HIGH LEVEL APPLICATION FOR FIRST TURNS COMMISSIONING IN NSLS-II STORAGE RING

S. Seletskiy[#], A. Blednykh, J. Choi, Y. Hidaka, B. Podobedov, G. Shen, L. Yang
BNL, Upton, NY 11973, USA

Abstract

The typical problems occurring during commissioning of the first turns in the storage rings include shorted coils or reversed polarity of the magnets, cross-cabling of magnet power supplies and reversed polarity of BPMs. In this paper we describe a dedicated high level control application, which was created and utilized for commissioning of the first turns in the NSLS-II storage ring.

INTRODUCTION

Years of experience in commissioning of storage rings (SR) show that obtaining first few turns commonly is one of the main challenges in establishing the beam in the SR. A typical storage ring is a complicated system consisting of hundreds (if not thousands) of devices. While it has been a routine practice to perform thorough integration tests prior to attempting the first injection into the ring, various hardware problems during SR commissioning with the beam are nonetheless rather common. Such problems usually hinder obtaining the first turns in the SR and result in significant loss of time.

Examples of the problems faced by various synchrotron facilities during commissioning the first turns in the storage rings include magnets with reverse polarities [1,2], cross-cabling of magnet power supplies [1], shorted magnet coil [3], obstruction of physical aperture in the SR [2,3], wrong polarity of beam position monitors (BPMs) [2] and mismatch between pulsed magnets waveforms [4,5].

Below we describe dedicated First Turn Analysis (FTA) software that was created with the goal of quickly finding possible problems described above. The FTA was successfully used for commissioning of the first turns in NSLS-II storage ring and proved to be a valuable tool [6].

GENERAL IDEA

The general idea of the First Turns Analysis tool was inspired by OPTIM [7]. Our goal is to combine interactive simulations of beam trajectory with live online measurements of beam position monitors. By interactive simulations we mean that when any parameter of the accelerator model is changed in the interactive GUI the user immediately sees the updated results on respective plots.

By updating the model simultaneously with the settings of the actual magnets and by plotting together the resulting beam trajectories predicted by the model and the live BPM readings one can obtain a rather powerful tool for analysis of the beam dynamics. Indeed, from comparing the simulated and the real trajectories one can easily deduce the source of their digression. The wrong polarity of the magnets can be quickly detected from trajectory response to the change in magnet settings. The same is true for cross-cabled magnets. As for the shorted magnet coils, the comparison of simulated and real differential trajectories shall be enough for their detection.

The beauty of this approach is that it does not require any additional off-line analysis. Any accelerator physicist equipped with such software shall be able to quickly pin-point the origin of the problem.

CODE DEVELOPMENT

As many other NSLS-II software applications [8] our code is written in Python [9]. For communications with the BPMs and magnet power supplies we rely on APHLA [10] – Python middle layer developed specifically for NSLS-II.

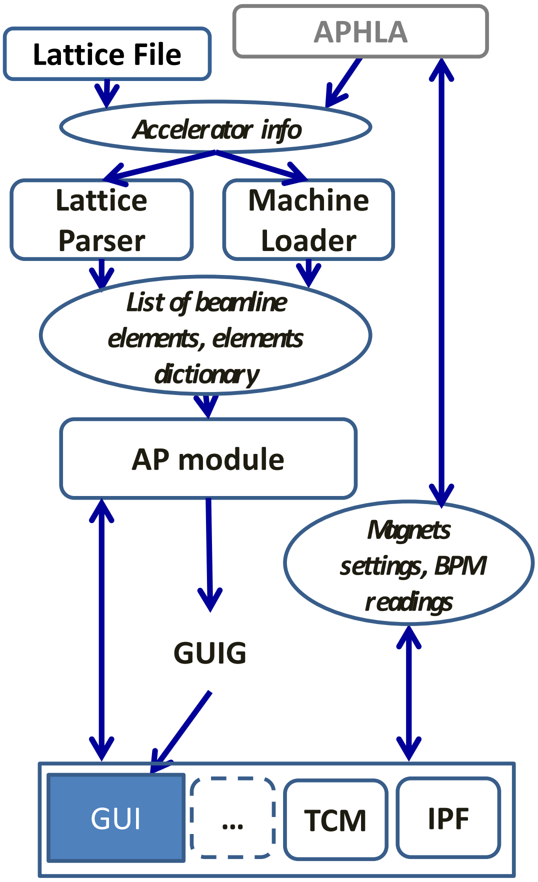


Figure 1: General overview of the FTA software architecture.

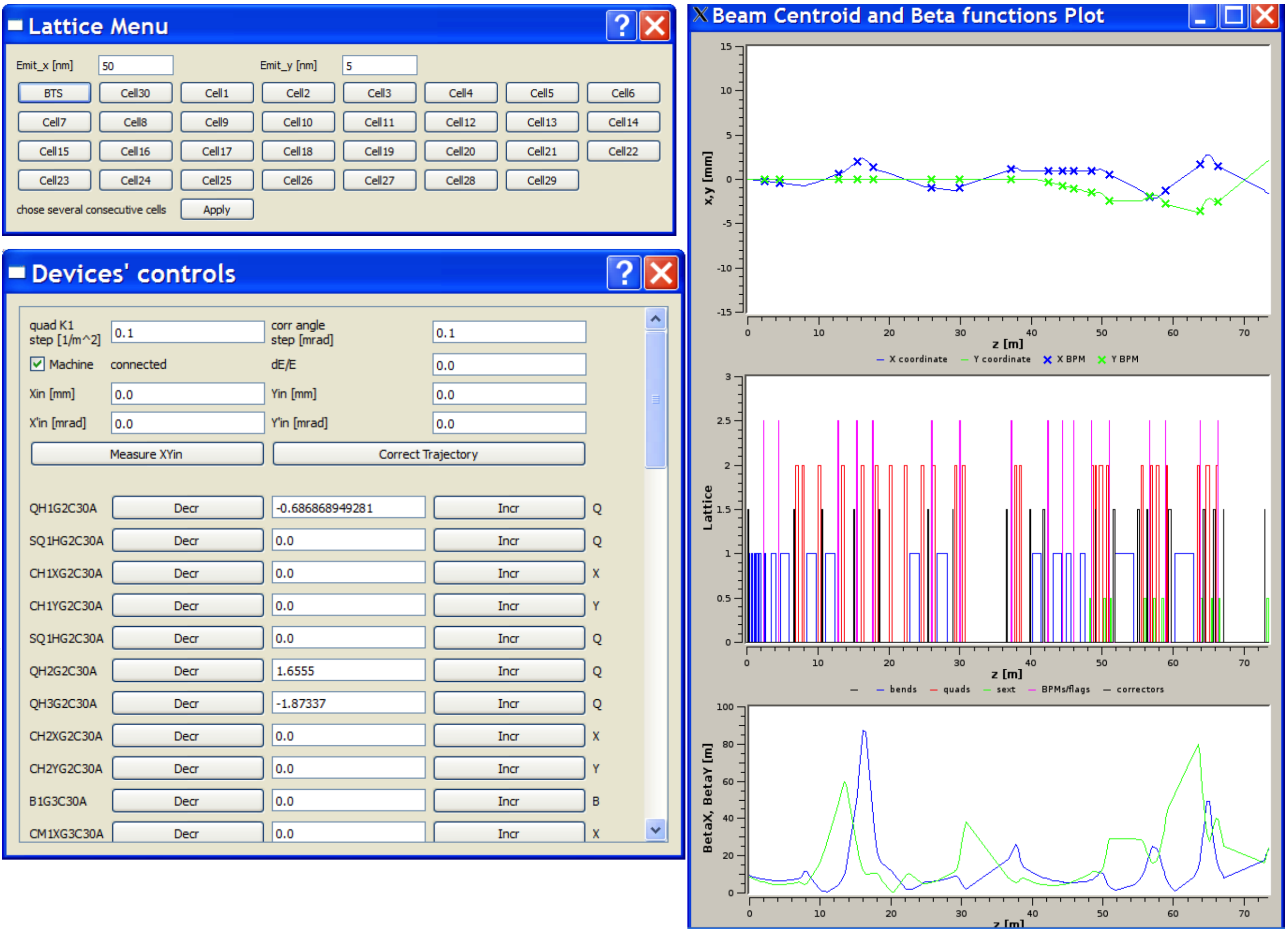
We took a modular approach to the design of the FTA code. Its most important modules include: the Accelerator

[#] seletskiy@bnl.gov

Physics Module (APM), the Machine Loader (ML), the Lattice Parser (LP), the Initial Parameters Finder (IPF), the Trajectory Correction Module (TCM) and the GUI Generator (GUIG). The architecture of the FTA software is schematically shown in Fig. 1.

All relevant information about the accelerator (such as the locations of the magnets and BPMs, magnet settings, and BPM readings) is obtained either from the lattice file or from the middle layer and processed by the Lattice Parser or the Machine Loader, respectively. The resulting Python list and dictionary of beamline elements are handed to the Accelerator Physics module, which generates a relevant accelerator model. We choose to work with the linear model only, since for the first turns commissioning all the SR sextupoles were switched off.

Finally, based on the model created by APM the GUI Generator creates the user interface shown in Fig. 2.



Emit_x [nm] 50
Emit_y [nm] 5

quad K1 step [1/m^2] 0.1
corr angle step [mrad] 0.1
☑ Machine connected
dE/E 0.0
Xin [mm] 0.0
Yin [mm] 0.0
X'in [mrad] 0.0
Y'in [mrad] 0.0

QH1G2C30A -0.686868949281 Q
SQ1HG2C30A 0.0 Q
CH1XG2C30A 0.0 X
CH1YG2C30A 0.0 Y
SQ1HG2C30A 0.0 Q
QH2G2C30A 1.6555 Q
QH3G2C30A -1.87337 Q
CH2XG2C30A 0.0 X
CH2YG2C30A 0.0 Y
B1G3C30A 0.0 B
CM1XG3C30A 0.0 X

Figure 2: The screenshot of the FTA GUI. The Lattice Menu Panel (top left) allows user to choose any number of Storage Ring cells for which the control panel shall be generated. The Devices Controls Panel (bottom left) allows for changing the magnet settings both in the accelerator and in the model as well as changing various model parameters. The right panel shows interactive plots with model trajectories, live BPM readings, and beta-functions.

From the FTA GUI a user can control both the real accelerator and its model. The model can be controlled separately allowing for quick and convenient fitting of the ideal model to the live machine. The GUI plot panel shows a clickable beamline lattice. Clicking the mouse on any lattice element displays the complete information about this element and allows one to plot the modelled transverse beam profile at this element. This feature is useful for comparing the model beam size to the one measured by insertion flags. Model beam energy and transverse emittances can also be modified from the GUI.

The initial displacement and angle of the beam trajectory used by the model can either be manually typed into the GUI or measured from BPM readings by the IPF module.

Trajectory can be corrected to any (as long as physics allows) user-defined positions at any set of BPMs with the Trajectory Correction Module.

TESTING THE CODE

While developing the FTA code we performed numerous simple checks. For instance, the APM was tested against OptiM and MAD8. The IPF and the TCM were tested with simulated BPM data and (in case of the IPF) predefined initial trajectory parameters.

The complete functional test of the software was performed with Virtual Accelerator (VA) [11].

Virtual Accelerator is dedicated software emulating the operation of the real NSLS-II machine in many aspects. At NSLS-II the Virtual Accelerator was named "JMS" (James Model Services) in acknowledgement of James Rowland's contributions to the field. VA uses TRACY [12] to simulate the beam dynamics. The APHLA provides communication to VA via EPICS channel access. Thus, for high level applications interaction with VA is very similar to interaction with the real accelerator.

The FTA code was tested by intentionally creating various "magnet errors" in VA and finding them with our software. An example of such work is demonstrated in Fig. 3.

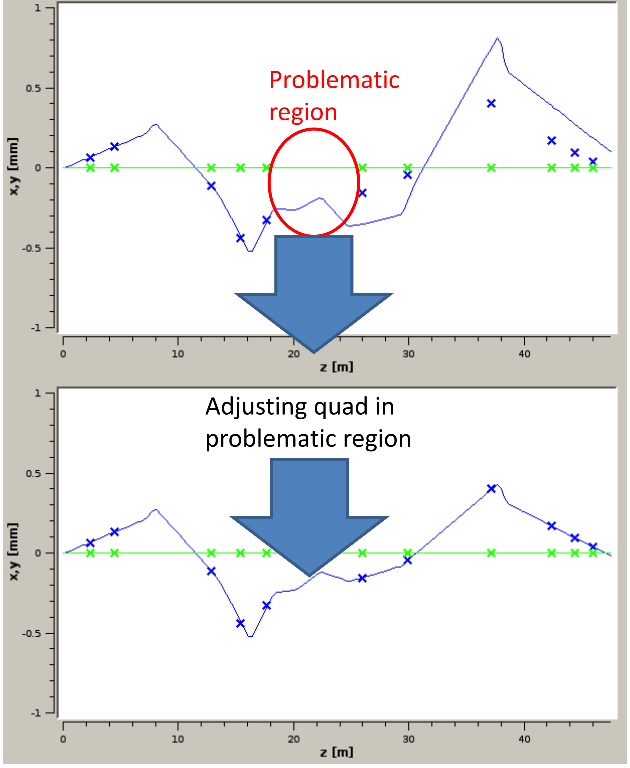


Figure 3: An example of functional test of the FTA software.

In this example when exciting the beam trajectory with one of the correctors we observe that BPM measurements and the model trajectory start diverging at some point. Next, we try adjusting the magnets in the problematic region one by one. We quickly find that adjusting the strength of one of the quadrupoles in the model by ~10% gives a perfect match between the model trajectory and the BPM readings.

DEPLOYING THE CODE

The FTA software was successfully deployed and utilized for first turns commissioning in NSLS-II storage ring. The details of first turns commissioning and description of how the FTA code was used for the task are given in [6]. Here we just point that it proved to be an extremely useful tool that allowed us to find and resolve hardware problems in timely fashion.

CONCLUSIONS

In this paper we presented dedicated software for first turns commissioning in the SR. This software was developed for quick and convenient beam-based diagnostics of various hardware problems one meets during initial commissioning of accelerators. The software is flexible and easily configurable. It provides a user-friendly interface that combines interactive simulations with live online measurements. It can be used as an on-line tool as well as for an off-line analysis. The described software proved to be a valuable tool for commissioning first turns in the NSLS-II storage ring.

ACKNOWLEDGMENTS

We are grateful for support from the NSLS. This work is supported in part by the U.S. Department of Energy (DOE) under contract No. DE-AC02-98CH1-886.

REFERENCES

[1] D. Einfeld, Proc. of IPAC11, MOXAA01 San Sebastián, Spain (2011).
[2] L. Dallin, Proc. of EPAC04, THPKF007, Lucerne, Switzerland (2004).
[3] A. Jackson, Proc. of PAC1993, p. 1432 (1993).
[4] R.P. Walker, Proc. of APAC'07, TUYMA03 (2007).
[5] R.P. Walker, Proc. of EPAC'06, THXPA01 (2006).
[6] S. Seletskiy et al., "Experience with First Turns Commissioning in NSLS-II Storage Ring," these proceedings, Proc. of IPAC2015, TUPMA053 (2015).
[7] V. Lebedev, http://www-ap.fnal.gov/~ostiguy/OptiM/
[8] L. Yang et al., Proc. of IPAC2012, THPPR018 (2012).
[9] Python Language, http://www.python.org/
[10] G. Shen, "NSLS II Physics Application and its Open Architecture," presented at the Fall 2014 EPICS Collaboration Meeting, Oct 23, 2014; http:// http://irfu.cea.fr/Meetings/epics/presentations/Thursday/NSLS%20II%20HLA%20Architecture%20G.Shen.pdf
[11] G. Shen et al., Proc. of ICALEPCS2013, MOPPC156 (2013).
[12] J. Bengtsson, TRACY-2 User's Manual, SLS Internal Document, February 1997; M. Böge, Update on TRACY-2 Documentation, SLS Internal Note, SLS-TME-TA-1999-0002, June 1999.