# THE RHIC/AGS ONLINE MODEL ENVIRONMENTS: EXPERIENCES AND DESIGN FOR AGS MODELING*

K. Brown†, J.Niederer, T. Satogata, A. Alai Tafti, N.Tsoupas,
J. van Zeijts, BNL, Upton, NY

*Abstract*

The RHIC/AGS online modeling environment, a general client-server modeling package that supports cdev and straightforward integration of diverse computational modeling engines (CMEs), is being adapted to model the AGS and Booster at BNL. This implementation uses a version of MAD modified at BNL that allows traditional lattice structure analysis, single pass beam line analysis, multi-particle tracking, interactive graphics, and the use of field maps. The on-line model system is still under development, a real working prototype exists and is being tested. This paper will describe the system and experience with its design and use for AGS and AGS Booster online modeling.

## 1 BNL MAD

One of the paths of model development at BNL has emphasized graphics based enhancements to the usual collection of traditional calculations. This emphasis has also provoked efforts to speed up these calculations, and to introduce various tools that make it easy to add features and new sets of calculations. These graphics based programs are similarly attractive as on line tools which can aid in operating the various accelerators and transfer lines in the AGS RHIC complex. Techniques are centered around a library of prepared commands stored in the BNL MAD data base, which are called by menu from the console. Graphics services are contained within the program and coupled directly to the orbit and tracking calculations. Parameters can be changed by means of menus and sliders, and steps repeated as the effects of parameters are observed and evaluated.

### *1.1 Calibration of the Model*

Ideally, live data is drawn from the various components of the accelerator, and used to describe the corresponding elements of the model. Orbits are computed, and compared with measured orbits. If sufficient care has been taken both to verify the data and to build the model, perhaps the orbit of the model agrees with the measurements. In our experience, there have often been glaring discrepancies between the two, compromising any reliance on the use of models. The kinds of problems are all too familiar. A few unreliable measurements, misaligned apparatus, inaccurate elements in the models all take their toll. While all of these irritants can be improved upon over time, they are rather unhealthy obstacles to satisfactory on line models.

We can make use of a series of virtual correctors in our models to bring model computed orbits into agreement with measured values. In this technique various harmonics are placed on the virtual correctors, and fitted so the resulting computed orbit best matches the measured orbit. The scheme is rather insensitive to missing monitors or correctors, works particularly well for our smaller machines, and produces decent matches to the measured orbits without marked distortions. Accompanying graphics shows the build up of the fitted harmonics, and whether the results can be trusted. The fitted virtual corrector patterns follow the details of the actual orbits considerably better than a simple harmonic fit to the measurements. In effect, the model is now realistically matched to the actual machine, and further work with it is likely to be more credible.

Figure 1 illustrates this kind of fit to a ragged Booster orbit. This fit is drawn beyond the actual orbit to show its continuity.

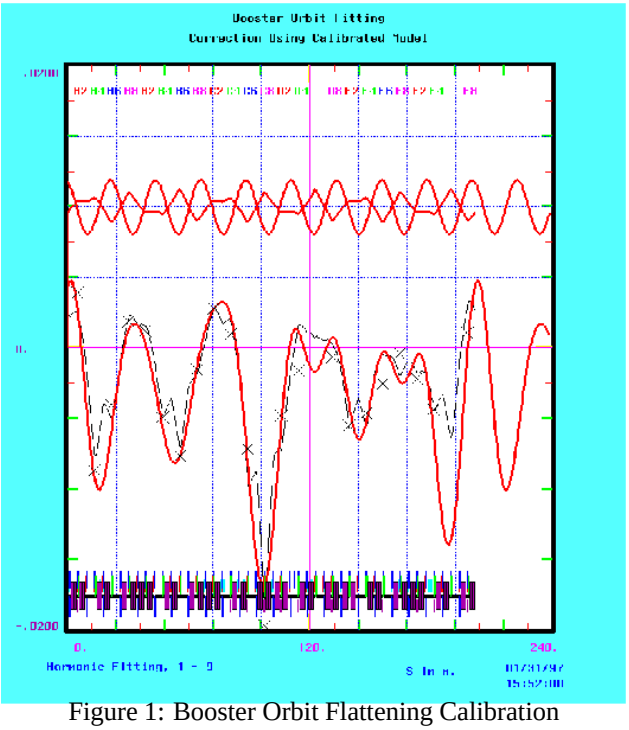


Figure 1: Booster Orbit Flattening Calibration

* Work supported by the U.S. Department of Energy.
† Email: kbrown@bnl.gov

### 1.2 *Dynamic Orbit Correction*

We have developed models of automatic orbit correction systems with displays that help to evaluate stability. Components include neuron like objects that collect signals, apply filters, and pass the signals on to overall controllers and then correctors. The components are coupled non linearly, like actual neurons, which is easily shown to increase the stability of these schemes. A Micado based controller is particularly effective at localizing responses to orbit distortions. Graphics aids help to demonstrate convergence, effect of gain, and onset of instabilities. Various time delays can be introduced, and their effects on responses observed. In Figure 2, we show the response of a 24 corrector system to an artificial orbit spike in an AGS model, over six cycles of read, filter, analyze, and set correctors. Here the gain of .4 is a little too large, so there is a noticeable overshoot.

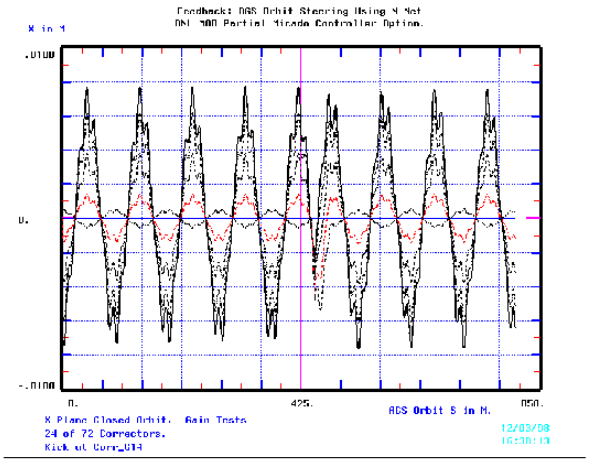


Figure 2: AGS Orbit Flattening

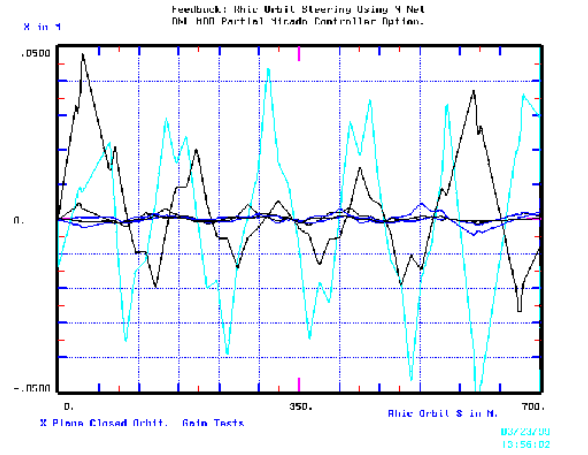


Figure 3: RHIC Micado based Orbit correction

A similar configuration can be used to observe convergence with the controller algorithm. In Figure 3, we show the response of a Micado based correction scheme to an exaggerated static error distribution used for spin preservation studies on RHIC. Both X and Y planes are shown for three cycles of a Micado iteration with all the signal filters turned off. The insertion region correctors are not included.

### 1.3 *Stability Studies*

The graphics offer a particularly dramatic way to evaluate the stability of a particular lattice configuration. A calculation of orbits is easily iterated for a few values of some parameter, such as the strength of a line quadrupole, and the lattice functions observed. The range of iterated values typically covers plausible error margins. If the envelopes of these iterated functions are narrow, the design is most likely very stable. Similarly, the behavior of the lattice functions can also be observed for varying momentum offsets and the various other optical input parameters. Competing designs, and set up parameters can be easily compared. In Figure 4, the stability of a suggested transfer line design is probed by varying all quads in the line by a few amperes. In Figure 5, the stability of an early LHC insertion design with respect to quad strengths at injection is displayed. Both designs appear to be unusually stable with respect to drifts in quad values. Both are similarly insensitive to a range of particle momenta.

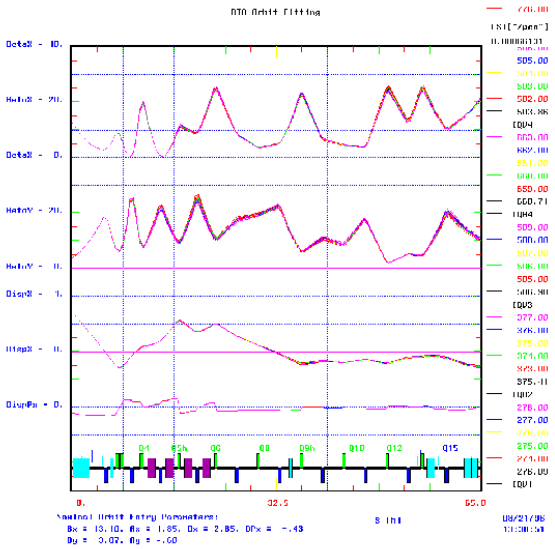


Figure 4: AGS Booster to AGS transfer line

While lattice designs usually involve complicated multiparameter fitting via conventional matching calculations, once in place their operation is greatly aided by graphics displays that show relative sensitivities of orbit functions and dispersion to various parameters. Knob twiddling is seldom the best way to optimize among twenty or so variables involved in steering beams.

A related issue in accelerator designs is the sensitivity to various misalignment and mispowering errors. There is often some doubt about bias introduced by the use of random distributions, which tends to be explored by iterating with numerous other distributions. Again, our orbit function graphics can be used to show the spread of effects due to a variety of random number seeds used to generate error distributions.

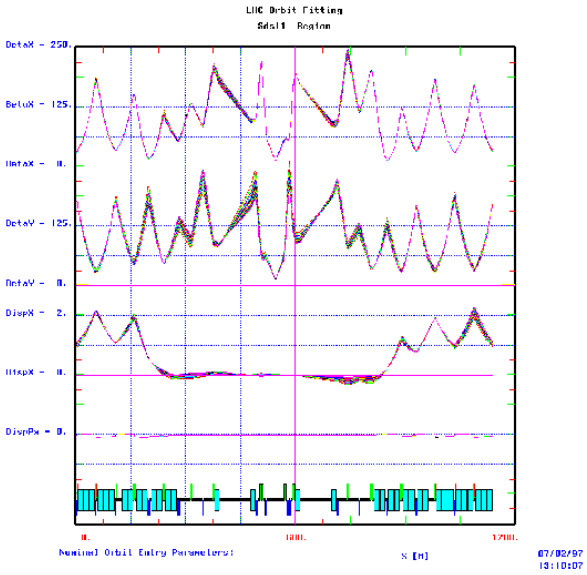


Figure 5: LHC Insertion Design

### 1.4 *Particle Tracking*

A graphics based procedure has been developed for tracking charged particles in the presence of magnetic fields changing in time relative to the usual reference conditions, as in acceleration and ejection scenarios. For tracking models to be realistic, accurate aperture data and time profiles of the changing fields must be included in the description of individual lattice elements. Groups of particles can be selectively captured in septum elements, and then further tracked down branch lines radiating from the septum. Various initial track patterns can be generated by selecting from a stored library of commands, and single or multiple track groups can be followed. This tool is particularly helpful for operating auxiliary beam lines.

A rather simple example is shown in Figure 6 in which an AGS Booster beam is coasting inwards as the main field increases.

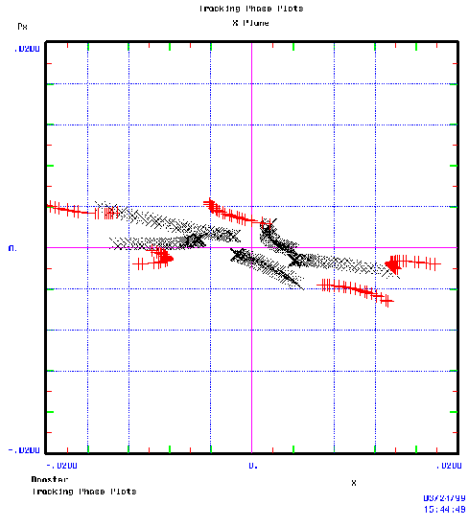


Figure 6: AGS Booster, Tracking with Time Varying Fields

## 2 THE ON-LINE MODEL SERVER

The model server is described in detail in [Satogata [1]]. It is built using the cdev generic server and is designed to allow integration of diverse computational model engines. Integration of BNL MAD is more involved (as would be any monolithic application that interfaces solely with file I/O), since it is built with no support for general Unix interprocess communication.

### 2.1 *Server Interface*

To connect BNL MAD to the model server a set of "wrapper" functions are built which are encapsulated into the model server. The wrapper functions for BNL MAD connect to the executable using named pipes and can be put into a separate server, to allow the model server to communicate over a network. The main part of this work builds a middle-ware which allows the model server to communicate to any model engine, locally or over a network. The reason for taking this approach is that other model engines are in use at BNL (transport, turtle, marylie, etc.), and we will want to be able to interface these model engines to the system. Much of this work is still under development. To first order the server is only interested in obtaining numerical results for other controls applications to make use of (using BNL MAD as just a sophisticated calculator). But we also intend to have clients that make use of the BNL MAD graphics utilities and the full power of BNL MAD, and so the interface is designed to be encapsulated directly into client applications also.

### 2.2 *Supported Platforms*

Presently BNL MAD with all its functionality is only supported on Silicon Graphics workstations. This is because the graphics utilities make use of the pre-OpenGL tools IrixGL. The computational engine of BNL MAD has been ported over to the Sun OS platforms, and the graphics is currently being ported over to OpenGL.

## 3 CONCLUSIONS

BNL MAD is a versatile and powerful set of tools which is well established in the BNL community and beyond. These tools are capable of solving complicated design problems as well as being used to test against experimental results. With the added capabilities of its graphics interfaces, problems can be studied and solved very quickly. The design process is greatly enhanced as a result. It is natural to want to incorporate these sophisticated applications into the actual accelerator controls, or at least, put it into the hands of the accelerator operators to be used as a tuning guide. We believe this is feasible and have taken the first steps in implementing it.

## 4 REFERENCES

[1] T.Satogata et al, "The RHIC/AGS Online Model Environments: Design and Overview", These proceedings.