BNL-221682-2021-TECH

NSLSII-ASD-TN-365

# Program on development of standard High Level Application (HLA) common tools for beam studies and operations

V. Smaluk

June 2021

Photon Sciences

**Brookhaven National Laboratory**

**U.S. Department of Energy**

USDOE Office of Science (SC), Basic Energy Sciences (BES) (SC-22)

| NSLS II TECHNICAL NOTE<br>BROOKHAVEN NATIONAL LABORATORY | NUMBER<br>NSLSII-ASD-TN-365 |
|---|---|
| AUTHOR<br>**V. Smaluk, T Shaftan, G. Wang** | DATE<br>**06/01/2021** |
| TITLE<br>**Program on development of standard High Level Application (HLA) common tools for beam studies and operations** | |

During NSLS-II commissioning, accelerator physicists created a powerful suite of high-level applications with capabilities sufficient to successfully put the machine into operation. A tight commissioning schedule made it difficult to debug HLAs and convert them into a common format with a uniform interface.

After the commissioning, NSLS-II entered user operations with machine ramping up the current regularly, and new insertion devices were installed almost every shutdown. To sustain our progress we needed highly efficient machine studies, which, in turn, required reliable and efficient software tools.

We were facing a number of problems with the existing high-level applications:

- There was no common standard for HLAs design. Applications were written in different environments and there was no uniform approach to their development.
- Most of the codes were developed by and for a single author. Programs often could not be used by anyone but their author, which lead to difficulties in arranging machine study shifts and working out corrections to the codes.
- The codes resided in personal workspaces and therefore were not accessible to the other team members.
- Most codes had no user manuals, which made their use nearly impossible for the team.

In 2017, we proposed the development of a Python Middle Layer (PyML) software package with the same or better functionality as Matlab Middle Layer (MML) used for accelerator controls at most of the light sources worldwide. Key features of the proposed PyML are summarized in the Table 1 in comparison with MML.

Table 1. PyML vs MML

| **MML** | **PyML** |
|---|---|
| Lattice input file:<br>Accelerator Toolbox format → THERING | Lattice input file:<br>ELEGANT format (lte) |
| Lattice model: matlab structures (ao, ad) | Lattice model: python structures, should be as convenient as the MML ones |
| Functions can be used from matlab command line, function, and script | Functions can be used from python command line, notebook, function, and script |
| >150 functions available for lattice modeling, characterization and correction | 40 functions available in the aphla package (2017) |

As a first step, we initiated the HLA common tools program to develop a set of well-tested and documented HLA functions available for all AD staff from a common tools collection. These functions were assumed to have interface to EPICS PVs and to the virtual machine model.

Basic requirements for the HLAs are formulated:

- Thorough equipment safety assessment must be carried out prior to the function development and testing.
- Every function must be available from console as a command line
  `function_name [-options] [input_file] [output_fie];`
- Fixed format of the function with following required fields:
  - Function header with parameters type and description;
  - PV list and description: machine system, access type (read/write) and frequency in the header (if too long, may be omitted, but must be present in documentation);
  - Comments.
- Universal functions: no machine-specific parameter is defined inside the function.
- No constant values or strings inside the function code.
- All cross-dependences must be clear.
- An interface to the machine lattice file, if applicable.
- Every function developed for a beam measurement must restore all used hardware (i.e. BPM settings, pinger status, RF frequency, etc.) back to its initial status after the measurement.

All functions must have clear and detailed specifications:

- Name of application;
- Description/Goal;
- List of input/output parameters;
- Related data files: such as input/output file/database and their format;
- Description of theory and data analysis/processing;
- Sequence of action/procedure, exit condition (if applicable);
- PV list: which PVs are read and written, how often (once per launch/every 60 machine cycles/every cycle etc.), to what machine system do they belong;
- Package, class, environment/external requirement;
- GUI (if any);
- Developer/maintainer;
- Error behavior, error codes or exceptions;
- Possible faults/damages of hardware, beam loss during operations;
- Version of developer software the function was tested with.

All accelerator physicists contribute to the code development, testing and debugging. Beam studies are allocated for testing the applications. The tested and debugged code with a manual is transferred to the central storage with documents containing the list of HLAs in use for NSLS-II operations and beam studies. All accelerator physicists should be trained on independent use of the HLA common tools.

The functions in the HLA library are logically divided into several blocks:

- Measurement and correction of orbit and linear optics.
- Measurement and optimization of non-linear beam dynamics.
- Compensation of perturbations caused by IDs.
- Optimization of injection.
- Beam stability and feedbacks.
- Beam diagnostic tools.

The HLA common tools program has been successfully completed in 2020. The list of ready-to-use applications is below:

- Injector energy tuning [7];
- Orbit correction with RF frequency adjustment [2];
- Local Orbit Bump program compatible with FOFB [10];
- Beam-based alignment [16];
- Measurement of chromaticity up to 5th order [15];
- Correction of betatron tunes [15];
- Correction of linear chromaticity [15];
- Local Bump Agent: automated python program for ID local bumps [11];
- Measurement of dynamic aperture and tune shift with amplitude [13];
- Noise locator (AC disturbances in beam orbit motion);
- Trajectory correction, energy and emittance measurement in LTB [17];
- Lattice characterization using gated BPMs [5];
- Vertical emittance control via linear coupling and vertical dispersion [14];
- Deviation of ID BPMs from the beam-based alignment;
- ID Orbit Feedforward Table generation/validation [12];
- Streak-camera interface;
- Measurement and correction of linear optics (beta functions and dispersion) [3];
- Measurement and correction of linear coupling [4];
- Measurement and correction of vertical dispersion [4];
- Fast Orbit Feedback interface [8];
- AC orbit bump [6];
- Active Interlock automatic testing [9];
- Standard Anakonda environments for studies and operations;
- Booster ramp manager;
- Local orbit feedback for canted/non-canted ID straights;

After thorough testing and debugging, all applications have been uploaded to a GitLab-based library [1]. Once each code is released, all NSLS-II accelerator physicists are trained on how to use the software. As a result, every physicist is now prepared for independent control and tune-up of the machine. A standard procedure of routine lattice characterization has been developed and implemented. We continue the program of routine lattice characterization using HLA common tools to keep tracking the machine status and to let every accelerator physicist get their skills updated. Now, the HLA common tools program is further evolved to the PyML. We are working on development, testing, and debugging Python functions using lattice models based on the input file in ELEGANT format with a built-in help and a detailed manual for every function.

## References


[1] https://gitlab.nsls2.bnl.gov/hlatools

[2] Y. Li, NSLS-II APHLA Orbit Correction Tools, BNL-221454-2021-TECH, 2021.

[3] Y. Li, NSLS-II APHLA Optics Correction Tools, BNL-221453-2021-TECH, 2021.

[4] Y. Li, NSLS-II APHLA Linear Coupling Correction Tools, BNL-221455-2021-TECH, 2021.

[5] Y. Li, NSLS-II APHLA Optics Correction with Gated BPM Tool, BNL-221456-2021-TECH, 2021.

[6] X. Yang, Implementing HLA common tool - AC-Local-Bump for impedance measurements and for FOFB compatible user operation mode, BNL-221563-2021-TECH, 2021.

[7] X. Yang, Implementing HLA common tool - Injector Energy Tuning Application, BNL-221564-2021-TECH, 2021.

[8] X. Yang, Implementing HLA common tool - FOFB Interface Application, BNL-221565-2021-TECH, 2021.

[9] R. Fliller, Active Interlock Automatic Testing Program, BNL-221567-2021-TECH, 2021.

[10] Y. Hidaka, Fast-Orbit-Feedback-Compatible ID Local Bumps with RF and X-ray BPMs at NSLS-II, BNL-221576-2021-TECH, 2021.

[11] Y. Hidaka, Local Bump Agents (Feedback) at NSLS-II, BNL-221577-2021-TECH, 2021.

[12] Y. Hidaka, A High-Level Application Tool for Generation and Validation of Insertion Device Orbit Feedforward Tables at NSLS-II, BNL-221578-2021-TECH, 2021.

[13] Y. Hidaka, A Multi-Shot Measurement Tool for 1-D Dynamic Aperture and Tune Shift with Amplitude based on Turn-by-Turn BPM Data at NSLS-II, BNL-221579-2021-TECH, 2021.

[14] Y. Hidaka, Vertical Emittance Control Application for NSLS-II Operation, BNL-221580-2021-TECH, 2021.

[15] G. Bassi, HLA Common Tools: Tune Correction, Chromaticity Measurement and Chromaticity Correction, BNL-221581-2021-TECH, 2021.

[16] J. Choi, Beam Based Alignment and Response Matrix Measurement, BNL-221582-2021-TECH, 2021.

[17] J. Choi, Injector High-Level Applications, BNL-221583-2021-TECH, 2021.