Recent Developments On The EMMA On-Line Commissioning Software

J.S. Berg[*], Y. Giboudot[†], D. J. Kelliher[‡], S. Machida[‡], F. Méot[§], B. Shepherd[¶], S. Tygier[‖]

Abstract

The EMMA (Electron Model for Many Applications) FFAG experiment at Daresbury will involve on-line modelling (a "Virtual EMMA") based on stepwise ray-tracing methods. Various aspects of the code of concern and of its interfacing to real world - machine and users - are addressed.

INTRODUCTION

The stepwise ray-tracing code Zgoubi [1] is the on-line model code for the world's first non-scaling FFAG experiment, EMMA [2], in support of machine control/command and general operation. This paper presents particular aspects of recent software developments (i) which the Zgoubi code has been subject to for that purpose, (ii) regarding its user interfacing software PyZgoubi, (iii) regarding the high level EPICS machine interfacing software.

A companion paper [3], useful for better understanding, contains additional details concerning the way the EMMA cell, ring, injection and extraction regions are simulated ; details of the Zgoubi method can be found in the users' guide [1] and in Ref. [4]. A development site has been created on the SourceForge repository [5].

RECENT DEVELOPMENTS IN ZGOUBI

Tab. 1 shows a typical structure of a EMMA data list to Zgoubi ("zgoubi.dat" in the following). Data lines are commented for understanding. Many aspects (methods, optical elements, tricks...) of the "Virtual EMMA" application are subject to detailed insight in Ref. [6]. Those various procedures as appearing in Tab. 1 data list which have been subject to developments in view of the EMMA experiment are addressed in the following.

Zgoubi coordinate frame nomenclature is used here, namely, X, Y, Z stand for longitudinal, horizontal and vertical coordinates.

• CHANGREF is used for magnet [mis-]positioning. In Tab. 1 are positions of the reference frame at the corner of the EMMA 42-sided polygon. The normal procedure has three arguments, X-shift, Y-shift, rotation about the Z axis, applying in that particular order ; a different order can be achieved by stacking several CHANGREFs. A new-style CHANGREF allows a series of up to 9 arbitrary transforms,

[*] BNL, Upton, NY
[†] Brunel University, Middlesex
[‡] STFC/RAL/ASTeC, Chilton
[§] CEA & IN2P3 LPSC INPG UJF, Grenoble
[¶] STFC/DL/ASTeC, Daresbury
[‖] University of Manchester

Table 1: EMMA ring, from object definition at injection, to extraction.

```
'OBJET'
+5.171103865921708e+01 15MeV
2
1 1
9.95 -113.446 0. 0. -85.486 0.67721442 'i'
1
'MULTIPOL' septum                                           Injection line : septum
0020 .plt
10. 10. -2.9 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 1.00 1.00 1.00 1.00 1.00 1. 1. 1. 1.
4 .1455 2.2670 -.6395 1.1558 0. 0. 0.
0. 0. 1.00 1.00 1.00 1.00 1.00 1. 1. 1. 1.
4 .1455 2.2670 -.6395 1.1558 0. 0. 0.
0. 0. 0. 0. 0. 0.00e+00 0.00e+00 0.00e+00 0.00e+00
#320—20—320
2 0.00e+00 5. 0.00e+00
'DRIFT' dr .plt ! distance to septum exit
2.1011412
'DRIFT' dr .plt ! distance to septum vessel opening
2.7151711
'MARKER' RingS
............                                                 kicker section (not shown)
'CHANGREF'                                                   Begin EMMA cell
ZR -8.571428571429                             reference frame orientation at polygon corner
'MULTIPOL'                                                   Simulation of QD magnet
0
7.5699 5.3 0. 2.49312 0 0 0 0 0 0 0     length, radius, $B_{dip}$, $B_{quad}$, ..., $B_{20-pole}$
0. 0. 0. 0. 0. 0. 0. 0. 0. 0.                               fringe field data, entrance
4 0.296571 4.533219 -2.270982 1.068627 0. 0.           Enge coefficients, entrance
0. 0. 0. 0. 0. 0. 0. 0. 0. 0.                               fringe field data, exit
4 0.296571 4.533219 -2.270982 1.068627 0. 0.              Enge coefficients, exit
0. 0. 0. 0. 0. 0. 0. 0. 0. 0.                               poles' skew angles
0.1                                                         integration step size
1. 0. 0.0.                                  magnet positioning : Kpos, $\Delta X$, $\Delta Y$, $\theta_Z$
'DRIFT' sd
5.                                                          short drift
'MULTIPOL' QF                                               Simulation of QF magnet
0
5.8782 3.7 0.5030623 -2.47715 0 0 0 0 0 0 0
0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
4 0.296571 4.533219 -2.270982 1.068627 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
4 0.296571 4.533219 -2.270982 1.068627 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0.1
1. 0. 0.0.                                   magnet positionning : Kpos, $\Delta X$, $\Delta Y$, $\theta_Z$
'DRIFT'
21.                                                         long drift
'MARKER'                                                    End EMMA cell
............                                                 RF cavity, every two other cells
'DRIFT' ld
10.5
'CAVITE' #cav1
7
0. 1.300525350355e+09
-6.4e+04 -9.771174812308e+00
'MARKER' BPMCAV
'DRIFT' ld
10.5
'REBELOTE' 8 0.2 99                                         multipass acceleration
.........                                                    extraction line follows
'END'
```

taken in arbitrary order from XS, YS, ZS, XR, YR, ZR with the S (respectively R) suffix standing for "shift" ("rotation"). Tab. 1 shows instances of the use of the new-style CHANGREF. Not all transforms XS, YS, ZS, XR, YR, ZR have been installed yet, this remains to be completed.

• DIPOLES, compared for instance with MULTIPOL, has the advantage of adding overlapping fringe fields [7]. An option as been introduced to provide a field model compatible with (QD,QF) doublet, namely [3, 6]

$$B_Z = Bz_{0,i} + \sum_{i=1}^{N} \left[\mathcal{F}_i(R, \theta) \sum_{m=1}^{M} b_{m_i}(R - RM_i)^m \right]$$

• EMMA is a new field map handling procedure specific to the EMMA doublet. It accepts polar or Cartesian mesh maps. The existing TOSCA procedures allow (Tab. 2)
(i) a single field map, describing the FD cell with frozen

Table 2: Manipulating field maps with `EMMA`.

```
'EMMA'
0 0
-1E-3 1. 1. 1.                    Global normalization of map data
emma cell field map                                     Comment
197 81 1 0           Number of nodes in Y,X,Z. Mode (=0 : d_FD frozen)
1. 1. 0.                               a_F, a_D, distance d_FD
Dax265.Fon.cart.table                          Name of F quad map
Dax265.Don.cart.table                          Name of D quad map
0 0 0 0
2                                          Interpolation method
.1                                    Integration step size, cm
2 0 0 0                                   Field map positionning
```

quadrupole arrangement and fields, (the "TOSCA" keyword also does this job [1]),

whereas, in addition, `EMMA` manipulates field maps in the following way [8] :

(ii) the FD cell is described by a single pair of field maps, one for QD (QD/on, QF/off), one for QF (QF/on, QD/off), with QD-QF radial distance d_{FD} frozen, and free field amplitude coefficients a_F, a_D,

(iii) the FD cell is described by an ensemble of pairs of field maps of (ii) type, in arbitrary number, each pair with its d_{FD} value attached. EMMA then interpolates in this set to get the field map corresponding to an arbitrary d_{FD} as specified by the user. This working mode allows flexible use of the Zgoubi fitting procedure, with three variables : fields (a_F and a_D quantities in Tab. 2) and d_{FD}.

(iv) a unique pair of maps is used with arbitrary d_{FD}. Possible mutual inductance is in this case assumed small enough that the method yields a conveniently precise estimate of a_F, a_D and d_{FD} values.

• `FAISTORE` creates a ASCII coordinate output file. The ASCII output file format has been modified. There are now header lines giving the name and unit for each column. The entire record is now stored on one line instead of five in earlier releases. This makes communication with other software such as PyZgoubi more reliable, as the read functions can be simpler, and will be robust against future changes.

• `FIT, FIT2` matching procedures are useful in various exercises in EMMA, as for instance tuning of the septum and kicker magets at injection and extraction (a beam injected or extracted with arbitrary energy must be placed on, or come from, the appropriate orbit in the ring). New-style `FIT, FIT2` procedures accept value ranges for the variables defined by an interval, whereas in the previous version the value ranges for the variables can only be defined by a percentage of variation from the initial value which is not convenient if an initial value is close to zero.

• `REBELOTE` allows multi-pass through the structure. The normal procedure has three arguments, $NPass$ (number of passes through the data list), $switch$ (switch on/off verbose), $Option$ (for instance, 99 for multiturn tracking in a ring : final coordinates after one pass are taken as initial coordinates for the next pass). A new-style `REBELOTE` has a fourth argument, $Lmnt\#$, which tells the pointer to Zgoubi data list where to begin the loop (rather than stepping back to the begining, as in the old style). As a consequence, the multi-pass structure is that part of the data list standing between element number $Lmnt\#$ and `REBELOTE`. A practical application is to allow the whole structure to be defined in one Zgoubi input file : beginning at the start of the injection line, making a number of turns around the EMMA ring and finally extraction into the diagnostic line [6, 9].

The `REBELOTE` keyword can be emulated by PyZgoubi (see below) by using loops when adding elements to the lattice. For example the line sections inject_line, extract_line, and ring_cell, could be combined like so :

```
lattice.add(injection_line)
for lap in range(50):
    lattice.add(ring_cell)
    lattice.add(extraction_line)
```

PYZGOUBI INTERFACE

PyZgoubi [10] is a Python [11] library that wraps around Zgoubi. Zgoubi is traditionally used by creating an input file containing the beam and lattice, calling the Zgoubi program to run it, and producing output files. With PyZgoubi a Python program is written that can create lattices, and track beams through them. Zgoubi is run in the background, and the results return into the program as arrays. This allows large flexibility. For example it is possible to modify the lattice or beam depending on the result of a previous run, or based on data stored in a file.

PyZgoubi has the ability to simultaneously start multiple Zgoubi processes each with a subset of the beam, and then recombine the results. On computers with multiple CPU cores this can reduce computation time significantly.

The script containing the description of the beam line has been written in PyZgoubi. It is composed of :
• the ALICE to EMMA injection line starting from the entrance dipole situated after the ALICE linac and finishing after the injection septum,
• the EMMA ring with the possibility of only considering the four sector option (only 23 cells),
• the extraction or also called diagnostic line up to the final Faraday cup.

The magnets are modelled with hard edges and considering their magnetic length. A vacuum chamber is implemented using the Zgoubi key word "CHAMBR" that stops particles with larger excursion than the threshold given for each section of the beam line. The script reads the positions of the ring magnets and the currents of all the magnets from a text file created by the EPICS server (see below). The text file contains as well the characteristics of the input beam (initial kinetic energy, number of particles, 6D initial conditions, emittance) and the number of turns requested to be performed by the beam (in full ring configuration). The septum and kicker magnets (injection and extraction) are implemented in such a way that their field value decreases turn after turn following a tunable function. This allows the study of remanent septum stray field and kicker ringing. The output of the script is a list of BPM readings corresponding to the average value at each BPM location of the transverse coordinates of all the particles launched (just

as a real BPM returns the position of the center of charge of a bunch). Similarly, the transverse coordinates at the YAG locations can be written to an image file.

EPICS INTERFACE

Online modelling is implemented using a *virtual accelerator* paradigm [12]. Using the EPICS Portable Channel Access Server (PCAS) [13], an EPICS server has been developed which runs an instance of Zgoubi. This "virtual IOC" publishes a set of process variables (PVs) which mirror those of the real machine, preceded by a prefix to distinguish them from "real" PVs. Upon receiving a change in one of these process variables (a change in magnet current, for instance), the Zgoubi model is re-run and the beam positions at the BPMs are updated in the relevant "virtual PVs". The block diagram of EpicsZgoubi is shown in Fig. 1.
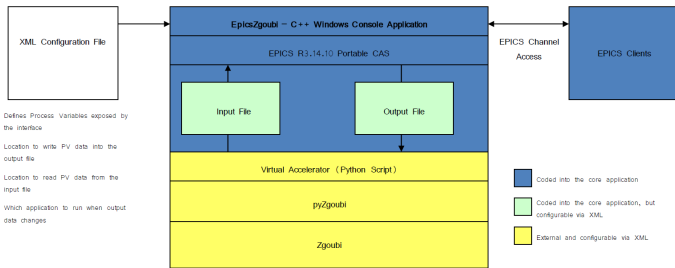

Figure 1: Block diagram of EpicsZgoubi server.

The main advantages of this system are twofold. Firstly, high-level software can be developed offline, using an instance of the virtual IOC. The interaction between high-level software and the machine can be tested prior to deploying it on the real accelerator, enabling bugs to be ironed out faster and techniques to be refined. Secondly, it allows a direct comparison of results between the real and virtual accelerators to be performed online during operation. Settings can be tested on the virtual accelerator before applying them to the real one, and problems can be rapidly diagnosed on the virtual accelerator without worrying about hardware issues.

The EpicsZgoubi server, like EPICS itself, is agnostic both in terms of the top and bottom levels. Any accelerator code can be used as a model, providing it can accept input from a text file in a defined format. Equally, at the top level, any software development environment can be used as long as it supports an EPICS interface. In our case, Python was chosen as the "main" development environment due to its openness, cross-platform compatibility and extensibility. The PyEpics3 package [14] was used to interface with EPICS. This does not preclude the use of other codes to develop EMMA software.

Fig. 2 is an example of a high-level software panel we have developed. The pull down menu at the top controls allow one to operate the virtual or real machine, save the data into the archive, take a snapshot of the window, and so on. There are several tabs for each task. The example

shows the reading of beam position monitors and a zoomed view. The data was read through EPICS, which is created by the virtual accelerator. However, the all software can read the data from the real accelerator exactly in the same way when it becomes ready. For more details on software for EMMA commissioning see Ref. [15].
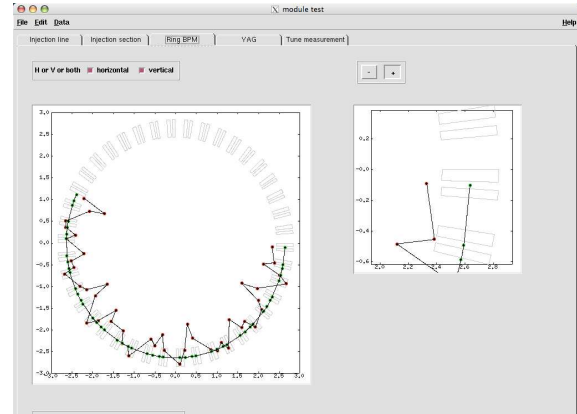

Figure 2: Screenshot of high-level software panel.

REFERENCES

[1] The ray-tracing code Zgoubi, F. Méot, NIM-A 427 (1999) 353-356.

[2] The EMMA Non-scaling FFAG, T.R. Edgecock et al., these proceedings.

[3] Beam dynamics simulations regarding the experimental FFAG EMMA, Y. Giboudot, D. Kelliher, F. Méot, T. Yokoi, THPD023, these proceedings.

[4] Beam dynamics simulations in FFAGs : the ray-tracing code Zgoubi, F. Méot, BD-Newsletter 43, C.R. Prior & W. Chou Eds. (Aug. 2007) 47-54.

[5] http ://sourceforge.net/projects/zgoubi/

[6] Virtual EMMA using Zgoubi, F. Méot, EMMA commissioning workshop, Daresbury, Dec. 2009.

[7] Developments in the ray-tracing code Zgoubi for 6-D multi-turn tracking in FFAG rings, F. Lemuet, F. Méot, NIM A 547 (2005) 638-651.

[8] Optical matching of EMMA cell parameters using field map sets, Y. Giboudot, F.Méot, Procs. PAC 09, Vancouver.

[9] Injection into EMMA, septum and kicker tunings, F. Méot, rep. LPSC-10-24 (2010) ; Simulation of injection into EMMA, injection section, F. Méot, rep. LPSC-10-33 (2010).

[10] PyZgoubi interface to Zgoubi, S. Tygier, D. Kelliher ; http ://sourceforge.net/projects/pyzgoubi/

[11] Python programming language ; http ://python.org/

[12] Beam-commissioning study of high-intensity accelerators using virtual accelerator model, H. Harada *et al*, NIM A 602 (2009) 320-325.

[13] Portable Channel Access Server ; http ://www.aps.anl.gov/epics/extensions/cas/index.php

[14] PyEpics3; http ://cars9.uchicago.edu/software/python/pyepics3/

[15] First results from EMMA commissioning, S. L. Smith et. al., THPEC090, these proceedings.