SLAC–TN–91–9
SSCL–N–760
July 1991
(TN)

MODELING LONG–TERM COLLIDER PERFORMANCE

D. RITSON

*Stanford Linear Accelerator Center**
Stanford University, Stanford, CA 94309

and

Superconducting Super Collider Laboratory◇
2550 Beckleymeade Ave., Dallas, TX 75237

ABSTRACT

A model for the SSC arcs is described with multipole lattice field errors agglomerated into 32 lattice points, and with first order lattice errors and modulation provided by discrete transfer elements. Numerical solutions for long term dynamic aperture studies are obtained by multipole kick-drift tracking. The CPU time required to track through one turn is minimal, and comparable to that required to implement a one-turn fifth-order Taylor series map. Comparisons with previous tracking results using a fine grained representation of the lattice are made, and found to be satisfactory.

The effects of tune modulation are studied and can substantially degrade long-term dynamic aperture. The effects of small relativistic momentum corrections, usually neglected for the large momenta at the SSC, are shown to have negligible influence on tracking results.

*Work supported by Department of Energy contract DE–AC03–76SF00515.

◇Operated by Universities Research Association, Inc., for the U.S. Department of Energy, under contract DE–AC02–89ER40486.

INTRODUCTION

Tracking simulation provides an essential tool for the economical and safe design of accelerators, and for anticipating their future performance. Modern colliders such as the SSC are very complex devices made up of many thousands of magnetic elements with substantial non-linear field errors. The SSC will inject into the main collider at 2 TeV for a period of one hour, corresponding to $\sim 10^7$ turns. With present computer technology, tracking for millions of turns at the SSC, while CPU intensive, is possible and has been used extensively via a program SSCTRK.[1] SSCTRK was designed to fully exploit the computing power of vector processing machines such as the CRAY[2]

Under collision conditions at 20 TeV, the beam is expected to be stored for periods of ~ 1 day, corresponding $\sim 10^8$ turns. The interaction physics is dominated by the beam-beam interactions, both head-on and long range, and by the IR quads that are located in very high β regions. The effect of the non-linearities in the arc guide fields is small. If only the IR quad effects and beam-beam effects were to be included in SSCTRK, tracking for 10^8 turns is possible. However, including the arc elements in the SSCTRK program increases the required CPU times by one to two orders in magnitude.

Speeding up the tracking through the arcs by the use of one-turn maps has been suggested by many authors. The expectation is that a map, applied once a turn, is much faster than tracking through several thousand thin non-linear kick elements. In fact, this is not correct for mapping with one-turn maps constructed from truncated Taylor series. An adequate description of tracking behavior for the full SSC dynamic aperture requires inclusion of terms up to eleventh order.[3] To implement numerically an eleventh-order map applied to five coordinates requires at minimum 20,000 multiplies and 20,000 additions. To achieve such coding economy requires implicit addressing, which adds substantial overhead to the CPU requirements. SSTRK simulations that include multipoles up to ninth order require roughly 50,000 multiplies and additions. Therefore such one-turn mapping is, at best, marginally better.

To provide speed we used a fully symplectic model of the SSC arc lattice, obtained by conglomerating errors onto a small number of lattice points. The model

includes synchrotron motion, modulation effects, and the effects arising from only partial cancellations of first-order errors. It appears to provide a good representation of SSC tracking behavior over the full dynamic aperture. The model can track 3×10^8 turns per day on an IBM RISC workstation, and could be run on CRAYS or parallel processing devices at least one to two orders of magnitude faster.

We describe below details of the simulation model and some results obtained from its use with injection optics. A later note will discuss the effects arising from the inclusion of high-β IR quad and beam-beam interaction effects.

THE SIMULATION MODEL

Error levels for position and field errors for a projected machine are defined in terms of systematic and random errors. The random errors are generally assumed to have a Gaussian distribution (sometimes with a 2 to 3 σ cutoff). Because of the random nature of the errors, it is necessary to generate by Monte Carlo techniques an ensemble of machines, and to make statistical predictions from this ensemble of machines as to what the expected behavior of the real machine will be. Even for an already constructed machine, the knowledge of individual errors levels is highly imperfect. Further, these errors are likely to change with time, so that again prediction is statistical in nature. Therefore the only requirement for a model to be used in conjunction with a Monte Carlo generator is that it provide an ensemble of machines with the statistically correct overall ensemble properties. Herein lies the main difference between our approach—which aims to reproduce only the ensemble properties—and more restrictive mapping techniques which attempt to provide exact case-by-case representations.

We follow the widely used and accepted procedure of representing large collider lattices as successions of thin multipole elements, separated by first-order drift-bend elements. The thin multipoles elements give discrete symplectic kicks to particles traversing via their multipole fields B_n

$$B_n = (a_n + ib_n)(x + iy)^n \ .$$

An SSC half-cell contains a quadrupole, a chromatic sextupole corrector, and five bend magnets. All of these components may be incorrectly placed, or contain systematic or random multipole errors. The representation of these errors is now described.

Systematic errors

Systematic errors—following a scheme suggested by Neuffer—are lumped so that the average, first, and second moments of the errors are maintained. Using this procedure, a series of multipoles of order n, and strength s_{ni} at points t_i, where t_i is the displacement from cell center and $2t_0$ is the half cell length, are replaced by three values S_{1n}, S_{2n}, S_{3n} at the beginnings, midpoints and ends of the half-cells, such that

$$S_{1n} + S_{2n} + S_{3n} = \int s_n dt \ , \tag{1}$$

$$-t_0.S_{1n} + t_0.S_{3n} = \int t s_n dt \ , \tag{2}$$

$$t_0^2.S_{1n} + t_0^2.S_{3n} = \int t^2 s_n dt \ , \tag{3}$$

where the integration is over a cell from $-t_0$ to $+t_0$.

This corresponds in the limit of many discrete errors to distributing errors in the ratio 1:4:1. This procedure can be analytically shown to provide non-linear frequencies shifts good to a few percent.

The SSC is constituted of FODO half-cells, with $\pi/2$ phase advance per cell. The machine consists of interaction clusters connected by two arc sectors. The new element in the procedure is to further lump the systematic multipole errors for each sector into a π phase advance cell consisting of four FODO cells (with sixteen quad and mid-cell positions). For a sector of N cells, the errors are distributed such that the quad and mid-cell errors in the four FODO cells are $N/4$—what they would have been in the simple lumping scheme—with the remainder of the sector now left free of errors.

This scheme gives frequency shifts that to first order are still close to those found from analytic calculation. It further correctly incorporates the lattice symmetries. The $\pi/2$ condition provides that only phase advance, and not beta-beat errors, are introduced to first order. The π condition causes terms from even multipole terms to cancel to first order. These properties are valid for systematic errors distributed over many betatron wavelengths.

Random errors

Random errors use a similar lumping scheme with Gaussian, not linear, addition.

Formally, each error on a given dipole at the ith position in a cell is lumped via a set of weights w_{ni} onto the end and mid-cell positions, where n runs from 1 to 3, corresponding to the three cell positions, with

$$S_n = \sum w_{ni} s_i \ . \tag{4}$$

The set of weights, w_{ni}, was chosen to be symmetric about the midcell position, and to give unchanged averages and first moments. Further requirements on the weight set were that the lumped errors at the three cell positions be statistically uncorrelated and, for the case with constant errors, that the ratio of S_ns reduce to 1:4:1—the same as for systematic errors. This choice of weights provides a representation of lattice behavior correct to better than 10%.[4] The resultant errors are further condensed from the N-sector cells onto four cells per sector by Gaussian addition.

While the above scheme could be directly implemented in the Monte Carlo generator, a much simpler but entirely equivalent procedure is available. The weighting scheme was specified to make the S_n uncorrelated. Thus the S_n are Gaussian, and independently distributed, and a statistically identical ensemble of lattices can be directly generated with a random Gaussian generator, using the rms value for each S_n. The rms values can be derived directly from the w_{ni}, and are in the ratio of 1:2:1. This is the actual procedure used in the program.

First-order lattice errors and chromaticity

In any real machine, subsequent to commissioning, the machine will go through substantial error corrections procedures; for instance: dipole steering errors, quadrupole errors, cross-coupling, and chromaticity will be largely corrected out. It is possible to go through operational simulation of the procedures to be used. In a model involving substantial concatenation, this can be tedious, and we have chosen instead to explicitly input first-order errors or chromaticity into the lattice. This procedure permits tolerances to be placed on the required correction levels for the lattice.

Cross-coupling errors are represented by a first-order matrix transfer R_c of the form

$$R_c = \begin{pmatrix} 1\ ,\ 0\ ,\ 0\ ,\ 0 \\ 0\ ,\ 1\ ,\ F\ ,\ 0 \\ 0\ ,\ 0\ ,\ 1\ ,\ 0 \\ F\ ,\ 0\ ,\ 0\ ,\ 1 \end{pmatrix}$$

where F is a variable that controls the coupling.

Tune shift adjustments, modulation, and alpha or beta-beat are put in via matrices R_1 and R_2

$$R_1 = \begin{pmatrix} 1\ ,\ 0\ ,\ 0\ ,\ 0 \\ A+B,\ 1\ ,\ 0\ ,\ 0 \\ 0\ ,\ 0\ ,\ 1\ ,\ 0 \\ 0\ ,\ 0\ ,C+D,\ 1 \end{pmatrix}$$

$$R_2 = \begin{pmatrix} 1\ ,\ 0\ ,\ 0\ ,\ 0 \\ A-B,\ 1\ ,\ 0\ ,\ 0 \\ 0\ ,\ 0\ ,\ 1\ ,\ 0 \\ 0\ ,\ 0\ ,C-D,\ 1 \end{pmatrix}$$

where A(C) introduces tune shift and B(D) introduces β beat. A(C) and B(D) can be functions of time (modulation) or momentum (chromatic effects.)

The matrices are incorporated into a variable tune and/or modulation, and/or beta-beat section, using the transfer matrix R_{err} formed by the combination

$$R_1 \times \left(\frac{\pi}{2} \text{ phase advance}\right) \times R_2 \times \left(-\frac{\pi}{2} \text{ phase advance}\right) .$$

Without errors, the transfer matrix reduces to a unit transfer matrix. With this representation, the first-order properties of the lattice can be modified orthogonally, permitting exploration of permissible tolerances in a corrected lattice.

CPU usage

Six additions and four multiplications are required to implement a single multipole kick. A transfer bend drift requires four additions and four multiplications. Therefore, the total operations required for the above model evaluated for multipoles out to ninth order is slightly above $32 \times (9 \times 4 + 4)$ or 1380 multiplies and 1380 additions per turn. The minimum operations required for mapping are one addition and one multiplication per term. Achieving this minimal set of operations requires "indirect" addressing, which adds substantial overhead to the CPU requirements. A truncated one-turn Taylor series map running at equivalent speeds would be, at most, a fifth-order map (1,008 terms).

CHECKS AND RESULTS

We have checked the model against previous full SSCTRK results for dynamic aperture, both as a function of tune and of storage time. We have also used the model to observe effects from modulation or "finite momentum" correction terms.

Figures 1 and 2 show typical dynamic-aperture-versus-tune runs for equal horizontal and vertical tune, for a full SSCTRK set of simulations and for the current model. Figure 1 is run with the detailed SSCTRK lattice representation for one seed, and Figure 2 is a simulation using the model for 10^5 turns at 100 equally spaced tunes and run for four different random generator seeds. The results are closely comparable.

Figure 3 shows a typical "loss" plot obtained with the full SSCTRK tracking program. A loss plot plots points corresponding to the number of turns at which

particle losses occurs versus aperture. The set of points are generated by starting particles at amplitudes that lie outside the dynamic aperture. When a given particle is lost out of the physical machine aperture, it is restarted at a smaller amplitude, and this process is continued until the particle runs stably for a full complement of turns. This procedure delineates an envelope for the stable aperture. A "survival" plot is a complementary plot histogramming starting amplitudes for the particles that survive to the full complement of turns. The plots are generated for a mix of starting phases, amplitudes, and machines (different Monte Carlo generator seeds).

Figures 4, 5, and 6 are loss curves for a full SSC injection time, corresponding to 2×10^7 turns evaluated with our model. They compare results respectively for tune modulations at the synchronous frequency .001, .0025, and .005. Tune modulation is accomplished by introducing modulation of the appropriate magnitude via the variable phase shift section, while leaving the rest of the machine identical. Comparison of Figure 3 with Figure 4 again demonstrates the close similarity of results obtained with the model and with the full lattice.

Comparison of Figures 4, 5, and 6 demonstrates the degradation of dynamic aperture for large storage times associated with increasing tune modulation. The SSC design specifications are for modulations below $\Delta\nu \leq 10^{-3}$, and the large modulation values in Figures 4 and 5 should not occur in practice. The plots demonstrate that the functional dependance of available aperture on storage time is not a universal curve.

Figure 7 investigates the effect of introducing finite momentum correction terms[5] into the tracking code. These terms arise because the variables operated on by first-order bend drift sections are displacements and transverse momenta divided by longitudinal momenta, p_t/p_l. The corresponding variable for thin-lens multipole kicks is transverse momenta divided by total momenta p_t/p_0. Formally, prior to a thin-lens multipole kick, the variable should be transformed by p_l/p_0 and, subsequent to the kick, transformed back by a factor p_0/p_l (using the values subsequent to the kick). In the approximation of infinite momenta, these correction vanish. While these correction factors are very small for large colliders, it has been argued that such small effects might cause bounded motion to become unbounded and thus cause observable

changes to long-term dynamic apertures. The inclusion of such corrections requires the evaluation of square roots, and increase CPU usage by a factor ~ 3. As they are generally believed to be inconsequential, most codes omit such corrections. To investigate the effect of this correction, we compare in Figures 4 and 7 runs, differing only by the absence or inclusion of these corrections. In a chaotic domain, exact numerical agreement is not to be expected. However the results are statistically indistinguishable, and this demonstrate that it is unnecessary to include such corrections terms.

CONCLUSIONS

The model obtained with strong conglomeration of multipole errors onto 32 lattice points provides a good representation of long-term behavior of the dynamic aperture. The model is symplectic, and is more than adequate to represent arc effects for interaction optics at 20 TeV, where beam–beam and IR-quad effect restrict the dynamic aperture to about 20% of what it would be including only the arc errors.

The CPU usage required for tracking one turn is minimal, and comparable with that required for fifth-order single-turn Taylor maps.

The long-term dynamic aperture is strongly affected by tune modulation, and care should be exercised that the nominal limits on modulation are not exceeded.

Inclusion of "finite momentum" correction terms does not result in changes to dynamic apertures for runs extending to 2×10^7 turns and little purpose is served by the inclusion of such terms.

The model simultaneously incorporates the full range of physics, provides results close to those observed with detailed lattice representations, is fully symplectic and can be numerically evaluated for the full range of turns to be expected in actual machine operation. Therefore, it can be used to provide benchmark tests to verify the precision of mapping techniques applied over the number of turns expected in actual operation.

REFERENCES

1. Garavglia, Kauffman, Stiening, and Ritson, SSCL–268 (1990).
2. D. Ritson, Proc. Workshop on Non-Linear Problems in Future Accelerators, Capri, 1990, p. 304.
3. See for instance, S. Kauffman, Y. Yan, and D. Ritson, SSCL–321 (1990). This study gives a good representation for seventh-order maps out to 2 mm, or one third of the dynamic aperture. However, Y. Yan found in concomitant studies that an eleventh-order map was required to model out to the full aperture of 6 mm.
4. J. Peterson and E. Forest, Proc. Part. Acc. Conf., Rome, 1988, p. 827, and private communication from E. Forest relative to the exact scheme described.
5. L. Schachinger and R. Talman, Part. Acc., Vol. 22, 1987.

FIGURE CAPTIONS

1. Dynamic aperture as a function of tune for $\nu_x = \nu_y$ as evaluated via the full element SSCTRK code (one input seed).
2. Dynamic aperture as a function of tune for $\nu_x = \nu_y$ as evaluated via the present model (four input seeds).
3. Typical loss curve for SSCTRK full simulation.
4. Loss curve with 0.001 chromatic tune modulation for present model (eight input seeds). The hardline is drawn for comparison purposes and is the envelope of the "loss curve."
5. Same as Fig. 4, but for loss curve with 0.0025 chromatic tune modulation.
6. Same as Fig. 4, but for loss curve with 0.0050 chromatic tune modulation.
7. Same as Fig. 4, but for loss curve with 0.001 chromatic tune modulation, with "finite momentum" correction included.

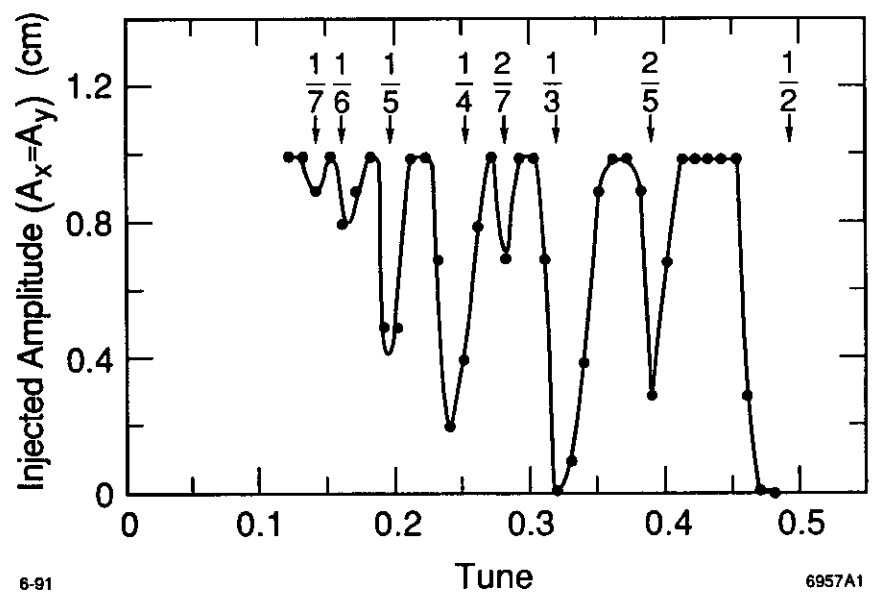

Fig. 1

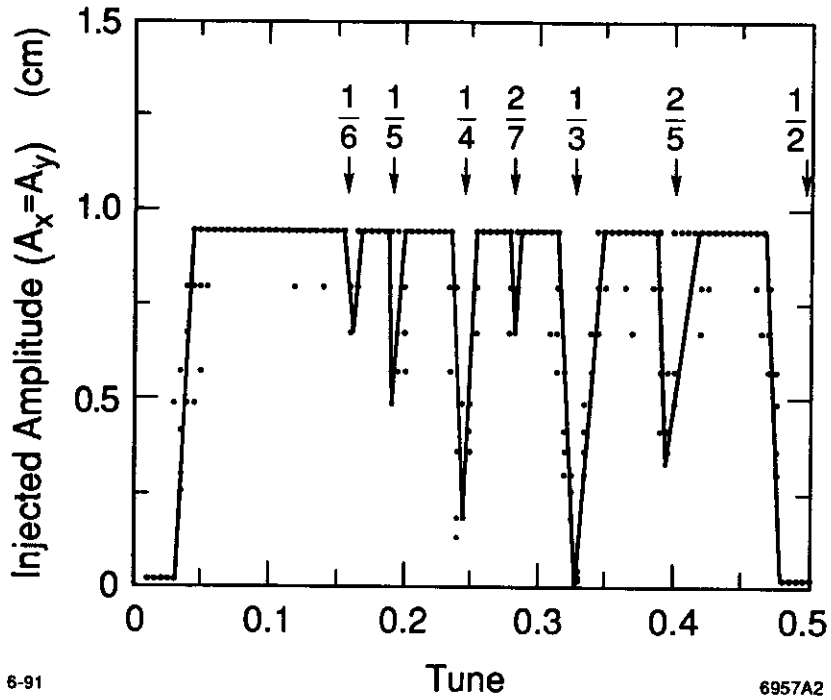
Injected Amplitude (A_x=A_y) (cm)
1.5
1.0
0.5
0
1/6
1/5
1/4
2/7
1/3
2/5
1/2
0
0.1
0.2
0.3
0.4
0.5
Tune
6-91
6957A2

Fig. 2

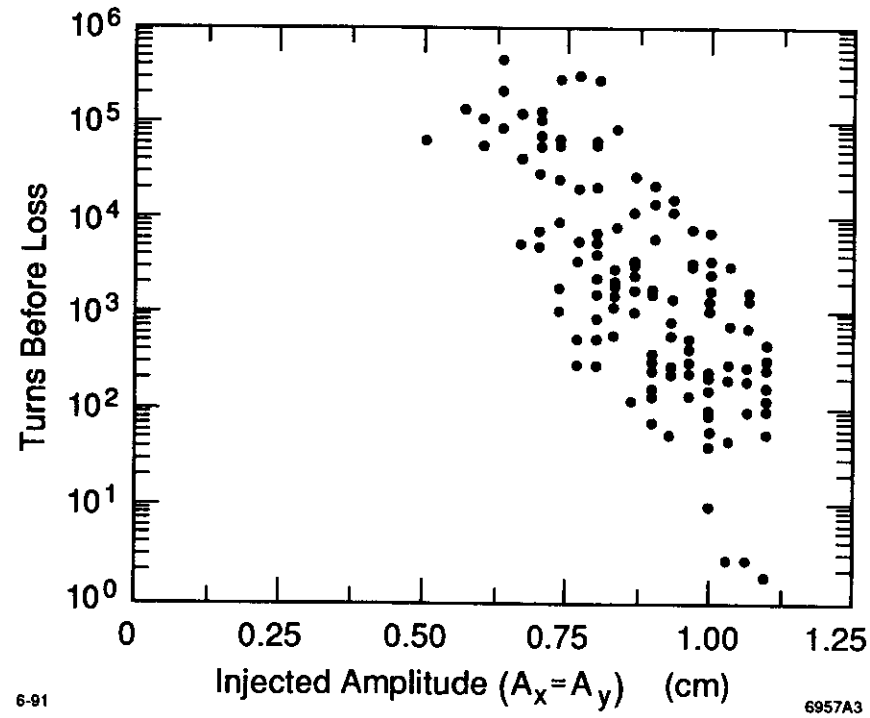
Turns Before Loss
10^6
10^5
10^4
10^3
10^2
10^1
10^0
0
0.25
0.50
0.75
1.00
1.25
Injected Amplitude (A_x=A_y) (cm)
6-91
6957A3

Fig. 3

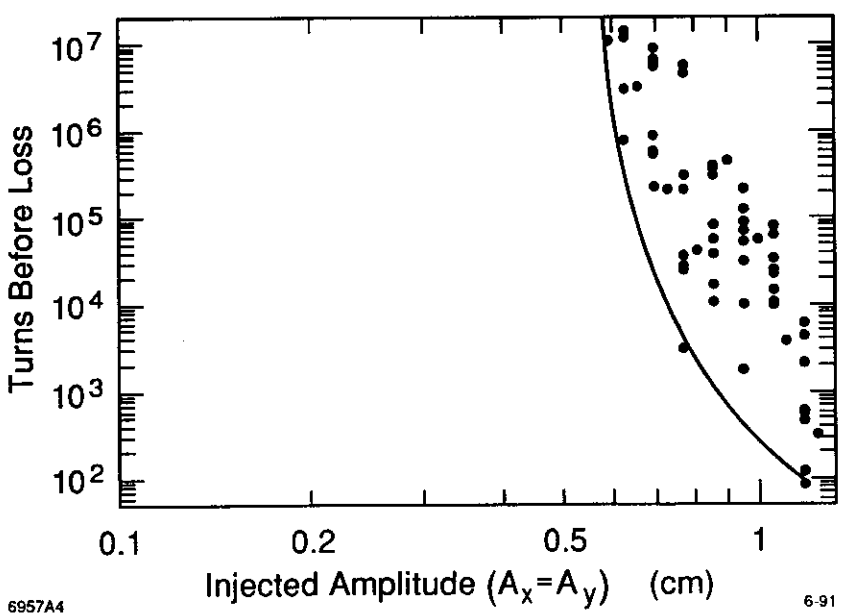
Turns Before Loss
10^7
10^6
10^5
10^4
10^3
10^2
0.1
0.2
0.5
1
Injected Amplitude (A_x=A_y) (cm)
6957A4
6-91

Fig. 4

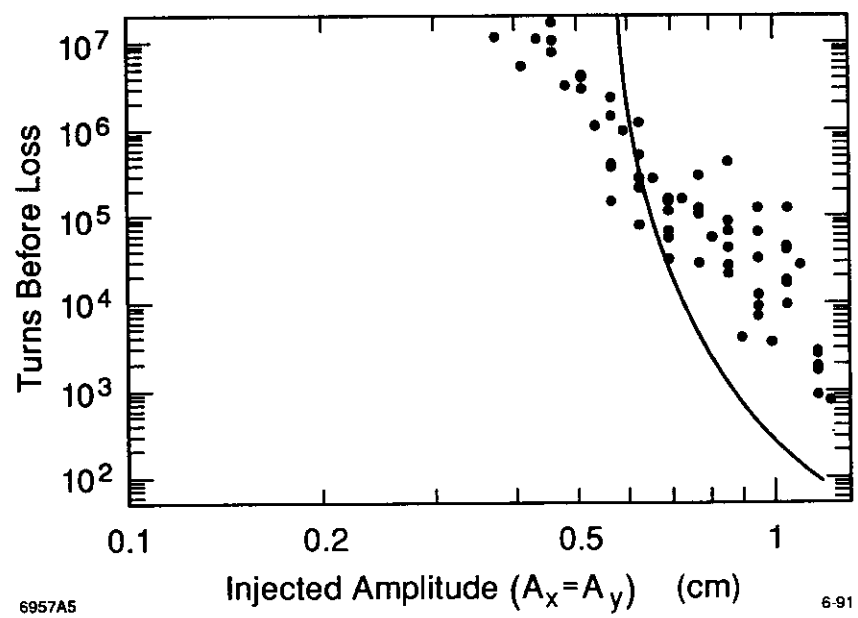
Turns Before Loss
10^7
10^6
10^5
10^4
10^3
10^2
0.1
0.2
0.5
1
Injected Amplitude (A_x=A_y) (cm)
6957A5
6-91

Fig. 5

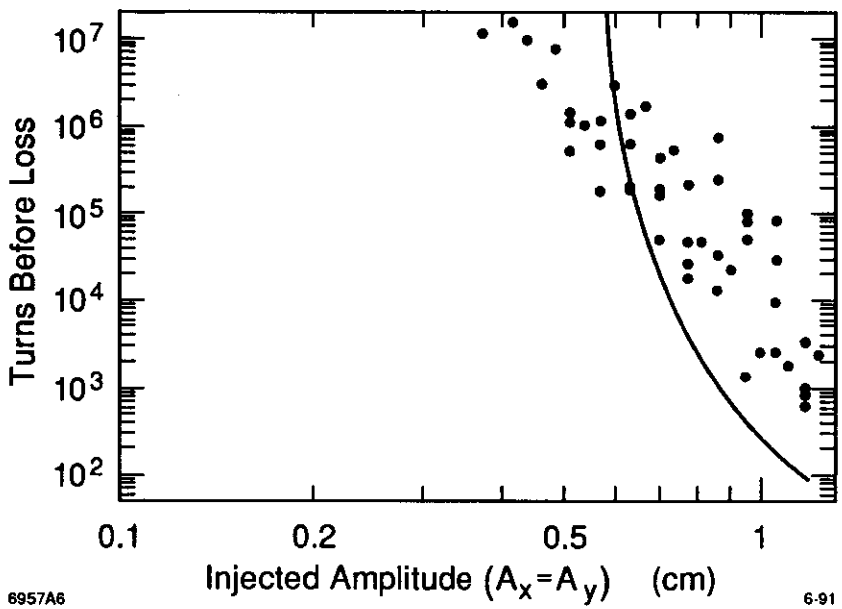
Turns Before Loss
10^7
10^6
10^5
10^4
10^3
10^2
0.1
0.2
0.5
1
Injected Amplitude (A_x=A_y) (cm)
6957A6
6-91

Fig. 6

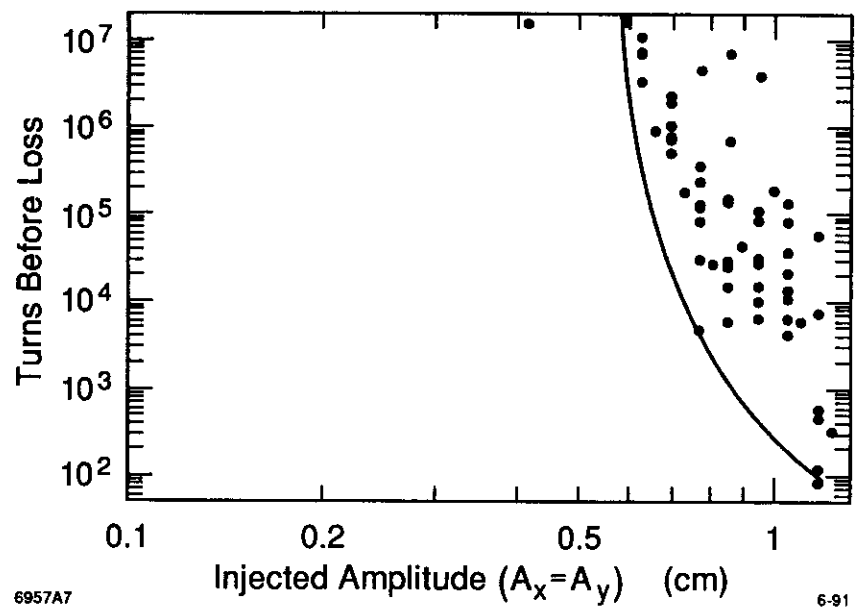
Turns Before Loss
10^7
10^6
10^5
10^4
10^3
10^2
0.1
0.2
0.5
1
Injected Amplitude (A_x=A_y) (cm)
6957A7
6-91

Fig. 7