



**An Amalgamation of work on
the Ultra-Cold Neutron Source
and Neutron Electric Dipole
Moment Experiment at
TRIUMF**

Nicholas Christopher

Mechanical Engineering Student, University of Waterloo

September 5, 2016

Contents

List of Figures	iv
List of Tables	vi
1 Introduction	7
1.1 UCN Source	7
1.2 nEDM experiment	10
2 PENTrack	14
2.1 Background	14
2.2 Implementation of general B_0 and B_1 fields	14
2.2.1 Explanation of C++ code	15
2.2.2 Verification of Performance	19
2.3 Quartz Window Study	21
2.4 Horizontal EDM Cell Study	23
2.5 PENTrack Verification - Ramsey Cycle Simulation	30
2.5.1 Comparison with Theoretical Model	30
2.6 The Effect of Magnetic Field Gradients on the Ramsey Cycle .	32
2.6.1 Background	32
2.6.2 Ramsey-Bloch-Siegert Shift	33
2.6.3 Bloch-Siegert Shift	34
2.6.4 Shift due to Gravity	35
2.6.5 Function Fitting	35
2.6.6 Simulation of Various Magnetic Field Gradients	36
2.6.7 Simulation of the GPE	38
2.7 Future PENTrack Work	41
3 Kicker Control	43
3.1 Background	43
3.2 Commissioning the Kicker	45
3.3 Demodulation	45
3.3.1 IQ Demodulation	47
3.3.2 Precision Rectifier Demodulation	48
3.3.3 Digital Demodulation	50
3.4 Kicker Timing Module (KTM)	50
3.4.1 ADC	53

3.4.2	FPGA Programming	53
3.4.3	HPS Programming	61
3.5	Guide for Operating the KTM	65
3.5.1	Uploading FPGA and C programs to the KTM	65
3.5.2	HPS communication via USB	66
3.5.3	Recording Data	66
4	EDM Cell Coatings	70
4.1	Background	70
4.1.1	Solution Mixing	71
4.2	Procedure for Si wafer coatings	71
4.2.1	Cleaning	71
4.2.2	Deposition	72
4.3	Results from PE/PS Coatings	72
4.3.1	PS Coatings	73
4.3.2	PE Coatings	73
4.4	Results from DPE/DPS Coatings	73
5	Testing of UCN's Cryogenic Pressure Switch	75
5.1	Background	75
5.2	Experimental Set-up	75
5.3	Findings	76

List of Figures

1	UCN sources around the world [2]	8
2	UCN beamline layout [2]	9
3	UCN production cross section [2]	9
4	experimental nEDM upper limits [4]	11
5	geometry.in file	15
6	edmfields.cpp	18
7	fields.cpp	19
8	B_0 field verification of transformations	20
9	B_0 field verification of origin offset	20
10	B_1 field verification of performance	21
11	Storage lifetime: comparison between ROOT program and PENTrack	22
12	Storage lifetime: comparison between normalized ROOT pro- gram and PENTrack	22
13	Vertical EDM cell geometry	24
14	Horizontal EDM cell geometry	24
15	Horizontal EDM cell filling efficiency	26
16	Vertical EDM cell filling efficiency	27
17	Horizontal EDM cell UCN total energy distribution	28
18	Vertical EDM cell UCN total energy distribution	28
19	Vertical EDM cell UCN kinetic energy distribution	29
20	Vertical EDM cells, false edm comparison	29
21	Horizontal EDM cells, false edm comparison	30
22	Ramsey cycle with circularly rotating B_1 field	31
23	Ramsey cycle with B_1 field oscillating in 1D	32
24	Ramsey cycle with and without magnetic gradients (gradients are in nT/m , in the z direction)	36
25	Ramsey cycles with $\frac{\partial B_{0z}}{\partial z} = 1nT/m$, $E \uparrow$ compared to $E \downarrow$ (Neutrons per point $\approx 1,000,000$)	39
26	Beam Structure [1]	43
27	Kicker rise time	45
28	Kicker fall time	46
29	Kicker cycle	46
30	AD8333 performance	48
31	OPA698, 1MHz sine wave	49
32	OPA698, 46MHz sine wave	50

33	KTM trigger output, large time scale	51
34	KTM trigger output, small time scale	52
35	KTM trigger output, triggering voltage	52
36	Pulser state diagram	55
37	1VM4 state diagram	56
38	Trigger state diagram	57
39	State change scenarios	58
40	Mean period and trigger offset flowchart	59
41	trigger switching flowchart	60
42	FPGA error conditions	63
43	HPS code flowchart	64
44	USB Connection Settings	66
45	Ethernet Connection Settings	67
46	Dektak plot of surface roughness	74

List of Tables

1	Magnetic Gradient Fit Results	37
2	Magnetic Gradient Larmor Shifts Comparison	37
3	GPE Fit Results	39
4	AD8333 performance	49
5	Polyethylene Coatings	77
6	Polystyrene Coatings	78

1 Introduction

This section is intended to give the reader a general understanding of the Ultra-Cold Neutron (UCN) source and the Neutron Electric Dipole Moment (nEDM) experiment at TRIUMF. Background information specifically about the projects discussed in this report are included in their respective background subsections.

1.1 UCN Source

Neutrons are hard to detect because they do not interact with matter as frequently as other particles such as protons. This is in part due to their zero charge. If neutrons are travelling slow enough, they can have a wavelength that is larger than the spacing between atoms within many materials. Therefore, neutrons can be reflected off the surface of various materials. The equation that is used to determine if reflection will occur is:

$$E \sin^2 \theta \leq V \tag{1}$$

where E is the neutron's kinetic energy, θ is the angle between the incoming neutron and the material, and V is the Fermi potential of the material.

Suppose a source of ultra-cold neutrons is created with a distribution of neutron energies which satisfy Equation 1, for $\theta = 90^\circ$. Neutrons can then be stored for their entire lifetime within material walls, and experiments can be done to measure the properties of these neutrons. Several UCN sources have been created or are under construction around the world (Figure 1 [2]). One of the most challenging aspects of a UCN source is to obtain a high UCN density. By creating a dense neutron source, many data points can be gathered in any subsequent experiments, which will result in a small statistical error.

The UCN source layout at TRIUMF is shown in Figure 2 [2]. The beam-line begins at the kicker magnet, where a portion of the protons from BL1A are "kicked" away from BL1A and into BL1U. A bending magnet then orients the beam towards a target. Subsequent quadropole magnets keep the beam directed towards the target. The target is made of tungsten, and is water cooled. When the proton beam hits the target, spallation neutrons are produced. these neutrons then pass through liquid D₂O at 300°K, and solid

	RAL/ SUSSEX /ILL (Grenoble, FR)	PSI (Villigen, CH)	TUM (Munich, DE)	US (Oakridge)	CryoILL (Grenoble, FR)	Russian (Grenoble, FR \Rightarrow Dubna, RU)	TRIUMF (Vancouver, CA)
temp	RT	RT	RT	0.7 K	0.7 K	RT	RT
comag	Hg	Hg	Hg	^3He	none	none	Xe+Hg
source	reactor, turbine	spall., sD ₂	reactor, sD ₂	spall, internal ^4He	reactor, internal ^4He	reactor, turbine, (^4He)	spall., ^4He
nr of cells	1	2	2	2	4	>1	1-2
goal [ecm]	$3 \cdot 10^{-26}$	$5 \cdot 10^{-28}$	$5 \cdot 10^{-28}$	$3 \cdot 10^{-28}$	$1.6 \cdot 10^{-27}$	$3 \cdot 10^{-28}$	$< 10^{-27}$
date	2006	2018	2019	2020	2015	2018	2019

Figure 1: UCN sources around the world [2]

D₂O at 10 - 30°K in order to slow them down. D₂O is used instead of H₂O because hydrogen absorbs neutrons. The neutrons then enter a superfluid helium bottle which is cooled to 0.7°K. At this point the neutrons are travelling slow enough that they are "ultra-cold", and will stay inside the helium bottle due to Equation 1. Figure 3 shows a cross section of the UCN creation [2].

Once the helium bottle is filled for a sufficient amount of time, a door will open which releases the neutrons into the UCN beamline. They pass through a polarizer, which allows only one spin state through, and polarizes the neutrons. The beamline then splits into two beams. One beam is intended for the nEDM experiment, and the other can be used for experiments proposed in the future.

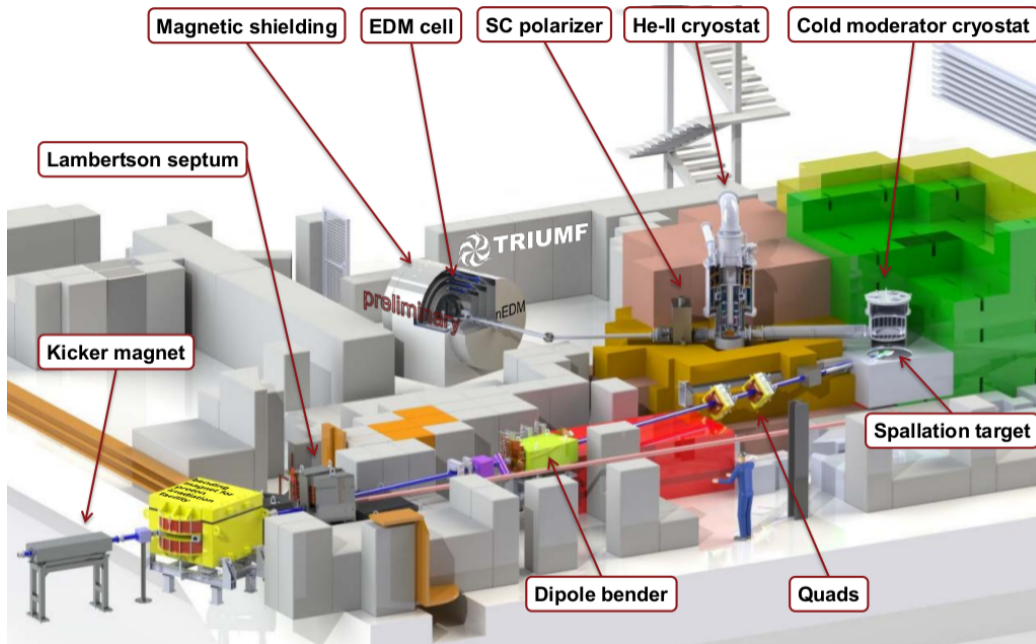


Figure 2: UCN beamline layout [2]

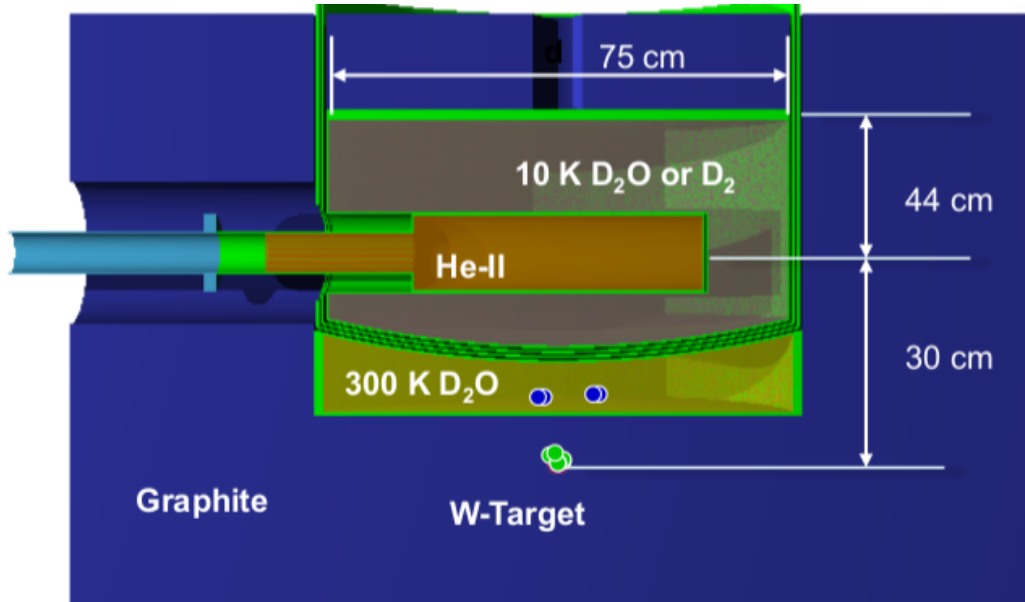


Figure 3: UCN production cross section [2]

1.2 nEDM experiment

One of the great unanswered questions of physics is why there is more matter than anti-matter in the universe. In the big bang, matter and anti-matter should have been created in equal parts. This means that eventually all the matter and antimatter should recombine to create energy. However, this did not happen, and matter is far more abundant than anti-matter in our observable universe.

One explanation for the matter anti-matter asymmetry in the universe is CP (charge-conjugation, and parity) violation. Particles should be CP invariant. If they are not, then this violation can contribute to the observable matter to anti-matter asymmetry. Note that a CP violation also infers a T (time) violation due to CPT symmetry. It has been experimentally proven that there is CP violation in the weak force [3], however it is not enough to explain the asymmetry in the universe. There are other places where experimentalists are looking for CP violation. One potential CP violation is in neutrinos. Another place likely to have CP violation is a non-zero value of the neutron electric dipole moment. An experiment to measure this value is currently under construction at TRIUMF.

There have been many experiments conducted with the purpose of measuring the nEDM. Unfortunately, none of them have been sensitive enough to measure the nEDM. The current upper limit on the nEDM comes from an experiment at ILL, which measured $d_n < 2.9 \times 10^{-26} ecm$ [5]. This upper limit corresponds to a value of the nEDM which Supersymmetry predicts, as shown in Figure 4 [4]. The standard model predicts a much lower nEDM value. If an experiment can measure the value of the nEDM (or reduce its upper limit), it could rule out many theories of physics, and put strong constraints on others.

The nEDM experiment at TRIUMF will use Ramsey's method of separated oscillatory fields. In this method the neutrons enter the nEDM cell polarized in the direction of a magnetic field (B_{0z}) with a magnitude of approximately $1\mu T$. The Larmor precession is then:

$$\omega_0 = -\gamma B_{0z} \tag{2}$$

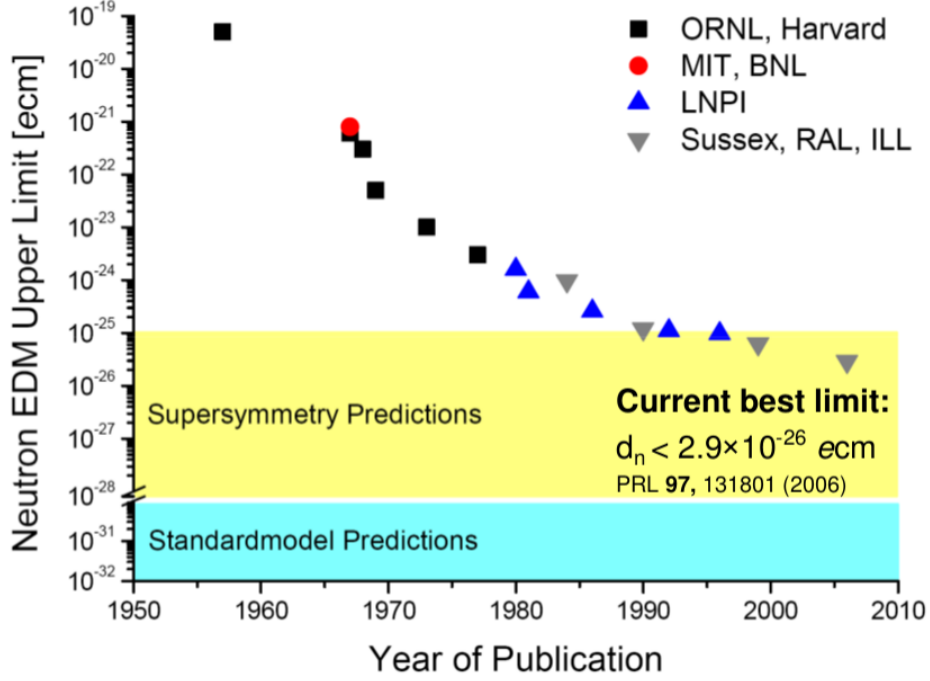


Figure 4: experimental nEDM upper limits [4]

Where γ is the gyromagnetic ratio (it is a negative value for the neutron). An oscillating B_1 field perpendicular to the B_0 field will act at a frequency ω for a time τ . this oscillating field's intensity must satisfy the equation:

$$\frac{\pi}{2} = \frac{-\gamma B_1 \tau}{2} \quad (3)$$

This will cause a $\pi/2$ spin flip if $\omega_0 = \omega$. After this, the neutron will precess freely for a time T . if $\omega_0 = \omega$, when a second $\pi/2$ pulse is applied, the neutron should theoretically always change state. However, if $\omega_0 \neq \omega$, the free precession T allows the neutron to change phase with respect to the B_1 field. This means that when the second $\pi/2$ pulse is applied, depending on $\lambda = \omega_0 - \omega$, the neutron may change states, or it may return to its original state. This can be modelled by the following equation, which defines the probability of a neutron changing from state p to state q , where state p is in the direction of B_0 , and state q is anti-parallel to B_0 :

$$P_{p,q} = 4 \sin^2 \theta \sin^2 \frac{1}{2} a \tau [\cos \frac{1}{2} \lambda T \cos \frac{1}{2} a \tau - \cos \theta \sin \frac{1}{2} \lambda T \sin \frac{1}{2} a \tau]^2 \quad (4)$$

where:

$$\Omega = -\gamma B_1 = \omega_0 \frac{B_1}{B_0} \quad (5)$$

$$a = [\lambda^2 + \Omega^2]^{\frac{1}{2}} \quad (6)$$

$$\sin \theta = \frac{\Omega}{a} \quad (7)$$

$$\cos \theta = \frac{\lambda}{a} \quad (8)$$

Note that Equation 4 is intended for a B_1 field that rotates in a circle. It can be modified for a B_1 field oscillating in one direction by dividing B_1 in Equation 5 by 2. Further analysis on this can be found in subsubsection 2.5.1.

The measurement of the nEDM is done by adding an electric field parallel to B_0 ($E \uparrow$) in one set of cycles, and anti-parallel to B_0 ($E \downarrow$) in another set of cycles. four different frequencies are applied for both $E \uparrow$ and $E \downarrow$, two on either side of the central fringe. The Larmor frequency calculated from these points is denoted as $\omega \uparrow$ for $E \uparrow$, and $\omega \downarrow$ for $E \downarrow$. The difference between these two values is then:

$$\Delta\omega = \omega \uparrow - \omega \downarrow \quad (9)$$

and the nEDM value will equal:

$$d_n = \frac{h\Delta\omega}{4E} \quad (10)$$

In reality, there are many systematic and statistical errors associated with the nEDM experiment, which makes it difficult to measure the nEDM. The statistical error of the experiment is:

$$\sigma = \frac{h}{2ET\alpha\sqrt{N}} \quad (11)$$

Where N is the sample size (number of neutrons), E is the electric field strength in (V/m), and α is the visibility of the central fringe:

$$\alpha = \frac{N_{\max} - N_{\min}}{N_{\max} + N_{\min}} \quad (12)$$

where N_{max} is the number of effectively polarized UCN at the peak of the resonance curve, and N_{min} is the number of ineffectively polarised UCN at the peak of the resonance curve [6].

Although the statistical error is the dominating factor in the most accurate nEDM experiment to date [5], the systematic error is also large. A significant portion of the systematic error is due to the geometric phase effect (GPE).

There are a few ways that an experiment can reduce the false EDM due to GPE. The first way, is to reduce the field gradient to as small of a value as possible. The second way, is to have a magnetometer inside the EDM cell with the neutrons. By measuring the Larmor precession of the magnetometer atoms, their GPE can be determined, and can be correlated to the GPE of the UCNs. This way, the GPE can be accounted for and eliminated in post data analysis. TRIUMF will be using a co-magnetometer of Xenon and Mercury.

2 PENTrack

2.1 Background

PENTrack is a Monte Carlo program used to simulate protons, electrons, and neutrons. The user defines parameters such as the particle source, guide geometry and material, as well as electric and magnetic fields. The program then simulates the particle trajectories as well as other parameters, and outputs the results to a text file. This text file can then be analysed with software such as ROOT.

PENTrack is a C++ based program. For dealing with the .stl files that the user inputs (typically via SolidWorks) It includes CGAL libraries. It also includes BOOST libraries for tasks such as numerical integration.

2.2 Implementation of general B_0 and B_1 fields

Previously, PENTrack had a static B_0 field implemented [9]. However, it was limited to act in the Z direction, with its origin at (0,0,0). The field was changed so that it could be oriented in any direction, with its origin at any point. The B_1 field was then created by simply adding a time varying sine function to the B_0 field. The input parameters for the added fields are shown in Figure 5.

```

84 #EDMStaticB0GradZField  x_origin_offset(m) yoff(m) zoff(m) pol_ang azm_ang Strength(T) Gradient(T/m)
85 #EDMStaticB0GradZField                                0      0      0      0      0      1E-6      1E-9
86
87 #EDM_AC_B1Field  xoff(m) yoff(m) zoff(m) pol(deg) azm(deg) B_1(T) Grad(T/m) freq(Hz) time_on(s) time_off(s) shift(s)
88 #EDM_AC_B1Field      0      0      0      90      0      1E-6      1E-9      1      2      4      0
89
90 #EDMStaticEField      MagnitudeAlongX (V/m)      MagnitudeAlongY (V/m)      MagnitudeAlongZ (V/m)
91 #EDMStaticEField                                0      0      1E6

```

Figure 5: geometry.in file

2.2.1 Explanation of C++ code

The improvements to the B_0 field were done by means of transformation matrices. The user defines a polar angle (θ) and an azimuthal angle (ϕ). When the B field is called, it takes the position of the particle in space, and rotates it into a reference frame where the B field is acting in the positive Z direction. This is done with the following transformation:

$$\begin{bmatrix} \cos(\theta) \cos(\phi) & \cos(\theta) \sin(\phi) & -\sin(\theta) \\ -\sin(\phi) & \cos(\phi) & 0 \\ \cos(\phi) \sin(\theta) & \sin(\theta) \sin(\phi) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad (13)$$

Where $'$ coordinates denote a coordinate system with its Z axis in the direction of the B_0 field, and coordinates without a $'$ are in the coordinate system that the rest of PENTrack uses (global coordinates). The magnetic field equations are then applied, and the value of the B field rotated back into the original coordinate system.

$$\begin{bmatrix} \cos(\theta) \cos(\phi) & -\sin(\phi) & \cos(\phi) \sin(\theta) \\ \cos(\theta) \sin(\phi) & \cos(\phi) & \sin(\theta) \sin(\phi) \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} B_{x'} \\ B_{y'} \\ B_{z'} \end{bmatrix} = \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} \quad (14)$$

The code is included in Figure 6.

For the first spacial derivatives of the field, the method is similar, but slightly different. The transformation matrix has to be applied to the derivatives twice in order to obtain the correct values. We begin with three vectors

of first order partial derivatives:

$$\begin{bmatrix} \frac{\partial B_{x'}}{\partial x'} \\ \frac{\partial B_{x'}}{\partial y'} \\ \frac{\partial B_{x'}}{\partial z'} \end{bmatrix}, \begin{bmatrix} \frac{\partial B_{y'}}{\partial x'} \\ \frac{\partial B_{y'}}{\partial y'} \\ \frac{\partial B_{y'}}{\partial z'} \end{bmatrix}, \begin{bmatrix} \frac{\partial B_{z'}}{\partial x'} \\ \frac{\partial B_{z'}}{\partial y'} \\ \frac{\partial B_{z'}}{\partial z'} \end{bmatrix}$$

The first transformation achieves the following:

$$\begin{bmatrix} \cos(\theta) \cos(\phi) & -\sin(\phi) & \cos(\phi) \sin(\theta) \\ \cos(\theta) \sin(\phi) & \cos(\phi) & \sin(\theta) \sin(\phi) \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} \frac{\partial B_{x'}}{\partial x'} \\ \frac{\partial B_{x'}}{\partial y'} \\ \frac{\partial B_{x'}}{\partial z'} \end{bmatrix} = \begin{bmatrix} \frac{\partial B_{x'}}{\partial x} \\ \frac{\partial B_{x'}}{\partial y} \\ \frac{\partial B_{x'}}{\partial z} \end{bmatrix} \quad (15)$$

This is done for all three vectors. The vectors are then rearranged to form:

$$\begin{bmatrix} \frac{\partial B_{x'}}{\partial x} \\ \frac{\partial B_{y'}}{\partial x} \\ \frac{\partial B_{z'}}{\partial x} \end{bmatrix}, \begin{bmatrix} \frac{\partial B_{x'}}{\partial y} \\ \frac{\partial B_{y'}}{\partial y} \\ \frac{\partial B_{z'}}{\partial y} \end{bmatrix}, \begin{bmatrix} \frac{\partial B_{x'}}{\partial z} \\ \frac{\partial B_{y'}}{\partial z} \\ \frac{\partial B_{z'}}{\partial z} \end{bmatrix}$$

and a second transformation is applied to return to global coordinates:

$$\begin{bmatrix} \cos(\theta) \cos(\phi) & -\sin(\phi) & \cos(\phi) \sin(\theta) \\ \cos(\theta) \sin(\phi) & \cos(\phi) & \sin(\theta) \sin(\phi) \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} \frac{\partial B_{x'}}{\partial x} \\ \frac{\partial B_{y'}}{\partial x} \\ \frac{\partial B_{z'}}{\partial x} \end{bmatrix} = \begin{bmatrix} \frac{\partial B_x}{\partial x} \\ \frac{\partial B_y}{\partial x} \\ \frac{\partial B_z}{\partial x} \end{bmatrix} \quad (16)$$

For completion, the code is included in Figure 7

For the oscillating B_1 field the same class as B_0 is used, but when the object is created some extra values are set. A sine wave scaling factor is applied when the B_1 field is called, but the field is only applied during the time that it is defined as being "on". The implementation is shown in Figure 6.


```

1 #include <iostream>
2 #include <cmath>
3 #include "edmfields.h"
4
5
6 TEDMStaticB0GradZField::TEDMStaticB0GradZField(double xoff, double yoff, double zoff,
7 double angl1, double angl2, double abz, double adB0zdz, int AC, double frq, double
8 tstart1, double tend1, double pshift, double dB_0, double dB_1, double dB_2, double
9 dB_3, double dB_4, double dB_5, double dB_6, double dB_7, double dB_8): edmB0xoff(
10 xoff), edmB0yoff(yoff), edmB0zoff(zoff), pol_angl1(angl1), azm_ang2(angl2), edmB0z0(abz
11 ), edmdB0z0dz(adB0zdz), ac(AC), f(frq), onl(tstart1), off1(tend1), phase(pshift),
12 dB0(dB_0), dB1(dB_1), dB2(dB_2), dB3(dB_3), dB4(dB_4), dB5(dB_5), dB6(dB_6), dB7(
13 dB_7), dB8(dB_8) {}; //end constructor
14
15 void TEDMStaticB0GradZField::BField(double x, double y, double z, double t, double B
16 [4][4], int partnum, int jobnum){
17
18 if(ac>=1 && (t<onl || t>off1)){
19 else{
20
21     double t1[3]={x-edmB0xoff,y-edmB0yoff,z-edmB0zoff}; //point to be rotated into
22     Bfield reference frame
23     double Rot1[3][3]; //Rotation Matrix
24
25     Rot1[0][0]=cos(pol_angl1)*cos(azm_ang2);
26     Rot1[1][0]=-sin(azm_ang2);
27     Rot1[2][0]=sin(pol_angl1)*cos(azm_ang2);
28     Rot1[0][1]=cos(pol_angl1)*sin(azm_ang2);
29     Rot1[1][1]=cos(azm_ang2);
30     Rot1[2][1]=sin(pol_angl1)*sin(azm_ang2);
31     Rot1[0][2]=-sin(pol_angl1);
32     Rot1[1][2]=0;
33     Rot1[2][2]=cos(pol_angl1);
34
35     double t2[3]={0};
36     double Rot2[3][3];
37
38     for(int i=0;i<3;i++){
39         for(int j=0;j<3;j++){
40             t2[i]+=Rot1[i][j]*t1[j]; //apply rotation matrix to point in space to
41             rotate to Bfield coordinate system
42             Rot2[j][i]=Rot1[i][j];
43         }
44     }
45
46     double BF1[3]; // B field in this coordinate system to be transformed into
47     original coordinate system
48     double BF2[3]={0};
49     //Apply known field
50     BF1[0] = -t2[0]/2*edmdB0z0dz; // Bx
51     BF1[1] = -t2[1]/2*edmdB0z0dz; // By
52     BF1[2] = edmB0z0 + edmdB0z0dz*t2[2]; // Bz
53
54     for(int i=0;i<3;i++){
55         for(int j=0;j<3;j++){
56             BF2[i]+=Rot2[i][j]*BF1[j]; //rotate Bfield to global coordinates
57         }
58     }
59
60     if(ac>=1){
61         double scalar;
62         scalar=sin((f*t+phase)*2*pi);
63         // scalar=sin(((partnum/f+phase+0.1*jobnum)*t)*2*pi);
64         BF2[0]*=scalar;
65         BF2[1]*=scalar;
66         BF2[2]*=scalar;
67         dB0*=scalar;
68         dB1*=scalar;
69         dB2*=scalar;
70         dB3*=scalar;
71         dB4*=scalar;
72         dB5*=scalar;
73         dB6*=scalar;
74         dB7*=scalar;
75         dB8*=scalar;
76     }
77 }

```

```

66         B[0][0] += BF2[0]; //set B matrix to B field parameters
67         B[1][0] += BF2[1];
68         B[2][0] += BF2[2];
69
70         B[0][1] += dB0; // set these to predetermined values calculated in the BFieldChange
71         function in fields.cpp
72         B[1][1] += dB1;
73         B[2][1] += dB2;
74         B[0][2] += dB3;
75         B[1][2] += dB4;
76         B[2][2] += dB5;
77         B[0][3] += dB6;
78         B[1][3] += dB7;
79         B[2][3] += dB8;
80         // std::cout << "partnumbah: " << partnum << "\n";
81     }
82 } // end TEDMStaticB0GradZField::BField
83
84 TEDMStaticEField::TEDMStaticEField (double aexMag, double aeyMag, double aezMag): exMag(
    aexMag), eyMag(aeyMag), ezMag(aezMag) {}; //end TEDMStaticEField constructor
85
86 void TEDMStaticEField::EField (double x, double y, double z, double t, double &V, double
    Ei[3], double dEidxj[3][3]) {
87     Ei[0] += exMag;
88     Ei[1] += eyMag;
89     Ei[2] += ezMag;
90
91     //V = integrate(E*r) => V = E_z * z
92     V += Ei[2]*z;
93
94 } //end TEDMStaticEField::EField

```

Figure 6: edmfields.cpp

```

1 void TFieldManager::BFieldChange(double theta, double phi, double dBn, double dB2[9]) //
    more info on code theory in edmfields.cpp
2 {
3     dB2[0] = -dBn/2; //dBxdx
4     dB2[3] = 0; dB2[6] = 0; //dBxdy, dBxdz
5     dB2[1] = 0; dB2[7] = 0; //dBydx, dBy/dz
6     dB2[4] = -dBn/2; //dBydy
7     dB2[2] = 0; dB2[5] = 0; //dBzdx, dBzdy
8     dB2[8] = dBn; //dBzdz
9
10     double Bd[6][3] = {{ dB2[0], dB2[3], dB2[6] }, { dB2[1], dB2[4], dB2[7] }, { dB2[2], dB2[5], dB2
        [8] }, { 0, 0, 0 }, { 0, 0, 0 }, { 0, 0, 0 } };
11 // change in B field in this coordinate system to be transformed into original
    coordinate system. //Bd[0]= Bdx/d(x,y,z) //Bd[1]= Bdy/d(x,y,z) //Bd[2]= Bdz/d(x,y,z)
12
13     double Rot[3][3]; //Rotation Matrix
14     Rot[0][0] = cos(theta)*cos(phi);
15     Rot[0][1] = -sin(phi);
16     Rot[0][2] = sin(theta)*cos(phi);
17     Rot[1][0] = cos(theta)*sin(phi);
18     Rot[1][1] = cos(phi);
19     Rot[1][2] = sin(theta)*sin(phi);
20     Rot[2][0] = -sin(theta);
21     Rot[2][1] = 0;
22     Rot[2][2] = cos(theta);
23
24     for(int k=0; k<3; k++){
25         for(int i=0; i<3; i++){
26             for(int j=0; j<3; j++){
27                 Bd[k+3][i] += Rot[i][j]*Bd[k][j]; //apply rotation matrix to Change in Bfield
                    parameters
28             }
29         }
30     }
31
32     Bd[0][0] = Bd[3][0]; //rearrangements of vectors for 2nd transform of Bfield. Bdx = Bd(
        x', y', z')/dx, where z' is in the direction of the B field
33     Bd[0][1] = Bd[4][0];
34     Bd[0][2] = Bd[5][0];
35     Bd[1][0] = Bd[3][1]; //Bdy = Bd(x', y', z')/dy

```

```

36     Bd[1][1]=Bd[4][1];
37     Bd[1][2]=Bd[5][1];
38     Bd[2][0]=Bd[3][2]; //Bdz = Bd(x',y',z')/dz
39     Bd[2][1]=Bd[4][2];
40     Bd[2][2]=Bd[5][2];
41
42
43     for(int k=0;k<3;k++){
44         for(int i=0;i<3;i++){
45             Bd[k+3][i]=0;
46             for(int j=0;j<3;j++){
47                 Bd[k+3][i]+=Rot[i][j]*Bd[k][j]; //apply rotation matrix again
48             }
49         }
50     }
51
52
53     dB2[0]=Bd[3][0];
54     dB2[1]=Bd[3][1];
55     dB2[2]=Bd[3][2];
56     dB2[3]=Bd[4][0];
57     dB2[4]=Bd[4][1];
58     dB2[5]=Bd[4][2];
59     dB2[6]=Bd[5][0];
60     dB2[7]=Bd[5][1];
61     dB2[8]=Bd[5][2];
62 }

```

Figure 7: fields.cpp

2.2.2 Verification of Performance

Tests were done to verify that the accuracy and processing speed of the general fields were within reasonable limits. It was found that while the new code was slower than the original (about half the speed for a B_0 field), it was not significant enough to cause processing time issues.

To test the functionality of the B_0 field, BFieldCut simulations were run in PENTrack, and plots of the results were created in matlab. Results are shown in Figure 8 and Figure 9. To ensure that the B_1 oscillations preformed as expected, a simulation was run and the magnitude of the field over time was plotted. The results are shown in Figure 10.

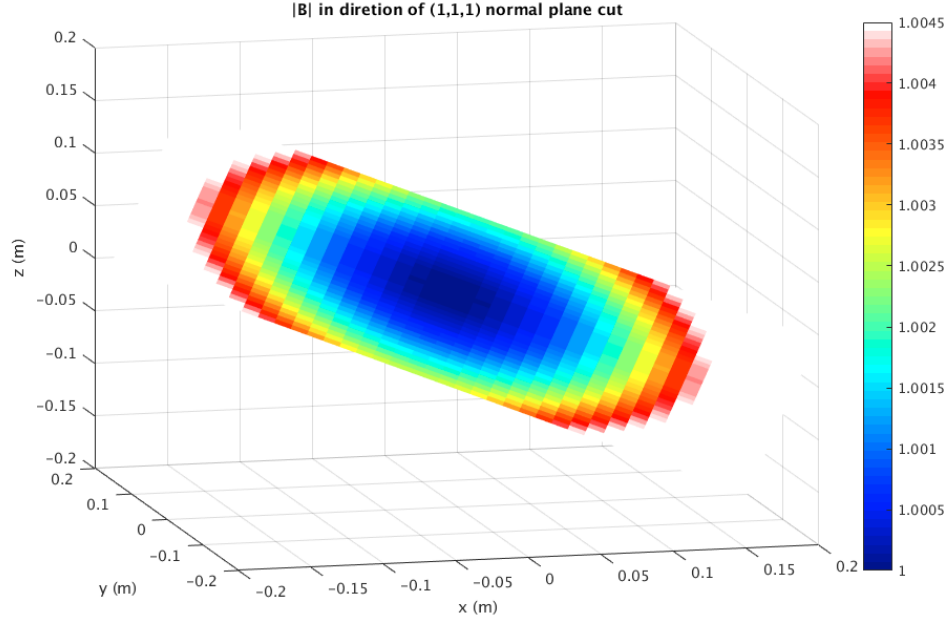


Figure 8: B_0 field verification of transformations. $B_0 = 1\mu T$ and $\frac{\partial B_0}{\partial z} = 1\frac{nT}{m}$. The B_0 field is in the (1, 1, 1) direction and centered at (0, 0, 0). Note that the colour scale has units of μT

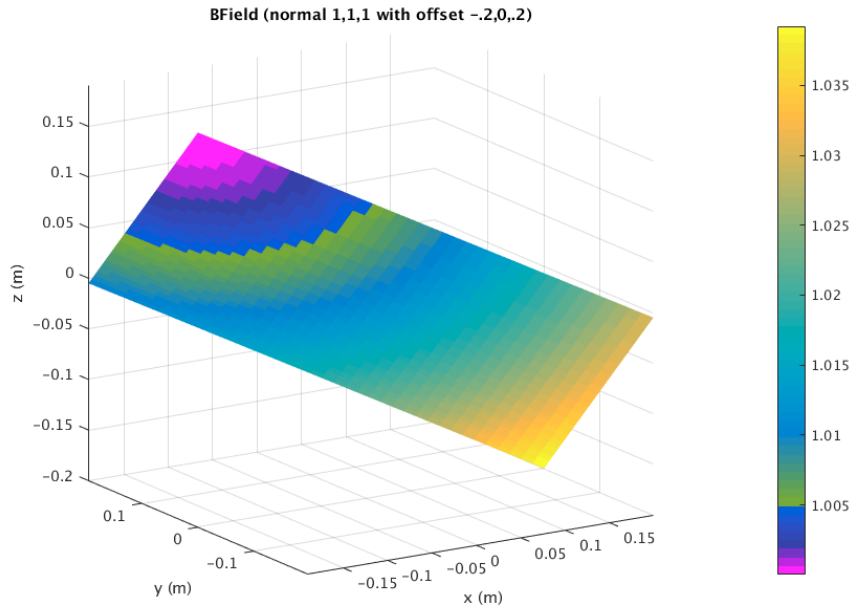


Figure 9: B_0 field verification of origin offset. $B_0 = 1\mu T$ and $\frac{\partial B_0}{\partial z} = 1\frac{nT}{m}$. The B_0 field is in the (1, 1, 1) direction and centered at (-0.2, 0, 0.2). Note that the colour scale has units of μT

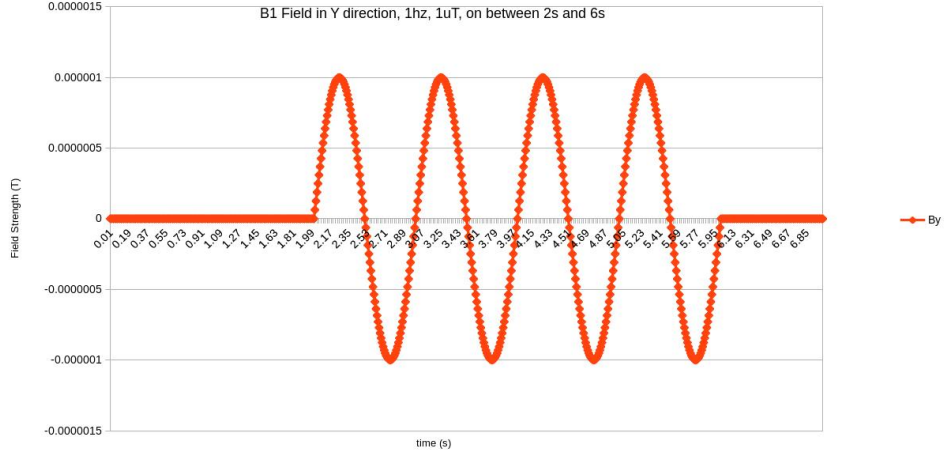


Figure 10: B_1 field verification of performance

2.3 Quartz Window Study

In order for the co-magnetometer in the EDM cell to work, the windows that the lasers pass through must be UV transparent. Unfortunately, UV grade fused silica (the proposed window material) does not have a high Fermi potential. It is possible to coat the windows with a thin layer of deuterated polyethylene (DPE), which has a high Fermi potential and is still UV transparent. However, the DPE may not be transparent enough for TRIUMF's magnetometer, which requires 99.9% or greater transparency. In order to determine how uncoated windows would effect the neutron lifetime, a study was performed.

Ruediger Picker created a program in ROOT to simulate the effect of quartz windows on the mean neutron storage time, however his program assumed a spherical volume. It was decided that Ruediger's program should be verified by running a simulation in PENTrack.

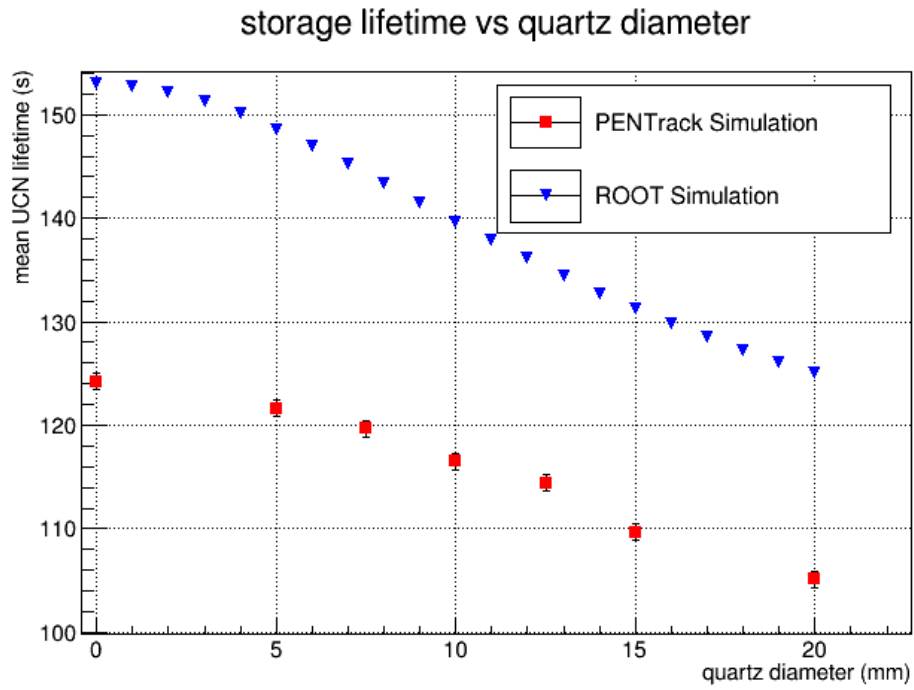


Figure 11: Storage lifetime: comparison between ROOT program and PENTrack

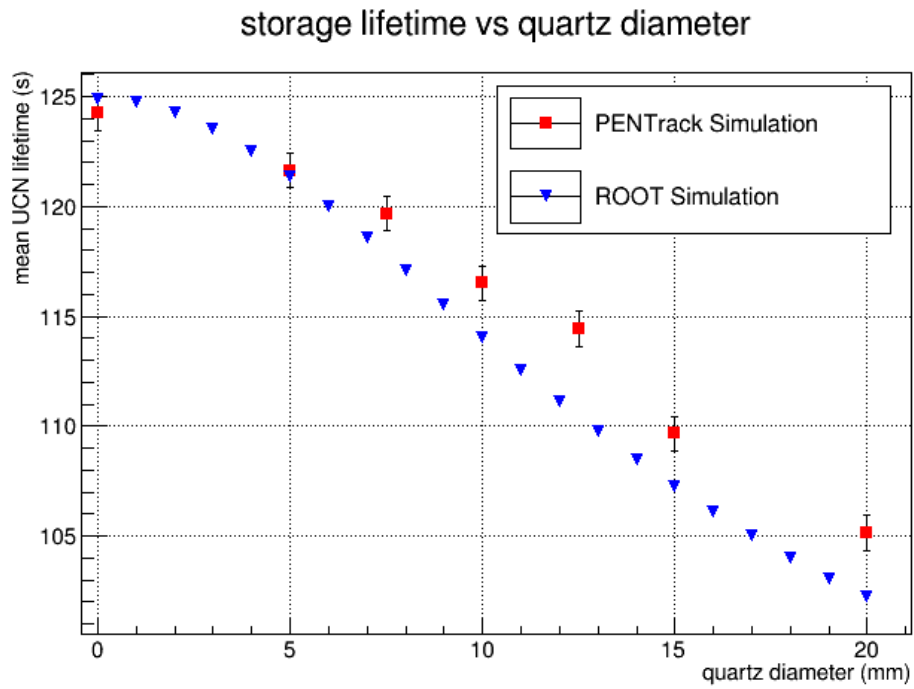


Figure 12: Storage lifetime: comparison between normalized ROOT program and PENTrack

Figure 11 and Figure 12 show the comparison between Ruediger’s ROOT program and the PENTrack simulation. although the values differ significantly in Figure 11, the difference in lifetime is similar for both models, as shown in Figure 12. the ROOT program seems to produce a slightly larger change in UCN lifetime with a change in the quartz window’s diameter.

2.4 Horizontal EDM Cell Study

The proposed nEDM apparatus at TRIUMF has two separate EDM cells. If one cell has a greater average UCN kinetic energy, it will have a larger false EDM due to Equation 21 (subsubsection 2.6.2). A difference in the false EDM may make correlating the results between the two cells difficult. This study measures the energy distribution of the neutrons in each cell, and then calculates the $d_{f,n}$ due to the GPE for each cell. This study was touched upon previously [9], however a more comprehensive study was deemed necessary.

Two different experiment geometries were evaluated in this study. In the first geometry, the two cells are stacked vertically as shown in Figure 13. In the second geometry, the cells are stacked horizontally (Figure 14). In the vertically stacked cells, there is a potential energy difference between the top cell and the bottom cell. This is expected to cause a difference in the $d_{f,n}$ between the top and bottom cells.

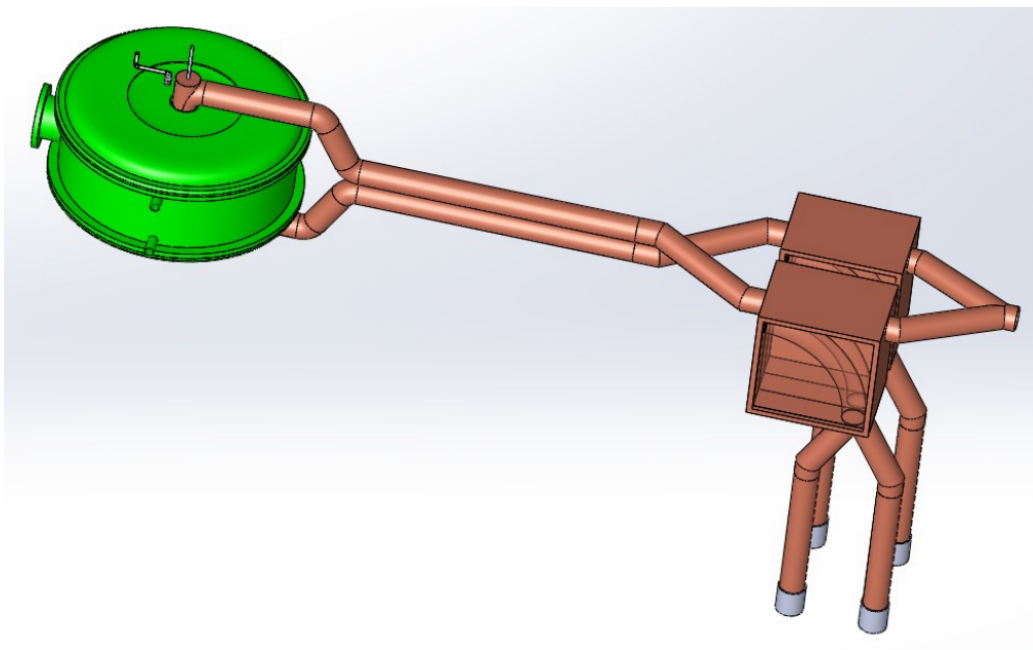


Figure 13: Vertical EDM cell geometry

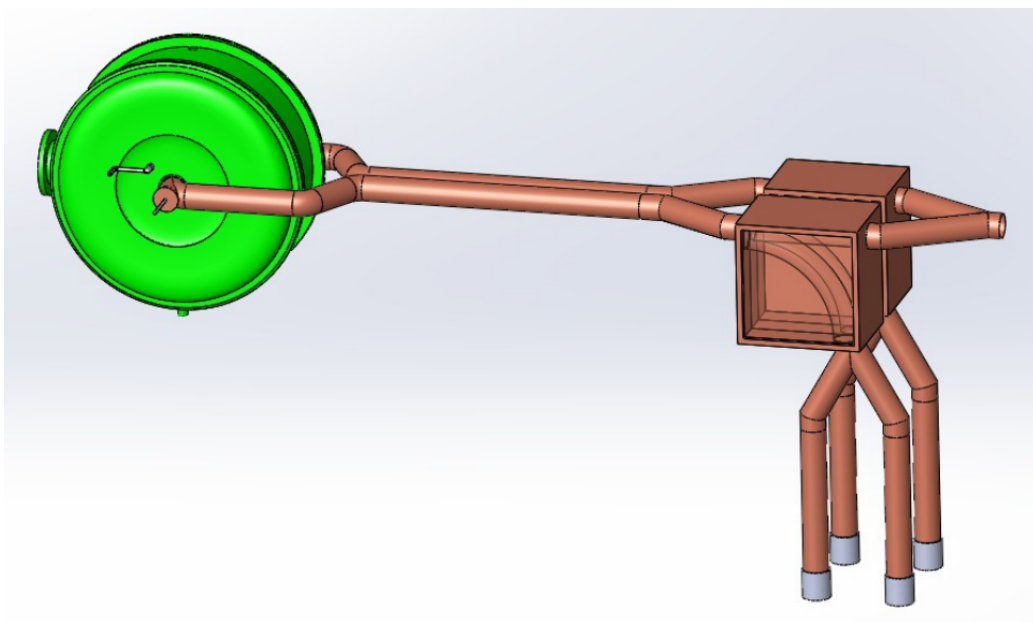


Figure 14: Horizontal EDM cell geometry

This study was broken up into two separate simulations. The first simulation followed the neutrons from the helium bottle into the EDM cells. The outputs of this simulation were the filling efficiency, and the energy distribution of the neutrons inside each cell. The filling efficiency results are shown in Figure 15 and Figure 16. It shows that the optimal filling time is approximately 11 seconds, and that the horizontal orientation has a slightly better filling efficiency than the vertical orientation.

The difference in energy distributions between cells shows the importance of this study. The total energy (potential energy plus kinetic energy) distributions of the two horizontal cells (Figure 17) have a difference that is not statistically significant. On the other hand, the difference in the total energy distributions between the two vertical cells (Figure 18) is statistically significant.

The logic behind this energy difference is more easily explained when looking at the kinetic energy distributions (Figure 19). In the top cell, the neutrons will slow down due to an increase in potential energy. The top cell should also filter out the neutrons with total energies less than the energy it takes to travel up the UCN guide. In the bottom cell, the neutrons will speed up due to a decrease in potential energy. Neutrons travelling too fast at the bottom of the UCN guide due to an increase in speed will also be lost.

The second part of this study involved using the Gaussian fits of energy distributions, and running a simulation to determine the false nEDM due to different magnetic field gradients. The Gaussian fits are not ideal, however they give a general idea of the difference in $d_{f,n}$ between cells. the $d_{f,n}$ due to the GPE with magnetic field gradients of different strengths is shown in Figure 20 and Figure 21 for the vertical and horizontal cells, respectively. In order to reduce the number of statistics, the simulation was run so that the program measures the larmor frequency of the neutrons with both $E\uparrow$ and $E\downarrow$ simultaneously. Since the $E\uparrow$ and $E\downarrow$ are measured along the exact same path, the statistics are greatly reduced.

From this study, it can be determined that if the average velocity difference between the top and bottom cells is not accounted for, it will cause an

un-accounted for false edm. However, uncompensated magnetic field inhomogeneities cause the most significant systematic error in the most accurate nEDM measurement, not the GPE [5]. There are some advantages to having the cells in the vertical orientation. One of these is that the average height of the UCNs and magnetometer atoms is different, and $\partial B_{0z}/\partial z$ can be calculated due to this difference. However, since $\partial B_{0z}/\partial z = -2\partial B_{0xy}/\partial r$, it may be possible to still determine this. There would be some complications however, due to the fact that B_{xy} also depends on Equation 18. Some analysis would need to be done to see if once can still deduce $\partial B_{0z}/\partial z$ with the addition of Equation 18.

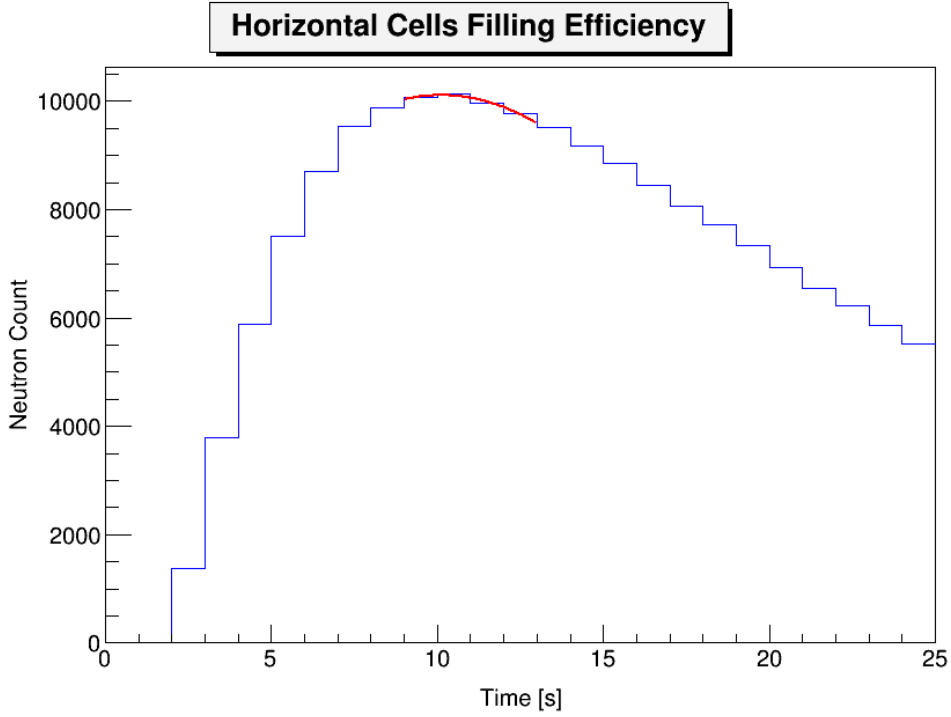


Figure 15: Horizontal EDM cell filling efficiency

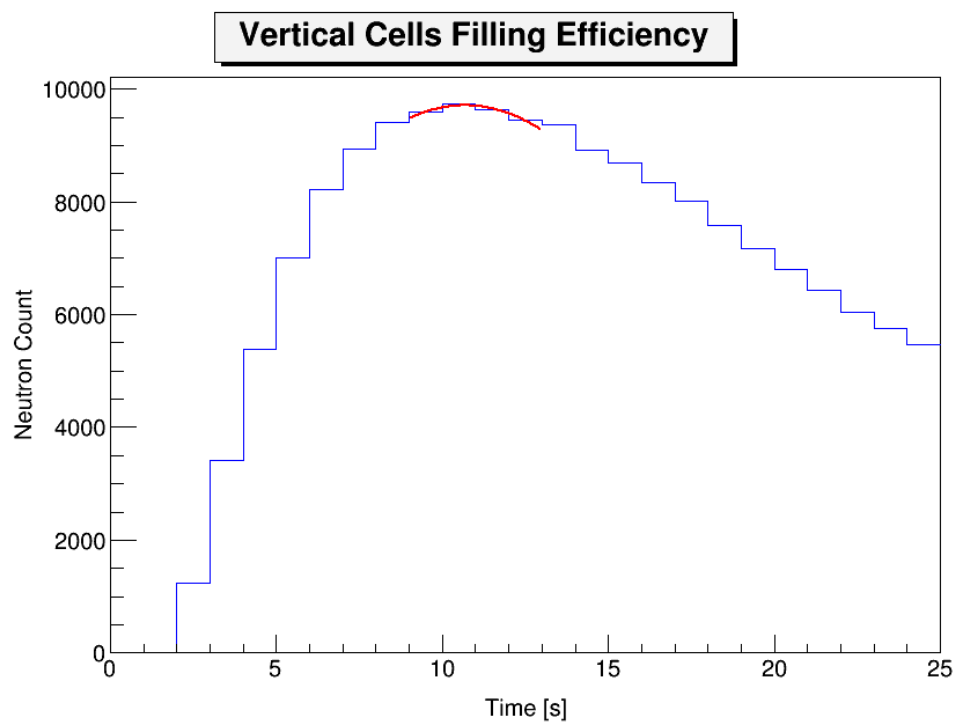


Figure 16: Vertical EDM cell filling efficiency

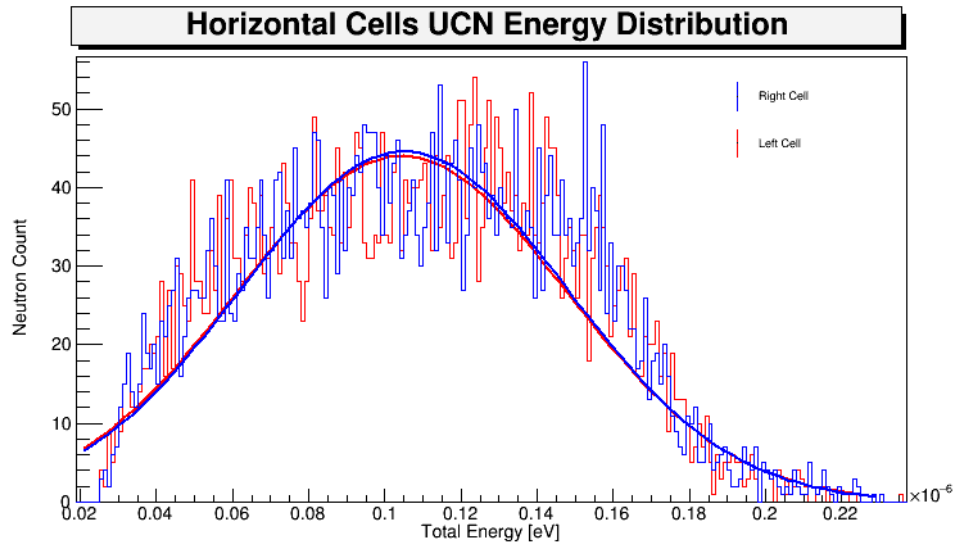


Figure 17: Horizontal EDM cell UCN total energy distribution

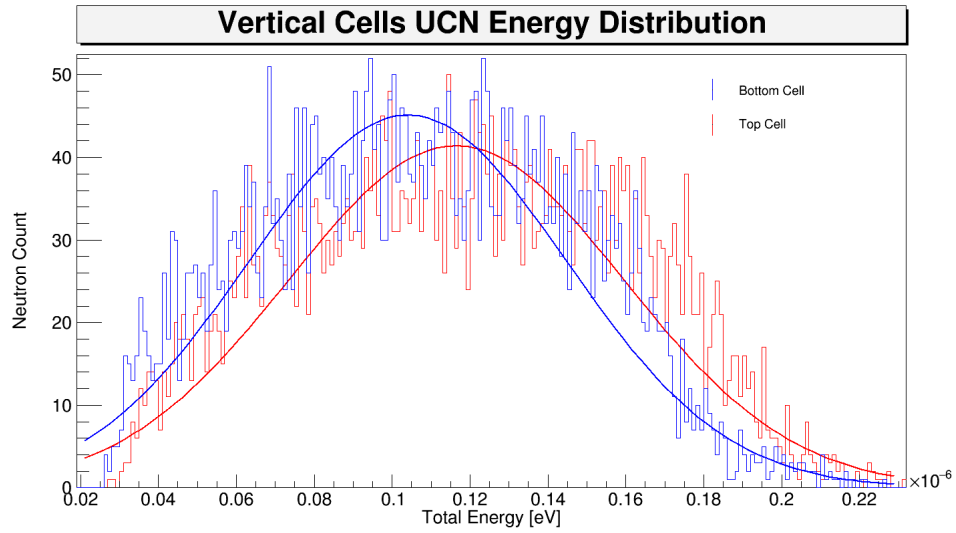


Figure 18: Vertical EDM cell UCN total energy distribution

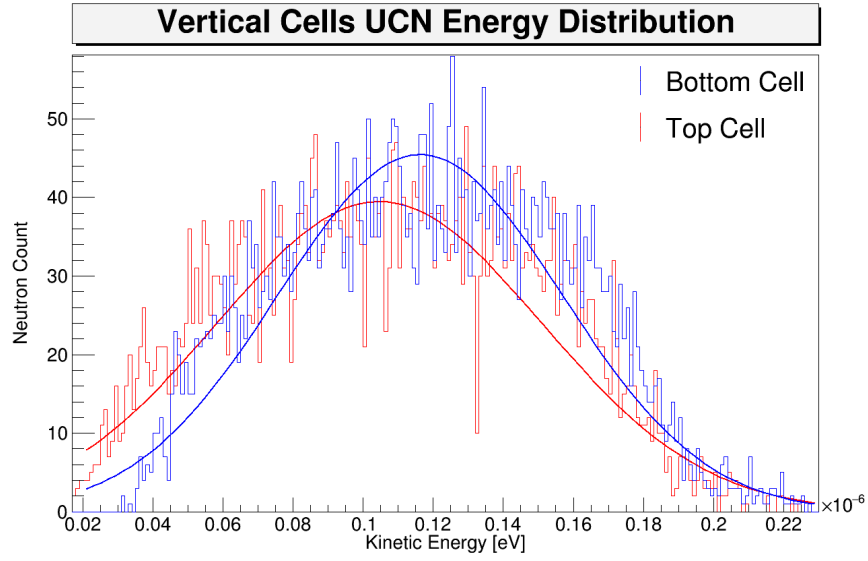


Figure 19: Vertical EDM cell UCN kinetic energy distribution

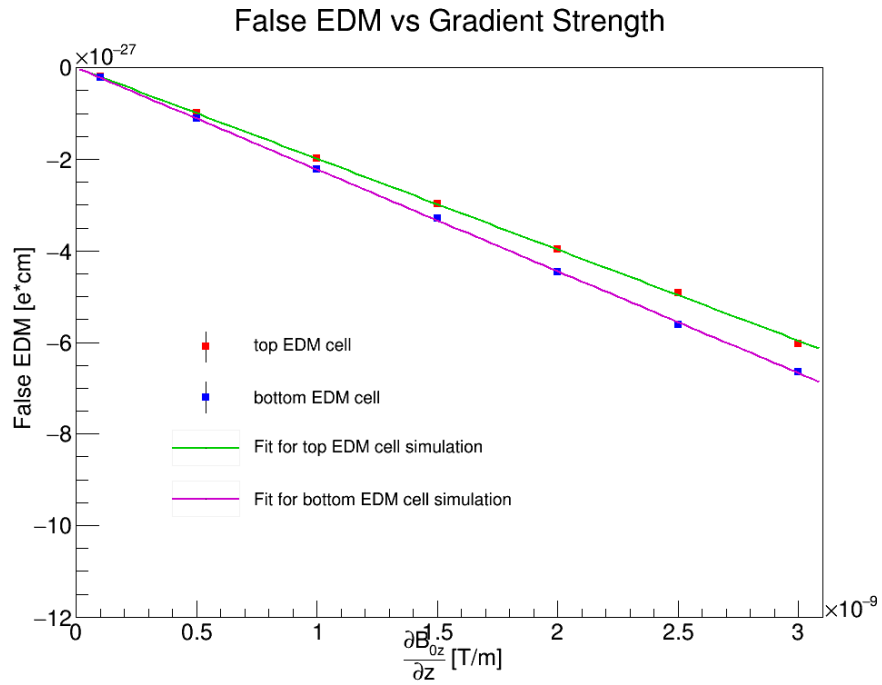


Figure 20: Vertical EDM cells, false edm comparison

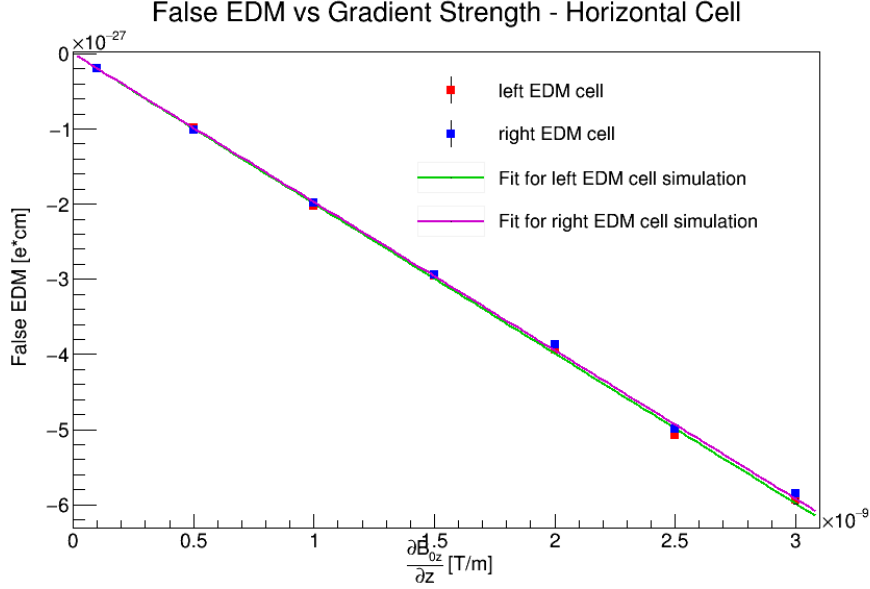


Figure 21: Horizontal EDM cells, false edm comparison

2.5 PENTrack Verification - Ramsey Cycle Simulation

The advantage to implementing the B_1 and B_0 fields in PENTrack, is that it is now possible to run a full simulation of the nEDM experiment. This opens the door to simulating the effects of various parameters on the output of the experiment.

The value of the B_0 fields used in the Ramsey cycle PENTrack simulations was based off of the anticipated field strength ($1\mu T$). The value of the B_1 field was based off of Equation 3 with a value of $\tau = 1second$. This value of τ was chosen rather arbitrarily. Previous experiments have used a τ value of around 2 seconds [5]. The electric field used was $1 \times 10^6 V/m$

2.5.1 Comparison with Theoretical Model

The first Ramsey Cycle simulations were run to compare the results from PENTrack to the theoretical model. Equation 4 is intended for a B_1 field which is rotating in a circle ($B_{1,2D}$), whereas the field being used at TRIUMF oscillates in one dimension ($B_{1,1D}$). In order for Equation 4 to be valid, TRIUMF's $B_{1,1D}$ field must have twice the magnitude of the B_1 field in

Equation 5. To verify this, a simulation was run with a $B_{1,2D}$ field, and another simulation was run with a $B_{1,1D}$ field (where $B_{1,1D}/2 = B_{1,2D}$). Both simulations were compared to Equation 4 with the same parameters. The results are shown in Figure 22 and Figure 23. Note that a spin of -0.5 in the Z direction correlates to a spin flip probability of 1.

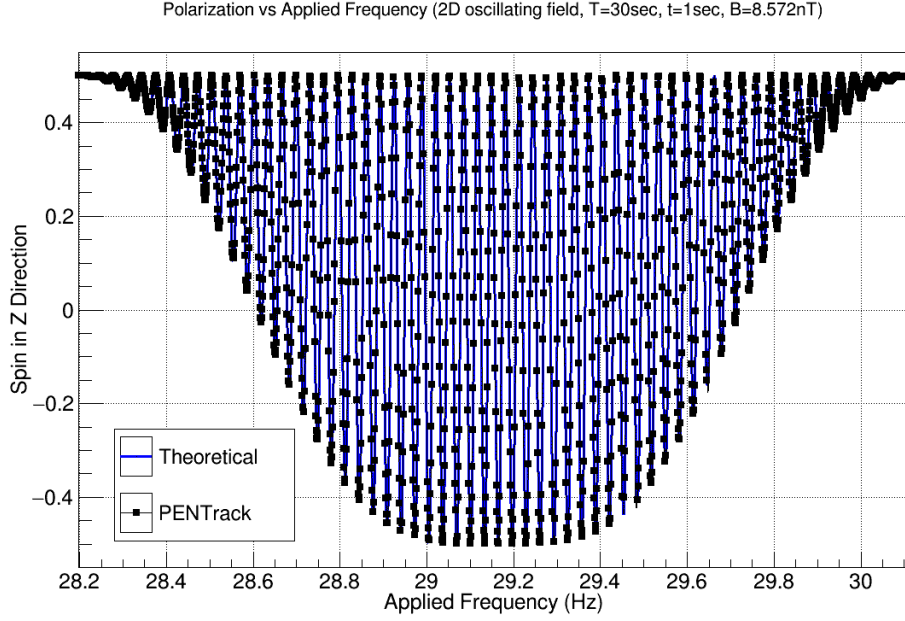


Figure 22: Ramsey cycle with circularly rotating B_1 field

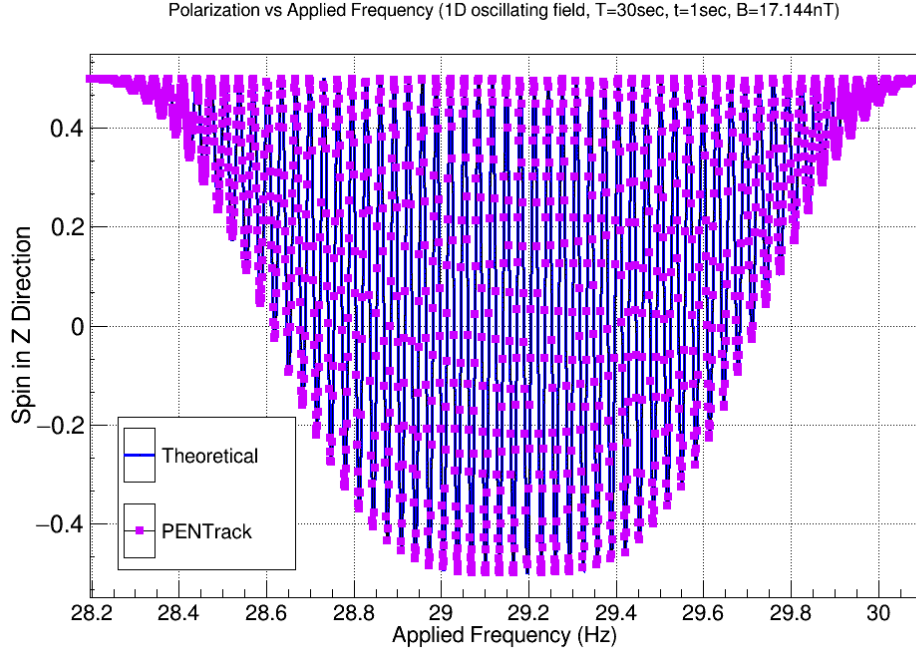


Figure 23: Ramsey cycle with B_1 field oscillating in 1D

On a macro-scale, Figure 22 and Figure 23 appear identical. There is a small shift expected (known as the Bloch-Siegert shift), which will be analysed further in the next section.

2.6 The Effect of Magnetic Field Gradients on the Ramsey Cycle

2.6.1 Background

Magnetic field inhomogeneities account for the largest source of systematic error in the most accurate nEDM experiment to date [5]. By understanding what effect these inhomogeneities will have on our experiment before it is built, TRIUMF will have a better chance of accounting for, and reducing those errors. In this study, Ramsey cycles are run with various magnetic field gradient magnitudes. The objective is to explain all of the shifts seen in the central fringe.

There are three main shifts that are expected in these simulations. The first is the Bloch-Siegert (BS) shift. This shift is due to the one dimensional B_1 field. The second shift is due to the average height of the neutrons in the cell, which will be slightly higher than the center (due to gravity). The third shift is the Ramsey-Bloch-Siegert (RBS) shift. This shift is due to the neutrons travelling in the xy plane while a field gradient is present on that plane.

For the simulations in this section, a "perfect" cylinder was created with a radius of 18.1 cm and a height of 14 cm. This was done to make mathematical analysis simple.

2.6.2 Ramsey-Bloch-Siegert Shift

If the B_0 field is oriented in the z direction (B_{0z}) and $\frac{\partial B_0}{\partial z} \neq 0$ or $E \neq 0$, an orthogonal magnetic field (B_{xy}) will be present in the rest frame of the neutron. This field will equal:

$$B_{xy} = \frac{1}{\sqrt{1 - v^2/c^2}}(B_{0xy} + B_v) \simeq (B_{0xy} + B_v) \quad (17)$$

where v is the UCN's velocity, and c is the speed of light. The B_v field is due to relativity, and is sometimes referred to as the ExV effect. It equates to:

$$B_v = \frac{E \times v}{c^2} \quad (18)$$

Assuming that the B_{0z} field has a small gradient ($\frac{\partial B_{0z}}{\partial z}$), it will also cause a small field in the xy plane

$$B_{0xy} = -\frac{\partial B_0}{\partial z} \frac{r}{2} \quad (19)$$

where r is the radial position ($r = \sqrt{x^2 + y^2}$). In calculating the value of B_{0xy} for theoretical predictions in this report, the average value of r is assumed to equal to two thirds of the radius of the EDM cell. This is done because it is the average position of the neutrons in the cell.

The B_{xy} field will shift the Larmor precession frequency away from ω_0 . The actual Larmor frequency ω_L , is given by:

$$\omega_L = \omega_0 + \frac{\omega_{xy}^2}{2(\omega_0 - \omega_r)} \quad (20)$$

Where $\omega_{xy} = -\gamma B_{xy}$, and ω_r is the rotational speed of the B_{xy} field. This is known as the Ramsey-Bloch-Siegert (RBS) shift [12].

Since Equation 18 changes sign between the $E \uparrow$ and $E \downarrow$ Ramsey cycles, the value of ω_{xy} will change. This creates a difference in the value of ω_L between the $E \uparrow$ and $E \downarrow$ cycles, which is known as the geometric phase effect (GPE).

In the case of UCN ($|\omega_r| < |\omega_0|$) the false EDM ($d_{f,n}$) due to the GPE, can be approximated to [12]:

$$d_{f,n} = -\frac{j\hbar}{2} \left(\frac{\frac{\partial B_{0z}}{\partial z}}{B_{0z}^2} \right) \frac{v_{xy}^2}{c^2} \left[1 - \frac{\omega_r^{*2}}{\omega_0^2} \right]^{-1} \quad (21)$$

$$\omega_r^{*2} = \frac{\pi^2}{6} \left(\frac{v_{xy}}{R} \right)^2 \quad (22)$$

where R is the radius of the storage cell, j is the total angular momentum quantum number, \hbar is reduced Planck's constant, and v_{xy} is the velocity in the xy plane. Note that the approximation to ω_r^{*2} is valid if there is partly diffuse reflection. More information on various models of the GPE can be found in [12]

2.6.3 Bloch-Siegert Shift

The Bloch-Siegert (BS) shift is a shift due to the non-circular B1 Field (1D instead of 2D) [6]. It is equal to:

$$\omega_L \simeq \omega_0 + \frac{\omega_1 \tan(\frac{\omega_1 \tau}{4})}{4\omega_0 T (1 + \frac{8}{\omega_1 T} \tan(\frac{\omega_1 \tau}{4}))} \quad (23)$$

where:

$$\omega_1 = -\gamma B_1 \quad (24)$$

Due to the chosen parameters for Equation 3 ($\tau = 1\text{second}$), Equation 23 reduces to:

$$\omega_L \simeq \omega_0 + \frac{\pi}{4\omega_0 T(1 + \frac{8}{\pi T})} \quad (25)$$

2.6.4 Shift due to Gravity

The third shift in the Larmor frequency is due to the fact that the average height of the neutrons will be significantly shifted from the center of the EDM cell due to gravity. The equation that defines this shift is based off of the classical kinematics equations. The average field in the z direction equals:

$$\bar{B}_z = B_{0z} + \frac{\partial B_{0z}}{\partial z} \left(\frac{-v_{0\uparrow}^2}{6g} - \frac{H}{6} + \frac{v_{0\uparrow} \sqrt{v_{0\uparrow}^2 + 2gH}}{6g} \right) \quad (26)$$

where $v_{0\uparrow}$ is the velocity the neutron has at the bottom of the cell after reflection, g is gravity (-9.81m/s^2), and H is the height of the cell.

Since the neutrons see a different B_z field strength than in Equation 2, the Larmor frequency should shift to:

$$\omega_L = \omega_0 - \gamma(\bar{B}_z - B_{0z}) \quad (27)$$

2.6.5 Function Fitting

To determine the peak of the central fringe, the real experiment uses the following equation (or a slight variation of it) [6]:

$$N_{\omega_i} = N_{\min} + (N_{\max} - N_{\min}) \cos^2\left(\frac{\pi}{2\Gamma}(\omega_i - \omega_L)\right) \quad (28)$$

Where N_{\max} and N_{\min} are the same as in Equation 12, ω_i is the frequency of the B_1 field, and Γ is:

$$\Gamma \simeq \frac{\pi}{T + \frac{4\tau}{\pi}} \quad (29)$$

Since there are four unknowns in Equation 28, four different frequencies are chosen for ω_i . The values of those frequencies are determined by:

$$\omega_i = \omega_L \pm \Gamma(1 \pm S) \quad (30)$$

Where S is known as the separation, and is chosen based on α .

2.6.6 Simulation of Various Magnetic Field Gradients

Multiple Ramsey cycle simulations were run with different field gradient strengths. For all of these simulations, the electric field magnitude was $1 \times 10^6 \text{V/m}$. The results are shown in Figure 24 and Table 1.

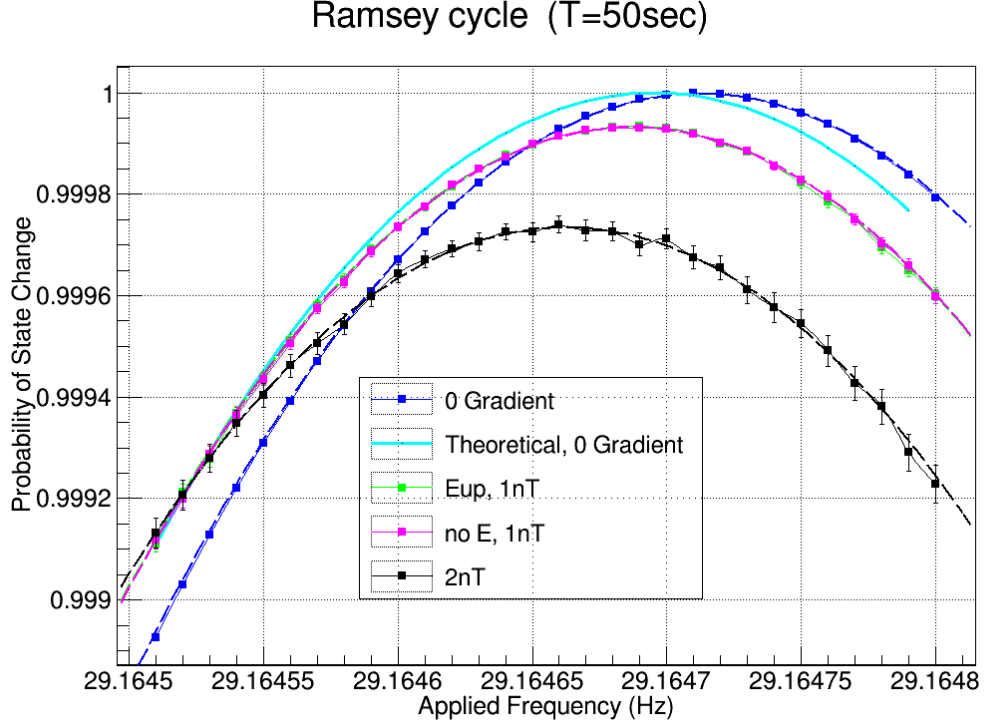


Figure 24: Ramsey cycle with and without magnetic gradients (gradients are in nT/m , in the z direction)

The difference between ω_L from Equation 2 and PENTrack for 0 gradient ($1.696 \times 10^{-5} \text{Hz}$) should be explainable from the BS shift (Equation 25) and the RBS shift (Equation 20). As shown in Table 2, The shift due to BS effect should be $1.298 \times 10^{-5} \text{Hz}$. The shift due to the RBS effect should be $2.06 \times 10^{-8} \text{Hz}$. There is still a discrepancy of $3.96 \times 10^{-6} \text{Hz}$.

The difference between the 0 gradient and 1 nT/m gradient PENTrack simulations should be explainable as a combination of the RBS, and average

Table 1: Larmor frequency fit results from simulation with various magnetic field gradients

Gradient Strength	Larmor Frequency (Hz)	error (Hz)	error (ecm)
0 (Equation 2)	29.1646953	N/A	N/A
0 (PENTrack) ($E \uparrow$)	29.1647123	6.7e-09	6.968e-28
1nT ($E \uparrow$)	29.1646869	4.5e-07	4.653e-26
1nT (No E)	29.1646872	4.6e-07	4.725e-26
2nT ($E \uparrow$)	29.1646624	1.1e-06	1.157e-25

height due to gravity shifts. As shown in Table 2, The shift due to Equation 27 should be $8.028 \times 10^{-5}\text{Hz}$. The shift due to RBS effect should be $7.713 \times 10^{-8}\text{Hz}$. PENTrack shows a shift of $-2.539 \times 10^{-5}\text{Hz}$. Based on the equations here, the shift should be in the opposite direction, and be larger.

Table 2: Magnetic gradient larmor frequency shifts. Comparison with theoretical equations

Simulations to compare ω_L offset:	Theoretical and PENTrack ($\frac{\partial B_{0z}}{\partial z} = 0$)	$\frac{\partial B_{0z}}{\partial z} = 0$ and $\frac{\partial B_{0z}}{\partial z} = 1nT/m$ (both from PENTrack)
Expected shift due to RBS	$2.06 \times 10^{-8}\text{Hz}$ ($2.1 \times 10^{-27}\text{ecm}$)	$7.713 \times 10^{-8}\text{Hz}$ ($8.0 \times 10^{-27}\text{ecm}$)
Expected shift due to BS	$1.298 \times 10^{-5}\text{Hz}$ ($1.34 \times 10^{-24}\text{ecm}$)	0
Expected shift due to gravity	0	$8.028 \times 10^{-5}\text{Hz}$ ($8.30 \times 10^{-24}\text{ecm}$)
ω_L offset (from simulation)	$1.696 \times 10^{-5}\text{Hz}$ ($1.75 \times 10^{-24}\text{ecm}$)	$-2.539 \times 10^{-5}\text{Hz}$ ($2.63 \times 10^{-24}\text{ecm}$)
difference between simulated ω_L offset and theory	$3.96 \times 10^{-6}\text{Hz}$ ($4.09 \times 10^{-25}\text{ecm}$)	$1.058 \times 10^{-4}\text{Hz}$ ($1.09 \times 10^{-23}\text{ecm}$)

The reason for a decrease in ω_L due to an increase in $\frac{\partial B_{0z}}{\partial z}$ cannot be explained based on the shifts presented here, and so alternative reasons need to be analysed. One reason could be that when the B_1 field was turned on, there could have been a large step in the magnetic field the neutrons saw.

This is because the B_1 phase was not optimized at all. Unfortunately, due to time constraints further analysis could not be performed before this report was finished. James Marcellin was tasked with further analysis on this study.

Further analysis on this subject can include many different simulations. Running simulations with positive and negative field gradients with and without a B_1 field would be a good start. Isolating for different shifts would also be useful. This could include Running simulations with a 2D oscillating B_1 field, and simulations without gravity. It could also include putting different envelopes on the B_1 field, and optimizing the phase (using values of T and τ so $T = n/(2\omega_L)$ and $\tau = m/(2\omega_L)$, where n and m are positive integers would result in possibly the best optimization).

2.6.7 Simulation of the GPE

The second study for analysis on magnetic field gradients was done to analyse the GPE. to choose the B_1 frequencies for this simulation, Equation 30 was used with an S value of 0.05 (can be small because the value of α is assumed to be 1).

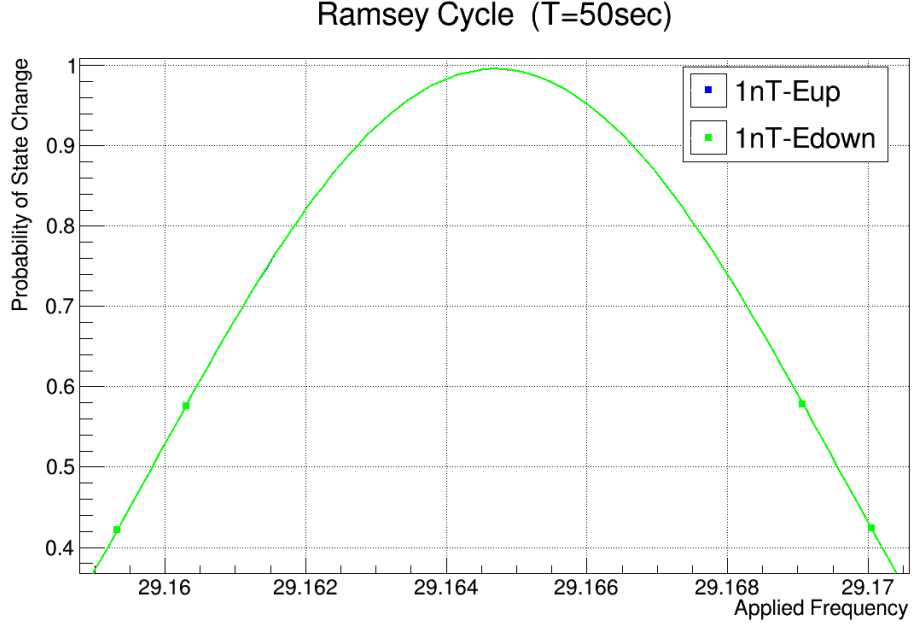


Figure 25: Ramsey cycles with $\frac{\partial B_{0z}}{\partial z} = 1nT/m$, $E \uparrow$ compared to $E \downarrow$ (Neutrons per point $\approx 1,000,000$)

Table 3: Larmor frequency fit results from simulation looking at GPE (fits use Equation 28)

Simulation	$1nT/m$ $E \uparrow$	$1nT/m$ $E \downarrow$	$\omega_r \uparrow - \omega_r \downarrow$
Larmor Frequency (Hz)	29.16469342165	29.16469342050	1.15e-9 (1.19e-28 ecm)
LF error (Hz)	2.56e-08	2.69e-08	3.71e-8
LF error (ecm)	2.65e-27	2.78e-27	3.84e-27
N_{\max}	0.9959	0.9959	N/A
N_{\max} error	2.6e-05	2.8e-05	N/A
N_{\min}	0.0041	0.0042	N/A
N_{\min} error	2.6e-05	2.7e-05	N/A
$\frac{\pi}{2\Gamma}$	25.63661977	25.63661977	N/A
$\frac{\pi}{2\Gamma}$ error	0	0	N/A

From Table 3, It can be seen that $\omega_r \uparrow -\omega_r \downarrow = 1.19 \times 10^{-28} \pm 3.84 \times 10^{-27}$ ecm. The GPE was expected to cause a shift on the order of -2×10^{-27} ecm. After looking at each of the 4 points individually, the one with the highest applied frequency (farthest right in Figure 25) does not appear to agree with the other 3 points (it is shifted in the opposite direction). By filtering this point out, and removing Γ from the fitting function, the GPE was calculated to be $2.74 \times 10^{-27} \pm 5.59 \times 10^{-27}$ ecm.

In both of these fits, the error is still too large, although it is close to the size of the false EDM expected due to the GPE. The sign change here may be accounted for by the fact that in PENTrack, the Equation 18 is implemented as $B_v = \frac{v \times E}{c^2}$. This would create a sign flip.

2.7 Future PENTrack Work

The following is a list of things that can and should be done in the future. They are in order of what is recommended to be done chronologically. There are many other studies that can be done. Discuss with the nEDM team at TRIUMF before starting work on one of these. If a future student or employee has any questions, they may contact the author of this report (Nicholas Christopher).

1. More studies of magnetic field gradient influence

The results from subsection 2.6.6 do not match the theory. More analysis should be done to explain the shifts in Larmor frequency in the PENTrack simulation. It would be advisable to run multiple simulations, each isolating the effects of one shift.

To look at the BS shift, a simulation should be run with a circular B_1 field with no gradient. This can be compared to the results from subsection 2.6.6.

A good test would be to run some simulations with only a B_0 field (no B_1 field and no E field), and plotting the Larmor frequency vs field gradient strength (for both positive and negative field gradients). If the velocity distribution is the same as in subsection 2.6.6, this should show that the Larmor frequency is directly proportional to the gradient strength. This is because the only shift should be due to subsection 2.6.4.

Optimizing the phase angle of the B_1 field would also be useful, to see if this is what is causing issues. With a square envelope, the best way of doing this would be to find a time T such that $\omega_L T \approx n/2$, and choosing a B_1 magnitude and applied time t such that $\omega_L t \approx m/2$ (where m and n are integers). Also, it should be made sure that the phase offset is such that the B_1 field begins and ends with its magnitude at 0 when $\omega = \omega_L$. There will still be a small field present at the start and end (since ω is not constant), but this will minimise it. Another thing that can be done is to implement a triangle envelope.

The GPE simulation should also be re-done. It may be easier to use a larger gradient (5nT/m or 10nT/m) since this would create a larger false EDM. However, this may also increase the error. It would be best to do this last, since there is an unidentified issue which is causing discrepancies between the simulation and theory.

2. Slightly misaligned field

Another study that can be done is to look at the effect of having a slightly misaligned B_0 and B_1 field. Plotting the Larmor frequency vs misalignment angle would be interesting.

3. Optimization of statistics

It is important to optimize the parameters in Equation 11 in order to obtain the lowest statistical error possible. Since increasing T will decrease N , these must be optimized based on the EDM cell conditions. it is unclear if α will be significantly effected by increasing T , however I would assume that it would decrease due to background neutrons and noise in the detector.

4. Addition of the nEDM to PENTrack

At some point, it would be useful to implement the EDM into the PENTrack code and try to observe it.

3 Kicker Control

3.1 Background

as discussed in subsection 1.1, a portion of the proton beam must be "kicked" from BL1A into BL1U. This is done with a kicker magnet. The microstructure of the beam is such that it arrives in pulses a period of approximately 1ms long, with a 50-100 μ s period of no beam, as shown in Figure 26 [1]. The difficulty in kicking the beam is that the kicker magnet must be ramped up or down during the no beam period. A power supply was built by Danfysik specifically for this application, and can ramp up the kicker magnet to 200A within 50 μ s.

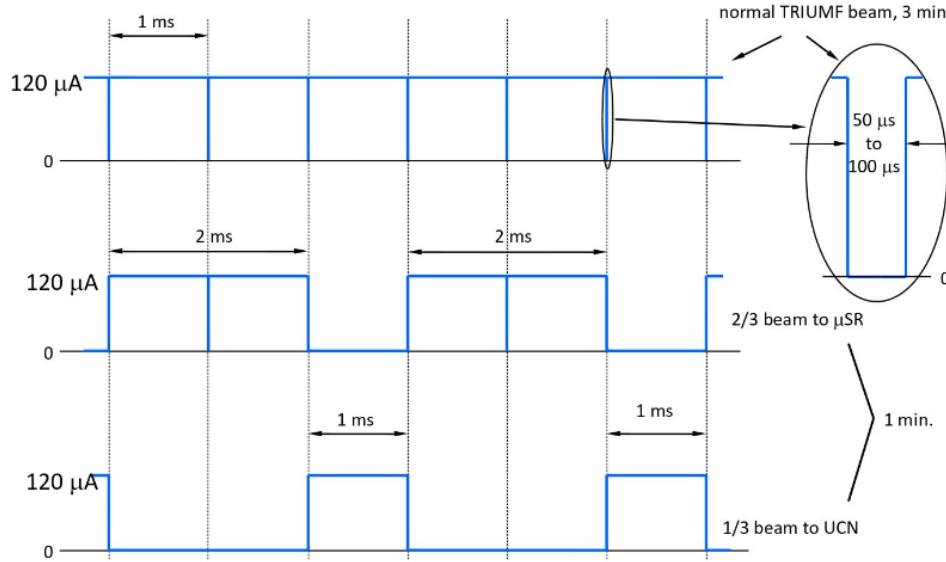


Figure 26: Beam Structure. The beam will be separated into 2 beams with the kicker magnet. 1/3 of the beam will go to the UCN source for 1 minute, followed by 3 minutes where the UCN source gets no beam [1]

Determining when to ramp up the kicker magnet is not as trivial a task as it may seem. There must be some way of knowing when the beam is off, and an optical trigger signal must be sent to the kicker power supply at the beginning of this period. An FPGA is the best choice for sending this

signal, since it operates fast enough to make decisions in under a μs . the magnet should also be shut off for various safety issues, so this should also be implemented within the electronics.

There is a capacitive pickoff monitor (1VM4) located close to the kicker magnet. Since the proton beam is travelling at three quarters the speed of light, the signal from this monitor tells us the beam intensity at almost exactly the time it is inside the kicker magnet. This monitor outputs an AM signal at the natural frequency of the cyclotron, which is approximately 23Mhz. This signal is fed through an amplifier, which uses the second harmonic (46Mhz). The envelope of this signal gives the beam structure. This monitor can detect how much residual beam there is during the beam off periods. If the residual beam is too high, the kicker magnet should not kick. If it does kick when the beam is too strong, there will be some beam spillage onto the septum between the two beamlines, and components will become radioactive.

The issue with using the 1VM4 signal to generate the trigger that ramps up and down the kicker magnet is that it is an analog signal. There is a digital signal of the beam structure (referred to as the pulser signal from here on), but it is from before the H^- ions enter the cyclotron. Therefore, There is a large offset in time between this signal and when the protons reach the kicker magnet (roughly $330\mu\text{s}$). As long as this time offset does not change much, the pulser signal can be used to create the trigger signal. However, it is known that in certain operating modes of the cyclotron this offset value does shift (when the RF booster is on).

In normal operation, BL1A has $120\mu\text{A}$ of current. The UCN source was designed for a maximum of $40\mu\text{A}$ on target. This means that the kicker should kick every third beam pulse towards BL1U. There is expected to be 1 minute of beam on target, followed by 3 minutes to run the experiment (Figure 26). Both of these times may need to be changed, so these parameters should be easily adjustable.

3.2 Commissioning the Kicker

In February 2016, the kicker power supply and magnet were commissioned. This involved operating the kicker power supply, and using an oscilloscope to look at the current being sent to and from the magnet. Everything functioned as expected. Information on controlling the kicker power supply can be found at https://ucn.triumf.ca/triumf/magnets/kicker/Kicker_PS_SOP_R2.pdf/view. Figure 27 and Figure 28 show that from 2% to 98% of maximum current, the rise and fall times are $47.2\mu s$ and $44\mu s$ respectively. If you time from when the trigger occurs, the rise and fall times are approximately $52\mu s$. This delay must be considered when sending the trigger signal in order to kick at the correct time.

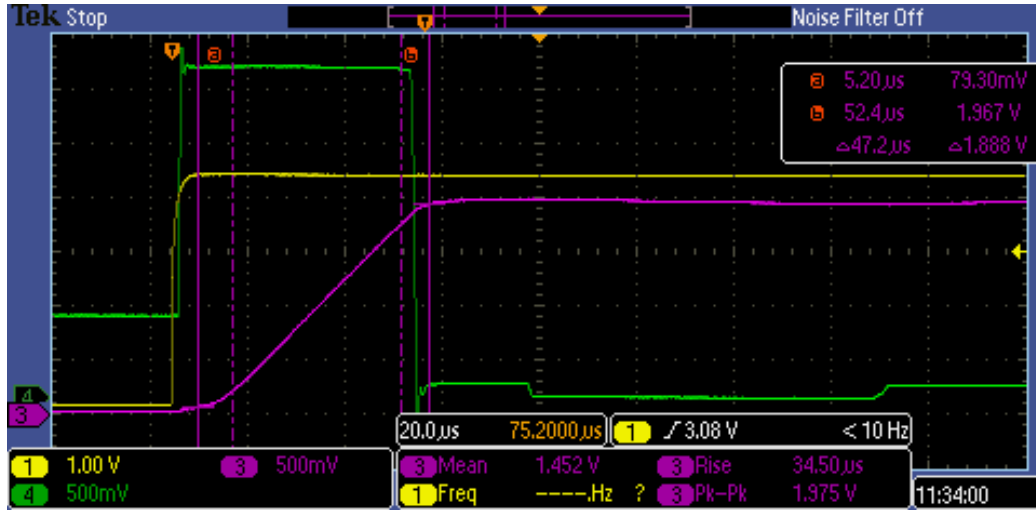


Figure 27: Kicker rise time at 200A. Yellow is the trigger, and purple is the magnet current. a and b are set to 2% and 98%

3.3 Demodulation

There were three different methods considered for demodulating the AM signal from the capacitive pickoff monitor. Two methods considered in analog were IQ demodulation, and precision rectification. Converting the signal to digital and doing the demodulation in an FPGA was also considered. Each

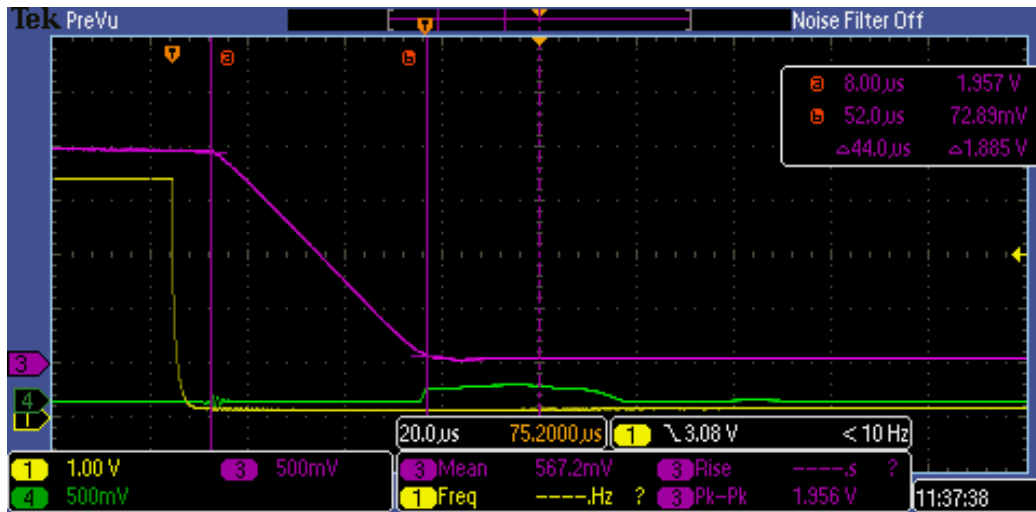


Figure 28: Kicker fall time at 200A. Yellow is the trigger, and purple is the magnet current. a and b are set to 98% and 2%

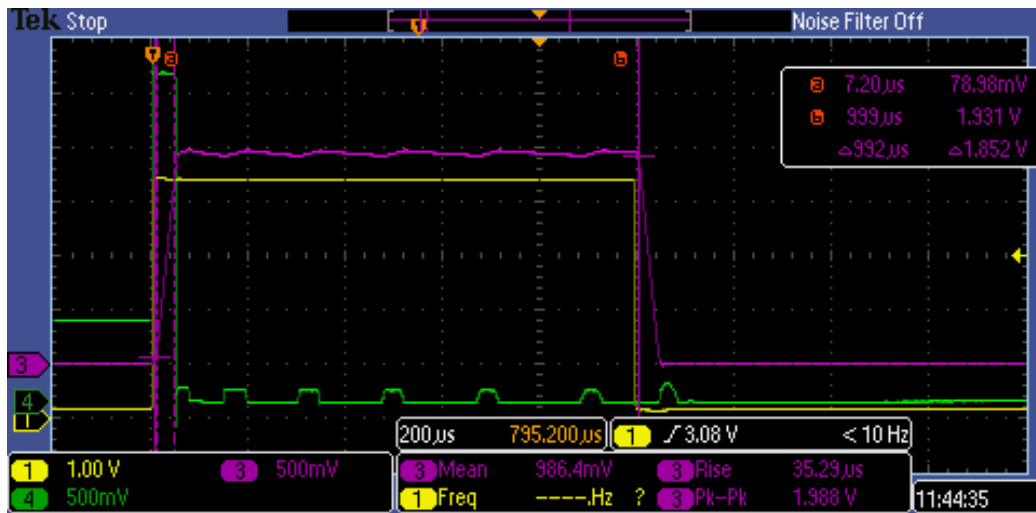


Figure 29: Kicker cycle at 200A. Yellow is the trigger, purple is the magnet current, and green is the HV capacitor voltage.

method had its advantages and disadvantages, which will be discussed in this section.

3.3.1 IQ Demodulation

The most promising demodulation method for this application is IQ demodulation. It takes very little effort to set up, and has a good accuracy. It is also not very expensive. The idea behind IQ demodulation is that you mix the incoming signal with two different signals, which are identical except that they are 90° out of phase. The output is two signals, I and Q . Each signal gives you the envelope with reference to the phase between the input and the mixing signal. By having two orthogonal mixing signals, no data is lost due to phase shifts. The envelope signal equals $\sqrt{I^2 + Q^2}$, which can be calculated very easily inside the FPGA.

There are IQ demodulator evaluation boards on the market. The one that the UCN group is using is the AD8333. The difficulty with IQ demodulation with the AD8333 is that a local oscillator (LO) signal must be sent to the IQ circuit, and it must be four times the frequency of the input signal. In this case, that means the LO signal must be around 184MHz. The frequency of the cyclotron may change over time, and so the LO frequency will need to change in accordance. The best way to lock onto the correct LO frequency is to take a signal of the cyclotron's natural frequency (23MHz) and multiply it by eight. However, there are only a few products on the market for this application.

The most promising frequency multiplier for this application appears to be BXMF1001 from API Technologies. However, it is roughly a thousand dollars. Another module was considered, which multiplies a square wave clock signal (Si5356-EVB), and is ninety dollars. The concern with mixing with a square wave is that it is not clear how much distortion would be present if a sine wave is mixed with a square wave. At the time of this report, a decision had not been made. More analysis should be done before coming to a conclusion. More testing with a signal generator would be useful for making a decision. For the purpose of testing the KTM, a LO signal was generated from the FPGA. This signal is not suitable for long term use because it introduced too much noise to measure the residual beam current.

The AD8333 evaluation board was tested using an input signal of 200mVpp and 30MHz. The LO signal was set at 120MHz and 340mVpp. 120MHz was the highest frequency of the available signal generator. The results are shown

in Figure 30 and Table 4. These results show that with inputs that have very little noise, the output has the accuracy needed for measuring the residual beam. More testing must be done with the signal from 1VM4, since it is expected to have more noise than this example. Note that the fall time of the IQ demodulator can be reduced by changing a capacitor on the evaluation board.

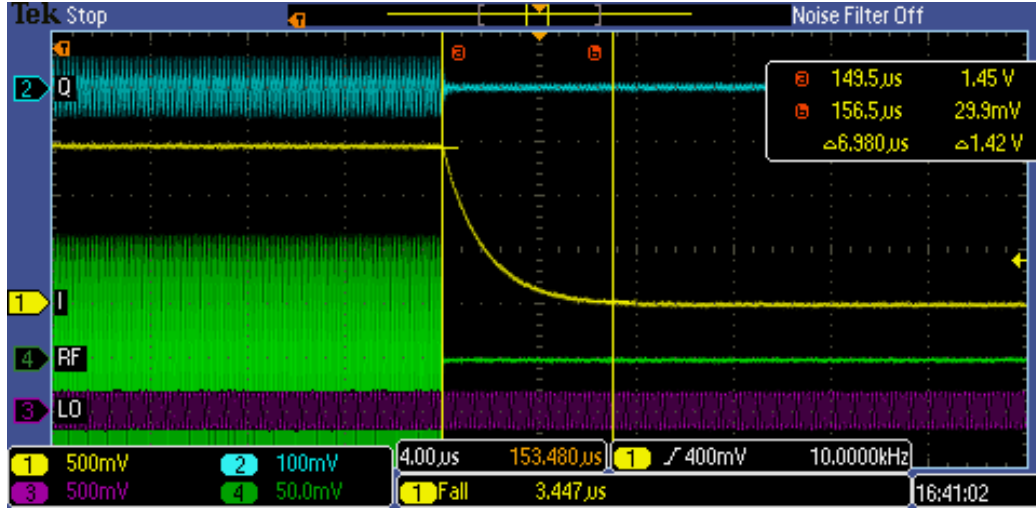


Figure 30: AD8333 performance with a square envelope input signal

3.3.2 Precision Rectifier Demodulation

Another common method for AM demodulation is using a diode as a rectifier, followed by an RC filter to produce the signal's envelope. The advantage to this is that it is the simplest AM demodulation circuit, and the most affordable. The issue with this method is that the diode's forward voltage drop is large, and the residual beam current could not be measured.

A method similar to diode rectification is using an operational amplifier to create a precision rectifier. The setback in this case is that at high frequencies they do not work. After some research on the subject, an operational amplifier (OPA698) which boasts half wave rectification at 100MHz was found. It was tested, but unfortunately it did not perform as anticipated. It performed perfectly up to about 1MHz (Figure 31). At higher frequencies, the

Table 4: AD8333 performance. Means were calculated with 2.25mv offset

signal generator duty	signal generator ratio of residual beam to full beam	I: Peak mean	I: Valley mean	demodulator output duty	demodulator output ratio of residual beam to full beam
1	0	1472.25	0	1	0
0.995	0.002506	1462.25	3.71	0.994938	0.002537
0.99	0.005025	1462.25	7.37	0.989970	0.005040
0.98	0.010101	1452.25	14.65	0.980026	0.010088
0.97	0.015228	1442.25	21.85	0.970152	0.015150
0.95	0.025641	1432.25	36.65	0.950099	0.025589
0.9	0.052632	1392.25	73.35	0.899904	0.052685
0.8	0.111111	1322.25	146.25	0.800817	0.110607
0.5	0.333333	1102.25	367.25	0.500170	0.333182

output became distorted (Figure 32). At 100MHz the signal did look usable, but that was about the only frequency above 1MHz that did not look distorted. Unfortunately, this demodulation method cannot be used for this application.

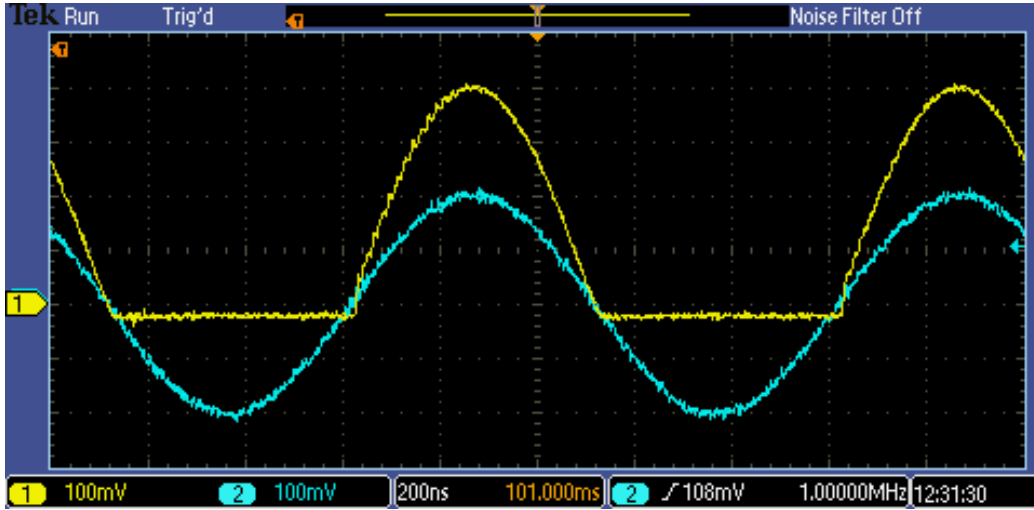


Figure 31: OPA698, 1MHz sine wave. Blue is the input signal, yellow is the output signal.

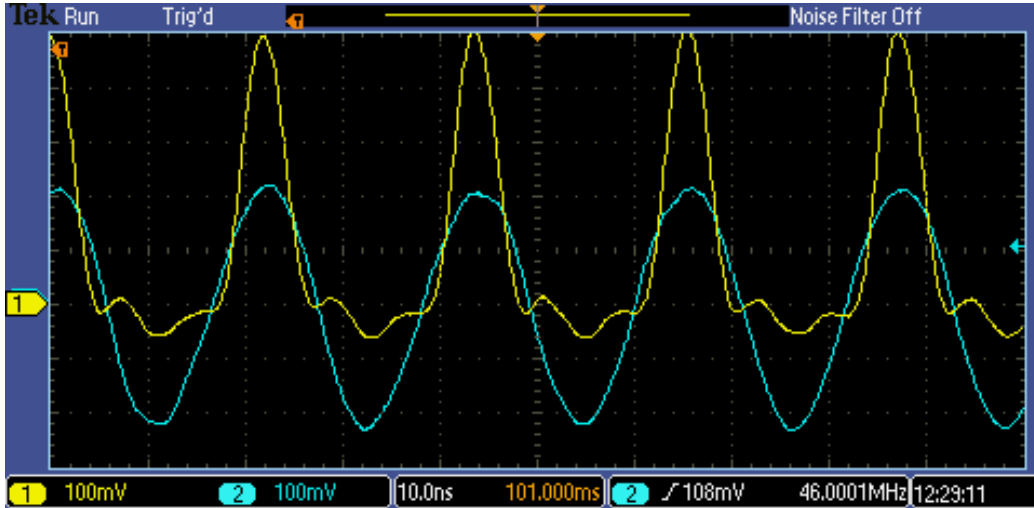


Figure 32: OPA698, 46MHz sine wave. Blue is the input signal, yellow is the output signal.

3.3.3 Digital Demodulation

Digital demodulation was considered, due to its superior accuracy with a high sample rate, high accuracy ADC (no noise would be introduced by additional electronics). The idea would be to calculate the rms value of the signal inside the FPGA. In order for this to be effective, the ADC sampling rate would need to be very high (at least 200MSPS, but may need to be up to 1GSPS). buying an ADC with a high sample rate could be for more expensive then demodulating in analog. It also requires a program to be written to do the demodulation (this may not be as simple as it appears).

It was decided that demodulation in analog using an IQ demodulator would be attempted before considering digital demodulation. If the accuracy is deemed insufficient, then switching to digital demodulation is the next best choice.

3.4 Kicker Timing Module (KTM)

The KTM takes the demodulated signal from 1VM4, and a pulser signal from the cyclotron as its inputs. The pulser signal is offset in time from the beam monitor signal. The offset is the transit time through the cyclotron.

The main purpose of the KTM is to monitor the residual beam from the 1VM4 signal while the kicker is ramping up or down. Its secondary purpose is to measure the time offset between the falling edge of the pulser signal and the falling edge of the 1VM4 signal. In the future, this module may also create the trigger signal for ramping up and down the kicker. Alternatively, it may check if the triggering system in place is triggering at the right time, and send inhibit signals for various safety issues.

The board being used for the KTM is a DE0-Nano-SoC board from Terasic. This board has an Altera Cyclone V SoC. The FPGA programming is done in Verilog using Quartus Prime. The HPS programming for communication with the FPGA is done in C. The programs can be found at <https://github.com/nchristopher04/KTCM-TRIUMF>. This section will give an overview of the programming logic and implementation.

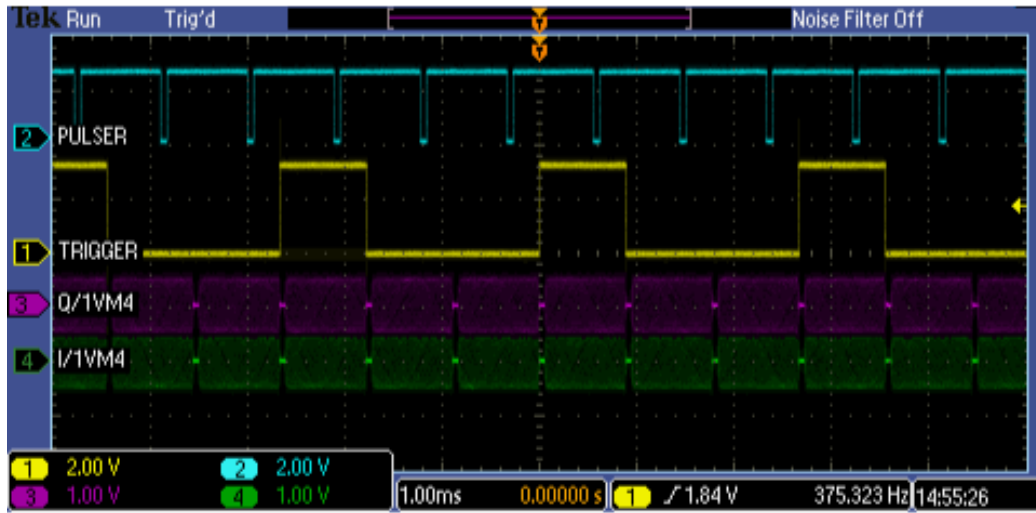


Figure 33: KTM trigger output on a large time scale. The trigger signal is the trigger generated in the KTM. The I and Q values appear as bands here because infinite capture time was set on the scope

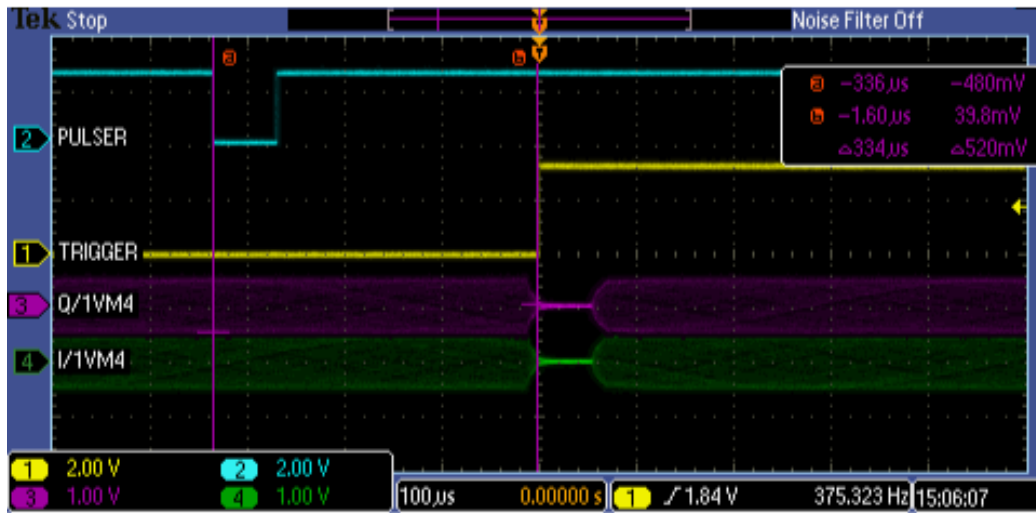


Figure 34: KTM trigger output on a small time scale. The trigger signal is the trigger generated in the KTM

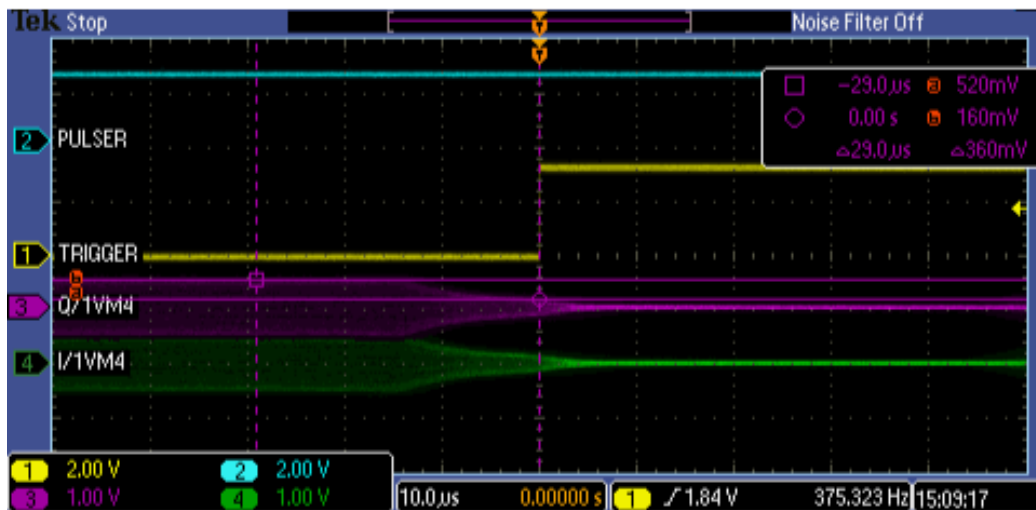


Figure 35: KTM trigger output, showing corresponding 1VM4 Voltage. The trigger signal is the trigger generated in the KTM

3.4.1 ADC

The ADC on the KTM is a 500Ksps throughput 12 bit ADC (LTC 2308). It measures 0 to 4V, however -0.5V was attainable in differential mode. The ADC's resolution is 1mV. With the measurement of two signals (I and Q) and restrictions of the driver provided in the Quartus IP Catalogue, the sample rate is roughly 227Ksps ($4.4\mu s$). This sample rate is not ideal, and 1Msps or higher would be preferable. However, by doing averaging over multiple cycles and knowing the sample rate, the timing of the next 1VM4 falling edge with respect to the falling edge of the pulser signal can be anticipated to an accuracy that should be sufficient for TRIUMF's purpose. This assumes that the trigger timing offset does not change significantly during the averaging period ($32ms$).

3.4.2 FPGA Programming

The program has a few main sections. Most of the logic discussed in this report is inside the KCM_LOGIC.v file. It also contains some pre-programmed functions from the Altera IP catalogue. IP's are used for creating a PLL clock generating an 80MHz signal, logic to control the ADC, and some other logic such as a square root function.

The ADC code was modified slightly to get the best performance out of the on board ADC. Although the ADC says it can do 500Ksps (one sample per $2\mu s$), the clock signal for acquiring data could only reach half of what it was supposed to in the driver. Any higher and the output was wrong. For this reason, the ADC was slightly slower than it should be ($2.2\mu s$). Further analysis could be done to obtain 500Ksps, however the difference was deemed insignificant for the time it would take to resolve.

Once the ADC values have been obtained, the envelope of the signal is calculated as $\sqrt{I^2 + Q^2}$. This is done with some square and square root functions from the Quartus IP catalogue. There is a signal that comes from the ADC IP that indicates when a new ADC value is being written. During this pulse the value of the envelope is uncertain. For this reason, no logic is determined from the envelope during this pulse

To determine if the pulser and 1VM4 signals are high or low, there are states implemented. The state diagrams for these are shown in Figure 36 and Figure 37. It may be easier to visualize the state changes by looking at Figure 39. Once the states reach record1State, they cycle between record1State and record2State. The reason for the states prior to these is to prevent recording of false signals.

The state diagrams are simplified, and in reality they include debouncing buffers so that the states do not switch early due to bouncing. For the 1VM4 signal, less than half of the mean voltage during the high period is defined as the signal being low.

The 1VM4 and pulser states set four values. these values are countValley, countPulse, countValleyStart, and countPulseStart. These values are used to calculate means based off the last 32 pulse cycles, as shown in Figure 40. countValley and countPulse are used to indicate that the program should increment the counter for the beam being low, and the beam being high, respectively. countValleyStart and countPulseStart are high for one clock cycle, at the time when the states change. This indicates that the counters should be reset, and the data from the last pulse should be stored. The logic for calculating the mean period and mean trigger offset values is based on the last 32 pulse cycles. Figure 40 is slightly simplified. One of the things removed is that since the buffer on the 1VM4 signal is larger than the buffer for the pulser signal, it is accounted for when the trigger offset is calculated.

The way that the trigger timing was configured is shown in Figure 38 and Figure 41. An average value of the offset between the pulse signal and the 1VM4 signal is obtained from the previous 32 pulses (Figure 40). This value is then used to approximate when the next pulse will occur. The trigger is then switched on and off according to the ratio of beam that BL1U will take Figure 41. Although this is not currently being used as the trigger, the module will gather data on when it would have triggered. This data can then be evaluated to see if the KTM trigger is better or worse than the current trigger.

The error states in Figure 38 are very important. If the trigger gets stuck in the on state, it would direct all of BL1A ($120\mu A$) into BL1U. The UCN

target is only designed to take a maximum of $40\mu A$, and any more than this would melt the target, and other components downstream. If an error does occur, and the trigger is on, an attempt should be made to turn it off at the correct time. This ramp down time is determined by taking the second last falling edge of the pulser signal, and offsetting by the mean period plus the mean trigger offset. If this does not occur within two milliseconds, the kicker is ramped down.

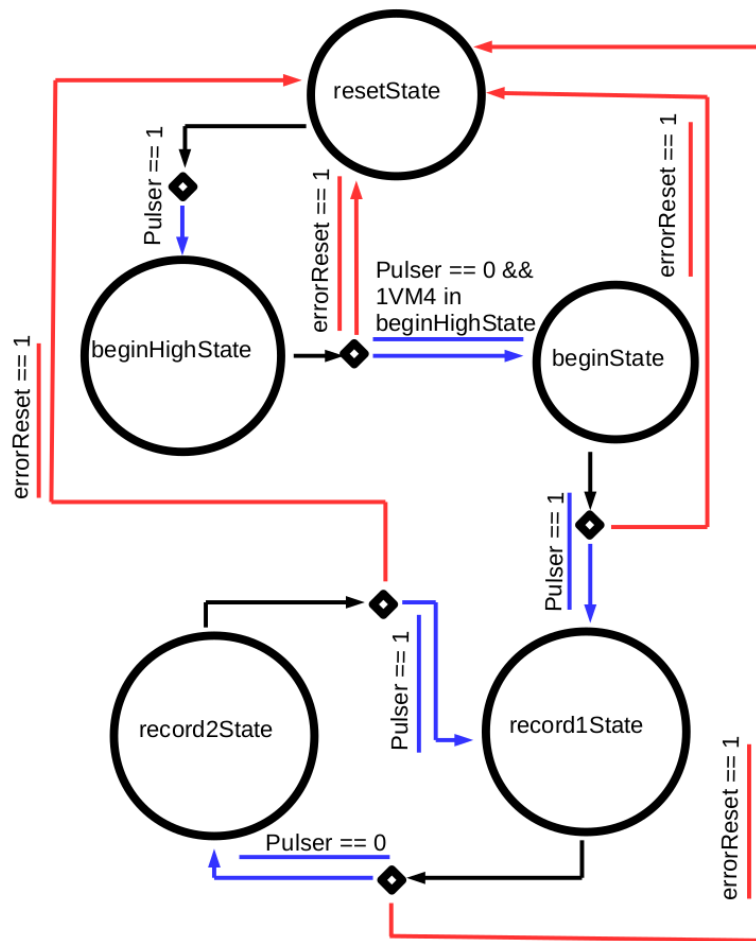


Figure 36: Pulser state diagram. **errorReset** is programmed so it always has priority

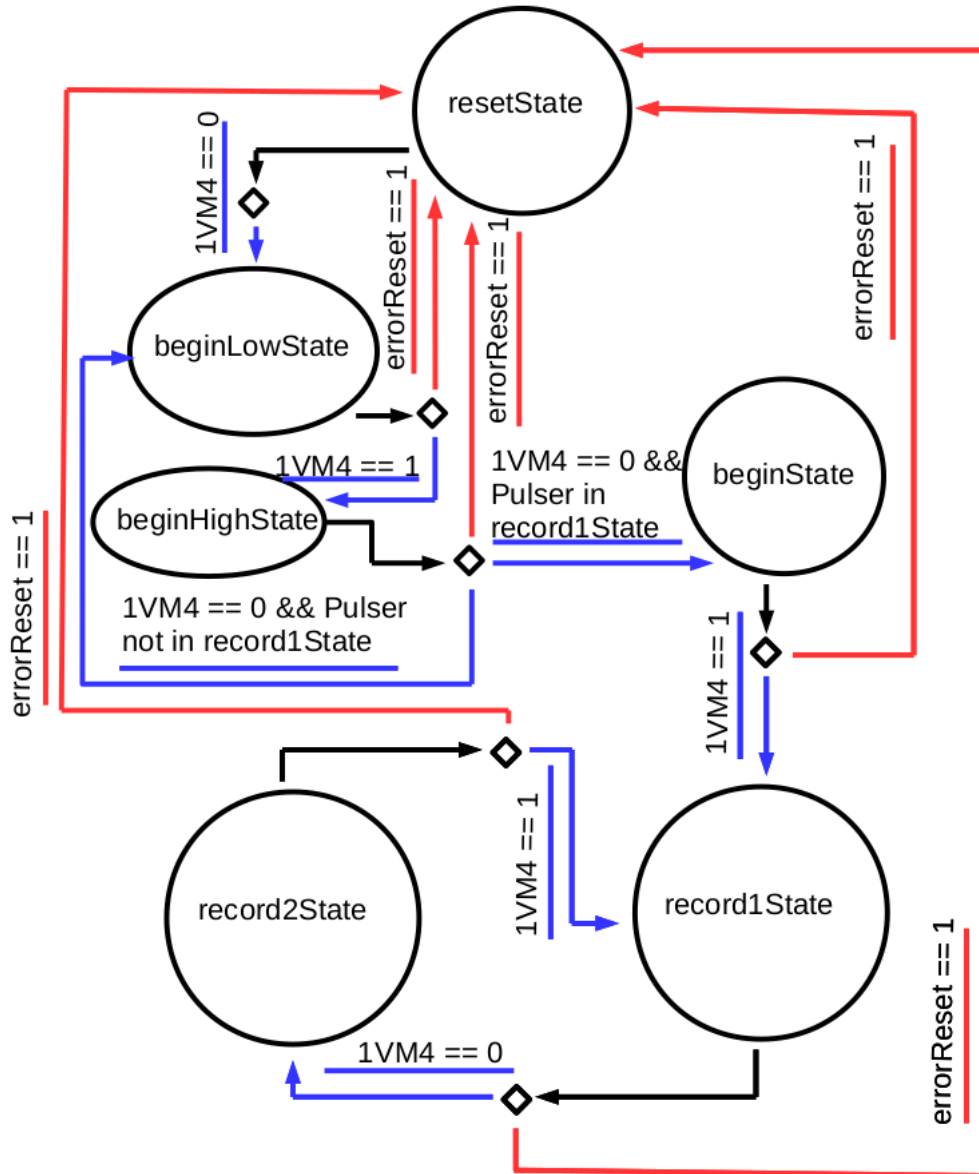


Figure 37: 1VM4 state diagram. **errorReset** is programmed so it always has priority

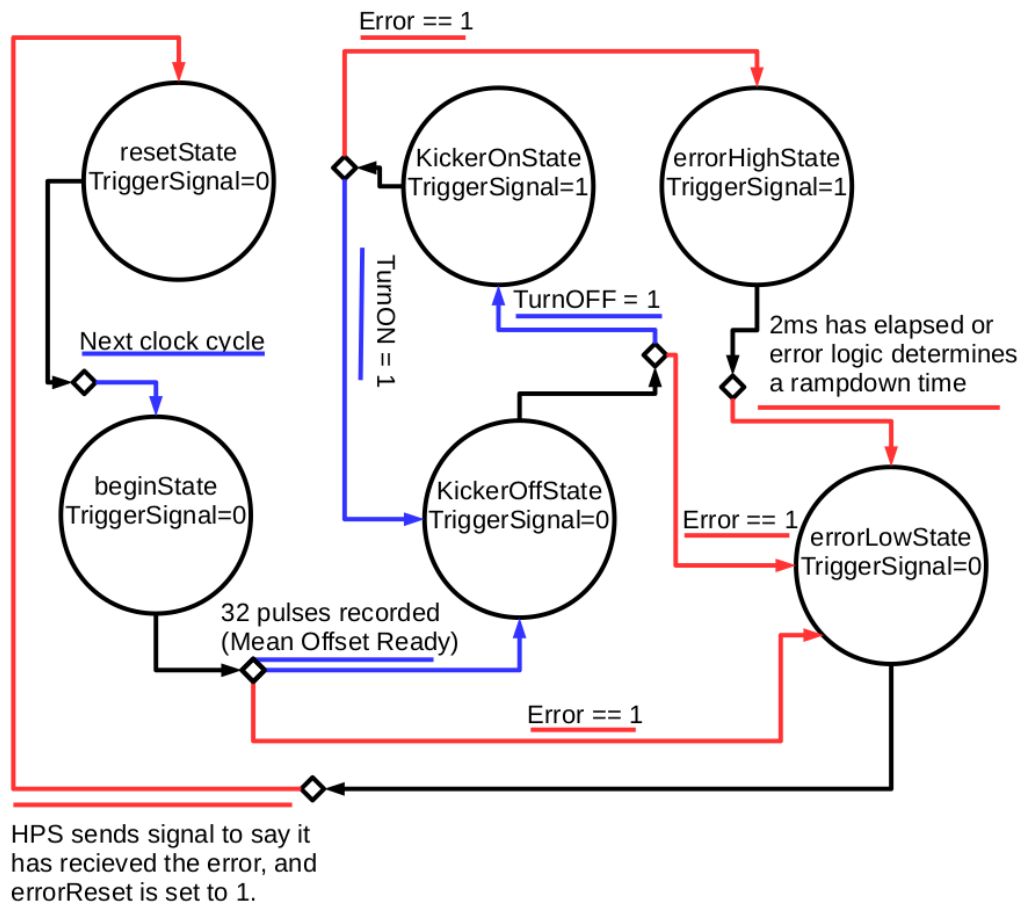


Figure 38: Trigger signal state diagram. error is programmed so it always has priority

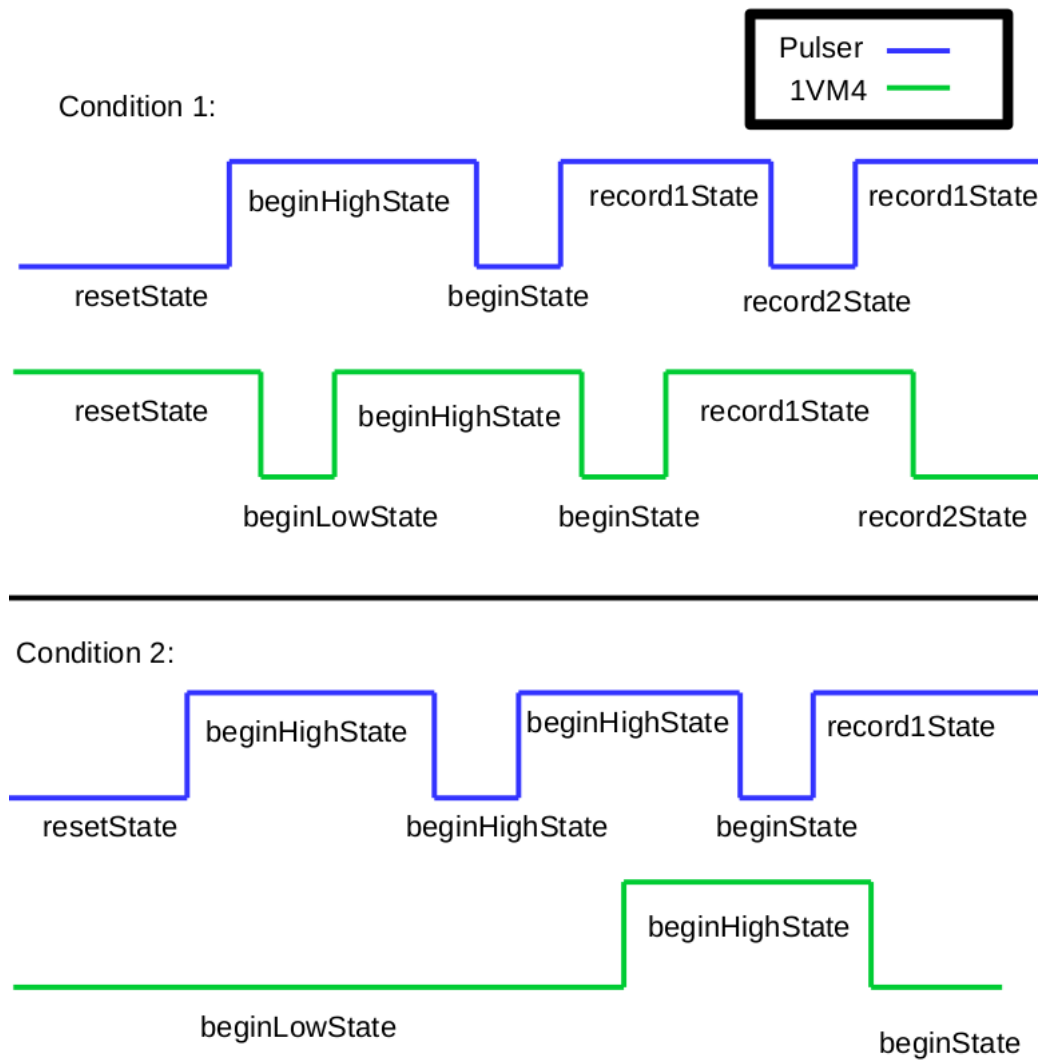


Figure 39: State change scenarios for the beginning of the program

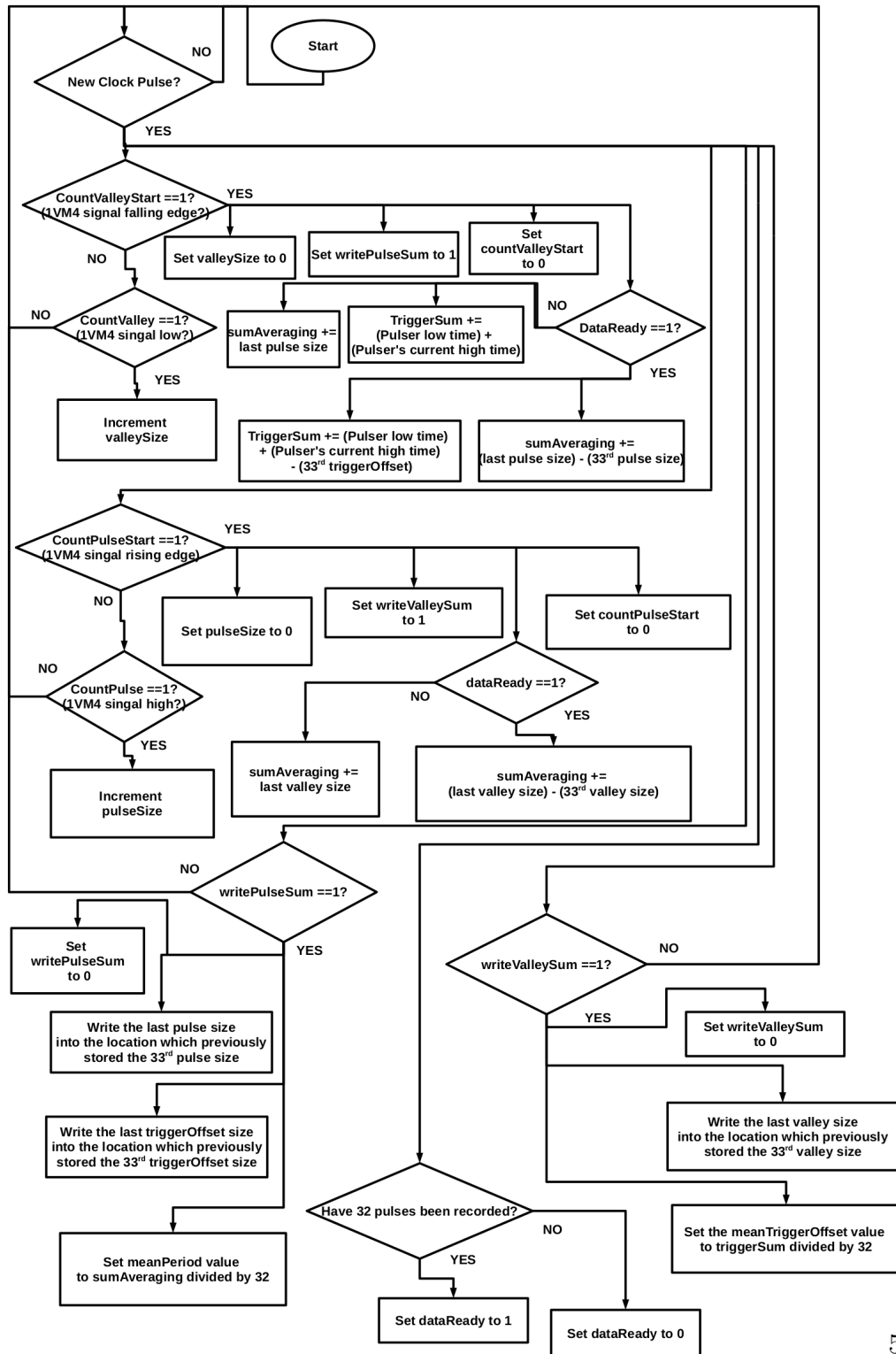


Figure 40: Mean period and trigger offset logic flowchart. Note that this varies from a typical microprocessor flowchart because most actions are preformed simultaneously in the FPGA. All squares are followed by a return to the new clock pulse check.

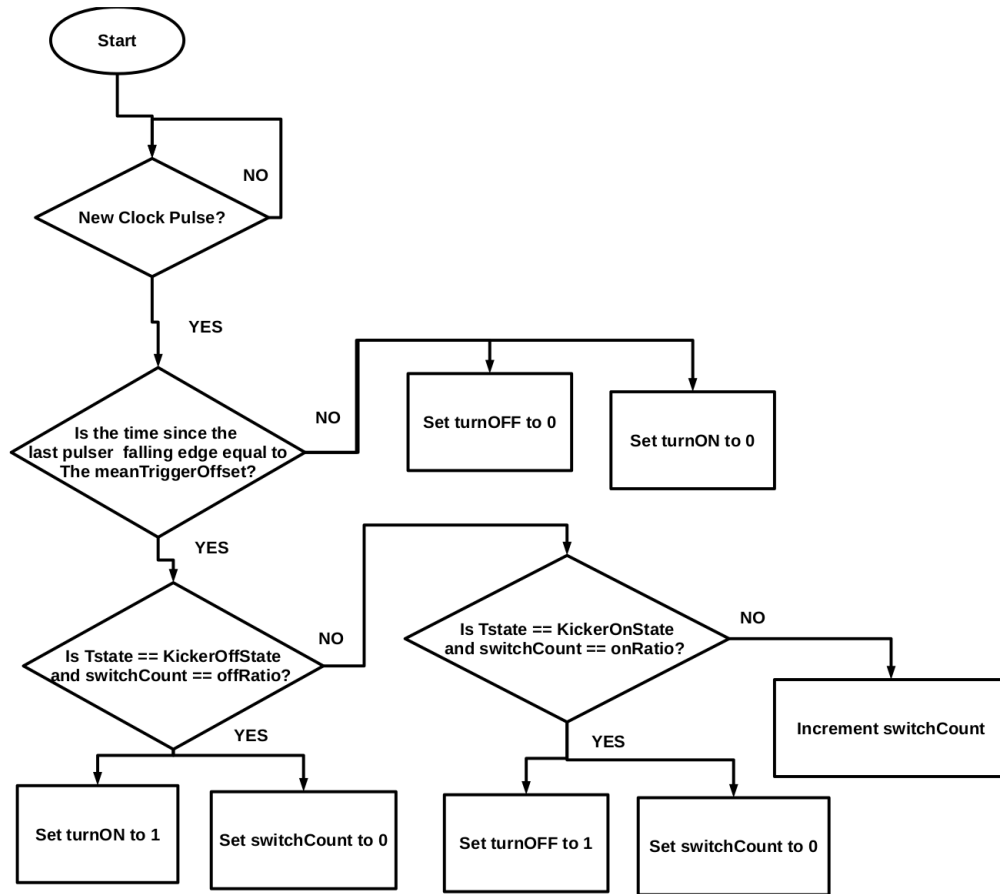


Figure 41: Trigger switching logic flowchart. Note that this varies from a typical microprocessor flowchart because most actions are prefomed simoul-
taneously in the FPGA. All squares are followed by a return to the new clock
pulse check.

There are many error conditions that trip the error states in Figure 38.
The error conditions are shown in Figure 42. Many of them are timing
checks to ensure that the pulser and 1VM4 frequencies are within a sensible
limit. There is also a safety check to prevent the case where the trigger

signal has been high for too long. In any of these error situations, the FPGA will communicate what the error was with the HPS. After this, it will reset the pulse analysis data so that no errors carry over to any of the mean calculations. 32 more cycles will need to be gathered before any more data or trigger outputs occur.

Another section is used for measuring the average beam on voltage and the average residual beam voltage. These values, along with the mean trigger offset, and trigger offset for the current pulse are sent to the HPS upon request. Also, all of the ADC values during the 1VM4 low period and their respective times relative to the pulser's falling edge are sent to the HPS.

The FPGA code for sending sending and receiving data from the HPS is located in the `ghrd.v` file. The logic for this section is the inverse of the logic on the HPS. The FPGA waits for requests from the HPS, and then sends the appropriate data.

3.4.3 HPS Programming

The program on the HPS is fairly simple, since it only has to ask for and receive data from the FPGA. The logic for the C program to write data to the terminal is shown in Figure 43. This should be fairly easy to modify so that the system sends data to MIDAS or EPICS.

```

1 always@(posedge CLOCK_80MHZ)
2 begin
3   if(errorReset == 1 || state==resetState)
4     begin
5       error <= 1'b0;
6       errorType <=6'd0;
7       errorDataR <= 24'd0;
8     end
9   else if(error)
10    begin
11
12    end
13  else if(KickerOnCounter>(20'd100000*OnRatio))
14    begin//if the kicker is on for longer than it should be, give an error
15      error <= 1'b1;
16      errorType <=6'd1;
17    end
18  else if((OffRatio/OnRatio)<7'd2)
19    begin//if ratio of beam on vs off is too large, give an error
20      error <= 1'b1;
21      errorType <=6'd2;
22    end
23  else if(PdataReady && (PvalleyAveraging [Pdatanum] < 14'd3600))
24    begin//If valley is smaller than 45us, give an error
25      error <= 1'b1;
26      errorType <=6'd3;
27      errorDataR <= PvalleyAveraging [Pdatanum];
28    end
29  else if(PdataReady && (PmeanPeriod > 19'd112000))
30    begin//If mean period is larger than 1.4ms, give an error
31      error <= 1'b1;
32      errorType <=6'd4;
33    end
34  else if(PdataReady && (Pperiod > 19'd112000))
35    begin//If last period was larger than 1.4ms, give an error
36      error <= 1'b1;
37      errorType <=6'd5;
38    end
39  else if(PdataReady && dataReady && (PmeanPeriod < 19'd67200))
40    begin//If mean period is smaller than 0.84ms, give an error
41      error <= 1'b1;
42      errorType <=6'd6;
43    end
44  else if(PdataReady && (Pperiod < 19'd67200))
45    begin//If last period was smaller than 0.84ms, give an error
46      error <= 1'b1;
47      errorType <=6'd7;
48    end
49  else if(dataReady && (valleyAveraging [datanum] < 14'd2000) && (writeValleySum == 0))
50    begin//If valley is smaller than 25us, give an error. The reason this is so small is
51      error <= 1'b1;//because this region is smaller than the pulser region (due to
52        capacitance, beamstretch, etc..).
53      errorType <=6'd8;//It was tripping too much at higher values.
54      errorDataR <= valleyAveraging [datanum];
55    end
56  else if(dataReady && (period < 19'd64000))
57    begin//If last period was smaller than 0.8ms, give an error
58      error <= 1'b1;
59      errorType <=6'd9;
60    end
61  else if(dataReady && (period > 19'd112000))
62    begin//If last period was larger than 1.4ms, give an error
63      error <= 1'b1;
64      errorType <=6'd10;
65    end
66  else if(valleySize > 14'd12000)
67    begin//if valley is larger than 150us, give an error
68      error <= 1'b1;
69      errorType <=6'd11;
70      errorDataR <= meanHighVoltage;
71    end
72  else if(pulseSize > 18'd100000)
73    begin//if pulse is larger than 1.25ms,give an error
74      error <= 1'b1;
75      errorType <=6'd12;
76      errorDataR <= pulseSize;

```

```

76 end
77 else if(PvalleySize > 14'd12000)
78 begin//if valley is larger than 150us, give an error
79     error <= 1'b1;
80     errorType <=6'd13;
81 end
82 else if(PpulseSize > 18'd100000)
83 begin//if pulse is larger than 1.25ms,give an error
84     error <= 1'b1;
85     errorType <=6'd14;
86 end
87 else if((datanum + 6'd1 != Pdatanum) && (datanum != Pdatanum) && ((datanum == 6'd32 ||
88     datanum == 6'd31) && (Pdatanum != 6'd0)))
89 begin//if trigger pulses are misaligned
90     error <= 1'b1;
91     errorType <=6'd15;
92 end
93 else if(errorFlag)
94 begin//due to PLL not being locked
95     error <= 1'b1;
96     errorType <=6'd16;
97 end
98 else if(offsetReady && (TrigTime == (meanTriggerOffset + 18'd1200)) && (IQvalue > (
99     meanHighVoltage/2)) && (done == 0))
100 begin//if 1VM4 is high 15us after mean offset time. send error and reset.
101     error <= 1'b1;
102     errorType <=6'd17;
103 end
104 end
105 end

```

Figure 42: FPGA error conditions

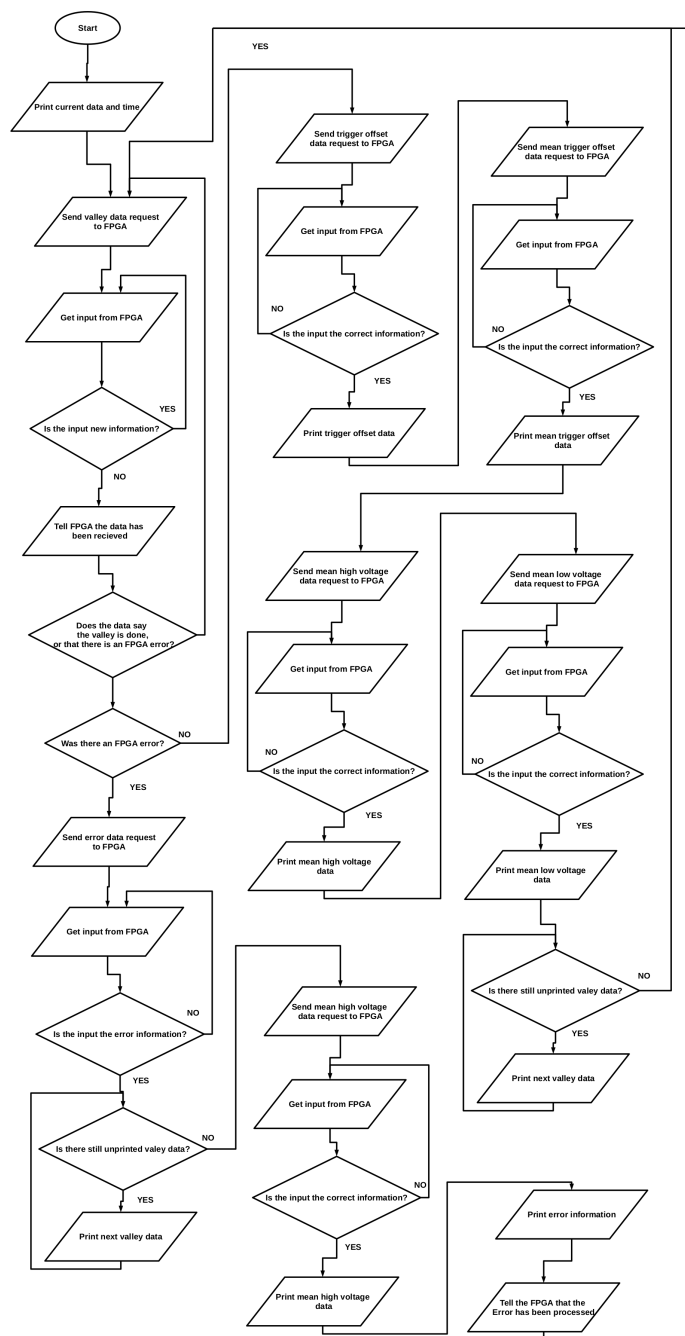


Figure 43: HPS code flowchart. The real implementation varies slightly, but this is the general flow. This as well as all other flowcharts can be found at ”https://ucn.triumf.ca/triumf/magnets/kicker/ktm/ktm_programming_info”

3.5 Guide for Operating the KTM

3.5.1 Uploading FPGA and C programs to the KTM

There are multiple ways to upload programs to the KTM. Here I will explain two different methods. The first is through Quartus. Once your program is compiled, select Tools→Programmer. Choose the correct file and click start. With Linux it may be difficult to connect to the FPGA this way (it was for me) Windows may be a simpler OS for Programming via Quartus. Make sure the MSEL switch array on the DE0 board is set correctly (2 and 4 set to '1', the rest set to '0')

The alternative to the above method is to compile your program and convert it into a .rbf file by running the following command in the program's directory

```
1 $ quartus_cpf -c ProgramName.sof ProgramName.rbf
```

where "ProgramName" is the name of the program.

Upload the .rbf file to the SD card on the HPS. To do this you can either plug the microSD card into your computer and transfer the file, or transfer the file via ethernet (recommended). To connect via ethernet, follow instructions in in subsection 3.5.3. In the terminal, run:

```
1 $ scp ./ProgramName.rbf root@10.42.0.150:~/ProgramName.rbf
```

Using the IP address of the KTM.
program the FPGA by running:

```
1 ./program_fpga.sh ProgramName.rbf
```

Make sure the MSEL switch array on the DE0 board is set correctly (all switches set to '0')

If you would like to reconfigure the HPS program, change the C files and re-compile with the command

```
1 $ make
```

To upload the C program to the SD card, you can use the same method for transferring files to the HPS as outlined above.

3.5.2 HPS communication via USB

Connecting to the HPS of the KTM via USB requires a mini USB to USB cable, and a computer. You will need to download a serial port connection program (GtkTerm for Linux, or PuTTY for Windows). Here I will explain how to use GtkTerm to connect. Open up the program and click Configuration→Port. Select the port that the KTM is connected to. The baud rate is 115200. The rest of the settings are shown in Figure 44. Once connected, the login is root.

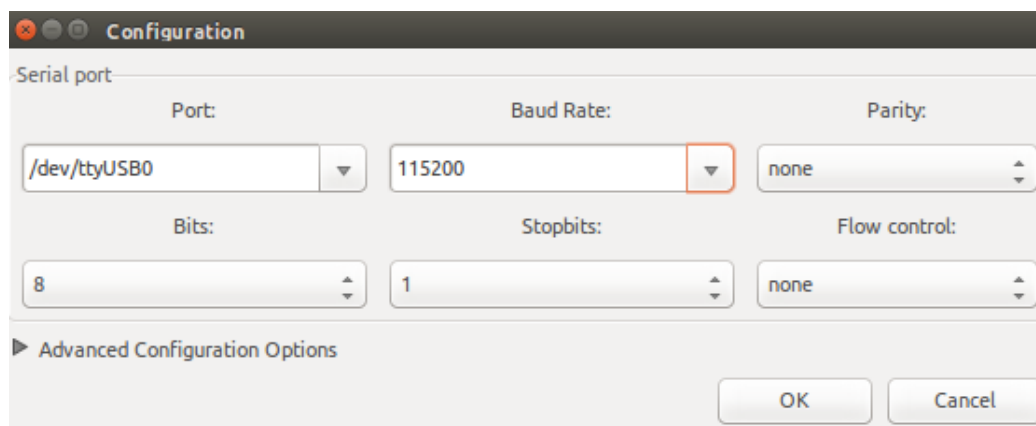


Figure 44: USB Connection Settings

3.5.3 Recording Data

There are 3 different ways to record data. The best way for long term use is via ethernet. In the future, it may be beneficial to obtain a permanent IP address for the KTM on TRIUMF's system. A method to connect directly to the board without a router follows (using Linux). Connect a computer directly to the ethernet port of the DE0. Go to the network settings on the computer and edit IPv4 settings to "Shared to other Computers" as shown if ??.

First we need to find the Bcast address for the ethernet connection. Open a terminal and enter:

```
1 $ ifconfig
```

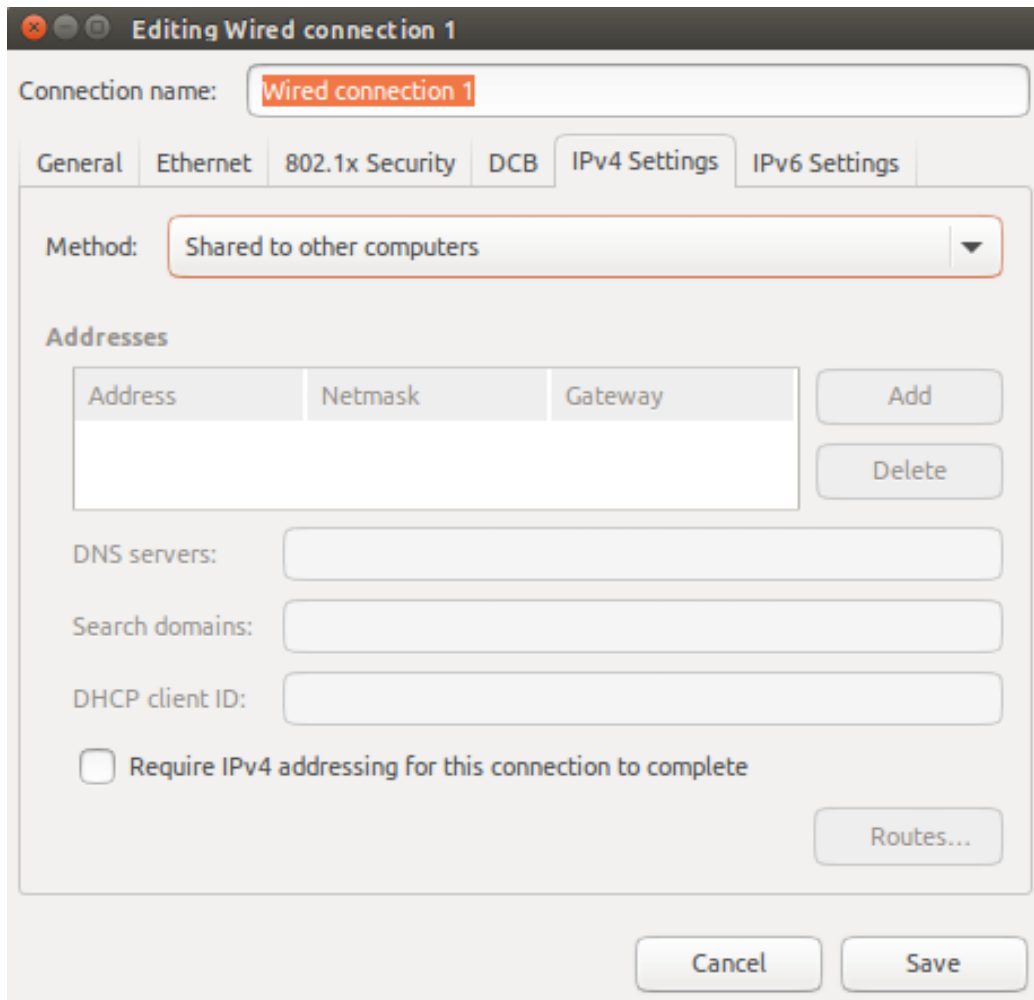


Figure 45: Ethernet Connection Settings

An example output is shown below. In this example, Bcast is 10.42.0.255 and the computer's IP address is 10.42.0.1

```

1 enp2s0f0  Link encap:Ethernet  HWaddr c8:2a:14:5a:46:0c
2          inet addr:10.42.0.1  Bcast:10.42.0.255  Mask:255.255.255.0
3          inet6 addr: fe80::ca2a:14ff:fe5a:460c/64  Scope:Link
4          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
5          RX packets:497434  errors:0  dropped:0  overruns:0  frame:0
6          TX packets:182147  errors:0  dropped:0  overruns:0  carrier:0
7          collisions:0  txqueuelen:1000
8          RX bytes:114513162 (114.5 MB)  TX bytes:12863464 (12.8 MB)
9          Interrupt:16

```

Next, we need to find the IP address of the KTM. You can do this from your

computer or by connecting to the HPS via usb. From the HPS terminal you can find the IP with:

```
1 $ ifconfig
```

and look for the inet addr of the ethernet. For connecting to the HPS via USB, see subsection 3.5.2

Alternatively, from your computer, install nmap:

```
1 $ sudo apt-get install nmap
```

Then run the following command using your Bcast value:

```
1 $ nmap -n -sP 10.42.0.255/24
```

You should get an output that looks like this:

```
1 Starting Nmap 6.47 ( http://nmap.org ) at 2016-07-18 12:06 PDT
2 Nmap scan report for 10.42.0.1
3 Host is up (0.00035s latency).
4 Nmap scan report for 10.42.0.150
5 Host is up (0.00038s latency).
6 Nmap done: 256 IP addresses (2 hosts up) scanned in 2.41 seconds
```

The first address is the IP address of the computer. The second is the address of the KTM board. If your output does not look like this, then you may need to restart the HPS or check the connection.

Now you should be able to connect to the HPS with the following command (using the KTM's IP):

```
1 $ ssh -Y root@10.42.0.150
```

The password for root is: UCNsRslow

Set up data logging on your computer's terminal. This way, all the data will be stored on your computer. Make sure you have enough space available. You may want to set the date and time before running (the program will output the time roughly every hour, and at the time of any error).

```
1 date -s '2016-07-18 12:34:56'
```

check the current time with:

```
1 date
```

You can program the FPGA by running:

```
1 ./program_fpga.sh ProgramName.rbf
```

To begin acquisition, run:

```
1 ./KTM_Terminal_Print
```

To end the program, use ctrl+z.

Another way to record data is to write to a file on the HPS. This has the disadvantage that the SD card connected to the HPS is only 4GB. 21MB/min. This program should not be left running for more than an hour. To write data to a file on the HPS, run:

```
1 ./KTM_File_Print
```

The third way to record data (not recommended) is to connect via USB, and save the terminal output to a file (Log→To File... in GtkTerm). Using Ethernet is better because the data transfer rate is not fast enough via USB. For connecting via USB, see subsubsection 3.5.2.

4 EDM Cell Coatings

4.1 Background

As discussed in subsection 1.1, in order to store UCN, the storage walls must have a high Fermi potential. Although this parameter is very important, there are other constraints for UCN storage materials. The materials chosen must have low neutron absorption rates and be mechanically strong enough for any applied loads. Another major constraint is that any material after the neutron polariser must be non-magnetic, and have a low spin flip probability.

The EDM cell walls have to fulfil not only the basic UCN storage constraints, but also a few more. The EDM cell walls must be electrically insulating, and not cause any electrical breakdown. In addition, any material covering the magnetometer windows must be UV transparent. For the co-magnetometer at TRIUMF, the windows must be 99.9% transparent.

There are very few materials that satisfy the above criteria. The two that are the most promising are deuterated polyethylene (DPE), and deuterated polystyrene (DPS). DPS has a slightly lower Fermi potential than DPE, and quickly deteriorates under UV light. Although DPE does seem like a superior material for the nEDM cell, it is mechanically weaker than DPS. Also, in order to dissolve DPE for coating it must be heated to about 130°C. This makes the procedure for coating DPE much more complicated, however it can be done. The current (as of 2016) nEDM experiment running at PSI uses a DPS coated cell with DPE coated magnetometer windows [7].

In June 2016, Cold neutron reflectometry was conducted at J-PARC in order to measure the properties of various UCN storage materials. Neutron reflectometry requires a very smooth coating in order to obtain good angle retention. For this reason, DPE and DPS were coated onto 2" diameter silicon wafers. The samples were produced via spin coating.

Neutron reflectometry works by varying parameters in Equation 1 in order to determine the fermi potential a material. A sample is placed in the path of a beam of cold neutrons. The angle between the sample and the beam is very small in order to see the reflection. Either the energy or incident angle

of the beam is varied. The lowest energy (or smallest angle) in which some energy is absorbed by the material is directly correlated to the material's fermi potential. Equation 1 is a simple model, and the real measurement takes into account other parameters. For more information, see [10] [11].

Since deuterated polymers are expensive, their non-deuterated counterparts were used to test the procedure outlined below. Although the produced non-deuterated coatings are expected to be slightly different than the deuterated coatings, the procedure is similar.

4.1.1 Solution Mixing

For polystyrene (PS) coatings, the PS or DPS was first weighed out and placed inside an erlenmeyer flask and covered with glass stopper. Toluene, or d-Toluene was then transferred into the flask. This solution was then left to sit for at least one day. It was found that running the solution through a qualitative filter produced coatings with less contaminants. Better filters (fine particle size) would be preferable.

For polyethylene (PE) coatings, the PE or DPE was measured and mixed with o-xylene or o-xylene d10 inside an erlenmeyer flask. the solution was heated to just below the boiling point of o-xylene (around 130°C). The glass stoppers being used did not create a good enough seal to boil the solution, however if an adequate seal can be made, it would be preferable to boil the solution slightly. The time it takes for the PE to dissolve greatly depends on the molecular weight. For the PE used, the solution became completely dissolved within fifteen minutes. For the DPE, it took two or more hours, and was mixed with a magnetic stirrer so the DPE would dissolve faster.

4.2 Procedure for Si wafer coatings

4.2.1 Cleaning

The coatings were produced in a class 1000 cleanroom, under a fume hood. Before coating, the Si wafers were cleaned first with acetone, then with 2-propanol, then with deionized water. This was done in an ultrasonic bath for five to ten minutes per cleaning solution. The wafers were then sprayed with pressurised nitrogen. Additional cleaning with piranha and HF was

done on some of the PE coatings, but did not appear to effect the coatings significantly. All other materials that touch the coating solution were cleaned thoroughly. This included the erlenmeyer flasks for the solvents, the syringe, and the pipettes. The spin coater and oven (for PE) were also cleaned.

4.2.2 Deposition

For PS, deposition was fairly simple. The wafer was placed onto the spin coater and spin parameters were set. The solution was then either poured or syringed onto the wafer, and then spun. It was found that 2-4 ml of solution was adequate for completely wetting the 2" diameter wafer.

Deposition of PE was more involved, and requires two people. This is because the wafer and syringe both had to be placed in an oven to heat them to the deposition temperature. It was discovered that the oven being used had a lot of particles in it. To avoid contamination, the oven was set to 150°C, and the first person would quickly spray the wafer with pressurised nitrogen before placing it on the spin coater. At the same time, the second person would take the syringe out of the oven and retrieve 2-4ml of solution from the hot beaker. They would then deposit it onto the wafer, and turn the spin coater on. This should be done as quickly as possible (10-15 seconds), since it is important that the wafer and solution do not cool down much before spinning.

4.3 Results from PE/PS Coatings

Extensive preparation for DPE and DPS coatings was done with PE and PS, in order to have a clear understanding of the procedure and the expected quality of the coatings. The objective was to reproducibly create contaminant free polymer coatings 200 nm thick or greater, with surface roughness as low as possible.

A fimetrics F20 spectrometer was used to measure the thickness of the coatings, and a Dektak XT Profilometer was used to measure the roughness. Figure 46 shows a PS roughness measurement. Parameters that were expected to effect coating properties were recorded. Table 5 and Table 6 show the deposition parameters and coating measurements.

4.3.1 PS Coatings

For the PS coatings, the coating procedure was quickly optimised. It was found that spinning below 500 rpm created inconsistent coatings, despite previous papers reporting 10 rpm as their rotating speed. It appears that between 2% and 5% PS by weight is the optimal concentration for TRIUMF's desired coating thickness. Any lower concentration, and the coatings are too thin. Any higher concentration will make the solution too viscous, and the coating may become too rough. The roughness of the PS coatings measured were reasonable for the neutron reflectometry at J-PARC.

4.3.2 PE Coatings

The PE coatings took many iterations in order to achieve the desired properties. This is because the added complication of heating the solution and wafer made the coatings difficult to control. The optimal rotating speed used was 2000 rpm. It was found that any solution with less than 1% PE by weight produced a coating thickness below 200 nm. The roughness of the coatings was approximately one third of the average thickness, which is typical for spun PE coatings [8]. For this reason, the optimal thickness for PE was as close to 200 nm as possible since any thicker coating would cause an increase in roughness.

There was an issue with contaminants constantly occurring on the PE coatings. Some theorised sources of contamination were from the beaker used, from the PE and o-xylene solution, and from the oven used to heat the wafer. It was found that the oven was the main source of contamination. In order to reduce the contamination, the oven was cleaned thoroughly. The wafers were also heated to a temperature higher than the PE solution, and cleaned of dust with pressurised nitrogen before being placed onto the spinner. This procedure appeared to reduce the contaminants on the wafers.

4.4 Results from DPE/DPS Coatings

While the DPS coatings deposited as expected, some issues were encountered while producing the DPE coatings. The issue had to do with dissolving the DPE in o-xylene d10. The dissolution time was much longer than it was for the PE. This was most likely due to the high molecular weight of the DPE



Figure 46: Dektak plot of surface roughness. 5% PS by weight solution, spun at 2000rpm

($M_w=361,000$, $M_n=212,000$, $M_w/M_n=1.70$). The molecular weight of the PE used is unknown. It was found that using a magnetic stirrer to mix the solution decreased the time it took for the DPE to dissolve. Unfortunately, due to the limited amount of deuterated o-xylene, the DPE coatings produced for measurement at J-PARC were dissolved in non-deuterated o-xylene. This should not effect the results, since most of the o-xylene will evaporate, and it is not definitive that using deuterated solvents will decrease the neutron absorption rate. Also, introducing traces of hydrogen in the coating should not cause a significant change in absorption due to a change in incident angle or neutron energy.

The results from reflectometry showed that the high surface roughness of the DPE samples caused them to appear as if they had a lower fermi potential than expected. We know that $V \propto \sin(\theta + \frac{Y-Y_0}{2L_2})^2$ where L_2, Y_0, Y have to do with the positioning of the detector and the neutron collision distance from the center of the wafer. we know that $\theta = 0.3$ if the wafer is perfectly smooth. However, the roughness of the DPE wafers will make this parameter larger. a rough estimate of the maximum θ value of the DPE wafers measured at J-PARC is $\theta = 0.453$. To get a realistic value of the fermi potential for DPE, the roughness measurement should be done more thoroughly, and must be applied to the fermi potential analysis.

5 Testing of UCN's Cryogenic Pressure Switch

5.1 Background

The UCN source at TRIUMF plans to use Superfluid Helium with a temperature below 1°K. As a safety measure, it is important to probe the pressure of this Helium. The issue with this is that there are no reliable pressure sensors for such low temperatures. A burst disk could be used, but with the disadvantage that it must be replaced if it bursts.

The solution decided on at TRIUMF is to have a bellow with a screw connected to it. when enough pressure is applied on the helium side, the bellow will compress and the screw will come in contact with an electrical switch. The length of the screw can be adjusted so that the switch will trip at the desired pressure.

The pressure that the switch will trip at is temperature dependant. In order to determine this pressure for superfluid helium temperatures, the pressure switch was tested both at room temperature and at liquid nitrogen (LN₂) temperature.

5.2 Experimental Set-up

Gaseous helium was used as the source of pressure for both the room temperature and LN₂ tests. A pressure regulator (HFR-1524-1515-R-FG-L from Kelly Pneumatics, Inc.) with a maximum outlet pressure of 15 psi was used. This pressure regulator also records the outlet pressure. A valve was placed on both the inlet and the exhaust of regulator. They were both set to be almost closed. When the pressure on the pressure regulator was set to 15 psi, the pressure would increase slowly. When the pressure was set to 0, the pressure would decrease slowly. This way, the pressure which caused the switch to trip could be determined for both the rising and falling edges. The controls and data recording were done with LabVIEW.

For each test the screw length was set, measured with a caliper, and recorded. For LN₂ tests, the pressure switch was slowly lowered into the

LN₂ over a period of 10-15 minutes. This was done so that the electrical switch was not broken due to temperature shock. The switch was then pressurised on and off a number of times. The pressure, time, and switch state data was recorded to a file.

5.3 Findings

The screw used for the tests was a 50mm M3x0.5 slotted cheese head screw. The issue with using this screw was the slot. Sometimes, the switch head would slide inside the slot, causing slight variances between tests. This screw should be replaced with a 50mm M3x0.5 screw with a hex head.

Table 5: Polyethylene Coatings

Polyethylene Coatings								
Concentration (WT%)	rpm	rpm/s	Wafer (°C)	Solution (°C)	Coating (nm)	Non-uniformity (nm)	Volume (ml)	Duration (mins)
0.5	1000	220	130	130	120	?	2	1
0.5	1000	990	140	130	91	20	2	1
0.5	1000	990	150	130	120	?	2	1
0.5	1000	990	165	130	120	?	2	1
0.5	1500	?	?	?	100	?	2	1
0.5	2000	?	150	130	82	38	2	1
0.5	2000	?	140	130	93.8	38.6	2	1
0.565	2000	?	140	130	95.5	22.6	2	1
0.75	1000	990	140	130	116.2	14.8	2	1
0.75	2000	990	140	130	106.2	17	2	1
0.75	2000	1650	140	130	91.6	9.95	2	1
0.75	3000	1650	140	130	103.6	26.5	2	1
0.81	2000	990	130	130	196	57	2	1
0.81	2000	990	130	130	244	76	2	1
1	1000	?	150	130	?	?	2	1
1	1500	?	150	130	?	?	2	1
1	2000	?	150	120	?	?	2	1
1.1	2000	990	130	130	180	47	2	1
1.1	2000	990	130	130	204	61	2	1
1.2	2000	990	150	130	168	43	2	1
1.2	2000	990	130	130	183	53	2	1
1.5	2000	990	140	130	368	110	2	1
Deuterated Polyethylene Coatings								
Concentration (WT%)	rpm	rpm/s	Wafer (°C)	Solution (°C)	Coating (nm)	Non-uniformity (nm)	Volume (ml)	Duration (mins)
0.4	2000	990	150	130	80	25	2	1
>0.82	2000	990	150	130	220	50	2	1
>0.82	2000	990	150	130	210	40	2	1
>0.93	2000	990	150	130	200	?	2	1
>0.95	2000	990	150	130	140	30	2	1
>1	2000	990	150	130	75	35	2	1

Table 6: Polystyrene Coatings

Polystyrene Coatings								
Concentration (WT%)	rpm	rpm/s	Wafer (°C)	Solution (°C)	Coating (nm)	Non-uniformity (nm)	Volume (ml)	Duration (mins)
0.5	40	?	25	25	50-500	?	0.5	4
1	40	?	25	25	70-500	?	0.5	Until Dry
2.5	500	?	25	25	305	?	1.5	1
2.5	400	?	25	25	355	?	1.5	1
2.5	300	?	25	25	410	?	2	1
2.5	200	?	25	25	305	?	2	1
5	2000	?	25	25	500	?	2	1
5	500	?	25	25	950	?	1	1
5	500	?	25	25	1000	?	2	1
Deuterated Polystyrene Coatings								
Concentration (WT%)	rpm	rpm/s	Wafer (°C)	Solution (°C)	Coating (nm)	Non-uniformity (nm)	Volume (ml)	Duration (mins)
2.8	1000	550	25	25	480	10	2	1
4	1000	550	25	25	1070	10	2	1

References

- [1] T. Linder, UCN Kicker Diagnostics and Control, UCN weekly meeting, TRIUMF, 23rd of December 2015
- [2] R. Picker, The UCN facility and EDM experiment at TRIUMF, TRIUMF International Peer Review 2013, 14th of November 2013
- [3] J. H. Christenson, J. W. Cronin, V. L. Fitch and R. Turlay, "Evidence for the 2π Decay of the K_2^0 Meson," Phys. Rev. Lett. 13, 138 (1964)
- [4] Andreas Knecht, The status of the search for a nEDM and the new UCN sources, Workshop on the New, the Rare and the Beautiful, University of Zurich, 6.-8. January 2010
- [5] Improved Experimental Limit on the Electric Dipole Moment of the Neutron PRL 97,131801 (2006)
- [6] D.J.R. May, A High Precision Comparison Of The Gyromagnetic Ratios Of The 199Hg Atom And The Neutron, Thesis: The University of Sussex, September, 1998
- [7] K. Bodek, M. Daum, R. Henneck, S. Heule, M. Kasprzak, K. Kirch, A. Knecht, M. Kuźniak, B. Lauss, M. Meier, G. Petzoldt, M. Schneider, G. Zsigmond, Storage of ultracold neutrons in high-resistivity, non-magnetic materials with high Fermi potential, Nucl. Instr. and Meth. A, doi:10.1016/j.nima.2008.09.018 (2008).
- [8] O. Mellbring, S. K. Øiseth, A. Krozer, J. Lausmaa, T. Hjertberg, Spin coating and characterization of thin high-density polyethylene films, Macromolecules 34 (7496-7503).
- [9] S. Chahal, Layout and EDM Monte Carlo Simulations for the UCN Experiment at TRIUMF. Vancouver: TRIUMF, 2016.
- [10] E. Pierre, Report on UCN guides. NEDM weekly meeting, TRIUMF, 28th of June 2016
- [11] Mitamura K, Yamada NL, Sagehashi H, Torikai N, Arita H, Terada M, Kobayashi M, Sato S, Seto H, Goko S, Furusaka M. Novel neutron reflectometer SOFIA at J-PARC/MLF for in-situ soft-interface characterization. Polymer journal. 2013 Jan 1;45(1):100-8.

- [12] Pendlebury JM, Heil W, Sobolev Y, Harris PG, Richardson JD, Baskin RJ, Doyle DD, Geltenbort P, Green K, Van Der Grinten MG, Iaydjiev PS. Geometric-phase-induced false electric dipole moment signals for particles in traps. *Physical Review A*. 2004 Sep 9;70(3):032102.