

# BNL MAD: Harmonic Analysis Based Orbit Correction Commands AGS Booster Applications

J. Niederer

February 1997

Collider Accelerator Department  
**Brookhaven National Laboratory**

**U.S. Department of Energy**

USDOE Office of Science (SC)

Notice: This technical note has been authored by employees of Brookhaven Science Associates, LLC under Contract No. DE-AC02-76CH00016 with the U.S. Department of Energy. The publisher by accepting the technical note for publication acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this technical note, or allow others to do so, for United States Government purposes.

## **DISCLAIMER**

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

*For Internal Distribution Only*

Accelerator Division  
Alternating Gradient Synchrotron Department  
BROOKHAVEN NATIONAL LABORATORY  
Upton, New York 11973

Accelerator Division  
Technical Note

**AGS/AD/Tech. Note No. 456**

BNL MAD

Harmonic Analysis Based Orbit Correction Commands

AGS Booster Applications

J. Niederer

February 24, 1997

# BNL MAD

## Harmonic Analysis Based Orbit Correction Commands

### AGS Booster Applications

*J. Niederer*

AGS Department  
Brookhaven National Laboratory

February 24, 1997

#### 1. Introduction

A method of using charged particle orbit data to compute coefficients of a Fourier series for sets of orbit corrections is described. An accelerator lattice is defined as well as practical with what is known about machine conditions. Any remaining differences, monitor errors, and other unknowns are then lumped together and approximated via a harmonic series applied to a family of calibration correctors. This family can be real correctors for convenience, or virtual ones distributed throughout the lattice for calibration / correction purposes only. This method has been added to the local BNL MAD programming environment, and demonstrated with such elderly AGS Booster data as available.

Various kinds of harmonic based orbit correction schemes have been practiced on most accelerators now for years. What may be a little different here is using such a scheme to bring accelerator models into essential agreement with actual measurements, thus providing an absolute basis for guiding orbit controls. Furthermore, the algorithms to be described appear to be extremely stable computationally, and hence very suitable for automatic orbit control application. The scheme can tolerate individual components popping in and out of service, and a fair amount of the other disorder that often accompanies actual machine operation.

In the calibration mode, orbit measurements are used to adjust a set of conventionally arranged correction elements (real or virtual) to match the measurements. A closed orbit is computed for each of a set of trial amplitudes for a harmonic pattern on the calibration correctors to search for an amplitude that minimizes the least squares residuals between the orbits by computation and by snapshot. The result is a set of harmonic amplitudes, and a "best fit" momentum offset term  $\Delta P$  for the X plane, and an  $a_0$  term for the Y plane. Harmonics are subtracted off one at a time, and are relatively insensitive to the order of subtraction, and to gaps in the recorder and corrector patterns. In the AGS Booster application noted here, all fitted amplitudes are well behaved, being well under the maximum orbit excursions. The expected  $J = 4$  and  $J = 5$  contributions are larger, and the others are smaller, and presumably help to measure the noise level of the method. One can watch the reduction of the least squares progress as each harmonic is fitted. This procedure effectively calibrates the program, using the smoothing properties of the harmonic approximation and the lattice optics to adjust the program as needed to yield the measured orbit. The program with the fitted adjustments is now able to reproduce the measurements within some quite acceptable margin of error. The resulting configuration typically reduces the discrepancies between the calibration measurement set and the closed orbit found by the program by a factor of ten or better.

In a second step, the now calibrated program may be asked to alter the orbit from the original measurements to something else, such as a flatter orbit. Orbits are now computed for trial harmonic patterns on the real correctors, and a penalty function comparing the desired with the measured orbit is minimized. The change in the real correctors is now in principle an *absolute* set of numbers, as the program has been calibrated to match the starting orbit values. The procedures are readily iterated to reduce further any

remaining discrepancies.

The notes describe the somewhat experimental MAD command setups, the graphics capabilities, and some demonstration results. A new command, **Vcorrs**, gathers the necessary information and performs the computations. The emphasis here is to report the apparent possibilities of the method rather than provide polished code and a manual. The method is a rather straightforward extension of the constrained amplitude subtraction procedure described previously, and implemented in the **Vharm** vector harmonic analysis command. In their current state of development, the commands and data needed are probably more complicated than they really have to be, and if any interest is aroused, may be expected to improve.

These notes use the language of orbit positions, and orbit monitors and correctors, in the conventional sense of the usual MAD lattice element definitions. Procedures are based on the notion of families of similar monitors and correctors, identified by names in lists. Actually the program can work with any family of distributed magnets, windings, or fudges with common attributes. Similarly it can provide fits to other practical kinds of distributed measurements as well.

## 2. Constrained Fits

Experience with the **Vharm** vector harmonic analysis command of the local BNL version of the MAD accelerator program indicated that constrained amplitude subtraction of individual harmonics produced reasonable results that were rather insensitive to missing items or regions of the sampled data set. The fitted functions behaved properly between the fitted data points, as well as at the points themselves. Input values could be irregularly spaced: spacing was not restricted.

In the familiar problem, a pattern of  $I$  measurements  $V_i$  taken at locations  $S_i$  over a distance  $S_{\max}$  is to be fitted to a harmonic series:

$$V_i = A_0 + \sum_{j=jmin}^{j=jmax} A_j \cos(2\pi j S_i / S_{\max}) + \sum_{j=jmin}^{j=jmax} B_j \sin(2\pi j S_i / S_{\max})$$

When treated as an extremal problem, the set of  $A_j$ ,  $B_j$  are found by minimizing the difference at each point between the measurement and the fitted function. The measurements may also be weighted individually, with weights noted by  $W_i$ . The function to be minimized is:

$$\sum_{i=1}^I W_i (V_i - A_0 - \sum_{j=jmin}^{j=jmax} A_j \cos(2\pi j S_i / S_{\max}) - \sum_{j=jmin}^{j=jmax} B_j \sin(2\pi j S_i / S_{\max}))^2$$

This general form of the fitting criteria is adjusted somewhat for the realities of working with corrector sets. In the present application, the intent is to place an optimal harmonic pattern on a family of correctors, rather than merely analyze orbit measurements in terms of harmonics. The function to be minimized now becomes:

$$\sum_{i=1}^I W_i (V_i - T_i)^2$$

where the  $T_i$  are computed values at the monitors, obtained from closed orbit tracking computations. The trial amplitudes  $A_j$  and  $B_j$  are considered in the orbit measurement space of the monitors, and constrained to be within some fraction of the largest measured amplitudes. These trial values translate to the family of corrector values as:

$$C_i = F_i \left( \sum_{j=jmin}^{j=jmax} F_{corr_j} (A_j \cos(2\pi j S'_i / S_{\max}) + B_j \sin(2\pi j S'_i / S_{\max})) \right)$$

where the  $S'_i$  are now the corrector locations along the orbit. The individual factors  $F_i$  for the correctors are applied as given, and the common  $F_{corr}$  factors computed from the changes in all of the  $T_i$  produced by an incremental harmonic change  $dC_i$  in all of the correctors.  $F_{corr}$  is somewhat artificial, and is introduced only to help with the amplitude constraint, which is expressed in terms of the orbit measurements. These factors are a measure of the efficiency with which a given harmonic is produced in the orbit when placed on the correctors. The practical range of actual corrector values is limited as well. While the program offers an optional feature to clamp corrector values, the values achieved so far have been sufficiently reasonable that further constraints have not been considered.

In a textbook styled implementation, corrector and monitor sets have similar placements and periodicities, and perhaps the two kinds of analysis may then be essentially equivalent, but such regularities are not a precondition here. Except for institutions where there is some deep seated religious aversion to working monitors and correctors, the scheme should be adequate for any reasonable configuration of monitors and correctors.

In the calibration mode, the model is asked to modify itself in terms of harmonic based corrections to satisfy given orbit measurements (snap shots). In the X plane the fixed A0 term represents an average displacement, which physically is a result of a momentum offset from the so called reference orbit, so this offset delta P is adjusted in the model to match the snap shot data. This adjustment should cause the A0 term to vanish. It provides a smooth adjustment to the model at every element of the orbit, which is preferable to a set of kicks at sparsely located correctors.

After the momentum offset has been obtained, the constrained method selects for each harmonic j an even (A) and odd (B) coefficient that produce the largest reduction of the above penalty function. Initially, the weighted squares of the difference between each point of the data set and the computed contribution from harmonic Aj, Bj are summed to obtain the penalty. The program searches in an Aj, Bj space for this reduction. The Aj and Bj pairs must satisfy the relation

$$A_j^2 + B_j^2 < \text{Constraint} * \text{Initial\_Amplitude}^2$$

Trial sets of Aj and Bj amplitude pairs are cycled, over a grid that is reduced over succeeding passes. First an Aj is cycled over a few intervals of da, where da is formed from the largest residual amplitude as:

$$da = \sqrt{\text{Constraint} * \text{Abs}(R_{\max})} / \text{Ndivs}$$

Ndivs is some specified number of divisions, perhaps about 8 or 10, which balances accuracy against minimizing the number of rather slow orbit computation passes. In the first pass, both Aj and Bj are taken as zero. A crude search is made by computing penalty functions of residuals, starting with the initial value for Aj. A minimum point and its two neighbors are sought. These three points are interpolated quadratically to estimate a minimum. Using this estimate for Aj, the steps are repeated for the corresponding Bj over a range subject to the constraint. The interval da is reduced about the best Aj, the penalty calculations repeated using the now best Bj, and Aj further localized. Then Bj is similarly cycled in steps of smaller db using the best Aj. The steps are repeated a few times, as needed to converge. The contribution of this Aj and Bj is added to the accumulating values on the correctors. The process is repeated for the next, and remaining harmonics. If iterated further, the starting Aj and Bj are taken from the previous best values, and any further changes to the Aj and Bj are sought over a reduced search region with a smaller constraint. When finished, the Vi - Ti residuals are a measure of the fit over the given range of harmonics. The sum of the squares of the residuals, the penalty function, is now in a sense a least squares for the fit.

As with the **Vharm** procedure, several options are provided to help test the validity of the fits. In a *Permute* option, the order of processing of the harmonics can be scrambled. The fits of a set of harmonics are repeated in a pattern which begins with a pivot harmonic, *jmain*, followed by the next lower one, then by the next higher one, until the range has been covered. The pivot is then advanced by one, and the next version of the pattern repeated. For low order harmonics, the method has appeared insensitive to the order in which harmonic contributions are subtracted. There are some minor differences among these trials with permuted orders, but they appear well within expected errors, and tend to involve harmonics which have little overall effect. In the *Drop Mode* option, the fits can be cycled over a pattern which drops one of the data points, performs the fit, restores the dropped point, and repeats with the next point dropped. A dropped data point that leads to a markedly better fit may cast some doubt on the point. Results may have some minor sensitivity to the dropped point in the overall orbit averaging, but for the most part they do not have the wild swings often produced by other methods.

The constrained subtraction method is somewhat slow because of the relatively large amount of searching and attendant closed orbit calculations, especially if the permute or drop mode options are used. For reasonable search patterns, and with debugging code, a 24 point Booster data set and orbit takes about 9 seconds per harmonic on a 100mips workstation. From about 15 to 25 closed orbit passes are involved, which is perhaps a little larger than needed once convergence is established. For someone in more of a hurry, most of the closed orbit passes can be done in parallel on a suitable parallel computer.

### 3. Results

This procedure works rather well for the rather small and regular Booster lattice. It is not yet at all clear how general it may be, as it needs a lattice understood well enough to yield a closed orbit when modeled. The example calculations have not yet been exercised with markedly non linear conditions. It can probably be argued that both the choice of machine and of the data sample used to develop the codes plays to the particular strength of the method. One or two bad data points cause the phases of the various harmonics to line up to satisfy the excursions at these points, which dominate the penalty function. Similarly where the phases are out of alignment, the orbits are better behaved. Yet when offending points are removed, the phases of the principal harmonics remain about the same. For the rather dirty Booster orbits considered here, there is always a clean minimum in both  $A_j$  and  $B_j$  as the cycling is carried out. The results do not cluster about the end points of the constrained ranges of  $A$  or  $B$ , as if a preferred result were outside of the range. The order in which the  $A_j$ ,  $B_j$  are evaluated and subtracted off does not seem to matter. When apparent bad points are skipped, the scheme produces even smoother results. The scheme is insensitive to missing points, and so far has not encountered the artifacts of the usual Fourier analyses when data is missing and patterns are broken. The resulting corrector patterns are balanced, and the changes have little effect on tunes. More elaborate trials with a wider range of data and afflictions are clearly needed, but these preliminary results are rather encouraging.

Several representative screen plots obtained by analyzing data from an AGS Booster orbit snapshot are shown in Figures 1 - 5. These examples are intended to illustrate the methods rather than to offer conclusions from the data sampled. This particular data, from an off momentum orbit of 1993, was obtained to test the extraction of the proper momentum from orbit data back when these things still worked. It is quite likely that something in the C8 region is turned on and didn't make it into the model, or maybe the data point itself is off by a centimeter or so. The **Pvcorrs** drawing command displays amplitudes and phases of individual harmonics, and the buildup in the form of curves of sums of harmonics  $J_{min} \rightarrow N$ , as  $N$  is increased from  $J_{min}$  through  $J_{max}$ . Monitors are missing at D6 and F6, and the point from monitor C8 is rather obviously suspect, which further strains the resolving power. Calibration studies using the dropped point mode also cast doubt on the A2 data point. Correlated distortions do appear to suggest a noticeable second and third harmonic in this particular set of readings. Spurious harmonics are weaker in similar curves run without the C8 entry, but such efforts to clean up the data are not considered further here. The seventh harmonic, perhaps an artifact of the fitting, is often a useful measure of the resolution and noise in the method. The apparent  $J = 7$  amplitude here is spurious, because it has little effect on the fit, and because the spacing of the correctors limits the efficiency with which it is expressed.

The Figures present the results of an exercise which calibrates the model in terms of a specific snapshot taken at a momentum offset of about  $- .0015$ , and moves the orbit from the snapshot values to a centered flat one, operating in the X plane. In Figures 1, and 2, all points of the sample are used, including the dubious A2 and C8 ones. In Figure 3, calculations similar to those of Figure 2 with these two points removed is shown. The program doesn't seem to mind the difference. These results are for a single iteration. A second fitting pass removes completely the remaining trace of the mainly fifth harmonic.

In these Booster data samples, the harmonic analysis averages rather broadly as the orbit is measured at essentially regular intervals, rather widely spaced with respect to dimensions of likely sources of orbit distortions. But as noted before, in one sense the Booster is perhaps an extreme case in that the number and spacing of monitors barely covers the range of orbit harmonics present in the machine. Details of finer structure are obviously limited by the number of sampling points. (Figure 1) A few X plane monitors between these intervals, perhaps by adding an X coordinate to several of the Y plane monitors, would be very helpful.

A subset of statements copied from a larger test file which applies the harmonics commands discussed here is appended, and described in Section 4 below. The Figures are generated from these inputs as well. Parts of the resulting PRINT file of the **Vcorrs** fits to this sample data are also included, and described in Section 5.

#### 4. Command File for Vcorrs

A partial set of **Vcorrs** and related commands from a library file of MAD input statements is discussed in this section. This information, together with a plot description file, should be adequate for running the corrector based features of this part of the program. While complicated, as a library of useful commands the file is readily edited for new applications. This style permits tailoring of conditions and graphics from input statements, rather than hard coding the morass of details about tuning computations and what is to be drawn. The various array names used here are generally enclosed in quotes, making them upper / lower case sensitive, to avoid accidental conflicts with lattice element names. This example fits the calibration to 20 Booster orbit data points of the 22 Bpms available, dropping the points at A2 and C8. A data point is dropped by setting its corresponding weight to zero. (*Wgts\_C20* array) The program reads the listed commands in store mode, that is it parses and stores them in the data base, but does not execute them until called to do so after the library has been assembled. Drawing commands for each case are included in the picture list, and called from menus by the *T.plx Pcalls* screen manager command. Details of the **Vcorrs** fitting and **Pvcorrs** drawing commands are given in later sections.

##### 4.1. Calibration Command Group

The example **Vcorrs** command here works with lists of the actual 24 bpm and 24 corrector names, and corresponding weight and factors arrays defined as MAD **Vectors**. Both arrays are given the same typical values for all valid elements. These values are the same as the program defaults. For convenience, our calibration corrector set is taken to be the actual Booster corrector set. Alternatively, a finer grained virtual corrector set with zero lengths could have been defined and inserted in the lattice description.

As this part discusses a calibration run, orbit measurement data in an array called *X\_Data* is needed, and so the array is listed here. Bpms are missing for the D6 and F6 positions, but entries are filled in anyhow by the controls operation that records orbit displacements, a convenience to preserve the general symmetry of the lattice naming conventions. The two false entries are marked by large values, which are rejected by the *Filter = 999.* feature. This data is in millimeters, and so the *Convert = .001* feature is applied to get the data into meters. The names of the attributes of the quantities to be read from the bpms and to be set in the correctors are given as *Bparams = /X* and *Cparams = /Dpx*; these could be omitted as they are the same as the defaults. These particular attributes are specialized cells of BNL "Fast" version monitors and correctors that can be read or set dynamically while the program is doing tracking. A range of harmonics from 1 through 9 are to be fitted. The total length of the orbit *Smax* is given, along with typical *Cycles* values for number of iterations, number of steps in searching for minima, and number of passes over decreasing search regions.

Also included in the group is an **FTwiss** command to initialize and perform closed orbit calculations for the fitting, and commands to save, archive, and retrieve fitted corrector and momentum offset results needed by later activities. When applied to the lattice elements involved in a closed orbit calculation, these results should reproduce the original measurements to within some error. The presumption is that someone will review the calibration results, and perhaps even repeat them with dubious data points removed, before applying them. As a calibration in principle can be used for a number of further correction adjustments, calibration results are preserved separately. In this example, the perhaps cumbersome notation for commands and saved vectors and files applies the suffix *C20* to note a calibration using 20 data points.

The *Cal20* subroutine group combines these stored commands to carry out the calibration, and saves results as files suitable for running in the actual correction mode. The detailed printing from **FTwiss** should be shut off, as closed orbit tracking passes will be made at the beginning and ending of the fitting to monitor tune shifts.

##### 4.2. Correction Command Group

The run mode finds the changes needed to move the orbit from the one calibrated in the model to some other one, given as recorder values held in the *Goodorbit* vector. In this example the goal is a flat orbit with all values 0. at the bpms. Actual bpm data are not needed for this move. The lattice calibration corrector family and momentum offset are set initially to the calibration values, using **Retrieve** and **VtoD** (Vector to lattice family Data) commands defined in the groups. The momentum offset value from the calibration is largely useful for an initial **FTwiss** pass that reproduces previous conditions. The offset most



likely will be changed as the orbit is moved to the desired one.

Data conversion and filtering is generally not needed for the **Vcorrs** command in the run mode, unless actual data for the intended orbit warrants them. Results to be saved optionally for later screen plotting are noted: the table of coefficients *Brun\_R20*, and the momentum offset *P\_savR20*. The coefficients of the pattern on the correctors are also in the table.

The *Run20* subroutine group combines these library entries, and produces a listing of the corrector and momentum offset changes that yield the given orbit changes. It also saves these results as archived files, suitable for input into the drawing commands.

### 4.3. Drawing Commands

The **Pvcorrs** drawing command draws curves from the coefficients computed during the fitting, stored in a table, and saved for displays. As the fitting is usually slow, the drawing is done separately from the preparation of the table. The same command can draw both individual amplitude and synthesis curves. Each refers to a **Vcorrs** command to find the table, and organize the arrays and other features needed by the original fitting command to prepare the table. These operations need the coefficients table *Brun\_R20*, and the fitted factors *F\_savR20* which relate the effect of correctors changes to orbit displacements. The table carries the fitted momentum offset. The operations further depend upon the *FTW\_Drw* **Ftwiss** command keyed to the *DrawC6* **PldrawR** vector drawing command. The vector drawing command works within an **Ftwiss** run, accepting an array of orbit variables after each lattice element. Entries in the plotter description file are keyed to these variables, through the *Cdef6* **Curvedef** command and its descriptive arrays, which handle selection of variables, colors, line types, and plotting factors and offsets.

At the very end, the *T.plx* **Pcalls** screen plotting manager is called to carry out the plotting via various menus and elaborate description arrays from a separate plot command library file not considered further here.

## 5. Vcorrs Outputs

Sample outputs are given using the commands noted above, for both calibration and correction runs. The two corrector families need not be the same for calibration and correction runs, although they are here. This extravagance with paper is mostly to give the reader a proper feel for the stability of the method. All data points are included in these results. Listings begin and end with a Twiss run to show closed orbit summaries for starting and fitted orbits.

A short table gives the progress of the fits as the several harmonics reduce the least square penalties. As the computations involve a number of closed orbit calculations which can be slow, this table is also printed on line to give some hint that the program is doing something. The table gives the fitted amplitudes relative to the bpm data, the current penalty, clock time, the reduction of penalty due to the current harmonic, and the number of closed orbit passes needed to reach a tolerance. The progression rather dramatically picks out the relevant harmonics, and sizes are bounded for the harmonics which are expected to be small, or noise.

Next a *Fit Summary Table* lists the fits to the measurements in the case of calibrations, or to the requested orbit in the case of applying corrections. The *V Initial* column gives the Twiss computed values from the uncalibrated program in the case of a calibration run, or the computed values from the calibrated model before adjusting to the requested orbit. The *Adjusted* columns are intended to measure any left over offsets after the harmonics have been fitted, perhaps a measure of noise, or asymmetric bpm data sampling.

The *Correctors Summary Table* has columns giving the final corrector values, the initial ones, and the differences, presented as angular kicks in radians. The *Change* column is an absolute adjustment to the real correctors computed by the fitting.

## 6. The Vcorrs Command

Like its predecessor **Vharm**, the **Vcorrs** command analyzes data arrays expressed as MAD **Vectors**. Vectors are identified by name in the inputs. Several options describe the range of sampling and outputs. In the *Permute* mode the fits are repeated with different sequences for the subtraction of the several harmonic contributions. In the *Drop* mode the fits are repeated in a sequence which drops one input point at

each step. The drop feature offers a way to examine potentially bad monitors, aid in data selection, and explore sensitivity of fits to particular monitors. Data selection can be obtained dynamically with the filter option and selective weighting.

The command accepts sets of measurements or program output expressed in vector format, at given named lattice monitors at locations *S*, with optional weights. The calibration mode expects one or two data sets, nominally *X* and *Y* orbit displacement values at their respective sampled locations. If wanted, data from both dimensions can be treated by one command. A range of harmonics consistent with the number of data points may be selected. In the *X* plane, instead of the fixed term, the momentum offset is always computed, and the program adjusted accordingly. The fits can be iterated to remove the biasing effect of ostensibly missing data upon the fixed term, if experience suggests that this effect is significant. In successive iterations, missing data points are computed from the fitted results of previous iterations. Output is to printed tables of inputs and fits, and optionally to an internal harmonic table which can be directed to harmonics synthesis plots, and individual harmonics plots.

In the following, the attributes are grouped into categories of name lists and vector arrays, program options, harmonic selections, and output options. The essential information includes the names of the lists holding names of the monitor and corrector families, and measurement data. Names are still limited to eight characters, and monitors and correctors must be uniquely named. Emplacements of both monitors and correctors are taken from the lattice used for the orbit computations, and so are not needed as inputs. Most other quantities have defaults adequate to get the calculations started. Some of these details will change with further experience.

As the families of bpms and correctors can differ for calibration and correction runs, separate data groups may be involved. Normally both groups work off the same lattice description, which can be quite littered with parameter expressions, some of which are likely to depend on the momentum offset. Hence the passing of calibration results to later runs can involve considerable complication in lining up the various parameters. There is a further possible confusion in that correctors are reset from the user supplied *KICK* attribute at the beginning of a closed orbit pass, but operate from the derived internal *IDpx* and *IDpy* attributes once initialized.

## 6.1. Input Data Vectors

**Bpms(2)** The *Name(s)* of simple list(s) of data point label names, *X* and / or *Y*. ( **List**, **Menu**, or **Xmenu** form) Normally these are names of members of a family of monitors, but any family of similar elements that yield a common kind of measurement may be submitted. These names are also used to label entries in least squares fit tables. Null entries may be included to maintain systematics and habits of listing. All monitor related arrays, such as measurements, weights, and good orbit values, must have the same length and indexing as this defining *Bpms* list.

**Bpmdata(2)** The *Name(s)* of simple array(s) of measurements, for example, a set of bpm readings, needed to define the calibration standard in the calibration mode only. *X* or *Y* (or both) must be given, and the number of entries in the vector(s) must agree with that of the *Bpms* list. The array of decimal values is defined as a **MAD Vector**. Units should be in meters, but conversion to meters can be arranged with the use of the *Convert* attribute noted below. Erratic input values can be rejected by the use of the *Filter* attribute noted below.

In an automatic orbit controller application, each logical bpm would normally be connected to a practical filter, such as a MAD filter neuron. The differences between the current filtered signals and the calibration values amount to error signals, that inspire a response as computed by the **Vcorrs** acting as a controller.

**Goodorbit(2)** The *Name(s)* of simple array(s) of decimal values, for example, a set of readings to be achieved at the bpms, needed to define the desired orbit in the run mode only. (**MAD Vector** form). The default is a flat orbit with 0. displacement. The number of entries in the vector(s) must agree with that of the *Bpms* list.

- Weights(2)** The *Name(s)* of simple decimal array(s) of weights. (MAD **Vector** form). Optional. If individual weights are used, each weight should be the inverse square of the nominal mean error for the measurement, typically .001m. Printing formats are centered about assumed residuals of the order of .001m, and least squares near unity. The default of 1,000,000. will be provided for weights if no weight vectors are given. Setting a weight to zero is a way to remove a data point from the calculations. This is probably more important in calibration than running, as there is little point in blanking out a value when prescribing a good orbit.
- Corrs(2)** The *Name(s)* of simple list(s) of corrector names, X and / or Y. ( **List**, **Menu**, or **Xmenu** form) Normally these are names of members of the family of correctors, but any family of similar elements that yield a common kind of orbit change may be submitted. These corrector names are also used to label entries in least squares fit tables. Null entries may be included to maintain systematics and habits of listing. All related corrector arrays, such as factors and saved result values, must have the same length and indexing as this defining *Corrs* list.
- Corrdata(2)** The *Name(s)* of simple array(s) of settings, for example, a set of corrector kicks in radians. When the same correctors are used for calibration and correction, these data would be the calibration values for the actual corrector set, applied for running or drawing. Optional. (MAD **Vector** form.)
- Corrmax(2)** The *Name(s)* of a simple array(s) of maximum permitted settings, to match the *Corrdata* variables, units, and formats. Optional. (MAD **Vector** form.) May be helpful as a diagnostic. The fitting method inherently leads to rather small, balanced corrector values.
- Factors(2)** The *Name(s)* of simple decimal array(s) of factors. (MAD **Vector** form). Optional. Factors are used internally to translate coefficients based on orbit measurements into corrector settings. While the program will compute its own average factor for this translation from the data it is given, the use of individual factors allows for variation in response among the individual correctors. A value of 0. shuts off a corrector. The default is 1.
- Csaves(2)** The *Name(s)* of simple decimal array(s) to receive the values of the family of correctors determined by the command. (MAD **Vector** form) Optional, and normally just used for calibration outputs. The saved array(s) can be initial configuration input for the run mode of fitting, by means of the *Corrdata* attribute. Alternatively they can be loaded by a **VtoD** command into the lattice previously assembled by a **USE** command. The program will generate a new save vector if a named one does not yet exist. No defaults.
- If run in an on line analysis mode, with the calibration configured with virtual correctors, *Csaves* contains the actual corrector values needed to maintain the prescribed orbit, and presumably would be fed back to the control system.
- Psaves(2)** The *Name(s)* of simple parameter command(s) to receive the values of the fitted momentum offset Delta P. Optional. The fitted offset is also placed in the preamble part of the coefficient table. In the run mode, saving and archiving these offsets can be helpful for passing fitted results for plotting. The program will generate a new parameter module if a named one does not yet exist. No defaults.

## 6.2. Necessary Attribute

- Smax** A REQUIRED decimal number, given in meters. This is the total length over which the input measurement set has been sampled. *Smax* defines the full period over which harmonics are to be fitted. It is the length of a circular orbit when a set of bpm values is analyzed. Phases for each harmonic J at corrector positions S(i) are computed as:

$$Theta(j,i) = 2. * \pi * J * S(i) / Smax$$

### 6.3. Attributes with Defaults

**Bparams(2)** The *Name(s)* of the attribute common to each member of the bpm or similar family from which the program is to obtain computed orbit data. Defaults are */X* and */Y* for bparams. Bparams are defined as **MAD Monitor**, **Hmonitor**, or **Vmonitor** elements. The */X* and */Y* attributes are specialized monitor cells that record orbit information during each pass of a tracking program. Internal names are often in quotes prefixed by a slash character to be case sensitive, and to avoid name conflicts.

**Cparams(2)** The *Name(s)* of the attribute(s) common to each member of the X and / or Y correctors or similar family into which the program is to place computed settings. Defaults are */Dpx* and */Dpy* for Correctors. Correctors as used here are defined as the **MAD Hkick** or **Vkick** elements. In the BNL version the */Dpx* and */Dpy* attributes of these elements can be set dynamically by processes running within the program, such as orbit correction and feedback schemes.

Note: the user supplied *KICK* attribute for correctors only initializes */Dpx* and */Dpy*, but is not further involved in any changes during tracking runs.

**Twiss** The *Name* of a BNL **FTwiss** command to be employed for the model calculations. The default will do, unless initial orbit conditions are to be prescribed.

**Constraint** A decimal number which constrains the choice of *Aj* and *Bj* coefficients of each trial according to

$$A_j^2 + B_j^2 < \text{Constraint} * \text{Initial\_Amplitude}^2$$

Ideally the value should be about 1., and the supplied default is 1.2. In these studies, the constraint is seldom tripped, and then only for background harmonics that the correctors are rather ineffective in expressing. The search routines give a fairly detailed diagnostic if an apparent penalty minimum lies outside of the constraint, suggesting adjustment.

**Dropmode** Logical Flag: If True, the fitting will be iterated, dropping a different input reading on each successive pass. Output is not stacked, so the *Htable* holds only the last computation. These *Drop* and *Permute* options are useful for breaking in the program to a new problem or configuration, to establish sensitivity and confidence levels, and to identify questionable data points during calibrations. They are both rather time consuming for routine use. Computation time increases as the number of points in the measurement sample.

**Permute** Logical Flag: If True, the fitting will permute the order of cycling harmonics. Computation time increases as the number of harmonics in the sample.

**Calibrate** Logical Flag: If True, the program will operate in calibration mode. Otherwise, in run mode.

**Filter** A decimal number which should be used to screen accelerator produced data such as monitor readings. Measurements greater than the absolute value of *Filter* will be rejected for fitting. Takes same units as, and applied only to *Bpmdata* and *Goodorbit* input arrays.

**Cycles(3)** 1. An integer giving the number of iteration cycles, which tend to adjust the fixed delta P result, and may influence others. Useful in reducing a harmonic that persists after a previous iteration. Successive iterations may shake up the fitting, producing more attractive fits, which may, however, differ little in final least squares criteria from others. The proper value should be obtained by trial, viewing the listing which traces the penalty function, hopefully to some convergence. The default is set for 1, to restrict inadvertent long runs, but is probably too small for most cases.

The method computes a fit once without the missing points, and subsequently includes values derived from the previously fitted function at the missing points to compute the next iteration. For the AGS Booster, the contributions to the offset from the two

missing monitors out of 24 tend to balance each other.

2. An integer giving the number of steps into which the span between the upper and lower constrained amplitudes is to be divided. Default is 8. The method of search favors about this number of steps, as searching moves in the direction of lower penalty results until an increase is obtained.

3. An integer giving the number of internal cycling passes used in searching the local grid for the best  $A_j$ ,  $B_j$  pairs. Default is 5, which has proved adequate for Booster work. The program stops cycling earlier if the relative least squares improvement between cycles is less than .001, which is usually smaller than the data warrants. Each extra pass takes at least six closed orbit calculations, so there is some incentive to keep this attribute small.

**Convert** An optional decimal number, that if present, is used as a conversion factor for all *Bpmdata* and *Goodorbit* input array values, and the optional *Filter* attribute. (Orbit measurement data)

#### 6.4. Harmonic Selection Options

Harmonics are indexed here by the letter J. In the X plane, the lead term,  $J = 0$ , ( $A_0$ ), is treated as 0, and the momentum offset fitted instead.  $A_0$  is not included in the following J ranges.

**Jmain** An optional integer. The number of the central harmonic to be fitted, if *Jmin* and / or *Jmax* are not given. *Jmin* and *Jmax* will be defined in terms of the number of measurements in the *Bpmdata* as

$$J_{\max} = J_{\text{main}} + j_x$$

$$J_{\min} = J_{\text{main}} - j_x$$

$$N_{\text{meas}} > 2 * (J_{\max} - J_{\min}) + 1$$

**Jmin** An optional integer. The lowest harmonic to be fitted. If given, *Jmax* will be constrained so:

$$N_{\text{meas}} > 2 * (J_{\max} - J_{\min}) + 1$$

**Jmax** An optional integer. The highest harmonic to be fitted. If given, and *Jmin* is not given, *Jmin* will be constrained so

$$N_{\text{meas}} > 2 * (J_{\max} - J_{\min}) + 1$$

The program may reduce the range *Jmin*, *Jmax* actually used to agree with the number of data points accepted after filtering and non zero weight criteria have been applied. For these cases *Jmax* is adjusted downwards. The use of these J attributes provides alternatives, such as dropping the low order harmonics instead.

#### 6.5. Output Options

**Htable** The *Name* to be given to the internal table that receives the Fourier (harmonic) coefficients and momentum offset fitted to the measurement vectors. Other analyses and display programs need to refer to this table by its name. The table can hold both X and Y fits, consistent with the usual MAD harmonic features. The table will be saved for possible further use only if an *Htable* name is given.

Run tables should be archived for reuse, as they may take a minute or more to compute.

**C\_save(2)** Already noted, the *name(s)* of array(s) to receive the corrector settings from calibration runs, intended as input to the lattice used for succeeding correction runs.

**P\_save(2)** Already noted, the *name(s)* of parameter definitions to receive the fitted momentum offset from correction runs, intended as input to succeeding plotting sessions.

**Print** Logical Flag: If True, a table of harmonic coefficients will be printed.

**Verbose** Logical Flag: If True, an extended trace of fitting progress will be printed. This trace can be helpful for evaluating sensitivity.

## 6.6. Internal Attributes

As currently patched together, a number of temporary arrays are involved in a **Vcorrs** run, many of which are also needed to produce display plots which follow the orbit correction process. Several plotting commands may be called by menu action during the same run, each tied to a particular **Vcorrs** calculation with its own ensemble of lists and arrays. All of this dependency has to be recorded, and provision made for swapping this information among the commands and the program when a particular command is called from the display manager. The program attaches the necessary pointers and working values to each parent **Vcorrs** command. This section gives the internal storage cells of the command layout, which perhaps can be useful when dumped for debugging purposes. (*A program user does not have to supply the following material when preparing the command.*)

- Atwiss(2) Internal name group with pointers to **FTwiss** command used by this **Vcorrs** command.
- Ahtable(2) Internal name group pointing towards harmonic table, either archived or produced by **Vcorrs**.
- Ablist(4) Two internal name groups, for X and / or Y lists **BL** as given, pointing to original *Bpms* list.
- Abdata(4) Internal name groups pointing to originally supplied *Bpmdata* vector(s) **BD**, if any.
- Ab0data(4) Internal name groups pointing to internally generated vector(s) **B0** holding originally computed model orbit bpm data.
- Abpos(4) Internal name groups pointing to internally generated vector(s) **BP** holding bpm lattice position data.
- Abvect(4) Internal name groups pointing to internal vector(s) **B** used to gather monitor values from Courant - Snyder - Twiss calculation of closed orbit model.
- Aclist(4) Two internal name groups, for X and / or Y lists **CL** as given, pointing to original *Corrs* list.
- Acdata(4) Internal name groups pointing to originally supplied *Corrdata* vector(s) **CD**, if any.
- Ac0data(4) Internal name groups pointing to internally generated vector(s) **C0** holding original corrector settings, from original lattice configuration, or as supplied in *Corrdata*.
- Acmax(4) Internal name groups pointing to originally supplied *Corrmax* data vector(s) **CM**, if any.
- Acpos(4) Internal name groups pointing to internally generated vector(s) **CP** holding corrector lattice position data.
- Acvect(4) Internal name groups pointing to internal vector(s) **C** used to place computed values in correctors for Twiss calculation of closed orbit model.
- Addata(4) Internal name groups pointing to internal vector(s) **D** derived from *Bpmdata* vector(s) **BD** under calibration mode, or from *Goodorbit* vector(s) **G** under run mode, with suitable conversions if needed.
- Afdata(4) Internal name groups pointing to given *Factors* vector(s) **F**, or equivalent internal default vectors produced by the program.
- Agdata(4) Internal name groups pointing to given *Goodorbit* vector(s) **G**, or equivalent internal default vectors produced by the program.
- Arvect(4) Internal name groups pointing to internally generated vector(s) **R** holding model computed orbit position results.
- Atvect(4) Internal name groups pointing to internally generated vector(s) **T** holding orbit position residuals: given or measured - computed.
- Auvect(4) Internal name groups pointing to internally generated vector(s) **U** holding corrector settings prior to current trial in fitting coefficient. Initially each **U** is set to values from a **C0** vector, and incremented with contributions from best fit coefficients for each J.

Avvect(4)	Internal name groups pointing to internally generated vector(s) <b>V</b> holding a constraint function, such as <b>D - B0</b> , used to limit amplitudes in search for best fit of trial coefficients.
Awdata(4)	Internal name groups pointing to given <i>Weights</i> vector(s) <b>W</b> , or equivalent internal default vectors produced by the program.
Acsvave(4)	Internal name groups pointing to given <i>Csave</i> vector(s) <b>Cs</b> , if any.
Apsaves(4)	Internal name groups pointing to given <i>Psave</i> parameter modules(s) <b>Ps</b> , if any.
Avgetb0(4)	Internal name group pointing to internally generated <b>DtoV</b> command used to fetch bpm values produced by initial run of model, and place them in a <b>B0</b> vector.
Avgetc0(4)	Internal name group pointing to internally generated <b>DtoV</b> command used to fetch corrector values at initial run of model, and place them in a <b>C0</b> vector.
Avgetb(4)	Internal name group pointing to internally generated <b>DtoV</b> command used to fetch bpm values produced by runs of model, and place them in a <b>B</b> vector.
Avputc(4)	Internal name group pointing to internally generated <b>VtoD</b> command used to place computed values in <b>C</b> vector into correctors before runs of model.
Avputc0(4)	Internal name group pointing to internally generated <b>VtoD</b> command used to place initially submitted corrector values into lattice from <b>C0</b> vector.
/wfave(2)	Decimal working values of average given corrector factor. (X, Y)
/wconvert	Decimal working value of conversion factor, given or default.
/wfilter	Decimal working value of filter limit, given or default.
/wdeltap	Decimal working value of momentum offset Delta P from fitting.
/nc(2)	Integer value of number of available correctors. (X, Y)
/nv(2)	Integer value of number of available bpms. (X, Y)
/ifreq(2)	Integer pointers to frequency column of temporary internal table. (X, Y)
/icos(2)	Integer pointer(s) to start of internal table(s) of $\cos(2\pi j \cdot si / smax)$ (X, Y)
/isin(2)	Integer pointer(s) to start of internal table(s) of $\sin(2\pi j \cdot si / smax)$ (X, Y)
/ijdim(2)	Integer increment(s) by point in cos / sin tables. (X, Y)
/jjdim(2)	Integer increment(s) by harmonic in cos / sin tables. (X, Y)
/imin(2)	Integer minimum harmonic(s) to be considered. (X, Y)
/imax(2)	Integer maximum harmonic(s) to be considered. (X, Y)
/jml(2)	Integer starting pointer(s) in drop mode. (X, Y)
/jrm2(2)	Integer ending pointer(s) in drop mode. (X, Y)
/jmain(2)	Integer central harmonic(s). (X, Y)
/wniter	Integer working number of iterations, given in <i>Cycle(1)</i> , or default.
/wndivs	Integer working number of divisions in searching for individual harmonic, given in <i>Cycle(2)</i> , or default.
/wnpass	Integer working number of passes in searching for individual harmonic, given in <i>Cycle(3)</i> , or default.
/jdeltap	Integer pointer to Delta P value in attached FTwiss command.
/flags(2)	Logical flags indicating X and / or Y fits and data active.
/lsqc(2)	These two cells receive the final X and Y values of the least squares penalty function for the fit(s). If more than one cycle has been performed, results from each cycle overwrite those of the previous, so only the last cycled values are preserved. As a part of each <b>Vcorrs</b> command module (object) in the data base, these values may be reached by any of the data logging, parameter change, and display facilities of the program.

/status        An internal cell used to mark the command status after the last use of the command.  
 /stamp        An internal cell used to mark the last use of the command.

## 6.7. Contents of Coefficient Table.

All table fields are in 8 byte decimal. Each row of the table has the following fields.

Nx            Number of harmonic, X fit.  
 Abx           Combined Aj and Bj amplitudes for X, square root of sum of squares.  
 Ax            Aj for X, in monitor coordinates.  
 Bx            Bj for X.  
 Ny            Number of harmonic, Y fit. Because of possible differences in the data, these may not all agree with the corresponding Nx.

Aby, Ay, By

Freqx        The ratio of the harmonic Nx \* twopi to the total length S.  
 Nabx        Abx, normalized so the largest entry among the harmonics is set to 1. Historical.  
 Nax, Nbx    Similarly for Ax, Bx.

Freqy, Naby, Nay, Nby

Fcorrx       The ratio of a small increment in Aj on the correctors to the change in orbit produced by the change.  
 Cabx        The amplitude of the harmonic on a corrector, for example a kick in radians.

Cax, Cbx

Fcorry, Caby, Cay, Cby

The header of the table has a preamble which is used by plotting and other services.

1.        Name of Vcorr command creating table.
2.        a). Type code: 1 = Vharm single fit, 2 = Vharm stacked table, 3 = Vcorr calibration, 4 = Vcorr run.  
           b.) Number of groups, > 1 for stacked table.
3.        Initial Delta P.
4.        Fitted Delta P. (X Plane)
- 5.
6.        a). Kind of Fit for Vharm: 1 = Constrained, 2 = General.  
           b). Length of a table row: 24 fields for Vcorr, 16 fields for Vharm.

## 7. Plotting

Plots displaying orbit correction by harmonic fitting are available through the **Pcalls** plot manager command. **Pcalls** is a Silicon Graphics based menu driven display, that can also be connected to orbit computations and display commands, and to utilities that enable parameter changes and comparisons. Curves computed from tables (*Htable* attribute) of harmonic coefficients produced by **Vcorrs** commands may be displayed by the **Pvcorrs** drawing command, which may be attached to the plot manager. Pictures may show individual harmonics and the buildup of the fitted function as harmonics are added. This command resembles a similar **Pvham** command used for displaying harmonic analysis of the contents of a vector. Original orbit values in the measurement vectors, Twiss generated orbits, lattice schematics, and orbits computed from the harmonic fits may also be displayed on the same picture. The program may also be compiled with an alternative X Windows based version for similar graphics.

Rather than create a series of larger processes that depend upon similar computations, many of the BNL MAD analysis and drawing commands work by linking together more familiar computations. The **Pcalls** manager works through a **Plotdef** command and its related picture list. The picture list holds the names of all drawing commands and their associated **Curvedef** commands and retinue of descriptive



arrays. This supporting material is usually held on a separate call file, handled as a command library. **Pcalls** commands present a selection panel with a number of buttons, of which the *Pick* button presents a pop up menu holding the kinds (classes) of drawing commands submitted to the picture list by the program user. By selecting on a class, a further submenu is presented with the names of all drawing commands of that class, such as **Pvcorrs**. Selecting on one of these names leads to a corresponding **Pvcorrs** menu or drawing.

A number of drawings may share the same tracking and drawing commands and setups, presumably with considerable opportunities for producing mischief as well. Many of the drawing features push parts of the program rather further than they were designed to go. For example, values for corrector families may be changed by the drawing commands involved. If wanted, there are ways to restore lattice conditions by attaching appropriate set up and restore subroutines to the **Pcalls Command** list, which will display them for selection on a command submenu. In time, the more offensive shortcomings will be eased.

### 7.1. Pvcorrs Attributes

Vcorrs	The <i>Name</i> of a <b>Vcorrs</b> command configured to produce a table of fitted harmonic coefficients or operate from an archived one.
Jharm	Integer noting a single harmonic to be displayed. Optional. If not given, the range of harmonics fitted by <b>Vcorrs</b> will be displayed. Ignored on <i>Menu</i> option.
Menu	Logical Flag: If True, each harmonic to be displayed will be selected from a pop up menu.
Masks	The <i>Name</i> of an optional integer array of masks (MAD <b>Mvector</b> form), keyed to harmonics. In non menu mode, if a mask array is given, only those harmonics for which the corresponding mask is greater than zero will be drawn. The drawing routine uses a drawing server directly attached to the <b>Ftwiss</b> command which computes orbits from the harmonic data. This server operates from a vector containing all 18 variables derived by the Twiss tracking at each element position. The variable(s) to be drawn, such as the closed orbit optics or displacements, are noted by positive values in an integer mask array keyed to the wanted variables. (ex. $X = \#11$ ) All of the other <b>Curvedef</b> plotting service arrays are normally keyed to these variables in the Twiss vector, and so are not available to be keyed also to a particular harmonic. Hence an extra masking feature is provided for selecting out particular harmonics.
Xycode	A <i>Name</i> , X or Y, denoting which results to display from the harmonic table.
NoSums	Logical Flag: If True, the sums (synthesis) curves are not drawn. Can also be set from menu.
NoAmps	Logical Flag: If True, the amplitude curves are not drawn. Can also be set from menu.
Ypos(2)	Decimal values in pixels telling where the 1) amplitude and 2) sums curves are to be centered vertically, relative to inside (upper) edge of the lower border. Optional, but pictures can get very busy if both amplitude and synthesis curves lie about the same center.

Additional internal attributes are attached to the command stored in the data base. These items are supplied and used within the program.

Acdef(2)	Internal name group pointing to <b>Curvedef</b> command, which is linked through a <b>PldrawR</b> command attached to a <b>Ftwiss</b> command. The program traces all of these links.
Adefault(2)	Internal name group pointing to default <b>Pvcorrs</b> command.
Avcorrs(2)	Internal name group pointing to attached <b>Vcorrs</b> command.
Amasks(2)	Internal name group pointing to attached <i>Masks</i> vector, if any.
/ay(2)	Internal decimal cells giving actual values for <i>Ypos</i> in plotting coordinates.

/status, /stamp

## 7.2. Pvcorrs Menu

The *Menu* option presents a pop up menu with the following panels. The first three display the currently selected harmonic *Jharm*, and the lowest (*Jmin*) and highest (*Jmax*) available. The set and change panels only modify *Jharm*; curves are produced only when the *Return* or *AllJ* panel is picked.

<i>Jharm</i> =	Shows current value.
<i>Jmin</i> =	Shows <i>Jmin</i> value.
<i>Jmax</i> =	Shows <i>Jmax</i> value.
Quit Vcorrs	Leaves current drawing routine and returns to main menu panel.
All J	Leaves menu to draw all J within <i>Jmin</i> - <i>Jmax</i> .
Return	Leaves menu to draw current <i>Jharm</i> .
J +	Increases <i>Jharm</i> by 1. Stays in menu.
J -	Decreases <i>Jharm</i> by 1.
J = 0	Sets <i>Jharm</i> to 0.
J = <i>Jmax</i>	Sets <i>Jharm</i> to <i>Jmax</i> .
J = <i>Jmin</i>	Sets <i>Jharm</i> to <i>Jmin</i> .
Sums	
No Sums	A toggle that displays current state of <i>Sums</i> (Synthesis curves) option, and switches it to alternate state.
Amps	
No Amps	A toggle that displays current state of <i>Amps</i> (Amplitude curves) option, and switches it to alternate state.

## Documentation

Unix troff typeset format

<b>Host:</b>	<b>rapt.ags.bnl.gov</b>
This report:	/usr/disc2/jn/Docum+/Vcorrs.man
Vharm Report:	/usr/disc2/jn/Docum+/Vharm.man
Plcalls:	/usr/disc2/jn/Docum+/Plcalls.man
Vcorrs Test File:	/usr/disc2/jn/Docum+/Vcorrs.data
Vcorrs Results:	/usr/disc2/jn/Docum+/Vcorrs.print22

# Booster Orbit Fitting Correction Using Calibrated Model

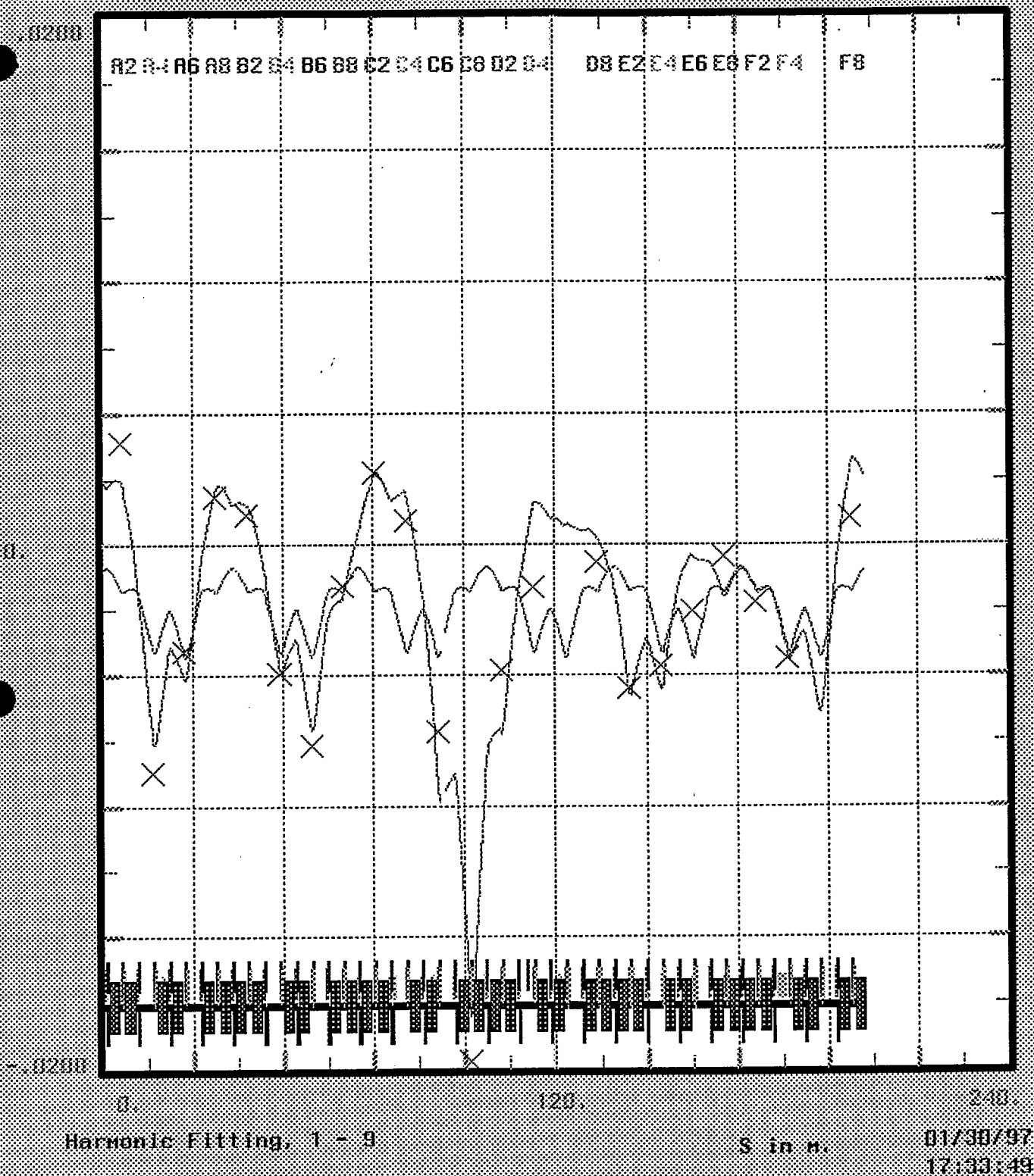


Figure 1.

The initial calibration fit to all data points produced by a Vcorrs command in the X plane. Data is from an AGS Booster orbit snapshot taken in April of 1993 with a marked momentum offset of about -.0015. Data points are shown as crosses. The inside, more regular curve is the X displacement produced by the usual bare orbit application of the model. It simply doesn't match the data very well. The average deviation here is about 7 mm per point. The second curve is the fit to the data produced by the calibration using the regular Booster corrector set. Nearly all of the penalty comes from points about A2 and C8.

# Booster Orbit Fitting Correction Using Calibrated Model

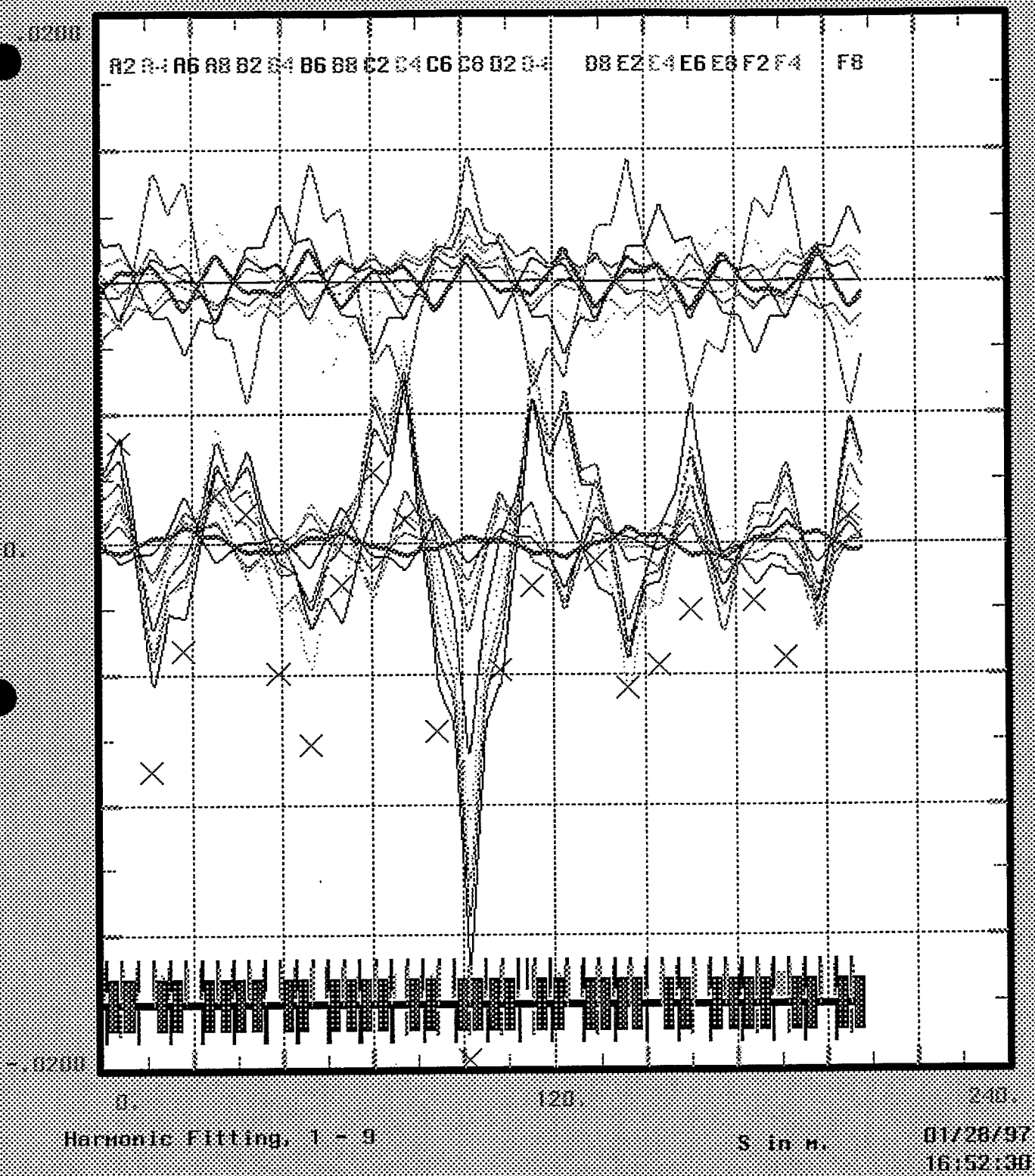


Figure 2.

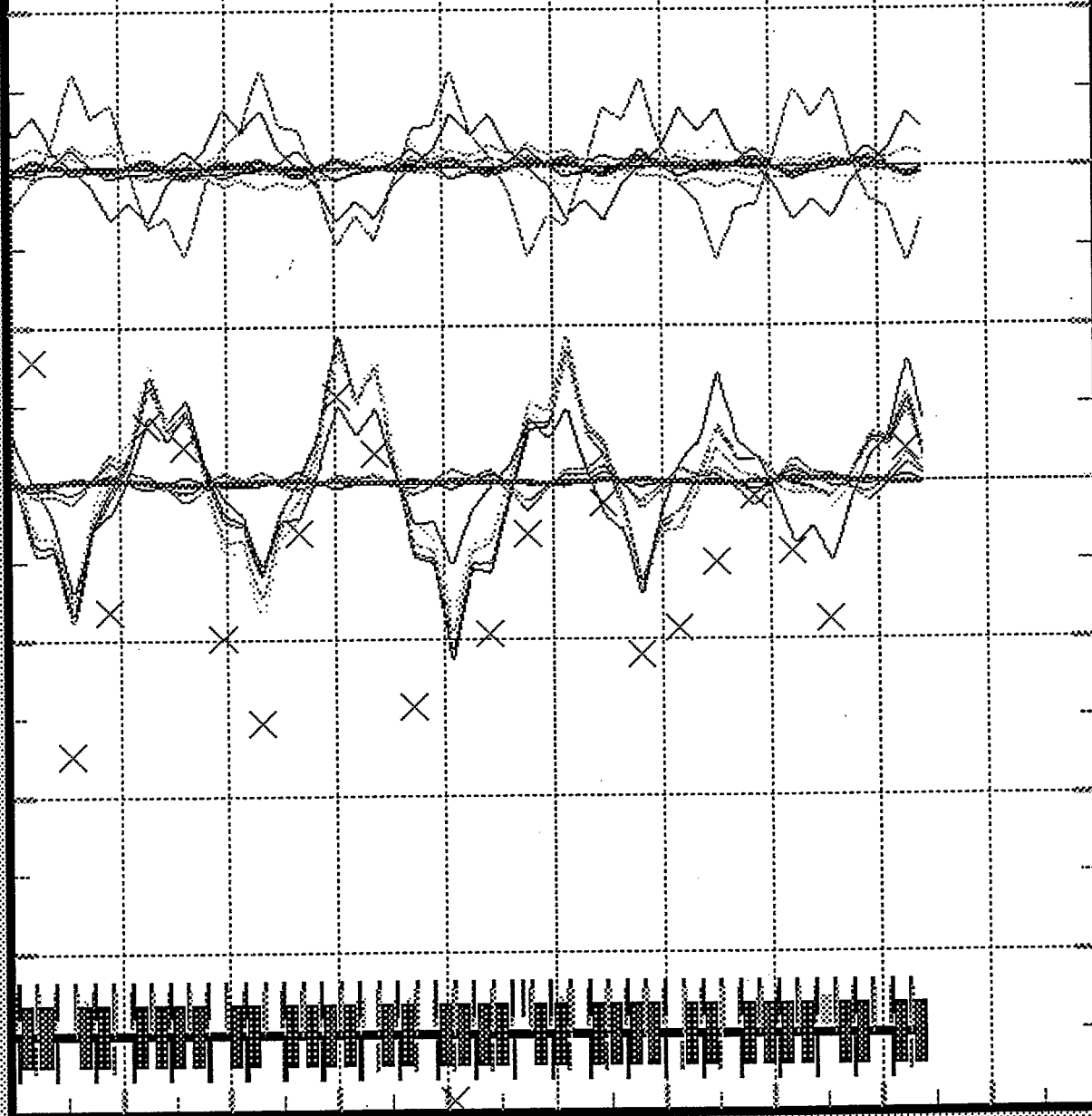
An example harmonics analysis plot produced by the Pvcrrs drawing command from coefficients fitted in a correction run applied to the original Booster orbit snapshot data. The program is asked to adjust momentum and correctors to produce a flat orbit in the center of the machine. The lower series of curves shows the buildup of the fitted function, in the form of the sum of the first J harmonic amplitudes, one curve each for J = 1 through 9. (19 Coefficients from 22 Data Points) The heavier central solid curve is the final result, after J = 9 has been added. The remaining amplitude essentially vanishes with further iteration. Measurement data points are shown as crosses. Both scales are in meters. The original in color distinguishes much better among the harmonics.

The upper curves are individual harmonic amplitudes which contribute to the synthesis curves. J = 4

# Booster Orbit Fitting Correction Using Calibrated Model

0200

A2 A4 A6 A8 B2 B4 B6 B8 C2 C4 C6 C8 D2 D4 D8 E2 E4 E6 E8 F2 F4 F8



Harmonic Fitting, 1 - 9

S in n.

01/28/97

16:55:17

Figure 3.

Synthesis and amplitude curves similar to Figure 2, computed using a calibration from which errant data points A2 and C8 have been omitted.



# Booster Orbit Fitting Correction Using Calibrated Model

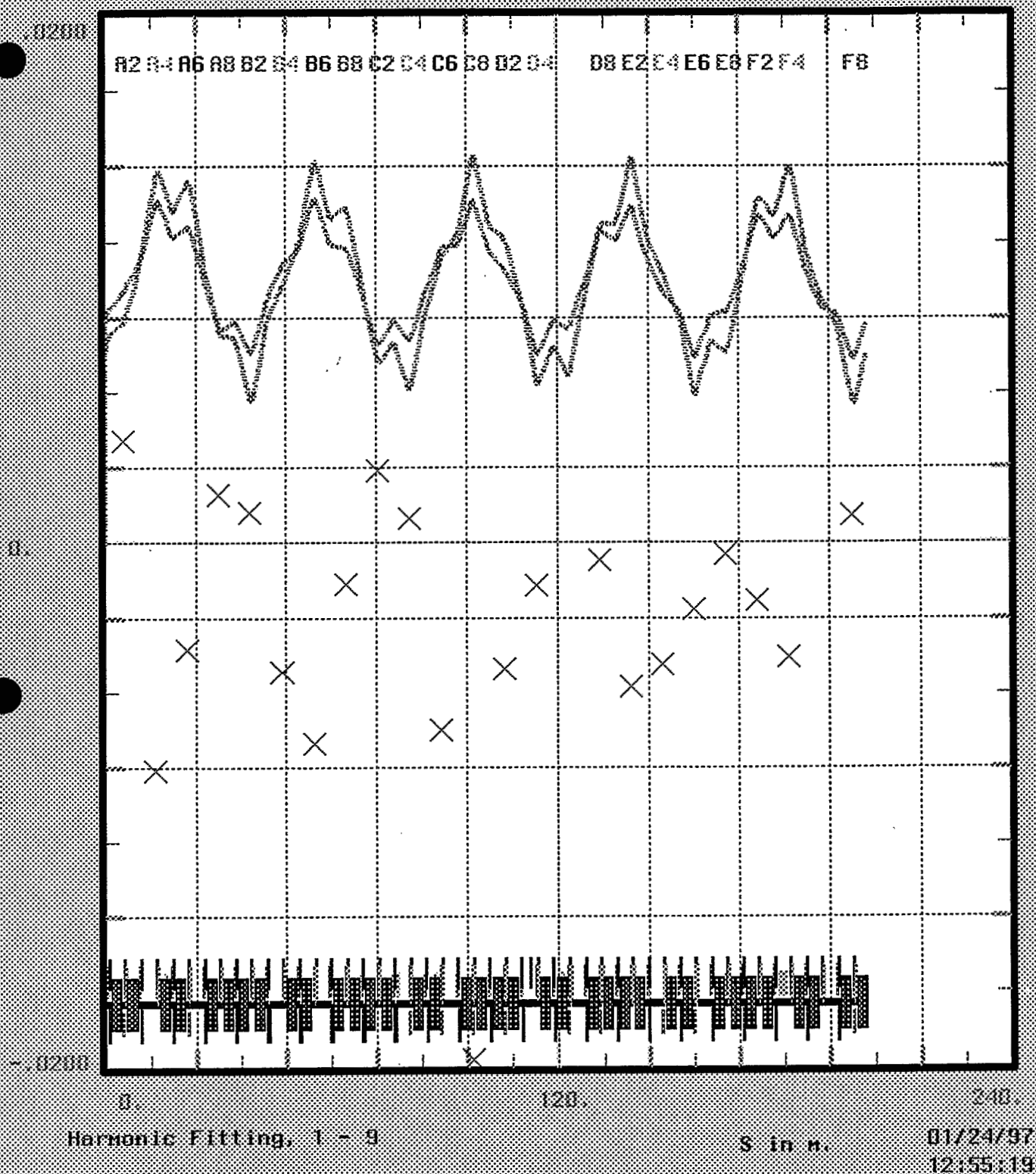


Figure 4.

Amplitude curves for harmonic  $J = 5$ , computed using calibrations with and without errant data points A2 and C8. The curves show the reduction of the orbit displacement caused by placing this fifth harmonic on the correctors for the two cases, for negligible momentum offset. The phase is not affected.

# Booster Orbit Fitting Correction Using Calibrated Model

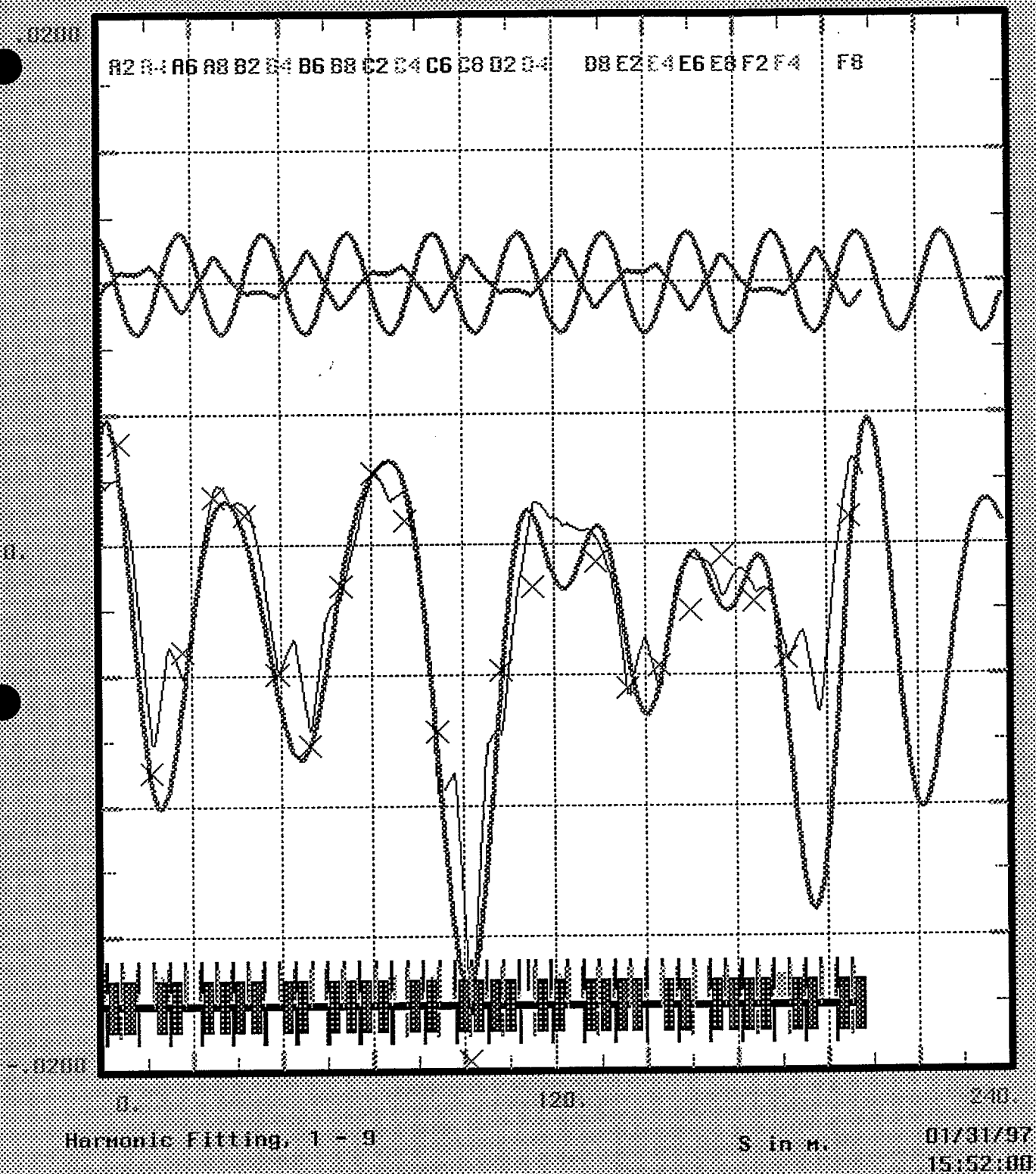


Figure 5.

Comparison of fits to measured data points by the constrained harmonic method (heavy curve) and the corrector calibration method of this paper (thin curve). The latter produces better detail, especially at the extraction region. The upper curves compare the apparent  $J = 9$  amplitudes. The correction curve is phase reversed relative to the plain fit.

# Vcorrs Command Library for Booster Runs

February 21, 1997

File = "/usr/disc2/jn/Docum+/Vcorrs.data"

Part of: "/usr/disc2/jn/Booster+/Hruns.boos"

\*\*\*\*\*

Pdelta Param = -.001462

Store

! Calibration Mode Command, Arrays, and Related Commands.

```
Boos_C20 Vcorrs Calib, Bpmdata = "X_Data", &
Bpms = L_XBPM, Corrs = L_XKICK, &
Weights = "X_Wgts20", Factors = "X_Facts", &
Print, Cycles = 1,10,3, &
Convert = .001, Filter = 999., &
BParams = "/X", "/Y", &
CParams = "/Dpx", "/Dpy", &
Twiss = FTW_Cal, Htable = "Hc_table", &
Csave = "C_savC20", Psave = "P_savC20", &
Jmin = 1, Jmax = 9, &
Smax = 201.78
```

```
"X_Data" Vector, &
3.956113 -8.693634 -4.038608 1.899860 1.212470 -4.886150 &
-7.630552 -1.532651 2.786204 0.946376 -7.146722 -19.704966 &
-4.802379 -1.614516 44444. -0.625031 -5.475378 -4.604772 &
-2.491268 -0.376475 -2.166401 -4.313535 44444. 1.058067 &
```

! Suppress A2, C8 for Calibration ==> 20 Points.

"X\_Wgts20" Vector 24 ^ 1000000., Value(1) = 0., Value(12) = 0.

"X\_Facts" Vector 24 ^ .01

FTW\_Cal Ftwiss, Deltap = pdelta

Arch\_C20 Archive "C\_savC20", "P\_savC20"

Rtrv\_C20 Retrieve Force, "C\_savC20", "P\_savC20"

\*\*\*\*\*

! Running Mode Command, Arrays, and Related Commands.

```
Boos_R20 Vcorrs Bpmdata = "X_Data", &
Bpms = L_XBPM, Corrs = L_XKICK, &
Weights = "X_Wgts", Factors = "X_Facts", &
Print, Cycles = 1,10,3, &
BParams = "/X", "/Y", &
CParams = "/Dpx", "/Dpy", &
Twiss = FTW_Run, Htable = "Brun_R20", &
Csave = "C_savR20", Psave = "P_savR20", &
Smax = 201.78, &
Jmin = 1, Jmax = 9
```

"X\_Wgts" Vector = 24 ^ 1000000.

Arch\_R20 Archive "P\_savR20", "Brun\_R20"

Rtrv\_R20 Retrieve Force, "P\_savR20", "Brun\_R20"

"VtoD\_C20" VtoD List = L\_XKICK, Param = "KICK", Vname = "C\_savC20"

"VtoD\_R20" VtoD List = L\_XKICK, Param = "KICK", Vname = "C\_savR20"

FTW.Run Ftwiss, Deltap = "P\_savC20"

\*\*\*\*\*

! Drawing Commands for Synthesis, Amplitude Curves.

```
T_Pvcor PVcorrs Vcorrs = T.Bos_R1, &
Xycode = X, Menu, &
Ypos = Yyused * .50, Yyused * .75
```

! Vcorrs Command for Drawing Results in Coefficient Table.

```
T.Bos_R1 Vcorrs Bpms = L_XBPM, Corrs = L_XKICK, &
```



BOOS\_C22 !! Begin Calibration for All 22 Points.

LINEAR LATTICE PARAMETERS FOR BEAM LINE: "BOOSTER ", RANGE = "#S / #E"

DELTA(P)/P = -0.001462 symm = F

PAGE 1

ELEMENT SEQUENCE			HORIZONTAL										VERTICAL						ELEMENT	LENGTH	STRENGTH
POS.	ELEMENT	OC.	DIST I	BETAX	ALFAX	MUX	X(CO)	PX(CO)	DX	DPX	I	BETAY	ALFAY	MUY	Y(CO)	PY(CO)	DY	DPY			
NO.	NAME	NO.	[M]	[M]	[1]	[2PI]	[MM]	[.001]	[M]	[1]	I	[M]	[1]	[2PI]	[MM]	[.001]	[M]	[1]			
BEGIN	BOOSTER	1	0.000	5.412	0.980	0.000	-0.956	0.138	0.644	-0.093		9.755	-1.556	0.000	0.000	0.000	0.000	0.000			
END	BOOSTER	1	201.780	5.412	0.980	4.810	-0.956	0.138	0.644	-0.093		9.754	-1.556	4.806	0.000	0.000	0.000	0.000			
TOTAL LENGTH =			201.780000		QX		=		4.809952		QY		=		4.806365						
DELTA(S) =			-12.434270 mm		QX'		=		-7.268288		QY'		=		-3.010743						
ALFA =			0.419520E-01		BETAX(MAX)		=		13.566733		BETAY(MAX)		=		13.671140						
GAMMA(TR) =			4.882290		DX(MAX)		=		2.966057		DY(MAX)		=		0.000000						
					XCO(MAX)		=		4.327952		YCO(MAX)		=		0.000000						
					XCO(R.M.S.)		=		2.546150		YCO(R.M.S.)		=		0.000000						

Harmonics Program. Starting Value for WDSq = 507.908										Mean Dev. = 0.0048	
X	Harm	A	B	Lsq	Time	Change		Passes			
Begin		-0.001462		507.91	17:10:18						
End Deltap	0	-0.001621	0.000000	505.56	17:10:22	2.349		7			
End Harmonic	5	-0.003184	0.001788	281.64	17:10:31	223.924		14			
End Harmonic	4	-0.001594	0.000509	201.75	17:10:38	79.890		13			
End Harmonic	6	0.000384	-0.000620	190.11	17:10:45	11.637		13			
End Harmonic	3	0.001346	-0.000419	139.47	17:10:53	50.640		13			
End Harmonic	7	-0.000761	0.000050	123.99	17:11:01	15.482		13			
End Harmonic	2	-0.000791	0.000383	104.20	17:11:08	19.788		13			
End Harmonic	8	-0.000394	-0.000831	80.59	17:11:15	23.610		13			
End Harmonic	1	0.001005	0.000061	53.89	17:11:23	26.695		13			
End Harmonic	9	-0.000597	0.000326	40.56	17:11:31	13.334		13			
Iteration 1,	Penalty	40.56, Pen Change		-467.35, Orig Dev.		-0.000320, Dev Change		-0.000320			

Summary Table for X Calibration Fits.

N	Monitor	S	V Measured	V Fitted	Difference	Penalty	Weight	V Initial	V Change
1	PUEHA2	5.08	0.003956	0.002325	0.001631	2.661	1000000.000	-0.001654	0.003979
2	PUEHA4	13.49	-0.008694	-0.007924	-0.000769	0.592	1000000.000	-0.004003	-0.003921
3	PUEHA6	21.90	-0.004039	-0.005597	0.001558	2.428	1000000.000	-0.004194	-0.001403
4	PUEHA8	30.30	0.001900	0.001944	-0.000045	0.002	1000000.000	-0.001748	0.003692
5	PUEHB2	38.71	0.001212	0.001407	-0.000195	0.038	1000000.000	-0.001654	0.003061
6	PUEHB4	47.12	-0.004886	-0.005102	0.000216	0.046	1000000.000	-0.004003	-0.001099
7	PUEHB6	55.53	-0.007631	-0.007438	-0.000192	0.037	1000000.000	-0.004194	-0.003244
8	PUEHB8	63.93	-0.001533	-0.002244	0.000712	0.506	1000000.000	-0.001748	-0.000497
9	PUEHC2	72.34	0.002786	0.002442	0.000345	0.119	1000000.000	-0.001654	0.004096
10	PUEHC4	80.75	0.000946	0.001633	-0.000686	0.471	1000000.000	-0.004003	0.005636
11	PUEHC6	89.16	-0.007147	-0.009463	0.002316	5.364	1000000.000	-0.004194	-0.005269
12	PUEHC8	97.56	-0.019705	-0.017886	-0.001819	3.309	1000000.000	-0.001748	-0.016138
13	PUEHD2	105.97	-0.004802	-0.007360	0.002558	6.543	1000000.000	-0.001654	-0.005706
14	PUEHD4	114.38	-0.001615	-0.001033	-0.000582	7.010	1000000.000	-0.004003	0.005036
15	PUEHD6	122.79	0.000000	0.000368	0.000000	0.000	1000000.000	-0.004194	0.004562
16	PUEHD8	131.19	-0.000625	0.000100	-0.000725	0.526	1000000.000	-0.001748	0.001848
17	PUEHE2	139.60	-0.005475	-0.005806	0.000331	0.110	1000000.000	-0.001654	-0.004152
18	PUEHE4	148.01	-0.004605	-0.005918	0.001313	1.724	1000000.000	-0.004003	-0.001915
19	PUEHE6	156.42	-0.002491	-0.000869	-0.001622	2.632	1000000.000	-0.004194	0.003325
20	PUEHE8	164.82	-0.000376	-0.002086	0.001710	2.924	1000000.000	-0.001748	-0.000339
21	PUEHF2	173.23	-0.002166	-0.001929	-0.000238	0.056	1000000.000	-0.001654	-0.000275

22 PUEHF4 181.64 -0.004314 -0.004678 0.000364 0.133 1000000.000 -0.004003 -0.000675  
23 PUEHF6 190.05 0.000000 -0.006774 0.000000 0.000 1000000.000 -0.004194 -0.002580  
24 PUEHF8 198.45 0.001058 0.002883 -0.001825 3.330 1000000.000 -0.001748 0.004630  
Iter = 1 Final Lsq = 40.560 Mean Dev. = 0.001358  
Initial Lsq = 507.908 Mean Dev. = 0.004805

Summary Table for X Correctors Calibration.

N	Corrector	S	C Fitted	C Initial	Difference	Factors
1	DHCA2	4.77	-0.000102	0.000000	-0.000102	0.100000
2	DHCA4	13.18	0.000470	0.000000	0.000470	0.100000
3	DHCA6	21.59	0.000083	0.000000	0.000083	0.100000
4	DHCA8	30.00	0.000004	0.000000	0.000004	0.100000
5	DHCB2	38.40	0.000118	0.000000	0.000118	0.100000
6	DHCB4	46.81	0.000071	0.000000	0.000071	0.100000
7	DHCB6	55.22	0.000071	0.000000	0.000071	0.100000
8	DHCB8	63.63	0.000143	0.000000	0.000143	0.100000
9	DHCC2	72.03	0.000293	0.000000	0.000293	0.100000
10	DHCC4	80.44	-0.000376	0.000000	-0.000376	0.100000
11	DHCC6	88.85	-0.000570	0.000000	-0.000570	0.100000
12	DHCC8	97.26	-0.000086	0.000000	-0.000086	0.100000
13	DHCD2	105.66	-0.000644	0.000000	-0.000644	0.100000
14	DHCD4	114.07	-0.000353	0.000000	-0.000353	0.100000
15	DHCD6	122.48	0.000372	0.000000	0.000372	0.100000
16	DHCD8	130.89	-0.000025	0.000000	-0.000025	0.100000
17	DHCE2	139.29	0.000250	0.000000	0.000250	0.100000
18	DHCE4	147.70	0.000081	0.000000	0.000081	0.100000
19	DHCE6	156.11	-0.000369	0.000000	-0.000369	0.100000
20	DHCE8	164.52	0.000328	0.000000	0.000328	0.100000
21	DHCF2	172.92	-0.000024	0.000000	-0.000024	0.100000
22	DHCF4	181.33	-0.000197	0.000000	-0.000197	0.100000
23	DHCF6	189.74	0.000520	0.000000	0.000520	0.100000
24	DHCF8	198.15	-0.000060	0.000000	-0.000060	0.100000

RAW FOURIER STRENGTHS IN X, Y

N	FREQ(N)	ABX(N)	AX(N)	BX(N)	N	ABY(N)	AY(N)	BY(N)
0	0.00000	0.00000	0.00000	0.00000	0	0.00000	0.00000	0.00000
1	0.03114	0.00101	0.00101	0.00006	0	0.00000	0.00000	0.00000
2	0.06228	0.00088	-0.00079	0.00038	0	0.00000	0.00000	0.00000
3	0.09342	0.00141	0.00135	-0.00042	0	0.00000	0.00000	0.00000
4	0.12456	0.00167	-0.00159	0.00051	0	0.00000	0.00000	0.00000
5	0.15569	0.00365	-0.00318	0.00179	0	0.00000	0.00000	0.00000
6	0.18683	0.00073	0.00038	-0.00062	0	0.00000	0.00000	0.00000
7	0.21797	0.00076	-0.00076	0.00005	0	0.00000	0.00000	0.00000
8	0.24911	0.00092	-0.00039	-0.00083	0	0.00000	0.00000	0.00000
9	0.28025	0.00068	-0.00060	0.00033	0	0.00000	0.00000	0.00000

NORMALIZED FOURIER STRENGTHS IN X, Y

N	ABX(N)	AX(N)	BX(N)	N	ABY(N)	AY(N)	BY(N)
0	0.00000	0.00000	0.00000	0	0.00000	0.00000	0.00000
1	0.27574	0.27524	0.01666	0	0.00000	0.00000	0.00000
2	0.24077	-0.21672	0.10489	0	0.00000	0.00000	0.00000
3	0.38599	0.36857	-0.11467	0	0.00000	0.00000	0.00000
4	0.45804	-0.43636	0.13927	0	0.00000	0.00000	0.00000
5	1.00000	-0.87189	0.48970	0	0.00000	0.00000	0.00000
6	0.19978	0.10515	-0.16987	0	0.00000	0.00000	0.00000
7	0.20881	-0.20836	0.01371	0	0.00000	0.00000	0.00000
8	0.25179	-0.10802	-0.22745	0	0.00000	0.00000	0.00000
9	0.18622	-0.16342	0.08929	0	0.00000	0.00000	0.00000

LINEAR LATTICE PARAMETERS FOR BEAM LINE: "BOOSTER ", RANGE = "#S / #E"

DELTA(P)/P = -0.001621      symm = F

PAGE 1

ELEMENT SEQUENCE			H O R I Z O N T A L										V E R T I C A L							ELEMENT	LENGTH	STRENGTH
POS.	ELEMENT	OCC.	DIST I	BETAX	ALFAX	MUX	X(CO)	PX(CO)	DX	DPX	I	BETAY	ALFAY	MUY	Y(CO)	PY(CO)	DY	DPY				
NO.	NAME	NO.	[M]	[M]	[1]	[2PI]	[MM]	[.001]	[M]	[1]	I	[M]	[1]	[2PI]	[MM]	[.001]	[M]	[1]				
BEGIN	BOOSTER	1	0.000	5.447	0.983	0.000	2.586	-0.387	0.507	-0.062		9.695	-1.547	0.000	0.000	0.000	0.000	0.000				
END	BOOSTER	1	201.780	5.447	0.983	4.811	2.586	-0.387	0.507	-0.062		9.694	-1.547	4.807	0.000	0.000	0.000	0.000				
TOTAL LENGTH =			201.780000		QX		=		4.811105		QY		=		4.806699							
DELTA(S)			=		-13.972943 mm		QX'		=		-7.267469		QY'		=		-3.011682					
ALFA			=		0.419143E-01		BETAX(MAX)		=		13.611467		BETAY(MAX)		=		13.675287					
GAMMA(TR)			=		4.884488		DX(MAX)		=		3.097710		DY(MAX)		=		0.000000					
					XCO(MAX)		=		18.179518		YCO(MAX)		=		0.000000							
					XCO(R.M.S.)		=		4.722295		YCO(R.M.S.)		=		0.000000							

LINEAR LATTICE PARAMETERS FOR BEAM LINE: "BOOSTER ", RANGE = "#S / #E"

DELTA(P)/P = 0.000000 symm = F

PAGE 1

ELEMENT SEQUENCE			H O R I Z O N T A L										V E R T I C A L						ELEMENT	LENGTH	STRENGTH
POS.	ELEMENT	OCC.	DIST I	BETAX	ALFAX	MUX	X(CO)	PX(CO)	DX	DPX	I	BETAY	ALFAY	MUY	Y(CO)	PY(CO)	DY	DPY			
NO.	NAME	NO.	[M]	[M]	[1]	[2PI]	[MM]	[.001]	[M]	[1]	I	[M]	[1]	[2PI]	[MM]	[.001]	[M]	[1]			
BEGIN	BOOSTER	1	0.000	5.444	0.982	0.000	3.437	-0.493	0.542-0.068			9.701	-1.548	0.000	0.000	0.000	0.000	0.000			
END	BOOSTER	1	201.780	5.444	0.982	4.799	3.437	-0.493	0.542-0.068			9.701	-1.548	4.802	0.000	0.000	0.000	0.000			
TOTAL LENGTH =			201.780000		QX		=		4.799341		QY		=		4.801815						
DELTA(S)			=		-0.189805 mm		QX'		=		-7.274764		QY'		=		-2.988117				
ALFA			=		0.423552E-01		BETAX(MAX)		=		13.613012		BETAY(MAX)		=		13.671588				
GAMMA(TR)			=		4.858995		DX(MAX)		=		3.067570		DY(MAX)		=		0.000000				
							XCO(MAX)		=		15.955692		YCO(MAX)		=		0.000000				
							XCO(R.M.S.)		=		3.710790		YCO(R.M.S.)		=		0.000000				

Harmonics Program. Starting Value for Wdsq = 513.787				Mean Dev. = 0.0046		
X Harm	A	B	Lsq	Time	Change	Passes
Begin		0.000000	513.79	17:11:33		
End Deltap	0	0.000034	0.000000	513.17	17:11:37	0.616 6
End Harmonic	5	0.003655	-0.001725	258.90	17:11:44	254.268 14
End Harmonic	4	0.001538	-0.000573	170.77	17:11:52	88.129 13
End Harmonic	6	-0.000404	0.000600	158.82	17:12:00	11.949 13
End Harmonic	3	-0.001296	0.000403	104.79	17:12:07	54.033 13
End Harmonic	7	0.000721	-0.000048	89.48	17:12:15	15.310 13
End Harmonic	2	0.000775	-0.000369	68.06	17:12:22	21.423 13
End Harmonic	8	0.000403	0.000843	43.77	17:12:30	24.284 13
End Harmonic	1	-0.000926	-0.000048	19.27	17:12:37	24.499 13
End Harmonic	9	0.000626	-0.000343	3.62	17:12:45	15.655 13
Iteration 1,	Penalty	3.62,	Pen Change	-510.17,	Orig Dev.	-0.000009, Dev Change -0.000009

Summary Table for X Adjustment Fits.

N	Monitor	S	Good Orbit	V Fitted	Difference	Penalty	Weight	V Initial	V Change	Adj Diff.	Adj Pen.
1	PUEHA2	5.08	0.000000	-0.000316	0.000316	0.100	1000000.000	0.004021	-0.004338	0.000379	0.144
2	PUEHA4	13.49	0.000000	0.000242	-0.000242	0.059	1000000.000	-0.003214	0.003455	-0.000179	0.032
3	PUEHA6	21.90	0.000000	0.000675	-0.000675	0.456	1000000.000	-0.000679	0.001355	-0.000613	0.376
4	PUEHA8	30.30	0.000000	0.000369	-0.000369	0.136	1000000.000	0.003771	-0.003402	-0.000306	0.094
5	PUEHB2	38.71	0.000000	-0.000264	0.000264	0.070	1000000.000	0.002936	-0.003200	0.000327	0.107
6	PUEHB4	47.12	0.000000	-0.000262	0.000262	0.069	1000000.000	-0.000716	0.000454	0.000325	0.105
7	PUEHB6	55.53	0.000000	0.000283	-0.000283	0.080	1000000.000	-0.002531	0.002814	-0.000220	0.049
8	PUEHB8	63.93	0.000000	0.000443	-0.000443	0.197	1000000.000	-0.000102	0.000546	-0.000381	0.145
9	PUEHC2	72.34	0.000000	-0.000051	0.000051	0.003	1000000.000	0.004135	-0.004186	0.000113	0.013
10	PUEHC4	80.75	0.000000	-0.000402	0.000402	0.162	1000000.000	0.005794	-0.006196	0.000465	0.216
11	PUEHC6	89.16	0.000000	-0.000146	0.000146	0.021	1000000.000	-0.004804	0.004658	0.000209	0.044
12	PUEHC8	97.56	0.000000	0.000266	-0.000266	0.071	1000000.000	-0.015662	0.015928	-0.000203	0.041
13	PUEHD2	105.97	0.000000	0.000198	-0.000198	0.039	1000000.000	-0.005457	0.005655	-0.000135	0.018
14	PUEHD4	114.38	0.000000	-0.000311	0.000311	0.097	1000000.000	0.005214	-0.005525	0.000374	0.140
15	PUEHD6	122.79	0.000000	-0.000500	0.000500	0.250	1000000.000	0.004833	-0.005333	0.000562	0.316
16	PUEHD8	131.19	0.000000	0.000040	-0.000040	0.002	1000000.000	0.002192	-0.002152	0.000023	0.001
17	PUEHE2	139.80	0.000000	0.000622	-0.000622	0.386	1000000.000	-0.003689	0.004310	-0.000559	0.312
18	PUEHE4	148.01	0.000000	0.000332	-0.000332	0.110	1000000.000	-0.001500	0.001833	-0.000269	0.073
19	PUEHE6	156.42	0.000000	-0.000405	0.000405	0.164	1000000.000	0.003478	-0.003883	0.000467	0.218
20	PUEHE8	164.82	0.000000	-0.000530	0.000530	0.280	1000000.000	-0.000297	-0.000233	0.000592	0.351
21	PUEHF2	173.23	0.000000	0.000201	-0.000201	0.041	1000000.000	0.000104	0.000097	-0.000139	0.019
22	PUEHF4	181.64	0.000000	0.000759	-0.000759	0.575	1000000.000	0.000030	0.000729	-0.000696	0.484
23	PUEHF6	190.05	0.000000	0.000461	-0.000461	0.213	1000000.000	-0.002146	0.002608	-0.000399	0.159

24 PUEHPs 198.45 0.000000 -0.000201 0.000201 0.040 1000000.000 0.004498 -0.004699 0.000264 0.070  
Iter = 1 Final Lsq = 3.620 Mean Dev. = 0.000388 Ave G = 0.000000 Ave B = 0.000063  
Adjusted: Final Lsq = 3.525 Mean Dev. = 0.000383  
Initial Lsq = 513.787 Mean Dev. = 0.004627

Summary Table for X Correctors Run.

N	Corrector	S	C Fitted	C Initial	Change	Factors
1	DHCA2	4.77	0.000010	-0.000102	0.000112	0.100000
2	DHCA4	13.18	0.000012	0.000470	-0.000458	0.100000
3	DHCA6	21.59	0.000008	0.000083	-0.000075	0.100000
4	DHCA8	30.00	0.000007	0.000003	0.000003	0.100000
5	DHCB2	38.40	0.000014	0.000118	-0.000104	0.100000
6	DHCB4	46.81	0.000010	0.000070	-0.000060	0.100000
7	DHCB6	55.22	-0.000007	0.000071	-0.000077	0.100000
8	DHCB8	63.63	-0.000012	0.000143	-0.000155	0.100000
9	DHCC2	72.03	-0.000003	0.000293	-0.000296	0.100000
10	DHCC4	80.44	-0.000001	-0.000377	0.000376	0.100000
11	DHCC6	88.85	-0.000009	-0.000571	0.000561	0.100000
12	DHCC8	97.26	-0.000018	-0.000086	0.000068	0.100000
13	DHCD2	105.66	-0.000023	-0.000644	0.000621	0.100000
14	DHCD4	114.07	-0.000016	-0.000353	0.000337	0.100000
15	DHCD6	122.48	-0.000001	0.000372	-0.000372	0.100000
16	DHCD8	130.89	0.000000	-0.000025	0.000025	0.100000
17	DHCE2	139.29	-0.000009	0.000250	-0.000259	0.100000
18	DHCE4	147.70	-0.000007	0.000081	-0.000088	0.100000
19	DHCE6	156.11	0.000001	-0.000369	0.000370	0.100000
20	DHCE8	164.52	0.000000	0.000328	-0.000328	0.100000
21	DHCF2	172.92	-0.000001	-0.000024	0.000023	0.100000
22	DHCF4	181.33	0.000009	-0.000197	0.000206	0.100000
23	DHCF6	189.74	0.000017	0.000520	-0.000504	0.100000
24	DHCF8	198.15	0.000013	-0.000060	0.000072	0.100000

RAW FOURIER STRENGTHS IN X, Y

N	FREQ(N)	ABX(N)	AX(N)	BX(N)	N	ABY(N)	AY(N)	BY(N)
0	0.00000	0.00000	0.00000	0.00000	0	0.00000	0.00000	0.00000
1	0.03114	0.00093	-0.00093	-0.00005	0	0.00000	0.00000	0.00000
2	0.06228	0.00086	0.00077	-0.00037	0	0.00000	0.00000	0.00000
3	0.09342	0.00136	-0.00130	0.00040	0	0.00000	0.00000	0.00000
4	0.12456	0.00164	0.00154	-0.00057	0	0.00000	0.00000	0.00000
5	0.15569	0.00404	0.00366	-0.00173	0	0.00000	0.00000	0.00000
6	0.18683	0.00072	-0.00040	0.00060	0	0.00000	0.00000	0.00000
7	0.21797	0.00072	0.00072	-0.00005	0	0.00000	0.00000	0.00000
8	0.24911	0.00093	0.00040	0.00084	0	0.00000	0.00000	0.00000
9	0.28025	0.00071	0.00063	-0.00034	0	0.00000	0.00000	0.00000

NORMALIZED FOURIER STRENGTHS IN X, Y

N	ABX(N)	AX(N)	BX(N)	N	ABY(N)	AY(N)	BY(N)
0	0.00000	0.00000	0.00000	0	0.00000	0.00000	0.00000
1	0.22943	-0.22913	-0.01179	0	0.00000	0.00000	0.00000
2	0.21232	0.19168	-0.09131	0	0.00000	0.00000	0.00000
3	0.33576	-0.32065	0.09960	0	0.00000	0.00000	0.00000
4	0.40606	0.38050	-0.14179	0	0.00000	0.00000	0.00000
5	1.00000	0.90431	-0.42688	0	0.00000	0.00000	0.00000
6	0.17892	-0.09985	0.14846	0	0.00000	0.00000	0.00000
7	0.17869	0.17829	-0.01182	0	0.00000	0.00000	0.00000
8	0.23124	0.09967	0.20865	0	0.00000	0.00000	0.00000
9	0.17666	0.15499	-0.08477	0	0.00000	0.00000	0.00000

LINEAR LATTICE PARAMETERS FOR BEAM LINE: "BOOSTER ", RANGE = "#S / #E"

DELTA(P)/P = 0.000034 symm = F

PAGE 1

ELEMENT SEQUENCE			H O R I Z O N T A L										V E R T I C A L								
POS.	ELEMENT	OCC.	DIST I	BETAX	ALFAX	MUX	X(CO)	PX(CO)	DX	DPX	BETAY	ALFAY	MUY	Y(CO)	PY(CO)	DY	DPY	ELEMENT	LENGTH	STRENGTH	
NO.	NAME	NO.	[M]	[M]	[1]	[2PI]	[MM]	[.001]	[M]	[1]	[M]	[1]	[2PI]	[MM]	[.001]	[M]	[1]				
BEGIN	BOOSTER	1	0.000	5.409	0.978	0.000	-0.198	0.013	0.676	-0.097	9.762	-1.557	0.000	0.000	0.000	0.000	0.000				
END	BOOSTER	1	201.780	5.409	0.978	4.799	-0.198	0.013	0.676	-0.097	9.762	-1.557	4.802	0.000	0.000	0.000	0.000				
TOTAL LENGTH =			201.780000		QX		=	4.799098		QY		=	4.801866								
DELTA(S)			=	0.298352 mm		QX'		=	-7.274935		QY'		=	-2.989095							
ALFA			=	0.423541E-01		BETAX(MAX)		=	13.566623		BETAY(MAX)		=	13.666313							
GAMMA(TR)			=	4.859063		DX(MAX)		=	2.965588		DY(MAX)		=	0.000000							
						XCO(MAX)		=	0.779494		YCO(MAX)		=	0.000000							
						XCO(R.M.S.)		=	0.312237		YCO(R.M.S.)		=	0.000000							