ORNL/TM-2006/62

Neutron and Gamma-Ray Pulse Shape Data Acquisition from a Cf-252 Source

April 2006

Prepared by
Sara A. Pozzi
Senada Avdic

ORNL/TM-2006/62

Nuclear Science and Technology Division

NEUTRON AND GAMMA RAY PULSE SHAPE DATA ACQUISITION FROM A CF-252 SOURCE

Sara A. Pozzi and Senada Avdic

Date Published: April 2006

Prepared by
OAK RIDGE NATIONAL LABORATORY
P.O. Box 2008
Oak Ridge, Tennessee 37831-6285
managed by
UT-BATTELLE, LLC
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This report describes a data acquisition program that can be used in conjunction with the Tektronix TDS-5104 oscilloscope running Matlab to acquire neutron and gamma-ray pulses from an instrumented Cf-252 source. The program was used to acquire pulse shapes from the anode of a liquid scintillation detector. The pulses were then analyzed and fitted to determine the decay constants of the scintillation components. The Matlab program can also be used to measure the neutron response matrix of this type of detector for neutron energies up to a few MeV.

1. INTRODUCTION

The knowledge of the neutron spectra from nuclear materials is of interest in many applications ranging from nuclear nonproliferation to homeland security. Recent studies[1] have shown that the information on the energy spectrum of neutrons in such measurements can increase the sensitivity of the assays performed on nuclear materials, whose scope is often to determine the mass and composition of the fissile samples. Organic scintillators (liquid and plastic) are currently being investigated as the main components of various measurement systems, such as portal monitors.

Neutron spectrum unfolding with organic scintillators is challenging because neutrons of a given incident energy generate a wide distribution of pulse heights in the detector. These distributions are the components of the neutron response matrix, which links incident neutron energy and detected pulse height. The problem is generally ill posed, so small variations in the pulse height distribution lead to a large variation in the unfolded spectrum.

The purpose of this report is to describe a Matlab program that can be used with the Tektronix TDS-5104 oscilloscope to acquire pulses from the anode of a scintillation detector from the neutrons and gamma rays emitted by an instrumented Cf-252 source.[2]

2. DETECTOR

The detector is a cylindrical liquid scintillator manufactured by Bicron, model 4.62MAB-1F3BC-501A/5. It is 7.7 cm thick, with a 15.2-cm diameter. The container is made of aluminum. The front face is approximately 2 mm thick. The side is a double aluminum wall with a 2-mm-thick external layer and a 0.5-mm-thick internal layer. The inside of the container has a clear-anodized surface treatment. The photomultiplier tube is mounted on the back circular surface.

3. MEASURED PULSES – GAMMA RAYS

Gamma-ray pulses were obtained from a Cs-137 source placed on the face of the detector. The oscilloscope fast frame acquisition mode was used to capture the pulses, with 0.2-ns resolution. Approximately 4200 pulses can be acquired at a time using this acquisition mode. The pulses were binned into 14 groups of increasing and equal pulse area, obtaining approximately 500 to 1000 pulses per area bin. The pulses were then averaged to obtain average waveforms, which are shown in Fig. 1 without normalization. Figure 2 shows the average waveforms normalized to the maximum value: it can be seen that the pulse shapes do not change significantly as a function of their total area.

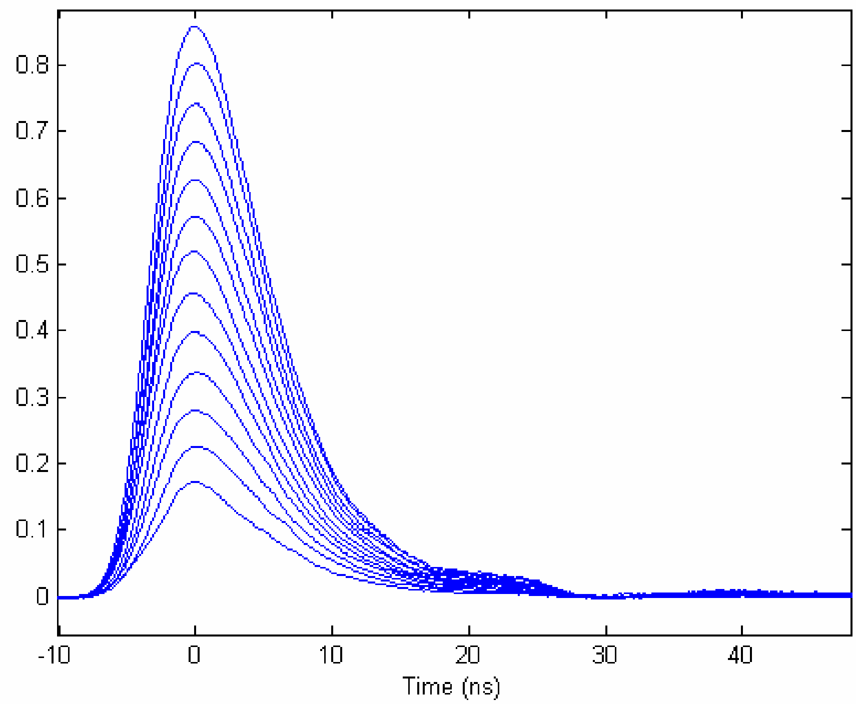

Fig. 1. Measured average pulse waveforms from gamma rays obtained from the anode of the liquid scintillator.

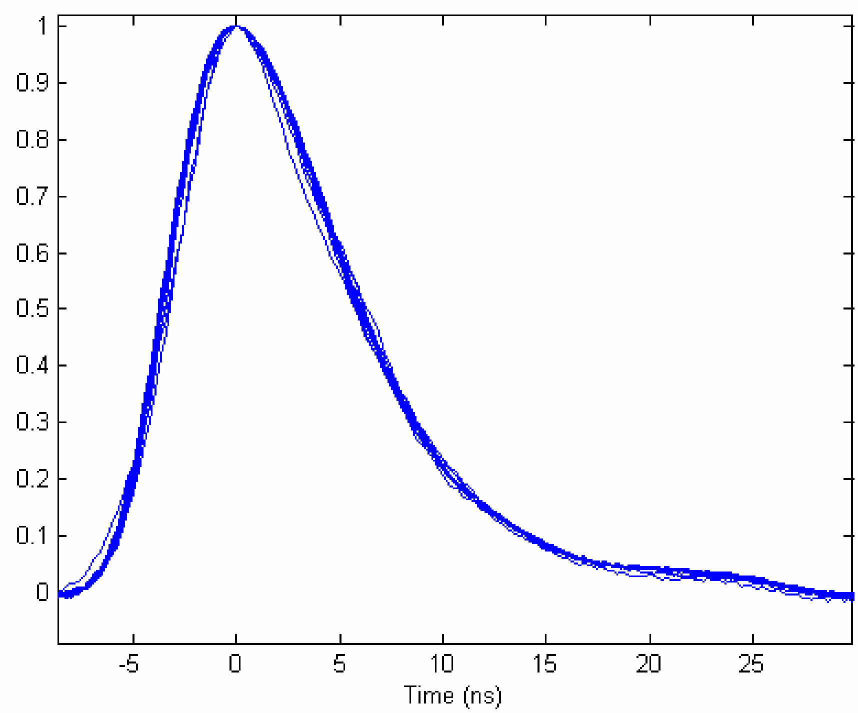

Fig. 2. Measured average pulse waveforms of Fig. 1 normalized to maximum value.

4. MEASURED PULSES – NEUTRONS

The measurement of neutron pulses with the liquid scintillator was performed using a timed Cf-252 spontaneous fission source. The time-of-flight method was used to discriminate neutrons from gamma rays and to determine the incident neutron energy. In the experiment, the Cf-252 source was placed at a 1-m distance from the liquid scintillator. The use of the Matlab Instrument Control Toolbox allowed a totally automated data collection with the program *matrixmeas.m* and functions *acquire.m* and *getpulse.m*, which are given in Appendices A through C, respectively.

Figure 3 shows an example of coincident waveforms from the liquid scintillator and from the Cf-252 ion chamber. The timing is represented by the dot and the star, respectively. The difference between these two times is used to determine the incident neutron energy. In the example of Fig. 3, the difference in times is 40.4 nsec and the corresponding incident neutron energy is 3.2 MeV.

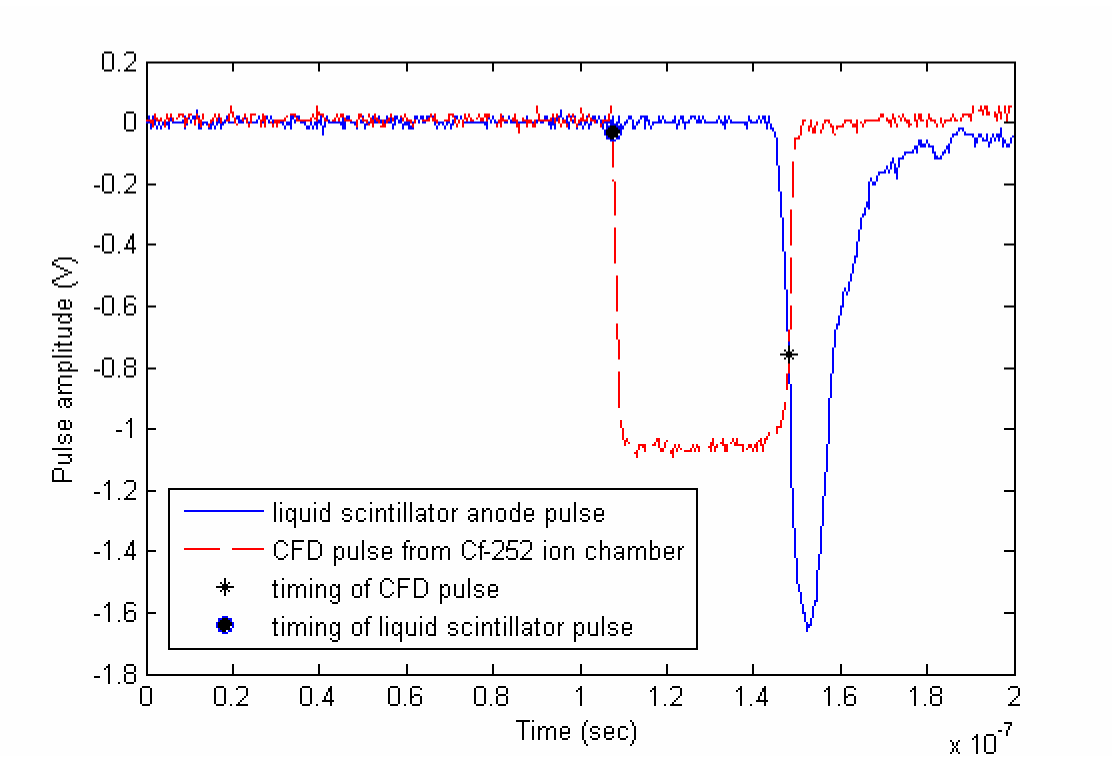

Fig. 3. Example of measured waveforms from liquid scintillator (solid line) and CFD pulse from the Cf-252 ion chamber (broken line). The timing of the rise times of the two pulses is also shown.

The program stores many waveforms of the type shown in Fig. 3. In addition to the total waveforms, which are typically stored with resolution of 0.4 nsec (500 points in this example), the neutron pulse heights (minimum values of the waveform) and their total integral are stored in separate matrices.

Figures 4 and 5 show measured average pulse waveforms neutrons from incident on the liquid scintillator.

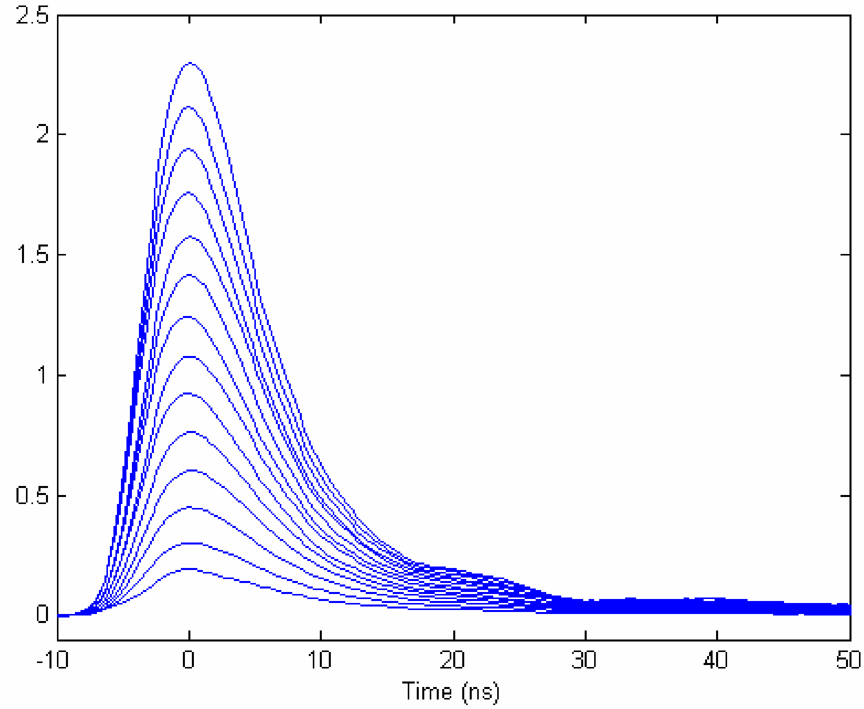

Fig. 4. Measured average pulse waveforms from neutrons obtained from the anode of the liquid scintillator.

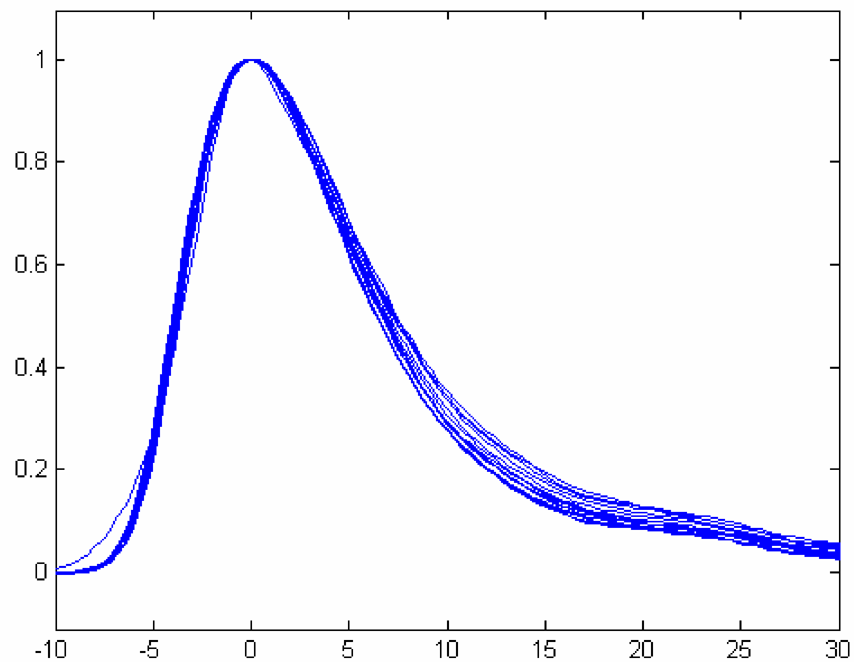

Fig. 5. Measured average pulse waveforms of Fig. 4 normalized to maximum value.

Figure 6 shows the result of a measurement of the detector response matrix, obtained by binning the measured pulse amplitudes (matrix *pampt*). Table 1 lists the main output variables of program *matrixmeas.m* and gives a short description of these variables.

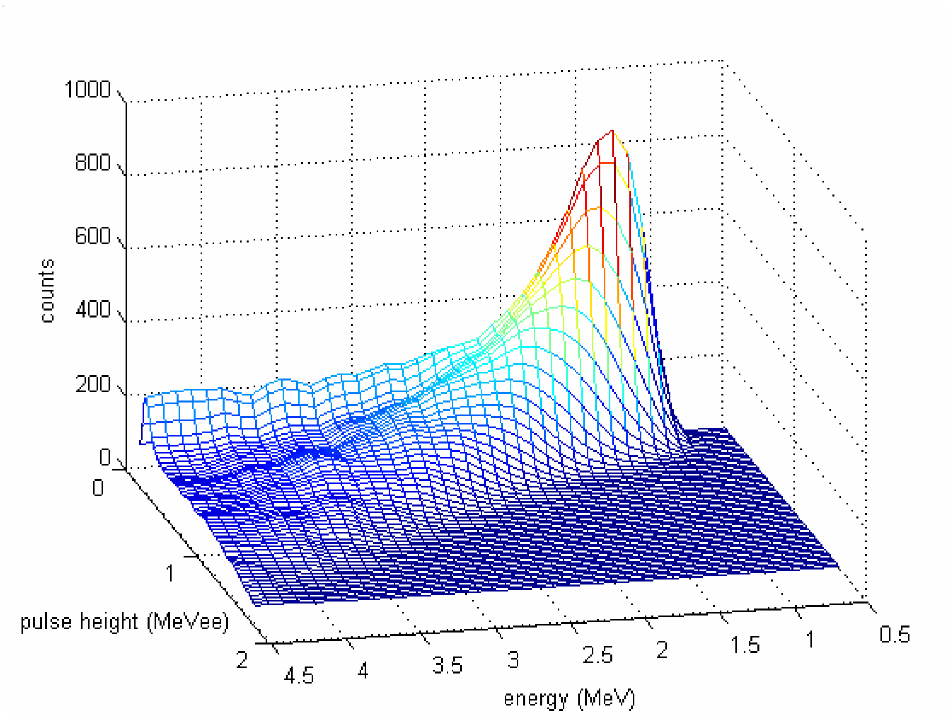

Fig. 6. Measured response matrix.

Table 1. Description of variables used in program *matrixmeas.m*

Variable name	Variable description
pampt	Matrix of pulse amplitudes - determined by taking the minimum of the pulse - rows = energy - columns = pulse heights
pareascint	Matrix of pulse areas - determined by taking the total area of the pulse - rows = energy - columns = pulse areas
pshapes	Matrix of pulse shapes - rows = time - columns = pulses
npbin	Row of number of pulses collected per energy bin

5. MEASURED AND SIMULATED SENSITIVITY FUNCTIONS

Monte Carlo simulations of detector response functions are often useful as a complement to experimental data. A simulated response matrix with the same energy and pulse height bins as experimental data was generated by the MCNP-PoliMi code.[4] Figure 7 shows the detector's sensitivity functions for a few pulse heights with neutron energy bins of 0.1 MeV. These functions represent the incident neutron spectrum that generates pulses of a given height in the detector. The sensitivity functions correspond to the columns of the response matrix shown in Fig. 6.

Good agreement was observed between simulated and experimental data for pulse heights up to 2.0 MeVee (MeV electron equivalent: a measure of the pulse height generated in the detector). However, it was not possible to perform meaningful comparisons in higher pulse height groups because of poor counting statistics obtained with the Cf-252 source. Preliminary results are encouraging, and it is expected that data from a new experiment will allow us to compare these functions over the entire energy range.

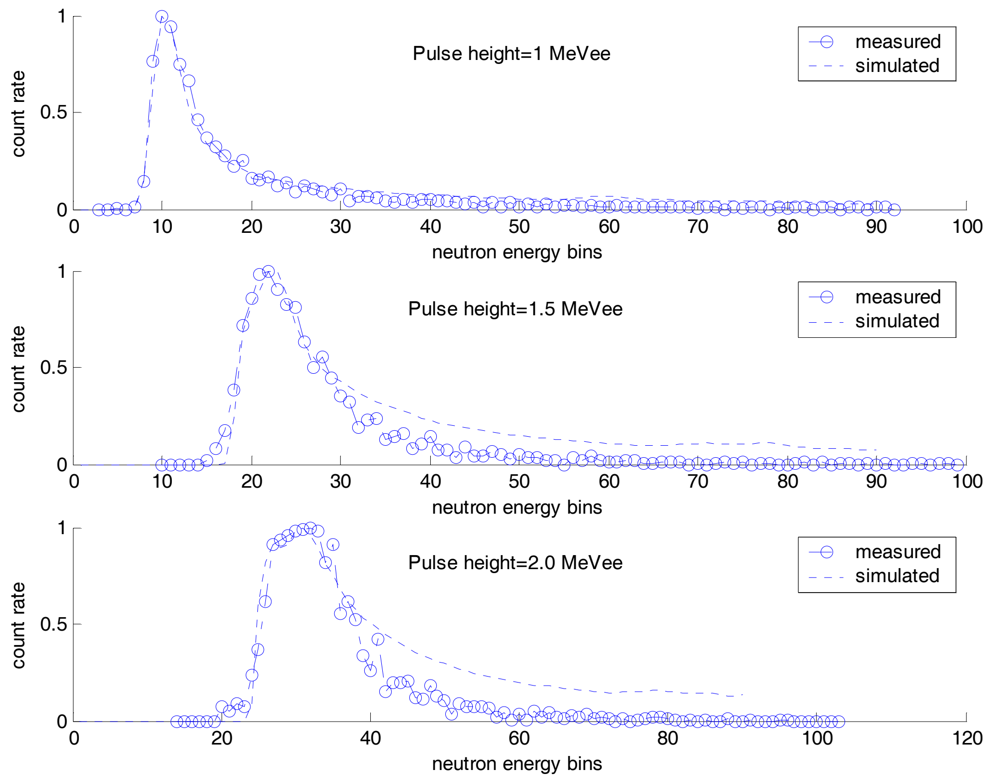

Fig. 7. Comparison between the simulated and measured sensitivity functions.

6. CONCLUSIONS AND FUTURE WORK

A Matlab code has been developed to measure the pulse shapes of liquid scintillator pulses obtained from neutrons emitted by a Cf-252 source. The time-of-flight method is used to determine the energy of the incident neutrons. The program can be used to analyze pulse shapes and to provide a measurement of the detector response matrix. This type of measurement can be used in place of measurements performed at linear accelerator facilities, when the neutron energies of interest are below a few MeV.

These measurements are currently being used to test an unfolding procedure that is based on the use of artificial neural networks to obtain estimates of the incident neutron spectrum on the basis of the neutron pulse height distributions.

7. ACKNOWLEDGMENT

This work was supported by the U.S. Department of Energy National Nuclear Security Administration Office of Nonproliferation Research Engineering NA-22.

8. REFERENCES

1. J. A. Mullens, J. D. Edwards, and S. A. Pozzi, "Analysis of Scintillator Pulse-Height for Nuclear Material Identification," Institute of Nuclear Materials Management 45th Annual Meeting, Orlando, FL, July 18–22, 2004.

2. J. T. Mihalczo, *Use of Cf-252 as a Randomly Pulsed Neutron Source for Prompt Neutron Decay Measurements*, Y-DR-41, UCC-ND, Y-12 National Security Complex, 1970.

3. L. Cartegni and S. A. Pozzi, *Simulation of the neutron response matrix for a liquid scintillator and spectrum unfolding*, ORNL TM 2004/315.

4. S. A. Pozzi, E. Padovani, and M. Marseguerra, "MCNP-PoliMi: A Monte Carlo Code for Correlation Measurements," *Nuclear Instruments and Methods in Physics Research A* **513,** 550–558 (2003).

APPENDIX A

Program "matrixmeas.m"

```
% Build a response matrix for scintillation detector
% using TDS 5104 with neutron source in Channel 1 and scintillator
% in Channel 2
% Signal from the neutron source is CFD type (cfdp)
% Signal from the scintillator is from the anode (scint)
% Set scope to trigger on the scintillator pulse (more rare event): 'stop'
% The signal from the neutron source is the 'start' pulse


global recordLen dt

% definition of constants used for the time to energy conversion
c=0.5227;      % time to energy conversion constant
dist=100;      % source-detector distance (cm)
dtimen=20e-9;  %tof for n classification

% definition of matrix neutron energy grid
enbin=0.5:0.1:10;         % energy bins
nen=length(enbin);
npulses=4000;             % maximum number of pulses recorded for each energy bin
pampt=zeros(npulses,nen); % matrix where pulse amplitudes are stored
npbin=zeros(1,nen);       % row of number of pulses registered
pareascint=zeros(npulses,nen); % matrix where pulse areas are stored
pshapes=[];                    % matrix where pulses shapes are stored

for i=1:1000  % adjust outer loop to vary number of pulses stored
   recordLen=400000;
   g=visa('tek','GPIB8::1::INSTR');  % assign Tekvisa info
   set(g,'InputBufferSize',recordLen*2);
   fopen(g)                          % open Tekvisa to communicate with scope
   idn=query(g,'*IDN?');
   fprintf(g,'HEADER OFF');
   fprintf(g,'DATA:ENCDG SRIBINARY');
   fprintf(g,'DATA:WIDTH 2');
   fprintf(g,'ACQUIRE:STATE OFF');
   fprintf(g,'ACQUIRE:MODE NORMALSAMPLE')

   for j=1:100

      d=acquire(g);  % this is equivalent to pressing 'single' on the scope
      strCh1=['ch1'];  % set channel number to be read for 'start' pulse
      [cfdp,tim1]=getpulse(g,strCh1); % acquire pulse and time from channel strCh1

      if min(cfdp)<-0.6  % if .true. we have a CFD pulse
         strCh2=['ch2']; % set channel number for scintillator pulse
         [scint,tim2]=getpulse(g,strCh2);
         if tim1~=tim2
            error('adjust timing in the two channels')
         end

         [mcfd,icfd]=min(diff(cfdp)); % determine position of leading edge of CFD pulse
         [pamp,ind]=min(scint);  % register pulse amplitude
         if ind>50
            [msc,isc]=min(diff(scint(ind-50:ind)));  % determine position of edge for scint pulse
            isc=isc+ind-50-1;
            tcfd=icfd*dt;  % time of CFD pulse
```

```
            tsc=isc*dt;    % time of scint pulse
            if tcfd<tsc
               dtime=tsc-tcfd; % register time difference
               if dtime>dtimen % check to see that the time difference is greater than the min time for neutrons
                  energyn=(c*dist^2./(dtime*1e9).^2);  % determine energy of neutron

                  if energyn<=enbin(nen)
                     ennbin=floor(energyn*10)-(enbin(1)*10-1); % determine energy bin number
                     % for this n energy
                     if ennbin >0

                        if npbin(ennbin)<npulses % determine if npulses pulses have already been stored in this energy
bin
                           pampt(npbin(ennbin)+1,ennbin)=pamp; % register pulse amplitude to matrix pampt
                           pareascint(npbin(ennbin)+1,ennbin)=sum(scint); % register pulse area to matrix pareascint
                           npbin(ennbin)=npbin(ennbin)+1;                 % update total number of pulses in this energy bin
                           pshapes=[pshapes scint];                       % register pulse shape to matrix pshapes

                           % insert debug plot commands here when
                           % needed.


                        elseif npbin==npulses  % determine if all energy bins contain npulses and if so, return
                           fclose(g);
                           delete(g);
                           return
                        end
                     end
                  end
               end
            end
         end
      end

   end

   fclose(g);
   delete(g);
end

return

% debug plot commands

plot(tim2,scint,'b')
hold on
plot(tim1,cfdp,'r')
plot(tim2(isc),scint(isc),'*r')
plot(tim1(icfd),cfdp(icfd),'ob')
plot(tim2(ind),pamp,'*k')
dtime
energyn
pause
clf
```

APPENDIX B

Function "acquire.m"

```
function d=acquire(g)


fprintf(g,'ACQUIRE:STOPAFTER SEQUENCE');
fprintf(g,'ACQUIRE:STATE RUN');
while query(g,'BUSY?','%s','%e'); end;
d=0;
```

APPENDIX C

Function "getpulse.m"

```
function [waveform,time]=getpulse(g,strCh)
global recordLen dt

fprintf(g,['DATA:SOURCE ' strCh]);
check=query(g,'DATA:SOURCE?');
horizLen=query(g,'HORIZONTAL:RECORD?','%s','%e');
fprintf(g,'DATA:START %d',1);
fprintf(g,'DATA:STOP %d',recordLen);
fprintf(g,'CURVE?');
dummy_string1=fscanf(g,'%s',2);
dummy_string2=fscanf(g,'%s',str2num(dummy_string1(2)));
recordLen2Transfer=min(recordLen, horizLen);
[waveform_raw,count]=fread(g,recordLen2Transfer,'int16');
dummy_string3=fscanf(g,'%s',1);
dt=query(g,'WFMOUTPRE:XINCR?','%s','%e');
ymult=query(g,'WFMOUTPRE:YMULT?','%s','%e');
yoff=query(g,'WFMOUTPRE:YOFF?','%s','%e');
yzero=query(g,'WFMOUTPRE:YZERO?','%s','%e');

% scale data
waveform=ymult*(waveform_raw-yoff)-yzero;
time=0:dt:dt*horizLen-dt;
```

INTERNAL DISTRIBUTION

1-2. S. Avdic
3 Z. W. Bell
4. C. S. Blessinger
5. N. A. Chaykin
6. J. J. Henkel
7. J. A. March-Leuba
8. J. T. Mihalczo
9. J. A. Mullens
10. J. S. Neal
11-12. S. A. Pozzi
13. T. Uckan
14. J. A. Williams
15. M. C. Wright
16. Central Research Library
17. ORNL Laboratory Records