Generic Automated Beam Steering and Beam Shaping Programs with an Object Oriented Approach

M. Arruat, A. Jansson, F. di Maio, G.-H. Hemelsoet, M. Lindroos, O. Tungesvik
CERN, CH-1211 Geneva 23, Switzerland

Abstract

The acceleration of the high brightness beam for the Large Hadron Collider (LHC) in the pre-injectors, which will operate at the limit of their performance, will demand that even small perturbations of beam trajectory, orbit and beam shape are corrected on a daily basis. The corrections applied have to be based on the existing machine model to avoid a slow drift of the machine parameters away from the nominal values. To help the operators in this work, Automatic Beam Steering and Beam Shaping (ABS) application programs are being introduced.

We will present the development of a generic and modular software for this purpose. The aim is to end up with a data-driven, portable application package, that has a common look for the different machines and transfer lines, and for the different types of corrections for which it applies. The programs have interfaces to several different instrumentation devices, normally external software applications, used to measure the actual beam positions or shapes.

For calculation of new machine settings, the programs use the CERN developed Mathematica™ program Beam Optics.

1 Introduction

The CERN PS injector complex consists in total of seven accelerators, interconnected by transfer-lines. Large parts of the complex can handle particles in a Pulse to Pulse Modulation mode (PPM) which permits electrons to be accelerated 1.2 s after protons in the same machine. To assure high beam quality in this complex injector chain trajectories, orbits, injection and ejection parameters have to be optimised in each machine part. Traditionally this has been done manually and consequently the corrections have been based on the operators personal experience of the machine steering and optics. The problem with such an approach is that the success of the correction depends on the skill of the individual operator and that it can be tempting to use other elements than the dedicated corrector elements for corrections. This can lead to a drift of parameters which can later cause problems in other machine domains.

2 Existing programs

In 1994 a project was launched to provide operation with a set of generic (re-usable) programs for computer Aided Beam Steering and shaping (ABS) [1] .

The part of the program dealing with the analytical treatment of the machine optics is developed in a symbolic beam optics routine while the user interface and the control interface are implemented in C in the PS control X11/Motif environment [2] .

All the necessary beam optics functions have been implemented as a part of a general Mathematica™ accelerator physics package BeamOptics [3] .

Corrections are calculated whenever new measurement data appears on the MathLink from the application program, and the result is returned to the calling module. The data for the necessary correction matrixes used by the BeamOptics package is generated off-line and stored locally.

The kernel can access the "correction data" in the ABS /Optics database directly via a "Selector" interface function, which is a small program using MathLink.

There are five different application programs , each with a separate Graphical User Interface (GUI). Three of the programs deal with steering in transfer lines, one deals with coherent oscillations at injection into the 26 GeV PS and one with correction of the closed orbit at low energy in the PS.

3 The next generation

The main objectives for the new generation of ABS application software are: i) to create a generic software architecture for different machine parts and for different kinds of corrections, ii) to retrieve all machine data from the optics database i.e. data driven software, iii) to assure maximum re-use of software components between different applications, iv) to create a uniform GUI (Fig. 1), v) to simplify the exchange of measurement data between Measurement software and the ABS correction application through a clearly defined protocol and vi) to make control by means of physical parameters (optic or magnetic) possible instead of electric (Amps or Volts). Furthermore, we plan to extend the use of ABS to more machine domains, such as betatron matching between the PSB and the PS.

To enable efficient management of all necessary optics input data the optics database will be used and we refer to [4] for a detailed discussion of the database and its structure. However, we will, discuss the points of importance for the application program here. The database contains operation-objects in which information relevant to the chosen operation (e.g. injection steering at 1 GeV from the PSB to the PS) can be found. This information can be a list of monitors that can measure beam properties, a list of correctors that can be used, the order in which the beam passes the elements, parameters needed such as energy and Q-value, calibration coefficients, and target values for the chosen type of corrections. The latter will, in the case of steering, be beam position and, in the case of betatron matching, be the Twiss values of the beam at a certain monitor.

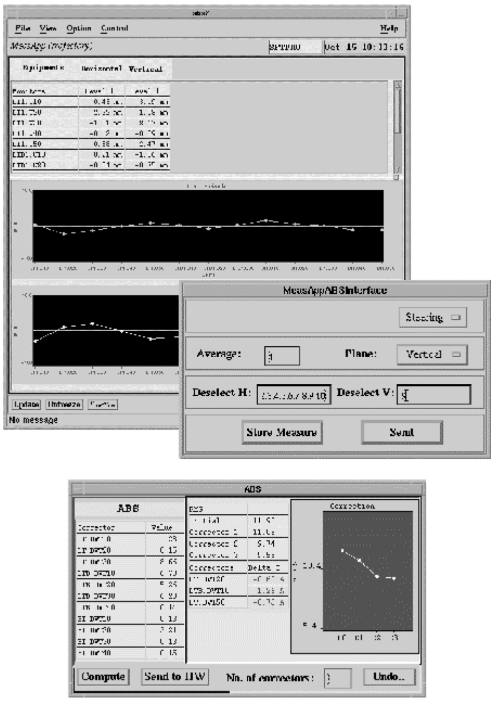

Fig. 1: The new ABS graphical user interface for a steering application. In the back the measurement results can be seen showing the trajectory in an arbitrary beam line, the small window in the middle is the measurement window which sends the measured and treated data to the correction window visible in the bottom of the figure.

To provide easily reusable software components, the interface to the correction application will be implemented as library functions, that can be used by all ABS-supported measurement applications.

As mentioned above, one of the goals has been to make as much as possible of the correction software generic. To achieve this we found it necessary to abstract theinstrumentation and the measured data from the correction application, by means of a well definedinterface or proto-col, for the exchange of data between instrumentation systems and the correction application.

At the PS, it has been common practice to control magnets and other elements in the machine, using electrical values such as Amperes and volts. The use of the optics database provides the possibility to have the coefficient required to convert this into physical values. Consequently, the correction application will operate with physical values like field strengths or bending angles, which the beam optics calculations are based upon.

We have found that we could benefit from using Object oriented Methods and Technology (OMT) for the project. OMT was chosen for the analysis and early design phases, using the CASE-tool Rose, from Rational Software Corp. As the programming language, C++ was chosen. Libraries developed will have a C interface to enable integration with existing measurement applications.

4 Software architecture

The software architecture is schematically shown in Fig.2. It is divided into distinct software components, which communicate by exchanging messages.

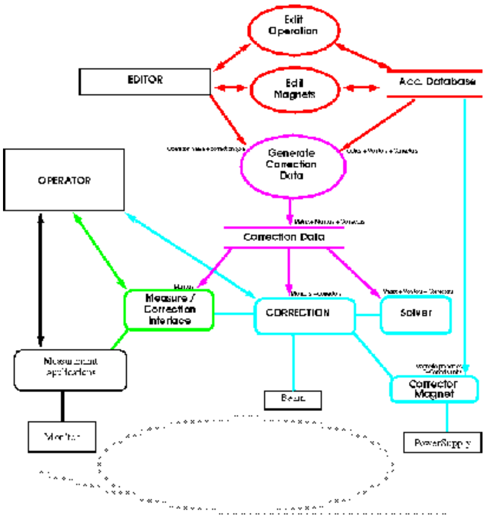

Fig. 2: A schematic view of the software architecture.

Correctors, correction applications and measurement applications are described by C++ classes which define the message protocol between these three categories of components.

A description of the main components follows.

4.1 Correction data

A set of correction data is produced for each "correction" (i.e. operation + correction type). The correction data is composed of a list of monitors, a list of correctors and a correction matrix.

For each monitor and for each corrector, the correction data contains the element's name and its position, relative to the origin of the machine part. In addition, for each monitor, the correction data includes a nominal value for the corresponding measurements.

The correction matrix coefficients describe the effect of each corrector, seen by each monitor. These are obtained from the element position and from the machine optics. This optics is computed from the list of magnetic elements and their nominal settings, which are both associated with the "operation".

4.2 Measurement programs

Measurement programs are dedicated to families of beam instrumentation device (e.g. pick-up, seconadry electron emission grid, wire-beam-scanner ..). These programs are in charge of i) controlling the monitor devices, ii) preparing measurements for the correction application: filtering wrong measurement, and averaging and iii) transmitting the measurement to the correction application.

Any measurement program manages a list of monitors and is associated with one or many correction types (profile monitors can be used for steering and matching). The list of possible "corrections" is derived from these monitors and correction type(s). Once a correction is activated, the measurement programs use the list of monitors from the correction data, to identify the measurements that must be sent to the correction entity.

In our context, measurement programs are existing applications, which are extended, by means of a dedicated library, in order to incorporate the activation of a correction and the transmission of the measurements. These extensions shall be encapsulated into the orrection Interface” objects.

4.3 Correction

A Correction is activated from an operation and correction type selection by a measurement program. Its interface defines “measurement” objects, used by the measurement programs to transmit the monitors’ measurement. Upon reception of measurements from one or several measurement applications, it computes correc-tions and will, on user request, send them to correction magnets.

Correction values, expressed in optical units are computed from the correction matrix and from the requested differences of the beam measurement using a minimisation routine.

Correction objects rely on “Correction Magnet” objects as the interface to the correctors (dipoles and quadrupoles). In our case, while the control system implements “Power Supply” objects, the implementation relies on the accelerator database to provide the “Correction Magnet” with: i) the connection with the control’s elements and ii) magnetic measurement points used for the conversion between magnetic and electrical units.

Correction objects also rely on “Solver” objects that implements the Micado minimisation algorithm. They are implemented, in Mathematica using Mathlink but could also be implemented in C++, if required.

5 Conclusions and status

The move from a set of specific steering programs to a re-usable architecture based on a central database which will be used for steering and matching on many different machines will continue. The plan is to introduce the new software in steps in between the 1998 start-up and the 1999 start-up.

The first components will be tested at the end of November 1997. In a first phase we will start with replacing the Booster steering programs, then continue with the software for the correction of coherent beam oscillations and end with the closed orbit application for the CPS. The next phase will be to introduce the betatron matching routines and to approach the different correction applications needed for the Anti-proton Decelerator (AD) project [5] .

The aim has been to create a set of reusable routines with a clear separation between the public interface and the private implementation in an object oriented programming environment. This will enable the application of the same software solution for a maximum of existing and planned computer aided beam steering and shaping applications.

Acknowledgements

We thank Elena Wildner and the ABS project leader Bruno Autin for helpful comments and useful discussions.

References

[1] M.Arruat, B.Autin, F.Di Maio and M.Martini. *Beam Steering:* “A test Bench for Generic Algorithms in Accelerator Controls”, CERN internal note: CERN/PS 94-09 (AR)

[2] B.Autin, G-H.Hemelsoet, M.Martini and E.Wildner, “Automatic beam steering in the CERN PS complex”, CERN internal note: CERN-PS-95-021 and published in the Proceedings of Particle Accelerator Conference, Dallas, TX, USA, May 1-5, 1995, p. 2178

[3] B. Autin (editor), V. Ducas, O. Grobner, M. Martini and E. Wildner. “BeamOptics, a program of analytical beam optics”, to be published as CERN yellow report

[4] B. Autin, F. Di Maio, M.Gourber-Pace, M. Lindroos, J. Schinzel, “Database for Accelerator Optics”, see proceedings of this conference

[5] D.Berli, J.Boillot, J.Booser, M.Brouet, J.Butkus, F.Caspers, M.Chanel, V.Chohan, T.Eriksson, R.Garoby, R.Giannini, O.Grobner, J.Gruber, J.Y.Hemery, H.Koziol, R.Maccaferri, S.Maury, C.Metzger, K.Metzmacher, D.M, H.Mulder, M.Paoluzzi, F.Pedersen, J.P.Rianaud, C.Serre, D.J.Simon, G.Tranquille, J.Tuyn and B.Williams, “Design study of the antiproton decelerator: AD”, CERN internal note CERN/PS Note 96-43 (AR)