

STATUS OF OpenXAL AT ESS

N. Milas*, J. F. Esteban Müller, E. Laface, Y. Levinsen
European Spallation Source ERIC, Lund, Sweden

Abstract

The Open XAL accelerator physics software platform is being developed through international collaboration among several facilities since 2010. The goal of the collaboration is to establish Open XAL as a multi-purpose software platform supporting a broad range of tool and application development in accelerator physics and high-level control. This paper discusses progress in beam dynamics simulation and updated application framework along with new generic accelerator physics applications for the ESS branch of the collaboration. We present the current status of the project, a roadmap for continued development and an overview of the future developments needed for ESS future commissioning work.

INTRODUCTION

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

The decision to use Open XAL at ESS was taken in 2015 [3] taking into account the similarities between SNS and ESS and the possibility of re-using part of the work already developed. Between 2015 and 2019 a fair amount of changes to the core part of the code were done in order to fit Open XAL to site specific needs at ESS. The main efforts were directed towards improving the site specific on-line model (JELS) and in the development of applications required for the first stages of the accelerator commissioning. Further changes were done in the applications framework, that was shifted from Swing to JavaFX, on the data saving format with the addition of HDF5 as the new standard among many others.

In this work a summary of the specific development of Open XAL functionalities will be described as well as the first set of application used for the Ion Source and LEBT commissioning between 2018 and 2019. An overview of the future developments will also be outlined.

* natalia.milas@esss.se

CORE AND EXTENSIONS

Development and Deployment

Open XAL has been upgraded to Java 12 with minor modifications and it will be available at the Control Room for the next commissioning phase starting on the second quarter of 2020. At ESS we use a Gitlab server to have the sources locally stored, and builds are now done using Gitlab-CI. We split the deployment process for Open XAL core and applications, so that each application can follow its own deployment schedule. During the commissioning of the LEBT and the Ion Source in 2019 we noticed that the deployment process of Open XAL as a single unit made the process of debugging and releasing new features for the Applications cumbersome. We now have several Git repositories, one for the core and services, one for the lattice, and for the applications.

Model and Machine Description

The machine description is now imported from the official lattice repository, which uses the TraceWin [4] format, and the SMF xml files are generated accordingly. 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.

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. This new field map model has a 4th Runge-Kutta 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 been refactored to take into account all elements, except for dipole magnets, and is valid only for small angles, when the approximation $\tan \theta \approx \theta$ holds. We use the standard yaw-pitch-roll convention (Tait-Bryan angles). For the misalignment, sequence type elements like the RF create a global misalignment and single elements inside the sequence can have their own parameters, the final quantities used are the sum of all misalignments for each single element.

Originally, Open XAL was designed for bunched beams only. We included a new algorithm in the EnvelopeTracker-Base class to simulate space-charge effects for continuous beam [5], which is needed to model the ESS LEBT. This algorithm can be activated by enabling a flag either while

running, `setUseDCBeam(True)`, or in the xml at the model parameters input file, `useDCBeam="true"`. The space charge routine considers only linear space charge effects and assumes a uniform distribution. During the last commissioning period we tested the new space charge model and compared with measurements as show in Fig. 1 which shows a very good agreement between data and model.

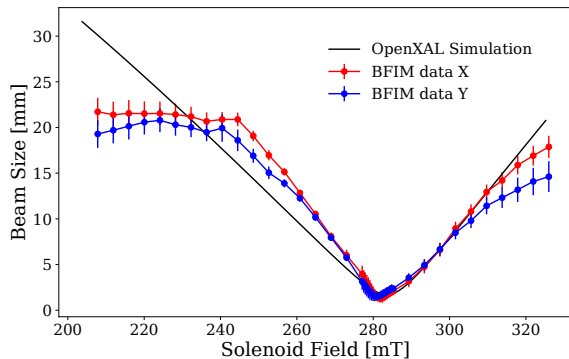


Figure 1: Comparison between Open XAL model using the DC beam space charge model and data from the ESS LEBT.

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 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.

Finally, a model for the RFQ is currently under development [6], which is the only section of the accelerator missing an online model. As a proof of concept, we developed a model based on the classical two-term potential function. We started from the simplest model because we are concerned about the computation speed, since JELS [7] is mainly used in Open XAL as an online model. Figure 2 shows the results of our model compared to a simulation run using TraceWin [4]. Unfortunately, it is clear that the accuracy of the model using the two-term potential is not sufficient and a more precise model is under study.

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, a `pvAccess` plugin has been developed using the `pvAccessJava` libraries to add EPICS7 support to Open XAL. 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.

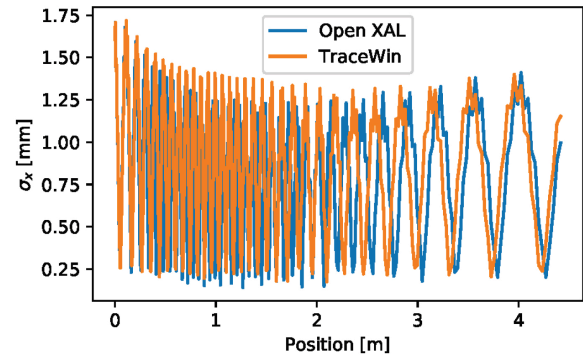


Figure 2: Horizontal beam size in the ESS RFQ simulated using Open XAL (blue) and TraceWin (orange). Initial beam kinetic energy 74.6 keV and no space charge.

At ESS, all functionalities of Open XAL core are also available as a library in the ESS Jupyter server and can be readily accessible via python.

JavaFX FRAMEWORK

A new framework for physics applications is being developed based on JavaFX called `FxApplication` and is being used for all new applications. It adds a tool bar with the same functionality as the Swing version and also a link to the User documentation Wiki. The ESS logbook was integrated in Open XAL so that applications can post new entries from a dialog or generate automatic messages.

For JavaFX applications a new plotting library was created, which is now part of the XAOS framework [8]. XAOS is written using the Java Platform Module System (JPMS) and runs in Java 12. This library includes many plotting formats ranging from line and histograms to density plots. A set of plugins can be added to any plot, and those include: error bars, plot statistics, cursor, zoom, axis properties among others. A example plot screen is shown in Fig. 3.

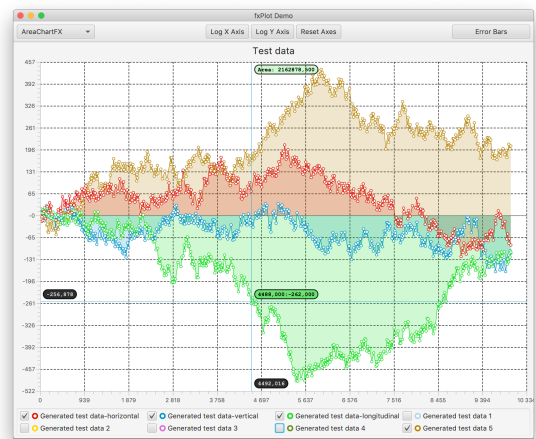


Figure 3: Example area plot from the JavaFX XAOS library.

APPLICATIONS AND SCRIPTING

The Open XAL configurator application is a simple tool to configure some of the Open XAL settings, as the default optics, logbook server address, and the status of the authentication system RBAC. The GUI for the configurator is shown in Fig. 4.

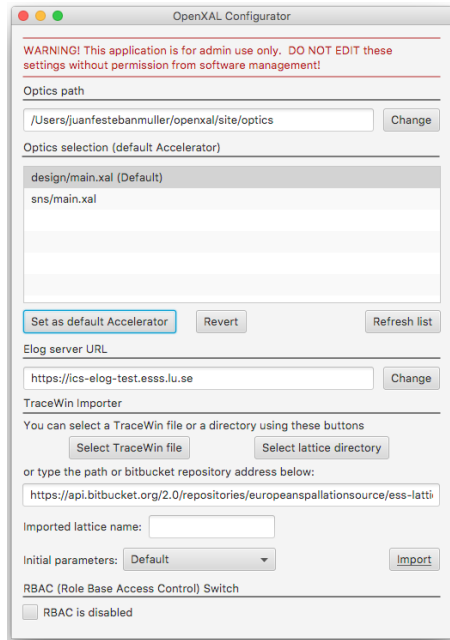


Figure 4: Configurator application GUI.

For commissioning a LEBT viewer application was created. In this application the trajectory and envelope along the LEBT and the commissioning tank are displayed together with readings from diagnostics. From the application it is possible to manipulate some of the magnets and power supplies from the source and LEBT and it is also possible to perform a simple matching based on diagnostics readings (from profile and emittance monitors). A more detailed description of this application can be found in [9] and an example showing the LEBT GUI is in Fig. 5.

The Scanner application allows a multidimensional scan of any number of variables. A new widget has been developed to quickly select multiple channels to read from or write to. As for other applications, the data is stored in HDF5 or XML files afterwards. It is possible to filter the scan ranges by analytical formulas that accept most basic mathematical operations and comparators. Configuration such as time delay between data acquisitions and multiple acquisitions for each setting is available. There is minimal analysis of the data in the application itself, expecting instead that the user develop scripts to fulfil their specific needs. Some additional features were added to the Scanner application based on the experience using it during commissioning which increase the 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 generic command to be executed in between each scan step (for example



Figure 5: LEBT application GUI.

an executable script), and the command in turn can control if the scan should continue or not based on its return value.

For the next commissioning step a trajectory correction application is under development. Two correction methods are included so far: a 1-to-1 and a SVD correction. The matrices used can be extracted from the machine model or can be measured. It is possible to define machine blocks where a subset of the machine BPMs and steerers are selected and are independent of the XML file sequences definitions. The overall trajectory can be corrected to the BPM zeros or to a predefined reference trajectory. To complement this application a Trajectory Display Application was also developed (see Fig. 6).

For the phase scan and transverse matching applications we plan to start using scripting language, instead of going straight to GUIs. They will be implemented as Jupyter notebook scripts, tested in the control room and transported to its final form as GUIs after debugging is completed. This was needed in order to optimize the time developers spend between application writing and commissioning.

NEXT STEPS

In conjunction with SNS we plan to define a universal diagnostics data format in HDF5 allowing easier exchange of information between the laboratories. Further development of the HDF5 Data Adaptor is also foreseen in order to allow document writing and saving simultaneously as well as fast writing of long array like channels.

We are considering implementing a multi-particle tracker together with a PIC code or another space-charge solver into a new Tracker. That can considerably slow down the simulations in the control room but those can be run independently from the main application for further analysis of experiment results and can be a valuable ally.

Another addition to Open XAL, this time aiming at future upgrade with the neutrino ESS Super Beam project [10] would be the addition of H^- stripping probability calculation

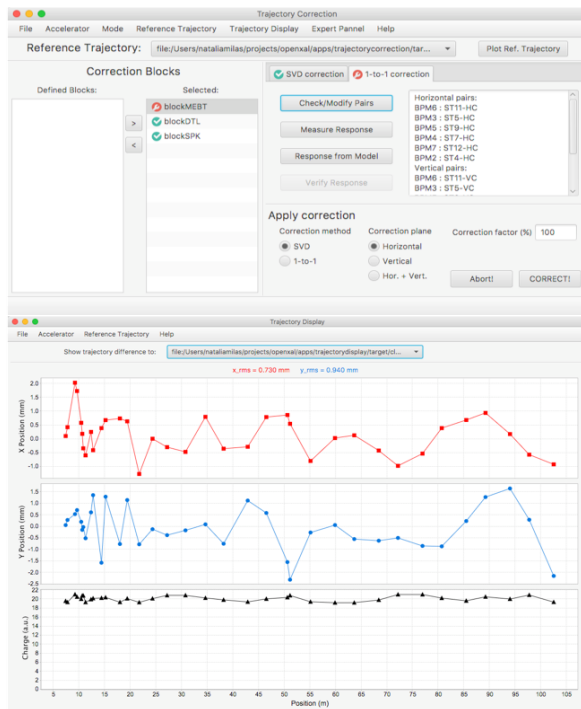


Figure 6: Trajectory Correction and Display application GUI.

along the machine. This is still a work in progress and is not the highest priority for ESS at the moment.

Finally, after this many changes the ESS branch and the main branch of Open XAL have diverged considerably. We have been in touch and we are planning next year to try to build and run the ESS version of Open XAL at SNS to evaluate if all changes can be merged to the main branch.

REFERENCES

[1] Open XAL, 2003, <http://openxal.github.io/>
 [2] J. Galambos *et al.*, “XAL Application Programming Structure”, in *Proc. 21st Particle Accelerator Conf. (PAC’05)*, Knoxville, TN, USA, May 2005, paper ROPA001, pp. 79–83.

[3] M. Munoz and Y. I. Levinsen, “Evaluation of OpenXAL”, ESS, Lund, Sweden, Rep. ESS-0017729, May 2015, <https://chess.esss.lu.se/enovia/link/ESS-0017729/21308.51166.60928.13621/valid>
 [4] D. Uriot, “TraceWin Simulation Software”, <http://irfu.cea.fr/Sacm/en/logiciels/index3.php>
 [5] B. Bolling and N. Milas, “DC beam space-charge modeling for OpenXAL”, in *Proc. 10th Int. Particle Accelerator Conf. (IPAC’19)*, Melbourne, Australia, May 2019, pp. 3177–3180. doi:10.18429/JACoW-IPAC2019-WEPTS037
 [6] J. F. Esteban Müller and E. Laface, “New RFQ and field map model for the ESS linac simulator”, in *Proc. 10th Int. Particle Accelerator Conf. (IPAC’19)*, Melbourne, Australia, May 2019, pp. 3181–3183. doi:10.18429/JACoW-IPAC2019-WEPTS038
 [7] E. Laface, M. Eshraqi, and R. Miyamoto, “Space charge and cavity modeling for the ESS Linac simulator”, in *Proc. 4th Int. Particle Accelerator Conf. (IPAC’13)*, Shanghai, China, May 2013, paper WE-PEA040, pp. 2588–2590.
 [8] XAOS, <https://github.com/ESSICS/XAOS>
 [9] N. Milas, K. S. Louisy, and D. C. Plostinar, “Transverse dynamics and software integration of the ESS low energy beam transport”, in *Proc. 9th Int. Particle Accelerator Conf. (IPAC’18)*, Vancouver, BC, Canada, Apr.-May 2018, pp. 882–885. doi:10.18429/JACoW-IPAC2018-TUPAF066
 [10] B. Gålnander, M. Collins, M. Eshraqi, A. Farricker, C. A. Martins, and R. Miyamoto, “Status of ESS linac upgrades Studies for ESSnuSB”, in *Proc. 10th Int. Particle Accelerator Conf. (IPAC’19)*, Melbourne, Australia, May 2019, pp. 1038–1041. doi:10.18429/JACoW-IPAC2019-MOPTS082

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.