ORNL/TM-2002/236

Initial Evaluation of a New Electromechanical Cooler for Safeguards Applications

September 2002

R. L. Coleman, J. S. Bogard and M. E. Murray

ORNL/TM-2002/236

INITIAL EVALUATION OF A NEW ELECTROMECHANICAL COOLER FOR SAFEGUARDS APPLICATIONS

R. L. Coleman, J. S. Bogard and M. E. Murray

September 2002

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

ACKNOWLEDGMENTS

The authors wish to acknowledge the generous support of Dr. David P. Spears, Radiation Detection Technologies Program Manager, at the U.S. Department of Energy. Daniel L. Upp, ORTEC® Vice President, was our liaison with PerkinElmer™ Instruments/ORTEC® Products and provided assistance during X-Cooler™ and germanium detector system procurement, installation and troubleshooting phases of the project.

ACRONYMS

CV	Coefficient of variation
DOE	U.S. Department of Energy
FWHM	Full width at half-maximum
FWTM	Full width at tenth-maximum
FW*x*M	Full width at *x*-maximum
FW(1/25)M	Full width at 1/25th maximum
HPGe	High-purity germanium
LN_2	Liquid nitrogen
NIST	National Institute of Standards and Technology
ROI	Region of interest
t	Time

EXECUTIVE SUMMARY

The ORTEC® X-Cooler™, because of its potential for unattended or covert use, for use in remote or inaccessible locations, and other advantages over the use of liquid nitrogen (LN_2), merits consideration as a viable alternative to LN_2 for cooling high-purity germanium (HPGe) detectors in many applications. Any decision to use alternative cooling for safeguards monitoring applications must also consider effects on performance characteristics such as total photopeak area, peak area distribution, full width at *x*-maximum (FW*x*M), FW*x*M distribution, and susceptibility to background artifacts, particularly at lower photon energies important in identifying and quantifying special nuclear materials. These parameters bear directly on data quality and usefulness for the intended purpose. This report investigates the performance characteristics of the X-Cooler™ both in higher-photon-energy regions (662 kev, 1173 kev, and 1332 keV) typical of (-emitting radionuclides and from a low-energy region (59.5 keV) more typical of characteristic X-rays of uranium and the transuranic elements.

Appropriate decisions relating to the use of the X-Cooler™ in place of LN_2, and effective use of the X-Cooler™ when it is the chosen alternative, depend on a complete understanding of its performance characteristics in relation to the alternative. Results reported here are germane to three important aspects of the performance of the X-Cooler™ used to cool HPGe detectors as compared with the benchmark LN_2: (1) stability and reproducibility, (2) precision and (3) limits of detection in HPGe systems cooled by the two alternatives. Stability and reproducibility are reflected through stochastic (random) and non-stochastic (data trends and anomalies) variability when operating under unchanging conditions for long periods of time. Precision is determined by photopeak width, since narrow, well-defined peaks allow more accurate identification of radionuclides, especially in mixtures. Limits of detection are directly related to background levels and variability, so that low backgrounds with minimal variability provide the best limits of detection.

Use of the X-Cooler™ compared more favorably with LN_2 cooling in the higher-energy ROIs than at lower energies. Comparison of photopeak areas in the monitored regions of interest shows that mean peak areas are nearly the same using the X-Cooler™ and LN_2 at 1332 keV and 1173 keV (although slightly lower using the X-Cooler™), but are significantly lower for X-Cooler™ cooling at 662 keV and 59.5 keV. The distribution of peak areas in a given ROI, expressed as the coefficient of variation, is almost twice as great using the X-Cooler™ than when using LN_2 in all four of the ROIs investigated. Smaller photopeak areas and greater variability would translate into lower detection efficiencies and greater variability in estimates of quantities of monitored radioactive material.

Peak widths and their distributions were greater with the X-Cooler™ than with LN_2, with the difference between the two cooling methods become greater at lower photon energies. The mean FWHM with the X-Cooler™ was about 2.4% greater than that using LN_2 at 1332 keV, increasing to around 17% greater at 59.5 keV. The corresponding coefficients of variation ranged from 65% greater for the X-Cooler at 1332 keV to 112% greater at 59.5 keV. The data for FWTM and FW(1/25)M indicate similar trends for these performance measures. The resolution of complex spectra, especially at low energies, becomes more difficult as the photopeaks widen.

Background increased significantly with one of the the X-Coolers™ after about 2.5 days (live time) of operation. Such behavior during safeguards monitoring would result in a corresponding change in the limit of detection, perhaps without of the knowledge of the operator. Microphonics transferred to the detector through the X-Cooler™ coupling is suspected to have been the cause of this particular increase in background. This speculation is supported by the absence of simultaneously increased background levels in the higher-energy ROIs, but the actual cause was not systematically determined.

1. INTRODUCTION

The use of liquid nitrogen (LN_2) constitutes the current state of the art in cryogenic cooling for high-purity germanium (HPGe) detectors, which are widely used for (-ray and characteristic X-ray spectroscopy because of their excellent energy discrimination. Use of LN_2 requires a liquid nitrogen supply, cumbersome storage tanks and plumbing, and the frequent attention of personnel to be sure that nitrogen levels are sufficient to maintain the detectors at a sufficiently low operating temperature. Safety hazards also are associated with the use of LN_2, both because of the potential for severe frostbite on exposure to skin and because it displaces ambient oxygen when it evaporates in closed spaces.

Existing electromechanical coolers have, until now, been more expensive to procure and maintain than LN_2 systems. Performance and reliability have also been serious issues because of microphonic degradation of photon energy peak resolution and cooler failures due to compressor oil becoming entrained in the refrigerant.

This report describes the results of tests of a new HPGe detector cooling technology, the PerkinElmer ORTEC® Products X-Cooler™ that, according to the manufacturer, significantly reduces the lifetime cost of the cooling system without degradation of the output signal[1,2]. The manufacturer claims to have overcome cost, performance and reliability problems of older-generation electromechanical coolers, but the product has no significant history of use, and this project is the first independent evaluation of its performance for safeguard applications.

Total cost savings for the DOE and other agencies that use HPGe systems extensively for safeguards monitoring is expected to be quite significant if the new electromechanical cooler technology is shown to be reliable and if performance characteristics indicate its usefulness for this application. The technology also promises to make HPGe monitoring, characterization and detection available for unattended or covert operation and in remote or inaccessible locations where the unavailability of LN_2 and signal degradation from existing mechanical coolers prevent its use at the present time.

2. EXPERIMENTAL APPROACH

The performance of spectrometry systems operated with HPGe detectors cooled using LN_2 was compared with the performance of the same systems using the PerkinElmer ORTEC® Products X-Cooler™ in place of LN_2. The HPGe detector tested was a medium-efficiency coaxial detector with an active volume of ~128 cm^3. Performance indicators consisted of spectroscopic parameters important in safeguards monitoring applications: spectral peak resolution and stability in several energy regions of interest (ROIs).

Testing was performed both in the presence of radioactive source standards and in their absence (background). Regions of interest chosen for evaluation were 58-61 keV, 659-664 keV, 1170-1176 keV, and 1329-1336 keV, corresponding to principal photon emissions of the radioactive source standards.

2.1 EQUIPMENT

An ORTEC model SGD-GEM-25175-P-S high-purity germanium (HPGe) coaxial photon detector (serial number 40-TP21490A), with an active volume of ~128 cm^3, was operated at the recommended +3600-V bias. High voltage and signal processing were provided by a DSP^{EC}® digital (-ray spectrometer* (serial no. 421).

*The DSP^{EC}® package includes a high-voltage supply and provides analog-to-digital detector signal conversion, digital signal processing and storage, and data transfer by ethernet connection to a personal computer.

2.2 RADIOACTIVE SOURCE STANDARDS

Radioactive source standards were fabricated from a mixture of radionuclides in aqueous solution that were obtained from Analytics (Atlanta, Georgia) and were traceable to the National Institute of Standards and Technology (NIST). The 5.51137 g of 4-**M** HCl solution included the radionuclides with principal photopeaks used in this research having total photon fluence rates as indicated in the table below. Individual standards were prepared from the standard solution by evaporation of a weighed aliquot onto an 0.0175-inch-

Table 1. Concentrations of radionuclides with principal emissions used for spectral stability determinations

Radionuclide	Photopeak Energy (keV)	Half-life (y)	(-Photon Fluence Rate[a] (s^{-1})
^{241}Am	59.54	432	1980
^{137}Cs	661.66	30.0	1861
^{60}Co	1173.24	5.27	3460
^{60}Co	1332.5	5.27	3467

[a]Total uncertainty in the photon fluence rate is #5%; the fluence rate was determined on 01 October 2000.

thick (0.444-mm) aluminum planchette. Three individual standards were fabricated using 0.5538 g, 0.2956 g and 0.2988 g of standard solution, respectively.

2.3 DATA COLLECTION

Data collection and analysis methods were automated for the majority of the research. Digital multi-channel analyzers were controlled directly using scripts within ORTEC® GammaVision™ software, while analysis results were extracted and stored using specialized code written in C++. Figure 1 shows a simplified overview of the logic used for the automation sequence.

Each detector was first initiated by clearing the analog-to-digital converter (ADC) memory and setting the acquisition parameters such as ‘acquire time’. A spectrum collection cycle would then begin and, once completed, the entire spectrum would be transferred from the ADC to the computer for manipulation. The spectrum would then be analyzed for peaks specific to the study followed by fine-tuning adjustments to the calibration parameters. Data would be extracted and stored into a comma-delimited file using a format similar to the following:

```
Detector name, date, time, livetime(sec), TOTAL, peak data, ROI data <new line>

Where,
  TOTAL        = total number of counts in spectrum;
  Peak data    = net counts, peak bkg, fwhm, fw1/10m and fw1/25m; and
  ROI data     = total counts in pre-selected regions of interest.
```

Once initiated, this algorithm was typically allowed to run for extended periods of time without need for extensive manual intervention. Appendix A contains the C++ source code written specifically for extracting and storing spectral information from the DSP^{EC}® and GammaVision™ files. Libraries marketed by the hardware manufacturer were found to be inadequate for the tasks needed in support of this research hence the reason for developing task-specific code. Note that this code does not require the use of any specialized libraries such as those sold by the hardware manufacturer.

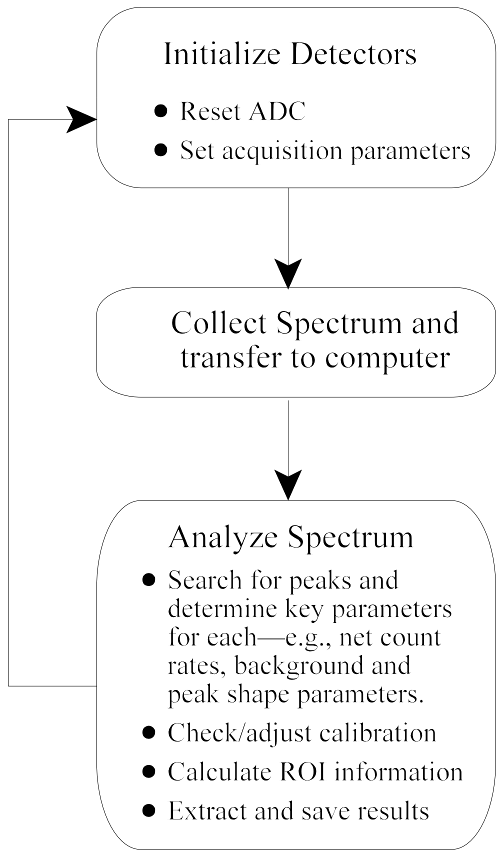


Fig. 1. Logic sequence for automated data collection.

3. RESULTS

Recorded data were analyzed for information about the reproducibility and stability of the HPGe system for long periods of time ($t > 500$ h, detector live time), both when LN_2 cooled and when cooled using the X-Cooler™. Sample means and uncertainties were determined for total counts in the ROI (photopeak area), which are adjusted to account both for radioactive decay and detector dead time. Means and uncertainties were also determined for the full photopeak width at half-maximum (FWHM), tenth-maximum (FWTM) and 1/25th maximum [FW(1/25)M]. The coefficient of variation (CV), or ratio of standard deviation to the mean expressed as a percent, is reported here as a statistic for comparing data sets. The data were also examined for trends and anomalies.

3.1 DETECTOR PERFORMANCE MONITORING STANDARD SOURCES

The coaxial HPGe safeguards detector SG1 was used to collect photon spectral data from standard radioactive sources while cooled with LN_2 and again while cooled using the X-Cooler™. Performance measures are described and compared in the following sections.

3.1.1 LN_2 Cooled

The spectroscopy system utilizing detector SG1 and monitoring a standard radioactive source was operated for 602 h with LN_2 cooling. Summary statistics for peak areas (total counts) in the ROIs are provided in Table 2. Uncertainty (± 1 standard deviation) in the total counts for each photopeak was less than 1.3%

Table 2. Photopeak area statistics for detector SG-1 cooled with LN2 and monitoring standard source

ROI centroid (keV)	59.5	662	1173	1332
Number of trials	602	602	602	602
Mean (counts)	20209	11615	10634	9478
Maximum (counts)	21012	12266	10981	9869
Minimum (counts)	19381	11168	10025	9178
Standard deviation (counts)	223	146	124	107
CV	1.10%	1.26%	1.16%	1.13%

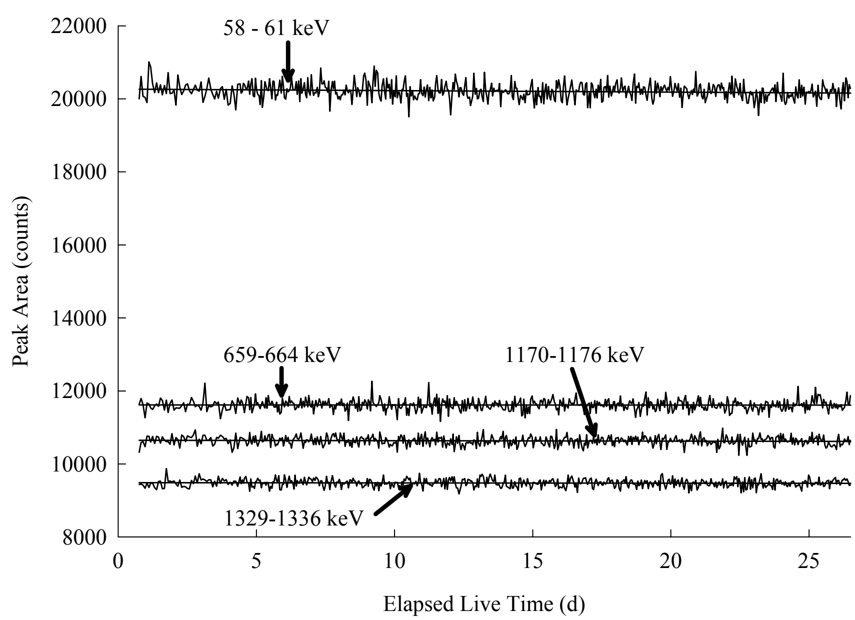


Fig. 2. Peak areas (counts) in regions of interest for safeguards detector SG1 cooled with LN_2 and monitoring a standard radioactive source.

over the test period. Maximum and minimum values were within 6% of the means. Figure 2 shows peak areas in the ROIs as a function of time. Linear regressions of the data (trend lines in the figure) show that the average detector response was stable over the long monitoring period with slopes of -4.07 d^{-1} (59.5 keV), -0.17 d^{-1} (662 keV), -1.15 d^{-1} (1173 keV), and -0.79 d^{-1} (1332 keV), corresponding to changes in average peak areas of the ROIs that are within -0.70% of the starting average over the 602-h counting interval. No periodic features were noted in the data.

Photopeak widths and related statistics for detector SG-1 monitoring standard sources with LN_2 cooling are shown in Tables 3 - 5 below. The ratio of FWHM to the peak centroid energy (Table 3) was about 1.3% at 59.5 keV and less than 0.2% at 662 keV, 1173 keV and 1332 keV. Coefficients of variation for all the peak widths was less than 5%. Tables 4 and 5 provide corresponding statistics for the full width at tenth maximum and full width at 1/25th maximum.

Table 3. Photopeak FWHM statistics for detector SG1 cooled with LN2 and monitoring standard source

ROI centroid (keV)	59.5	662	1173	1332
Number of trials	602	602	602	602
Mean (keV)	0.763	1.317	1.640	1.732[a]
Maximum (keV)	0.881	1.463	1.749	1.878
Minimum (keV)	0.544	1.106	1.420	1.459
Standard deviation (keV)	0.032	0.041	0.049	0.050
CV	4.17%	3.15%	2.99%	2.91%

[a]FWHM at 1332 keV claimed by the manufacturer is 1.75 keV.

Table 4. Photopeak FWTM statistics for detector SG1 cooled with LN2 and monitoring standard source

ROI centroid (keV)	59.5	662	1173	1332
Number of trials	602	602	602	602
Mean (keV)	1.305	2.323	2.931	3.097
Maximum (keV)	1.438	2.513	3.085	3.664
Minimum (keV)	1.095	2.109	2.524	3.077
Standard deviation (keV)	0.035	0.046	0.051	0.049
CV	2.65%	1.96%	1.74%	1.59%

Table 5. Photopeak FW(1/25)M statistics for detector SG1 cooled with LN2 and monitoring standard source

ROI centroid (keV)	59.5	662	1173	1332
Number of trials	602	602	602	602
Mean (keV)	1.587	2.745	3.464	3.668
Maximum (keV)	1.694	3.050	3.664	3.863
Minimum (keV)	1.346	2.505	3.077	3.451
Standard deviation (keV)	0.034	0.059	0.063	0.065
CV	2.17%	2.14%	1.82%	1.76%

3.1.2 Cooled with X-Cooler™

The spectroscopy system utilizing detector SG-1 and monitoring a standard radioactive source was operated for 1248 h (live time) with cooling provided by the X-Cooler™. Summary statistics for peak areas (total counts) in the ROIs are provided in Table 6. (Fewer than 1248 trials are reported in the table for 1173-keV and 1332-keV centroids because of some excluded anomalous data that were determined to have origins in the data collection and reporting software, and were not due to any factors related to the X-Cooler™.)

Table 6. Photopeak area statistics for detector SG1 cooled by the X-Cooler™ and monitoring standard source

ROI centroid (keV)	59.5	662	1173	1332
Number of trials	1248	1248	1246	1234
Mean (counts)	19778	11473	10583	9409
Maximum (counts)	21220	12250	11052	9783
Minimum (counts)	15681	9564	7498	7739
Standard deviation (counts)	397	198	222	180
CV	2.01%	1.73%	2.10%	1.91%

Uncertainty (± 1 standard deviation) in the total counts for each photopeak was between 1.7% and 2.1% of the mean over the test period. Maximum values were within about 7% of the respective means, but variations in the minimum values were much greater, ranging to almost 30% below the mean in the case of the 1173-keV centroid. Figure 3, which shows peak areas in the ROIs as a function of time, reveals that the relatively greater minimum values in Table 6 are the result of several data spikes (~1 h duration) and of one broad dip (~1 d) in total counts of each of the 4 ROIs. Most of the spikes and the dip are closely related in time. These features are unexplained and cannot necessarily be attributed to the X-Cooler™ performance. Examination of the figure also reveals a significant drop in counts in the 58- to 61-keV ROI at the end of the test period. This anomaly is also unexplained; there is no obvious corresponding falloff in counts in the other ROIs. Linear regressions of the data (trend lines in the figure) show that the average detector response was otherwise stable over the long monitoring period with slopes of 2.78 d^{-1} (59.5 keV), -0.98 d^{-1} (662 keV), -0.048 d^{-1}

(1173 keV), and -0.31 d^{-1} (1332 keV), corresponding to changes in average total counts in the ROIs within -0.45% to 0.75% of the starting average over the 1248-h counting interval. No periodic features were noted in the data.

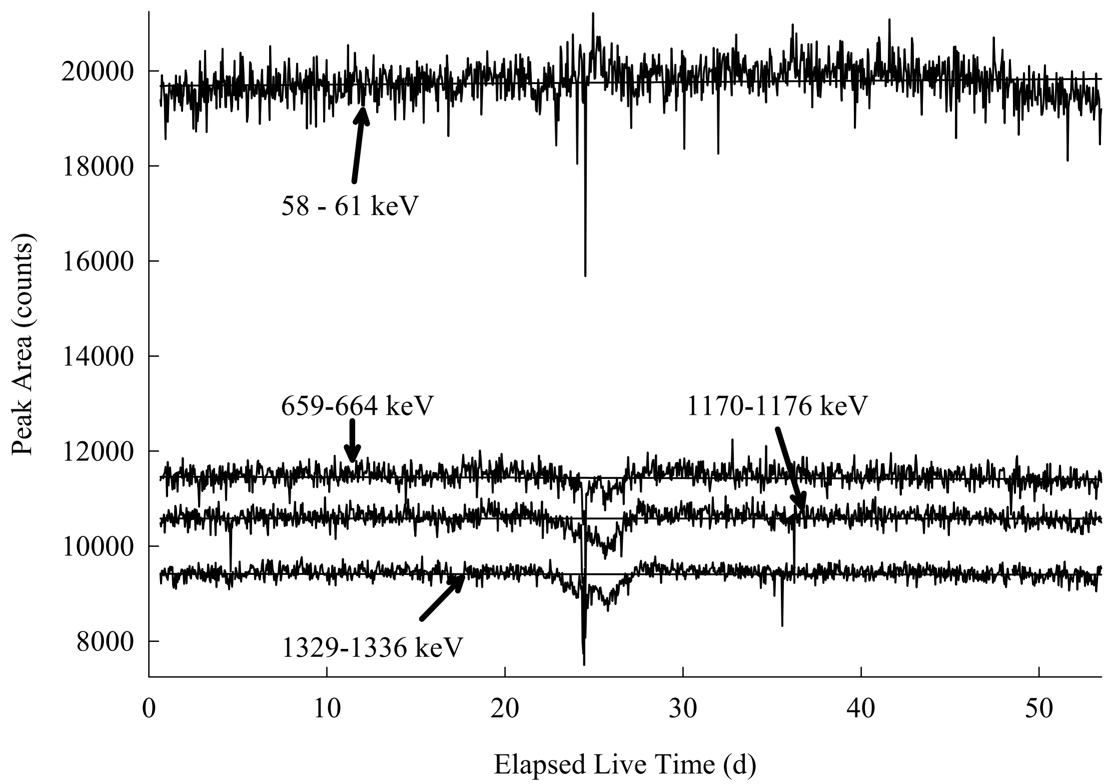


Fig. 3. Peak areas (counts) in regions of interest for safeguards detector SG1 cooled using the X-Cooler™ and monitoring a standard radioactive source.

Photopeak widths and related statistics for detector SG-1 monitoring standard sources and cooled with the X-Cooler™ are shown in Tables 7 - 9 below. The ratio of FWHM to the peak centroid energy was about 1.5% at 59.5 keV and about 0.2% or less at 662 keV, 1173 keV and 1332 keV. Coefficients of variation for the peak widths was between 5.7% (1332 keV) and about 11.4% (59.5 keV). Standard deviations in peak widths tended to be relatively constant with centroid energy. Mean peak widths tended to increase with energy, however, so that CVs decreased correspondingly. This behavior is in contrast to that of the detector cooled with LN2 (Tables 3 - 5), where both mean peak widths and their standard deviations increase with centroid energy and CVs stayed about the same or decreased slightly.

Table 7. Photopeak FWHM statistics for detector SG1 cooled with the X-Cooler™ and monitoring standard source

ROI centroid (keV)	59.5	662	1173	1332
Number of trials	1248	1248	1246	1234
Mean (keV)	0.921	1.412	1.718	1.798[a]
Maximum (keV)	1.664	2.115	2.408	2.347
Minimum (keV)	0.500	0.985	1.098	1.409
Standard deviation (keV)	0.105	0.103	0.106	0.103
CV	11.42%	7.33%	6.17%	5.72%

[a]FWHM at 1332 kev claimed by the manufacturer is 1.75 keV.

Table 8. Photopeak FWTM statistics for detector SG-1 cooled with the X-Cooler™ and monitoring standard source

ROI centroid (keV)	59.5	662	1173	1332
Number of trials	1248	1248	1246	1234
Mean (keV)	1.820	2.668	3.234	3.391
Maximum (keV)	3.109	4.338	5.964	6.117
Minimum (keV)	1.206	1.994	2.143	2.766
Standard deviation (keV)	0.171	0.160	0.172	0.162
CV	9.39%	6.00%	5.32%	4.78%

Table 9. Photopeak FW(1/25)M statistics for detector SG1 cooled with the X-Cooler™ and monitoring standard source

ROI centroid (keV)	59.5	662	1173	1332
Number of trials	1248	1248	1246	1234
Mean (keV)	2.385	3.290	3.964	4.134
Maximum (keV)	3.894	5.425	6.877	7.127
Minimum (keV)	1.535	2.499	2.666	3.347
Standard deviation (keV)	0.234	0.212	0.228	0.217
CV	9.81%	6.44%	5.75%	5.25%

3.1.3 Comparison of Key Statistics

Figure 4 compares net count distributions for safeguards detector SG1 cooled with LN_2 and with the X-Cooler™. Equal numbers of samples (a sample being a count with live-time duration of 1 h) were used for comparison in each ROI. The total dataset (602 samples) from Figure 2 was used as representative of data obtained when using LN_2 cooling (shaded bars). A selection of 301 samples from each side of, but excluding, the data anomalies that occurred at around 25 days in Figure 3 (602 samples total), was taken as representative of data in each ROI obtained when using the X-Cooler™ (solid bars). It is apparent from the figures that distributions from detectors cooled by the X-Cooler™ are broader, have lower means, and are more likely to contain outliers than those from detectors cooled by LN_2, and that the differences between the distributions are more pronounced at lower photon energies.

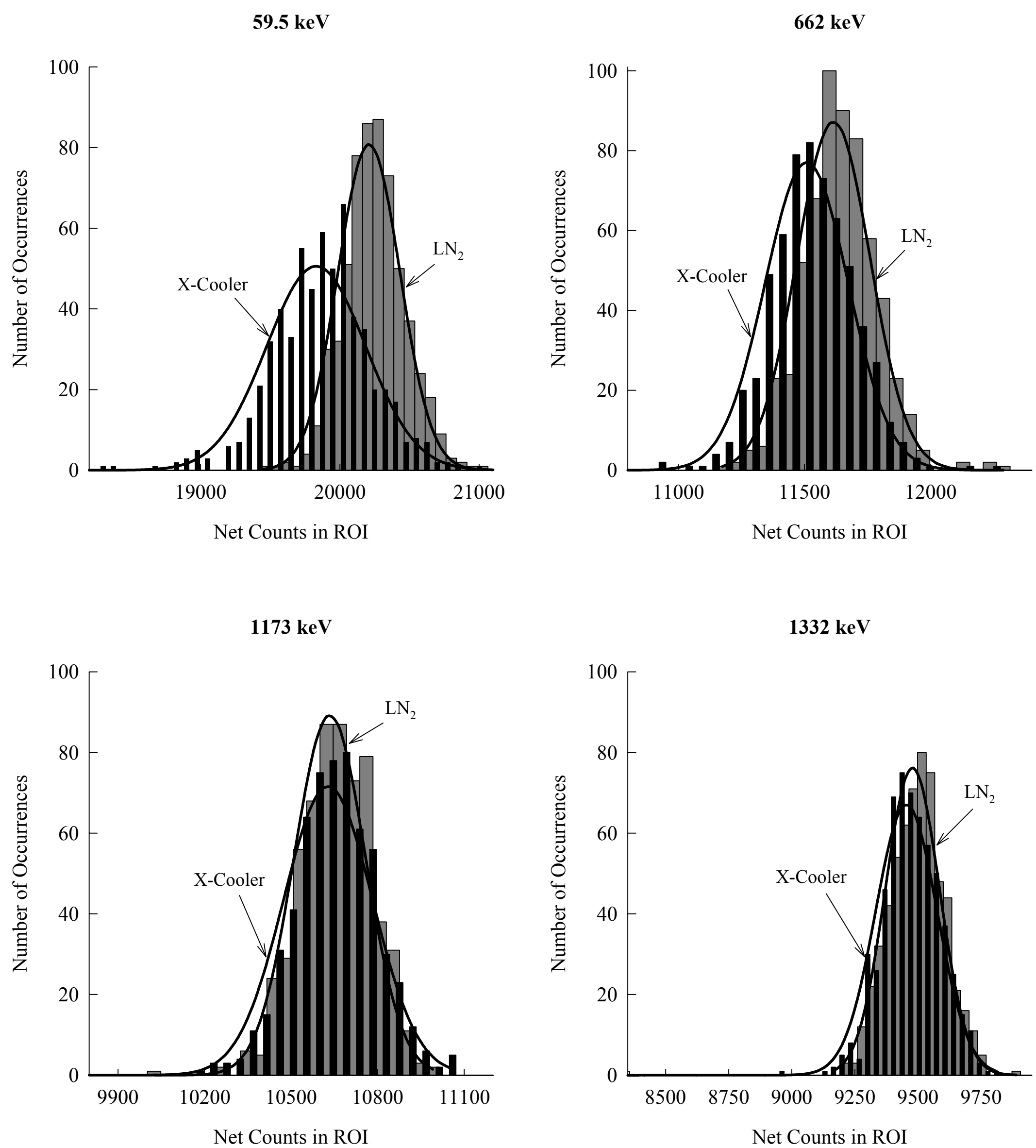


Fig. 4. Distributions of net counts in ROIs from detector SG1 monitoring standard sources and cooled by LN_2 (shaded bars) and the X-Cooler™ (solid bars).

Distributions of values of FWHM around the centroids of the four ROIs were also determined from the same data sets and are shown in Figure 5. Mean FWHM values were 2% - 17% higher for the same detector cooled with the X-Cooler™, and the standard deviations about the means were 70% - 148% larger than when cooled with LN_2.

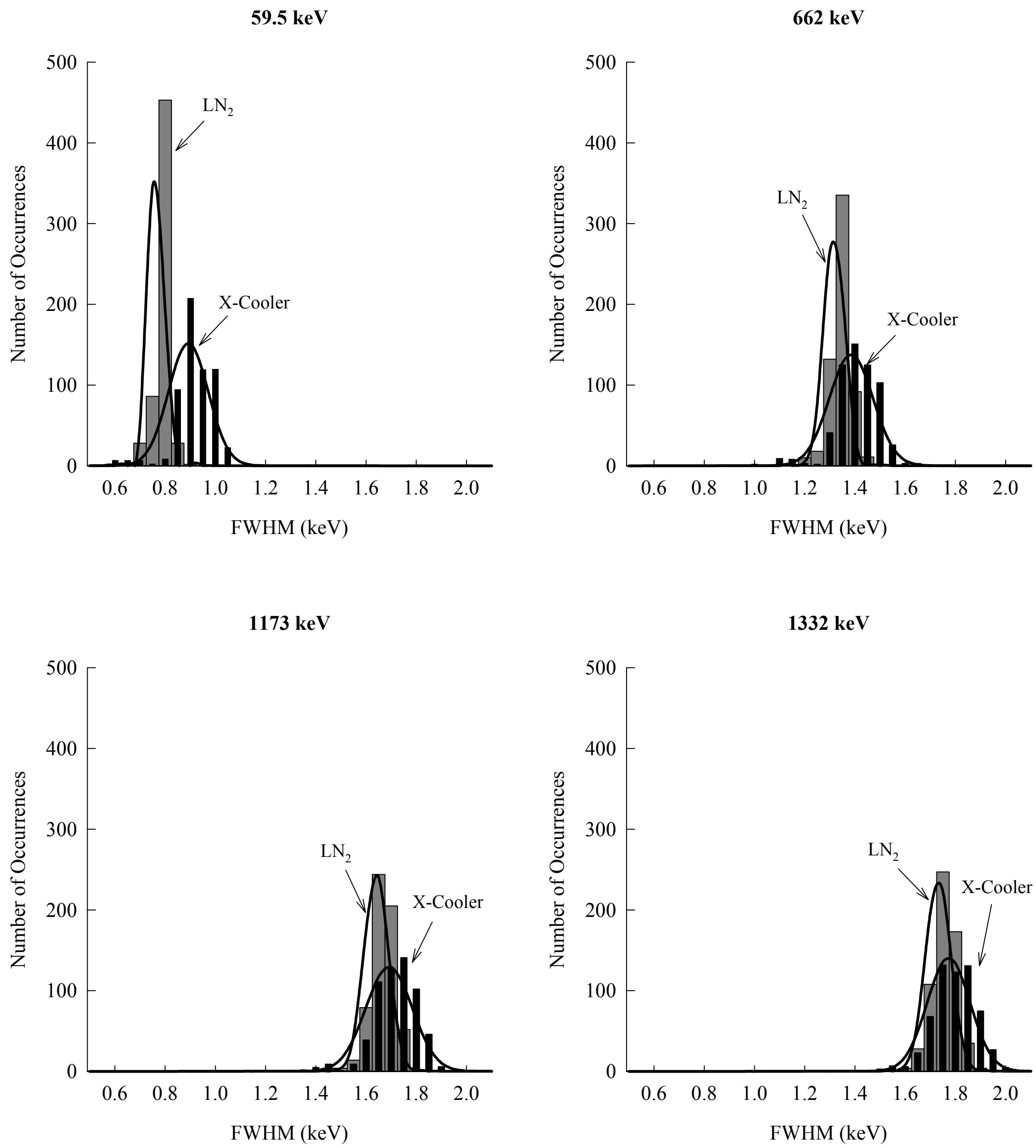


Fig. 5. Comparison of the distributions of full width at half-maximum for detector SG1 monitoring standard radiation sources and cooled with LN_2 and the X-Cooler™ in four regions of interest. Shaded bars correspond to the LN2 distributions; solid bars, to those of the X-Cooler™.

3.2 DETECTOR PERFORMANCE MONITORING BACKGROUND AND COOLED WITH X-COOLER™

The coaxial HPGe safeguards detector SG1 was used to collect background photon spectral data while cooled using the X-Cooler™. The spectroscopy system utilizing detector SG1 and monitoring background was operated for 187 h (live time). Summary statistics for peak areas (total counts) in the ROIs are provided in Table 10. Background counts were quite low, with no more than 15 counts in any trial, except in the low-energy (59- to 61-keV) region. Examination of Figure 6, which shows the background count rate in consecutive 1-h trials plotted with time, reveals that background response in the low-energy ROI was also low [(26 ± 9) h^{-1}] for about 2½ d, after which it gradually increased to around (325 ± 125) h^{-1}. Replacement of the X-Cooler™ with another identical X-Cooler™ eliminated this increase in background. The source of the anomalous behavior in the first X-Cooler™ was not determined.

Table 10. Photopeak area statistics for detector SG1 cooled by the X-Cooler™ and monitoring background

ROI (keV)	58 - 61	659 - 664	1170 - 1176	1329 - 1336
Number of trials	187	187	187	187
Mean (counts)	--	6.93	3.63	3.13
Maximum (counts)	603	15	9	9
Minimum (counts)	15	2	0	0
Standard deviation (counts)	--	2.38	1.87	1.79
CV	--	34.4%	51.6%	57.1%

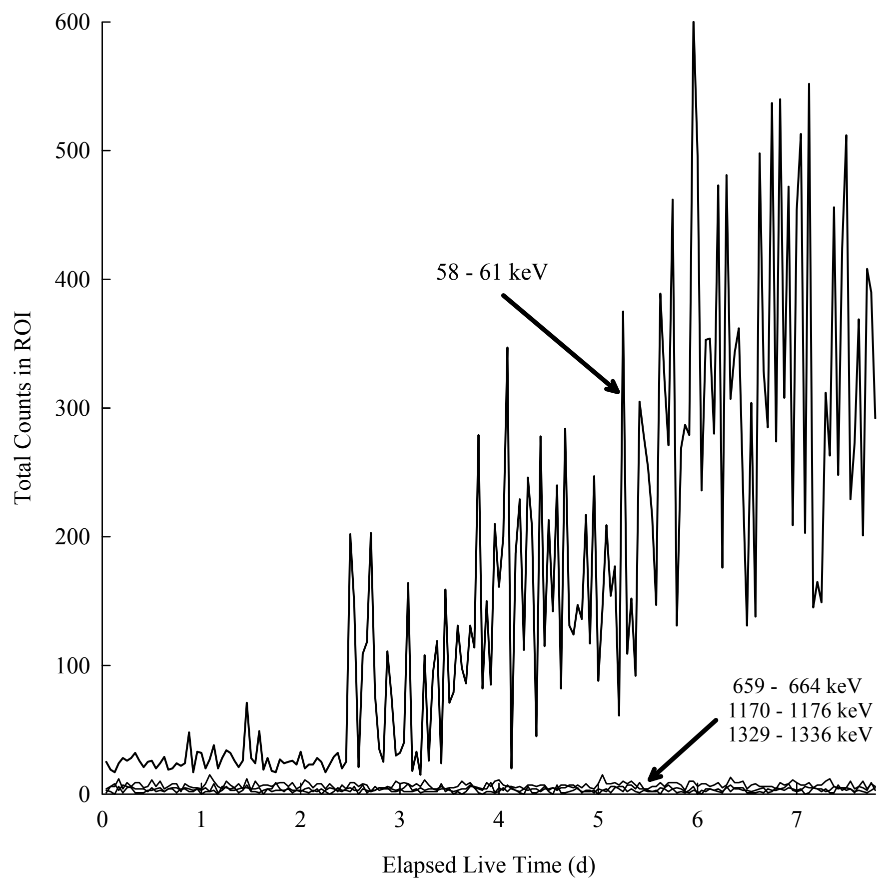


Fig. 6. Background counts from SG1 cooled by the X-Cooler™ recorded during consecutive 1-h live-time intervals in four energy regions of interest. Increased background in the 58 - 61 keV ROI appears to be due to the onset of microphonic noise from the X-Cooler™.

Figures 7 through 9 show plots of individual background rates with time and the count rate distributions in 187 trials for ROIs centered on 662 keV, 1173 keV and 1332 keV. The empirical distribution is reasonably represented by a Poisson distribution with the same mean in each case.

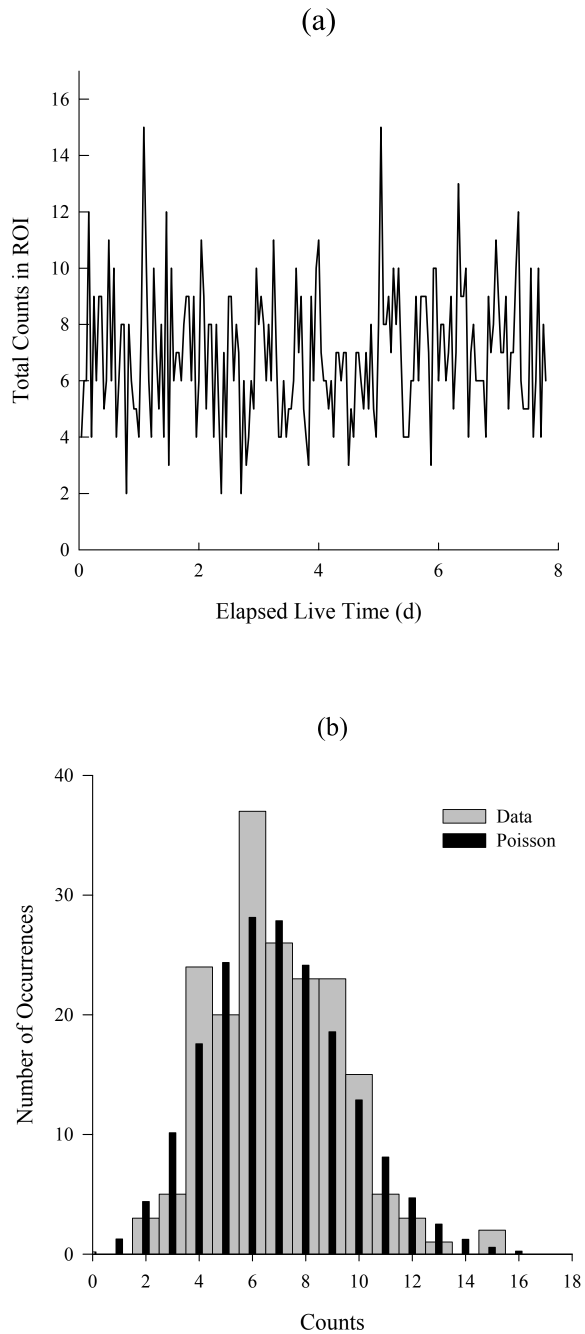


Fig. 7. Total background counts (a) in consecutive 1-h live-time intervals and (b) distribution of counts for SG1, cooled by the X-Cooler™, in the 659- to 664-keV ROI. Solid bars in (b) show the Poisson distribution having the same mean as the data.

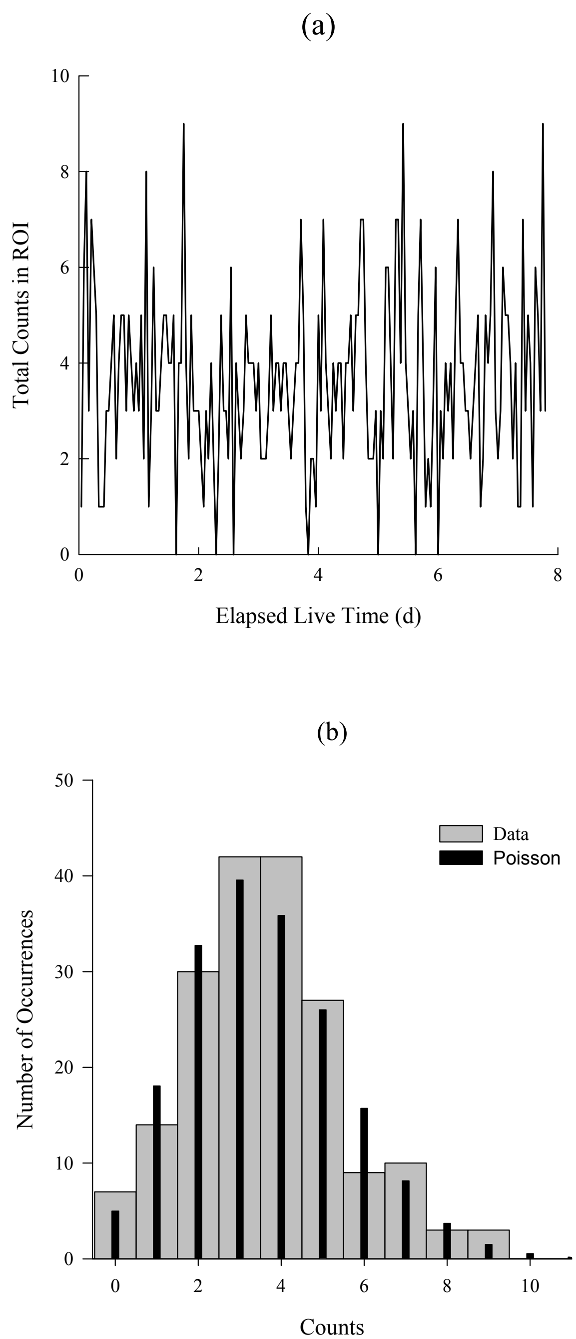
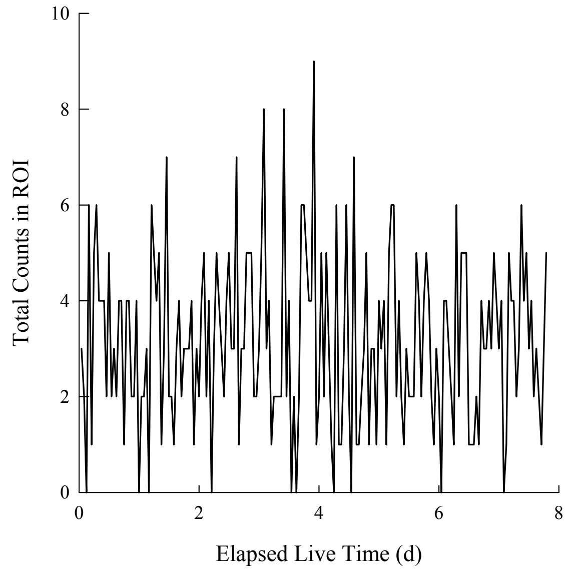


Fig. 8. Total background counts (a) in consecutive 1-h live-time intervals and (b) distribution of counts for SG1, cooled by the X-Cooler™, in the 1170- to 1176-keV ROI. Solid bars in (b) show the Poisson distribution having the same mean as the data.

(a)



(b)

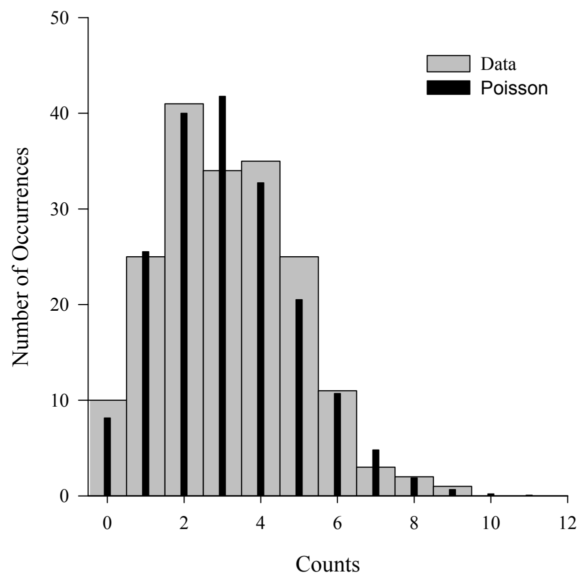


Fig. 9. Total background counts (a) in consecutive 1-h live-time intervals and (b) distribution of counts for SG1 cooled by the X-Cooler™, in the 1329- to 1336-keV ROI. Solid bars in (b) show the Poisson distribution having the same mean as the data.

4. DISCUSSION

The X-Cooler™ was developed as an alternative to LN_2 cooling of HPGe detectors. Among the many potential users of such technology are those involved in nuclear safeguards activities, including special nuclear material monitoring and verification. The data chosen for analysis include both those from higher-energy regions (662 kev, 1173 kev, and 1332 keV) typical of (-emitting radionuclides and from a low-energy region (59.5 keV) more typical of characteristic X-rays of uranium and the transuranic elements and having perhaps greater interest in nuclear safeguards applications.

Appropriate decisions relating to the use of the X-Cooler™ in place of LN_2, and effective use of the X-Cooler™ when it is the chosen alternative, depend on a complete understanding of its performance characteristics in relation to the alternative. Results reported here are germane to three important aspects of the performance of the X-Cooler™ used to cool HPGe detectors as compared with the benchmark LN_2: (1) stability and reproducibility, (2) precision and (3) limits of detection in HPGe systems cooled by the two alternatives. Stability and reproducibility are reflected through stochastic (random) and non-stochastic (data trends and anomalies) variability when operating under unchanging conditions for long periods of time. Precision is determined by photopeak width, since narrow, well-defined peaks allow more accurate identification of radionuclides, especially in mixtures. Limits of detection are directly related to background levels and variability, so that low backgrounds with minimal variability provide the best limits of detection.

Use of the X-Cooler™ compared more favorably with LN_2 cooling in the higher-energy ROIs than at lower energies. Comparison of photopeak areas in the monitored regions of interest (Tables 2 and 6, and Figure 4) shows that mean peak areas were nearly the same using the X-Cooler™ and LN_2 at 1332 keV and 1173 keV (although slightly lower using the X-Cooler™), but were significantly lower for X-Cooler™ cooling at 662 keV and 59.5 keV. The distribution of peak area results, expressed as the coefficient of variation, was almost twice as great using the X-Cooler™ than when using LN_2 in all four of the ROIs investigated.

Peak widths and their distributions were significantly greater with the X-Cooler™ than with LN_2, with the difference between the two cooling methods increasing as photon energy decreases. Figure 5 shows graphically the statistics reported for full width at half-maximum using LN_2 (Table 3) and using the X-Cooler™ (Table 7, but with anomalous data and trends not believed related to X-Cooler™ performance removed), illustrating the differences between the two cooling methods. The mean FWHM with the X-Cooler™ was about 2.4% greater than that using LN_2 at 1332 keV, increasing to around 17% greater at 59.5 keV. The corresponding coefficients of variation ranged from 65% greater for the X-Cooler at 1332 keV to 112% greater at 59.5 keV. The data for FWTM and FW(1/25)M indicated similar trends for these performance measures.

Increased background from SG1 after about 2.5 days (live time) of operation, shown in Figure 6, raises an additional issue about long-term stability and reproducibility of X-Cooler™ performance, particularly for safeguards applications. Such behavior during safeguards monitoring would result in a corresponding change in the limit of detection, perhaps without of the knowledge of the operator. Microphonics transferred to the detector through the X-Cooler™ coupling is suspected to have been the cause of the increased background. This speculation is supported by the absence of simultaneously increased background levels in the higher-energy ROIs, which, we note from past experience, tend to be less susceptible to vibration. The actual cause was not systematically determined, however.

Additional data (including backgrounds collected with LN2 cooling and detailed analysis of X-Cooler™ maintenance and repair history) that would have been helpful in the comparison of X-Cooler™ and LN_2 cooling of HPGe systems are not available for comparison because of early withdrawal of project support.

5. CONCLUSIONS

The X-Cooler™, because of its potential for use in remote or inaccessible locations and its other advantages over the use of LN_2, merits consideration as a viable alternative to LN_2 for cooling HPGe detectors in many applications. Its suitability for use in specific safeguards monitoring applications and procedures for its use in such applications should be determined by considering performance characteristics (total peak area, peak area distribution, full width at *x*-maximum, FW*x*M distribution, and susceptibility to background artifacts) at lower photon energies important in identifying and quantifying special nuclear materials. HPGe performance when cooled by the X-Cooler™ tended in this study to compare favorably with that when cooled with LN_2 at higher photon energies, but less favorably at lower energies. Use of the X-Cooler™ may be worthy of consideration for many safeguards applications, but any decision for its utilization should be made with cognizance of its limitations and the implications for data quality and usefulness for the intended purpose.

6. REFERENCES

1. E. Broerman, R. Keyser, T. Twomey, and D. Upp, "A New Cooler for HPGe Detector Systems", PerkinElmer Instruments-ORTEC. Paper presented at the 23rd meeting of the European Safeguards Research and Development Association on Safeguards and Nuclear Materials Management, Brugge, Belgium, May 2001. Available on the World-Wide Web at http://www.ortec-online.com/papers/reprints.htm.

2. E. Broerman, D. Upp, and T. Twomey, “Performance of a New Type of Electrical Cooler for HPGe Detector Systems.” Paper presented at the 42nd Annual Meeting of the Institute of Nuclear Materials Management, Indian Wells CA, July 18, 2001. Available on the World-Wide Web at http://www.ortec-online.com/papers/reprints.htm.

APPENDIX A

C++ Code for Extracting and Storing Spectral Data

APPENDIX A. C++ Code for Extracting and Storing Spectral Data

(excludes headers and 'include' files)

```
/*
****************************************
GVExtract.cpp
Collects spectrum data from GammaVision files. Runs as a DOS application.
Robert L. Coleman, 2002
Oak Ridge National Laboratory

Parameters:
   #1 - UFO File name
   #2 - Spectrum file name (complete)
   #3 - Spectral data format file indicating which peaks &/or ROIs to analyze
   #4 - format type for output (.dat or .xml formats)
   #5 - create new (unique) file

NOTE:  Must include afx.h, rlcspc.cpp
****************************************
*/

//declarations
#include "StdAfx.h"
#include <rlcspc.cpp>

// internal functions
bool FindIniEntry(FILE* inifile, CString entrytxt);
int GetIniPair(FILE* inifile, float* formatset);

void main(int argc, char *argv[])
{
   // get file name passed from command prompt
   // note: argv[0] is the exe file name
   rUFOData ufomain;
   rSpecData spdata;
   float area, bkg, fwhm, fw10m, fw25m, dataset[2], centroid;
   CString startdate, cstrTxt, outputfmt, nuclide;
   char cTxt[30];
   int i, misc;
   FILE *inifile, *outfile;
   CFileFind cffFile;
```

```
bool DataHeader=0;

outputfmt="DAT";
if (argc < 4)
{
   printf("\n");
   if (DumpFile("GVExtract.hlp")==0)
      // during debug
      DumpFile ("c:\\data\\code\\c++\\gvextract\\distfiles\\gvextract.hlp");
   printf("\n");
   return;
}

if (UFOGet(argv[1], ufomain) != 1)
{
   printf("\nError opening UFO file.  Note that file extension MUST be included.\n");
   return;
}

if (SpectrumGet(argv[2], spdata) != 1)
{
   printf("\nError opening SPECTRUM file.  Note that file extension MUST be included.\n");
   return;
}

// look for optional parameters
for (i=1; i<argc; i++)
{
   cstrTxt = argv[i];
   cstrTxt.MakeUpper();
   if (cstrTxt == "F=XML")
      outputfmt="XML";
}

CString detname=ufomain.detdesc1;

// open INI file
if ((inifile=fopen(argv[3],"r")) == NULL)
   {
      printf("\nError opening INI file!");
      return;
   }

detname.Replace(" ", "");  // remove spaces from detector name
// open output file
```

```
if (outputfmt == "DAT")
   // append to existing file or create new
   outfile=fopen(detname+".dat", "a");
else
{
   // create unique file everytime
   time_t now;
   time(&now);
   outfile=fopen(detname+"_"+_ltoa(now,cTxt,10)+".xml", "a");
}

if (fgetc(outfile) == EOF && outputfmt=="XML")
{
   // new file -- output XML header entry
   fprintf(outfile,"<?xml version=\"1.0\" ?>\n");
   fprintf(outfile,"<!--GammaSpec ROI Measurement Results-->\n");
}

// output common data
if (outputfmt == "DAT")
   fprintf(outfile,"%s,%s,%s,%10.0f,%5.3e", detname, ufomain.datestr,
       ufomain.timestr, ufomain.livetime, spdata.sum);
else
{
   // XML output
   fprintf(outfile,"<SpecResult>\n");
   fprintf(outfile,"\t<Detector>%s</Detector>\n", detname);
   fprintf(outfile,"\t<Date>%s</Date>\n", ufomain.datestr);
   fprintf(outfile,"\t<StartTime>%s</StartTime>\n", ufomain.timestr);
   fprintf(outfile,"\t<LiveTime>%5.3e</LiveTime>\n", ufomain.livetime);
   fprintf(outfile,"\t<TotalCounts>%5.3e</TotalCounts>\n", spdata.sum);
}

// extract peak data using FORMAT template file
// get peak data and output to DAT file
if (FindIniEntry(inifile, "[PEAK]"))
{
   while (!feof(inifile))
   {
      if ((misc=GetIniPair(inifile, dataset))==1)
      {
         // valid data set retrieved-- output results
         if (!DataHeader)
         {
            // print opening Peak Data header for XML
```

```
            fprintf(outfile,"\t<PeakData>\n");
            DataHeader=1;
         }

         area= 0; bkg = 0; fwhm= 0; fw10m=0;  fw25m=0;
         nuclide=""; centroid=dataset[0];
         if ((i=UFOFindPeak(ufomain, dataset[0], dataset[1])) > -1)
         {
            area= ufomain.peak[i].netarea;
            bkg = ufomain.peak[i].background;
            fwhm= ufomain.peak[i].fw2m_e;
            fw10m=ufomain.peak[i].fw10m_e;
            fw25m=ufomain.peak[i].fw25m_e;
            nuclide=ufomain.peak[i].nuclidename;
         } //endif
         if (outputfmt == "DAT")
            fprintf(outfile, ",%8.0f,%8.0f,%6.3f,%6.3f,%6.3f",
                   area, bkg, fwhm, fw10m, fw25m);
         else
         {
            // XML output
            fprintf(outfile,"\t\t<Peak>\n");
            fprintf(outfile,"\t\t\t<Energy>%7.2f</Energy>\n", centroid);
            fprintf(outfile,"\t\t\t<NetArea>%8.0f</NetArea>\n", area);
            fprintf(outfile,"\t\t\t<BkgArea>%8.0f</BkgArea>\n", bkg);
            fprintf(outfile,"\t\t\t<Isotope>%s</Isotope>\n", nuclide);
            fprintf(outfile,"\t\t</Peak>\n");
         }
      } //endif
      else if (misc==0)
         // end-of-section or end-of-file
         break;

      else if (misc==-1)
         // error in data
      {
         printf("Error occurred reading PEAK data from INI file. Press any key.");
         keypress();
         return;
      } //end elseif
   } //endwhile
   if (outputfmt == "XML")
      // print closing section
      fprintf(outfile,"\t</PeakData>\n");
} //endif
```

```
DataHeader=0;
// parse and output ROI sums
if (FindIniEntry(inifile, "[roi]"))
{
   while (!feof(inifile))
   {
      if ((misc=GetIniPair(inifile, dataset))==1)
         // valid data set retrieved-- output result
      {
         if (!DataHeader)
         {
            // print opening Peak Data header for XML
            fprintf(outfile,"\t<ROIData>\n");
            DataHeader=1;
         }
         if (outputfmt == "DAT")
            fprintf(outfile,",%5.3e",
                SpcSumRegionE(spdata,dataset[0],dataset[1]));
         else
         {
            // XML output
            fprintf(outfile,"\t\t<ROI>\n");
            fprintf(outfile,"\t\t\t<LowE>%8.0f</LowE>\n", dataset[0]);
            fprintf(outfile,"\t\t\t<HiE>%8.0f</HiE>\n", dataset[1]);
            fprintf(outfile,"\t\t\t<Sum>%8.0f</Sum>\n", SpcSumRegionE(spdata,dataset[0],dataset[1]) );
            fprintf(outfile,"\t\t</ROI>\n");
         }
      } //endif
      else if (misc==0)
         // end-of-section or end-of-file
         break;
      else if (misc==-1)
         // error in data
      {
         printf("Error occurred reading ROI data from INI file. Press any key.");
         keypress();
         return;
      } //end elseif
   } //endwhile
   if (outputfmt == "XML")
      // print closing sections
      fprintf(outfile,"\t</ROIData>\n");
} //endif
if (outputfmt == "XML")
```

```
      // print closing sections
      fprintf(outfile,"</SpecResult>\n");

   fprintf(outfile,"\n");
   fclose(outfile);
   fclose(inifile);

}  //end main


bool FindIniEntry(FILE* inifile, CString CSentrytxt)
{
   // searches for entrytxt (case insensitive) in file inifile (text file)
   // returns 1 if found, 0 if not

   fseek(inifile,0,SEEK_SET);
   bool rtn=0;
   char txt[200];
   CString cstrTxt;
   CSentrytxt.MakeUpper();

   while (!feof(inifile) && rtn==0)
   {
      fscanf(inifile,"%s",txt);
      cstrTxt=txt;
      cstrTxt.MakeUpper();
      if (cstrTxt.Find("/")>-1)
         fgets(txt,200,inifile);      // dump remainder of line and loop
      else if (cstrTxt==CSentrytxt)
         rtn = 1;
   }//endwhile
   return rtn;
}


int GetIniPair(FILE* inifile, float* formatset)
   // get a set of format data x1,x2 from inifile and store into
   //  array formatset [0,1]
   //  Return = 0 if end of file or another section has been reached
   //         = 1 if VALID data is retrieved
   //        = -1 if INVALID data is retrieved

{
   char txt[200], *stopstr;
   CString cstrTxt;
```

```
    int rtn=0, i=0;
    formatset[0]=0;
    formatset[1]=0;

    while (!feof(inifile))
    {
        fscanf(inifile,"%s",txt);        // spaces are ignored
        cstrTxt=txt;
        if (cstrTxt.Find("[")>-1) break; // start of another section: abort
        if (cstrTxt.Find("/")>-1)
            fgets(txt,200,inifile);      // comment: dump remainder of line and loop
        else if (cstrTxt != "" && !feof(inifile))
        {
            formatset[i++]=(float) strtod(cstrTxt,&stopstr);
            if (i==2)
            {
                // set of data retrieved
                rtn=1;
                break;
            } //endif
        } //endif

    } //endwhile
    if (i>0 && (formatset[0]==0 || formatset[1]==0))
        // invalid data read
        rtn=-1;

    return rtn;
}
```

```
/*
*************************************************************
rlcspc.cpp
Functions for accessing SPC, CHN and UFO data

Robert L. Coleman, 2002
Oak Ridge National Laboratory

Notes:    1) Requires the following includes in
             the project: <afx.h>, <rlclib.cpp>,
                          <afxdisp.h>, <cmath>

Requires compilation using MFC
**************************************************************
*/

#if !defined (rlcspc_cpp_included)
#define rlcspc_cpp_included

#include <rlclib.cpp>
#define CHN -1
#define SPC 1

// User-called Functions -------------------------------------------

int SpectrumGet(char* infilename, struct rSpecData &spdata);
   // Opens infilename spectrum file and fills structure spdata
   // Works for real or int SPC files and also CHN files.
   //  Returns  -1 if input file does not exist
   //           0 if file cannot be a spectra file
   //           1 for success

float SpcSumRegionE(struct rSpecData spectrum, float e1, float e2);
   // sums channel counts between e1 keV and e2 keV of spectrum

int UFOGet(char*  infilename, struct rUFOData &udata);
   //  Opens infilename UFO file and fills structure udata

COleDateTime DATEortec(double datetime);
   //  converts ortec date+time real value to a COleDateTime DATE value

int UFOFindPeak(rUFOData &udata, float e1, float e2);
   //  returns index pointer into peak array contained in udata.
   //     e1 is the target peak energy wanted (keV)
   //     e2 is the tolerance allowed for the peak (keV)
```

```
float ChtoE(float c, float zero, float slope, float poly);
   // converts channel c to energy (keV) given a1,a2,a3 slope coefficients

int EtoCh(float c, float zero, float slope, float poly);
   // converts energy e (keV) to channel given a1,a2,a3 slope coefficients


// library-internal functions ---------------------------------------
void riSpectGetSpcBase(struct rSpecData &spdata);
void riSpectGetChnBase(struct rSpecData &spdata);
void riUFOGetPeak(struct rUFOData udata, struct rPeakData &peak,
                 int peakoffset, int nuclide_os);
int QSortPeakByE( const void *arg1, const void *arg2 );

// data structures  -------------------------------------------------
struct rSpecData {
   short  filetype,    //  SPC (real=5 or int=1) or CHN= -1
          start_chan,
          chan_count;
   float  ezero,
          eslope,
          epoly,
          livetime,
          sum,         // total number of counts in spectrum
          *chan_data;
   FILE   *infile;
};

struct rUFOData {
   short  peakcount,          // total number of peaks (lib1 + unknown)
          nuclidecount,       // number of nuclides identified
          detnumber;          // detector number
   struct  rPeakData *peak;   // pointer to array of Lib 1 identified peaks
   float  ezero,              // energy coefficients
          eslope,
          epoly,
          fzero,              // fwhm coefficients
          fslope,
          fpoly,
          livetime,
          realtime,
          starttime;
   double  datetime_double;

   COleDateTime datetime_DATE;
```

```
   CString   datestr,
          timestr,
          detdesc1,     // detector description
          detdesc2;
   FILE   *infile;      // spectrum file
};

struct rPeakData {
   float  channel,
          energy,
          netarea,
          background,
          bkgbelow,     //avg bkg below peak
          bkgabove,     //avg bkg above peak
          fw25m_ch,
          fw10m_ch,
          fw2m_ch,
          fw25m_e,
          fw10m_e,
          fw2m_e;
   CString   nuclidename;
};


// FUNCTION CODE  ---------------------------------------------------

int SpectrumGet(char* infilename, rSpecData &spdata)
{
   // return = 1 for success, 0 for incorrect file type,
   //          -1 for file not found

   int rtn=1;
   short i;
   if ((spdata.infile=fopen(infilename,"rb")) != NULL)
   {
      // check to insure proper file types
      fread(&i,2,1,spdata.infile);
      if (i !=CHN && i !=SPC) rtn = 0;
   }
   else
      rtn = -1;

   // populate base spectrum file data structure
   if (i==CHN && rtn==1)
      riSpectGetChnBase(spdata);
```

```
   if (i==SPC && rtn==1)
   {
      fread(&spdata.filetype,2,1,spdata.infile);  //real=5, int=1
      riSpectGetSpcBase(spdata);
   }

   if (rtn==1)
   {
      spdata.sum=ArraySum(spdata.chan_data,0,spdata.chan_count-1);
      fclose(spdata.infile);
   }
return rtn;
}

void riSpectGetChnBase(rSpecData &spdata)
{
   short chan_offset;  // bytes from beginning of file to start of chan data
   int temp;
   FILE *infile = spdata.infile;
   fseek(infile,12,SEEK_SET);
   fread(&temp,4,1,infile);   // chn time stored as number of 20 ms ticks
   spdata.livetime=(float) temp*20/1000;
   fseek(infile,28,SEEK_SET);
   fread(&spdata.start_chan,2,1,infile);
   fread(&spdata.chan_count,2,1,infile);
   chan_offset=32;

   int* ichan_data= new int[spdata.chan_count];
   fseek(infile,chan_offset,SEEK_SET);
   fread(ichan_data,4,spdata.chan_count,infile);
   spdata.chan_data=ArrayCopy(ichan_data, spdata.chan_count);
   delete[] ichan_data;

   fseek(infile,4,SEEK_CUR);
   // get energy cal information
   fread(&spdata.ezero,4,1,infile);
   fread(&spdata.eslope,4,1,infile);
   fread(&spdata.epoly,4,1,infile);
return;
}

void riSpectGetSpcBase(rSpecData &spdata)
{
   FILE *infile = spdata.infile;
```

```
    short calrecoffset, chan_offset;

    fseek(infile,94,SEEK_SET);
    fread(&spdata.livetime,4,1,infile);
    fseek(infile,60,SEEK_SET);
    fread(&chan_offset,2,1,infile);  // offset = record count
    fseek(infile,64,SEEK_SET);
    fread(&spdata.chan_count,2,1,infile);
    fread(&spdata.start_chan,2,1,infile);

    if (spdata.filetype==1)
    { // integer SPC file.  128 bytes per record, 4 bytes per channel
        int* ichan_data= new int[spdata.chan_count];
        fseek(infile,(chan_offset-1)*128,SEEK_SET);
        fread(ichan_data,4,spdata.chan_count,infile);
        spdata.chan_data=ArrayCopy(ichan_data, spdata.chan_count);
        delete[] ichan_data;
    }

    else
    { // real (float) SPC file. 128 bytes per record, 4 bytes per channel
        spdata.chan_data= new float[spdata.chan_count];
        fseek(infile,(chan_offset-1)*128,SEEK_SET);
        fread(spdata.chan_data,4,spdata.chan_count,infile);
    }

    // get energy cal information
    fseek(infile,34,SEEK_SET);
    fread(&calrecoffset,2,1,infile);
    // note: calrecoffset is the start of the calibration DATA record

    fseek(infile,(calrecoffset-1)*128+20,SEEK_SET);
    fread(&spdata.ezero,4,1,infile);
    fread(&spdata.eslope,4,1,infile);
    fread(&spdata.epoly,4,1,infile);

return;
}

float SpcSumRegionE(struct rSpecData spec, float e1, float e2)
{
    // sums counts between e1 keV and e2 keV in spectrum
    int c1=EtoCh(e1, spec.ezero, spec.eslope, spec.epoly);
    int c2=EtoCh(e2, spec.ezero, spec.eslope, spec.epoly);
    if (c2>(spec.chan_count-1)) c2=spec.chan_count-1;
```

```
   // get channel numbers for e1 and e2 using quadradric solution
   return ArraySum(spec.chan_data, c1, c2);
}


float ChtoE(float i, float zero, float slope, float poly)
{
   // converts channel count i to energy (keV) given
   // calibration coefficients a1=zero, a2=slope and a3=quadradic coeff.
   float a=zero, b=slope, c=poly, rtn;
   rtn=a + i*b + i*i*c;
   if (rtn<0) rtn=0;
   return rtn;
}

int EtoCh(float e, float a1, float a2, float a3)
{
   // converts energy e to channel number
   // uses quadradic solution for ax^2+bx+c=0 and always uses a (+) for sqrt term
   // calibration coefficients a1=zero, a2=slope and a3=quadradic coeff.
   float a=a3, b=a2, c=a1-e;
   return (int) ((-b+sqrt(b*b-4*a*c))/(2*a));
}


int UFOGet(char* infilename, rUFOData &udata)
{
   // return = 1 for success, 0 for incorrect file type,
   //          -1 for file not found
   int rtn=1, i, j;
   short peakcount1, peakcountU;
   short filetype1, filetype2;
   FILE *infile;

   if ((infile=fopen(infilename,"rb")) != NULL)
   {
      // check to insure proper file types
      fread(&filetype1,2,1,infile);
      fread(&filetype2,2,1,infile);
      if (filetype1 != 1 || filetype2 !=1024)
         return 0;
   }
   else
      return -1;
```

```
udata.infile=infile;

short  gen_os,          //GEN record
       peak1_os,    //lib 1 peak start record offset
       unpeak_os,       //unknown peak record offset
       nuclide1_os, //lib 1 nuclide start record offset
       calrec1_os,      //cal data record offset
       detdesc_os;      //det description record offset

// populate base ufo data structure.  UFO files use 64 word (128 byte) records
fseek(infile,12,SEEK_SET);
fread(&detdesc_os,2,1,infile);

fseek(infile,34,SEEK_SET);
fread(&calrec1_os,2,1,infile);

fseek(infile,128,SEEK_SET);
fread(&gen_os,2,1,infile);

// lib1 peaks header
fseek(infile,128+6,SEEK_SET);
fread(&peak1_os,2,1,infile);
fread(&peakcount1,2,1,infile);
peakcount1 = peakcount1/2;       // no idea why this is true...

// unknown peaks header
fseek(infile,128+24,SEEK_SET);
fread(&unpeak_os,2,1,infile);
fread(&peakcountU,2,1,infile);
peakcountU = peakcountU/2;       // no idea why this is true...
// nuclide header
fseek(infile,128+30,SEEK_SET);
fread(&nuclide1_os,2,1,infile);
fread(&udata.nuclidecount,2,1,infile);

// detector desc
fseek(infile,(detdesc_os-1)*128,SEEK_SET);
fread(udata.detdesc1.GetBuffer(64),64,1,infile);
fread(udata.detdesc2.GetBuffer(64),64,1,infile);
udata.detdesc1.TrimRight();
udata.detdesc2.TrimRight();
udata.detdesc1.ReleaseBuffer();
udata.detdesc2.ReleaseBuffer();

// cal data
```

```
fseek(infile,(calrec1_os-1)*128+76,SEEK_SET);
fread(&udata.detnumber,2,1,infile);
fseek(infile,(calrec1_os-1)*128+20,SEEK_SET);
fread(&udata.ezero,4,1,infile);
fread(&udata.eslope,4,1,infile);
fread(&udata.epoly,4,1,infile);
fread(&udata.fzero,4,1,infile);
fread(&udata.fslope,4,1,infile);
fread(&udata.fpoly,4,1,infile);

// GEN record data
fseek(infile,(gen_os-1)*128+46,SEEK_SET);
fread(&udata.livetime,4,1,infile);
fread(&udata.realtime,4,1,infile);
fread(&udata.starttime,4,1,infile);
fseek(infile,(gen_os-1)*128+76,SEEK_SET);
fread(&udata.datetime_double,8,1,infile);
COleDateTime tdate = DATEortec(udata.datetime_double);
udata.datetime_DATE = tdate;
udata.datestr = tdate.Format("%Y-%b-%d");
udata.timestr = tdate.Format("%H:%M:%S");

// dimension for ALL peaks to be retreived
int tmppeakcount = peakcount1+peakcountU;
rPeakData* tmppeak = new rPeakData[tmppeakcount];

// get library-1 peak records (identified peaks for Lib 1)
for (i=0; i<peakcount1; i++)
{
   // peak records are 128 bytes each
   // one blank record between each (no idea why... see peakcount1 above)
   riUFOGetPeak(udata, tmppeak[i], (peak1_os-1)*128+i*256, nuclide1_os);
}

// get unknown peak records and store into peak array
j=0;
for (i=peakcount1; i<tmppeakcount; i++)
   // peak records are 128 bytes each
   // one blank record between each (no idea why... see peakcount1 above)
{
   riUFOGetPeak(udata, tmppeak[i], (unpeak_os-1)*128+j++*256, 0);
}

// count non-zero peaks
j=0;
```

```
    for (i=0; i<tmppeakcount; i++)
    {
        if (tmppeak[i].netarea > 0)
            j++;
    }
    udata.peak = new rPeakData[j];
    udata.peakcount=j;

    // remove peaks with zero net counts
    j=0;
    for (i=0; i<tmppeakcount; i++)
    {
        if (tmppeak[i].netarea>0)
            udata.peak[j++]=tmppeak[i];
    }

    // sort peaks by energy
    qsort(udata.peak,udata.peakcount,sizeof(rPeakData),QSortPeakByE);

    // debug: print peak data to console
    printf("\nPeaks Found:\n\n");
    printf("Energy \t Net\tBkg\tIsotope\n");
    printf(" (keV) \t Counts\t\n");
    printf("-------\t ------\t-----\t---------\n");
    for (i=0; i<udata.peakcount; i++)
        printf("%.2f\t %.0f\t%.0f\t%s\n", udata.peak[i].energy, udata.peak[i].netarea,
                            udata.peak[i].background, udata.peak[i].nuclidename);

    fclose(infile);

return rtn;
}

//------------------------------------

void riUFOGetPeak(rUFOData udata, rPeakData &peak, int peakoffset, int nuclide_os=0)
    // udata = rUFOData structure where peak is stored
    // peak  = rPeakData structure where peak is stored
    // peakoffset is pointer to byte 0 of peak record
    // nuclide_os = offset to first nuclide record. 0 for unknown peaks.
{
        short misc_os;
        FILE *infile = udata.infile;
        fseek(infile,peakoffset,SEEK_SET);
        fread(&peak.energy,4,1,infile);
```

```
        fseek(infile,peakoffset+12,SEEK_SET);
        fread(&peak.netarea,4,1,infile);
        fread(&peak.background,4,1,infile);
        fseek(infile,peakoffset+28,SEEK_SET);
        fread(&peak.channel,4,1,infile);
        fseek(infile,peakoffset+80,SEEK_SET);
        fread(&peak.bkgbelow,4,1,infile);
        fread(&peak.bkgabove,4,1,infile);
        fseek(infile,peakoffset+32,SEEK_SET);
        fread(&peak.fw25m_ch,4,1,infile);
        fread(&peak.fw10m_ch,4,1,infile);
        fread(&peak.fw2m_ch,4,1,infile);

        if (peak.netarea <0) peak.netarea=0;
        if (peak.fw25m_ch < 1) peak.fw25m_ch=0;
        if (peak.fw10m_ch < 1) peak.fw10m_ch=0;
        if (peak.fw2m_ch < 1) peak.fw2m_ch=0;

        if (nuclide_os>0)
        {
           fseek(infile,peakoffset+48,SEEK_SET);
           fread(&misc_os,2,1,infile);
           // nuclide records are 64 bytes.  nuclide record offset is
           // given relative to 128-byte file records.
           fseek(infile,(nuclide_os-1)*128+(misc_os-1)*64,SEEK_SET);
           fgets(peak.nuclidename.GetBuffer(8),8,infile);
           peak.nuclidename.ReleaseBuffer();
        }

        else
           peak.nuclidename="Unknown";

        // calculate peak parameters in keV
        float a1=udata.ezero, a2=udata.eslope, a3=udata.epoly;
        peak.fw25m_e= ChtoE(peak.fw25m_ch, a1, a2, a3);
        peak.fw10m_e= ChtoE(peak.fw10m_ch, a1, a2, a3);
        peak.fw2m_e= ChtoE(peak.fw2m_ch, a1, a2, a3);
}


int UFOFindPeak(rUFOData &udata, float e1, float e2)
{
   // find peak closest to energy e1 from udata.  Must be within +/- e2 of e1.
   // e1 = target and e2 = tolerance are expressed in keV
   // returns index into peak array for udata
```

```
   // returns -1 if peak not found
   // Example:  rUFOData test;
   //           int i = UFOFindPeak(test, 662, 2);
   //           //find peak within 2 keV of 662 keV
   //           cout << test.peak[i].netarea
   int i, select=-1;
   double  d1, d2;
   for (i=0; i<udata.peakcount; i++)
   {
      // raw compare
      if ((udata.peak[i].energy < (e1+e2)) && (udata.peak[i].energy > (e1-e2)))
      {
         if (select > -1)
            // compare to previous find to select best pick
         {
            d1 = fabs(udata.peak[select].energy-e1);
            d2 = fabs(udata.peak[i].energy-e1);
            if (d2<d1)
               select = i;
         }
         else
            // first to be found
            select=i;
      }
   }
   return select;
}
#endif
```

ORNL/TM-2002/236

ELECTRONIC DISTRIBUTION

Internal

R. L. Coleman
J. S. Bogard
M. E. Murray
V. M. Baylor
ORNL Laboratory Records–CPPR

External

David P. Spears, U.S. Department of Energy, Office of Nonproliferation Research and Engineering, Routing NA-22, Forrestal GH-068, 1000 Independence Avenue, S.W., Washington, DC 20585

Daniel L. Upp, Vice President, ORTEC, 801 S. Illinois Ave., Oak Ridge, TN 37831