EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN - PS DIVISION

CERN/PS 95-22 (PA)

INTEGRATION OF SYMBOLIC COMPUTING IN ACCELERATOR CONTROL

M. Arruat, B. Autin, G.H. Hemelsoet, M. Martini, E. Wildner

Abstract

Modern symbolic computing programs address not only the manipulation of mathematical functions, but also process control. Symbolic computing can thus be a powerful tool for developing algorithmic engines that can be fully integrated in the controls environment and facilitates modular design of control systems. This technique has been applied to a special class of accelerator problems, namely the beam steering in transfer lines and accelerator rings in the CERN PS Complex. An optimizer for beam trajectories has first been developed off line then linked on line to process interface routines for prototyping. A beam optics program with a full graphics environment and symbolic functions was used to generate necessary information for the optimizer. An application program has then been written to integrate the system into the controls environment.

Fourth International Workshop on Software Engineering and Artificial Intelligence
for High Energy and Nuclear Physics, Pisa, Italy, 3-8 April 1995

Geneva, Switzerland
19 May 1995

© World Scientific Publishing Company

INTEGRATION OF SYMBOLIC COMPUTING IN ACCELERATOR CONTROL

M. ARRUAT, B. AUTIN, G.H. HEMELSOET, M. MARTINI, E. WILDNER
CERN, 1211 Geneva 23, Switzerland

Received (3 April 1995)
Revised

Modern symbolic computing programs address not only the manipulation of mathematical functions, but also process control. Symbolic computing can thus be a powerful tool for developing algorithmic engines that can be fully integrated in the controls environment and facilitates modular design of control systems. This technique has been applied to a special class of accelerator problems, namely the beam steering in transfer lines and accelerator rings in the CERN PS Complex. An optimizer for beam trajectories has first been developed off line then linked on line to process interface routines for prototyping. A beam optics program with a full graphics environment and symbolic functions was used to generate necessary information for the optimizer. An application program has then been written to integrate the system into the controls environment.

1. Introduction

With the advent of symbolic programs which are easy to use and offer powerful programming languages, the symbolic computing is no longer reserved to very specialized domains of theoretical physics but has spread over many scientific and engineering applications. At CERN, a forum called SAP (Symbolics in Accelerator Physics) has been created to debate the issues of symbolic computing in accelerator physics and to stimulate the production of dedicated applications in beam theory of course but also for radio frequency, vacuum, survey, magnets, data manipulation during machine physics experiments and accelerator controls. In the future, many of the applications which have been developed for personal use may become tools which alleviate the burden of the operation of complex machines and improve their reliability. Before proceeding to a general implementation of the symbolic techniques in accelerator controls, some prototyping is requested to test their feasibility and advantages with respect to other techniques[1].

The problem which has been selected concerns the transfer of particle beams from the four rings of the booster synchrotron (PSB) to the proton synchrotron (PS). The beam steering has indeed to be very precise and many parameters are involved in its adjustment so that an automation of the process is more than welcome. The various steps of the project consists of modeling the accelerators and the transfer lines with a beam optics program, connecting the correction algorithms to existing control programs and writing a graphics user interfaces.

2. Physics context

The PS Booster accelerates protons and heavy ions beams coming from a linear accelerator in four rings stacked vertically. The four beams are extracted sequentially at 1 GeV or 1.4 GeV for LHC and recombined vertically before they are transferred to the PS. Before any adjustment, the four beams arrive at the entrance to the PS with different momentum vectors (Fig. 1). The task of the magnetic correctors on the booster side is to make the momentum vectors collinear without an excessive trajectory distortion along the transfer and the PS system must put this common momentum vector tangent to the PS closed orbit. In each transfer channel, the particle trajectories are measured with nine pick-up electrodes and the corrections are provided by four magnets in the horizontal plane and six magnets in the vertical plane. In the PS, two turns are observed with 40 pick-up's in each plane and there are two correctors per plane.

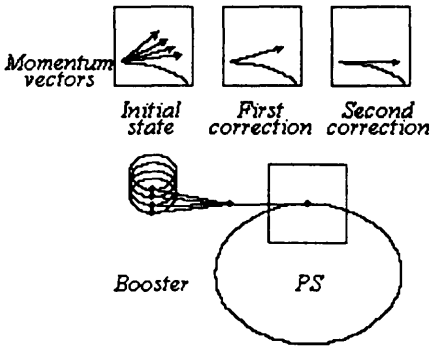

Fig. 1. Transfer from PSB to PS

3. Symbolic beam optics

To correct beam trajectories is a special aspect of linear corrections in presence of experimental errors or even breakdowns of the instrumentation or of the correcting elements. It consists basically of minimizing the norm of a residual vector

$$r = Ax + b \tag{1}$$

where b is a vector whose components are the position measurements at the pick-up's, x is the correction vector which contains the currents to be applied to the correction magnets and A a matrix which is rectangular in general and represents the response of the beam to a set of deflecting fields. The components of the matrix A are derived from the optical model of the machine through the formulae[2]

$$a_{ij} = \sqrt{\beta_i \beta_j} \sin(\mu_i - \mu_j) \tag{2}$$

$$a_{ij} = \frac{\sqrt{\beta_i \beta_j}}{2 \sin \pi Q} \cos(-\pi Q + |\mu_i - \mu_j|) \tag{3}$$

for a transfer line or a ring with Q oscillations per turn. The parameters $\beta_{i,j}$ and $\mu_{i,j}$ are the amplitude and phase of the particle oscillation at the pick-up's and the correctors.

3.1. *Modeling program*

In symbolic optics, an accelerator, a storage ring or a transfer channel is nothing but a mathematical expression made of the operator *Channel* which acts on a sequence of arguments which are the elements of the optical structure:

`Channel[el1,el2,...]`

In the framework of linear optics, three types of elements only have to be considered: straight sections, bending magnets and quadrupoles. The syntax of those objects is:

- `SS[l]` for a straight section of length l;
- `Bend[phi]` for a point-like deflection *phi*, `Bend[l,phi]` for a dipole of length l, `Bend[phi,l,k]` for a combined function magnet of focusing strength k;
- `Q[f]` for a thin lens of focal length f, `Q[l,k]` for a quadrupole of finite length l and focusing strength k.

A very classical focusing structure is defined in the horizontal plane by the commands:

`s=SS[l]; b=Bend[phi]; {qf,qd}=Q[{f,-f}]; ch=Channel[qf,s,b,s,qd];`

The various operations of list processing can be applied to a beam channel: inversion (*Reverse*, *Mirror*), repetition of a given structure (*Repeat*), concatenation with other beam channels (*Join*), selection of elements by pattern recognition (*Cases*), composition of operators (*Composition*), etc. Other operations are specific to beam optics such as the compression of adjacent elements of same type (*Compress*) or the derivation of the structure in the vertical plane (*ToVertical*).

Once the channel is defined, the amplitude β (*BetaFunction*) and the phase μ (*MuFunction*) of the particle oscillation can be determined and entered in the matrix A through the Eq. 2 or Eq. 3.

3.2. *Correction algorithms*

A correction vector x could be obtained directly from Eq. 1 with $r = 0$ by computing the pseudoinverse of the rectangular matrix A, but the result would be very sensitive to experimental errors. Instead, the method of correction minimizes the vector r by successive iterations. At each iteration, the "best" corrector is appended to the corrector set. The corrector strengths that yield the minimum residual vector are

$$x^{(k)} = -\left(A^{(k)T} A^{(k)}\right)^{-1} A^{(k)T} b \tag{4}$$

where the matrix $A^{(k)}$ is composed of k columns $\{i_1 \ldots i_k\}$ of the full matrix A.

The norm of the residual vector r after correction is

$$||r^{(k)}||^2 = b^T b + x^{(k)T} A^{(k)T} b \qquad (5)$$

The method of correction has been coded as a *Mathematica* package which contains two generic functions for beam steering.

- `MicadoMatrix[m,n,type,options]` returns the correction matrix A; m and n are the number of correctors and pick-up's, *type* is *open* for a transport channel or *closed* for a circular machine, and *options* are rules for the optical parameter file and experimental constraints like missing pick-up's or correctors;
- `Micado[A,b,k,options]` returns a list of the correctors and corrections at iteration k; A is the correction matrix, b the beam positions, and *options* are rules which filter the raw measurements to take reference values and experimental errors into account.

4. Link with a data base

In the future, the accelerators of the PS complex will be documented in a relational data base following the model adopted for LEP[3]. A data structure has been produced to analyze the organization of the data and their access from an external interface to the beam optics program. A machine, say the PSB or the PS, is described in a file which contains the topology of the machine and the properties of each element. The topology is a list of elements tagged by their name and placed in the right order. The example shown here uses the *Mathematica* symbolic language[4] and considers four elements out of which three are different:

```
top={el1,el2,el1,el3}
```

Each element is the row of a table and is defined by its name, its type and its properties.

```
tab={{el1,SS,l},{el2,Q,l,k0},{el3,Bend,l,phi}}
```

The purpose of the interface is to convert the raw data into the Channel structure required for *BeamOptics*[5]. This is done with the generic rule

```
rul={name_,type_,p__}->(name->type[p])
```

which is applied to the data table through the pure function

```
f=#/.rul&
```

The table tab can thus be transformed by a mapping of f which produces a set of specific rules. The statement

```
rset=f/@tab
```

returns

```
{el1->SS[l],el2->Q[l,k0],el3->Bend[l,phi]}
```

The channel is produced by application of the rules *rset* and replacement of the head *List* by the head *Channel*:

```
ch=Channel@@(top/.rset)
```

and the final output is

```
Channel[SS[l],Q[l,k0],SS[l],Bend[l,phi]]
```

The technique of list processing and rule applications presented for an elementary test case remains valid for a real machine containing an arbitrary number of elements and properties because it is only based on pattern recognition.

5. Graphics user interface

In the development stage, the *Mathematica* electronic notebook is used to perform a physics experiment. Instrumentation and corrections are accessed to through UNIX pipes which are treated on the same footage as file names. For instance,

```
ReadList["u00"]
ReadList["!eqp_read bt.u00 horiz"]
```

both read the horizontal position of a bunch of particles from the pick-up u00. Using the text processing and graphics facilities, the experiment can be reported in the form of a note without any delay, the measurements archived and further analyzed with all the information contained in a single interactive document.

For the operational application program the notebook is replaced by a standard graphics user interface integrated in the PS Controls System environment[6]. When the beam positions have been acquired and displayed, the operator can validate the measurements and choose the procedure of correction with the interface which monitors the various parts of the system. The C program of the interface is connected to the *Mathematica* algorithm of correction via the two way *MathLink* protocol. An example of the organization of the interface is shown in Fig. 2 for the PSB-PS transfer channel. The screen contains the display of the beam positions, the input to the correction algorithm and the calculated currents ready to be sent to the correction power supplies.

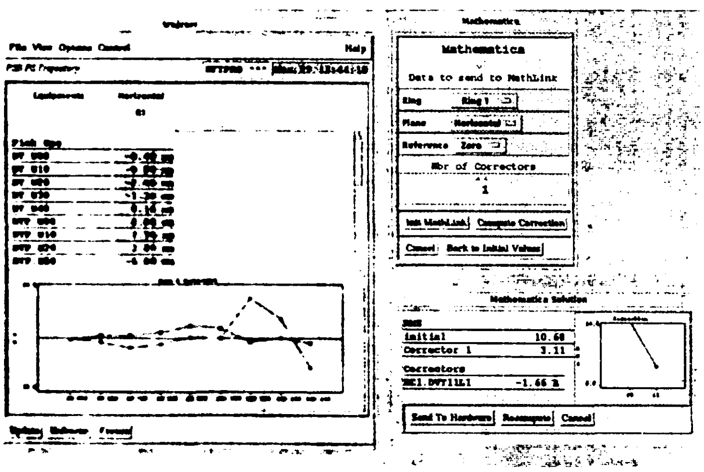

Fig. 2. Graphics user interface for the PSB-PS transfer channel

6. Conclusion

The integration of symbolic computing in accelerator control turns out to be an efficient and rapid way of automating the steering of a particle beam. The modular development of an operation program by separating algorithms and interface and using appropriate tools for each task is cheap since the cost is limited to the purchase of a commercial package and a satisfactory result can be reached within a rather short time with few people working part time on the project. In brief, the advantage of an automatic process is double: it improves the operation and makes the machine more reliable.

References

1. B. Autin, M. Arruat, F. diMaio, M. Martini, *BeamSteering: A test Bench for Generic Algorithms in Accelerator Controls*, CHEP 94, San Fransisco (1994).
2. E.D. Courant, H.S. Snyder, *Theory of the Alternating-Gradient Synchrotron*, Annals of Physics, vol. 3 (1958).
3. R. Bailey, A. Belk, P. Collier, M. Lamont, G. de Rijk, M. Tarrant, *Development of the LEP high-level control software using Oracle as an on-line database*, Proc. ICALEPCS (1993).
4. S. Wolfram, *Mathematica, A System for Doing Mathematics by Computer* (Addison-Wesley, Redwood City, 1991).
5. B. Autin, E. Wildner, *Beam Optics*, to be published (1995).
6. F. Perriollat, Ch. Serre, *The New CERN PS Control System - Overview and Status*, Proc. ICALEPCS (1993).