THE APPLICATION OF THE SXF LATTICE DESCRIPTION AND THE UAL SOFTWARE ENVIRONMENT TO THE ANALYSIS OF THE LHC[*]

W. Fischer, F. Pilat[†], V. Ptitsin, BNL, Upton, New York

Abstract

A software environment for accelerator modeling has been developed which includes the UAL (Unified Accelerator Library), a collection of accelerator physics libraries with a Perl interface for scripting, and the SXF (Standard eXchange Format), a format for accelerator description which extends the MAD sequence by including deviations from design values. SXF interfaces have been written for several programs, including MAD9 and MAD8 via the doom database, Cosy, TevLat and UAL itself, which includes Teapot++. After an overview of the software we describe the application of the tools to the analysis of the LHC lattice stability, in the presence of alignment and coupling errors, and to the correction of the first turn and closed orbit in the machine.

1 INTRODUCTION

The modeling of large accelerators such as the LHC requires a sophisticated, flexible and powerful modeling software environment. On the one hand, complex problems have to be studied requiring non-standard modeling techniques, such as tracking two beams, assigning measured values to individual magnets, dealing with complex alignment tolerances for triplet assemblies, etc. On the other hand, large accelerators are nowadays international collaborative efforts among laboratories and the necessity arises to exchange lattice and measurement information between groups in a transparent, model independent way. To fulfill the first set of requirements we built up our model in the framework of the UAL software, and to fulfill the need of information exchange we use a flat machine representation, the Standard eXchange Format (SXF) to describe the accelerator.

Section 2 will explain in more detail the software building blocks, UAL and SXF. Section 3 will discuss the unique challenges of modeling the LHC, in particular the study of alignment tolerances in the interaction region dipole and triplet assemblies. In Section 4 as an example of application, we will discuss a way to close the first turn and to correct the orbit in the LHC in the presence of realistic alignment errors.

2 THE MODELING SOFTWARE ENVIRONMENT: UAL AND SXF

The Unified Accelerator Library (UAL) [1] is an object oriented and modular software environment for accelerator physics which comprises an accelerator *object model*

for the description of the machine (SMF, for Standard Machine Format), a collection of *physics libraries*, and a *Perl shell* that provides a homogeneous interface for integrating and managing these components. Currently available physics libraries include TEAPOT++, a collection of C++ physics modules conceptually derived from TEAPOT [2], and ZLIB [3], a differential algebra package for map generation and manipulation.

UAL has already been used to build up a realistic model of the Relativistic Heavy Ion Collider (RHIC) where measured field and alignment data are automatically read from a relational database and applied to individual elements in the machine [4] [5]. It has been used to model CESR and is being used, as we will describe in the following, for the LHC. Access to the UAL suite is provided via a Perl-based *shell* environment. Each UAL project (RHIC, CESR, LHC, etc.) has developed a set of Perl modules which provide a user friendly means of calling the ensemble of UAL programs, e.g., correction procedures, particle tracking via TEAPOT++ or differential algebra and mapping techniques.

The SXF (Standard eXchange Format) is an ASCII accelerator description that carries lattice information and also individual quantities specific to an element such as measured field and alignment data. SXF has been developed as an answer to the need for easy exchange of lattice information among laboratories. SXF parsers to several existing codes, such as UAL, MAD, Cosy and TevLat, have been written and used.

An SXF lattice description is an ASCII listing that contains one named, *flat*, ordered list of elements, delimited as {...}, with one entry for each element. The list conceptually resembles a MAD "sequence" describing the entire machine. The difference is that an SXF sequence can carry additional information such as field harmonics, alignment errors, aperture limitations, etc. that are specific to the individual element. The syntax is supposed to be adapted for ease of reading by human beings and for ease of parsing by LEX and YACC. Here is an example of a simple SXF sequence of 2 elements, a dipole and a marker:

```
TwoElementExample sequence {

bh.r1   sbend { tag=bh at=200.0 l=10.0

body={ kl=[0.0045 0.001 0.0002 0.045]
       fint=0.01 e1=0.001 e2=0.002
       h1=0.1 h2=0.02 hgap=0.002 }
align ={al=[0.001 0.001 0 0 0.0005] }
body.dev={kl=[0.001..] kls=[0.001..]}
entry.dev={kl=[0.001..] kls=[0.001..]}
```

```
exit.dev={kl=[0.001..] kls=[0.001..]}
align.dev={al=[0.005 0.005 0 0 0.001] }
aperture={ap=[1 0.03 0.02 0.001 0.001]}
};
m1 marker {   };

endsequence at=200.0
}
```

SXF supports MAD attributes; attributes are grouped in logical "buckets" such as *body*, which contains the element general attributes, *align*, which describe intentional misalignments, etc. The buckets that end with ".*dev*" contain deviations from the design values, such as field (*body.dev*) and alignment (*align.dev*) errors. Quantities that are logically related, such as field multipoles, misalignments, apertures are grouped in vectors, such as *kl, al, ap*. The vector *kl* for instance contains normal field harmonics up to order 20; only orders up the last non-vanishing order need to be explicitly written. A more detailed description of the grammatical and lattice description rules of the SXF, as well as more examples of SXF sequences can be found in [6].

3   THE ALIGNMENT MODEL FOR THE LHC INTERACTION REGIONS

Since SXF is a *flat* ordered list of elements it does not support the simultaneous misalignment of adjacent elements. The introduction of an element *hierarchy* is needed to align several elements by the same amount. This is of interest for two reasons. First, magnets are often modeled by more than one element and all these elements should be misaligned by the same amount. For example, a magnet model may consist of a body and two end kicks; the body itself may be modeled by several kicks. Second, sometimes there are several magnets assembled in one cryostat. It is then desirable to displace the whole cryostat while allowing for the individual magnets within to be independently misaligned relative to the cryostat.

The proper description of alignment errors is of great importance to the US-LHC collaboration since the US will deliver assembled cryostats of interaction region magnets to CERN. In the design phase an estimate of tolerable alignment errors is needed. Once magnets are assembled in a cryostat, the best position of the cryostat must be found in order to minimize the harmful effects of field and alignment errors.

We developed a SXF-like description that allows the simultaneous misalignment of adjacent magnets and a filter program that can merge this information into a canonical SXF file. Here is an example of how several entities can be misaligned with respect to a commom cryostat, which also has a global misalignment relative to the reference axis.

```
// a misaligned element that contains
// misaligned elements
q2con = {
```

```
align = { al = [ 0.002 -0.001 0.003
                 0.005 -0.005 0.007]}
components = {
 q1t2.r1 = {
  align.dev = { al = [ .0 -0.002
                  0. 0. -0. 0.007]}}
 q1t3.r1 = {
  align.dev = { al = [ 0.001 0. 0. 0.
                   -0.006 0.]}
  components = {
    x1.r1 = {
     align.dev = {al=[ 0. 0.2 ]}}
    x2.r1 = {
     align.dev = {al=[ .2 0. 0.
                      0. 0. 0.1]}}
} } } }
```

we used this technique to assign alignment errors to the LHC interaction region assemblies and study the impact of such errors on the first turn and closed orbit correction. The analysis and preliminary results are summarized in the following section.

4   FIRST TURN AND CLOSED ORBIT CORRECTION

In the presence of misalignment the particle orbit is no longer ideal. The orbit excursions, even with moderate amount of misalignment, in a large machine like the LHC may become so big that a particle reaches the aperture limits defined and gets lost. As in the case of a real accelerator ring, in order to keep the beam closed orbit inside the physical aperture, a first turn beam steering method must be applied.

Two first turn beam steering techniques were implemented in TEAPOT++. One is the method recently developed by H. Grote for MAD [7]. This method performs the correction using two beam steering correctors and two beam position monitors in the region where the orbit excursions become too large. With the spacing between the two BPMs about $\pi/2$ in betatron phase it provides in fact correction of both the orbit position and the orbit angle.

The other method implemented in TEAPOT++ is the so called sliding-bumps method. The closed bumps are set with of three neighboring dipole correctors. The whole ring interval is divided in a series of consecutive bumps that are overlapping since every corrector can be used in the construction of three different bumps. Each bump by definition only affect the orbit inside itself.

One correction step consists of consecutive one-by-one bump corrections along the ring until a lattice point is reached, where a particle loss or a big particle trajectory deviation is detected. The limit value of the trajectory deviation that defines the start of the correction procedure is determined as a parameter to the first turn steering command. Because the trajectory downstream of the corrected interval should also be affected by the correction, the last
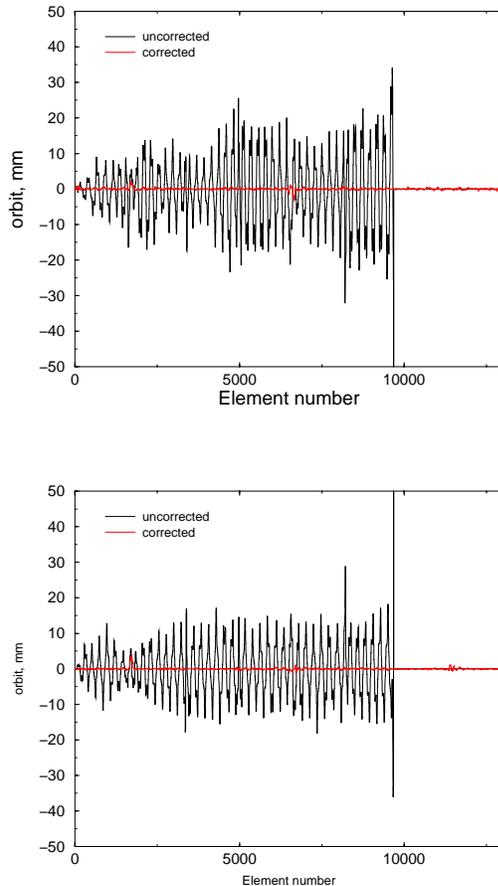
Figure 1: First turn correction in horizontal and vertical planes

bump of the corrected interval is made "open", meaning that the strength of one or two of the last bump correctors is set to zero. After one correction step is done, the particle is tracked again through the lattice (or, operationally, the beam is injected again) until the next limit trajectory deviation is encountered (or, with beam, the next beam loss point) and the correction is repeated. Using the sliding-bumps technique not only steers the particle through the first turn of the accelerator but also provides a corrected orbit at all BPM positions along the ring. However, this method is more time consuming than the one previously described. The sliding-bumps method is used in within TEAPOT++ also as a closed orbit correction algorithm. The only differences are that the input data are closed orbit instead of particle trajectory data and that there is no need to "open"any bumps.

Simulation studies confirmed that the correction methods implemented work well for the LHC lattice in the presence of alignment errors, both for first turn steering and closed orbit.

Figure 1 and 2 show the horizontal and vertical particle

trajectory during the first turn for the LHC, before and after correction. The method used here is the sliding-bumps.

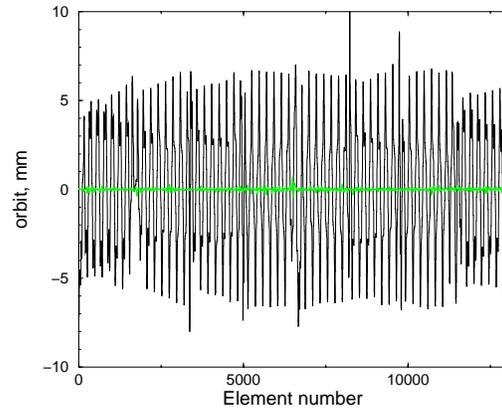Figure 2 show the horizontal closed orbit before and after correction.

Figure 2: Closed orbit correction in the horizontal planes

5 CONCLUSIONS

A software environment has been developed for LHC simulations whose flexibility and modularity ease the task of adding new modeling features. A standard interface, the SXF, extends the functionality of the MAD sequence and can interface several existing accelerator codes. We used this environment to study the effect of misalignment of the IR assemblies on beam dynamics, noticeably on the first turn trajectory and the closed orbit. The model and correction methods proved to work for the LHC lattice and we plan to use them for systematic studies in the near future. We plan on adding the capability of modeling roll alignment errors in the IR assemblies and study the resulting coupling effects in the next few months.

6 REFERENCES

[1] N. Malitsky and R. Talman, AIP 391 (1996).

[2] L. Schachinger and R. Talman, Particle Accelerators, 22, 35 (1987).

[3] Y. Yan, SSCL-300 (1990).

[4] F. Pilat, C. Trahern, J. Wei, T. Satogata, S. Tepikian, "Modeling RHIC using the Standard Machine Format Accelerator Description", PAC97 (1997).

[5] F.Pilat, M.Hemmer, V.Ptitsin, S.Tepikian, and D.Trbojevic, "Processing and analysis of the measured alignment errors for RHIC", PAC99 (1999)

[6] H. Grote, J. Holt, N. Malitsky, F. Pilat, R. Talman, C. Trahern, "SXF: definition, syntax, examples", RHIC/AP/155 (1998).

[7] H. Grote, "Beam Threading in the LHC", EPAC98 (1998)