ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-MOPV012

THE ESRF-EBS SIMULATOR: A COMMISSIONING BOOSTER

S. Liuzzo*, L.R. Carver, J.M. Chaize, L .Farvacque, A. Götz, D. Lacoste,
N .Leclercq, F. Poncet, E.Taurel†, S. White
ESRF, Grenoble, France

Abstract

The ESRF-Extremely Brilliant Source (ESRF-EBS) [1] is the first-of-a-kind fourth-generation high-energy synchrotron. After only a 20-month shutdown, scientific users were back to carry out experiments with the new source. The EBS Simulator (EBSS) played a major role in the success of the commissioning of the new storage ring. Acting as a development, sandbox and training platform, the machine simulator allowed control room applications and tools to be up and ready from day one. The EBSS can also be seen as the initial block of a storage ring digital twin. The present article provides an overview of the current status of the EBS Simulator and presents the current roadmap foreseen for its future.

INTRODUCTION

The ESRF storage ring was upgraded in 2019 to provide 100 times brigther X-rays [1]. The strong demand for experiments at ESRF imposed a very tight schedule for dismantling, installation and commissioning of the new storage ring. Overall, 20 months of dark time for external users were necessary [2]. Only three of these months were dedicated to Storage Ring commissioning. In order to cope with this schedule, the design and update of the whole software infrastructure had to be brought forward as much as possible. This is particularly true for the high-level applications needed on the first day of commissioning, such as the new magnets control. In addition, several applications specific to the commissioning such as beam threading algorithms [3] had to be prepared and tested to be used effectively, with minimal debug time.

For this purpose a full test of the software from high-level applications to power-supplies level (current input, not hardware level) was realized via an EBS control system simulator (EBSS). This control system simulator was strongly focused on the new EBS magnets control system that needed to be completely redesigned, but included as well all the frequently used diagnostic devices (beam position monitor (BPMs), tunes, emittances, etc..) needed for the development of the tools used for the commissioning. The output values of the simulated diagnostic devices are generated from a given lattice optics model that is updated upon a magnetic strength or RF frequency variations in the simulated control system - as depicted in Fig. 1.

From the user's point of view, the simulator is identical to the "physical" control system. The devices providing the information on the beam position have identical names for

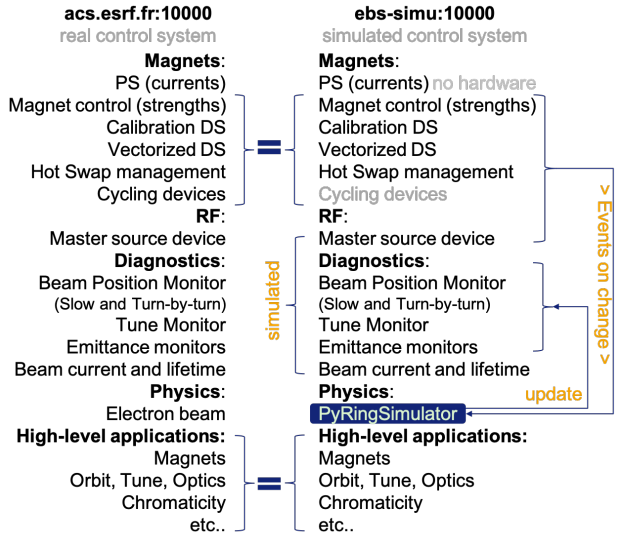

Figure 1: Real control system vs EBS simulator.

attributes and properties as their physical counterparts. This allows to port the applications from the simulator environment to the real machine by simply changing the environment variable pointing to the TANGO [4] database.

EBSS is a clone of a subset of the EBS control system that allows to interact with a simulated beam and to visualize the expected behaviour of most of the relevant electron beam and lattice observables: orbit, tunes, emittances, optics and coupling.

In the past, the ability to run off-line control room applications was already available, for example via toolkits such as the Matlab-Middle-Layer [5], used in many 3rd generation light sources or specific solutions - like the one on which the Virtual XFEL [6] relies.

For storage-rings, Matlab-Middle-Layer provides a high level switch from simulations to real beam experiments. This limits the development to anything above the matlab-middle-layer software infrastructure. The EBSS instead acts at a lower level, replacing directly the beam, thus giving access to all control levels. It notably enables real-time tests of features (1-2s/loop) not only for beam dynamics experts but also for control system engineers.

STRUCTURE OF THE SIMULATOR

The core of an EBSS instance [7] is composed of more than ~ 4000 Tango devices compared to ~ 25000 in the physical accelerator complex (including the 3 accelerators - linac, booster + storage ring). Each EBSS instance runs on its own Tango control system - the associated Tango database

* simone.liuzzo@esrf.fr
† taurel@esrf.fr

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-MOPV012

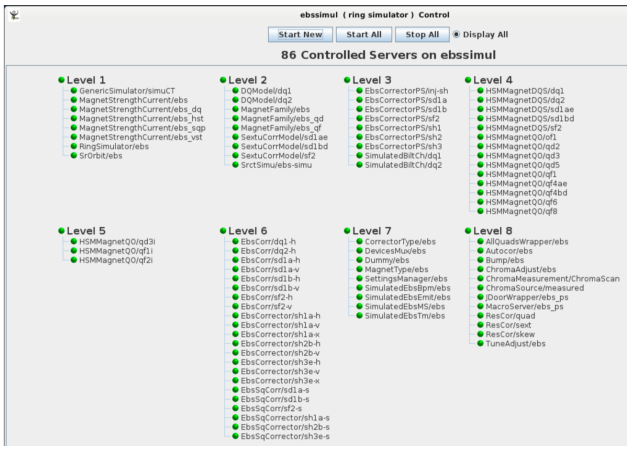

Figure 2: View of the 86 Tango device servers running within a docker.

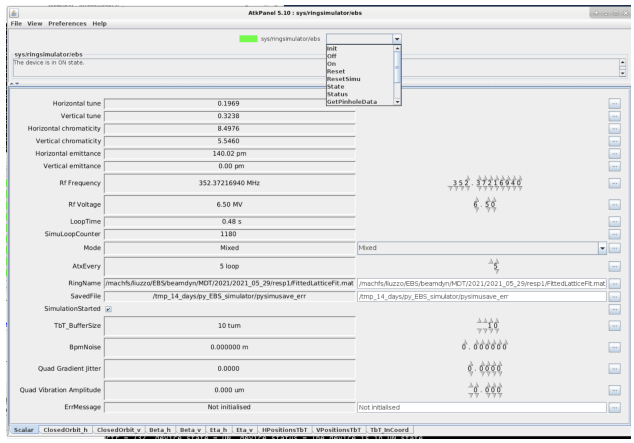

Figure 3: PyRingSimulator ATK panel view.

being isolated in a Docker container. This allows multiple EBSS to run simultaneously and independently on the same host. The device servers architecture is depicted in Fig. 2. Presently three EBSS instances are deployed at the ESRF - each one consuming about 25-30% of the CPU resources and 18-20% of the memory of a 16-CPUs/64-Gb host.

Tango Device Servers

Each EBSS Tango device exposes the exact same interface as its production counterpart. This ensures a total transparency to its clients and guarantees a smooth transition from simulation to production. At the lowest level of the control system, most of the devices are written in C++ and support a compilation flag providing a way to specify the target platform - i.e., simulation or production. For instance, the BPMs read the beam position from the dedicated hardware when running in production mode, whereas they obtain it from the ring simulator while running in the context of an EBSS instance. The behaviour of some devices has also been modified in order to broadcast or specifically notify (upon change) the EBSS environment - e.g., a strength modification on a magnet will trigger a modification of the EBS storage ring magnetic model and a consequent update of the relevant beam parameters. These asynchronous notifications rely only on the Tango events mechanism. A total of ~ 100 Device Classes and ~ 500 Device Servers are configured to run the EBSS.

Physics Core: PyRingSimulator The PyRingSimulator device server (see Fig. 3, and Fig. 2 level 1) runs high energy electron beam dynamics simulations [8]. It was initially developed in Matlab [9] and C++ [10], and has recently been ported to Python 3 [11].

The PyRingSimulator DS runs a dedicated process at startup (`ebs_simu.py`) that continuously[1] computes the optics using python Accelerator Toolbox (pyAT) [12]. For each EBSS instance, one CPU core is then fully dedicated to this process. This structure allows to better exploit the CPU resources and provides a clear separation among the issues and workload related to control and those related purely to beam dynamics. Moreover, this separation allows for future extensions of the beam dynamic simulations to exploit more CPU resources, GPUs or the ESRF computing cluster, keeping the pyTANGO DS unchanged.

The simulated diagnostic DS continuously reads/monitors the updated parameters in PyRingSimulator instead of measuring real beam properties. From the user's point of view, the devices providing orbit, tune, chromaticity, etc, behave as the physical ones available in the control room. Figure 1 depicts the structure of the Tango devices layers: the PyRingSimulator device is the only one not existing in the real control system: it actually replaces the electron beam.

The electron beam magnetic lattice (optics model) used by PyRingSimulator can be the same as the one used for operation, a measured one or a different one (yet, an EBS optics model). It is defined by three individual lattice files: a reference optics with no errors nor corrections, a lattice including errors (on any magnet or BPM, potentially unstable) and a lattice including the same errors as in the previous one and the corresponding corrections. This structure allows to hide the lattice errors (any error that is possible to specify with pyAT) to the user, just as in real life. On the other hand, these three models can be identical if needed to give full freedom in the conditions of the machine to mimic. For example, to simulate the first turns steering, the lattice with errors and corrections has all the corrections set to zero, and it is thus identical to the one with errors only. An unstable lattice is simulated and as the correctors strengths are set to progress in the first turns steering, the simulations are updated to display the trajectory signal at more and more BPMs, finally leading to a stable machine (see [3]).

In order to make the EBSS even closer to the physical storage ring a few stochastic effects are also included: BPM noise, magnets vibrations (quadrupole), and field gradient jitter.

The optics computations can be selected (for speed reasons) among the following options:

[1] the loop runs continuously and not only upon changes to allow the introduction of stochastic errors, see later.

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-MOPV012

1. *Emittance* mode (>~ 1.6 s/loop), computing all linear optics, including emittances but excluding Turn-by-turn data;
2. *LinearOptics* mode, as the *Emittance* mode but without the lengthy natural emittance computation (> 0.8s/loop),
3. *Mixed*, iterates one loop *Emittance* and few loops *LinearOptics*,
4. *Turn-by-turn* mode (0.02 s/loop/turn (not linear, 0.4 s for 1 turn, 2 s for 100 turns), only computes the T-b-T buffer for the required number of turns and with the specified injection coordinates.

The actual loop speed of the EBSS depends on the number of magnets (or RF) that are modified before the required computations. The times are approximately doubled when using radiation for all computations (rarely used case). A computation loop counter is used at the Tango device level as a trigger signal for new data.

The PyRingSimulator has the ability to set the strengths of the magnets from the optics model (with errors and corrections) to the magnets of the EBSS. This feature was introduced to be able to reset the initial conditions of the simulations and proved to be extremely useful.

Moreover it is possible to save a lattice to pyAT format at any moment for further studies or to debug the EBSS itself.

Simulated Lifetime/Current Decay The lifetime/current device is a standalone device implementing an exponential decay starting from a given current, with a given lifetime. Simulations of beam (Touschek) lifetime are impractically long (hours on a single core, minutes on a computing cluster), and are not crucial for the purpose of the EBSS.

DS AND APPLICATIONS DEVELOPED FOR EBSS

Before the commissioning of the EBS storage ring, the EBSS proved to be extremely useful. Several high level applications needed complete refurbishment, including some major ones such as the magnet control applications.

Magnets Control

Having to develop a completely new control of the magnets - starting from low level device servers to a user-friendly and intuitive graphical interface - was the occasion to also include new features. This is a risky action when it is not possible to test the new features. For EBS, the EBSS was the key to progress in this development, allowing to debug offline in a pure virtual/simulated environment. The new features included in the magnet control were:

- individual control of (more than 1000) power-supplies,
- control of magnets organized by arrays or family,
- magnet set points expressed in “strength” (calibration curves integrated in the control system, including combined function sextupoles with 5 current channels to pilot 4 strengths),

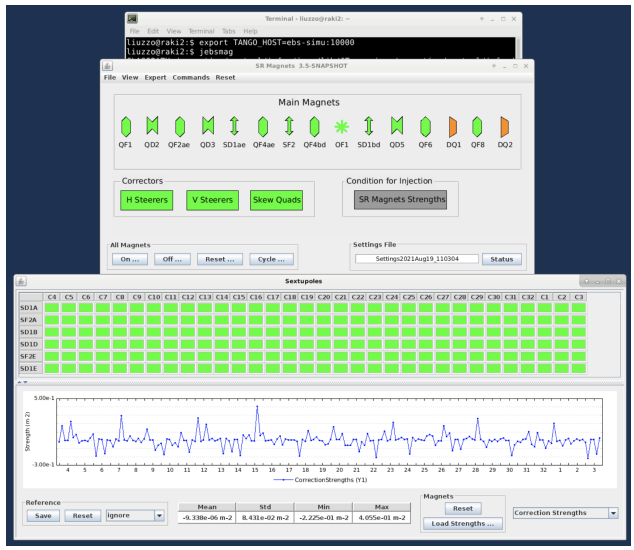

Figure 4: Application for the control of the individual magnets in strengths. Top, the main panel with family tuning, bottom, a view of the sextupole magnets.

- definition of new cycling and ON sequences that avoid the peak power consumption to run out of limits,
- ability to *freeze* magnets (inhibit any user modification of a specific magnet),
- integrate the hot-swap features [13]
- tuning of electron beam resonance knobs,
- save and load of currents (primary) and strengths (utility) of the storage ring magnets configuration files.
- a completely new user interface.

All these features were introduced and tested within the EBSS, leaving for the commissioning only a few issues and some unforeseen developments. A-posteriori, even more applications of the EBSS would have been beneficial before the commissioning - for example, to identify calibration issues.

The final look and feel of the EBS magnets control application is shown in Fig. 4. From the user’s point of view, it is impossible to distinguish an instance of the application running in the context of the simulator from an other attached to the physical machine. All actions are fully functional.

First Turns Steering

An other key feature to a successful commissioning was the ability to automate the first turns trajectory steering. For this purpose basic scripts were designed and tested in the EBSS again solving bugs, giving confidence in the use of the tools and leaving only a few developments to be tested with real beam. The successful use of the EBSS was here also enhanced by the test of applications designed for EBS but tested in the old ESRF storage ring. More details are available in Ref. [3].

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-MOPV012

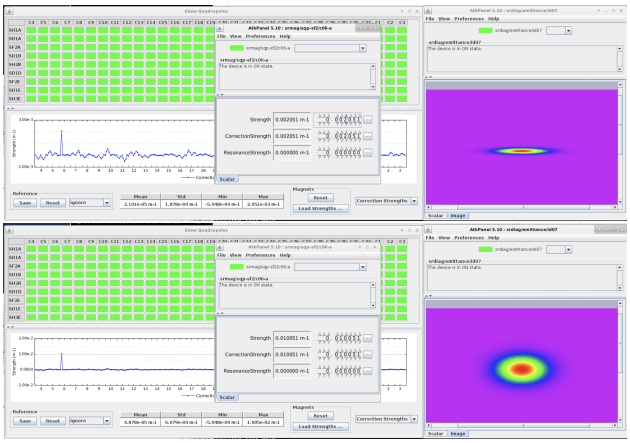

Figure 5: One skew quadrupole is powered (top, low field, bottom, large field) in the EBSS to show the effect on one of the simulated emittance diagnostics.

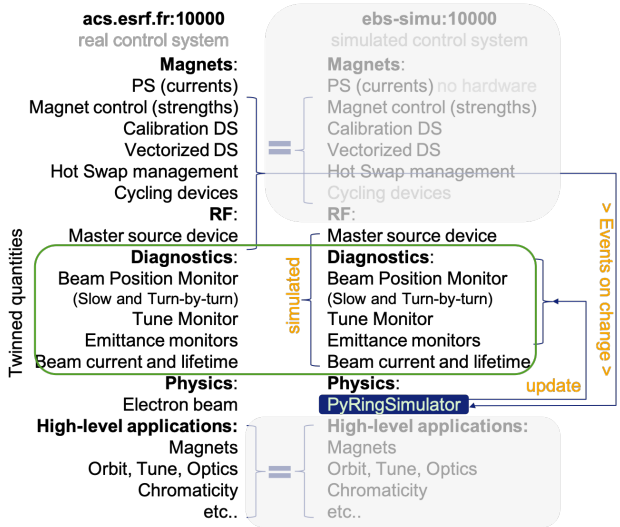

Figure 6: Digital twin configuration using the existing EBS Simulator.

Tune Correction

A new tune correction application was also deployed. The multiplication of the available knobs allowed to generalize the tune correction to any vector of quadrupole magnets, rather than simply two families. This new feature was tested in the EBSS and in the old ESRF storage ring and did not require any further tuning with real beam.

Optics and Coupling Correction

Some of the applications used during the operation of the previous ESRF storage ring could not be updated before commissioning because of the lack of time. Some were deliberately postponed with no detrimental impact on the commissioning schedule. An example is the optics and coupling correction application. This was completely redesigned and updated during the beamline commissioning time, making extensive use of the EBSS simulator to test all steps, from the response matrix measurement to the fit of the lattice errors and the application of quadrupole correction. An example of beam emittance display with a skew quadrupole mistuned on purpose is displayed in Fig. 5.

Slow and Fast Orbit Correction Interaction

The EBSS simulator has nevertheless a limitation, that is the speed of the computation loop. All fast systems such as fast orbit and emittance feedbacks are not available in the EBSS. Nevertheless, tests of the correct simultaneous use of Fast Orbit Feedback, Slow orbit correction, and setting of closed orbit bumps, were possible and allowed to exploit the EBSS to redesign the orbit correction logic for EBS in parallel to the User Service Mode operation (USM).

More Applications Developed for Commissioning

The EBS simulator also allowed the development and test of: beam based alignment functions, chromaticity measurement and correction, a Tango DS for the computation of optics at any given longitudinal coordinate, a procedure of measurement to evaluate the beam energy, the procedure to load new SR optics, and many more.

EBSS allows to perform realistic, real-time virtual Machine Dedicated Time (MDT). During real MDTs if a problem occurs, the solution can be looked for in parallel with the EBSS, while other MDT activities continue to take place.

TOWARDS A DIGITAL TWIN

The EBSS infrastructure is a first step towards a Digital Twin [14] of the Storage Ring. In fact, a complete digital twin continuously monitors the real machine settings and updates the expected output accordingly. Today it is manually possible to use a measured lattice to trigger the simulator computations and to set the strengths so that EBSS matches the real machine configuration. In order to move towards a digital-twin, this action would need to be automated. An instance of the PyRingSimulator device could then run in the real control system sourcing the currently loaded optics as a model - a trivial action requiring a simple modification of a Tango property - and use the real magnets settings - that are already expressed in strengths! - to update the optics model (see Fig. 6). This would allow to have constantly (at the speed of the simulator loop) an estimate of the expected orbit, tunes, dispersion and optics of the SR. The absolute comparison being far from easy to obtain with the existing lattice model at the ESRF, relative variations monitoring should be possible without major issues. A further improvement would be to include ID gaps models, such that it would be possible to monitor for example the optics modulations induced by ID gaps during USM.

Both improvements on the lattice modelling and on the control infrastructure are needed, since the magnet device servers are presently designed to work either on the real control system or in the EBSS. Regarding the control aspects of this problem, the idea would be to implement a bridge between the physical control system and its virtual counter-

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-MOPV012

part. The ability to access Tango devices across separated control systems makes such a task quite straightforward.

Another step towards a real digital twin is to connect the control device to either the real live devices or the archive database to display live or historical values as readings. This step would be useful for debugging applications, new algorithms and or using the digital twin as a time machine able to go back in time. Connecting the digital twin to the beamline control system simulator is another area to be explored. This would allow beamlines to test developments with a simulated copy of the accelerator i.e. the EBSS.

PORTING EBSS

There are a number of new projects started or planned to replace the current 3rd generation storage rings with a 4th generation one. For those projects which are using Tango for the accelerator controls or are interested in trying it out with a simulator, the EBSS offers a powerful solution. However there is a certain amount of work required to port the EBSS to a new storage ring.

To port the EBSS to another storage ring the following would need to be carried out:

- Define the devices according to the naming scheme of the new storage ring
- Implement simulation devices for each of the device classes. One option is to start off with a generic device simulator e.g. using the tango-simlib [15]
- Populate the Tango database with the simulated devices
- Define the optics of the new storage ring for pyAT
- Start the docker containers with the new simulators

CONCLUSIONS

The EBSS proved to be an extremely powerful tool for the preparation of the ESRF-EBS storage ring commissioning. It allowed to eradicate most of the coarse bugs and to include several additional features without any delay on the initial schedule. The EBSS was used with large profit also after the EBS storage ring commissioning and is presently used for the continuous development and upgrade of all the softwares for which a direct impact on the electron beam is expected.

The infrastructure that was set in place to realize the EBSS is such that future developments towards a Digital Twin are simplified. The final objective will be to continuously monitor the expected (simulated) beam properties from the storage ring control room, such as optics and response to magnetic fields variations. Such a development will also allow to implement Artificial Intelligence and Machine Learning algorithm that will constantly compare model vs measurement enabling: 1) early detection of faults, 2) slow drifts of parameters potentially linked to hardware failures, 3) continuous lattice optics model update, and 4) many other features.

The extension to other accelerators is presently not trivial, due to the intrinsic peculiarity of each control system. Nevertheless the strategy is set and at least the PyRingSimulator DS would require few modifications (the specific name of the devices) to be used for other storage rings, or high energy electron transfer lines. Specific developments will be needed instead to extend the use of the simulator to linacs, energy ramped accelerators and hadron accelerators.

The several aspects mentioned above will be addressed in the near future and are presently part of a proposal for a future EU collaboration project [16].

REFERENCES

[1] J. Biasci *et al.*, "A low-emittance lattice for the esrf," *Synchrotron Radiation News*, vol. 27, no. 6, pp. 8–12, 2014. DOI: 10.1080/08940886.2014.970931.

[2] S. White *et al.*, "Commissioning and Restart of ESRF-EBS," in *Proc. IPAC'21*, (Campinas, SP, Brazil), ser. International Particle Accelerator Conference, JACoW Publishing, Geneva, Switzerland, Aug. 2021, MOXA01, pp. 1–6, ISBN: 978-3-95450-214-1. DOI: 10.18429/JACoW-IPAC2021-MOXA01.

[3] S. Liuzzo *et al.*, "Preparation of the ebs beam commissioning," *Journal of Physics: Conference Series*, vol. 1350, p. 012 022, Nov. 2019. DOI: 10.1088/1742-6596/1350/1/012022.

[4] A. Götz *et al.*, "State of the Tango Controls Kernel Development in 2019," in *Proc. ICALEPCS'19*, (New York, NY, USA), ser. International Conference on Accelerator and Large Experimental Physics Control Systems, JACoW Publishing, Geneva, Switzerland, Aug. 2020, pp. 1234–1239, ISBN: 978-3-95450-209-7. DOI: 10.18429/JACoW-ICALEPCS2019-WEPHA058.

[5] G. Portmann, J. Corbett, and A. Terebilo, "An accelerator control middle layer using matlab," in *Proceedings of the 2005 Particle Accelerator Conference*, 2005, pp. 4009–4011. DOI: 10.1109/PAC.2005.1591699.

[6] R. Kammering *et al.*, "The Virtual European XFEL Accelerator," in *Proc. of International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'15), Melbourne, Australia, 17-23 October 2015*, (Melbourne, Australia), ser. International Conference on Accelerator and Large Experimental Physics Control Systems, Geneva, Switzerland: JACoW, Dec. 2015, pp. 578–580, ISBN: 978-3-95450-148-9. DOI: doi:10.18429/JACoW-ICALEPCS2015-TUD3O04.

[7] E. Taurel, D. Lacoste, and N. Leclercq. "Ebs simulator tango device servers," https://gitlab.esrf.fr/accelerators/Simulators/EbsSimulator (accessed: 08.10.2021).

[8] S. Liuzzo. "Python ring simulator optics computation," https://gitlab.esrf.fr/BeamDynamics/pythontools/optics_for_pyringsimulator (accessed: 08.10.2021).

[9] *Matlab*, The MathWorks, Natick, MA, USA, <2020a>.

[10] B. Stroustrup, "Thriving in a crowded and changing world: C++ 2006-2020," *Proc. ACM Program. Lang.*, vol. 4, no. HOPL, 70:1–70:168, 2020. DOI: 10.1145/3386320.

[11] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009, ISBN: 1441412697.

[12] W. Rogers, N. Carmignani, L. Farvacque, and B. Nash, "pyAT: A Python Build of Accelerator Toolbox," in *8th International Particle Accelerator Conference*, Copenhagen, Denmark, May 2017, THPAB060. DOI: 10.18429/JACoW-IPAC2017-THPAB060.

[13] ESRF, internal communication, 2012–2021.

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-MOPV012

[14] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technologies, challenges and open research," *IEEE Access*, vol. 8, pp. 108 952–108 971, 2020, ISSN: 2169-3536. DOI: 10.1109/access.2020.2998358.

[15] T. community. "Tango-simlib: Easily generate tango device simulators," https://github.com/ska-sa/tango-simlib (accessed: 05.10.2021).

[16] D. T. P. for Analytical Research Infrastructure Experiments (DiTARI), grant application proposal for European Commission's Horizon Europe framework programme HORIZON-INFRA-2021-TECH-01-01, 2021.