SAND2017-8169C

AUTOMATIC DATA PROCESSING AND DATA DISPLAY SYSTEM FOR THE HERMES III ACCELERATOR*

Sean K Coffey[ξ], Adam Circle, Benjamin Ulmen, Chris Grabowski, Nathan Joseph, Barbara Lewis and Victor-Harper-Slaboszewicz

Sandia National Laboratories
PO Box 5800 MS-1106
Albuquerque, NM 87185-1106

Abstract

This paper describes the software changes made to the data processing and display system for HERMES III accelerator at the Simulation Technology Laboratory (STL) at Sandia National Laboratories, New Mexico. The HERMES III accelerator is a gamma ray simulator producing 100kRad[Si] dose per shot with a full width half max pulse duration of ~25 nanoseconds averaging six shots per day. For each accelerator test approximately 400 probe signals are recorded over approximately 65 digitizers. The original data processing system provided the operator a report summarizing the start of probe signal timings for groups of probes located within the power flow conductors. This timing information is indicative of power flow symmetry allowing the operator to make necessary adjustments prior to the next test. The report also provided data overlays concerning laser trigger light output, x-ray diode currents and x-ray source output. Power flow in the HERMES III accelerator is comprised of many circuit paths and detailed current and voltage information within these paths could provide a more thorough understanding of accelerator operation and performance, however this information was either not quickly available to the operators or the display of the data was not optimum. We expanded our data processing abilities to determine the current and voltage amplitudes throughout the power flow conductors and improved the data display abilities so data plots can be presented in a more organized fashion.

We detail our efforts creating a software program capable of processing the ~ 400 probe signals together with an organized method for displaying the dozens of current and voltage probes. This process is implemented immediately after all digitizer data has been collected so the operator is provided timing and power flow information shortly after each accelerator shot.

I. INTRODUCTION

This paper describes a new software program for processing and displaying the output shot data from a HERMES III accelerator test. The HERMES III accelerator is a gamma ray simulator at Sandia National Laboratories, New Mexico. HERMES III performs an average of 6 shots per day and records from 370 data channels during each shot; the large quantity of data produced each day necessitates an automated method for analyzing signals quickly and accurately. Our new data processing and display (DPAD) program will combine the user-friendly features of LabVIEW with the data analysis and visualization tools of the Python language. The data processing and display routines originate from a text based input configuration file (CF) allowing users to uniquely process any data channel and customize its displayed format. The Python data processing routines will be made available for inspection, allowing users to examine all post processing details and ensure maximum transparency of processing algorithms employed.

This paper is divided into four sections. The second section presents the DPAD software program requirements followed by the section detailing the input configuration file format and Python program functions. The fourth section presents an example configuration file and examines the resultant software program operations. We close this paper with the summary, future efforts and acknowledgements.

II. PROGRAM REQUIREMENTS

DPAD program objectives are:

- Easy to use.
- Transparent data processing code.
- Scalable.

---

*Work sponsored by Sandia National Laboratories, a multi mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

[ξ] email: skcoffe@sandia.gov

- Compliable – distributable – low cost.
- Operated manually as a data viewer or automated for batch processing and display.
- Batch code is text based and customizable.
- Allow users to create and implement code.
- Create ASCII type data files with a configurable output structure.
- For each ASCII data file create a text file to include probe header and processing data.
- Create Plot overlay image files with configurable start and end times.
- Create shot report documents.
- Create file maker pro data files.
- Create setup document files detailing:
  1) Scope setups for all channels
  2) Processing information for all channels
  3) Diagnostic information for all channels

III. INPUT CONFIGURATION FILE

The data processing and display information for all data channels originates from a single Windows text based configuration file. This file format structure is associated with filenames containing the ".ini" extension. This file is composed of unique section names containing keywords and their keyword values. For section names exactly matching specific DAQ data channel names we list the keywords, corresponding to the Python programs, we want to execute.

The Python programs perform multiple type of operations to include mathematical signal processing, create ASCII type spreadsheet data files (original data format is binary), create waveform plot overlays & save as image files and create document files.

This configuration file format allows users to easily view, edit and create files to suit their own requirements and its sectional format inherently separates the data channels making it easy to understand. Presently, the last section within our configuration file is reserved for programs that analyze and or process groups of data channels but the order of sections can be changed as needed. Table I presents configuration file keywords presently linked to Python programs.

Table 1: CF Keywords linked to Python Programs

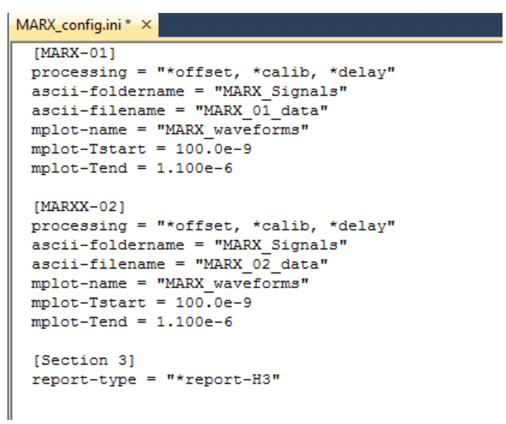
Configuration keyword	Python program	Python program information
*offset	Py_offset	Offset correct baseline
*calib	Py_calib	Multiply Y-values by calib-#
*delay	Py_delay	Add delay-# to X-values
*integ	Py_integ	Integrate data (Tstart, Tend)
*report-H3	Py_report-H3	HERMES III report program
*asci-folder *asci-file	Py_ascii	Create ASCII data set (requires folder name, file name & data set)
*mplot	Py_mplot	Create an X-Y waveform plot overlay of 1 or more data sets

IV. EXAMPLE CONFIG FILE

We now present a 2-channel example configuration file (CF) and "apply" this CF to a typical HERMES III accelerator shot data acquisition (DAQ) output file. In this example, we assume that:

1. The HERMES III DAQ data file contains two data channels named "MARX-01" and "MARX-02".

2. The configuration file keywords are linked to Python programs as shown in Table 1.




```
[MARX-01]
processing = "*offset, *calib, *delay"
ascii-foldername = "MARX_Signals"
ascii-filename = "MARX_01_data"
mplot-name = "MARX_waveforms"
mplot-Tstart = 100.0e-9
mplot-Tend = 1.100e-6

[MARXX-02]
processing = "*offset, *calib, *delay"
ascii-foldername = "MARX_Signals"
ascii-filename = "MARX_02_data"
mplot-name = "MARX_waveforms"
mplot-Tstart = 100.0e-9
mplot-Tend = 1.100e-6

[Section 3]
report-type = "*report-H3"
```

Figure 2: Screen shot of "MARX_config.ini"

The configuration file contents shown in figure 2, contains 3 separate sections. The first two sections pertain to specific data channels followed by the last containing a keyword & keyword value we recognize as linked to a Python report generation program.

The data processing and display program flow that follows when this CF file is "applied" to the DAS data file is as follows:

- The first step extracts all configuration file channel section names.
- Next, check if these names exactly match the channel names within the DAS data file.
- The first section name "MARX-01" is a match.
- The second name "MARXX-02" is misspelled and is not a match.
- For "MARX-01" section, extract keywords and keyword values:
  1) "processing" keyword value = "*offset, *calib, *delay" defines 3 Python data processing programs to be executed.
  2) "asci-foldername" keyword value = "MARX_Signals" defines the ASCII sub-folder name.
  3) "asci-filename" keyword value = "MARX_01_data" defines the ASCII file name.
  4) Create the ASCII data file.
  5) "mplot-name" keyword value = "MARX_waveforms" defines the waveform plot image filename.
  6) "mplot-Tstart" keyword value = 100.0e-9 defines the start time of the mplot waveform graph.
  7) "mplot-Tend" keyword value = 1.100e-6 defines the end time of the mplot waveform graph.
  8) Create mplot plot overlay, save the waveform as image file in a sub-folder labeled "Mplot-images". Note: This type waveform plot overlay can display 1 or more waveforms on a single X-Y type waveform graph.

- Finally, the last section of our example contains the keyword "*report-H3". This keyword is linked to "Py_report-H3" so the DPAD program starts the HERMES III report generation program.

Note: In the example, if the configuration section name "MARXX-02" had been correctly spelled then that DAS channel would have been processed, its ASCII data file created, its waveform data then plotted together with the MARX-01 data and this 2-signal plot overlay saved as an image file.

For processing **all** data channels in a typical HERMES III DAQ data file, the default configuration file will contain ~ 360 channel sections plus the last section containing the appropriate report generation program to execute. The DPAD will create the 360 ASCII data files in ~ 15 sub-folders (i.e. MTG, MARX, IS, PFL, TL, Cavity…) providing an organized layout for the data. After post-processing, the DPAD program then plots the data channels per the configuration file instructions. For typical HERMES III accelerator tests the DPAD program will create ~65 separate JPG type image files. Finally, the DPAD will execute the HERMES III shot report Python program creating a MS-Word type document containing detailed timing information and timing scatter plots of the accelerator power as it flows within the accelerator sections.

V. SUMMARY

This data analysis program is a work in progress and its present abilities are limited. However, we expect the system to be operational by late August 2017. The processing and display of the 360 probes signals will assist the DAS and machine operators to quickly access machine performance and guide corrective actions for the next HERMES III accelerator test. The ability to organize the 360 ASCII data files into sub-folders will help keep archived data more organized. The use of Python as the base code platform allows us access to many processing and display tools that have been well tested by others. A copy of the Python functions will be archived in a shared location providing maximum transparency of the processing details and methods.

Our method for introducing processing instructions at the data channel name level provides users with useful and flexible data processing and display tools using intuitive keyword commands. This method is expandable and as time permits more advanced Python programs will be developed and assigned new keyword names. Users may then run

the new Python programs simply by adding these new keywords within their configuration file.

VI. FUTURE EFFORTS

The present Python functions built into our DPAD system are limited but we expect many more to be developed soon. These additional functionalities include:

- Creating "flags" to the user / operator if key accelerator performance indicators have not been achieved.
- Calculate and apply mathematical cable compensation programs to correct for long diagnostic cable path length effects.

- Examining accelerator current and voltage in greater detail.
- Allow for multiple accelerator shot comparisons.
- Present a demonstration of the software at the 2018 IEEE International Power Modulator and High Voltage Conference in Jackson, Wyoming.

VII. ACKNOWLEDGEMENTS

Dr. Jim Degnan (AFRL) for the initial guidance for processing data and automating the procedures. Dr. Glen Wurden and the late Dr. Tom Intrator (LANL) for introducing the "MDS+" data file format and its "jScope" visualization concept, upon which the "mplot" display tool is loosely based. Finally, Dr. Ed Ruden (AFRL) for our talks concerning the importance of transparent data processing