ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-TUPHA005

# TOOLS FOR NSLS II COMMISSIONING

G.M. Wang#, T. Shaftan, G. Bassi, A. Blednykh, W.X. Cheng, J. Choi, M. Davidsaver, L. Dalesio, J. De Long, K. Ha, Y. Hidaka, Y. Hu, Y. Li, D. Padrazo, S. Seletskiy, G. Shen, K. Shroff, O. Singh, T. Summers, Y. Tian, H. Xu, L. Yang, X. Yang, F. Willeke
BNL, Upton, NY, 11973, U.S.A

*Abstract*

The National Synchrotron Light Source II (NSLS-II) is a state of the art 3 GeV third generation light source at Brookhaven National Laboratory. As many facilities worldwide, NSLS II uses the EPICS control system to monitor and control all accelerator hardware. Control system studio (CSS) is used for simple tasks such as monitoring, display, setting of PVs. browsing the historical data, et. al. For more complex accelerator physics applications, a collection of scripts are mainly written in Python and part from Matlab during commissioning. With the close collaboration and fully support from control group, more and more CSS features were developed for operation convenience and several high level applications are interfaced with users in CSS panels for daily use based on softiocs. This paper will present the tools that we have been used.

## INTRODUCTION

The National Synchrotron Light Source II (NSLS-II) is a 3 GeV, ultra-small emittance (1 nm-rad), high brightness third generation light source [1]. The storage ring commissioning started in Mar. 2014 and took ~ two months and reached the key performance parameters (goal: 25 mA), 50 mA with superconducting RF cavity in July 2014 [2,3]. Since Feb. 2015, it started to provide the user operation for phase I project beam lines.

As many other facilities [4,5], NSLS II chose the Experimental Physics and Industrial Control System (EPICS) as its control system to monitor and control the accelerator hardware. It interfaces to the accelerator instruments and devices (such as Power supply, digitizer, and motor) with IOC (Input Output Controllers). Channel Access is used as the interface to the machine process variables (PVs).

The typical control applications include two types, simple monitor/control device and complex accelerator physics application for studies and machine optimization (measurements, data analyses and applying correction). Mainly, we use cothread to access the process variables (PVs) in Python script for complex accelerator physics applications and the CSS (BOY) [6] panel for simple tasks such as monitoring, displaying and setting of the machine Process Variables (PVs). The PVs history can be saved from data archive system. Besides, MATLAB is also available for programming and EDM panels are also used. All source codes including panel configurations are controlled and managed by mercurial version in the control system.

#gwang@bnl.gov

To keep the beam commissioning time efficient, various tools were developed for data displaying, collecting and processing. Besides machine status and physics applications, we also developed the daily used operation tools [7], such as save/compare/restore tool, data archive and browsing, alarm and warning monitor, elog, eticket, channel finder service and applications.

In this paper, we'll present our commissioning tools experience.

## MACHINE STATUS APPLICATION

All of the monitor and control panels are created with software tools in Control System Studio (CSS), which is an Eclipse-based collection of tools to monitor and operate large scale control systems. The interface allows the operators to edit the desired ramp/soak profiles and provides the option to use predefined recipes.

The NSLS-II operation panels [8] include the user panels and the expert panels. They display the device status and parameters (setpoint and readback) and show the device performance. A user panel shows only information that user is required, such as the magnet current setpoint, readback or fault status. The expert panel shows all the information that expert can fully control the device or diagnose the device, such as power supply operation mode, fault details.

## HIGH LEVEL APPLICATION

Besides the devices status and control, we also develop high level applications to realize the accelerator physics application tools. It is to translate the raw/processed data from PV to the "physics" parameters, such as from beam size reading to beam emittance or orbit measurement and correction.

An extensive set of libraries and scripts [9] have also been written in Python for more dynamic and physics related study. A high level physics applications (HLAs) library, which maps flat EPICS process variables (PVs) to object oriented accelerator components, was developed. This mapping is dynamical and configurable from channel finder service (CFS). CFS associates PVs with properties and tags. HLAs library constructs an accelerator structure based on PVs properties and tags and make them with the search feature and group manipulatable. This makes high level application user more convenient, such as operation type of elements (BPMs, correctors) or individual element based on their name pattern, position… HLAs library also wraps the unit conversion functions between physics unit and hardware unit.

ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-TUPHA005



| | |
|---|---|
| elemName=FYM1G4C02A | element name |
| devName=FM1G4C02A | device name |
| elemType=VFCOR | element type |
| cell, girder=C02, G4 | cell name |
| symmetry=A | symmetry |
| elemField=y | element field |
| handle=READBACK | read/write |
| sEnd,length=65.5222,0.044 | physics location, length |
| ordinal=264 | index in lattice file |

Figure 1: Example of Channel Finder data properties tags.

There are different interfaces to use high level applications. Ipython notebook is the most popular environments for accelerator studies with the feature to show the input code and output results including plots in the same page. Figure 2 shows the example of beta beat and dispersion correction in IPython notebook.

```
# --- combined RM for phase advance and dispersion
combrm = np.append((1.0-dispWeight)*tparm,dispWeight*drm,axis=0)

# --- calculate needed corrections
dph = np.append(phaseBeatx,phaseBeaty)
detax = etax-etax0
dphetax = np.append((1.0-dispWeight)*dph,dispWeight*detax)

quads = ap.getElements('QUAD')
Kq = [q.get('b1')/q.length for q in quads]

dK = np.linalg.lstsq(combrm,dphetax,rcond=0.05)[0]

plt.figure(figsize=(15,5))
plt.bar(np.arange(len(dK)),dK)

plt.xlabel('Quadrupole index')
plt.ylabel(r'$\delta K$')
plt.show()
```

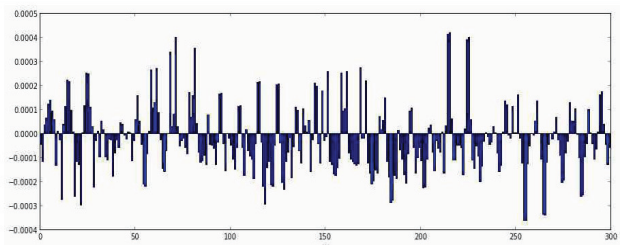

Figure 2: Ipython notebook for beta beat and dispersion correction.

These high level applications are highly demanded during commissioning, include orbit response matrix measurement, orbit correction, beam based alignment, local bump, beta function measurement and correction with BPMs turn by turn data, tune measurement and correction, dispersion measurement and correction, chromaticity measurement and correction, coupling measurement and correction, insertion device lookup table measurement.

## SOFTIOC

As python based high level applications has very powerful function for scientific calculation and logic control, CSS has the simple feature to display/control PVs with simultaneous data sharing environment, the soft IOCs based applications were developed and widely used for highly operation demanding high level applications, such as orbit correction, tune measurement. The source codes for scientific calculation are programmed with Python, but not limited, and the code operation and interesting input/output data communication are realized through IOCs. Their GUIs are implemented in the CSS, as part of operator panels, so that the operators have consistent operation interface environment and live information. The user can set parameters and monitor the progress in GUI as regular device PV set and monitor.

These are the advantages of using soft IOCs. Firstly, users do not need to know how to run different language code. Second, users simultaneously share the same information through PVs. Third, it avoids the code path dependence. CSS or user does not need to know where code is and what the script is. Finally, the related indirect hardware operation activity can be archived, which is very important for machine monitor and problem diagnoses.

But soft IOC also has limitations, including that the code depends on the soft IOC status and CSS has limited plot function feature.

Currently, these applications are developed in softiocs for operation purpose and general beam studies: magnets cycling, filling pattern control, slow orbit feedback, local bump, BPM slicer, tune measurement and correction, chromaticity measurement and correction, dispersion measurement.

### *Filling Pattern Control*

An important control function of the timing system is the triggering of the Electron Source, to determine the precise time that electrons enter the storage ring. By varying this time it shifts the whole injector timing relative to SR timing and place injection beam in SR different bucket. On top of this, the SR filling pattern application is developed for daily operation. The interface is shown in Figure 3. The user should set the desired current and pattern. Once the SR DCCT current reaches the desired current, this application will stop the injection.

User can also stop the injection any time manually. There are three patterns to choose, Fill one, uniform fill and pattern file. 'Fill One' is for single bucket injection, setting with the bucket number. 'Uniform fill' is for the fill pattern that can be described with the number of gaps and fill percentage. 'Pattern File' is for any general pattern fill, which user defines the filled bucket start number. It covers the above two pattern, but needs to set a waveform into the PV.

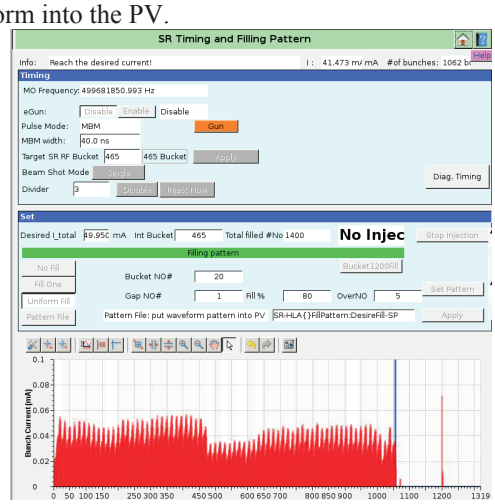


Figure 3: SR filling pattern control panel.

ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-TUPHA005

### *BPM Slicer*

BPMs slicer page was initially developed to monitor the beam injection status. Later more and more featured were added. Figure 4 shows BPMs slicer application page. It reads all the BPMs TBT and FA and displays/manipulate the same turn data in waveform. It displays BPMs healthy status, sets SR BPMs into different mode user defined delay, displays turn by turn data position and sum signal and compare the difference, 10 kHz data position and sum signal and compare the difference. This application is convenient to find beam loss location and loss turns, monitor the injection beam mismatched induced betatron and synchrotron oscillation, monitor the injection kickers mismatch induced residual oscillation amplitude, monitor RF induced synchrotron oscillation and coupling monitor.

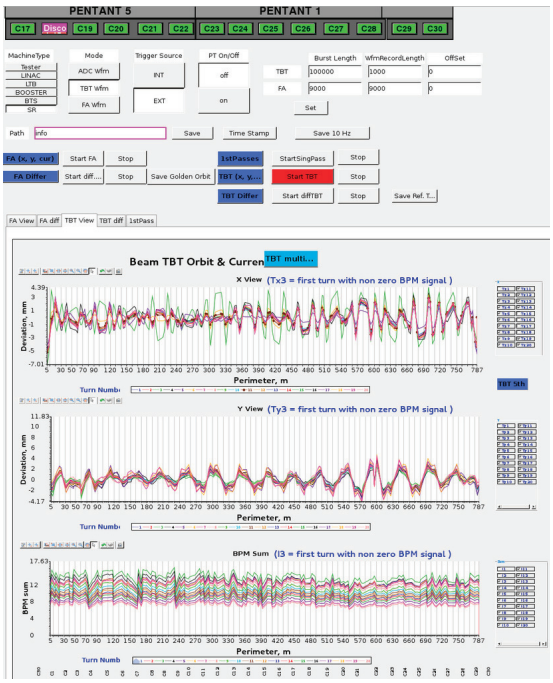


Figure 4: BPMs slicer application.

### *Slow Orbit Feedback*

NSLS-II slow orbit feedback (SOFB) system is designed to suppress the beam movement caused by the low frequency noises such as ground motion, slow drift etc. It runs at 0.5 H, including the BPM for orbit measurement, feedback calculation, and slow correctors for orbit correction. In the feedback calculation, the orbit response matrix includes 360 BPMs, 360 correctors and each plane correctors sum angle minimization. The SVD calculation is carried out by a python script with Tikhonov regularization method to remove the numerical instability caused by small eigen values. After optimization, the Kp and Ki are set to 0.5. The regularization parameter α initially was set to 0.5 and the orbit correction does not converge with residual orbit p2p ~100 um. After optimization, α was set to 0.005 and the residual orbit p2p was converged to < 10 um.

### *Tune Measurement from BPMs TBT Data*

Different methods have been used to measure the betatron tune in NSLS II, such as BPM TBT data Fourier spectrum with fast kickers or pinger excitation betatron oscillation, dedicated sweeping tune measurement system, bunch-by-bunch feedback system spectrum. Figure 5 shows BPMs TbT tune measurement application. It is based on python tools and interfaced with CSS for precisely determine and monitor tunes. The tunes are calculated from 20 BPMs TbT FFT. The average spectrum is inspected locally around the peaks of the FFT, with the use of Discrete Time Fourie Transform. This allows a precise peaks scan for accurate tune determination. Up to five local peaks are scanned by the TbT tune measurement system. It also allows precise determination of the synchrotron sidebands from the dispersive BPMs spectrum, very important for chromaticity studies and longitudinal dynamics characterization.

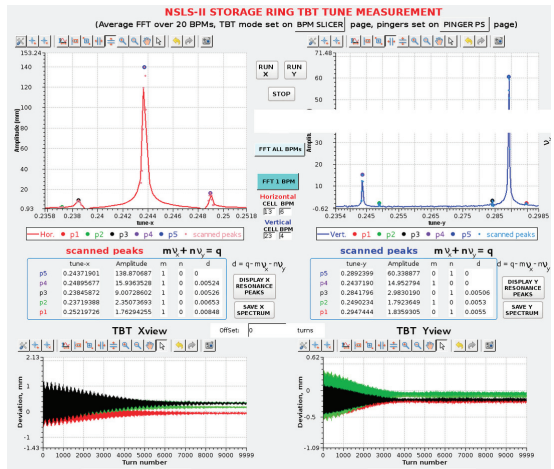


Figure 5: Tune measurement from BPMs TBT data.

## ACKNOWLEDGEMENTS

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

## REFERENCES

[1] F. Willeke, STATUS OF NSLS-II, PAC2011.
[2] G. Wang, Results of the NSLS-II commissioning, Bulletin of the American Physical Society, Vol. 60, Num. 4, APS April Meeting 2015.
[3] F. Willeke, Commissioning Results of NSLS-II, IPAC15, 2015.
[4] Emery, L., Commissioning Software Tools at the Advanced Photon Source, p 2238 - 2240, PAC1995.
[5] R. Bartolini et al., High level software for DIAMOND commissioning and operation, THPCH112, EPAC2006.
[6] http://cs-studio.sourceforge.net/
[7] G.M. Wang et al., NSLS II COMMISSIONING TOOLS, MOPHO17, PAC2013.
[8] G.M. Wang et al., Preparation for NSLS II linac to booster transport line commissioning, MOPPR094, IPAC12.
[9] L. Yang et al., The Design of NSLS-II High Level Physics Applications, TUPPC130, ICALEPCS2013.