ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-FR2A005

PYTHON LIBRARY FOR SIMULATED COMMISSIONING OF STORAGE-RING ACCELERATORS*

L. Malina†, I. Agapov, E. Musa J. Keil, and B. Veglia
Deutsches Elektronen-Synchrotron, 22607 Hamburg, Germany
N. Carmignani, L. R. Carver, L. Hoummi , S.M. Liuzzo, T. Perron and S. White
ESRF, Grenoble, France
T. Hellert, Lawrence Berkeley National Laboratory, Berkeley, CA, USA

Abstract

Simulations of the commissioning procedure became vital to the storage-ring lattice design process. The achievable tolerances on lattice imperfections, such as equipment misalignments or magnet gradient errors, would, without correction, prohibit reaching the design parameters. We present a Python library which includes an extensive set of error sources in the accelerator lattice and provides a variety of correction algorithms to commission a storage ring. The underlying beam dynamics simulations are performed with pyAT. This project builds upon previous works and expands them in the direction of realistic control room experience and software maintainability. The performance is demonstrated using example commissioning studies, and further development plans are discussed.

INTRODUCTION

Fourth-generation storage ring (SR) synchrotron light sources aim at improving the hard X-ray source brightness with a reduction of their horizontal electron beam emittance, obtained with the extensive use of Multi-Bend Achromat (MBA) schemes [1–5]. Such compact lattices inherently feature strong focusing elements and strong nonlinear magnets, yielding highly nonlinear beam dynamics. The combination of strong nonlinearities and compactness, which leave less space for correction and compensation, makes these schemes highly sensitive to alignment and magnet strength errors. Operability of newly built or upgraded ultra-low emittance SR is conditioned to accurate understanding of imperfections, analysis of turn-by-turn beam data prior to the application of efficient beam orbit and optics correction methods during beam measurements [6–8].

Commissioning Simulations

Simulations have become indispensable in this process, as they allow for a meticulous examination of the complex interplay between various factors that can affect the performance of these cutting-edge accelerators [9–17]. In response to this challenge, we have developed a Python library that offers a comprehensive suite of error sources and a wide array of correction algorithms. This library can help operators and accelerator physicists to commission storage rings effectively. The aim is a start-to-end simulation of the machine commissioning beginning from first-turn trajectory correction, progressing to orbit correction, and culminating in lattice correction and coupling adjustment. Commissioning simulations enable the development of efficient and well-structured procedures for beam commissioning. These procedures encompass beam injection, acceleration, and beamline setup. Next-generation synchrotron radiation sources often represent significant investments, and minimising downtime is crucial. Commissioning simulations help identify potential bottlenecks, inefficiencies, or issues that could lead to extended downtime. By addressing these concerns in advance, facilities can reduce their so-called “dark time”.

In this paper, we present a new computational tool developed to meet the requirements of 4th generation machines and its application to the PETRA IV, ALS-U and ESRF-EBS SR lattices. The Python Simulated Commissioning, shortened in pySC, leverages modern open-source scientific libraries such as NumPy [18] and SciPy [19] and bases its lattice manipulation and simulations on the Python-based Accelerator Toolbox (pyAT) [20, 21], which gains momentum both in usage and support. We aim to conduct realistic commissioning simulations for electron storage rings, encompassing a wide range of error sources while accurately modelling beam diagnostics. The simulation allows for tests of the applications to achieve stored beam and adjust properties of the electron beam, such as orbit, tune and optics. The project builds mainly up on the *Toolkit for Simulated Commissioning* (SC) [11, 22] (written in Matlab and already used in several laboratories) and the tools developed at the ESRF [23], which are based on pyAT. Following a discussion on synergies of the two projects, translation of the SC library and later inclusion of ESRF’s tools, most importantly analytical formulas for response matrices, was considered the most efficient further development. The development began with translating the SC library into Python with the support of the Artificial Intelligence (AI) tool, ChatGPT [24].

TRANSLATION BY CHATGPT

In the translation, ChatGPT’s engine code-davinci-002 was used. Due to limitations on the length of prompts, the SC code was first stripped of all comments on separate lines. It was submitted to the engine along with a description of the input (Matlab code), the action (conversion) and the requested output (Python code) followed by the first lines of the assumed Python code, i.e. import numpy and matplotlib

* This project has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement No. 871072
† lukas.malina@desy.de

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-FR2A005


```python
def remove_comments_in_file(ffile: str, source_dir: str = "") -> str:
    with open(f"{source_dir}{ffile}", "r") as mfile:
        lines = mfile.readlines()
    new_lines = [l.replace("\t", "    ").strip("\n") for l in lines if not l.strip().strip("\t").startswith("%")]
    new_lines = [l for l in new_lines if l.strip(" \t\n")]
    return "\n".join(new_lines)


def matlab_to_python(matlab_code: str, python_start: str = "") -> str:
    intro = "# Convert this from Matlab to Python:\n# Matlab version \n\n"
    ending = "\n# End\n# Python version\n\n"
    prompt = intro + matlab_code + ending + python_start
    response = openai.Completion.create(model="code-davinci-002", temperature=0, top_p=1, frequency_penalty=0,
                                        presence_penalty=0, max_tokens=len(prompt), stop=[intro[:20]], prompt=prompt)
    return python_start + response.choices[0].text
```

Figure 1: Code snippet to translate code using ChatGPT's Python API.

and define a function name. Figure 1 shows a code snippet applied to all files in the SC toolkit.

Since the functions were translated one by one, and they contain calls to the other functions of SC toolkit, the signatures of translated methods needed to be adapted manually. The following features "allowed" in Matlab were complicated in automated translation:

- Implicit definitions of objects (typically arrays) by one of their member elements
- Calls to "functions", which are only substitutes for sections of code, i.e. not a function in a general sense.

Adding type hints with modern IDE proved very helpful in resolving the latter.

LIBRARY COMPONENTS

In this section, we discuss the data structures that store real and ideal lattices and the information on various sources of uncertainties. Later, the interfaces between real and ideal observation/settings are discussed and finally, available correction algorithms are listed.

Data Structures

In contrast to lattice design tools, software for error analysis and commissioning simulations must treat each lattice element as a unique entity.

Typical usage of the library proceeds through the following steps:

1. Initialisation of the SimulatedCommissioning class,
2. Definition and registrations of error sources,
3. Generation of a machine realisation including errors,
4. Interaction with the machine.

After initialising the core structure, the user registers elements like magnets, Beam Position Monitors (BPMs) or cavities, including all the relevant error sources in the structure. It is important to note that the uncertainties are only stored in a dedicated structure at this step: the ideal and actual rings are still identical. When applying errors, the uncertainties are used to generate random error distributions, assigned as attributes of corresponding elements in the lattice. The error sources currently available for inclusion in the commissioning simulation are the following:

BPM Errors

- Horizontal and vertical offsets,
- Noise (closed orbit, turn-by-turn),
- Roll around beam direction,
- Calibration, i.e. scaling error (2D),
- Sum error, i.e. error of the beam intensity measurement,

Magnet Errors

- Offsets in all three dimensions,
- Rolls around all three axes,
- Strength calibration,
- Multipole errors (strength offsets and calibrations),

Errors of three stackable types of support structures

- Offsets in all three dimensions,
- Rolls around all three axes,

Errors of injected beam

- Static errors (6D) present in every injection,
- Injection jitter (6D), different shot to shot,

RF phase, frequency and voltage errors

Ring circumference errors

Real to Ideal Interface

The purpose of the software is to adjust lattice parameters, such as magnet strengths, to reach the operational state using mostly beam-based corrections. Once the real lattice was created, the errors were assigned. However, they also needed to be kept during the lattice correction. For example, a magnetic field does not lose a scaling error when the current is set to a different value. In this sense, errors are persistent differences (typically relative or absolute) between a set

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-FR2A005

point value set by the user and the parameter used in the underlying beam dynamics simulations.

Such an interface is available to set normal and skew magnetic field components and RF-cavity properties.

The output from beam dynamics simulation, represented by tracking, features a similar interface for beam position monitor (BPM) reading. The interface covers several features:

- Averages trajectories/orbits of all tracked particles
- Offsets and rolls the orbit values
- Adds noise and sum signal errors. Note that in the trajectory case, the noise is inversely proportional to charge, i.e. if some of the particles are lost, the signals downstream will be noisier.

Only the interfaces mentioned earlier need to be modified to commission an already-built synchrotron, i.e. not in simulation. This is due to the design choice of strict separation of methods available to real machines and the analysis methods, which evaluate the performance of commissioning simulations.

Basic Correction Algorithms

Several important correction algorithms optimize the machine's performance and precisely control the beam's trajectory, orbit, RF settings and lattice optics, specifically, the following are presently available in `pySC`:

Trajectory correction
: The trajectory is iteratively acquired and corrected (via a response matrix approach) using the steerers upstream of the last BPM reached by the beam. After achieving first-turn transmission, an approximate one-turn closed orbit is searched by correcting the second-turn reading of the first several BPMs such that they are equal to their reading in the first turn.

RF setup
: The averaged trajectory at specified BPMs is logged turn-by-turn whilst the phase of RF is scanned (full 360°). The RF phase is set to a value for which the average orbit does not change turn-by-turn. Similar algorithm is also available to adjust RF frequency.

Closed orbit correction
: Similarly to trajectory correction, the closed orbit is corrected using the theoretical orbit response matrix (filtered with a maximum number of singular vectors or using Tikhonov regularization).

Beam based alignment
: Performs a trajectory or orbit scan in a BPM together with changing the strength of a neighbouring magnet. Fits result in a BPM reading corresponding to the centre of the magnet.

Tune working point scan
: Betatron tune working is scanned and the beam transmission is observed and optimised.

Tune and chromaticity
: Families of quadrupoles and/or sextupoles are used for adjusting the betatron tunes and/or the chromaticity to the desired values. This correction is possible once a closed orbit is established. Tune correction based on trajectory is under development.

Linear Optics from Closed Orbit
: (LOCO [25]) for the correction of linear optics and coupling based on the measurement of orbit response matrix data. In particular, the closed-orbit correction based on precise measurements recorded at the BPMs in response to current changes at the correcting magnets (ORMs). The Linear Optics from Closed Orbit (LOCO) algorithm is used for the linear optics calibration. This involves adjusting the parameters of the lattice model and iteratively fitting the response matrices of the model to the measured data to detect machine errors accurately.

The correction methods mentioned above utilise only the information available (directly observable) in a real machine. Thus, the same methods can be used to commission these machines, the only differences being the interfaces to the control system. Analytical functions to evaluate the simulation progress may use true values of errors and are well separated from the correction functions.

EXAMPLE SYNCHROTRONS

In this section, various examples from PETRA IV, EBS and ALS-U are shown, demonstrating the pySC's components/capabilities.

Petra IV

The commissioning chain of PETRA IV starts with various first and second-turn trajectory corrections. A sample of a trajectory being corrected is shown in Figure 2. Once the two-turn trajectory (including the one-turn closed orbit) has been corrected, the typical transmission of about 20 turns is achieved.

Later, some form of beam-based alignment appears to be necessary, and Figure 3 compares the BPM misalignments with respect to neighbouring magnets before and after beam-

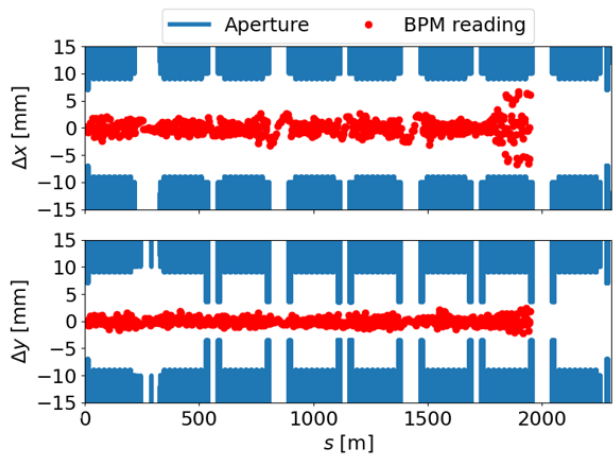

Figure 2: PETRA IV First-turn trajectory correction in progress using pySC.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-FR2A005

trajectory-based alignment, i.e. the figure only analyses the process of commissioning simulation and is therefore not available in actual commissioning.

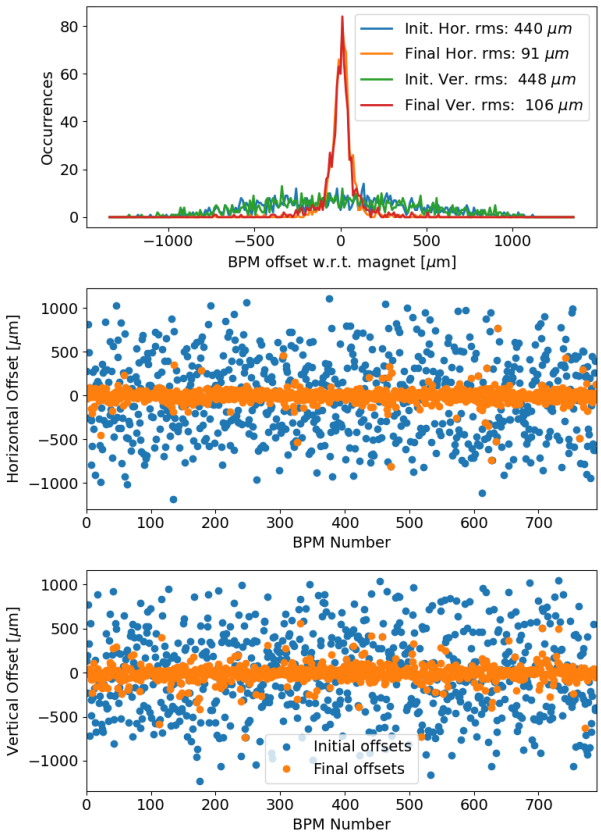

Figure 3: Beam-trajectory-based alignment of PETRA IV BPMs using pySC. The top plot shows distributions of misalignments of BPMs with respect to centres of neighbouring magnets: initial (in blue and green) and final (in red and orange). Middle and bottom plots show individual horizontal and vertical misalignments before alignment (in blue) and after alignment (in orange).

With the RF system still off, the typical transmission is limited to less than 100 turns due to energy lost via synchrotron radiation. Therefore, switching on and setting up the RF system (mostly phase and frequency) is vital. A full RF phase scan result is shown in Figure 4. Needless to mention, the result is fully observable, unlike the BBA example.

Similarly, the RF frequency is corrected, leading to most of the injected beam transmitted through 1000 turns, as shown in Figure 5 and the establishment of a closed orbit.

EBS

The pySC code is tested for the EBS lattice. Uncertainties are registered and applied in the lattice using the `register*` methods and `apply_errors`. The errors used are the alignment of quadrupoles and sextupoles of 60 µm rms in both planes and gradient errors in those magnets, for simplicity.

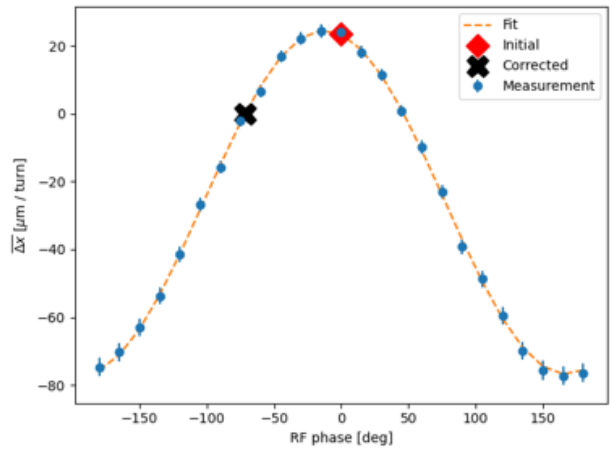

Figure 4: Fitted change of average trajectory turn-by-turn as a function of RF phase. The correction algorithm searches for zero-crossing, where the RF system replenishes the lost beam energy via synchrotron radiation.

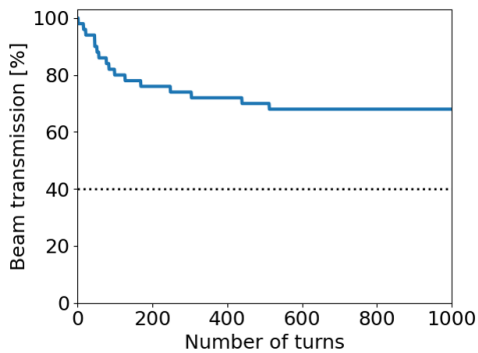

Figure 5: Typical PETRA IV beam transmission after RF phase and frequency correction.

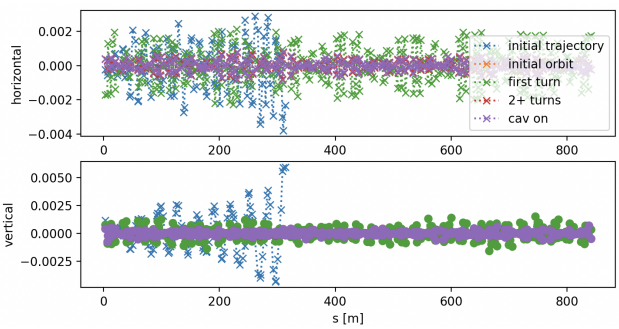

Figure 6: Trajectory and orbit correction for EBS using pySC.

With these errors, there is no single turn possible in EBS. Figure 6 shows how the trajectory/orbit correction functions can achieve transmission through the first turn and a corrected closed orbit without switching the sextupoles off.

Once closed orbit correction is achieved, `pySC` is instructed to run tune working point scan to optimize the

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-FR2A005

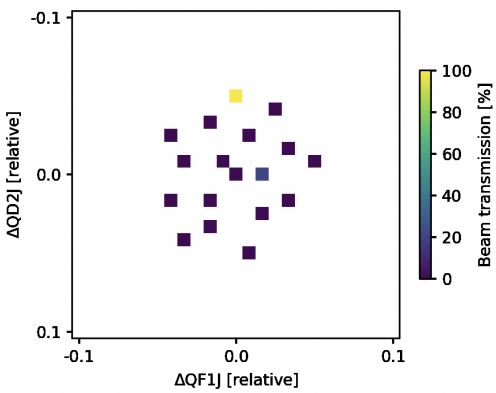

Figure 7: Tune working point scan to increase transmission efficiency.

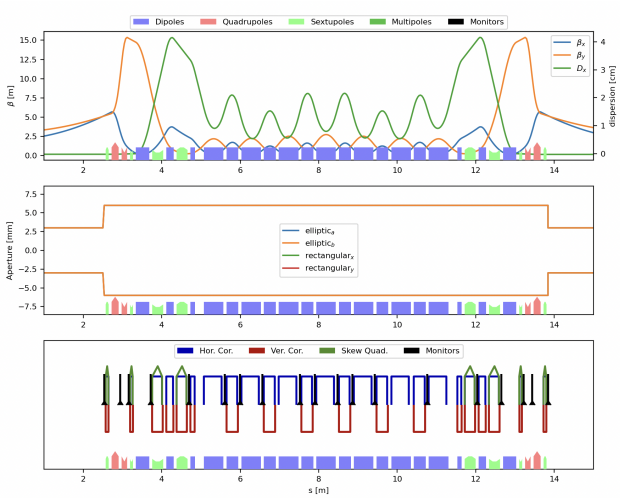

Figure 8: ALS-U Storage Ring cell as registered in pySC.

transmission over the maximum possible number of turns or higher transmission rate. Figure 7 depicts such a step of the correction. This computation is triggered only if no transmission over 200 turns is achieved.

ALS-U

The ALS-U Storage Ring correction chain [12] has begun to be ported into pySC. Preliminary steps include the registration of error sources and trajectory correction. For example, Figure 8 shows one lattice cell, including the registered diagnostic devices.

CONCLUSIONS AND OUTLOOK

A new library written in Python and based on standard widely used and well-tested software components, such as NumPy [18], has been developed and we are approaching its first (alpha) release. At the time of writing, it features more maintainable code measured by CodeClimate [26], and many automated tests run on every commit. Some of the correction algorithms have been elaborated utilising all the available statistical information. The effort to increase code maintainability, test coverage and performance will continue. Further elaboration of correction methods with modern statistical tools is also planned. This way, the commissioning simulation chain's automation level may increase and the necessary user input decrease.

REFERENCES

[1] P. Raimondi, C. Benabderrahmane, P. Berkvens, *et al.*, "The Extremely Brilliant Source storage ring of the European Synchrotron Radiation Facility", *Commun. Phys.*, vol. 6, no. 82, 2023. doi:10.1038/s42005-023-01195-z

[2] C. Steier *et al.*, "Status of the Conceptual Design of ALS-U", in *Proc. IPAC'17*, Copenhagen, Denmark, 2017, pp. 2824–2826. doi:10.18429/JACoW-IPAC2017-WEPAB104

[3] N. Carmignani *et al.*, "Operation Improvements and Emittance Reduction of the ESRF Booster", in *Proc. IPAC'18*, Vancouver, BC, Canada, 2018, pp. 4077–4080. doi:10.18429/JACoW-IPAC2018-THPMF017

[4] I. Agapov *et al.*, "PETRA IV Storage Ring Design", in *Proc. IPAC'22*, Bangkok, Thailand, 2022, paper TUPOMS014, pp. 1431–1434. doi:10.18429/JACoW-IPAC2022-TUPOMS014

[5] A. Loulergue *et al.*, "CDR BASELINE LATTICE FOR THE UPGRADE OF SOLEIL", in *Proc. IPAC'21*, Campinas, SP, Brazil, 2021, paper TUPAB054, pp. 1485–1488. doi:10.18429/JACoW-IPAC2021-TUPAB054

[6] P. F. Tavares *et al.*, "Commissioning and first-year operational results of the MAXIV 3GeV ring", *J. Synchrotron Radiat.*, vol. 25, no. 5, pp. 1291–1316, 2018. doi:10.1107/S1600577518008111

[7] P. Raimondi *et al.*, "Commissioning of the hybrid multibend achromat lattice at the european synchrotron radiation facility", *Phys. Rev. Accel. Beams*, vol. 24, no. 11, p. 110 701, 2021. doi:10.1103/PhysRevAccelBeams.24.110701

[8] L. Liu, M. Alves, A. Oliveira, X. Resende, and F. de Sá, "Sirius Commissioning Results and Operation Status", in *Proc. IPAC'21*, Campinas, SP, Brazil, 2021, paper MOXA03, pp. 13–18. doi:10.18429/JACoW-IPAC2021-MOXA03

[9] S. Liuzzo *et al.*, "Updates on Lattice Modeling and Tuning for the ESRF-EBS Lattice", in *Proc. IPAC'16*, Busan, Korea, 2016, pp. 2818–2821. doi:10.18429/JACoW-IPAC2016-WEPOW005

[10] V. Sajaev, "Commissioning simulations for the argonne advanced photon source upgrade lattice", *Phys. Rev. Accel. Beams*, vol. 22, no. 4, p. 040 102, 2019. doi:10.1103/PhysRevAccelBeams.22.040102

[11] T. Hellert, P. Amstutz, C. Steier, and M. Venturini, "Toolkit for simulated commissioning of storage-ring light sources and application to the advanced light source upgrade accumulator", *Phys. Rev. Accel. Beams*, vol. 22, no. 10, p. 100 702, 2019. doi:10.1103/PhysRevAccelBeams.22.100702

[12] T. Hellert, C. Steier, and M. Venturini, "Lattice correction and commissioning simulation of the advanced light source upgrade storage ring", *Phys. Rev. Accel. Beams*, vol. 25, no. 11, p. 110 701, 2022. doi:10.1103/PhysRevAccelBeams.25.110701

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-FR2A005

[13] Z. Martí *et al.*, "ALBA-II first tolerance studies", in *Proc. IPAC'23*, Venice, Italy, 2023, pp. 3104–3107. doi:10.18429/JACoW-IPAC2023-WEPL003

[14] H. Chao *et al.*, "Updates to Diamond-II storage ring error specifications and commissioning procedures", in *Proc. IPAC'23*, Venice, Italy, 2023, pp. 422–425. doi:10.18429/JACoW-IPAC2023-MOPA158

[15] O. Blanco-García *et al.*, "Status of the SOLEIL II robustness studies", in *Proc. IPAC'23*, Venice, Italy, 2023, pp. 3263–3266. doi:10.18429/JACoW-IPAC2023-WEPL070

[16] T. Hellert *et al.*, "Error Analysis and Commissioning Simulation for the PETRA-IV Storage Ring", in *Proc. IPAC'22*, Bangkok, Thailand, 2022, pp. 1442–1444. doi:10.18429/JACoW-IPAC2022-TUPOMS018

[17] S. Prakash, V. Meena, and R. Husain, "Preliminary simulation studies on closed orbit correction in hbsrs storage ring", presented at InPAC 2023, Mumbai, India, 2023, p. 64.

[18] C. R. Harris *et al.*, "Array programming with numpy", *Nature*, vol. 585, 2020. doi:10.1038/s41586-020-2649-2

[19] P. Virtanen *et al.*, "Scipy 1.0: Fundamental algorithms for scientific computing in python", *Nature Methods*, vol. 17, 2020. doi:10.1038/s41592-020-0772-5

[20] W. Rogers, N. Carmignani, L. Farvacque, and B. Nash, "pyAT: A Python Build of Accelerator Toolbox", in *Proc. IPAC'17*, Copenhagen, Denmark, 2017. doi:10.18429/JACoW-IPAC2017-THPAB060

[21] pyAT manual, https://atcollab.github.io/at/p/index.html

[22] SC manual, https://sc.lbl.gov/

[23] S. Liuzzo *et al.*, "Commissioning simulations tools based on python accelerator toolbox", in *Proc. IPAC'23*, Venice, Italy, 2023, pp. 386–389. doi:10.18429/JACoW-IPAC2023-MOPA142

[24] ChatGPT, https://www.openai.com/chatgpt

[25] J. Safranek, "Linear Optics from Closed Orbits (LOCO): An introduction", *ICFA Beam Dyn. Newslett.*, vol. 44, pp. 43–49, 2007.

[26] Codeclimate, https://codeclimate.com