

# OPEN XAL STATUS REPORT 2019

A. P. Zhukov\*, C. K. Allen, A. P. Shishlo, Oak Ridge National Laboratory, TN, USA  
T. Dodson, University of Tennessee, TN, USA  
J. F. Esteban Müller, E. Laface, Y. Levinsen, N. Milas, C. Rosati, B. Bolling  
European Spallation Source ERIC, Lund, Sweden  
P. Gillette, P. Laurent, G. Normand, A. Savalle  
Grand Accélérateur National d'Ions Lourds, Caen, France  
P. Chu, M. Li, Y. Li, X. Lu, J. Peng  
Institute of High Energy Physics, Chinese Academy of Sciences, Beijing, China

## Abstract

The Open XAL accelerator physics software platform is being developed through an international collaboration among several facilities since 2010. This paper discusses progress in beam dynamics simulation, new RF models, and updated application framework along with new generic accelerator physics applications. We present the current status of the project, a roadmap for continued development and an overview of the project status at each participating facility.

## INTRODUCTION

Open XAL [1, 2] is a generic open source software platform for accelerator physics. Open XAL is written mainly in the Java programming language but is accessible via any JVM based scripting language and Python. It is used worldwide by different accelerator labs including: China Spallation Neutron Source (CSNS) in Dongguan, China, European Spallation Source (ESS) in Lund, Sweden, Spiral2 program at the Grand Accélérateur National d'Ions Lourds (GANIL) in Caen, France, and SNS in Oak Ridge, TN.

## ESS

During the past year, the commissioning of the ESS ion source and low-energy beam transport (LEBT) started [3] and for the first time we tested Open XAL with the real ESS machine. Our efforts focused on improving the model of this part of the accelerator and debugging the applications needed at this stage.

### Model and Machine Description (SMF)

Originally, Open XAL was designed for bunched beams only. We included a new algorithm in the EnvelopeTrackerBase class to simulate space-charge effects for continuous beam, which is needed to model our LEBT [4]. This algorithm can be activated by enabling a flag.

Another issue with modelling the ESS LEBT was that the corrector magnets are too long and a single kick was not accurate. We added the possibility to split a dipole corrector in several slices to overcome this problem. Each slice can have a different weight, which is also useful when the magnetic profile cannot be approximated by a square

\* zhukovap@ornl.gov

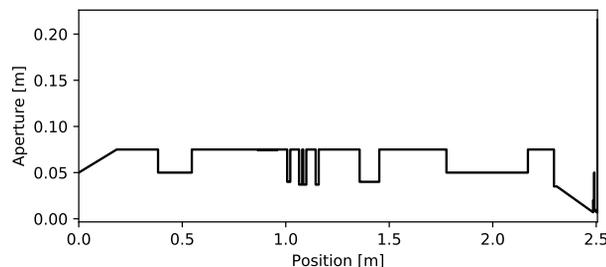


Figure 1: Aperture profile of the ESS LEBT generated by Open XAL.

function. Varying weight allows reduction of the error in the approximation. The sign of the kick for vertical correctors was changed to follow our convention.

The definition of apertures has been extended and now each element can have an array describing its aperture, and an integer to set the aperture shape (e.g. rectangular, elliptical). Markers can incorporate an aperture bucket to describe the aperture of the drift spaces. The AcceleratorSeq class includes methods to get the aperture for all its elements. Figure 1 shows an example of the aperture profile of the ESS LEBT.

We also simplified some of the existing elements. Dipole bending and steering magnets had different classes for vertical and horizontal orientations, and they have been merged. The field map classes have been completely refactored to allow quick development of new field map classes [5]. This new field map model has a generic integrator for the equations of motion of a particle in electromagnetic fields and allows the superposition of field maps.

Open XAL had a simple model for misalignments implemented for some elements. This has also been refactored to take into account all elements, except for dipole magnets. Some modifications were required to follow the standard yaw-pitch-roll convention (Tait-Bryan angles).

Finally, a model for the RFQ is currently under development [5].

### XAOS Application Framework

The XAOS project [6] is a JavaFX application framework made to facilitate building Open XAL applications requiring a graphical user interface.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

In the last year, the project was restructured using the Java Platform Module System (JPMS) available since Java 9, and updated to be compatible with Java 11 and 12. It was enriched with chart plotting capabilities, first implementation of window management and distributed, inter-application communication.

First examples of its usage will be presented at the ICALEPCS'19 conference [7].

### Other General Changes

A bug in the JCA plugin for EPICS communication was found and fixed. The plugin was throwing timeout exceptions when getting a PV after trying to connect to a non-existing PV. In the same context, the pvAccess plugin was updated to use the latest version of the pvAccessJava libraries. There are still minor differences in the behaviour of the JCA and pvAccess plugins that we plan to fix.

A new DataAdaptor was developed to allow applications to support files using the HDF5 binary format, in addition to XML that was currently supported.

Some additional features were added to the Scanning application based on the experience using it during commissioning. The HDF5 support as described above was added, in addition to a separate HDF5 file writing for array datasets. More flexibility in terms of handling the scan procedure for the user with a few flags, and a new command line feature was added. The latter allows the user to define a command to be executed in between each scan step, and the script in turn can control if the scan should continue or not based on its return value.

### Using Open XAL to Model the ESS LEBT

Using the continuous beam space charge implementation we are now able to simulate a simple solenoid scan observing the beam at the profile monitor (NPM, non-invasive profile monitor) placed in the middle of the LEBT. The values for the initial conditions come from emittance measurements combined with a simple minimisation algorithm to try to guess the initial Twiss parameters. A first result is show in Fig. 2.

### Development and Deployment Practices

Open XAL has been ported to Java 12 with minor modifications. Preliminary tests were successful, but further tests are still ongoing before deploying it in the Control Room.

Concerning the continuous integration, we migrated our code to a Gitlab server to have the sources locally stored, and builds are now done using Gitlab-CI. In the near future, We are planning to split the deployment process for Open XAL core and applications, so that each application can follow its own deployment schedule.

## SNS

The SNS team worked on several RF related applications for the SNS linac. Also we researched possibility of interfac-

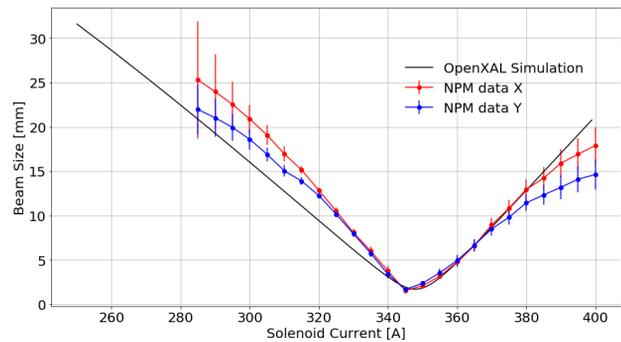


Figure 2: Comparison between simulation in Open XAL using a continuous beam and the rms values for the profiles observed at a profile monitor downstream the first solenoid at the ESS LEBT.

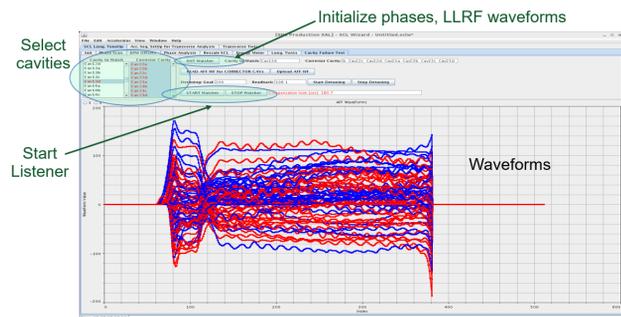


Figure 3: Cavity Failure Compensation in the SCL Tuning Wizard App.

ing Open XAL Java code with PyORBIT [8] code written in Python.

### Cavity Failure Automated Compensation

The SNS Superconducting Linac (SCL) is a good testbed for development of algorithms and methods for a future Accelerator Driven System (ADS). One ADS problem is the very limited number and duration of accelerator trips it can afford. If such beam interruption occurs the beam power should be restored in less than 10 sec. This implies the power restoration process for an ADS should be automated. For routine SNS operation, SCL cavity failure or parameter change do not happen frequently enough to affect availability of the accelerator. Nevertheless, SNS has an Open XAL application (SCL Tuning Wizard) to retune the SCL. The retuning process takes about 15-30 minutes, and it includes 43 steps which are mostly administrative. To demonstrate that the linac retuning process can be automated, and to find critical components, we modified the SCL Tuning Wizard app by including the automated beam restoration option. The screenshot of this part of the app is shown in Fig. 3. The new addition was successfully tested, and it detected the failed cavity, detuned it, rephased the SCL linac, and restored the production power on the SNS target in 120 seconds and with the acceptable beam loss. The restoration time includes 30 second of power ramping up (administrative

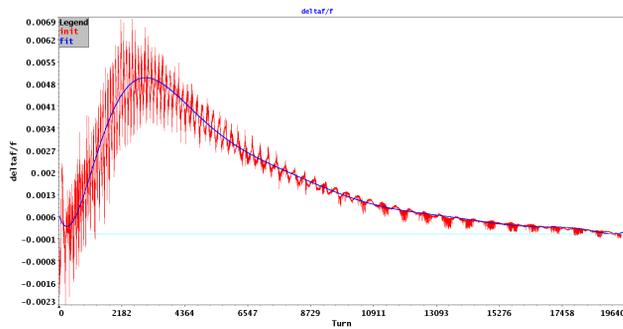


Figure 4: RRF frequency curve error before correction of the RRF frequency Curve.

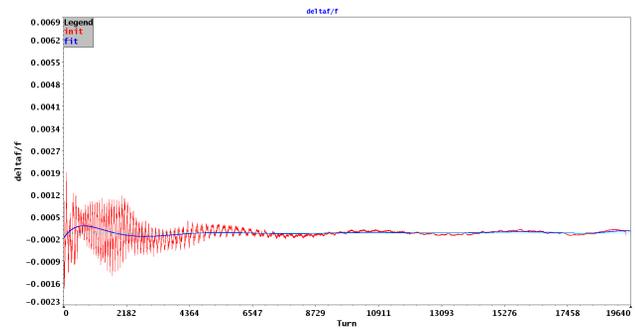


Figure 5: RRF frequency curve error after correction of the RRF frequency Curve.

rule at SNS), and 90 seconds of the failed cavity detuning from the resonant frequency. All necessary calculations and communications with controllers took less than 1 second. This test showed that the automated fast retuning of the superconducting linacs is achievable, but the future ADS projects should implement faster cavity detuning.

### Optimization of RF Learning

RF learning is an essential mechanism providing feedback for beam loading of the cavity. This procedure is automated and takes several macro-pulses to adjust RF settings to modified beam conditions (usually macro pulse length). During this process beam is often interlocked by loss monitors. One can record different beam scenarios and corresponding RF settings that were used. Then, instead of initiating RF learning for current beam condition, it's possible to look up a similar scenario among recorded ones and use it as a starting point for learning. This mechanism was implemented as an Open XAL app. It allows saving I and Q waveforms for all SCL cavities as well as amplitude and phase settings. It also defines a distance function in configuration space that includes amplitude and phase for all cavities, beam current length and other timing parameters. The current machine setup is read online via EPICS and the scenario minimizing the distance function, the closest one, is identified.

### CSNS

In the beginning of this year, CSNS reached another significant milestone by operating stably at proton beam power of more than 50 kW, which is half of the designed beam power of CSNS. In order to study the beam longitudinal dynamics, we developed an application called WCM Data Analysis for analyzing the data of the wall current monitor (WCM). The bunch length, bunching factor, longitudinal tune and longitudinal phase distribution can be obtained by this application. Using the longitudinal phase distribution data, the longitudinal phase space distribution can also be obtained based on the Simultaneous Algebraic Reconstruction Technique Algorithm. In order to optimize the RCS ring RF frequency curve, we developed an application called TBTAnalysis. As the phase advance between BPMs with index of 4 and 9 in each of the four folders in the lattice

of the RCS of CSNS was designed to be more or less 180 degree, the mismatch between the ring RF frequency and the dipole magnet field is corrected based on the TBT data of 4 and 9 BPMs. After modifying the ring RF frequency curve, the maximal mismatch decreases from more than 0.5% to 0.01%. Figure 4 shows the frequency curve error before optimization and Fig. 5 after optimization. This application also can analyze the betatron tunes of RCS based on the TBT data.

### ACKNOWLEDGEMENT

This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC0500OR22725 with the U.S. Department of Energy. This research was supported by the DOE Office of Science, Basic Energy Science, and Scientific User Facilities. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for the United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<http://energy.gov/downloads/doe-public-access-plan>).

### REFERENCES

- [1] OpenXAL website, <https://openxal.github.io>
- [2] A. P. Zhukov *et al.*, "Open XAL Status Report 2017", in *Proc. 8th Int. Particle Accelerator Conf. (IPAC'17)*, Copenhagen, Denmark, May 2017, pp. 4676–4678. doi:10.18429/JACoW-IPAC2017-THPVA093
- [3] R. Miyamoto *et al.*, "ESS Low Energy Beam Transport Tuning During the First Beam Commissioning Stage", presented at the 10th Int. Particle Accelerator Conf. (IPAC'19), Melbourne, Australia, May 2019, paper MOPTS084, this conference.
- [4] B. E. Bolling and N. Milas, "DC Beam Space-Charge Modeling for OpenXAL", presented at the 10th Int. Particle Accelerator Conf. (IPAC'19), Melbourne, Australia, May 2019, paper WEPTS037, this conference.
- [5] J. F. Esteban Muller and E. Laface, "New RFQ and Field Map Model for the ESS Linac Simulator", presented at the 10th Int.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

Particle Accelerator Conf. (IPAC'19), Melbourne, Australia, May 2019, paper WEPTS038, this conference.

[6] C. Rosati, XAOS. <https://github.com/ESSICS/XAOS>

[7] C. Rosati and K. Shroff, "New Java frameworks for building next generation EPICS applications", in *Proc. ICALEPCS'19*,

to be published, 2019.

[8] A. Shishlo, S. Cousineau, J. Holmes, and T. Gorlov, "The particle accelerator simulation code PyORBIT", *Procedia Computer Science*, vol. 51, pp. 1272–1281, 2015.