

Neutron-scattering data acquisition and experiment automation with Python

Piotr A. Zolnierczuk (zolnierczukp@ornl.gov) – Oak Ridge National Lab, USA

Richard E. Riedel (riedelra@ornl.gov) – Oak Ridge National Lab, USA

PyDas is a set of Python modules that are used to integrate various components of the Data Acquisition System at Spallation Neutron Source (SNS). PyDas enables customized automation of neutron scattering experiments in a rapid and flexible manner. It provides wxPython-based GUIs for routine experiments as well as IPython command line scripting environment. Matplotlib and NumPy are used for data presentation and simple analysis. PyDas is currently used on a number of SNS instruments and plans exist to deploy it on new ones as they come on-line.

Introduction

The neutron is a useful probe of matter as the wavelengths of the so called cold neutrons¹ are on the order of inter-atomic distances in solids and liquids [SQU]. It has no charge and therefore it can interact with the atomic nuclei, for example with the proton in hydrogen that is virtually transparent to the X-ray radiation. Moreover, the energies of cold neutrons are on the same order as many excitations in condensed matter. And finally the neutron possesses a magnetic moment which means that it can interact with the unpaired electrons in magnetic atoms.

Using neutrons scientists can glean details about the nature of materials ranging from liquid crystals to superconducting ceramics, from proteins to plastics, and from metals to micelles and metallic glass magnets [PYN].



Figure 1. Layout of the Spallation Neutron Source

¹Cold neutrons energies range from about 0.05 to about 25 meV, and their corresponding wavelengths range from 0.18 to 4.0 nm.

Basics of neutron scattering

Cold neutrons have to be produced either in a nuclear reactor or with the help of particle accelerators. The Spallation Neutron Source (SNS) at Oak Ridge National Laboratory, Tennessee is an accelerator-based neutron source that currently holds the Guinness World Record as the world most powerful pulsed spallation neutron source [ORN]. The neutrons are produced when a beam of very energetic protons (1GeV) bombards a liquid mercury target and in the process some neutrons are “spalled” or knocked out of mercury nucleus. Other neutrons are evaporated from the bombarded nucleus as it heats up. For every proton striking the nucleus about 20 to 30 neutrons are expelled. The neutrons emerging from the mercury target are too fast to be useful to study properties of materials. They need to be slowed down (or cooled) by passing through the moderator material - for example liquid hydrogen kept at the temperature of 20 K. The cold neutrons are then guided by a set of beam lines to specialized instruments.

The beam of neutrons then undergoes collimation, focusing and velocity selection (chopping) and is aimed at a sample. While many neutrons will pass through the sample, some will interact and bounce away at an angle. This scattering of neutrons yields important information about the positions, motions and magnetic properties of atoms in materials. Figure 2 shows a schematic lay-out of a neutron scattering experiment.

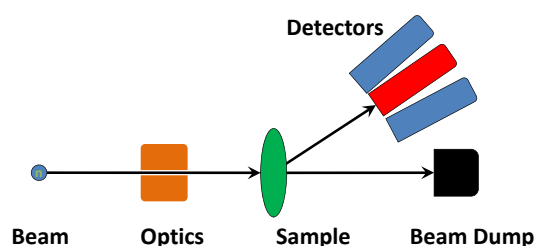


Figure 2. Basic Elements of a neutron scattering experiment

A number of devices and controls is required for a successful neutron scattering experiment. Such a set-up, conventionally called a neutron scattering instrument, needs first and foremost a neutron detector to record

the scattered neutrons. In most cases gaseous linear position-sensitive tubes or scintillator detectors are used [KNO]. In addition each instrument requires sample environment controls (temperature, pressure, and magnetic field), neutron beam optics controls (slits for collimation and choppers for velocity selection) and motors for mechanical alignment. An example of a real life neutron scattering instrument [CNC] is shown Figure 3.

A typical neutron experimental technique is called scanning. For example, a scientist chooses a set of temperatures that spans a phase transition in a sample. The experimental procedure then involves collection of a predefined amount of neutron data for each temperature set-point as neutron scattering experiments are often statistics limited, i.e. one has to record enough scattered neutrons in order to obtain useful results. Real-life experiments involve scanning of multiple variables, e.g. temperature, pressure, neutron polarization, etc.

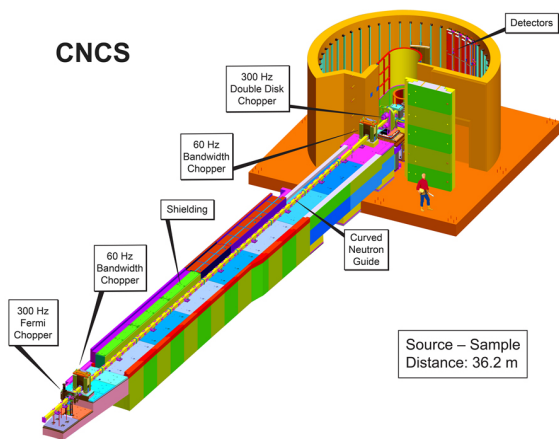


Figure 3. Example neutron scattering instrument: Cold Neutron Chopper Spectrometer [CNC] at SNS.

SNS Data Acquisition System

Such a complex instrumentation requires a very sophisticated data acquisition and control system that is usually custom designed and matches specifics of each neutron scattering facility. At SNS, the Data Acquisition System (DAS) [RIE] is a collection of real-time firmware running in FPGA-based boards, programs written in Microsoft Visual Studio C++ and National Instruments LabView [LAB] virtual instruments. Each piece of software typically performs one single task such as detector read-out or motor control and the programs communicate via TCP/IP networks. The SNS DAS is composed of four subsystems (see Figure 4):

- Real-Time System for detector electronics and read-out
- Timing System provides crucial timing information

- Slow Controls for sample environment, motors and ancillary equipment
- Control & Data for neutron data collection and main control

The Control Computer (see Figure 4) brings together information from each subsystem. It is designed to run a number of control programs that interact with its partners on the remote computers. For example motor control program communicates via TCP/IP with a motor satellite program.

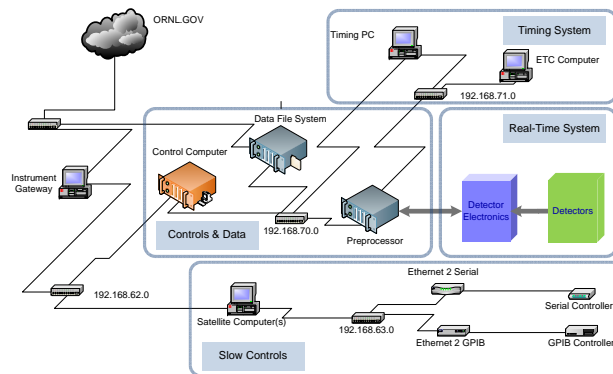


Figure 4. SNS DAS Network Architecture

PyDas - Python environment for SNS DAS

From the very start it was realized that the SNS DAS system needed a robust and convenient scripting environment. Python was chosen as a scripting language for its simplicity, elegance, power and ease to write extension modules. The SNS DAS scripting environment is called PyDas. It is designed to not only provide scripting capabilities but also the ability to integrate various experimental components into a single application.

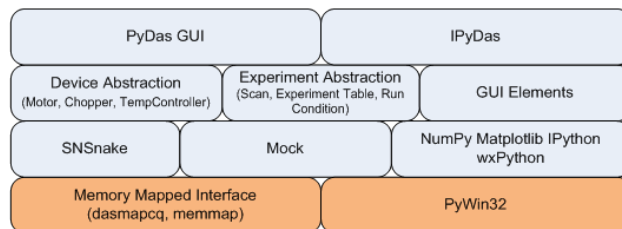


Figure 5. PyDas General Architecture

The general PyDas architecture is shown in Figure 5. Shared memory and Win32 API provide the interface to other DAS components. Two custom C++ modules (dasmapiq and memmap) and Mark Hammond’s pywin32 [HAM] module are at the foundations of PyDas. Next layer contains SNSnake.py and mock.py modules that “pythonize” the low-level interface. The

former provides access to real hardware, while the latter delivers the essential simulation (mock) environment. PyDas also uses standard scientific Python modules such as NumPy [NUM], Matplotlib [MAT] and IPython [IPY]. The GUI environment is provided by wxPython [WXP]. Above these base layers there are modules that abstract hardware devices (such as motors, choppers and temperature controllers), experiment procedures (such as scan, experiment table, run condition) and various GUI elements.

	SANGLE (Deg)	DANGLE (Deg)	Xsample (mm)	PolTrans (Deg)	PolAngle (Deg)	S1_Width (mm)	S2_Width (mm)	S3_Width (mm)	Flipper1	Flipper2	Magfield (T)	Temp (K)	Count Rate	Value
Point_0	1.00000	2.00000	2.450	-0.123	-4.500	5.649	10.062	5.961	ON	OFF	0.000	300.000	COUNTS	500000
Point_1	2.00000	4.00000	2.450	-0.123	-4.500	5.649	10.062	5.961	ON	OFF	0.000	300.000	COUNTS	500000
Point_2	3.00000	6.00000	2.450	-0.123	-4.500	5.649	10.062	5.961	ON	OFF	0.000	300.000	COUNTS	500000
Point_3	4.00000	8.00000	2.450	-0.123	-4.500	5.649	10.062	5.961	ON	OFF	0.000	300.000	COUNTS	500000
Point_4	5.00000	10.00000	2.450	-0.123	-4.500	5.649	10.062	5.961	ON	OFF	0.000	300.000	COUNTS	500000
Point_5	6.00000	12.00000	2.450	-0.123	-4.500	5.649	10.062	5.961	ON	OFF	0.000	300.000	COUNTS	500000
Point_6	7.00000	14.00000	2.450	-0.123	-4.500	5.649	10.062	5.961	ON	OFF	0.000	300.000	COUNTS	500000
Point_7	8.00000	16.00000	2.450	-0.123	-4.500	5.649	10.062	5.961	ON	OFF	0.000	300.000	COUNTS	500000
Point_8	9.00000	18.00000	2.450	-0.123	-4.500	5.649	10.062	5.961	ON	OFF	0.000	300.000	COUNTS	500000
Point_9	10.00000	20.00000	2.450	-0.123	-4.500	5.649	10.062	5.961	ON	OFF	0.000	300.000	COUNTS	500000
Point_10	11.00000	22.00000	2.450	-0.123	-4.500	5.649	10.062	5.961	ON	OFF	0.000	300.000	COUNTS	500000
Point_11														
Point_12														
Point_13														
Point_14														
Point_15														
Point_16														

Figure 6. PyDas GUI example

PyDas can be run in two basic modes: with graphical user interface (GUI) and via the command line. The PyDas GUI interface (see Figure 6) is used to run pre-defined experiments. Each instrument is provided with a customized PyDas GUI to meet instrument individual needs. The command line interface, called IPyDas, is based on the IPython shell embedded into a single wxPython window and allows for custom experiment scripting. Below, we present two short examples of IPyDas scanning scripts:

```
scan_motor('motor1', arange(10.0, 90.0, 0.5),
           scan.RUNTIME, 30.0, go_back=True)
scan_plot()

for temp in [273.0, 293.0, 313.0, 333.0]:
    tempctrl.value = temp
    scan_chopper('Energy', [25., 35., 50.],
                scan.PCHARGE, 1e10,
                title='The Nobel Prize Data at T=%s K' % temp)
```

Summary

In summary, we presented the complexities and opportunities of modern neutron scattering experiments and how PyDas helps to automate and integrate SNS

Data Acquisition System. The future challenges for PyDas lie in the requirement to support 24 different instruments with high level of customization. Based on lessons learned so far we are in the process of refactoring PyDas to better meet these challenges, provide live neutron data viewing (currently served by a C++ Measurement Studio application) and we are also exploring new Python technologies such as IronPython [IRP].

The authors would like to thank Xiaodong Tao for initiating the PyDas GUI project and members of the SNS DAS Group: Lloyd Clonts, Gayle Green, Andre Parizzi, Mariano Ruiz-Rodriguez and Madhan Sundaram for collaboration and help during the development. ORNL/SNS is managed by UT-Battelle, LLC, for the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

References

- [SQU] G.L. Squires, Introduction to the Theory of Thermal Neutron Scattering, Cambridge University Press (1978)
- [PYN] R. Pynn, Neutron Scattering, A Primer, http://neutrons.ornl.gov/science/ns_primer.pdf
- [ORN] ORNL Neutron Sciences, <http://neutrons.ornl.gov>
- [KNO] G.F. Knoll, Radiation Detection and Measurement, J.Wiley & Sons (2000)
- [CNC] Cold Neutron Chopper Spectrometer http://neutrons.ornl.gov/instrument_systems/beamline_05_cncc
- [RIE] R.E. Riedel, Overview of Data Acquisition at the SNS, talk given at NOBUGS 2004 conference, <http://lns00.psi.ch/nobugs2004>
- [LAB] National Instruments Labview, <http://www.ni.com/labview>
- [HAM] M. Hammond, A. Robinson, *Python Programming On Win32*, O'Reilly Media (2000), <http://pywin32.sourceforge.net>
- [NUM] T. Oliphant et al. NumPy, <http://numpy.scipy.org/>
- [MAT] J.D. Hunter, *Matplotlib: A 2D graphics environment*. Computing in Science and Engineering. 9: 90-95 (2007). <http://matplotlib.sourceforge.net>.
- [IPY] F. Perez and B. Granger: *IPython: a system for interactive scientific computing*, Computing in Science & Engineering 9(3) 21-29, 2007, <http://ipython.scipy.org>
- [WXP] WxPython, <http://www.wxpython.org/>
- [IRP] J. Huginin et al. IronPython, <http://www.codeplex.com/IronPython>