

Fluid Dynamics experiments with Tcl

Ron Fox¹, Vaibhav Khane²

Abstract—Computer Automated Radioactive Tracking (CARPT) has emerged as a powerful Technique for mapping fluid flow under a variety of conditions. This paper describes the adaptation of a general purpose nuclear physics event-based data acquisition system to the needs of the CARPT apparatus at Missouri University of Science and Technology. The resulting software is a C++ multithreaded framework which communicates via events with a thread running a Tcl interpreter. The Tcl scripts run by the interpreter provide Experimental control, online-data analysis and data storage for later offline analysis.

I. INTRODUCTION

Understanding how fluids flow through chemical reactor vessels is an important piece in the puzzle of optimizing many industrial processes. Ab-initio fluid dynamic calculations and even simulations are not able to provide solutions for realistic reactors. Furthermore since larger reactors, as well as those which several material phases (e.g. solids, liquids and gasses in the same vessel) are optically opaque, experiments to track fluid flow in these reactors are not trivial..

As one often quoted author in the field says: “Multiphase reactors are widely used in petroleum, chemical, petrochemical, pharmaceutical and metallurgical industries as well as in materials processing and pollution abatement...the physical phenomena that affect the fluid dynamics of such systems are not yet entirely understood. This makes a priori predictions of important process parameters... very difficult.”[1] Figure 1 at right is a schematic of a Circulating Fluidized Bed (CFB) reactor that gives an idea of the scale of these devices. An understanding of the flow of fluids through reactors like this and others is essential to the optimization of a large variety of industrial processes.

The remainder of this paper is organized as follows:

- The CARPT method is introduced and described.
- The computer problem being solved is described.
- We describe how the base NSCL VM-USB readout framework was modified to meet the needs of this system.

II. TRACING FLUID FLOW WITH RADIOACTIVE SOURCES

1. National Superconducting Cyclotron Laboratory Michigan State University.
2. Missouri University of Science and Technology Very High Temperature Reactor (VHTR) consortium.

In recent years, introducing radioactive tracers into the flow of fluidized reactors has been a fruitful way to perform experimental studies of the fluid dynamics of these devices. One such method, called Computer Automated Radioactive Particle Tracking involves injecting a radioactive source that has neutral buoyancy into the fluid being studied.

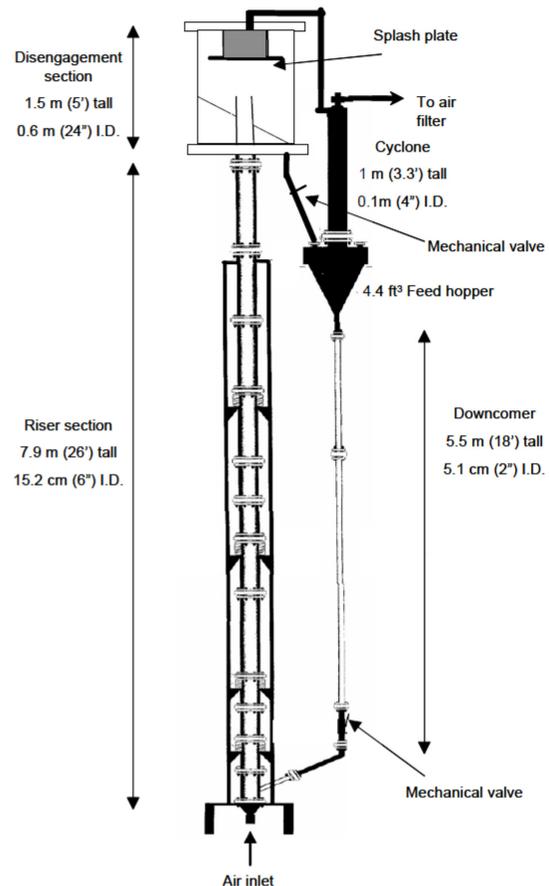


Figure 1 CFB reactor prototype [2]

Detectors placed around the reactor vessel track the position and velocity of the source as a function of time. Normally sources composed of ⁴⁶Sc are used. ⁴⁶Sc decays emitting a β and a γ-ray with characteristic energies of 900KeV, 1.0 and 1.3 MeV. NaI, detectors sensitive to the long ranged γ are used to track the particle. This is shown schematically in Figure 2.

The position of the source is determined by measuring the count rates in the detectors as a function of time. Position/intensity maps obtained during calibration runs are then used to derive the time evolution of the position and

vector velocity of the particle. This simple sounding procedure is actually quite complex because:

- A good knowledge of detector efficiencies is required to understand what a specific detector rate means.
- Detector acceptance depends on angle of incidence.
- Pile-up either due to a highly active source or the recovery time of the detectors need to be factored in (While the signal rise times of NaI crystal is quite fast, the fall time can be several 10s of microseconds).
- The reactor contents can absorb gamma rays resulting in a position dependent attenuation of the raw intensity.

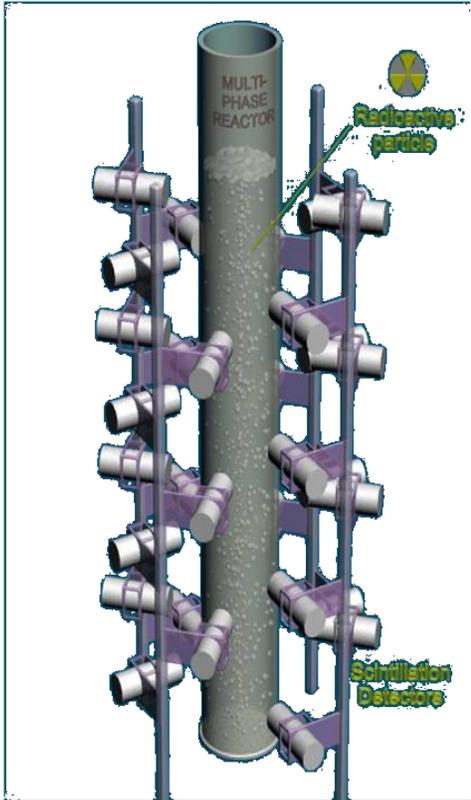


Figure 2 Schematic of a CARPT system.

The flow of data analysis is shown in Figure 3.

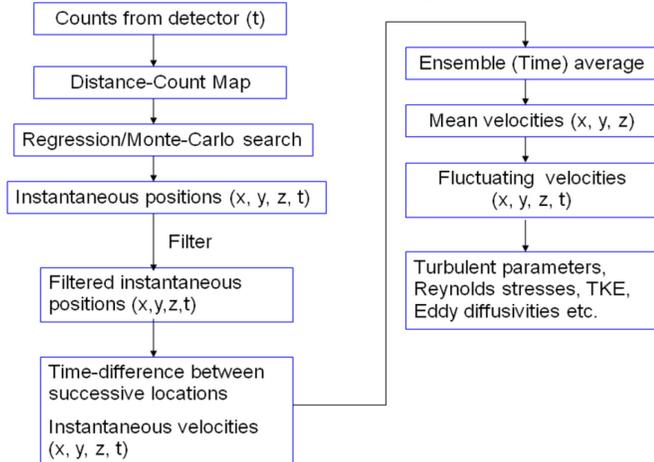


Figure 3 Flow of CARPT data analysis

III. HARDWARE PROBLEM AND SOLUTION

The Chemical & Biological Engineering Dept. at Missouri University of Science and Technology Rolla (MST) performs CARPT studies of several types of reactors. In early 2010 they contacted the first author of this paper about upgrading their system. Figure 4 shows a block diagram of their system at that time:

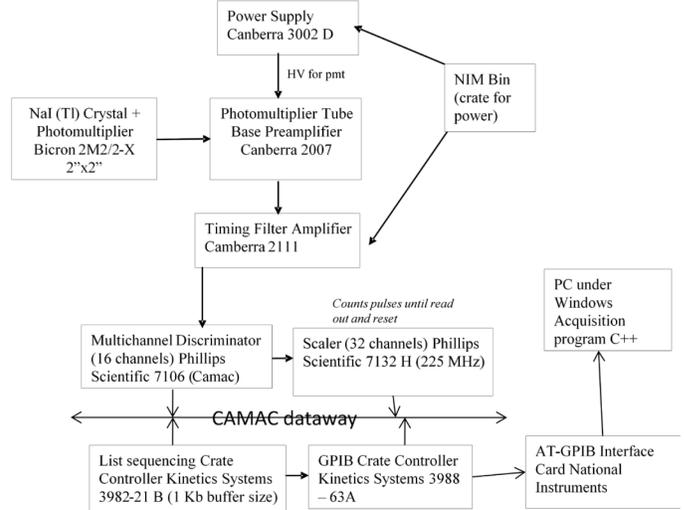


Figure 4 Original system block diagram.

Of concern to the MST group were the bottom three elements of Figure 4. Those devices are no longer manufactured. Additionally, the software on the Windows PC was not sufficiently flexible, nor reliable.

CAMAC [3] (Computer Automated Measurement and Control) is a very old instrumentation bus. It was first used in the Automotive industry adopted by the nuclear physics community about that time, and standardized for use in nuclear physics data acquisition systems in 1972 (ESONE/EUR 4100). The IEEE standardized the system, bus controllers and host interfaces and software APIs for FORTRAN Programs in 1982 (IEEE standards 585,683, 596, 595, 726, 675, 758 inter alia).

The goal of the project was to replace and modernize the CAMAC readout system/host interface and computers while retaining the existing analog electronics and CAMAC instrumentation. While it would have been nice to update the computer interfaces from CAMAC to something more recent, that was beyond the budgetary scope and capabilities of this project.

Many smaller nuclear physics labs have a significant investment in CAMAC electronics and are “trapped” in this technology either because they are insufficiently funded to replace their electronics or they don’t have sufficient on-site expertise to convert their software once they do replace their hardware. To meet the needs of those laboratories, Weiner Plein & Baus Elektronik markets a USB CAMAC crate controller the CC-USB. Thanks to modern gate array technology, this module provides much of the functionality of

the three CAMAC interface modules used in the original system in a single simply laid out module.

A block diagram of the CC-USB is shown in Figure 5. This is taken from the module's manuals. The block referred to as "Stacks" in the figure allows operation lists to be downloaded and triggered by various conditions.

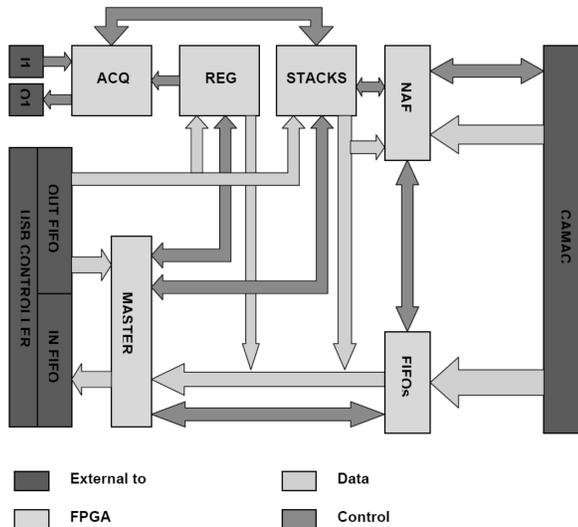


Figure 5 Block diagram of CC-USB

The specifications require that scaler readouts be triggered at relatively stable timed intervals. The CC-USB contains onboard resources that allow a periodic pulse to be generated on one of its outputs. The stability of this timing signal is on the order of one part in 12.5ns which is far superior to the required timing. The periodic output pulse is then cabled to the input that triggers execution of the CC-USB primary event readout list.

The CC-USB connects to a host computer via a USB A to B cable, such as that used to connect many USB printers.

IV. SOFTWARE

This section is subdivided as follows:

- First an overview of the requirements of the software is presented.
- Second a description of the NSCL DAQ CC-USB readout software is given.
- The modifications performed on the base software are described along with some of the operational characteristics of the resulting software.

A. Software requirements

In order to understand the software requirements and data acquisition modes it is important to know a bit about the electronics. Each gamma ray interacts with the NaI in the detector to produce light. This light is collected, amplified turned into an electrical pulse by means of a photo multiplier tube. The resulting electrical pulse is then electronically amplified. If the pulse rises above a threshold, it is counted

by a channel in the scaler module. The number of counts in a scaler channel per unit time represents the intensity of the source as seen by each detector. The algorithms in Figure 3 can map those intensities to positions given position./intensity map.

In production data runs, the source is moving, this leads to the common mode of data taking. The scalers are allowed to count for some dwell time, and then read and cleared. The output from a production run is a file that consists of a record of the scaler counts for each dwell period and the length of the dwell period. The precision to which the positions and velocities of the source can be reconstructed therefore depends on the dwell time as well as other factors.

In addition to production runs there are two other modes of data taking:

- Threshold hunt mode.
- Position calibration mode.

NaI detectors are sensitive to gamma-rays as well as cosmic background and the gamma rays emitted by background sources. It is the leading edge discriminator (LED) that determines which pulses are counted and which are not. The threshold hunt mode provides information to the experimenters to help them set the LED thresholds. In this mode, a source is placed where all detectors can see it. After each dwell time the discriminator threshold is incremented. Differentiating the counts with respect to the dwell time produces a crude spectrum with discriminator settings as the channel number. The position of the peak(s) due to the source provides the discriminator settings.

The original system required that the output file from threshold hunt data taking be fed into software that then did the peak location. The requirements for the new system were to produce the same output file but also provide support for visualizing spectra constructed from these data so that the results of the peak finding software could be double checked.

Position calibration mode is required to produce intensity to position maps. In this mode of data taking, a source is placed at a known position in the system with the detectors mounted in their production data taking positions. After each dwell time the source is moved to a different position. The output of this mode is a file that contains a map of counts as a function of position. These data are used to reconstruct positions from production runs. Having a correct setting for the LED is a precondition for this sort of run.

B. NSCL DAQ CC-USB base software

In [4] I reported on a software system that provides a domain specific language to describe nuclear physics experiments to data acquisition software. The system described in [4] interfaced to digitizers resident in a VME [5] backplane via a USB-VME controller. What was not mentioned in that paper was that a similar system was also written that interfaces to

CAMAC via the CC-USB. This CAMAC system was used as the starting point for the CARPT data acquisition system.

This section focuses on the structure of the Readout software, shown in Figure 6.

Readout Software Structure

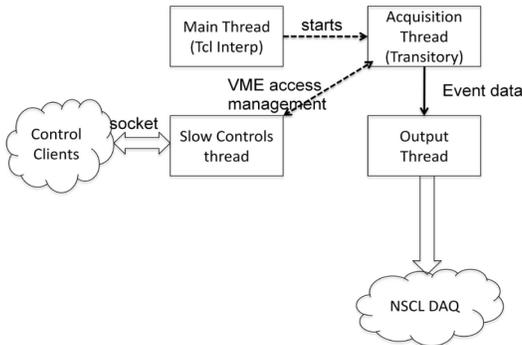


Figure 6 Structure of original CCUSB readout

The main thread embeds a Tcl interpreter with a few additional commands to control data taking. If the user requests that a data taking run start, the acquisition thread is started. The acquisition thread processes the configuration file through an extended captive Tcl interpreter, sets up the CC-USB, downloads readout lists and starts taking data.

Data from the CC-USB come in 8Kbyte buffers. As each buffer is received from the CC-USB it is passed to the output thread. The output thread normally does some light weight reformatting so that the resulting buffer is correctly structured for the NSCL data acquisition system. The reformatted buffer is sent to the NSCL DAQ data distribution server which makes it available to interested clients.

The slow controls thread runs an additional Tcl interpreter. It accepts TCP/IP connections and performs CAMAC operations on behalf of the client. If acquisition is active the slow controls thread requests a pause in data taking, performs the requested operation then resumes data taking.

This Slow controls thread is required because on Linux, USB programming via libusb allows only one process to access each USB device. Furthermore, single shot CAMAC operations cannot be executed on the CC-USB when data taking is in progress.

C. Modifications to the base software.

Figure 7 shows the modified structure of the Readout program.

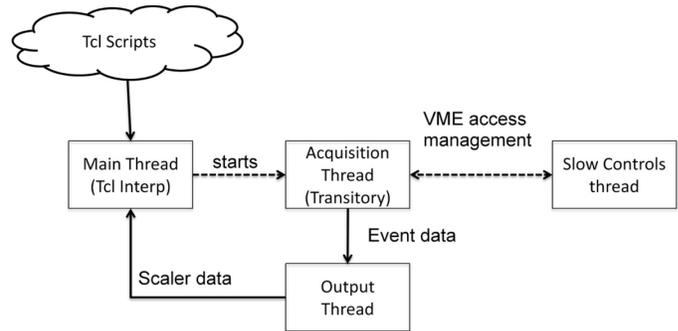


Figure 7 Structure of modified software.

The major difference between Fig. 6 and Fig 7 is that in Fig 7, the output thread has been rewritten to queue Tcl events back to the main thread's interpreter. The Tcl_Event struct used has been extended to hold the raw scaler data.

When the interpreter in the main thread dispatches the event, the event handler reformats the raw scaler data into a list of Tcl lists. Each inner list is the set of scaler values. Each outer list one dwell time worth of data (several scaler dwell times fit in a single VM-USB buffer). The resulting list is passed to the Tcl proc `onEvent` which performs mode dependent processing of the data.

This architecture:

1. Removes the need to run the NSCL data acquisition system, which is serious overkill for this application.
2. Allows the online processing of scaler data to be done in Tcl scripts.

The Tcl scripts sourced into the main thread's interpreter interact with the slow controls thread to set discriminator thresholds and output widths. These scripts provide GUI's that interact with the slow controls thread over a Tcp/IP socket allowing that part of the architecture to remain unchanged (normally a separate GUI process interacts with the slow controls thread).

Tcl scripts were then written for each of the three modes of data taking. A 'master' Tcl script allows the user to select which operating mode they want to run. In the case of the discriminator hunt mode, Plotchart [6] was used to display the spectrum and a cursor position indicator was provided to allow the users to read the discriminator threshold setting directly from the resulting spectrum.

Figure 8 below shows a spectrum from the threshold scan mode when a ^{137}Cs and a ^{60}Co source were both counted by the detectors. Note that ^{60}Co emits γ -rays at both 1.3 and 1.7MeV providing a pair of peaks close together in the energy spectrum.

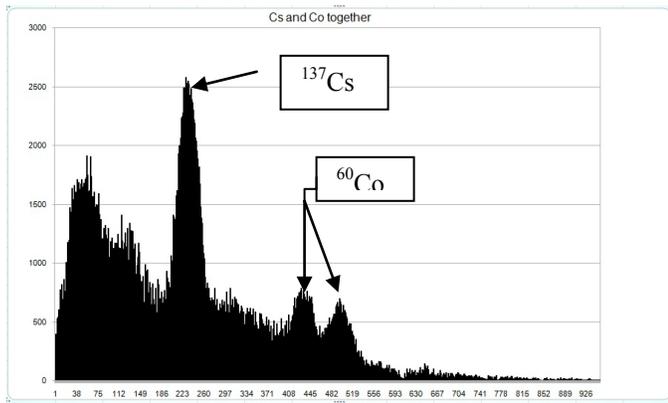


Figure 8 Spectrum from threshold hunt mode.

V. CONCLUSIONS

Tcl provided an ideal platform on which to develop the software for this project. Since the data rates are quite low (typical dwell times are a few seconds, each data point delivering 32 x 32 bits of data), Tcl scripts were more than equal to the task of handling these data, even in the more challenging threshold scan run where data were actually analyzed online.

The original structure of the readout software made the modifications required to handle the data at the script level relatively easy. Once those modifications were written, development went at the speed of Tcl. Thanks to Arjen Markus's Plotchart package, providing the spectra of the threshold hunt mode could be easily displayed and trivial mouse motion event handlers written to allow the user to pick off the correct LED thresholds from the plots.

VI. REFERENCES

- [1] *Opaque Multiphase Reactors: Experimentation, Modeling and Troubleshooting* M.P. Dudukovich *Oil & Gas Science and Technology Rev. IFP V 55 (2000) #2* pp 135-158.
- [2] *Final Report: Flow mapping in a Gas-Solid Riser via Computer Automated Radioactive Particle Tracking (CARPT)* M. Al-Dahan et al. Washington University
- [3] ANSI/IEEE *IEEE Standard Modular Instrumentation and Digital Interface System (CAMAC) 585* 1982 IEEE
- [4] *A Domain Specific Language for defining Nuclear Physics Experiments* Ron Fox *Proceedings of the 15'th Annual Tcl/Tk Conference* Tcl Association Press. ISBN 978-0-578-00296-5.
- [5] VME IEEE 1014-1987 (note more common is VME 64 which is specified by ANSI/VITA 1-1994).
- [6] *Plotchart* Arjen Markus <http://wiki.tcl.tk/11265>