ORBIT AND DISPERSION TOOL AT FLASH

E. Prat, V. Balandin, N. Golubeva, DESY, Hamburg, Germany
J. Kamenik, I. Kriznar, T. Kusterle, Cosylab, Ljubljana, Slovenia

Abstract

A java-based application to measure and correct orbit and dispersion has been developed at FLASH. In this paper we discuss the algorithm used in this tool as well as its functionality. First tests on machine operation are also presented.

INTRODUCTION

FLASH, i.e., **F**ree electron **LAS**er in **H**amburg, is a Vacuum Ultra-Violet FEL user facility at DESY and a pilot facility for the European XFEL project [1]. A schematic layout is shown in Figure 1. FLASH uses DOOCS as control system [2].

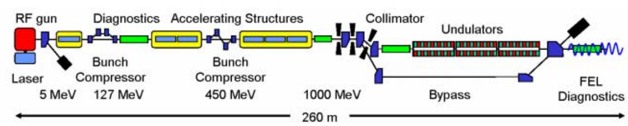

Figure 1: Schematic layout of FLASH.

Motivation

The beam trajectory has to be controlled for daily routine operation as well as for specific tasks at FLASH (i.e. for setting up the machine, to improve FEL radiation, etc.).

Moreover, for an optimal FEL performance the beam size in the undulator should not be increased by dispersive effects. The goal is to keep the contribution of the dispersion on the beam size below 10% for the whole electron energy range (450 MeV up to 1 GeV). The (dominant) energy spread generated during bunch compression in BC3 is about 0.3 MeV rms. Therefore, the dispersion generated between BC3 and the undulator must be corrected to less than one centimetre in both planes. Another important aspect of dispersion correction is the reduction of undulator orbit launch sensitivity to rf amplitude and phase jitter.

Sources of the (spurious) dispersion include field errors and stray magnet fields in the undulator beam line as well as spurious dispersion created upstream of the undulator by, for instance, rf coupler kicks, magnet misalignments and field errors. The impact of these errors on dispersion generation depends on the actual operating conditions of the accelerator, so the dispersion must be measured and controlled frequently.

Dispersion Measurement

Dispersion measurement is based on measuring the orbit for different energies of the beam, obtained by changing the gradient of the different accelerator modules in the machine.

To compensate the kick produced by the module when its gradient is varied an orbit correction is performed just downstream the accelerator module. Therefore, the orbit readings downstream the last correction BPM can be used to derive the horizontal and vertical dispersion (R_{16} and R_{36}) from that BPM up to any downstream position *s*. In our case, a second order fit is applied:

$$x(s) = x_0(s) + R_{16}(s_0,s)\frac{\Delta p}{p} + R_{166}(s_0,s)\left(\frac{\Delta p}{p}\right)^2$$

$$y(s) = y_0(s) + R_{36}(s_0,s)\frac{\Delta p}{p} + R_{366}(s_0,s)\left(\frac{\Delta p}{p}\right)^2$$

Figure 2 displays an example of a dispersion measurement for a single BPM.

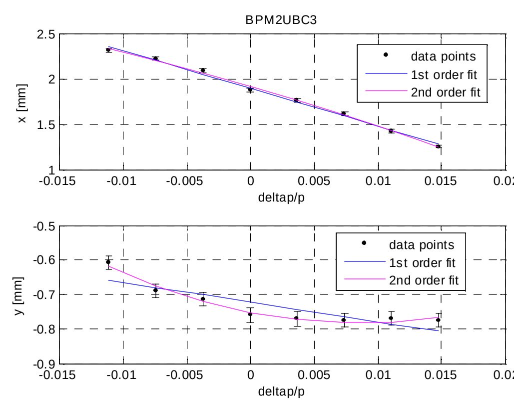

Figure 2: Dispersion measurement for a single BPM.

Correction Algorithm

We correct both orbit and dispersion using correction coils and quadrupole movers in the undulator and collimator sections. The optimal settings are calculated using the orbit and dispersion response matrices, which are defined as the shift of the orbit or dispersion due to a change of the corrector strength:

$$O_{i,j} = \Delta x_i / \Delta\theta_j \qquad D_{i,j} = \Delta d_i / \Delta\theta_j$$

where Δx_i and Δd_i is the change of the orbit and dispersion in the BPM *i*, and $\Delta\theta j$ is the change of the strength of the corrector *j*.

The algorithm consists in finding a setting of correctors $\underline{\Delta\theta}$ which induces some orbit and dispersion that minimizes in a least squares sense the finally resulting orbit and dispersion:

$$(1-w^2)\left\|\underline{x}_{meas} + \underline{\underline{O}}\underline{\Delta\theta}\right\|^2 + w^2\left\|\underline{d}_{meas} + \underline{\underline{D}}\underline{\Delta\theta}\right\|^2 = \min$$

More details about the algorithm procedure and successful correction results can be found at [3].

ORBIT AND DISPERSION CORRECTION APPLICATION

The Orbit and Dispersion Correction Application (ODCA) is a tool for measuring and correcting orbit and dispersion of FLASH. It calculates required corrections for steerers and quadrupole movers and it applies them in order to adjust orbit and dispersion towards a reference.

The ODCA is meant to be employed on a regular (daily) basis by the operators in the FLASH control room. This imposes a number of non-functional requirements on the software:

- **Robustness**: failures in I/O or communications channels should be handled without blocking or even crashing the application. Out of bounds values of inputs or responses should be handled properly by the application and presented in its GUI.
- **Responsiveness**: all the CPU intensive or lengthy operations including measurement or control calls to the control system should be handled asynchronously or in a separate thread to do not unnecessarily block the work flow – especially in time critical situations.
- **Flexibility**: since it is a general purpose tool, the GUI and the visualization of the data should be general and it should allow a certain degree of customization. Well defined additional control and data manipulation procedures should be easy to add at later stages during the application's life cycle.
- **Production Quality**: since the target users are not necessarily familiar with the algorithmic, structure and implementation details of the application, the GUI should provide intuitive layout and descriptive labelling of the individual controls as well as clear feedback messages on the user inputs or other events, when appropriate.

Application Architecture

We have selected the different elements of the application architecture in order to fulfil all the functional and non-functional requirements.

We have chosen the Java programming language since it is in principle platform independent. This leads to reduced implementation, deployment and maintenance effort. It also has support for native multi threading supplemented with useful concurrency utility libraries. We use the Swing GUI framework since it allows building rich and responsive GUIs on many of the today common OS platforms.

To ensure flexibility of the application design we followed an extended version of the Model-Controller-Display design pattern for the application, together with a modular design of the computational part of the model. The more detailed structure of the design is shown in Figure 3. All the user actions on the GUI display are handled through the controller which passes the commands asynchronously to the engine.

The engine is handling all processing logic: communication with DOOCS and optics server, execution of the modules, keeping a reference of the state of the execution progress, etc.

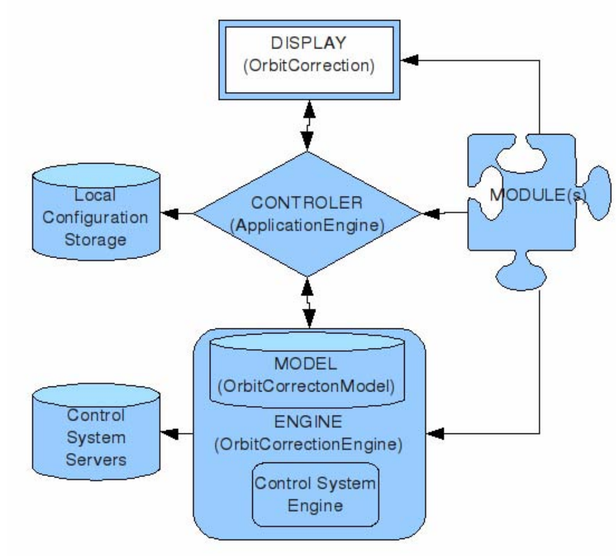

Figure 3: Application Design Scheme.

One of the strong features of the engine is the so-called "simulation mode". With a single boolean flag in the configuration, all the calls to DOOCS are diverted to a local cache (backed on HashMap). This enables testing of the whole applications and the major part of the computational modules in off-line, with limited access to the FLASH machine.

The engine also remembers the number of the last operations made, being able to perform "redo" and "undo" operations.

Specific tasks are implemented as modules and extension points of the application. They have a well-defined and well-controlled life-cycle, so they can be interrupted at any point. They have optionally their own GUI. In this way they can provide implementations of various specific cases for the application. The modules to be loaded by application are specified in the local application configuration. The following modules are finished or being tested: orbit correction module, orbit and dispersion correction module, dispersion measurement module and self closing localized bump module.

Optics Server

In order to be independent from hardware upgrades, the ODCA relies on the optics server. The server provides not only response matrices required for correction algorithms but also the most recent information on currently available number and locations of corrector magnets and beam position monitors. The optics server in its turn is always compiled form the last version of the online Matlab optics toolbox [4], which is supposed to contain at the current time the most accurate optics model and transfer coefficients between power supply currents and steerer deflection angles.

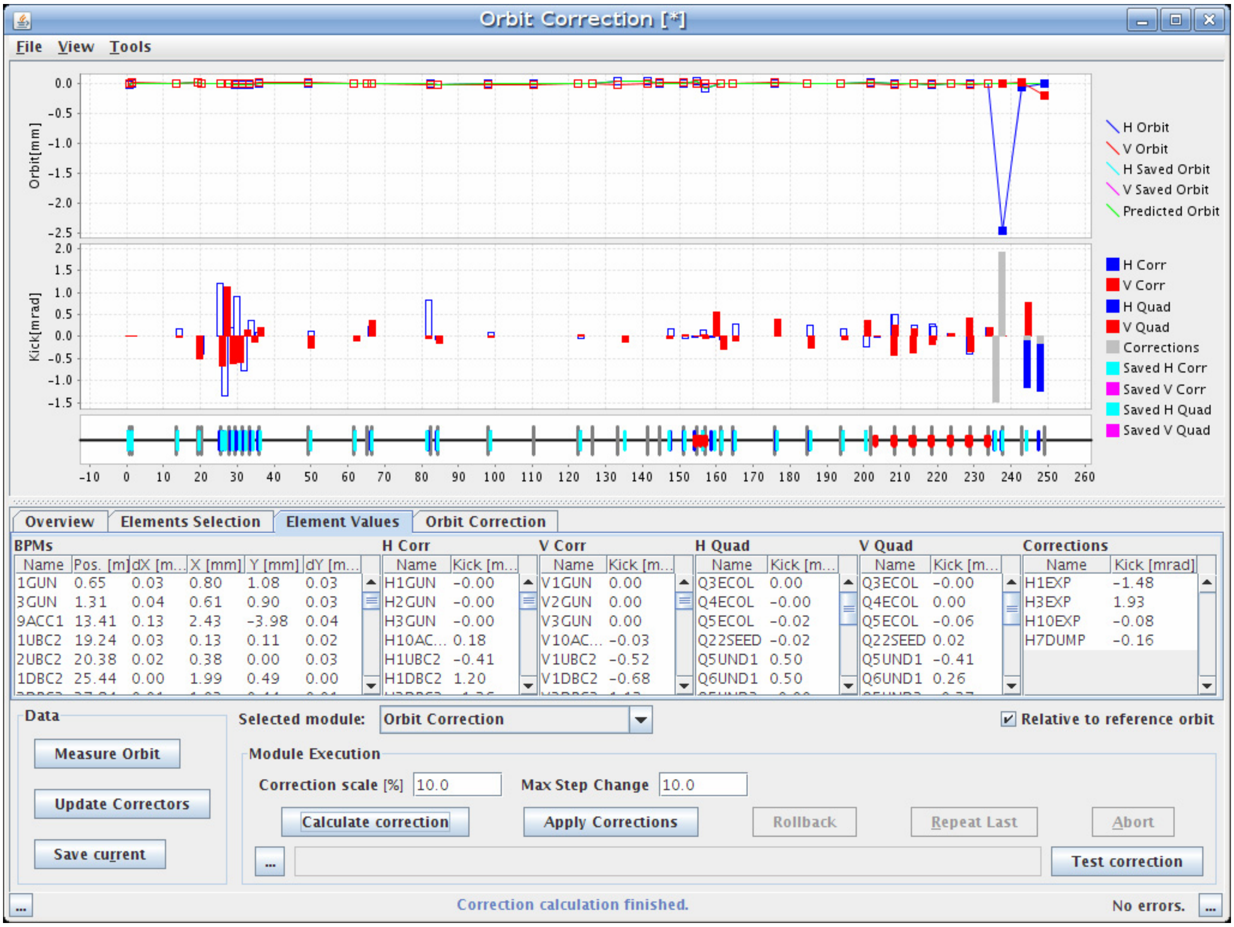

Figure 4: GUI for Orbit and Dispersion Correction Application. A horizontal orbit of 2.5mm was applied at BPM3EXP (upper plot). Corrector changes are indicated in grey (lower plot).

GUI

Figure 4 shows the GUI for the present version of the ODCA. The main application area contains the chart region on its upper half. The data for the different orbits is visualized in the upper plot. The lower plot shows the corrector values and the different changes applied during corrections. All the displayed elements can be selected or deselected for the correction by clicking on the charts or on the synoptics chart placed below the plot area.

Below the charts, all the plotted data is also displayed in tabbed tables. The GUI control part supplied optionally by the modules is also displayed here. The main application controls are displayed below the tables.

Beside the main area, the application contains a menu bar where the other application functions can be found. This includes local saving and loading of orbit data, customization of GUI display, configuration of other non-visual application parameters, etc. Finally, at the bottom of the GUI there is a status bar where the local application log and error histories can be accessed.

FIRST TESTS AND OUTLOOK

We have successfully tested a preliminary version of the ODCA. Figure 4 shows an example of orbit correction. A local horizontal bump of -2.5mm was set at 3EXP, a BPM placed downstream the undulator section. For that we used four corrector magnets – the applied changes are indicated in grey

We are presently implementing the option to measure the dispersion and improving some aspects like the bump set-up. We will have beam time to test the final version of the application this summer. We expect that ODCA will be available for FLASH operators by the end of 2009.

REFERENCES

[1] J. Rossbach, “A VUV free electron laser at the TESLA test facility at DESY”, Nuclear Instruments and Methods in Physics Research Section A, Volume 375, p. 269-273 (1996).

[2] http://doocs.desy.de

[3] E. Prat *et al*, “Measurement and Correction of Dispersion in the VUV-FEL”, EPAC’06, Edinburgh, June 2006, p. 1951.

[4] V.Balandin, N.Golubeva,"Online Matlab Toolbox for the FLASH Optics", unpublished user manual, 2006-2008.