.540
 SP1 C 0  .01  .04  .09  .16  .25  .36  .49  .64  .81 1.00
M1  55133.60c 0.5 &   $ this is CsI 4.510
    53127.60c 0.5
M2  48000.42c 1.0   $ target material, Cd 8.650
M3   3006.60c 1.0     $ 6Li shield .301
*F001:P 31
 C001  .00 .05 .10 .15 .20 .25 .30 .35 .40 .45 &
       .50 .55 .60 .65 .70 .75 .80 .85 .90 .95 1
 FS001 -70 -71 -72 -73 -74 -75 -76 -77 -78 -79 &
       -80 -81 -82 -83 -84 -85 -86 -87 -88
F706:P 11
 CF706 70
F716:P 11
 CF716 71
F726:P 11
 CF726 72
F736:P 11
 CF736 73
F746:P 11
 CF746 74
F756:P 11
 CF756 75
F766:P 11
 CF766 76
F776:P 11
 CF776 77
F786:P 11
 CF786 78
F796:P 11
 CF796 79
F806:P 11
 CF806 80
F816:P 11
 CF816 81
F826:P 11
 CF826 82
F836:P 11
 CF836 83
F846:P 11
 CF846 84
F856:P 11
 CF856 85
F866:P 11
 CF866 86
```



```
F876:P 11
 CF876 87
F886:P 11
 CF886 88
F896:P 11
 CF896 89
PRDMP 5000000 5000000 1
NPS   5000000
```

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] V. A. Vesna, E. A. Kolomenskii, V. M. Lobashov, V. A. Nazarenko, A. N. Pirozhkov, L. M. Smotritsky, Yu. V. Sobolev, and N. A. Titov, *Observationof parity nonconservation in the integrated γ spectrum from (n,γ) reactions in Cl, Br, Cd, Sn, and La nuclei*, JETP Lett. **36**, (1982) p169

[2] E. A. Kolomenskii, V. M. Lobashov, A. N. Pirozhkov, L. M. Smotritsky, N. A. Titov, V. A. Vesna, *Observation of parity violating effects in the polarized thermal neutron total and radiative capture cross sections of ^{117}Sn and ^{139}La*, Physics Letters, **107B**, (1981) p272

[3] M. Avenier, G. Baglieu, H. Benkoula, J. F. Cavaignac, A. Idrissi, D. H. Koang, B. Vignon, Richard Wilson, *Parity non-conservation in the radiative capture of polarized neutrons by ^{35}Cl*, Nuclear Physics, **A436**, (1985) p83

[4] W. M. Snow, A. Bashenov, C. S. Blessinger, J.D. Bowman, T. E. Chupp, K. P. Coulter, S. J. Freedman, B. K. Fujikawa, T. R. Gentile, G. L. Green, g. Hansen, G. E. Hogan, S. Ishimoto, G. L. Jones, J. N. Knudson, E. Kolomenski, S. K. Lamoreaux, M. B. Leuschner, A. Masaike, Y. Masuda, Y. Matsuda, G. L. Morgan, K. Morimoto, C. L. Morris, H. Nann, S. I. Pentilla, A. Pirozhkov, V. R. Pomeroy, D. R. Rich, A. Serebrov, E. I. Sharapov, D. A. Smith, T. B. Smith,

R. C. Welsh, F. E. Wietfeldt, W. S. Wilburn, V. W. Yuan, J. Zerger, *Progress toward a new measurement of the parity violating asymmetry in* $\vec{n}+p \rightarrow d+\gamma$, Nuclear Instruments and Methods in Physics A, **A440** (2000) p729-735

[5] W. S. Wilburn, J. D. Bowman, *Consistency of parity-violating pion-nucleon couplings extracted from measurements in* $^{1}8F$ *and* $^{1}33Cs$, Physical Review C57 (1998) p3435-3429

[6] E. G. Adelberger and W. C. Haxton, Ann. Rev. Nucl. Part. Sci. **35**, 501(1985)

[7] B. Desplanques, J. F. Donghue, and B. R. Holstein, Ann. Phys. **124**, 449(1980)

[8] W. C. Haxton and C. E. Wieman, *Atomic Parity Nonconservation and Nuclear Anapole Moments*, arXiv:nucl-th/0104026 6 Apr 2001

[9] V. V. Flambaum and O. P. Sushkov, *Angular and Polarization Correlation in the (n,*γ*) Reaction*, Nuclear Physics A435 (1985) p352-380

[10] V. P.Alfimenkov, L. Lason, Yu. D. Mareev, V. V. Novitsky, L. B. Pikelner, V. R. Skoy, M. I. Tsulaya, and A. N. Chernikov, *Parity-Violation Effect in the Interaction of Neutrons with Polarized Lanthanum*, Physics of Atomic Nuclei, **59**, No.11, 1861-1864 (1996)

[11] V. E. Bunakov, V. P. Gudkov, S. G. Kadmenskii, I. A. Lomachenkov, and V. I. Furman, *Parity nonconservation effects in radiative-capture cross-sections*, Yad. Fiz. **40**, 188-196 (July 1984)

[12] A.R. Berdoz, et al., TRIUMPH E497 Collaboration, *Parity Violation in Proton-Proton Scattering at 221MeV*, preprint: arXiv:nucl-ex/0107014 19Jul2001

[13] R. Balzer *et al.*, Phys. Rev. C **30**, 1409 (1984).

[14] A. Csoto, B. F. Gibson, and G. L. Payne, Physical Review C **56**, 631 (1997)

[15] W. Happer, *Polarization of the nuclear spins of noble gas atoms by spin exchange with optically pumped alkali metal atoms*, Physical Review A29,(1984) p3092

[16] T. G. Walker, W. Happer, *Spin exchange optical pumping of noble gas nuclei*, Reviews of Modern Physics 69,(1997) p629

[17] V. A. Vesna, Yu M. Gledenov, P. V. Lebedev-Stepanov, I. S. Okunev, A. V. Sinyakov, Yu. M. Tchuvil'sky, and E. V. Shulgina, *Searches for Parity-Violation Effects in the* γ *Transistion* $^{7}Li^{*} \rightarrow \gamma(M1) \rightarrow {}^{7}Li(E_{\gamma} = 0.478\ MeV)$, Physics of Atomic Nuclei, Vol. 62, No. 3, 1999, p522-532

[18] G. Mitchell, *Magnetic Field Effects on NPDGamma Photodiodes* Internal NPDGamma Technote

[19] O. Schaerpf, *Comparison of Theoretical and Experimental Behaviour of Supermirrors and Discussion of Limitations*, Physica B, **157** (1989) p631-638

[20] S. D. Penn, E. G. Adelberger, B. R. Heckel, D. M. Markoff, H. E. Swanson, *A low-noise* ^{3}He *ionization chamber for measuring the energy spectrum of a cold neutron beam*, Nuclear Instruments and Methods in Physics Research, **A457**, (2001) p332-337

[21] N. R. Roberson, C. D. Bowman, J. D. Bowman, P. P. J. Delheij, C. M. Frankle, C. R. Gould, D. G. Haase, J. N. Knudson, G. E. Mitchell, R. N. Mortensen, S. I. Penttila, H. Postma, S. J. Seestrom, J. J. Szymanski, S. A. Wender, S. H.

Yoo, V. W. Yuan, X. Zhu, *An apparatus and techniques of tests for fundamental symmetries in compound-nucleus scattering with epithermal polarized neutron beams*, Nuclear Instruments and Methods in Physics Research, **A326**, (1993) p549-565

[22] V. F. Sears, *Thermal-Neutron Scattering Lengths and Cross Sections for Condensed-Matter Research*, Atomic Energy of Canada Limited, June 1984

[23] J. Byrne, *Neutrons, Nuclei and Matter*, Institute of Physics Publishing Bristol and Philadelphia, (1994)

[24] P. A. Krupchitsky, *Fundamental Research with Polarized Slow Neutrons*, Springer-Verlag Berlin Heidelberg (1987)

[25] K. Grotz and H. V. Klapdor, *The Weak Interaction in Nuclear, Particle and Astrophysics*, IOP Publishing Ltd, 1990

[26] M. Ishii and M. Kobayashi, *Single Crystals for Radiation Detectors*, KEK Preprint 92-10, April 1992

[27] Piotr A. Rodnyi, *Physical Processes in Inorganic Scintillators*, CRC press LLC (1997)

[28] Charles P. Slichter *Principles of Magnetic Resonance*, Harper & Row, New York, (1963)

[29] Philip R. Bevington and D. Keith Robinson, *Data Reduction and Error Analysis for the Physical Sciences*, McGraw-Hill, Inc., New York, (1992)

Christopher S. Blessinger

Contact Information

Indiana University Cyclotron Facility

2401 Milo B. Sampson Lane, Bloomington, IN 47408

(812)855-2918

chblessi@indiana.edu

Education

Ph.D. Nuclear Physics, Indiana University, Bloomington, IN, October 2001

dissertation: *Measurement of the Parity Violating Gamma Ray Asymmetry in Polarized Neutron Capture on* ^{35}Cl, ^{113}Cd, *and* ^{139}La

M.S. Physics, Indiana University, Bloomington, IN, 1998

B.S. Physics, Rose-Hulman Institute of Tech., Terre Haute, IN, 1995

B.S. EE, Rose-Hulman Institute of Tech., Terre Haute, IN, 1995

Academic Positions

Teaching Assistant, Indiana University, 1996-1997

Research Assistant, Indiana University Cyclotron Facility, 1997-1998

Graduate Research Assistant, Los Alamos National Laboratory, 1998-2001

Memberships

American Physical Society

Publications

W. M. Snow, A. Bashenov, C. S. Blessinger, J.D. Bowman, T. E. Chupp, K. P. Coulter, S. J. Freedman, B. K. Fujikawa, T. R. Gentile, G. L. Greene, G. Hansen, G. E. Hogan, S. Ishimoto, G. L. Jones, J. N. Knudson, E. Kolomenski, S. K. Lamoreaux, M. B. Leuschner, A. Masaike, Y. Masuda, Y. Matsuda, G. L. Morgan, K. Morimoto, C. L. Morris, H. Nann, S. I. Pentilla, A. Pirozhkov, V. R. Pomeroy, D. R. Rich, A. Serebrov, E. I. Sharapov, D. A. Smith, T. B. Smith, R. C. Welsh, F. E. Wietfeldt, W. S. Wilburn, V. W. Yuan, J. Zerger, *Progress toward a new measurement of the parity violating asymmetry in* $\vec{n} + p \rightarrow d + \gamma$, Nuclear Instruments and Methods in Physics, **A440** (2000) p729-735

Conference Proceedings

W. S. Wilburn, A. Bazhenov, C. S. Blessinger, J. D. Bowman, T. E. Chupp, K. P. Coulter, S. J. Freedman, B. K. Fujikawa, T. R. Gentile, G. L. Greene, G. Hansen, G. E. Hogan, S. Ishimoto, G. L. Jones, J. N. Knudson, E. Kolomenski, S. K. Lamoreaux, M. B. Leuschner, A. Masaike, Y. Masuda, Y. Matsuda, G. L. Morgan, K. Morimoto, C. L. Morris, H. Nann, S. I. Penttila, A. Pirozhkov, V. R. Pomeroy, D. R. Rich, A. Serebrov, E. I. Sharapov, D. A. Smith, T. B. Smith, W. M. Snow, R. C. Welsh, F. E. Weitfeldt, V. W. Yuan, and J. Zerger, **Measurement of the Parity Violating Asymmetry** A_γ **in** $\vec{n}+p \rightarrow d+\gamma$, Proceedings of the 15th International Conference on the Application of Accelerators in Research and Industry, ed. J. L. Duggan and I. L. Morgan, AIP conference proceeding #475, 247 (1998).

Contributed Abstracts

C. S. Blessinger (for NPDG collaboration), *Parity Violation in Polarized Neutron Capture on Protons:Target and Detector Simulations*, Bulletin of the American Physical Society(April 1998)

C. S. Blessinger (for NPDG collaboration), *Measurement of parity violating* γ*-ray asymmetries of polarized neutron capture in* ^{35}Cl *and* ^{139}La, Bulletin of the American Physical Society(April 2001, Vol.46, No.2, p71)