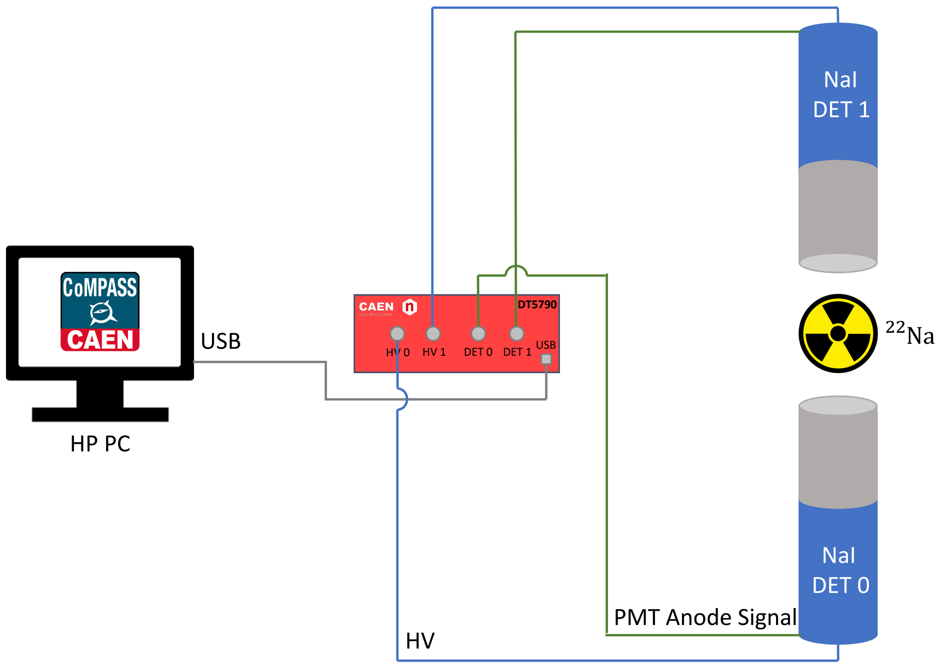
Viola, Jesse (LSA, Class of 2023) 2019 Dec 10

CAEN DT5790 ^{22}Na Gamma-Gamma Coincidence Experiment

General Gamma-Gamma Coincidence Setup



Software/Hardware

- Windows 10
- CAEN CoMPASS 1.3 and DPP-PSD
- CAEN DT5790 digitizer with USB driver

CoMPASS Settings

Acquisition Mode:

- Acquisition mode[i] – List only
- Timed run (s) – 10
- List saving – Save filtered data, CSV file format

Settings[ii]:

- Use all factory settings (see below for list of factory settings)

Time Selection:

- Correlation – Paired AND
- Correlation window (μs) – 0.1

[i] 'List only' does not allow waveforms to be visualized.
[ii] Settings are all equivalent for CH0 and CH1.

Sample Data File

```
TIMETAG;ENERGY;ENERGYSHORT;FLAGS
538540000;95;21;0x0
685136000;98;17;0x0
2218124000;161;23;0x0
2541228000;102;17;0x0
2672752000;337;46;0x0
…
```

^{22}Na Gamma-Gamma Coincidence Experiment Procedure

The setup for this experiment consisted of two NaI(TI) detectors connected to the CAEN DT5790 digitizer. The 1 µCi Na-22 source was placed between the two detectors at an equivalent distance from each detector. In the initial setup, the Na-22 sample was about 3 cm from each detector, and the detectors were 180° apart. Each 2 in. by 2 in. NaI scintillator, PMT and base was connected to a high-voltage output at approximately 1500 V [to match gains, Detector 1 (CH1) was set at approximately 100 V higher than Detector 0 (CH0)]. Using the CAEN CoMPASS software, a coincidence window was set at 0.1 µs, thus filtering out data points that were outside of the time window. The data file produced (.csv file) includes timestamps, energy long and short pulse integration, and flags. Visualization software written in Python (see below for programs) was then used to create histograms to represent the energy long pulse integration values. Using gating in the Python program, the energy peak can be separated from lower energy coincidences in order to have an accurate count (*N*) of high-energy gamma coincidence events. In this experiment, the energy peak can be calibrated to 511 keV, which is the energy of positron annihilation gamma rays released from the decay of Na-22. To confirm that the positron annihilation gamma rays are released 180° apart, the two detectors were moved to 45° apart. This change resulted in the predicted effect of decreasing the number of high-energy coincidence events (see Fig. 2. Left). Increasing the distance between the detectors also decreases the number of coincidence events, including the random coincidences (see Fig. 2. Right, for each detector 6 cm away from the source). The decrease in coincidence events, and radioactive intensity in general for a point source, can be described by the inverse-square law: $I \propto \frac{1}{r^2}$.

Figures:

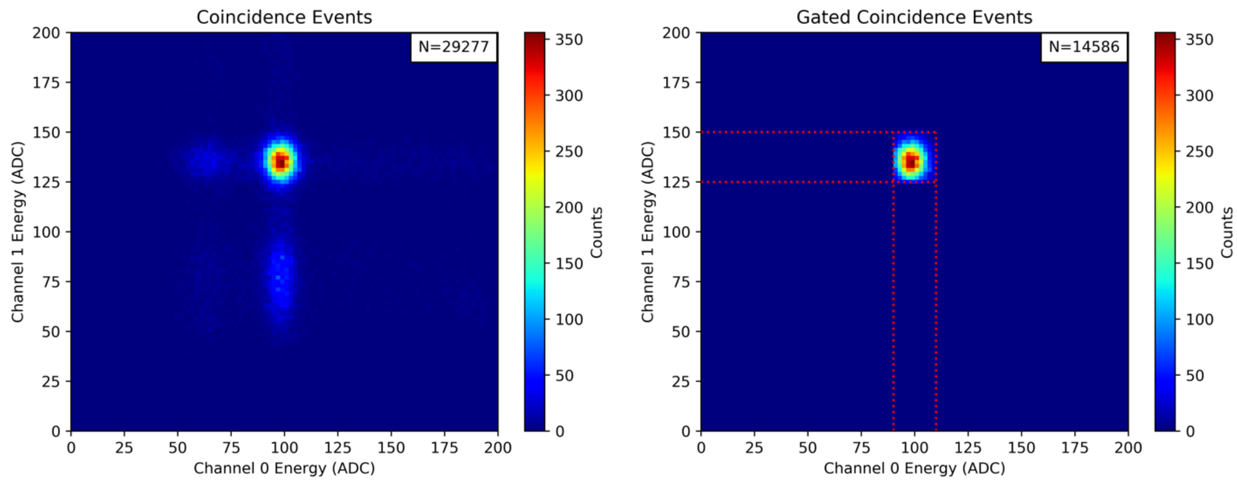


Fig. 1. Left: 2D histogram Na-22 gamma-gamma coincidence events with two detectors (CH0 and CH1) 180° apart in a 100 s interval. Each detector is placed about 3 cm from the Na-22 source. Right: Gated version of the Fig. 1. Left (90 < CH0 Energy < 110, 125 < CH1 Energy < 150). The number of events, *N*, is shown in the top right of each figure.

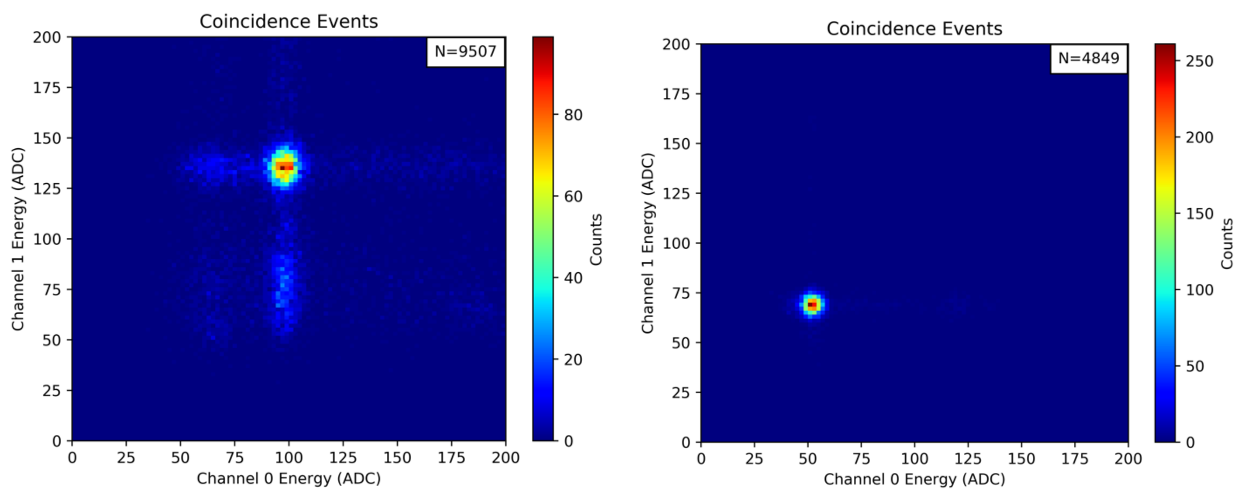


Fig. 2. Left: 2D histogram Na-22 gamma-gamma coincidence events with two detectors (CH0 and CH1) 45° apart in a 100 s interval. Each detector is placed about 3 cm from the Na-22 source. Right: 2D histogram Na-22 gamma-gamma coincidence events with two detectors (CH0 and CH1) 180° apart in a 100 s interval. Each detector is now placed about 9 cm from the Na-22 source, and attenuation is added.[iii] The number of events, *N*, is shown in the top right of each figure.

[iii] Channel (0 and 1) energy scale is approx. half of Fig. 2 Left channel (0 and 1) energy scale due to the use of a 50 Ω splitter.

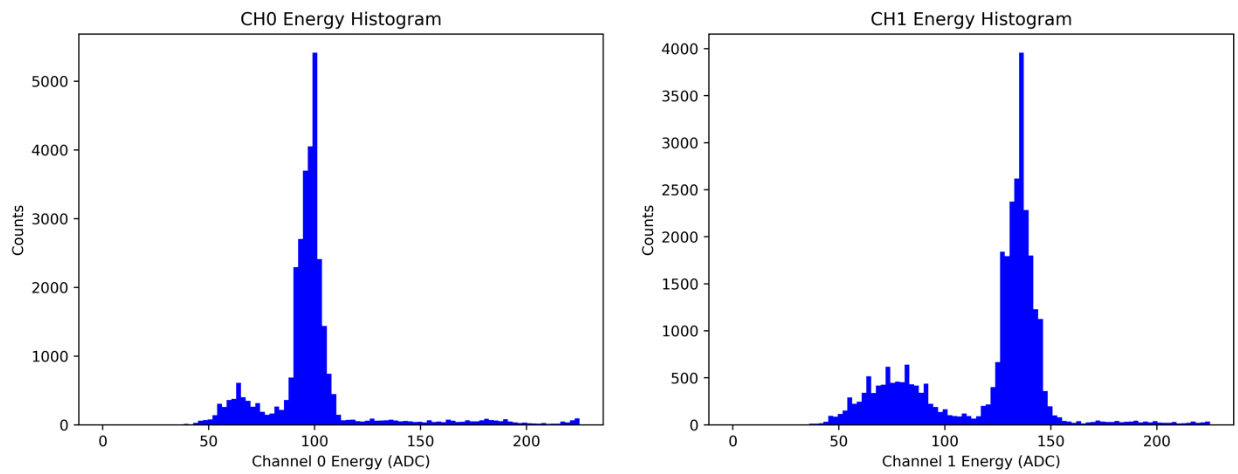


Fig. 3. Uncalibrated energy spectra of Na-22 gamma rays for each detector (CH0 and CH1) 180° apart in a 100 s interval. Each detector is placed about 3 cm from the Na-22 source (see Fig. 1).

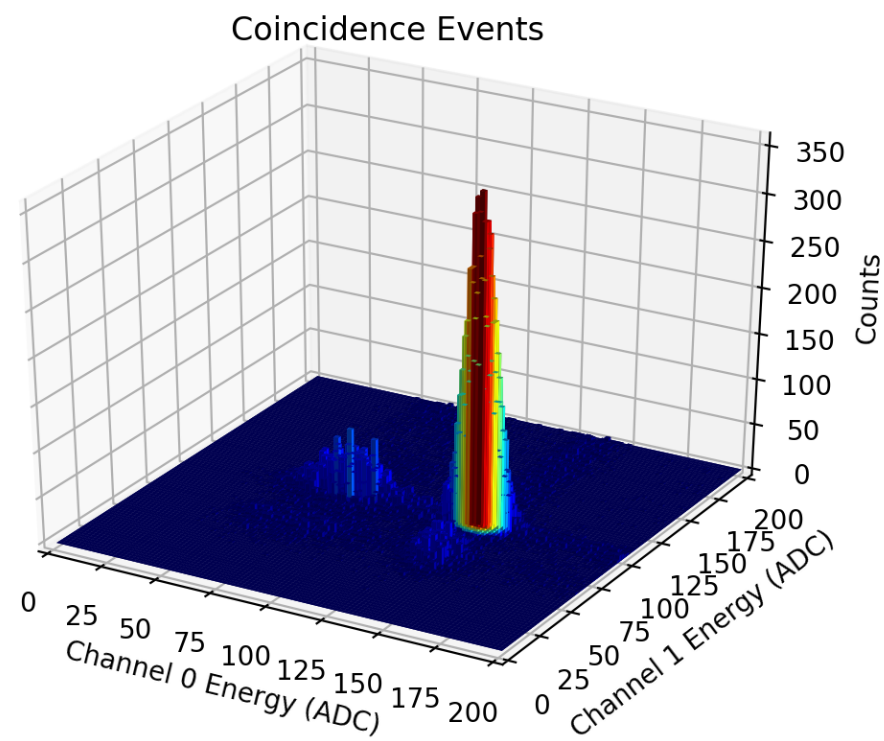


Fig. 4. 3D histogram of Na-22 gamma-gamma coincidence events with two detectors (CH0 and CH1) 180° apart in a 100 s interval. Each detector is placed about 3 cm from the Na-22 source (see Fig. 1).

Python Programs

1D Energy Histogram:

```
#Initialization
import matplotlib.pyplot as plt
import numpy as np
import csv

#Array
x = []

#.csv Reader
with open('FILE_NAME.csv') as csvfile:
    csv_reader = csv.reader(csvfile, delimiter=';')
    for lines in csv_reader:
      x.append(lines[1])

#Delete Label
del x[0]

#Create Float Array
x1 = np.array(x)
x1 = np.asfarray(x1,float)

#1D Histogram (change range to fit data)
plt.title('Energy Histogram')
plt.xlabel('Channel 0 Energy (ADC)')
plt.ylabel('Counts')
plt.hist(x1, bins=100, range=[0, 225], facecolor='blue')

#Save Histogram
plt.savefig('CH#_Energy_Histogram_ADC.png', dpi=500, bbox_inches='tight')

plt.show()
```

2D & 3D Energy Histograms:

```
#Initialization
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d
import numpy as np
import pandas as pd
import csv

#Gate Input (keV) (X0 < X1, Y0 < Y1) (change gating to fit data)
X0 = 0
X1 = 225
Y0 = 0
Y1 = 225

#Calibration Input (keV) (change calibration to fit data)
E = 1.0

#Bins (Resolution)
B = 100

#Arrays
x = []
y = []

#CH0 .csv Reader
with open('FILE_NAME0.csv') as csvfile:
    csv_reader = csv.reader(csvfile, delimiter=';')
    for lines in csv_reader:
      x.append(lines[1])

#CH1 .csv Reader
with open('FILE_NAME1.csv') as csvfile:
    csv_reader = csv.reader(csvfile, delimiter=';')
    for lines in csv_reader:
      y.append(lines[1])

#Delete Label
del x[0]
del y[0]

#Create Float Arrays
x1 = np.array(x)
x1 = np.asfarray(x1, float)

y1 = np.array(y)
y1 = np.asfarray(y1, float)

#Calibration
x2 = x1 * E
y2 = y1 * E

#Create Data Frames
```

```
colsx = ["Energy 0"]
dx2 = pd.DataFrame(x2, columns=colsx)

colsy = ["Energy 1"]
dy2 = pd.DataFrame(y2, columns=colsy)

#Merge Data Frames
DF = pd.concat([dx2, dy2], axis=1)

#Gated Data Frames
DF1 = DF[DF["Energy 0"] >= X0]
DF2 = DF1[DF1["Energy 0"] <= X1]

DF3 = DF2[DF2["Energy 1"] >= Y0]
DF4 = DF3[DF3["Energy 1"] <= Y1]

#Split Data Frame
x4 = DF4[["Energy 0"]]
y4 = DF4[["Energy 1"]]

#Create Arrays
x5 = x4.values
x5 = x5.transpose()
x5 = x5.flatten()
x6 = np.array(x5)
x6 = np.asfarray(x6, float)

y5 = y4.values
y5 = y5.transpose()
y5 = y5.flatten()
y6 = np.array(y5)
y6 = np.asfarray(y6, float)

#Events
A = len(x6)

#Graph Limits
G = Y1 / (225 * E)
H = X1 / (225 * E)

#2D Histogram
plt.title('Coincidence Events')
plt.xlabel('Channel 0 Energy (keV)')
plt.ylabel('Channel 1 Energy (keV)')
plt.hist2d(x2, y2, bins=B, range=[[0, 225 * E], [0, 225 * E]],
cmap=plt.cm.jet)
plt.colorbar(label='Counts')

#Save 2D Histogram
plt.savefig('Coincidence_Events_keV.png', dpi=500, bbox_inches='tight')

#Gated 2D Histogram
plt.title('Gated Coincidence Events')
```

```
plt.xlabel('Channel 0 Energy (keV)')
plt.ylabel('Channel 1 Energy (keV)')
plt.axvline(x=X0, c='r', ymax=G, linestyle=':')
plt.axvline(x=X1, c='r', ymax=G, linestyle=':')
plt.axhline(y=Y0, c='r', xmax=H, linestyle=':')
plt.axhline(y=Y1, c='r', xmax=H, linestyle=':')
plt.hist2d(x6, y6, bins=B, range=[[0, 225 * E], [0, 225 * E]],
cmap=plt.cm.jet)
plt.colorbar(label='Counts')

#Event Text
plt.text(0.98,.98,"N={}".format(A), bbox={'facecolor':'w','pad':5},
         ha="right", va="top", transform=plt.gca().transAxes )

#Save Gated 2D Histogram
plt.savefig('Gated_Coincidence_Events_keV.png', dpi=500, bbox_inches='tight')

#Gated 3D Histogram
xAmplitudes = x6
yAmplitudes = y6

x7 = np.array(xAmplitudes)
y7 = np.array(yAmplitudes)

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

hist, xedges, yedges = np.histogram2d(x7, y7, bins=(B, B), range=[[0, 225 *
E], [0, 225 * E]])
xpos, ypos = np.meshgrid(xedges[:-1]+xedges[1:], yedges[:-1]+yedges[1:])

xpos = xpos.flatten()/2.
ypos = ypos.flatten()/2.
zpos = np.zeros_like (xpos)

dx = xedges [1] - xedges [0]
dy = yedges [1] - yedges [0]
dz = hist.flatten()

cmap = plt.cm.jet
max_height = np.max(dz)
min_height = np.min(dz)

rgba = [cmap((k-min_height)/max_height) for k in dz]

ax.bar3d(xpos, ypos, zpos, dx, dy, dz, color=rgba, zsort='average')
plt.title('Gated Coincidence Events')
plt.xlabel('Channel 0 Energy (keV)')
plt.ylabel('Channel 1 Energy (keV)')
ax.set_zlabel('Counts')
ax.set_xlim3d(0, 225 * E)
ax.set_ylim3d(0, 225 * E)
```

```python
#Save Gated 3D Histogram
plt.savefig('3D_Gated_Coincidence_Events_keV.png', dpi=500,
bbox_inches='tight')

plt.show()
```

CoMPASS Factory Settings[iv]

Input
- Record length – 992 ns
- Pre-trigger – 100 ns
- Polarity – negative
- Ns baseline – 128 samples
- Fixed BLR – 0
- DC Offset – 20.0 %
- Input dynamic – 2.0 Vpp

Discriminator
- Threshold – 200 lsb
- Trigger holdoff – 496 ns

QDC
- Energy coarse gain – 160 fC/LSB
- Long gate – 300 ns
- Short gate – 80 ns
- Pre-gate – 48 ns

Spectra
- Energy N channels – 4096
- PSD N channels – 4096
- Time intervals N channels – 8192
- Time intervals Tmin – 0.000 μs
- Time intervals Tmax – 1000.000 μs
- Start/stop Δt Tmin – 10000 ns
- Start/stop Δt Tmax – 10000 ns
- 2D Energy N channels – 512
- 2D PSD N channels – 512

Rejections
- PSD low cut – 0.000
- PSD high cut – 0.000
- Time intervals low cut – 0 ns
- Time intervals high cut – 0 ns

Energy Calibration
- SET CALIBRATION VALUES

Sync/Trig
- Start mode – Software (asynchronous)
- TRIG OUT/GPO mode – Level 0
- Start delay – 0 ns
- Channel time offset – 0.000 ns

Miscellaneous
- Label – CH
- FPIO type – NIM
- Rate optimization – 1023

[iv] Not including settings changed as needed in ^{22}Na gamma-gamma coincidence experiment