AN OPTICS-SUITE AND -SERVER FOR THE EUROPEAN XFEL

S. Meykopff, DESY, Hamburg, Germany

Abstract

For the European XFEL and the upgraded FLASH facility we require a tool for beam optics calculations. A newly developed software library manages accelerator parameters and performs beam dynamics calculations. In addition a server offers an interface between the library and the control system. A MATLAB interface allows convenient access to the optics server. This framework provides an online model which is integrated in the control system. The online model is used for a simulated European XFEL environment with realistic control interfaces. We use this environment for extensive software developments and tests.

At the FLASH facility a MATLAB toolbox is in use for the calculation of optics parameters [1, 2]. The FLASH2 update adds a second undulator beam line to the facility. The second beam line is not supported in the MATLAB toolbox and must be considered in a successor. Currently the European XFEL is in construction which has up to 5 beam lines [3]. A beam optics tool will be required for commissioning and routine operation. We decided to develop a code for both facilities. This code will provide the accelerator layout in conjunction with the different beam lines. The code calculates transport and response matrices and allows fitting of optics parameters. The matching of beam parameters will be an important feature of the new code. To provide clearly arranged software the new code is developed as a library.

OPTICS LIBRARY

The description of the accelerator is stored in the optics library and will be initialised from a SQlite database [4]. In the case of the European XFEL the SQlite database is generated from an Excel sheet which is distributed by the machine layout coordinators [5]. The sources of the FLASH description are some MAD8 output files [6]. In both cases the generation of the database will be done with MATLAB. The optics library provides these descriptions as static data. The static data includes element names, type names, positions, and covers all columns of the Excel sheet.

The design of the optics library provides multiple setups in the same time. These setups cover variable parameters like steerer magnet angles or k values of quadrupole magnets. The setups are kept separated with a unique setup name. Each setup is independent.

An external code is called for the beam dynamics calculation. Currently the code ELEGANT is in use [7]. To start an ELEGANT run a lattice file and a command is required. The static information and the current setup information are used to generate both files. The output format of elegant is SDDS. The SDDS files were read from the library and stored internally. A dispatch queue collects all ELEGANT runs and executes them parallel. The code of the optics library is written in C. The design of the interface allows one to use it as a shared library.

OPTICS SERVER

The optics server offers a connection to the control system. The TINE interface was chosen as control system because it's in use at both facilities. The optics server provides all functions of the optics library. Every optics server call includes the setup name. This constraint allows a multi-user run with independent setup parameters.

The layer between the control system and the optics library is thin (see figure 1 for details). Mostly parameter checks and conversion is implemented in this layer. A major addition is the observe module. The observe module checks for updates of the RF system or changes of power supply currents. On demand a defined setup will be updated with these values.

We provide a virtual beam position monitor server for tests with simulated orbits. The orbit will be set by a call from the control system. In the optics server is a small push orbit module. This module delivers the simulated orbit to the virtual beam position monitor.

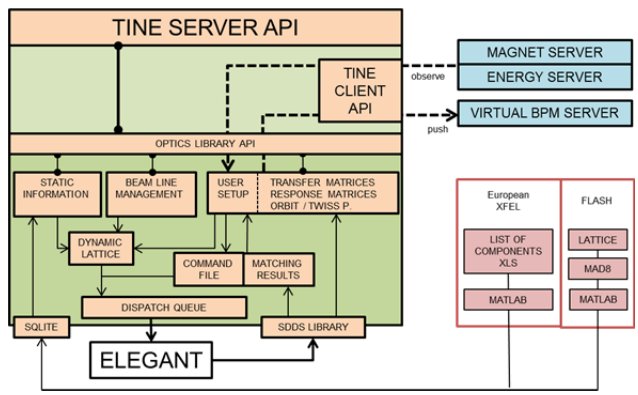

Figure 1: Optics server internals: TINE interface to clients and connections to other servers. Optics library internals with information databases, file generator, job dispatcher, access to SDDS output, and SQLite database as source.

INTERFACE TO MATLAB

In the DESY control room MATLAB is an important tool. The XOptics MATLAB object is a convenient interface to the optics server. The lower interface will be provided by XCOMM. XCOMM is a unified interface for TINE and DOOCS control systems (overview in figure 3) [8]. All functions of the optics server are covered by the XOptics object. MATLAB offers a tabulator expansion for object methods and properties. This feature is a notable simplification for the user of XOptics. Figure 2 shows a code example.

ISBN 978-3-95450-146-5

```
% get XOptics object:
x = XOptics();
% get all members of beamline I1D:
i1d_member =
     x.ListOfBeamlineMembers('I1D');
% fetch all informations:
info = x.ElementSetupInformation(
     'DEFAULT','I1D', i1d_member);
% plot beta x:
plot([info.z],[ info.betax]);
```

Figure 2: How to plot optics data in MATLAB.

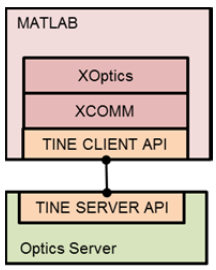

Figure 3: MATLAB interface to the optics server.

USE CASE: VIRTUAL XFEL

At DESY a large environment was built to develop and test software for the European XFEL. This environment includes elements of low-level, middle-layer and high-level controls software, and some hardware. It covers the timing system, the toroid server, the data acquisition server, the orbit server, the sequencer, the magnet server and the power supply servers. The test environment includes also software which runs in the control room like the display panels, the orbit bump tool and the emittance measurement and matching tool (see figure 4). Most of the software components are used with little or no modifications with respect to the production versions. Our environment allows us to test software components before the commissioning of the European XFEL.

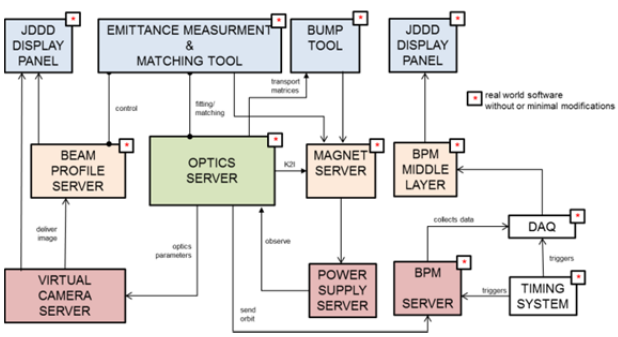

Figure 4: Components of the Virtual XFEL. High level applications on top, middle layer server and on the bottom front end server.

Test of Orbit Bump Tool

One test case is the orbit bump tool. The tool queries transport matrices from the optics server. The calculated bump angles are written to the magnet server. The magnet server uses the optics server to convert the angles into currents. The magnet server delivers the new currents to the power supply server. The new currents are read by observe module of the optics server. The optics server recalculates the orbit and pushes the new orbit data into the virtual beam position monitor server. The DAQ collects the new positons with the help of a trigger signal of the timing system [9]. The collected data will be read from the beam positon monitor middle layer server. The standard tool JDDD displays the new beam positions, showing a closed bump if everything tested successfully [10].

USE CASE: EMITTANCE MEASUREMENT AND MATCHING TOOL

At the FLASH facility we developed a beam profile server. This middle layer server offers emittance measurements. The server handles all different interfaces of the cameras and wire-scanners. If required the server switches the laser on or off. There are routines to handle the background noise and to calculate some beam parameters. The "Emittance Measurement and Matching Tool" was developed to operate the middle layer server. The tool was developed in MATLAB and allows starting a measurement with live view of the cameras. The tool plots the measurement results and displays some beam parameters like the emittance and BMAG (see figure 5). The fitting of the beam parameters are done by the optics server. The tool offers the matching of the twiss parameters. The matching will be done by the optics server. The tool displays the advised magnet currents and pushes the values to the power supplies one demand. A new measurement with the updated magnet currents should yield optimized beam parameters.

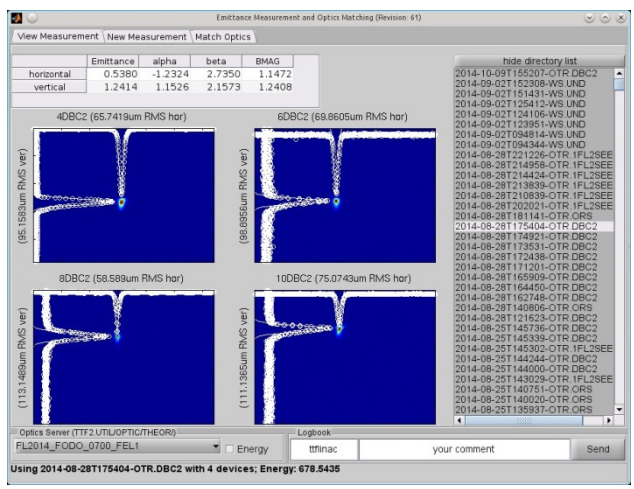

Figure 5: Screenshot of the "Emittance Measurement and Optics Matching Tool".

ISBN 978-3-95450-146-5

CONCLUSION

The optics library is an easy to use and powerful interface to do beam optics calculations. The optics server offers all library functions in the control system. The full-featured interface in MATLAB is a valuable tool in the control room allows the development of nice applications. The integration in the control system enables the installation of an unprecedented development and test environment. The developers are independent of a real accelerator. The time schedule of the commissioning of the European XFEL will be tight. Currently we are able to develop and test software before the commissioning. We expect a smoother commissioning phase with our already tested software.

REFERENCES

[1] Free-electron laser FLASH http://flash.desy.de/.
[2] V. Balandin, N. Golubeva, Matlab, “Functions for Calculations of the Linear Beam Optics of FLASH Linac”, http://ttfinfo.desy.de/TTFelog/data/doc/Physics/Optics/2008-02-15T13:59:48-00.pdf
[3] European X-Ray Free-Electron Laser Facility GmbH, http://www.xfel.eu/.
[4] SQLite, http://www.sqlite.org/.
[5] European XFEL – list of components, http://www.desy.de/fel-beam/
[6] MAD8, http://cern.ch/mad8
[7] M. Borland, "elegant: A Flexible SDDS-Compliant Code for Accelerator Simulation", Advanced Photon Source LS-287, September 2000.
[8] J. Wilgen, S. Meykopff, "XCOMM A Unified MATLAB API for TINE and DOOCS", Proc. PCaPAC2014, http://jacow.org/.
[9] V. Rybnikov et al., “FLASH DAQ DATA MANAGEMENT AND ACCESS TOOLS”, Proc. PCaPAC2010, http://jacow.org/.
[10] JDDD, A Java DOOCS Data Display, http://jddd.desy.de/.

ISBN 978-3-95450-146-5