

APPLICATION PROGRAMS FOR THE ELETTRA BOOSTER COMMISSIONING AND OPERATION

F. Iazzourene^{*}, C. Scafuri, V. Forchi[#], Sincrotrone Trieste, Trieste, Italy

Abstract

The application programs developed for the commissioning and operation of the new Elettra injector are all based on the TANGO control system, a new High Level Framework and a beam optics module named Vicky. The present paper summarizes the main developed application programs and their successful use during commissioning and operation of the new injector.

INTRODUCTION

Elettra is a 2.5GeV 3rd generation light source in operation since October 1993. For the commissioning and operation of the new Elettra injector [1, 2], complete new application programs based on the new high level framework and the new beam optics module named Vicky have been developed [3].

The programs are mainly: boosterTOCA, TOCA being an application program developed for the storage ring in use since 1998 [4], BoosterBumpers, PTBTrajectory, BTSTrajectory, PTBEnergy, BTSEnergy, and BoosterEnergy. PTB and BTS are the transfer lines Pre-injector To Booster and Booster To Storage ring, respectively.

HIGH LEVEL FRAMEWORK

Library

The High Level Framework (Hlf) is a collection of software libraries designed for the implementation of accelerator models. The design of the framework is fully object-oriented, and implemented in C++. The framework gives the programmer all the essential building blocks for assembling a software model of an actual accelerator. At the present state of development the library, mainly, deals with transverse beam dynamics.

The library is structured into 3 main modules.

- The first module, named *Field*, provides the basic model data of an accelerator magnet, like dipoles, quadrupoles, etc. The data are both static: like its geometry, its longitudinal position, its survey errors, the name of its power supply, its calibration tables, etc, and variable: like magnetic field, strength, etc. The conversion from excitation (current, voltage, etc...) to field is done by means of cubic spline interpolation. Although the library user can build its own magnet models "by hand", a utility class, designed according to the *factory* pattern is provided. This factory class completely automates the loading of the static data into the magnet models. Static data are stored in a MySQL database.
- The second module, named *Optics*, provides the data structures needed by the optics' calculation and

application programs. The two main classes are *AcceleratorElement* and *Lattice*. An *AcceleratorElement* is composed of 3 objects: a *Controller*, a *Field* and the data needed by application programs like Twiss functions, transfer matrices, etc, calculated by *Vicky*. A *Controller* handles all the communications with the real plant by means of the control system. The *Lattice* is an ordered container of *AcceleratorElement* objects. The *Lattice* provides also the *MagnetFamily* class, which is the string of magnets connected to the same power supply, which thus share the same excitation. This information is exploited by the optics programs to determine the effective number of independent parameters which can be used to manipulate the beam parameters.

- The third module, named *Latticedescription*, provides the *Latticefactory* class, designed according to the *factory* pattern. The *Latticefactory* class builds the lattice of the accelerator or transfer line, etc. The lattice is populated with all its *AcceleratorElement* and *Controller* objects correctly configured. The description of the lattice is stored in a MySQL database. The standard *Latticefactory* class manages all the connection and query procedures to the database.

Integration in the TANGO Control System

The Hlf library *Controller* is linked to the effective plant devices - mostly power supplies - by means of the Tango Control System [5] libraries. Thus a *Controller* object can both read and set the excitation of a string of magnets. *Controller* objects, although accessible by the library user, are seldom used directly. For example, when an application program issues a *set strength* command, the *AcceleratorElement* converts the strength to the excitation and then sets the power supply, via the *Controller*. All the necessary conversions are done transparently.

It is possible to use *Controller* objects also in simulated mode, without actually accessing the real power supplies. Using simulated mode it is possible to test application programs off-line or perform simulations. This feature has been quite useful for the debugging of the application programs before their use on the real field.

VICKY

Vicky is a package which provides all the classes and tools needed by application programs, like the computation of the optics and beam parameters like the equilibrium emittance, the measurement and correction of the orbit, tune, chromaticity and trajectory. The optics'

* fatma.iazzourene@elettra.trieste.it

Now at European Southern Observatory, Munich, Germany

calculation provides the “real” case, in which are taken into account the imperfections, like misalignments, magnetic errors, etc; and the “ideal” case. The package is written in C++ and uses itpp mathematical libraries [6]. Vicky has been successfully used for the commissioning and operation of the booster and its transfer lines.

APPLICATION PROGRAMS

Most of the needed tools for the commissioning of the new injector were available from the start. Some made use of the MATLAB package, like the linac emittance measurement and the integer part of the booster tunes. Others were written in Python, like the filling pattern setup of the storage ring and the one shot injection and ramping of the booster. After debugging and tuning using Python, the definitive programs have been written in C++.

The save and restore of the machine settings/readings is performed by a dedicated application program, based on TANGO tools. For details see [7].

The remaining application programs were directly written in C++ and use the Qt widget libraries for the graphic interface.

Transfer Lines and Booster Energy Scaling

The application programs, like BoosterEnergy, PTBEnergy and BTSEnergy developed to scale the energy to a wished value, use the calibration tables through the *Field* module of the Hlf package. The quadrupoles, sextupoles and correctors are scaled accordingly, keeping constant their strengths. PTB, BTS and Booster energy programs have actually the same source. For the operator convenience, they have a different name. Fig. 1 shows the interface panel for PTBEnergy, for example.

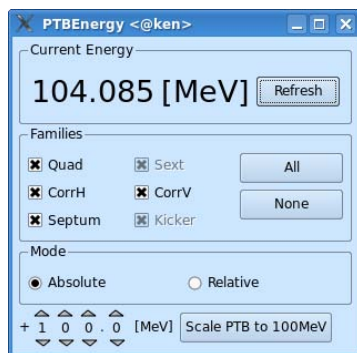


Figure 1: PTBEnergy panel.

Applications Programs Which Use Also Vicky

The applications which also require Vicky are BoosterBumpers, BoostertOCA, PTBTrajectory and BTSTrajectory. BoosterBumpers has been successfully used for the calculation of the currents to set to the bumpers for the extraction of the beam. It has not yet a graphic interface.

They all have a text window to printout the execution tasks results and so on.

On top of their main utilities, like the measurement and correction of the orbit or trajectory, for example, these application programs provide a “tools” menu: like the print out on the text window of the optics, the beam parameters, e.g. equilibrium emittance, damping times, etc for the booster, the plot of the optics, as shown in fig. 2, the plot of the tune diagram for the booster, and the scaling of the correctors, by a scaling factor.

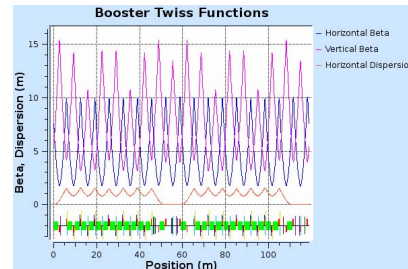


Figure 2: Booster Twiss functions.

They have also a “settings” menu. The 1st menu item allows the operator, through a popped up panel, as shown in fig. 3 for the PTB transfer line, to select the BPMs and correctors to use, the variation value of the current for each corrector for the response matrix measurement, the initial Twiss function values for the transfer lines, etc.

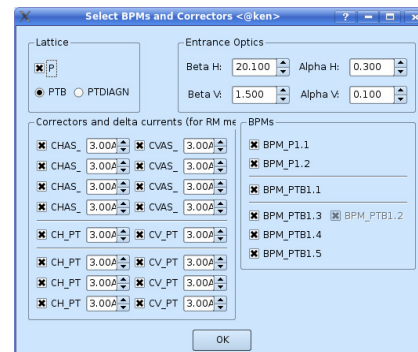


Figure 3: BPMs and correctors selection panel.

The 2nd menu item allows the operator to choose the energy of the beam to use for the optics calculation, from the magnets calibration data or from an assumed value.

For the booster, there is a 3rd menu item where the operator can use for the optics calculation, the DC values of the power supplies, selecting “static” or the values at a given time of the ramp, selecting “dynamic” and digitizing the wished time.

All these programs have a “read only”, “read and calculate correction”, “apply correction” and “undo apply”. For the correction of the orbit and the trajectory, the application programs provide two methods: Micado and SVD, together with the choice from the panel of the number of correctors or Eigen values, respectively.

The PTB and BTS trajectory programs have actually the same source. They are given a different name for the operator convenience. Fig. 4 shows, for example, the graphic interface panel for BTSTrajectory.

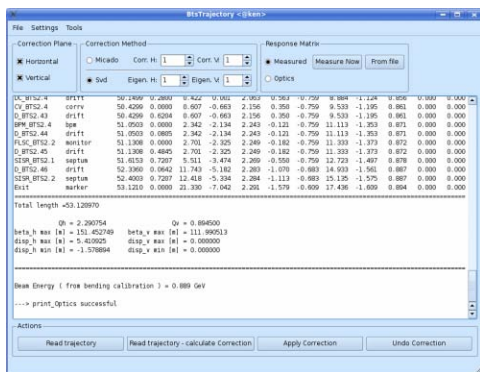


Figure 4: BTSTrajectory graphic interface panel.

The response matrix for the trajectory correction calculation can be measured or computed from the Twiss functions. The measured response matrix can be saved in a file and used later, selecting “read from file” and then the file. The trajectories of the transfer lines and the orbits of the booster are plotted after each “read” and “read and calculate correction”. Both Micado and SVD methods have been successfully used for both PTB (using measured response matrices) and booster. Figures 5 and 6 show, for example, the orbits of the booster before/predicted and after correction, respectively, when using SVD method for the correction of the horizontal orbit.

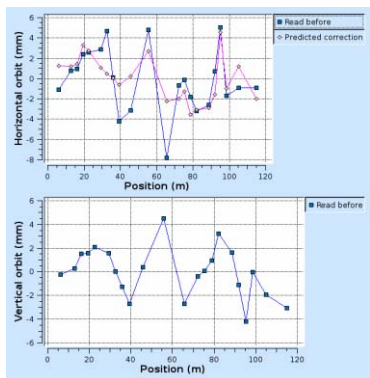


Figure 5: Booster orbits: before & predicted.

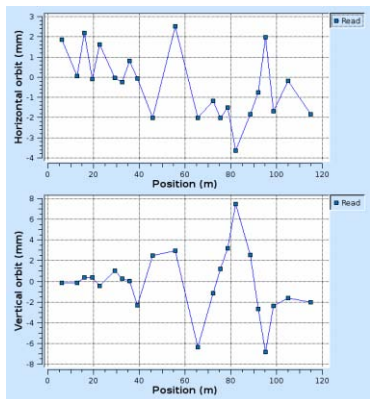


Figure 6: Orbits after correction (H. plane).

BoosterTOCA provides also the measurement and the correction of the tunes and chromaticities. Up to now, the tune measurement is not available via the control system, thus, the program provides to the operator a widget in which he can introduce the measured values, as shown in the graphic interface panel, fig. 7.

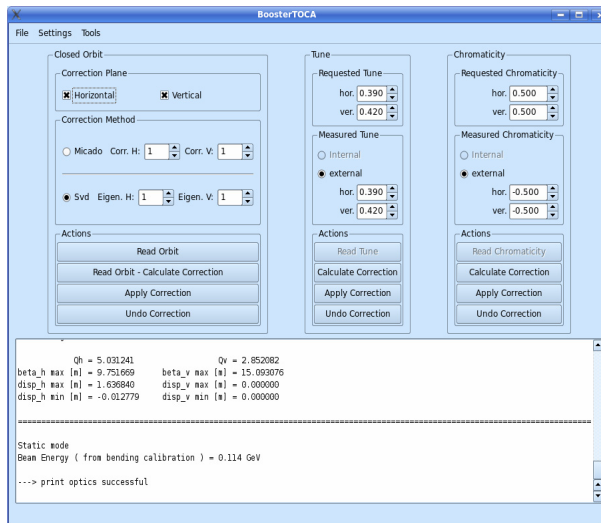


Figure 7: BoosterTOCA graphic interface panel.

CONCLUSION

The developed application programs have been successfully used during the commissioning and after. The needed improvements and further tools installation are in progress. A BoosterBump application program is foreseen to create bumps, both in horizontal and vertical plane. Once the Hlf of the storage ring will be available, Vicky and BoosterTOCA can be used for the storage ring, too. Vicky will also replace Racetrack [8]. The development of the relative tools is in progress.

REFERENCES

- [1] M. Svandrlik et al., “Overview of the Status of the Elettra Booster Project”, these proceedings
- [2] F. Iazzourene et al., “Elettra Booster Commissioning and Operation”, these proceedings
- [3] C. Scafuri & F. Iazzourene, “Elettra New Full Injector High Level Software”, EPAC2006, Edinburgh, 2006
- [4] F. Iazzourene, “TOCA: a Highly User Friendly Application Program for the Tune, Orbit, Dispersion and Chromaticity Correction”, PAC1999, NY 1999.
- [5] L. Battistello et al., “The Control System of the Elettra Booster Injector”, 10th ICALEPCS, Geneva 2005
- [6] ITPP, <http://itpp.sourceforge.net/>
- [7] M. Lonza et al., “Implementation and Operation of the Elettra Booster Control System”, these proceedings.
- [8] F. Iazzourene et al., “Racetrack Users’ Guide”, Version 4.01, internal report ST/M-92/7, July 1992.