THE DESIGN OF NSLS-II HIGH LEVEL PHYSICS APPLICATIONS*

L. Yang†, J. Choi, Y. Hidaka, Y. Li, G. Shen, G. Wang
Photon Science Directorate, BNL, Upton 11973, USA

Abstract

The National Synchrotron Light Source II (NSLS-II) is a state-of-the-art third-generation light source under construction at Brookhaven National Laboratory (BNL). In this paper we present the development of high level physics applications for both machine commissioning and routine accelerator physics studies. Both of the client applications and the supporting services are discussed.

INTRODUCTION

NSLS-II (National Synchrotron Light Source II) is a state-of-the-art third-generation light source under construction at BNL (Brookhaven National Laboratory). The commissioning of LINAC has been finished. The Booster commissioning is coming this winter by the vendors and the storage ring commissioning is being prepared for early 2014.

The high level applications for machine operation and study are collabrative work between the controls group and physics group [1] and the operators are also involved recently. We classified the high level applications into two categories: static panels for monitoring and scripts for measurement and active controls. The monitoring panels represent a group of relative machine parameter values shown in a variety of widgets (e.g. tables, meters, progress bars) in different colors. They are relatively static. When an monitored parameter goes out of a pre-defined range, alarms are triggered and alert the operators by a widget color change as well as an audible alarm. The second category is more dynamic and it involves some physics logic. The controling parameters could change with time or the machine status, and the output must be produced in a predefined format and location. Many physics applications and scripts are of this type.

In NSLS-II, we have chosen Control System Studio (CSS) [2] to build the static panels. CSS is a collection of tools based on Eclipse to monitor and operate large scale control systems. An extensive set of libraries, scripts and GUI applications have also been written in Python programming language [3] to do more dynamic and physics related controls. The Matlab Middle Layer Toolkit (MMLT) [4] that has been used worldwide by accelerator physicists in synchrotron light sources will also be available.

* Work supported by Brookhaven Science Associates, LLC under Contract No. DE-AC02-98CH10886 with the Department of Energy.
† lyyang@bnl.gov

STATIC PANELS

CSS has a full set of widgets to build static panels and its GUI (Graphical User Interface) designer program allows a user to readily create such a panel mainly by drags and drops. Each displayed numerical value and/or color in a CSS panel is linked to a process variable whose setpoint, readback, and alarm status are retrieved and updated by CSS.

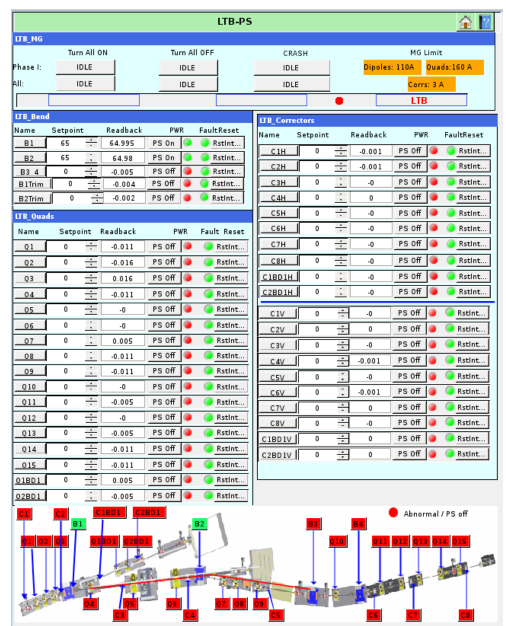

Figure 1: Linac to booster (LTB) transport line power supplies panel.

Fig. 1 is the panel for Linac to booster (LTB) power supplies and Fig. 2 is the LTB diagnostics panel. The expert panels with more detailed information can be opened from these panels. External Python or shell scripts can be executed from within these panels as well.

In the commissioning of Linac and LTB transfer line, CSS has proven to be a convenient tool for physicists and operators to build panels to view well processed results.

PHYSICS SCRIPTS AND APPLICATIONS

Scripting capability is very important for us to monitor and control the accelerator efficiently at a higher level, Matlab and Python are a few such environments and languages. In NSLS-II, we primarily use Python for new development, while keeping MMLT as our alternative tool. Due to its long history and wide acceptance, MMLT has plenty of useful tools and are well tested. Linear Optics from Closed Orbits (LOCO) is one of them. This approach allows us

ISBN 978-3-95450-139-7

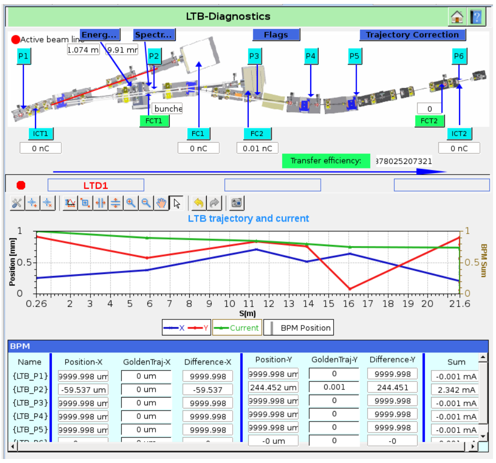

Figure 2: Linac to booster (LTB) transport line diagnostics panel.

to integrate better with the client-server mode which was initiated in controls group and makes the high level controls environment more comfortable for our collaborators and ourselves who are not very familiar with Python.

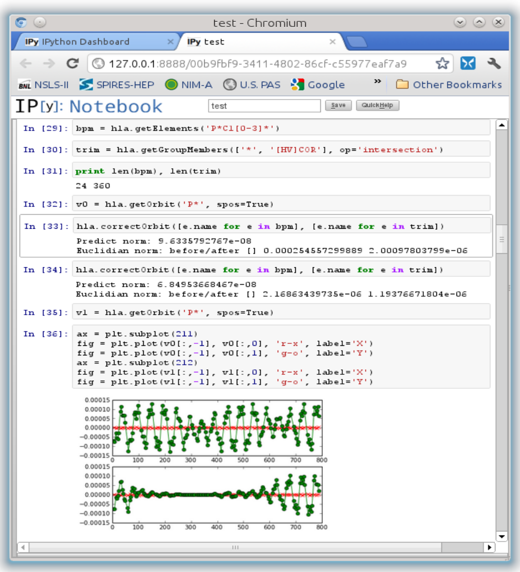

Figure 3: IPython notebook as an interactive environment for accelerator modeling and control. The input code and output results including plots are on the same page and easy to print and share.

Python comes with a lot of libraries and has strong support from scientific computing community. We have adapted some widely accepted and well supported tools to faciliate our scripting and GUI development. For example, *NumPy*, *SciPy*, *IPython*, *Matplotlib* and *PyQt4*. Fig. 3 shows an IPython notebook example for interactive scripting and control. *PyQt4* is our primary tool to build GUI applications as shown in Fig. 4. We have also developed Py-Tracy [5] that wrapps the C/C++ Tracy-III [6] for Python.

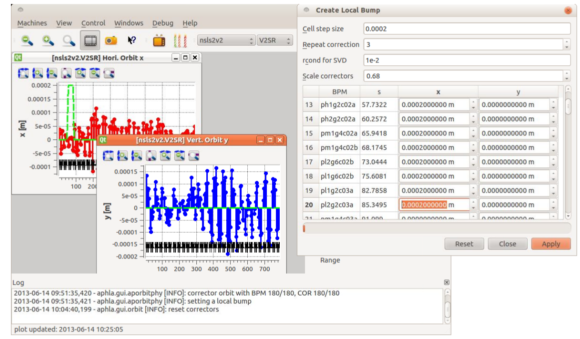

Figure 4: A GUI application for various high level controls developed with PyQt4.

The first layer of our high level physics applications (HLAs) is the core library which maps flat EPICS process variables (PVs) to object oriented accelerator components. This mapping is dynamical and configurable from channel finder service (CFS) [7]. As a directory service, CFS associates PVs with properties and tags. The properties could be element name or position for all physicists and operators, whereas tags are for a specific owner and usage. The HLA constructs an accelerator structure based on these properties and tags. There can be multiple accelerator structures initialized from different sources. This make it possible to have a virtual structure that is a combinations of several accelerating structures or is a subset of one big structure. This is done simply by adding an extra tag for each relevant component to state its extra owner.

On top of the accelerator components, we rely on searching or matching to find the elements needed, e.g. a magnet or diagnostics instrument. This matching can be based on the pattern of their name, position, type, family, all of which are all from CFS data.

The physics routines, e.g. measurements and feedbacks, are done on the matched elements with a high level grammar (higher than channel access). This makes the routines portable across accelerator facilities. The following is an example for dispersion measurement and plotting.

```
import aphla as ap
ap.machines.init("nsls2")
ap.machines.use("SR")
bpmobj = ap.getElements('p*c0[3-6]*')
bpmnames = [b.name for b in bpmobj]
f0 = ap.getRfFrequency()
f = np.linspace(f0 - 1e-5, f0 + 1e-5, 5)
for i,f1 in enumerate(f):
    ap.putRfFrequency(f1)
    time.sleep(6)
    obt2 = ap.getOrbit(bpmnames)
    codx[i,:] = obt2[:,0]
    cody[i,:] = obt2[:,1]

ap.putRfFrequency(f0)
dxc = codx - codx0
```

ISBN 978-3-95450-139-7

Table 1: Channel Finder Data for PV 'SR:C02-MG:G04BHCor:M1Fld-I'

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

```
df = -(f - f0)/f0/eta
p = np.polyfit(df, dxc, 1)
plt.plot(s1, p[0,:], 'o--', label="Fit")
```

Thanks to the Python programming language, in the above example code, the controllable properties of elemens in `bpmobj` can be introspected with the standard `dir` and `help` functions.

SERVICES

With the client-server approach initiated by the controls group, the physics group can focus more on the meansurements and control algorithms at the client side. The services provided by controls group cover almost all non-or-less physics related area which includes virtual accelerator [8], channel finder service, machine snapshot save and restore, unit conversion, lattice model service [9], twiss data service etc.

The channel finder service provides properties and tags to PVs available in all IOCs. Once the engineers or physicists prepared properties and tags, they are searchable across the whole controls network. This makes it easy for an operator or physicist to find the data he/she needs. Table 1 shows the properties of the PV for a vertical corrector from our virtual accelerator.

CFS also allows the user to tag each PV with different strings.

NSLS-II has done various measurements of the magnets and insertion devices. For some families, all the units have been measured, while only a few units have been measured for the other families. The unit conversion service can convert the driving current to the magnetic field. With a backend database, the service supports several ways of interpolating the data among several different units for the same quantity (one PV).

A lattice model server and twiss data server are provided by the controls group to manage the online model and its computed twiss data. The online machine data are first retrieved and converted to physics units. These readings are then feed into a simulator to calculate some quantities interesting to the physicists and operators. A set of predefined random errors can be applied optionally. The archived data will allow us to see and analyze the history of both the model and real machine.

INSERTION DEVICE

The 2nd order radia kickmaps are used to simulate insertion devices (IDs) in the TRACY-III-based virtual accelerator. The first and second field integrals from the measurement are simulated by four virtual kicks (two for each plane) located at the ID extremities. The closed orbit distortion due to the imperfection of ID will be compensated by four on-board correctors (as shown in green). The calibration function of corrector strength to realize this feed-forward compensation has been successfully implemented and tested on the virtual machine. In the next step, we will use different kickmaps to simulate an ID at different states (gap/phase) to calibrate other feed-forward tables if necessary.

The commissioning process of IDs has been planned. The first ID's on-board correctors will be calibrated with beam under various status. Then we adjust the IDs to their passive states (max. gap, and zero phase if applicable). The closed orbit under this passive states will be monitored and recovered to the golden orbit. Next we will vary the IDs to different status and measure some critical beam parameters, including tunes, closed orbit distortion, optics, injection efficiency, lifetime, transverse coupling and so on. Based on the measurement results, feed-forward tables for the on-board correctors, adjacent normal/skew quadrupoles (for optics/coupling compensation) will be created. All measurement results will be archived in a centralized database. For EPUs, we will measure the non-linear effects by displacing beam off-axis, to calibrate the current strip compensation. The calibrated feed-forward tables will be distributed to the local IOCs to realized the real-time compensation. Measurement data will be analyzed offline and compared with simulation.

ACKNOWLEDGMENT

We would like to thank K. Shroff and G. Carcassi for their support on CSS and channel finder service, O. Chubar for ID commissioning plan.

REFERENCES

[1] L. Yang, J. Choi, Y. Hidaka, G. Shen, G. Wang, *Development Progress of NSLS-II Accelerator Physics High Level Applications*, IPAC 2012.

[2] *Control System Studio*, `cs-studio.sourceforge.net`

[3] *Python*, `www.python.org`

[4] G. Portmann, J. Corbett, A. Terebilo, *Middle Layer Software Manual for Accelerator Physics*, LSAP- 302, 2005; J. Corbett, A. Terebilo, G. Portmann, *Accelerator Control Middle Layer*, PAC 2003.

[5] J. Choi, *PyTracy*, unpublished.

[6] J. Bengtsson, *Tracy-III*, unpublished.

[7] *ChannelFinder: Directory Service for EPICS Channels*, `http://channelfinder.sourceforge.net/`

[8] G. Shen etc, *Virtual Accelerator at NSLS-II Project*, ICALEPCS 2013.

[9] G. Shen etc. *NSLS-II Middle-Layer Services*, ICALEPCS 2013.