

# Modeling in Control of the Advanced Light Source \*

J. Bengtsson, E. Forest, H. Nishimura, and L. Schachinger  
Lawrence Berkeley Laboratory  
Berkeley, CA 94720

## Abstract

A software system for control of accelerator physics parameters of the Advanced Light Source (ALS) is being designed and implemented at LBL. Some of the parameters we wish to control are tunes, chromaticities, and closed orbit distortions as well as linear lattice distortions and, possibly, amplitude- and momentum-dependent tune shifts. In all our applications, the goal is to allow the user to adjust physics parameters of the machine, instead of turning knobs that control magnets directly. This control will take place via a highly graphical user interface, with both a model appropriate to the application and any correction algorithm running alongside as separate processes. Many of these applications will run on a Unix workstation, separate from the controls system, but communicating with the hardware database via Remote Procedure Calls (RPCs).

## Introduction

The ALS is a 1-2 GeV third-generation synchrotron light source under construction at Lawrence Berkeley Lab. The machine consists of an electron gun, followed by a 50 MeV linac, a 1.5 GeV booster ring, and a storage ring with 10 straight sections for insertion devices. The demands placed by the booster on correction algorithms are much less stringent than those of the storage ring — the booster orbit will probably only have to be corrected to 1mm *rms*, and chromaticity correction might not be needed. The storage ring, on the other hand, will require a residual closed orbit distortion of the order of 0.15mm *rms*. This is due to the fact that linear lattice distortions caused by feed-down from orbit offsets in the sextupoles break the symmetry of the lattice, and reduce the dynamic aperture. Also, fast (10-20Hz) correction of the orbit in the insertion devices is needed to maintain beam stability for users. Good chromaticity control is necessary as well. The distortion of the linear lattice by insertion devices also needs correction.

At present we have accelerated beam to 50 MeV in the linac, and are preparing for the imminent injection of the first beam into the booster. Storage ring commissioning is scheduled to begin in 1992.

\*Work supported by Director, Office of Energy Research, Office of Basic Energy Sciences, Materials Sciences Division, U.S. Department of Energy under Contract Number DE-AC03-76SF00098.

We have developed tools for linac and transfer line commissioning, two of which are mentioned in the next section. We describe two tools being developed for booster commissioning; one for orbit correction and another for adjusting the tunes. One storage ring application that corrects for linear lattice distortions caused by insertion devices is also included. First we describe our present vision of an ideal system, then report on a sample of applications.

## Planned System Structure

Each application will be composed of four parts — the graphical interface, the model of the machine, the correction algorithm, and the hardware database. These four pieces are seen as independent, with well-defined interfaces. In the system as it will eventually be implemented, each will be a separate process. The communication mechanism between them will probably be RPCs, perhaps coordinated by a sequencing language such as Glish [1].

Our current graphical interfaces are written for the X window system in C++ using InterViews[2], a freely available toolkit. In our interfaces, we want to display any relevant physics that will aid in adjusting a parameter; for example, we display resonance lines directly on an application that shows a measurement of the tunes and also allows tune adjustment (see Figure 1).

The model is a description of the accelerator that varies from application to application. In the case of orbit correction, our model is the ideal betas and phases at the beam position monitors (BPM's) and correctors. In setting the tune, the model is a map from tune changes to quadrupole strength changes, or a collection of maps, one for each area of the tune plane. Our models will be simple, though often created by doing studies with the full-blown simulation codes Tracy2[3] or COSY Exterminator[4], both based on the Pascal-S compiler/interpreter system[5] and therefore developed in Pascal. Tracy2 is a tracking code using 4x5 matrices and thin kicks, or alternatively, 2nd or 4th order symplectic integrators. COSY Exterminator generates maps, and is basically a generalization of a matrix code to arbitrary order. It is a cleaner, more powerful implementation of the idea behind COSY-Infinity[6]. Part of its power comes from the ability to write input files in Pascal, rather than COSY-Infinity's special-purpose, but

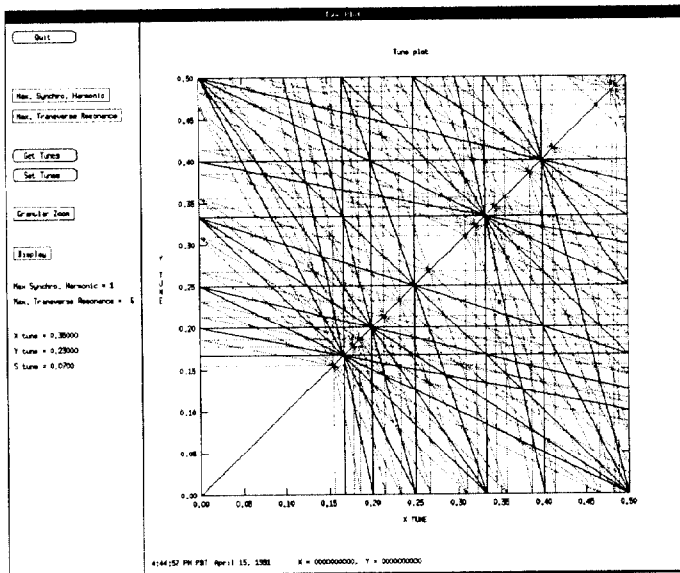


Figure 1: Graphical interface to the tune measurement and correction algorithm.

inappropriate input language. COSY Exterminator has been used to calculate maps to allow independent focusing of the beam in X and Y at waists in a transport line. For another application, MACSYMA[7] was used to simplify the steering in the solenoidal fields downstream of the electron gun by decoupling the effect of dipole correctors. This model also contains the solenoidal field, calculated by POISSON[8] from the solenoid currents.

The correction algorithm uses input from the model and the accelerator to compute new values of accelerator parameters, resulting in a change in a physics parameter of the machine. An example of a correction algorithm is the code which uses ideal betas and phases at the BPM's and correctors from the model and readings from the BPM's to compute new corrector settings, improving the orbit.

The hardware database is described in detail elsewhere [9], but for our purposes it can be viewed as a collection of readable and writable "values" of devices in the machine.

## Booster Applications

A simple application is the one that sets the tunes. Our interface is as shown in Figure 1. The model is a map produced by COSY Exterminator, and the correction algorithm simply applies the model to the requested tune changes, producing new quadrupole settings. The correction algorithm iterates, applying the model until the measured tunes equal the requested tunes. The desired tune is selected by pointing and clicking with a mouse, or by typing. The maximum order of resonance lines can be selected using a menu item on the left. Resonance lines identify themselves (e.g.,  $\nu_x + 2\nu_y = 3$ ) when pointed at.

The orbit correction application we will use for the booster is based on earlier work done at the SSC Central

Design Group [10]. The correction algorithm uses overlapping local bumps of three correctors, and can correct the orbit both on the first turn and for circulating beam. Minimization is done using the simplex method, so each corrector and monitor may be assigned a weight which need not be a continuous function. This is used to prevent the program from setting the corrector strengths above the maximum value — corrector weights are a step function which becomes very large for strengths above the maximum. The goal orbit can be any arbitrary trajectory.

The interface is highly graphical (see Figure 2). The user can zoom in on either the ring on the right, or the linear display of BPM readings on the left. Zooming in, then hitting the "correct" button causes just the zoomed region to be corrected. The individual monitors and BPM's are displayed under the plot on the left, and pointing at one with a mouse causes its value to be displayed.

Originally this algorithm was implemented in the context of a simulation; we have replaced the modeling program which generated BPM readings with RPC calls to the ALS hardware database. This turned out to be straightforward, due to the modular design of the original program.

## Storage Ring Applications

Due to lack of space for dedicated correction quads, local correction of insertion devices is not possible for  $B^2L$  greater than approximately  $10T^2m$ , which is not enough to correct the strongest insertion devices. Therefore, a global correction scheme has been worked out using only the maps in Floquet space between sextupoles[11]. By individually retuning either a subset or all of the existing 48 quadrupoles, it is possible to restore the symmetry of the bare linear lattice. We have used Tracy2 and the simplex method for this 48-dimensional non-linear optimization. This almost restores the dynamical aperture.

A complication in the implementation of this correction scheme is the fact that some of the insertion devices will be scanning. This makes it impossible to do the non-linear optimization in real-time. Therefore, a linear approximation to the relation between the insertion device and quadrupole settings is derived for each type of insertion device. This allows superposition of the contributions from several insertion devices, which is a good approximation for the cases we have studied. Tables of quad strengths as a function of insertion device setting will be used by the correction algorithm, which then only needs to do the superposition. At present, there is no user interface for this algorithm.

To improve the Touschek lifetime, we are working on an algorithm to reduce the amplitude-dependent tune shifts and non-linear chromaticity by retuning all 48 sextupoles in the ring. Analytical expressions for these tune shifts, which depend on properties of the bare lattice that have been calculated using DA techniques[12], have been calculated using MACSYMA[7]. These expressions could be used on line to adjust the sextupole strengths.

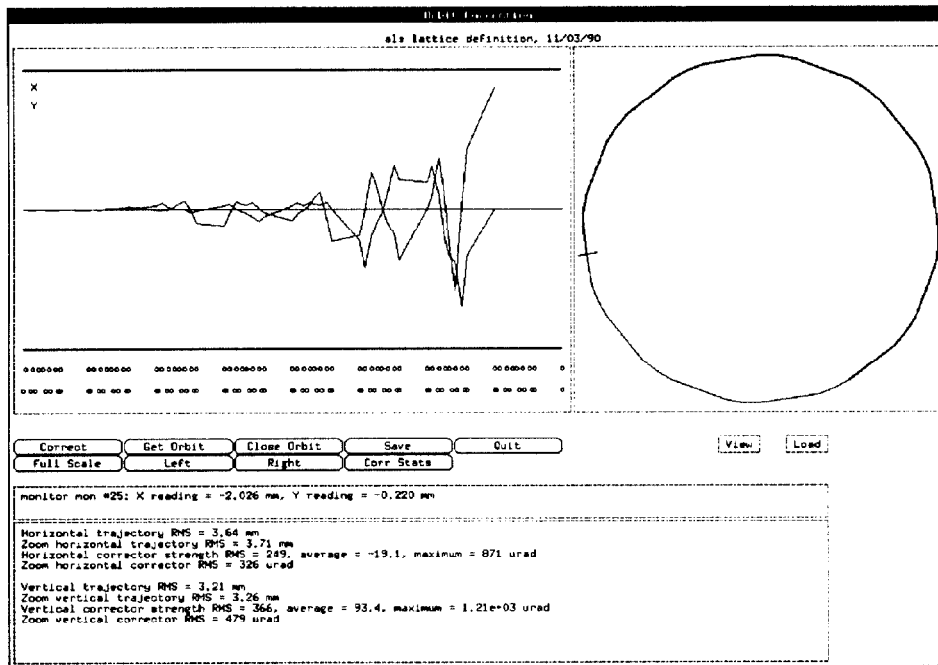


Figure 2: Graphical interface to the orbit correction algorithm. The bird's-eye view showing the entire ring is on the right; the window on the left shows the area currently zoomed-in on.

## Future Work

We hope that in the process of commissioning the booster we will learn much that will be of use in developing tools for the storage ring. For example, what sort of interfaces are really convenient to use? How do we want to view and manipulate the machine state? How do the components of the system function together? What other tools or interfaces might be useful?

We are also exploring the potential of other software development tools and languages for possible use in storage ring commissioning applications. For instance, a prototype model is being developed using Eiffel, and with it, a class library for accelerators. We are exploring other graphics toolkits such as *glistk*[13] which is being developed at SSC.

## Acknowledgments

Vern Paxson created the interface for orbit correction while at SSC/CDG. The interface for setting the tune is an outgrowth of one he and one of us designed for studying tracking. Many thanks to Chris Timossi for implementing the RPC calls to the hardware database and for supporting our efforts in general.

## References

- [1] V. Paxson *et al.*, "A Language, Server and C++ Class Library for Event Sequencing," NIM A293 (1990), p. 356-362.
- [2] M. Linton *et al.*, "Composing User Interfaces with Interviews," IEEE Computer, Feb. 1989, pp. 8-22.
- [3] J. Bengtsson and H. Nishimura, unpublished.
- [4] J. Bengtsson, "Applications Based on Tracy and COSY Exterminator," LBL LSAP-116, 1991.
- [5] N. Wirth, "Pascal-S: A Subset and Its Implementation, Pascal - The Language and Its Implementation," Ed. D. W. Barron, pp. 199-260, Wiley 1981.
- [6] Cosy-Infinity Reference Manual, LBL-28881 (1990).
- [7] "MACSYMA Reference Manual", Symbolics Inc. (1983).
- [8] C. Kim, private communication.
- [9] S. Magyary *et al.*, "The Advanced Light Source Control System," NIM A293 (1990), p. 36-43.
- [10] V. Paxson, *et al.*, "Interactive First Turn and Global Closed Orbit Correction in the SSC," Proceedings of the European Particle Accelerator Conference, Rome, August 1988.
- [11] J. Bengtsson and E. Forest, "Global Matching of the Normalized Ring," LBL ESG Tech Note 125.
- [12] M. Berz, "Differential Algebraic Description of Beam Dynamics to Very High Orders," Particle Accelerators 24, 109 (1989).
- [13] M. A. Kane *et al.*, The GLISTK Manual, unpublished.