

OCELOT INTEGRATION INTO KARA'S CONTROL SYSTEM

P. Schreiber*, E. Blomley, J. Gethmann, W. Mexner, M. Schuh, A.-S. Müller
Karlsruhe Institute of Technology, Karlsruhe, Germany

Abstract

The Karlsruhe Research Accelerator (KARA) at the Karlsruhe Institute of Technology (KIT) is an electron storage ring and synchrotron radiation facility. The operation at KARA can be very flexible in terms of beam energy, optics, intensity, filling structure, and operation duration. For different aspects of the operation of the accelerator separate and individual simulation models are in place using different simulation tools, custom lattice data and varying levels of maintenance. In a general push at the accelerator to provide unified access via Python, a new framework was implemented using Ocelot with a much closer integration to the accelerator control system and supplementary tools. This allows a better integration and lowers the effort necessary for simulations and predictions of actual changes to the beam properties based on live data. It also provides a good entry point for the various Python based machine learning activities at the accelerator and the goal to obtain an easier to maintain and test accelerator model. This paper presents the taken approach and current status of this project.

INTRODUCTION

The storage ring KARA (Karlsruhe Research Accelerator) is operated as an accelerator test facility and serves as the KIT light source. It is a ramping electron synchrotron from 0.5 GeV to 2.5 GeV with a 4 fold symmetry and a total of 8 double bend achromatic cells. The circumference of 110.4 m and frequency of the RF system of 500 MHz results in a bunch revolution time of 368 ns and a bunch spacing of 2 ns. Furthermore, a wide range of beam diagnostics tools is available. A lot of beam time is dedicated to machine physics including experiments performed by scientists from external institutions. KARA is available as R&D facility for TNA (TransNational Access) via Eurolabs [1]. In order to make best use of the beam time simulations beforehand are necessary. Therefore easy access and ideally pre-set-up simulations are required.

REQUIREMENTS

Performing simulations for an accelerator, especially for new students or external scientists working at KARA can be hard to set up as many details such as exact lattice parameters as well as machine specific conversion coefficients are mandatory. Therefore the integrated simulation should be easy to use and have reasonable defaults. It should furthermore allow a reasonable amount of customisations to adapt the simulations to the various needs.

Furthermore, future systems might rely on e.g. the optics functions for the currently used settings at KARA. Such

systems could consist of machine learning applications for various tasks such as operation optimisation. For these systems the simulation results should be readily available via network connections. The simulations for such systems should also be run periodically to ensure recent values.

A third requirement for the integration is easy maintain and extendability. Additional simulation features, such as tracking or new elements should be easily implemented in order to encourage extension of the system.

SIMULATION SETUP

The simulation setup consists of a lattice defined with LatticeJSON [2], an epics-IOC [3] with records for each element or family, ocelot [4] as simulation software and a control panel in CSS [5]. An overview of the involved objects and their connections is shown in Fig. 1.

From the LatticeJSON definition an intermediate object is created that acts as uniform interface between the lattice definition, the IOC and the actual magnetic lattice for ocelot. This approach was chosen in order to allow potential other simulation tools to make use of the same lattice input. For each element in the lattice, the intermediate object creates an element type specific object. These element objects are responsible for transforming the user facing values of current flowing through the magnets to magnet strengths and also for creating the elements used by ocelot.

The intermediate lattice object is used by the automatic creation of an epics-IOC. A record in the IOC is created for each specified PV of each element in the LatticeJSON definition (this allows for multiple PVs per element if necessary).

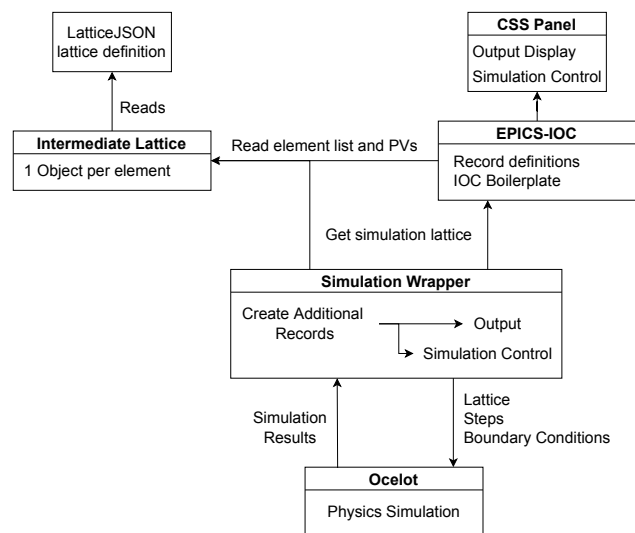


Figure 1: Structure and connections between different objects involved in the simulation integration system.

* patrick.schreiber@kit.edu

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

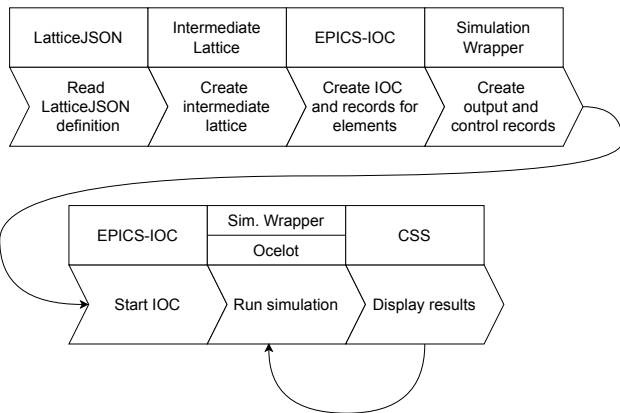


Figure 2: Control flow of the startup and simulation process of the simulation integration system.

These allow control over properties of the elements such as the magnet strength. However, the original LatticeJSON schema does not allow to specify PV names and lacks support for certain lattice elements such as BPMs or corrector magnets. As both the additional elements and the attributes for PVs are required, the LatticeJSON schema was extended to include them.

A wrapper around the simulation is used to manage all the previously mentioned elements. It creates additional records in the IOC to allow user control over the simulation such as the granularity of evaluation of the optics functions or the interval between automatic simulations. This wrapper also takes care of managing the simulation and updating the magnetic lattice for ocelot. The start-up order and flow is visualised in Fig. 2.

To conform to the general theme of using python for control system integration all elements are written in python, where for the IOC the pythonSoftIOC [6] package is used. This furthermore focuses on maintainability. As python is easy to read and is widely known it is relatively simple to add additional simulation features or fix bugs in the years to come.

To fulfill the requirement to have updated simulation values (e.g. optics functions or tunes) available for settings used at the accelerator at the moment (e.g. currents flowing through the magnets), periodic online simulations with updated values are run. These periodic runs are managed by the simulation wrapper with a customisable periodicity. For performance optimisations these simulations are performed with a fixed frequency but only when current input values are changed.

INTERFACE

The interface to the simulation is entirely handled via epics. PVs for elements with modifiable attributes allows control over the lattice. This includes the strength of quadrupoles, sextupoles and bending magnets. Additional PVs are provided to store the results or control the simulations. This includes calculated tunes and optics functions.

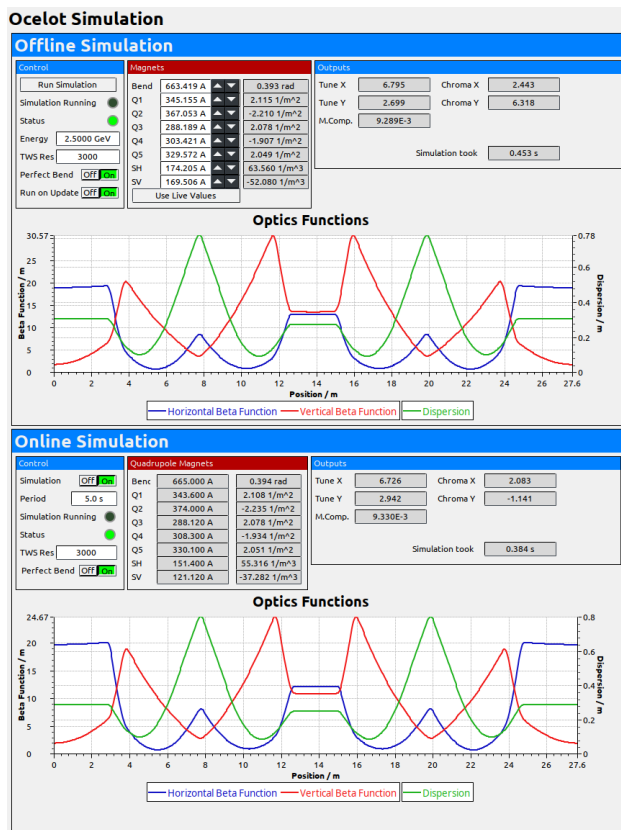


Figure 3: Screenshot of the user interface to the ocelot simulation. The shown panel is provided by CSS, the standard control system UI at KARA. Controls of the simulation as well as control over the magnets for the simulation is present, combined with the output values and optics functions.

Presently the control over the lattice is entirely performed via engineering units. This means the values for magnets are set as current in Ampere. This is also used for the simulations at the moment. It is possible to control the simulation via physical units as well, but the interface (extra PVs) is not available in the UI at the moment. The rationale behind this is to use the same interface for the machine as for the simulations to simplify the switch from simulations to the actual machine or vice versa. The magnet strengths in physical units is displayed in the UI, however, to help users accustomed to them.

To allow access to the simulations a graphical user interface was implemented based on CSS BOY. A screenshot of the interface is shown in Fig. 3.

The UI is split in two panels, the top panel shows the controls and results for offline simulation, where the user can manually select values for e.g. the magnet currents. The lower panel shows the controls and results for the periodic online simulations with limited control over the simulation as most values are taken from the currently used settings at KARA.

Some simulation results are displayed as numerical values, these include the tunes, chromaticities and the momentum compaction factor. The optics functions are displayed as a

combined plot for the beta function in both planes and the dispersion in the horizontal plane.

Apart from the obvious values for the magnets there are additional settings available such as the number of points used for evaluation of the optics functions as well as the beam energy. The latter is necessary to correctly convert from current flowing through the magnets to the magnet strength k . The UI offers to use the current of the dipoles or use a "perfect" magnet with a deflection angle equal to 360° divided by the number of bending magnets present in the lattice (22.5° at KARA). Furthermore it is possible to automatically run the simulation once the value for a magnet is changed to keep the simulation synchronised to the displayed values.

APPLICATIONS

Recent studies and efforts at KIT are aimed towards the usage of machine learning for autonomous accelerators. As the online simulation results are readily available via epics also within python, this system offers easy integration of the simulation results into such algorithms. In turn this allows the integration of machine learning applications into the control system. The centralised simulation would then reduce the computing power necessary as each application can rely on the same results and does not have to simulate individually. Furthermore, the centralised simulations guarantee that all systems relying on simulated values use the same simulation and will not differ due to different simulation lattices etc.

As the simulation can provide parameters that require time expensive measurements, such as beta functions and chromaticities, the online simulation can be used to optimise the regular operation conditions. For example, this can be used to optimise the sextupole magnets without having to redo the chromaticity measurements each time.

Since KARA is an accelerator test facility, often new operation modes and optics are implemented. By using the integrated simulation it is straightforward to test new optics before the actual beam time and reduces the time spent on optimisations. KARA offers its services to scientists of external institutes and as part of KIT as university also to students. For such users the tight integration and coherent interface to the rest of the control system offers a fast learning experience and removes the necessity of obtaining a simulation code, lattice and the knowledge how to operate it.

The code is setup in a way that multiple instances (with different prefixes) can be spawned to circumvent the problem of having only one simulation at a time for potentially multiple users. As an addition in the future, the individual instances could be lockable by users to avoid interference.

RESULTS & OUTLOOK

The system is successfully implemented and first comparison measurements have been carried out. While the simulations can be very well used for studies and beamtime preparations, some simulated parameters do not perfectly

Table 1: Measured and Simulated Tunes

	Measured Tune	Simulated Tune
Horizontal	6.77	6.73
Vertical	2.81	2.94

match the values measured at the machine. As an example, a comparison for the tunes is shown in Table 1.

The simulation lattice as well as the conversions between currents through the magnets and the magnet strengths should be improved. With the integration of the simulation into the control system and therefore direct availability algorithms that can help with those improvements can now be applied, possibly leveraging machine learning. Furthermore, the simulation uses perfect magnets without misalignments or fringe fields. The latter have been studied before in [7]. Also, the insertion devices are not implemented. Therefore, improvements to the simulation could be made by including the above mentioned effects and elements.

Improvements to the entire system could be made in the form of extensions of simulated features. That could include the simulation of emitted radiation or the inclusion of insertion devices in the lattice. Furthermore, an additional setting allowing for tracking simulation could be introduced to for e.g. estimations of beam lifetime or to identify critical apertures in the ring through loss simulations.

To overcome the centralisation issue with multiple users, the simulation could be put into a python package separate from the IOC control logic. This could allow for usage of the pre-setup simulation from within python scripts that offer almost the same convenience as the operation via the CSS panels.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no. 101057511.

REFERENCES

- [1] Eurolabs, <https://web.infn.it/EURO-LABS/>
- [2] LatticeJSON, <https://nobeam.github.io/latticejson/>
- [3] EPICS, <https://epics-controls.org/>
- [4] I. Agapov *et al.*, "OCELOT: A software framework for synchrotron light source and FEL studies", *Nucl. Instrum. Meth. A*, vol. 768, pp. 151-156, 2014. doi:10.1016/j.nima.2014.09.057
- [5] Control System Studio, <https://controlsystemstudio.org/>
- [6] pythonSoftIOC, <https://github.com/dls-controls/pythonSoftIOC>
- [7] M. Streichert *et al.*, "Simulations of fringe fields and multipoles for the ANKA storage ring bending magnets", in *Proc. IPAC'12*, New Orleans, LA, USA, May 2012, paper TUPPP011, pp. 1626-1628. <https://jacow.org/ipac2012/papers/tuppp011.pdf>