

AN ORBIT FEEDBACK FOR THE FREE ELECTRON LASER IN HAMBURG (FLASH)

R. Kammering DESY (Hamburg, Germany), John Carwardine ANL (Argonne, IL, USA)

Abstract

The lack of knowledge of the exact energy profile of the Free Electron Laser in Hamburg (FLASH) and thereby of the orbit response matrix, made the implementation of a conventional orbit feedback in the past very difficult.

The new run period started this spring after extensive modifications of the facility, showed that the responses matrixes seem now to be in good agreement with the theory, thereby allowing the application of standard orbit feedback techniques.

The physics concepts and the chosen architecture to implement such software on the middle layer and interplay with other high-level software components will be discussed. The development and implementation of this software using the DOOCS servers in combination with the dynamic components of the Java DOOCS data display (jddd) allowed a flexible and scalable implementation, which could also serve as a prototype for future implementations at e.g. the European XFEL.

MOTIVATION

The task of stabilizing beam jitter, as it is the case at most synchrotron radiation facilities, is not feasible for the FLASH linac, because the orbit can only be sampled at the maximum of pulse repetition rate of 10 Hz.

So the task of compensating fast-varying errors, for example in magnetic fields of corrector magnets or vibrations due to ground movements is here not the main focus of this orbit feedback implementation.

Instead of this the main objectives for an orbit feedback at a linear accelerator are to:

- restore saved orbits
- compensate long-term drifts
- stabilize the orbit downstream while tuning the machine further upstream
- making localized orbit changes

These are only the most important objectives an orbit feedback could attack. For FLASH it is even further envisioned to change the today practice of using individual steerers (dipole magnets) to tweak the orbit at a certain position along the machine (we will call this the *longitudinal position* in what follows), but instead of this modify beam positions using the orbit feedbacks target values at this longitudinal position.

BASIC SCHEMA OF A BEAM BASED ORBIT FEEDBACK

The basic principle of the FLASH beam based orbit feedback follows the standard techniques as e.g. described in [1]. A linear response matrix (\mathbf{R}) describes the action of small changes ($\Delta\mathbf{I} = [\Delta\mathbf{h}, \Delta\mathbf{v}]$) in the corrector magnet

fields (dipoles) on the beam position ($\Delta\mathbf{X} = [\Delta x, \Delta y]$) measured at the beam position monitors (BPMs).

$$\mathbf{R} \Delta\mathbf{I} = \Delta\mathbf{X}$$

Inverting the response matrix allows to derive the needed values to be applied to the correctors to yield a certain change in the beam position. In cases of unequal numbers of BPMs and correctors, the response matrix is non-square which can be inverted using the pseudo inverse or singular value decomposition.

$$\mathbf{I}_j = g \mathbf{R}^{-1} (\mathbf{X}_{\text{ref}} - \mathbf{X}_{\text{meas}}) + \mathbf{I}_{j-1}$$

With the gain factor $g = 1$ this would lead to a full correction of a given difference between the desired \mathbf{X}_{ref} and actual beam position \mathbf{X}_{meas} , if the new current \mathbf{I}_j will be written to the correctors in step j . One will usually work with a gain factor $\ll 1$ and also apply some filtering to the \mathbf{X}_{meas} data to avoid ringing and overcorrection.

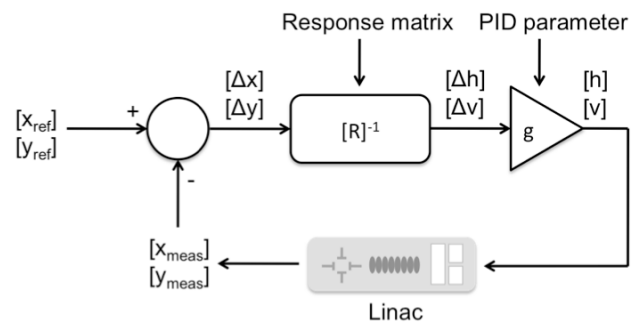


Figure 1: Basic structure of the beam based orbit feedback loop

ARCHITECTURE

The main objectives of the FLASH orbit feedback are not to damp high frequency position jitter, but more to assist operation and decouple actions within the machine.

Therefore it was planned from the beginning to implement this as a pure software feedback with moderate operation frequency (0.5-2 Hz).

The basic architecture for such a software-based feedback therefore follows the classical design of a middle layer server as described in the following section.

DOOCS as basic software infrastructure

The dominant control system at the FLASH facility is the Distributed Object Oriented Control System (DOOCS) [2]. Therefore a logical choice for the orbit feedback is to implement this software using C++ and the existing DOOCS application-programming interface (API). DOOCS offers a natural mapping of the monitors (BPMs) and correctors (steerers) to C++ objects, which significantly eases working with many devices, and thus understandability of the code.

Optics Toolbox

In standard operation the machine optics and hence the beam transfer matrix will not change, provided the energy and quadrupole magnetic fields variations are negligibly small. Therefore we excluded the determination and manipulation of the transfer matrix from the core feedback function and instead used the well-proven optics functions from the *optics toolbox* used at FLASH [3]. This toolbox is a collection of Matlab functions providing all relevant beam optics operations needed for standard optic tasks. This toolbox is used to create inverse response matrixes and stores these in files, which are read by the orbit feedback (see Figure 2).

The orbit server

The orbit feedback server is not reading the actual beam position from the front end servers attached to the beam position monitors (BPMs), but rather it is read from a server instance (called *orbit server*) used for synchronizing and pre-processing (e.g. the intra bunch train average) as shown in Figure 2.

The orbit server itself is embedded in the FLASH data acquisition system (DAQ), from where it collects the BPM readings (for details about the DAQ system see e.g. [4]).

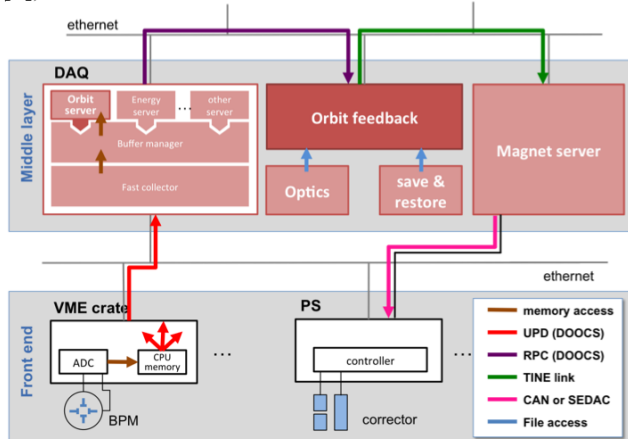


Figure 2: Architecture and data flow

The orbit feedback

The orbit feedback uses standard DOOCS RPC communication to collect BPM data and perform the calculations needed to create the corresponding vector of corrector setpoint changes. The updated corrector setpoints are written to the TINE-based magnet server, which distributes them to the relevant power supply (PS) controllers (for TINE see [5]).

The management of the reference orbits (vectors of setpoint values) will be handled through the existing FLASH Save and Restore system.

The display level

Java DOOCS data display jddd [6] is used for monitoring and control of internal states of the orbit feedback server.

Defining BPMs and steerers as DOOCS objects and subsequent mapping to jddd's *dynamic lists* makes it possible to work with many devices as if they were a single instance. (FLASH, even though its length is only about 300m, contains ~ 50 BPMs and ~ 70 correctors.) This is realized by the simple technique of “draw once, use many times”.

Figure 3 shows the jddd editor with the line representing a single BPM instance while in the lower right corner the same panel in run mode is showing the full list of BPMs.

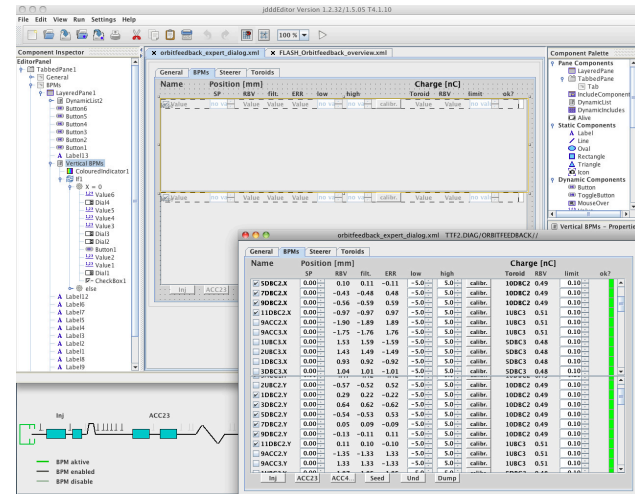


Figure 3: Java DOOCS data display, showing the BPM pane in edit and run mode

STATUS

The basic software interface for reading the beam positions and writing the corrected currents has been implemented using C++ as a standard DOOCS server. Rough estimates for total loop times (150-300 ms) have been made and show that operation with the targeted operation frequencies of 0.5–2 Hz are well suitable.

First routines for accessing response matrixes delivered by the optics toolbox have been integrated and the whole data flow chain is already operational.

All implementations have been accompanied by the continuous development of jddd panels mainly aimed for debugging, but these will also serve as a good starting point for final monitor and control panels.

First tests with the beam are planned to take place in the end of this year.

CONCLUSIONS

The implementation of a software based global orbit feedback for the FLASH facility using the existing software landscape is well on its way. The loose coupling and combination of the different control system components and protocols has allowed this new software to be developed without any need for modification of existing parts.

The combination of DOOCS object oriented approach and the dynamic generation of the displays and panels,

have proven to ease the development a lot. Such methods will be a must for working with the high device multiplicities, as one will have at e.g. the European XFEL.

REFERENCES

- [1] Herman Winick, "Synchrotron Radiation Sources", World Scientific Publishing, Singapore, 1994
- [2] K. Rehlich et al, "DOOCS: an Object Oriented Control System as the integrating part for the TTF Linac", Proceedings ICALEPCS '97, Beijing China, 1997
- [3] V. Balandin and N. Golubeva, "Current Status of the Online MatLab Toolbox for the FLASH Optics", XFEL Beam Dynamics Group Meeting, October 2007.
- [4] K. Rehlich et al, "Integrating a Fast Data Acquisition System into the DOOCS Control System", ICALEPCS'05, Geneva Switzerland, October 2005
- [5] P. Bartkiewicz, P. Duval, "TINE as an accelerator control system at DESY", Meas. Sci. Technol. 18 (2007)
- [6] E. Sombrowski, A. Petrosyan, K. Rehlich, P. Tege, "jddd: A Java Doocs Data Display for the XFEL", ICALEPCS'07, Knoxville, Tennessee USA, October 2007