Simulation Using Xorbit with EPICS*

Kenneth Evans, Jr.
Accelerator Systems Division
Argonne National Laboratory, Argonne IL 60439

ABSTRACT

The accelerator code Xorbit has an interface to the Experimental Physics and Industrial Control System (EPICS). This means that machine data such as magnet settings can be sent to Xorbit via EPICS, and the resulting orbit parameters such as beta functions, etc., can be calculated. In addition, Xorbit can be made to simulate the real machine, whether the latter is running or not. To accomplish this for the APS, there is a database of process variables in an IOC corresponding to each APS ring and beamline. These process variables are very similar to the real process variables that read and set power supplies and read monitors, except that when a setting is changed, Xorbit is notified via a callback, calculates a new orbit, and outputs the appropriate readbacks to the database. By attaching the string "Xorbit:" to a control name the control system will respond to the simulation rather than the real system. This allows the testing of control algorithms, orbit diagnostics, and many other components of the control system (as well as EPICS itself). It is fast enough to be visually similar to accessing the real system.

I. INTRODUCTION

There are a number of reasons to have an accelerator code that is directly linked to a control system. Physics application programs, such as orbit control, may be developed and tested even before the machine is operational. After the machine is operational, these programs may be tested without using valuable machine time and without the possibility of damaging the machine. In addition, when the machine is running, magnet settings can be readily transferred to the code for analysis and the actual orbit compared to the calculated orbit. The effect of possible changes in magnet settings can be easily tested and evaluated. Theoretical quantities such as beta functions, dispersion, phases, and tune, which often may not be readily available from the instrumentation (especially during the early stages of commissioning) become available, either in lieu of the real values or for comparison to them when they do exist.

It is an additional benefit when the orbit code has a graphical interface so that these data are displayed as soon as they are calculated. A history mechanism is useful for storing results and comparing the time evolution of important parameters, such as the tunes. These are capabilities that are not found in most of the standard physics lattice codes, such as MAD [1], however.

The most difficult problem of connecting an accelerator code to the control system lies in the fact that in the code, or for almost any orbit calculation for that matter, the orbit is calculated given the magnetic fields in the magnets, while the control system does not know these magnetic fields directly. The control system only sets and reads the currents in the power supplies that control the magnets. Consequently, it is necessary to know the somewhat complicated relationship between the currents and the magnetic fields to make the conversion. This problem is further compounded by the fact that this relationship has hysteresis effects, that is, depends on the past history of the magnets.

This paper describes Xorbit, a graphical accelerator code that has been developed at the Advanced Photon Source (APS), and how it is linked to the EPICS control system in use there. Familiarity with EPICS is assumed. Section II describes Xorbit, and Section III describes how it is connected to EPICS. Section IV describes the database that holds the Xorbit process variables (PVs). Section V discusses the problem of the

---

* Work supported by U.S. Department of Energy, Office of Basic Energy Sciences under Contract No. W-31-109-ENG-38.

conversion between current and field, and Section VI describes a special EPICS record type, the IK record, that performs the conversion.

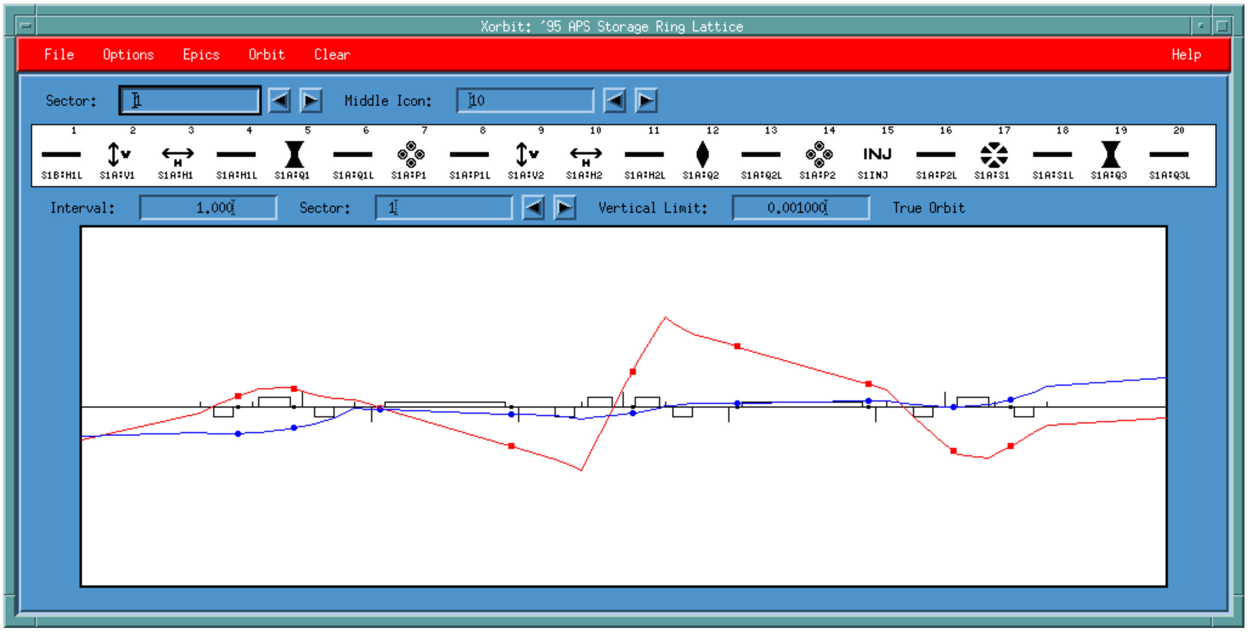

Fig.1 The Xorbit main window. Twiss parameters and a tune diagram are displayed separately.

II. XORBIT

Xorbit [2] is a UNIX computer code designed to help in the design and operation of circular accelerators, storage rings, and beamlines. It is a lattice design code that calculates the orbit primarily by using 2 x 2 transport matrices. (Nonlinear elements are simulated with kicks, and random element displacements are supported. Dispersion is obtained from the 2 x 2 matrices.) It has been designed to be fast and easy to use and to provide immediate feedback when changes are made in the lattice. There are many options and features that will not be described here.

Xorbit has a GUI interface based on Motif and is designed for ease of use and immediate feedback. The main window is shown in Fig. 1. After entering the lattice via an input file, almost all of the interaction with Xorbit is via the mouse and keyboard. All of the results including the orbit and Twiss parameters (beta functions, tunes, *etc*.) are readily displayed in graphic form along with previous values of these quantities if desired.

Several of the principal APS physics application codes, such as Orbitscreen [3] and ADT [4] (which both display data such as monitor readings) and Corbit [5] (which does orbit correction), have an option to use Xorbit data rather than data from the real machine. This means, for example, that orbit correction can be done with the same code whether simulated or real, and the resulting monitor readings during correction can be viewed the same way in both cases. There is very little conceptual difference in using the simulation compared to working on the real machine.

There is also a means to connect programs, such as Mathematica [6], or spreadsheets like Wingz [7] or Excel [8] to the orbit calculations in Xorbit. This capability, which is discussed in Reference 2, is actually an imitation of EPICS that uses UNIX interprocess communication (IPC) instead of channel access. It has the same channel access function calls and can be used somewhat interchangeably with EPICS tools [9-10] that connect to these same programs. (Re-linking is necessary.) These connections can be made at the same time the EPICS connections are in effect and extend the capability to analyze data from EPICS.

III. CONNECTION TO EPICS

In what follows we will use the mneumonic BPM for beam position monitors (hardware diagnostics that measure the position of the beam in the longitudinal directions, x and y) and just monitors for EPICS monitors (software which determines when a PV has changed).

The basic operation of Xorbit with EPICS is as follows: The user of Xorbit controls the EPICS connection via the EPICS menu after Xorbit is up and running with the lattice defined in the input file. When the EPICS connection is started by the user, Xorbit starts channel access and searches for the necessary PVs. It then calculates an orbit (the closed orbit for a ring or the trajectory corresponding to specified initial conditions for a beamline) and puts the orbit coordinates into the BPM readbacks and the magnet settings into both the magnet readbacks and the setpoints (all of which are PVs). It also establishes monitors on the magnet setpoints. Thereafter, when a channel access client changes the setpoints, Xorbit is notified via the monitors, calculates a new orbit, and changes the BPM and magnet readbacks accordingly. Notice that Xorbit **sets** readbacks. It is simulating the hardware.

The orbit calculation is nearly instantaneous for the smaller APS rings and beamlines, but takes a few seconds for the storage ring on a SPARCstation 10.

The user of Xorbit can also change magnet and other parameters himself inside of Xorbit. Or they can be changed through the connections to Mathematica, etc. All of the Xorbit features are available. As the orbit and magnet settings change, the BPM and magnet readbacks are reset. In most cases the setpoints are also reset so they will be consistent. The user can explicitly reset all the setpoints, as well. The mechanism for resetting the setpoints is described in the next section.

For those interested in more technical programming information, these simultaneous channel access and X-windows events are handled as follows: A monitor event triggered by channel access calls an Xorbit routine that changes the appropriate variable inside Xorbit and sets a switch that something has changed. A work procedure (one that runs only when Xorbit is not busy) does the resetting of the readbacks if there has been a change. A routine is also established that is called when activity occurs on a UNIX socket attached to channel access. This routine gives channel access a chance to calculate. In addition, connection monitors and associated Xorbit callback routines are set up in a similar fashion to handle lost connections and other channel access problems. All of these are handled by the X event loop, along with the user mouse and keyboard activity, in the order received.

Because of the necessity to convert between current and magnetic field, the setting of setpoints and reading of readbacks is more complicated than has been indicated. The details when the conversion is included are discussed in Sections V and VI.

Since all of the Xorbit PVs and all of the real machine PVs are simultaneously available from the control system, it is straightforward to write a program that reads the real values and puts them into the Xorbit setpoints, causing the orbit in Xorbit to be that predicted by the current machine settings. One of the EPICS menu items in Xorbit runs programs like this. The result is that the current machine values are easily "dumped" into Xorbit. One could go the other way, but this has not been implemented because of its potential for accidental misuse. The values that are read in this procedure are the magnet setpoints, rather than the readbacks, because this is a more stable and reproducible procedure.

IV. XORBIT DATABASE

The Xorbit database has a PV corresponding to each BPM readback and each magnet readback and setpoint in each of the APS rings and beamlines (with the current exception of the BTS and BTX beamlines). These process variables have the same names as the ones in the real machines, except that the prefix "Xorbit:" is attached to each one. For example, S1A:H1:CurrentAO is the name of a real magnet setpoint, and Xorbit:S1A:H1:CurrentAO is the name of the corresponding Xorbit setpoint. Thus, the applications programs, like ADT, that have an Xorbit option only have to add this prefix to their PVs to deal with Xorbit rather than the real world.

In addition, there are two process variables to handle the current-to-field-strength conversion for each magnet, one for going from Xorbit and one for going to Xorbit. Xorbit and similar codes actually use the parameter KnL rather than the field strength directly. Consequently, the conversion is made between KnL and the power supply current I. KnL is defined to be BnL/Bρ, where B is the main vertical dipole field in the device, BnL is its n*th* partial derivative with respect to x (n is 0 for a dipole, 1 for a quadrupole, 2 for a sextupole, etc.) integrated along the longitudinal axis, and ρ is the gyroradius. The quantity Bρ is related to the energy by Bρ[T·m] = 3.34E[GeV]. The KnL and hence the calculations in Xorbit do not depend on the energy. There needs to be one more PV for each ring and beamline to hold the value of Bρ. In addition, there are also two PVs for each ring that hold the readback values for the tunes.

The total number of process variables necessary to simulate all of the APS rings and beamlines is about 7500. These comprise the Xorbit database and are stored in one input/output controller (IOC) with a 16MB, 25MHz Motorola 68040 board.

The BPM PVs use a standard analog input (AI) record. The magnet PVs are more complicated because of the conversion: The original implementation, which is still used for the booster, is to use a standard calculation (CALC) record for the conversion as shown in Fig. 2. The input for the CALC record is described in the next section. In order to get more accuracy the CALC record has been replaced by a special current/KnL conversion (IK) record, described below, in the other cases. Xorbit writes to the VAL fields in the xxx:KAI and xxx:KAO records (where xxx stands for any particular magnet) and monitors xxx:KAO. The conversions to the xxx:CurrentAI and from the xxx:CurrentAO happen automatically because of the forward links (FLNKs).

For either type of record, the setpoints are set equal to the readbacks by setting the OMSL field in the AO record from **supervisory** to **closed_loop**, processing the record, then changing the OMSL field back. This causes the record to get its value from the DOL input, which is attached to the readback.

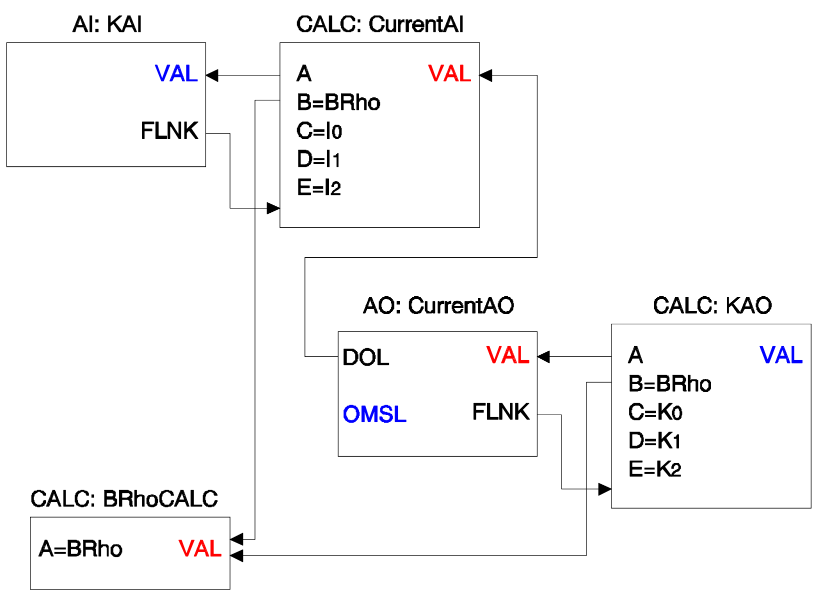

Fig. 2. The record structure for a magnet in the Xorbit database.

V. CURRENT VS. FIELD STRENGTH

The relationship between the current and the resulting field strength for each magnet in the APS was measured [11] before it was installed. From these data, files were created [12], one for each magnet, with sets of values for BnL vs. I. These data are called excitation curves, though tables would be more appropriate. It is the excitation-curve data in these files that is used by Xorbit.

The earliest attempts to do the conversion just used a proportionality constant based on the single point corresponding to the design reference value for BnL and the corresponding current. When the magnets were set differently from the reference value (as they often were), it quickly became apparent this was not accurate enough. The next approach was to make quadratic fits of the excitation-curve data. The fits were made to determine I as a function of BnL, and the fits to BnL as a function of I were inverted from these so the inversion process is 1:1 and hence consistent. This gave a satisfactory conversion for machines up to the storage ring; that is, the orbit dumped from the machine was very close to the predicted orbit as were the beta functions and other parameters, including the tunes, which are sensitive to magnet, especially quadrupole, changes.

The excitation files were not made for the booster. The reason is that the booster is time dependent and does not have or need the precision of the other machines. The fit coefficients used for the booster correspond to a linear fit passing through zero and the reference value.

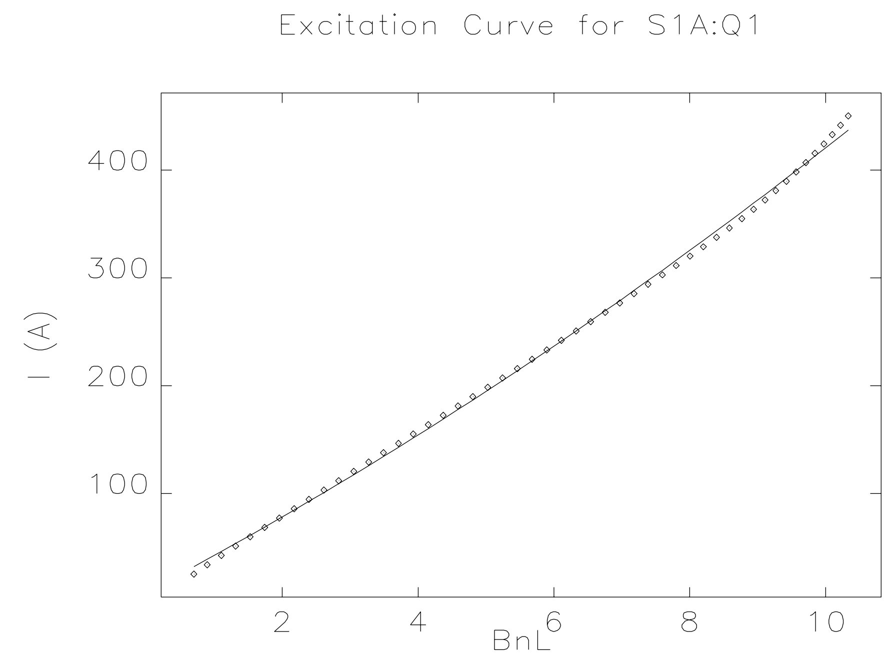

Fig. 3. Excitation curve points for a storage ring quadrupole and the quadratic fit to them.

For the storage ring these fits, though they are reasonably close as seen in Fig. 3, do not give accurate values for the tunes. In fact, the integer parts of the tunes are off by about two integers and the fractional parts are unrelated to the real values. (In practice the integer part of the tune is determined separately from the fractional part, and the fractional part can be measured to three figures or more.) The reason for agreement elsewhere and not in the storage ring is that (1) there are many more magnets in the storage ring and (2) the quadrupoles (the main influence on the tune) are stronger. This is verified by the fact that the empirical values could be reproduced by small, systematic changes of the quadrupoles inside of Xorbit. It seems apparent that to get sufficient accuracy in the conversion, it is necessary to do a table lookup on the values in the excitation curves. Since the excitation curves have on the order of up to 50 points, it is not possible to do this with a CALC record. Consequently, a new record, the IK record, was developed to do the conversion using table lookups.

VI. IK RECORD

In order to do the conversions using a table lookup, it is necessary to have a record that includes fields for the I and BnL arrays, which may contain a variable number of points. The IK record which was developed is similar to the standard CALC record with the addition of the arrays. The record structure for the Xorbit database looks as shown in Fig. 4.

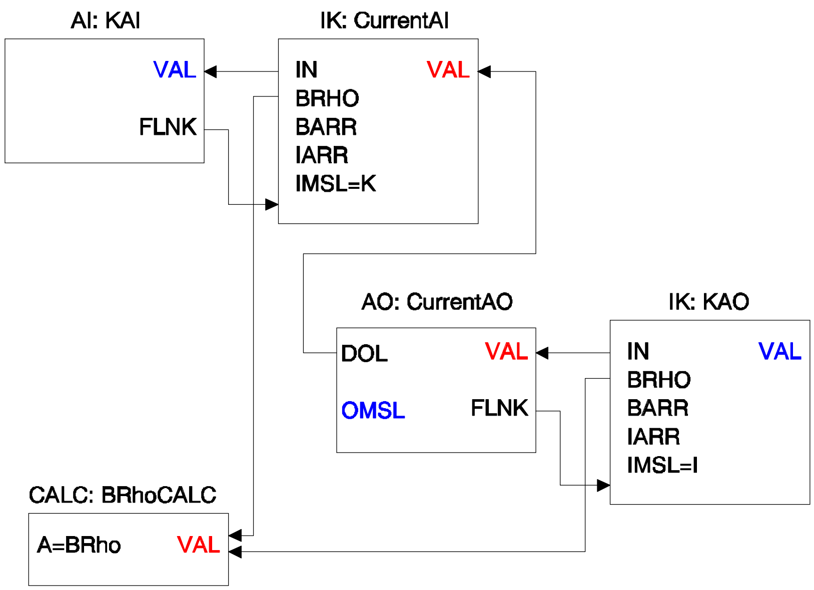

Fig. 4. The record structure using the IK record.

The input field is INP or IN (corresponding to, say, INPA and A in a CALC record). This may be either a KnL value or an I value, depending on whether the IMSL (input mode select) field is **I** (convert I to Knl) or **K** (convert Knl to I). The BRHO field is an additional input. The arrays are held in BARR and IARR. There is currently no way in EPICS to specify initial values for array fields. Therefore, the values must be stored in the arrays during or after record initialization. For the IK record there are two options. There are two string fields, PATH and FILE, that allow you to specify a file from which to load the data. If FILE is blank, then the record looks for a routine called getIKArrayValues(type), where type is any string. Users of this record can use this function to load the arrays using their own method. There is also a string field, TYPE, in the record that can be used to hold the string.

For the Xorbit database the excitation files are processed into C programs where the data are hard coded in static arrays along with the names of the magnets. The C programs create hash tables for the data and getIKArrayValues gets the data from the hash tables using the magnet names, which were stored in the TYPE field.

REFERENCES

1. H. Grote and F.C. Iselin, "The MAD Program," CERN Report No. CERN/SL/90-13(AP), 1991.
2. K. Evans, Jr., “Xorbit -- An X-Windows Accelerator Simulation,” Proc. Computational Accelerator Physics Conf. 1993, Pleasanton, CA, February 22-26, 1993, p. 450.
3. K. Evans, Jr., "Orbitscreen Reference Manual," World Wide Web URL: http://www.aps.anl.gov/asd/oag/oagSoftware.html.
4. K. Evans, Jr., "Array Display Tool ADT Reference Manual," World Wide Web URL: http://www.aps.anl.gov/asd/oag/oagSoftware.html.

5 K. Evans, Jr., "Corbit Reference Manual," World Wide Web URL: http://www.aps.anl.gov/asd/oag/oagSoftware.html.

6. S. Wolfram et. al., Mathematica (Addison-Wesley, Redwood City, CA,1991).
7. Computer code *Wingz* (Informix Software, Inc., Lenexa, KS).
8. Computer code *Excel* (Microsoft Corp., Redmond, WA).
9. B-C. Cha, "WingZ/CaWingz," World Wide Web URL: http://www.aps.anl.gov/asd/controls/software_tools.html.
10. B-C. Cha and R. T. Daly, "CaMath User's Guide," World Wide Web URL: http://www.aps.anl.gov/asd/controls/software_tools.html.
11. S. H. Kim, et al., "Statistical Analyses of the Magnet Data for the APS Storage Ring Magnets," Proc. 1995 Particle Accelelerator Conf., Houston, TX, May 1-5, 1995 (to be published).
12. L. Emery, private communication.