ISBN: 978-3-95450-208-0 doi:10.18429/JACoW-IPAC2019-TUPGW021

AN ACCELERATOR TOOLBOX (AT) UTILITY FOR SIMULATING THE COMMISSIONING OF STORAGE-RINGS *

Thorsten Hellert†, Philipp Amstutz, Christoph Steier and Marco Venturini
Lawrence Berkeley National Laboratory, Berkeley, CA, USA

Abstract

We present the development of an AT-based toolkit, which allows for realistic commissioning simulations of storage ring light sources by taking into account a multitude of error sources as well as diligently treating beam diagnostic limitations.

INTRODUCTION

To achieve small beam emittance, diffraction-limited light sources employ aggressive designs based on high-gradient, small-aperture focussing elements. These make them particularly sensitive to magnet and other lattice errors, thus complicating the machine start-up. In addition, many of the new-generation light-source projects are upgrades of existing facilities for which a short dark time is an important goal. As a result, the traditional view that tends to see machine commissioning as somewhat disjoined from the design phase and to be pursued by following a more empirical, hands-on approach is widely believed to be inadequate. In virtually all 4th generation light-source projects, commissioning simulations are integral to the design effort and preparation work for commissioning [1–4]. This is particularly true for the Advanced Light Source Upgrade (ALS-U) [5–7], where the design challenges common to all new-generation light sources are magnified by the tight space constraints and concern both the Storage Ring and the Accumulator Ring to be built as an expansion of the existing ALS injection system.

Here we report on an extension to the MATLAB®-based Accelerator Toolbox (`AT`) [8], the *Toolkit for Simulated Commissioning* (`SC`), which was being developed to assist with the specification of the lattice and other error tolerances, testing of orbit/lattice correction strategies, and the definition of commissioning procedures. The toolkit specifically addresses the challenges of rapid commissioning in support of the ALS-U project but has been designed in full generality to apply to any storage-ring light source. In this contribution we give an overview over the design of toolkit and some of its central capabilities. The source code, including examples of application is available online [9]. Detailed descriptions of all functions can be found in the manual.

* This work was supported by the U.S. Department of Energy (DOE) under Contract No. DE-AC02-05CH11231.
† thellert@lbl.gov

TOOLBOX DESIGN AND USAGE

Realistic simulations of the operation of a complex machine like an accelerator require not only a good model of the beam dynamics but also the recognition that only incomplete information about the actual machine state is available during operation, due to the many unknowns in the machine geometry, magnetic fields, and beam-diagnostic systems.

In this spirit, the `SC` toolbox makes a clear distinction between machine parameters that are accessible during operation on the one hand (e.g. a magnet set-point or a BPM reading) and the parameters that go into the beam dynamics simulations (e.g. the field coefficients entering the symplectic integrator through a lattice element) and their results (e.g. actual beam offsets) on the other hand, thus providing a framework for supporting high-level scripts that simulate with realisms the various procedures (e.g. Beam-Based Alignment) to be encountered during commissioning, that closely mirrors the reality in the control room. The logic of this approach is captured by the workflow schematically shown in Fig. 1. Typical usage of the `SC` toolbox follows the steps 1) Initialization of the `SC` core structure, 2) Error source definition & registration, 3) Generation of a machine realization including errors, 4) Interaction with the machine, which are described in the following.

Initialization: In a first step, the user initializes the toolbox by calling `SCinit()` with the AT lattice of his or her machine as input. This sets up an MATLAB®-structure,

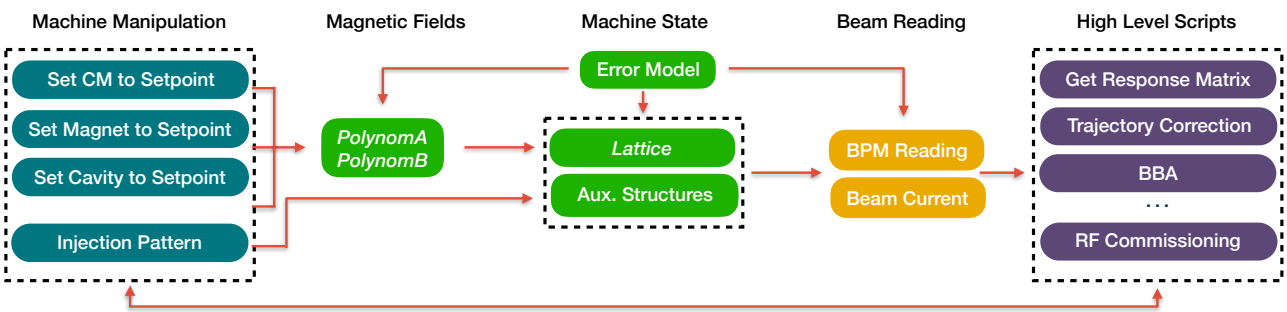

Figure 1: Schematic drawing of the workflow of the `SC` toolkit. The input for high-level scripts are only BPM readings as well as machine setpoints, which are distinguished from the actual machine parameters by using a transfer function.

ISBN: 978-3-95450-208-0 doi:10.18429/JACoW-IPAC2019-TUPGW021

usually assigned the variable name SC, with which nearly all subsequent functions of the toolbox interact. Within this central structure all relevant information about the machine and the error sources is stored. Having a single, isolated core data-structure like this allows for the state of the toolbox to be easily saved and loaded at the user's discretion.

Error Source Definition & Registration: In the next step, the user registers elements like magnets, BPMs or cavities including all error sources he or she would like to consider in the SC structure, using the `SCregister*()` function family. The `SCregister*()` functions typically take the ordinates of the elements in the lattice and values for the uncertainties for any of the parameters used by the AT tracking code, as well as some parameters specific to SC's error model as input. Element ordinates and uncertainties are centrally stored in `SC.ORD` and `SC.SIG`, respectively, and auxiliary fields in the lattice elements are initialized. Further, these functions are used to specify advanced properties of the elements which are subsequently accounted for by the toolbox; for instance the user here specifies which magnets are "split"- or combined-function magnets, or which magnets should be used as a dipole or skew quadrupole correctors including their limits etc.

Generation of a Machine Realization: Errors are randomly generated based on the uncertainties stored in `SC.SIG` and applied to the lattice via `SCapplyErrors()`. Typically, errors are modeled to follow a 2σ-truncated Gaussian distribution, where σ is the value specified in `SC.SIG`. Multiple calls to `SCapplyErrors()` produce a family of lattices realizations following the same error distribution, allowing to comfortably set up Monte-Carlo tolerance studies.

Interaction with the Machine: As motivated above, the operator / the control system of a particle accelerator only has incomplete information about the state of the machine and can influence the beam only indirectly. To account for this circumstance the SC toolbox implements the function `SCgetBPMreading()` and the function family `SCset*2SetPoints()`.

`SCgetBPMreading()` models the process of injecting a bunch into the machine and observing the readings of the BPMs previously defined by `SCregisterBPMs()`, taking into account injection errors, BPM offsets, BPM calibration errors, and more as described below.

Members of the `SCset*2SetPoints()` family model the process of assigning the set point of an experimentally accessible variable in the control system of the accelerator, for instance the strength of a quadrupole magnet. Based on these setpoints the actual simulation parameters going into the AT tracking routine are calculated by a subsequent call to, for instance, the `SCupdateMagnets()` function. This mechanism provides a powerful layer of abstraction, which also allows to easily extend and modify the underlying error models during further development of the toolbox, without the need to modify any existing user-side code.

All commissioning routines implemented in SC exclusively use these functions to interact with the machine, so that the commissioning simulation is conducted from the point of view of a real-world operator.

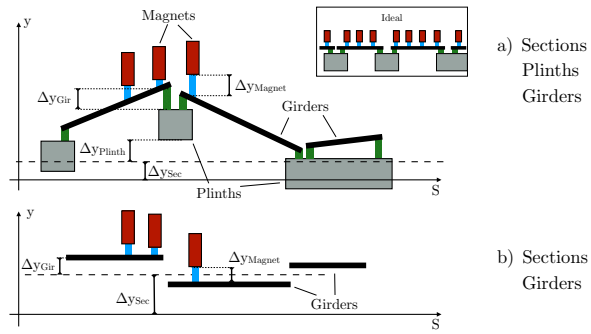

Figure 2: Illustration of different implementations of the misalignment model. Magnets are considered to be mounted on girders, which in turn may be mounted on plinths. In addition, section offsets may be considered. The lower plot shows a scenario without plinths and paraxial girders.

ERROR MODELS

Magnets: The magnetic multipole components used for tracking by AT, `PolynomA/B` are calculated by `SCupdateMagnets()` considering current setpoints $\vec{b}_{\mathrm{SP}}$, calibration errors $\vec{\delta}_{\mathrm{cal}}$ and field offsets $\Delta\vec{b}_{\mathrm{off}}$ according to $\vec{b}_{\mathrm{AT}} = \left[\underline{\mathrm{Id}} + \mathrm{diag}(\vec{\delta}_{\mathrm{cal}})\right]\vec{b}_{\mathrm{SP}} + \Delta\vec{b}_{\mathrm{off}}$. The field offset may include a bending angle error of a pure dipole magnet or can be used to specify (higher order) multipole errors, see `SCsetMultipoles()`. If the considered magnet is registered as a combined function magnet, the actual bending angle is considered to depend on the quadrupole set point. In order to capture that effect, the proper horizontal dipole field is added which results from a quadrupole set point variation from the design value. The field variations induced by a roll around the beam axis are calculated and applied on the `PolynomA/B` fields. It is thereby assured that the coordinate system remains unchanged by a rotation of dipole magnets.

Injection: The injected beam errors include a random shot-to-shot variation as well as a static offset from the 6D design injection-trajectory. Parameters of the injection pattern include the number of turns, particles per bunch, number of injections over which the BPM reading is averaged, the injected beam trajectory, the $6D$ beam σ-matrix and the choice of the tracking mode as described below.

- *turn-by-turn mode*: A bunch is tracked for the specified number of turns and the readings of the BPMs in each individual turn are returned.
- *pseudo-orbit mode*: The BPM readings are averaged over the turns, giving a good estimate of the orbit, without having actually achieved stored beam.
- *orbit mode*: It is assumed that stored beam has been achieved, so that the AT function `findorbit6()` can be used to determine the orbit.

ISBN: 978-3-95450-208-0 doi:10.18429/JACoW-IPAC2019-TUPGW021

Support and Alignment: The transverse misalignment model was developed to reflect the magnet support structure of the ALS-U facility and includes the concepts of girders, plinths, and sections as illustrated in Figure 2. Girders may have offset and roll errors, while sections and plinths (the concrete slabs on which girders are mounted) are currently considered to have offset errors only. The offset and roll errors of BPMs and magnets is a sum of their individual misalignment and the misalignment of the girder at the element location, see Figure 3 for an example. Individual longitudinal misalignments are not considered. However, a global circumference error is modeled by a scaling of all drift spaces.

Diagnostics: The calculation of BPM readings from particle trajectories takes into account BPM offsets, calibration errors, rolls, and BPM noise. If more than one particle is used for tracking, the BPM reading returns `nan` if more than a user defined fraction of the particles are lost. The toolbox can be configured to plot the trajectory of every injected beam, which is a valuable utility to discover potential problems of the commissioning procedures; an example of the 2-turn output for the ALS-U storage ring during early commissioning is shown in Figure 4.

CORRECTION ROUTINES

Diagnostic high level scripts include, among others, the simulated "measurement" of the response matrix and dispersion based on the current injection scheme. Further, many scripts have been implemented to determine various performance parameters of the lattice, such as the turn-by-turn beam transmission, dynamic- and momentum aperture as well as the beam life time. Based on these functions a variety of correction scripts for the simulation of the commissioning process are implemented, see Reference [7] for an example of a complete correction chain.

For the initial trajectory correction the toolbox implements an iterative approach using a Tikhonov-regularized version of the well-known SVD pseudo-inversion of the machine's trajectory response matrix, see [11]. Based

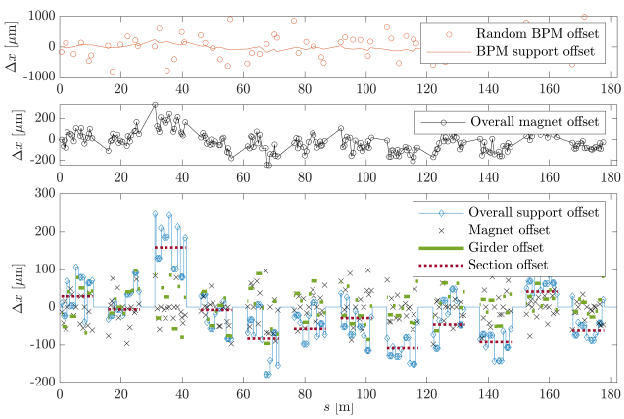

Figure 3: An example of the horizontal offset distribution for the ALSU accumulator ring using `SCplotSupport()`.

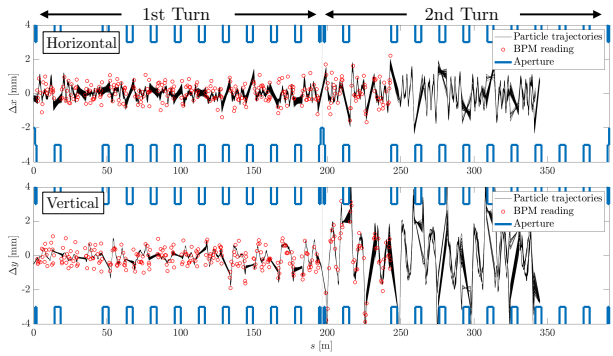

Figure 4: Plot of the injected beam for a realization of the ALS-U storage ring at an early commissioning step, showing the particle trajectories (black), BPM readings (red) and aperture (blue) for 200 particles and two turns.

on this method, the function `SCfeedbackFirstTurn()` can bring the machine from its uncorrected state to a state of full one-turn transmission. Subsequently, `SCfeedbackStitch()` achieves full two-turn transmission and `SCfeedbackBalance()` finally corrects the machine to a state with a period-one orbit, from which full transmission through a large number of turns can be expected. A final minimization of the BPM readings is achieved by the more generalized function `SCfeedbackRun()`, which also works in orbit mode and includes correction of the dispersion – with the RF-frequency as an adjustable parameter.

A simple but robust way to perform a coarse trajectory based linear optics correction in early commissioning is a tune scan `SCtuneScan()`; two quadrupole families are exercised coherently on a grid of setpoints in a spiral pattern until the target beam transmission is achieved. RF frequency and phase can be corrected using `SCsynchEnergyCorrection()` and `SCsynchPhaseCorrection()`. In both functions the horizontal turn-by-turn BPM variation is minimized in order to identify the synchronous frequency and phase.

Given the strong sextupole magnets in future machines, a single-pass beam-based alignment (BBA) procedure is most likely required in order to store beam while the perturbed lattice properties differ significantly from the design model. Hence a model-independent BBA procedure based on 2-turn trajectories is implemented.

Performing LOCO [12] based optics correction is an essential step during machine commissioning. To this end an SC-LOCO interface in terms of a library `SClocoLib()` has been developed to allow for convenient application of the established LOCO workflow while using the `SC` data structure.

REFERENCES

[1] Vadim Sajaev and Michael Borland. Commissioning Simulations for the APS Upgrade Lattice. In *Proceedings, 6th International Particle Accelerator Conference (IPAC 2015): Richmond, Virginia, USA, May 3-8, 2015*, page MOPMA010, 2015.

ISBN: 978-3-95450-208-0 doi:10.18429/JACoW-IPAC2019-TUPGW021

[2] Simone Liuzzo, Nicola Carmignani, Laurent Farvacque, and Boaz Nash. Lattice Tuning and Error Setting in Accelerator Toolbox. In *Proceedings, 8th International Particle Accelerator Conference (IPAC 2017): Copenhagen, Denmark, May 14-19, 2017*, page WEPIK061, 2017.

[3] H. Ghasem, M. Apollonio, R. Bartolini, J. P. Kennedy, and I. P. S. Martin, "Early Commissioning Simulation of the Diamond Storage Ring Upgrade", presented at the 10th Int. Particle Accelerator Conf. (IPAC'19), Melbourne, Australia, May 2019, paper TUPGW076, this conference.

[4] V. Sajaev. Commissioning simulations for the argonne advanced photon source upgrade lattice. *Phys. Rev. Accel. Beams*, 22:040102, Apr 2019.

[5] C. Steier *et al.*, "Design Progress of ALS-U, the Soft X-Ray Diffraction Limited Upgrade of the Advanced Light Source", presented at the 10th Int. Particle Accelerator Conf. (IPAC'19), Melbourne, Australia, May 2019, paper TUPGW097, this conference.

[6] T. Hellert *et al.*, "Simulation of Trajectory Correction in Early Commissioning of the Advanced Light Source Upgrade," J. Phys. Conf. Ser. **1067**, no. 3, 032001 (2018). doi:10.18429/JACoW-IPAC2018-THPMF078, 10.1088/1742-6596/1067/3/032001

[7] T. Hellert, S. C. Leemann, C. Steier, C. Sun, M. Venturini, and Ph. Amstutz, "Commissioning Simulation Study for the Accumulator Ring of the Advanced Light Source Upgrade", presented at the 10th Int. Particle Accelerator Conf. (IPAC'19), Melbourne, Australia, May 2019, paper TUPGW022, this conference.

[8] Andrei Terebilo. Accelerator toolbox for MATLAB. In *Workshop on Performance Issues at Synchrotron Light Sources Berkeley, California, October 2-4, 2000*, 2001.

[9] `https://sc.lbl.gov`

[10] `https://sc.lbl.gov/source`

[11] Ph. Amstutz and T. Hellert, "Iterative Trajectory-Correction Scheme for the Early Commissioning of Diffraction-Limited Light Sources", presented at the 10th Int. Particle Accelerator Conf. (IPAC'19), Melbourne, Australia, May 2019, paper MOPGW098, this conference.

[12] J. Safranek, G. Portmann, A. Terebilo, and C. Steier. MATLAB based LOCO. In *Particle accelerator. Proceedings, 8th European Conference, EPAC 2002, Paris, France, June 3-7, 2002*, pages 1184–1186, 2002.