

XAL – THE SNS APPLICATION PROGRAMMING INFRASTRUCTURE*

J. Galambos, C., M. Chu, M. S. Cousineau, T. Pelaia, A. Shishlo
SNS, Oak Ridge National Laboratory, C.K. Allen, C. McChesney
Los Alamos National Laboratory, I. Kriznar, A. Zupanc
Cosylab, Ljubljana Slovenia, W.D. Klotz, ESRF

Abstract

A Java programming infrastructure for high level applications has been developed and is being used for the Spallation Neutron Source (SNS). The framework provides a hierarchical view of the accelerator and hides much of the underlying control system details. The hierarchy is database configured, facilitating application sharing across different beamlines, shielding the programmer from detailed knowledge of signal names, and allowing wholesale updating of applications. An important aspect of the framework is an online model, which can be run for design values, live machine values or user selected tuning values.

1 INTRODUCTION

The XAL application-programming interface [1] has been developed at the Spallation Neutron Source and is being applied in the initial beam commissioning. This framework offers a Java programming interface for high level physics applications and includes a number of features and tools to facilitate quick application development. Some of these features are discussed in Ref. 2, such as the accelerator hierarchy, the connection to the EPICS [3] control system, configuration from a database, and the application GUI framework for a quick start in application development. This paper concentrates on more recent developments such as the online model enhancements, a client service capability and the addition of new applications. These areas are discussed in the following sections.

2 ONLINE MODEL DEVELOPEMENTS

The XAL beam model [4] is a simple envelope or particle tracker designed for use in applications. It can be set up using either design values, machine settings or user supplied values. Recently there have been two main areas of work with the model: applying the online model to data analysis, and extension of the model to transport lines and the ring.

2.1 Phase and Energy Advance Calculations

The original XAL acceleration model was similar to Trace 3D, in that the beam was assumed to always enter a cavity at the design phase. During commissioning, when the initial klystron phase and amplitude settings are not known, calculating both the phase and energy advance is

important for comparison with experimental results. We have implemented the method described in Ref 5. With this algorithm, phase differences between upstream Beam Position Monitors (BPMs) can be predicted, based on the beam initial conditions and downstream accelerating cavity parameters. This algorithm has been initially tested with data taken in the last commissioning run for the first two Drift Tube Linac (DTL) tanks. An example application is shown in Figure 1, which shows the measured phase difference between two downstream BPMs vs. the DTL phase, at several amplitude settings. The model is used inside an XAL optimization package to solve for the initial beam energy, the three different klystron amplitudes, and the phase calibration between the beam and klystron to match the measured data. The results shown here were done in a Jython [6] scripting environment, using XAL features from the online model, optimization and plotting packages.

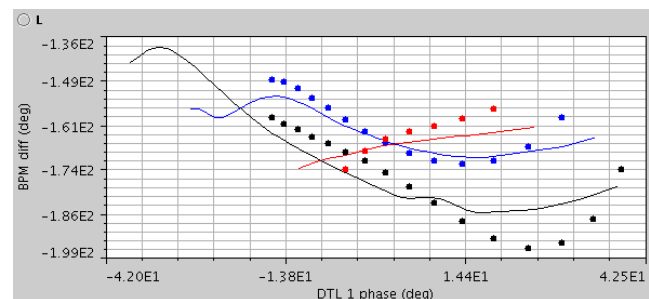


Fig.1. Phase difference between two BPMs downstream of Tank 1, vs. tank 1 klystron phase. Dots are model results, curves are measured data. Blue = nominal amplitude setting, red = 95%, black = 105%.

2.2 Twiss Parameter Calculations

Another application of data analysis with the online model is in the transverse phase space analysis of wire profile measurements. Wire profile data was taken in the Medium Energy Beam Transport (MEBT) beam line, for a variety of quadrupole settings. For each quadrupole setting, the calculated beam size at each wire is matched to the measured value, by varying the initial alpha and beta Twiss parameters. This procedure helps determine the beam conditions leaving the SNS RFQ and entering the MEBT. An example of the matching of the model predicted beam size to the measured values for the MEBT section, for one quadrupole setting is shown in Fig. 2. This analysis is also presently done with a Jython script.

*SNS is managed by UT-Battelle, LLC, under contract DE-AC05-00OR22725 for the U.S. Department of Energy. SNS is a partnership of six national laboratories: Argonne, Brookhaven, Jefferson, Lawrence Berkeley, Los Alamos, and Oak Ridge.

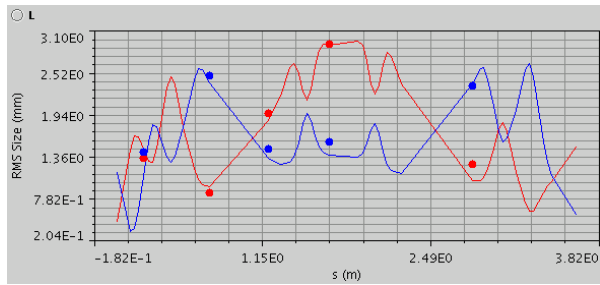


Fig.2. Beam size along the MEBT section. Dots are measured values and the curves are model predictions after solving for the initial Twiss parameters. Red = horizontal, blue = vertical.

2.3 Ring modeling

The online model was originally developed for modeling linac beams, but has been extended to model transport lines and rings. We intend to use the same online model for all of the SNS accelerator beamlines, to minimize redundant code development. For the ring modeling, space charge effects are ignored and a closed orbit solution is implemented. Many features developed for the linac beam modeling, such as database beam line configuration and automatic lattice generation are reused. Figure 3 shows an example of ring Twiss parameters around the SNS Ring calculated with the XAL online model, using design values. These results compare favorably with those predicted by MAD.

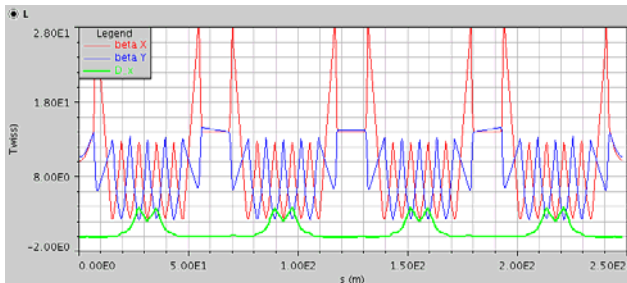


Fig. 3. Horizontal beta (red), vertical beta (blue) and dispersion (green) predicted by the online model for the SNS Ring.

3 SERVICES

A new area of work in XAL is the addition of service capability [7]. This feature allows applications to communicate with each other (in addition to the usual communication with control system provided signals). We have adopted an off-the-shelf approach to providing services. We use Rendezvous for service discovery. For actual passing of information we use XML-RPC. These are widely used standards and have simple interfaces.

One example use of the services is in the application framework – the base class for GUI applications. The framework provides a built-in service so that each application advertises its existence and a separate

“Application Viewer” application can monitor XAL application usage (and for example kill applications).

An important use of services is providing background services, ones that should not be killed by the closing of a single client program. One such case is our Machine Protection System (MPS) post-mortem application. Originally a standalone client application, this has been migrated to use the service framework. This service is always running in the background monitoring MPS events – capturing the stream of signals that emanate from each trip and sorting them to determine the root cause of each trip. It also provides statistics and views of the MPS trip history. Any number of client applications can view this data. Figure 4 shows a client view of the MPS trip history.

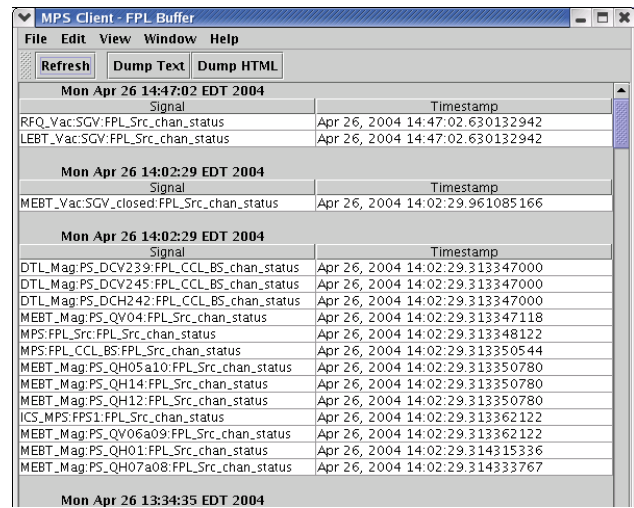


Fig.4. History log view from a Machine Protection System client application.

Another service application is the XAL “PV logger”. This application logs sets of control system signal values to a database, at specified intervals. It can also be triggered manually. One example use is to grab machine settings directly used by accelerator physics such as magnet, RF and BPM values. This provides complete sets of information needed to configure the online model, taken by a background process.

4 APPLICATIONS

There are now more than twenty XAL applications. They are all extensions of the application framework, which provides a simple GUI frame with commonly used menu items, out of the box. The framework follows the document pattern – a single application can have many open documents, each with a different save-able setup. Common features provided by the framework include open/save capability for documents, accelerator sequence browsing, and console error logging. This base provides developers a quick start, and offers users a common look

and feel for applications. A few sample applications are described briefly below.

4.1 The Online Model

The online model application (fig. 5) offers a GUI interface to the online model. The user selects a portion of the accelerator to analyze, selects a beam probe file (editable) containing the beam initial conditions, and selects whether to use live machine values or design values. Calculated results include Twiss parameters, as well as beam trajectories. Additionally there are feature to compare calculated trajectories to BPM values, and to display differences between the previous calculation. Comparing trajectory results before and after changing a dipole corrector provides orbit-difference capability.

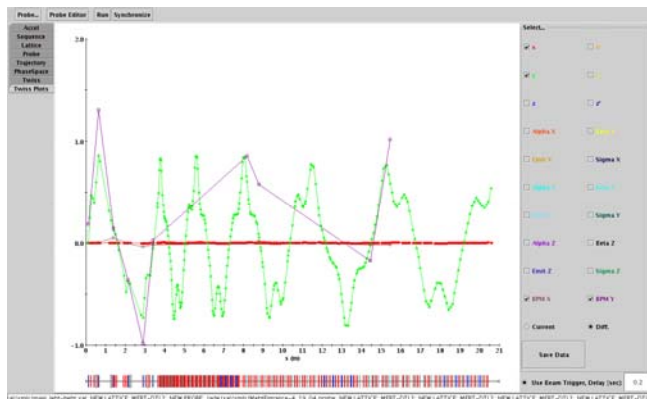


Fig. 5 The online model application run in the orbit difference mode - showing difference between the calculated position before and after a corrector change as well as the observed change in BPM positions.

4.5 Scan Application

The 1-D scan application (Fig. 6) provides a quick way to scan one quantity and monitor others. The user can specify the setup conditions for the scan (step size, averaging, etc.), and the signal names to monitor. Filtered data acquisition is possible based on the behavior of additional specified signals. Also a number of data analysis tools are provided such as fitting, locating curve intersections, locating curve minima and maxima, etc. A number of commonly used scan procedures are provided as permanent setups. A 2-D version of this application is also available to facilitate parametric scans.

4.5 Correlator

The correlator application (Fig. 7) is a general purpose application that monitors up to 3 signals, and displays their values against each other. For three signals, a color rendering in the third plane is used. A correlation tool ensures the signals values are from the same pulse. This application allows users to look for unanticipated cause-

effect phenomena. Features include data curve fitting, and data export.

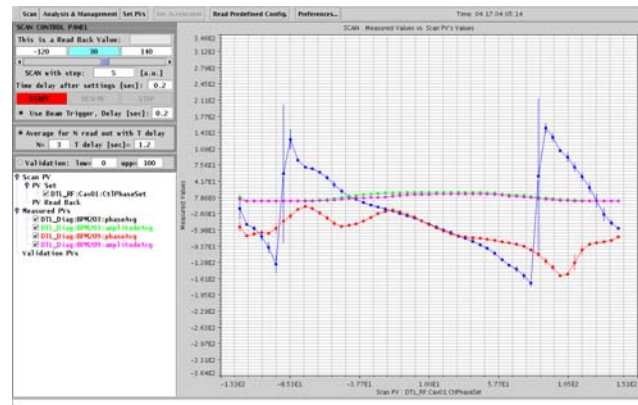


Fig. 6. Example use of the 1-D scan application, varying a klystron phase and monitoring BPM phases and amplitudes.

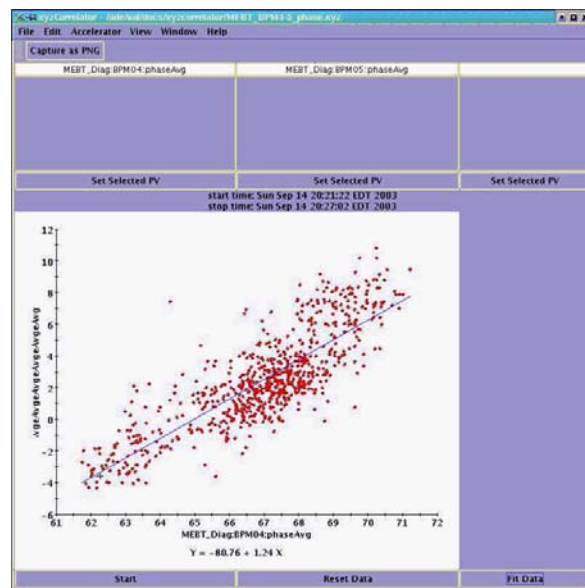


Fig. 7. The XYZ Correlator application running with 2 signals (phase signals from 2 BPMs).

6 REFERENCES

- [1] <http://www.sns.gov/APGroup/appProg/xal/xal.htm>
- [2] J. Galambos, et. al., "XAL-Application Programming Framework", <http://icalepcs2003.postech.ac.kr/>
- [3] <http://www.aps.anl.gov/epics>
- [4] C.K. Allen, et. al., "A Novel Online Simulator for High-Level Control Applications Requiring A Model Reference", <http://icalepcs2003.postech.ac.kr/>
- [5] T. Owens, et.al., "Phase Scan Signature Matching for Linac Tuning", Particle Accelerators, 1994, **48**, p. 169
- [6] <http://www.jython.org>
- [7] T. Pelaia, EPICS Collaboration Meeting, http://lansce.lanl.gov/epics/presentations/TomPelaiaXAL_Report.pdf