EXPERIMENT AUTOMATION WITH A ROBOT ARM USING THE LIQUIDS REFLECTOMETER INSTRUMENT AT THE SPALLATION NEUTRON SOURCE*

B. Vacaliuc†, P. A. Zolnierczuk, M. Sundaram, A. A. Parizzi, C. E. Halbert,
M. C. Hoffmann, G. C. Greene, J. F. Browning, and John Francis Ankner
Oak Ridge National Laboratory, Oak Ridge, TN 37831 USA

Abstract

The Liquids Reflectometer (LR) Instrument installed at the Spallation Neutron Source (SNS) enables observations of chemical kinetics, solid-state reactions and phase-transitions of thin film materials at both solid and liquid surfaces [1]. Effective measurement of these behaviors requires each sample to be calibrated dynamically using the neutron beam and the data acquisition system in a feedback loop. Since the SNS is an intense neutron source, the time needed to perform the measurement can be the same as the alignment process, leading to a labor-intensive operation that is exhausting to users. An update to the instrument control system, completed in March 2013 [2], implemented the key features of automated sample alignment and robot-driven sample management, allowing for unattended operation over extended periods, lasting as long as 20 hours. We present a case study of the effort, detailing the mechanical, electrical and software modifications that were made as well as the lessons learned during the integration, verification and testing process.

INTRODUCTION

The Liquids Reflectometer [1], installed as one of the first instruments at the Spallation Neutron Source, has now been functional for over six years. This instrument is designed to view liquid and solid surfaces in specular, off-specular, and near-surface small angle scattering geometries [3]. A typical experiment on LR begins with mounting a sample, then aligning it by adjusting the sample height and two axes of rotation under observation of the neutron beam. The alignment process was described in greater detail in a previous work [2].

Figure 1 shows the equipment present in the sample environment cave. Once the sample and the cave area are secured, the sample is aligned by adjusting the sample stage goniometer positions under observation of the reflected neutron beam. Following alignment, a series of data collections are performed with the detector and sample at a variety of geometries according to the specific type of reflectivity profile that is sought.

The LR instrument can be configured such that the possible scattering paths shown in Figure 2 can be isolated

Figure 1: Shows the sample table (center, low), the detector assembly (left), the robot arm (center, yellow), the neutron beam orifice (right, back) and user (Jim).

and detected individually by appropriate motions of the incident, the sample and detector positioning goniometers. Coupling sample positioning with the ability to select neutrons of different wavelengths (λ) and scan over Q by adjusting θ, the incident beam angle, makes the LR instrument a most versatile tool for characterizing surface features of a very wide variety of materials.

Automation

Operating the LR instrument is a labor intensive process, taking approximately 1-3 hours to mount, align and then collect the neutron counts over the several positions and wavelengths. Since the sample stage has multiple axis of freedom, incorporating a sample changer presents a unique challenge. A robot arm, as shown in Fig. 1, was commissioned as part of the 2012 summer maintenance cycle. The arm is able to translate samples from one location to another without coupling to any mechanical axes of the instrument. Controls for the robot arm are integrated into the DAS system and are exposed as process variables by the motor control application. They provide the functions of: i) moving and restoring a collection of selected instrument motors, ii) commanding the robot to move samples between the instrument and the sample carrier cells, iii) ensuring safe operation of the robot by implementing interlocks between motor movements and the robot operations.

Since these functions are implemented as virtual motors, integration of the robot functions into the Python-based user interface "PyDas" [4] was straightforward.

* Work supported by Oak Ridge National Laboratory, managed by UT-Batelle, LLC for the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

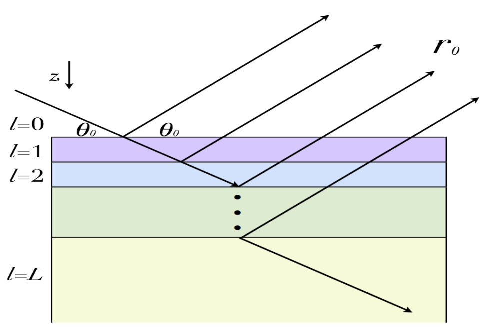
† vacaliucb@ornl.gov

ISBN 978-3-95450-139-7

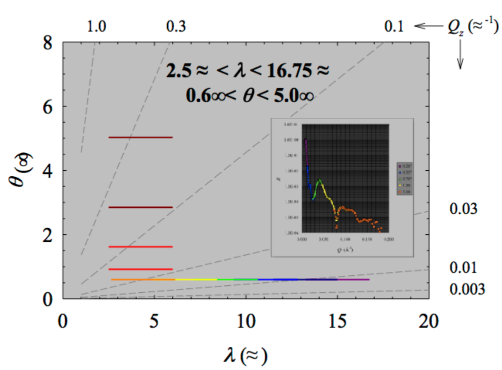
ROBOT SYSTEM

Figure 3 illustrates the hardware components of the robot control system. A robot arm is driven by a high power controller. The controller accepts movement commands over an ethernet network that are formatted as XML strings. A separate PC/104 embedded controller contains a position sequencer that converts a single sample select process variable into a sequence of points.

The sample stage position optimal for pickup is different than the position used for alignment and experiment data taking. Since the sample stage will translate each time a new sample is to be loaded, it is necessary to verify that the sample stage is in the correct position. This is accomplished by a two-camera vision system that is focused on two targets that are mounted on the sample stage plate. The vision system is connected to the Personnel Protection System (PPS) that governs the permit signals to all the motors to prevent any motion under unauthorized conditions, such as cave door open, that could lead to unsafe operation.



(a) Neutron beam reflection from surface layers



(b) Angle vs. Wavelength

Figure 2: Shows the how a reflectivity profile is constructed on the LR instrument by illuminating the sample with a neutron beam (from left) at specific geometries. (a) Illustrates the possible paths that neutrons can take in a layered sample. The relation $Q = (4\pi/\lambda)\sin\theta$ holds, where λ is the neutron wavelength and θ is the incidence/reflected angle. (b) Shows the series of data collections that are needed (colors) overlaid over lines of constant-Q. These compose a complete reflectivity profile (inset).

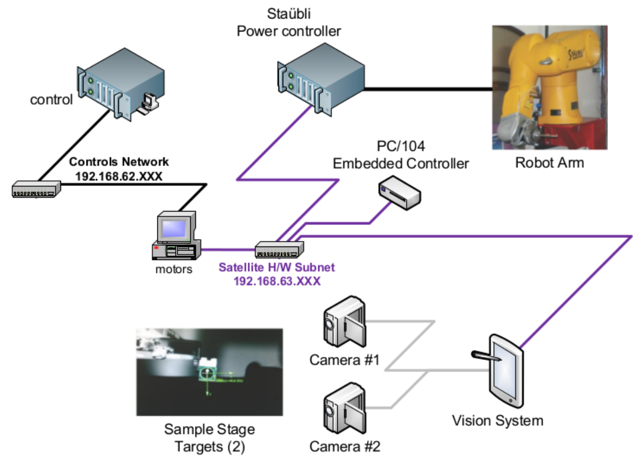


Figure 3: Robot system physical component diagram showing Staübli controller, PC/104 sequencer, vision system components, PPS and interlock system, and the motors computer.

Control System Design

Every movement of the robot arm is an action in two abstract steps: a pick and a put. Figure 4 shows the basic sequence of motions that are pre-programmed into the robot arm motion control unit. The robot controller interpolates the movement of the arm between these points. The points themselves are all referenced to a particular point in space, and the individual sample magazine positions are calculated as *offsets* to the pick/place locations according to the selected value. In this way, the waypoints are defined once, and all the other possible positions can be parametrically determined.

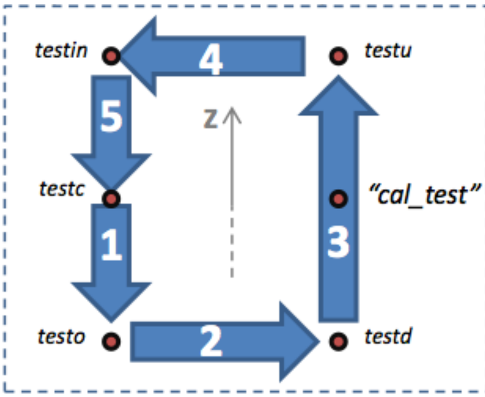
The robot coordinate system as shown in Figure 4(b) is used to compute fixed positions that are used in the sequencer. The 5 fixed positions are the sample landing position, the sample park position, the magazine park position, the sample magazine calibration position and the robot park position. With these 5 fixed position in space, any combination of samples from the magazine can be accommodated with just the variable positions described for the offsets between the magazine slots and the calibration position. These calculations and the communication to and from the robot arm power controller are performed by the PC/104 embedded controller.

Control System Interface

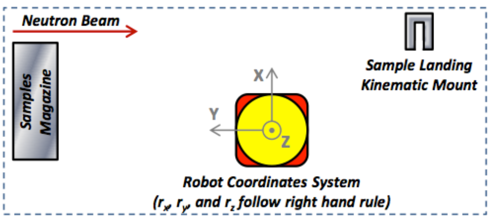
The control system interface follows the model described in [5]. In this system, the process variable distribution system is based on National Instruments DataSockets. C++ applications (CPA) on the control computer implement a shared memory and perform the work of keeping the values synchronized. On the remote end, C++ applications (Satellite) act as device drivers to translate the DataSockets protocol into device-specific commands.

The interface produces three key process variables as

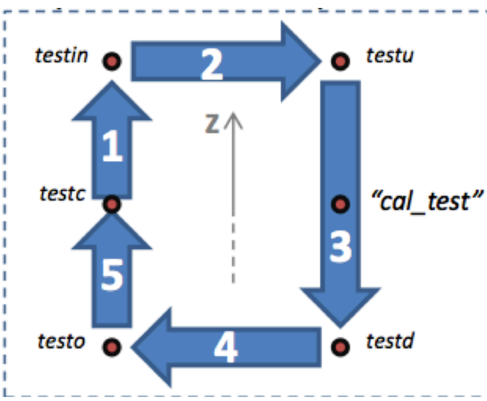
ISBN 978-3-95450-139-7



(a) Pick Function Move Sequence



(b) LR Robot Coordinate System



(c) Put Function Move Sequence

Figure 4: Describes the two fundamental motions that are programmed into the robot control sequencer. These motions are always executed in pairs: pick, put. (a) The motion to pick a sample (sample holder or sample magazine) and deliver it to a neutral position. (b) Graphical view of robot coordinate system in which the waypoints are described in three dimensional space. (c) The motion to take a sample from neutral position and place it onto a destination (sample holder or sample magazine).

shown in Fig 5. A variable that controls the enabling/disabling of the robot motion is used to interlock movement of the goniometers and positioning motors with the action of the robot sample changer. It effectively controls a mutex that is shared between two Satellite applications. On the motor side, a set of "preset" motor positions is kept to allow scripting to easily move to the sample pickup position and return to the sample analysis position. A method of memorizing the current position is implemented in a similar way to a calculator's MEMORY function. This

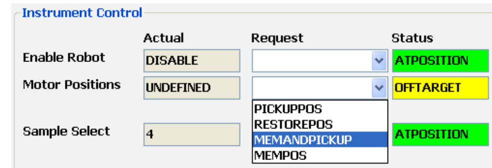


Figure 5: Illustrates the robot-related process variables available to PyDas simplifying the experiment control intrface

is used by instrument staff to define the "starting" position for sample alignment and can vary between experiments and sample carriers used by a user. The names of the motors memorized are defined in a configuration file used by the motors Satellite application. Finally, a sample select variable initiates the actions of the robot to move samples to and from the sample stage and the sample magazine.

PRODUCTIVITY

Table 1 captures several metrics that were obtained from historical analysis of the performance of the instrument for two run cycles. In the 2012-A cycle, all alignments were performed manually, the process of which was detailed in [2]. A trained user can observe a global maximum soon after it is scanned by the neutron beam, and so can use the software to "fit and continue" before the total number of positions have been collected. An algorithm must (*generally*) collect all the points before fitting, and so we see the automatic implementation taking somewhat longer.

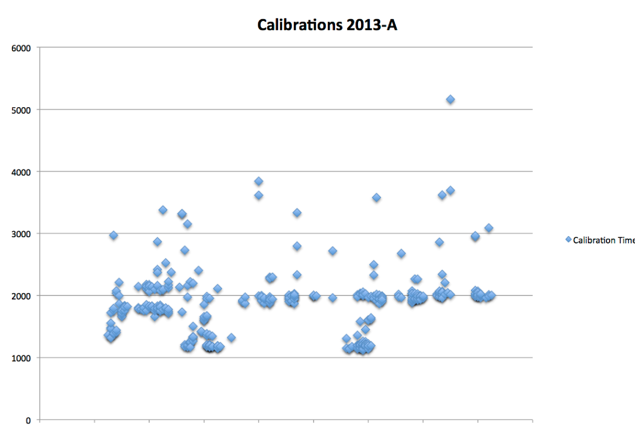


Figure 6: Shows time required (in seconds) to perform automatic alignment from January thru May 2013.

Figure 6 plots all the calibrations made during the 2013-A run cycle. Variability between the runs is generally attributed to the need for additional axis scans in order to converge on the global maximum, with some amount of instrument control system related delays occurring sporadically.

Following the end of the 2013-A run cycle, an effort was undertaken to analyze the source of the slow alignment

ISBN 978-3-95450-139-7

Table 1: Compares Several Kinds of Metrics Between run Cycle 2012-A (January thru May) which did not have the Robot and Alignment Integration, and run Cycle 2013-A in which the Robot was Commissioned.

Productivity Metrics			
Metric	2012-A	2013-A	Notes
Number of neutron runs that could have used the current robotic sample changer	2696	3805	Number of neutron runs that *did* use the robotic sample changer
Number of days used to run experiments	30 days	51 days	
Average energy on target/experiment day	16.98 MWh/day	16.20 MWh/day	*includes* downtime during experiments scheduled and non-scheduled
Average number of runs/energy on target	5.29 runs/MWh	4.61 runs/MWh	
Estimate average MWh used per sample	1.52 MWh/sample	1.76 MWh/sample	based on 8 runs/sample (typical)
Typical time of manual alignment scan	14 min/sample	32 min/sample	Average time of automatic alignment
Energy equivalent spent for alignment	165 kWh/sample	360 kWh/sample	
Future Projections			
		2013-B	
Benchmarked "markers"-based alignment time		12 min/sample	
Energy equivalent estimate using "markers"-based alignment and accelerator at 875 kW		*135 kWh/sample*	
Expected number of runs/energy on target using "markers"-based alignments		*5.29 runs/MWh*	

performance, and these were addressed during the 2013 summer maintenance; implementation of a "marker"-based scanning system that avoids acquisition start and stop, as well as repair of a memory leak in a core DAS program. These enhancements are now in production for the 2013-B run cycle and Table 1 describes "Future Projections" regarding what productivity may be achieved with them.

One of the lessons learned in programming the vision system, was the need to schedule time with the vendor's field application engineers (FAE). Originally, the vision system was programmed based on a study of the user manual and the system was setup to identify several metrics on the target as shown in Fig. 3. This worked well during development, but exhaustive sample change testing showed a sensitivity to shadowing (loss of contrast) of the illuminator that proved insufficiently tolerant.

During a meeting with the FAE, the engineers were advised not to use certain features of the system, and to use only a certain type of locator tool based on pattern matching. The FAE had intimate knowledge of the strengths and weaknesses of the software, as well as an understanding of which aspects would work well in a given situation. Using the FAE recommended metrics, enabled the vision system to demonstrate a marked repeatability in the classification leading to far fewer vision system trips that required user intervention.

Finally, the vision system cameras are sensitive to position. A metal shield was designed to protect the cameras and prevent them being "bumped" during maintenance activity.

ACKNOWLEDGEMENT

The authors wish to acknowledge valuable contributions made by Lloyd G. Clonts, Xiaodong Tao and Rick Reidel for their original work on the DAS control system. Robert Viola and "Cooper" at Square-One Systems for the design and programming of the robot arm. Brad Lokitz, Juan Pablo Hinestrosa, and Mike Kilbey for being the "first" users of the automated robotic experiment processing system.

REFERENCES

[1] LR- liquids reflectometer. http://neutrons.ornl.gov/instruments/SNS/LR/, 2006. [Online; accessed 09-Sep-2012].

[2] P.A. Zolnierczuk, B. Vacaliuc, M. Sundaram, A.A. Parizzi, C.E. Halbert, M.C. Hoffmann, J.F. Browning, and J.F. Ankner. "Old wine in new wineskins:" Upgrading the liquids reflectometer instrument user control software at the Spallation Neutron Source. In *Future of Instrumentation International Workshop (FIIW), 2012*, pages 1–4, 2012.

[3] Ankner J. F., Tao X., Halbert C. E., Browning J. F., Kilbey II S. M., Swader O. A., Admun M. S., Kharlampieva E., and Sukhishvili S. A. The SNS liquids reflectometer. *Neutron News*, 19(14):14 –16, 2008. Available at http://dx.doi.org/10.1080/10448630802210545.

[4] P.A. Zolnierczuk and R.A. Riedel. Neutron scattering experiment automation with python. In *Real Time Conference (RT), 2010 17th IEEE-NPSS*, pages 1 –3, May 2010.

[5] Richard Reidel. Overview of data acquisition at the SNS. In *Fifth NOBUGS Conference, 2004*, Paul Scherrer Institut, Villigen, Switzerland, October 2004. Available at http://lns00.psi.ch/nobugs2004/papers/paper00055.pdf.