Interactive First Turn and Global Closed Orbit Correction in the SSC

Vern Paxson, Steve Peggs, and Lindsay Schachinger
SSC Central Design Group*
c/o Lawrence Berkeley Laboratory
Berkeley, CA 94720

We describe an algorithm for correcting the orbit of the SSC using closed bumps; an implementation of the algorithm utilizing a highly graphical user-interface, designed with portability in mind; and results from using the algorithm to correct lattices with simulated errors.

Introduction

An early step in storing useful beam in the SSC is achieving a first-turn orbit. This is a non-trivial problem since the quadrupole misalignments are such that the *rms* orbit deviation, without correction, would be 16 mm in the arcs, while the beam pipe is 16.5 mm in radius. The procedure is to first correct the trajectory so that the beam goes around the entire circumference of the ring, then to establish circulating beam, and finally to fine-tune the orbit with a global orbit correction algorithm. The task is complicated by the monitors' limits of resolution and by displacement errors in the monitor positions.

We address three aspects of a program we have developed for simulating orbit correction: the algorithm used, the graphical interface to the program, and the results of the simulation.

Orbit correction is essentially a minimization problem, in which one computes the set of corrector strengths which will minimize the deviation of the actual trajectory from the desired trajectory. The algorithm described below breaks down the problem of global orbit correction into a series of overlapping local orbit correction problems. Each of these can be parameterized so that to solve them involves minimizing a function of only one independent variable, which can be done in straight-forward fashion using a simplex method.

Software for exploring and manipulating models, such as studying the behavior of the orbit correction algorithm and its simulated effects on the SSC, benefits greatly from a highly graphical user-interface. [1] With a visual way to interact with the model the user can much more readily build intuition and understanding as to how the model operates. We endeavoured to create such a user-interface to the orbit correction algorithm, and also to use this as an opportunity to explore ways to manageably yet effectively represent a model as complex as the SSC lattice. Furthermore, we took pains to design the user-interface software such that it is reusable, both by being modular to the extent that many components are generic and can thus be used in other interfaces, and by adhering to emerging standards so that the software can be readily moved to other hardware.

There are several motivations for implementing the orbit correction algorithm: to determine design requirements for the SSC's dipole correctors, to help with the design of corrector configurations for the interaction regions, to provide machines with corrected orbits for aperture studies, and to provide a more realistic machine model for studies of the correction of other errors. [2] The eventual goal is to realistically simulate injection into the SSC with all known errors. We found that the algorithm is able to correct orbits in the presence of expected machine errors [3] to a satisfactory degree, and below we summarize the corresponding design requirements.

* Operated by the Universities Research Association, Inc. for the U. S. Department of Energy.

Algorithm

We use an orbit-correction algorithm which is based on closed bumps. [4] Bumps are made using three correctors as shown in Fig. 1. Each bump can be parameterized by a, its height at the central corrector. To correct a region of the accelerator one finds the set of bumps which minimize the difference between the actual orbit and the desired orbit, given constraints such as physical aperture size and maximum corrector strengths.

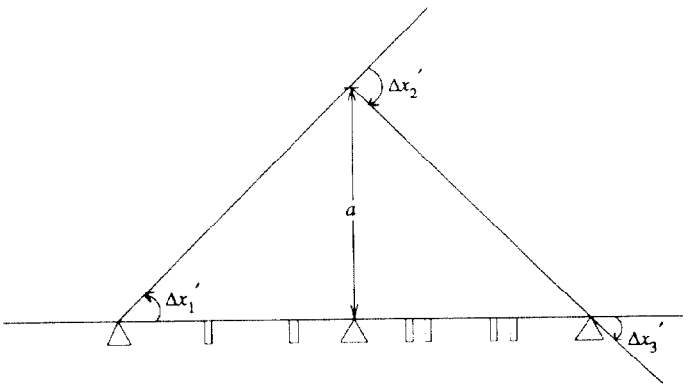

Figure 1. Structure of a closed bump. The triangles indicate correctors, the thin rectangles are monitors.

In general, the effect on the beam at a point s due to a kick $\Delta x_i'$ at i is given by

$$\begin{aligned}\Delta x(s) &= \sqrt{\beta_i \beta_s}(\sin\Delta\phi)\Delta x_i' \\ \Delta x'(s) &= \sqrt{\beta_s/\beta_i}(\cos\Delta\phi - \alpha\sin\Delta\phi)\Delta x_i'\end{aligned} \qquad (1$$

where $\Delta\phi = \phi(s) - \phi(i)$.

If the bumps are *closed*, that is, the effects of the bump are wholly localized to the area between the first and third correctors then

$$\sum_{i=1,3} \sqrt{\beta_i}\Delta x_i' \begin{pmatrix} \sin\Delta\phi \\ \cos\Delta\phi \end{pmatrix} = 0$$

for s outside of the bump. Then the law of sines gives equation and using the law of sines yields

$$\frac{\sqrt{\beta_i}}{\sin(\phi_j - \phi_k)}\Delta x_i' = \text{constant} \qquad (2$$

where i, j, k are (1, 2, 3), (2, 3, 1), or (3, 1, 2).

We now can use (1) and (2) to express the corrector strengths in terms of a, giving

$$\Delta x_1' = \frac{a}{\sqrt{\beta_1\beta_2}\sin(\phi_2-\phi_1)}$$

$$\Delta x_2' = \frac{a\sin(\phi_1-\phi_3)}{\beta_2\sin(\phi_3-\phi_2)\sin(\phi_2-\phi_1)} \qquad (3)$$

$$\Delta x_3' = \frac{a}{\sqrt{\beta_2\beta_3}\sin(\phi_3-\phi_2)}$$

Since (1) gives the additional displacement for a monitor inside a bump due to an additional kick at one of the correctors, and (3) expresses the kicks in terms of a, by combining them we can introduce a *penalty function*, with a as the sole independent variable, which quantifies how close the beam trajectory within a bump is to the ideal trajectory:

$$P(a) = \sum_{b=\mathrm{BPM}} w_b(x_b - x_b^*)^2 + \sum_{c=\mathrm{corr.}} w_c P_c(x_c') \qquad (4)$$

where w_b is a weight associated with each monitor, w_c a weight associated with each corrector, x_b is the present monitor reading plus the change in position due to the bump, x_b^* is the desired reading at the monitor, and P_c is a function which penalizes excessive corrector settings.

Since $P(a)$ is a function of only one independent variable, it can easily be minimized using a one-dimensional simplex algorithm.[5] With this simplicity comes the strong advantage that $P(a)$ needn't have a restricted form such as being quadratic, or even being particularly well-behaved (for example, $P(a)$ can be discontinuous). In particular, the penalty function for corrector strength can be

$$P_c(x_c') = \begin{cases} k|x_c'|, & \text{if } |x_c'| > x_c'^{\max}; \\ 0, & \text{otherwise.} \end{cases}$$

for some large constant k. This function is quite cheap to compute, yet will prevent correctors from being set beyond their maximum strengths.

The region of the accelerator to be corrected is divided into overlapping closed bumps, and then the penalty function for each bump is minimized in turn, incorporating the effects of overlapping bumps in the x_b of (4). This iteration over the set of bumps is repeated until the change in the *global* penalty function, $G(a_1,\ldots,a_n)$ equal to the sum of the $P(a_i)$, is less than some small ϵ. At this point, since we minimize the $P(a_i)$ separately, we have $\partial G(a_i)/\partial a_i = 0$ for all i, so we have arrived at a local minimum.

The flexibility of this algorithm makes it quite appealing—it can correct an arbitrary region of the accelerator, both first-turn and closed orbit; since the solution generated is composed entirely of closed bumps, its effects are confined to the particular region of interest; the goal orbit can be any arbitrary trajectory; the penalty function can have non-quadratic forms, incorporating effects such as maximum corrector strength and limits on monitor readings; both monitors and correctors can be weighted to emphasize or remove particular elements; it is stable in regions which have no BPM's; and the algorithm is straight-forward to implement in software.

Graphics Interface

We implemented a highly graphical user-interface for the orbit correction algorithm, both to facilitate use of the algorithm and to explore ways of effectively displaying information about a machine as large as the SSC.

The model we adopted was that of a *bird's eye view*, in which a top-view of the entire ring is shown in one window, and a selected, zoomed-in region of the ring shown in greater detail in another window (Fig. 2). Navigation (zooming in or out, shifting the region of interest) can be done in either window, with both windows updating to show the effects. The zoomed-in view shows the horizontal and vertical beam positions at each monitor (along with, optionally, the positions predicted by the orbit correction algorithm), and along the bottom appear blips indicating the position of individual correctors and monitors. These can be selected using the mouse to produce information on the element.

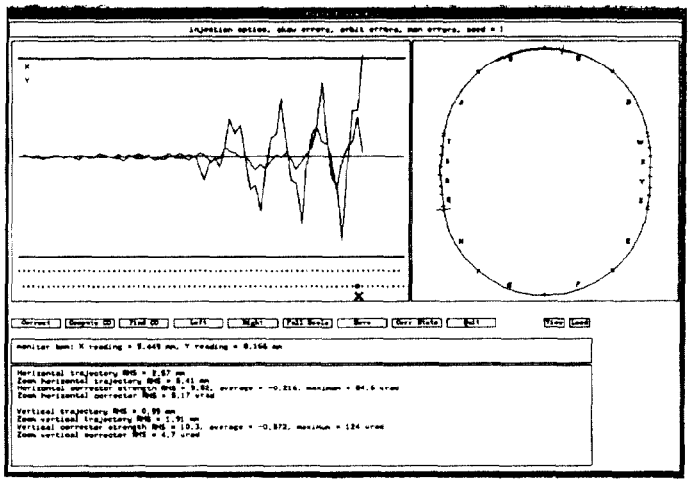

Figure 2. Graphical interface to the orbit correction algorithm. The bird's-eye view showing the entire ring is on the right; the window on the left shows the area currently zoomed-in on.

The basic operation is to zoom in on the region to be corrected, correct the region using the algorithm, reinject, and observe the resulting progress. In addition, the program can find the closed orbit, and allows for ready archiving of both any stage of the orbit correction process and statistics summarizing the distribution of corrector strengths that were necessary to correct the orbit. This is not an overwhelming amount of functionality; a key point here, however, is that due to modular design, adding further functionality (e.g., turning off bad monitors or correctors; diagnosing faults) is a straight-forward task. The difficult work—the basic design of the graphics—is in place.

In implementing the orbit correction algorithm and its graphics interface, we endeavoured to follow the tenets of sound, portable program design.[6] The algorithm itself is written in standard-compliant (with the exception of **include** statements) FORTRAN-77. While the lattice information and machine functions it needs happen to be provided by Teapot[7], the implementation of the algorithm is wholly independent of the modeling program: it has no knowledge of any Teapot common blocks, and is isolated from the format of Teapot machine files.

The graphical interface is written using the industry-standard X Window System,[8] which runs on a wide variety of computers

(DEC, Apollo, Sun, Masscomp, Apple, IBM PC, etc.; we used Sun workstations). We introduced a further level of graphics-independence by isolating the X-specific code in a package of library routines. The orbit correction user-interface is written solely in terms of these routines, and thus is insulated from changes in the underlying window system.

The only coupling between the graphics and the orbit correction algorithm is a set of interface routines which the graphics code calls to gather information to be displayed (such as element types, positions, and strengths) and to run the orbit correction algorithm. This design keeps the algorithm fully separate from the particular graphics interface used to interact with and animate it, and allows either to be changed without impacting the other.

Simulation Results

The algorithm was used to correct the SSC 90° injection-optics lattice. Five random number seeds were used to generate different sets of field strength and positional errors. The dipoles were assigned errors with standard deviations $\sigma_{a0} = 5.9$, $\sigma_{b0} = 8.5$, $\sigma_{a1} = .72$, $\sigma_{b1} = .72$, $\sigma_{a2} = .64$, $\sigma_{b2} = .40$, $\sigma_\theta = .6$ mrad, and $\sigma_{x,y} = 1$ mm. Multipole errors are in units of parts per 10^{-4} at 1 cm. The quadrupoles were rotated with $\sigma_\theta = .5$ mrad and misaligned with $\sigma_{x,y} = .5$ mm. The distributions used were Gaussian, truncated at four standard deviations. In the simulation the monitors had a resolution σ of 100 microns, with errors in the readings not exceeding 10σ. The beam was injected on-axis and with no energy errors.

The bulk of the study of the algorithm's performance is limited to the arcs, as the corrector and monitor configuration in the IR's needs further study. In the arcs, there is a monitor positioned 0.1 meters downstream from each quadrupole, and 3.285 meters downstream from the monitor is either a horizontal or vertical corrector (for horizontal or vertical focussing quadrupoles, respectively). The monitors are misaligned with $\sigma = 0.1$ mm with respect to the quadrupoles.

The correction procedure was to steer the beam completely around the ring by progressively correcting the trajectory. The program then automatically found the closed orbit using tracking and binary search. The closed orbit was then globally corrected, the resultant closed orbit found again, and these two steps repeated until the *rms* orbit errors were less than .4 mm (this never required more than two subsequent iterations).

Seed	x_D	y_D	x_B	y_B
1	.36	.37	.28	.29
2	.35	.36	.30	.28
3	.35	.33	.29	.30
4	.37	.39	.29	.32
5	.37	.36	.28	.27

Table 1: RMS orbit deviations in mm. x_D is with respect to the design orbit, while x_B is with respect to the center of the BPM.

Seed	$X_{\rm rms}$	$Y_{\rm rms}$	$X_{\rm max}$	$Y_{\rm max}$
1	8.45	8.53	20.63	19.95
2	8.11	8.29	19.15	18.84
3	7.68	8.06	19.99	18.14
4	7.97	8.49	19.31	19.17
5	8.50	8.16	19.41	21.59

Table 2: Corrector strengths needed for final, corrected orbit, in μrad.

Table 1 shows the *rms* of the monitor readings in each plane of the final closed, corrected orbit. Two values are given for each plane, one with respect to the design orbit and one with respect to the center of the monitors. The values in the table include the monitor readings in the IR's, which are not substantially different from those in the arcs.

Table 2 summarizes the *rms* and maximum corrector strengths needed in the arcs for the final corrected orbit. Slightly higher values were needed during the course of correcting the orbit. For the five seeds the greatest corrector strength ever needed was 26.38 μrad in the X plane and 26.20 μrad in the Y plane. The design maximum corrector strength is 60 μrad at 20 TeV.

Peggs and Chao[9] have calculated the *rms* orbit deviations and corrector strengths for the 60° SSC lattice, for the linear machine with no coupling. A similar calculation for the 90° lattice with the errors used for this study yields an *rms* orbit deviation with respect to the center of the beam position monitors of 0.24 mm in both planes and an *rms* horizontal and vertical corrector strength of 7.6μrad, in good agreement with the results.

References

1. V. Paxson, V. Jacobson, E. Theil, M. Lee, and S. Clearwater, "A Scientific Workstation Operator Interface for Accelerator Control," 1987 IEEE PAC, p. 556, Washington, D. C.
2. L. Schachinger, "Interactive Global Decoupling of the SSC Injection Lattice," these proceedings.
3. SSC Central Design Group, "Superconducting Super Collider Conceptual Design," SSC-SR-2020, March 1986.
4. S. G. Peggs, "Some Aspects of Machine Physics in the Cornell Electron Storage Ring," Ph.D. Thesis, Cornell University, 1981.
5. W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, **Numerical Recipes**, Cambridge University Press, 1986.
6. V. Paxson and E. Theil, "Towards Portability in Model-Based Control Software," LBL-24723, November 1987.
7. L. Schachinger and R. Talman, "Teapot: A Thin-element Accelerator Program for Optics and Tracking," **Particle Accelerators**, Vol. 22, 1987.
8. R. Scheifler, J. Gettys, "The X Window System," **ACM Transactions on Graphics**, No. 63, 1986.
9. S. Peggs and A. Chao, "An Updated Look at Long Arc Orbit Correction," SSC-N-96, 1985.