Brookhaven National Laboratory

BNL-104892-2014-TECH

AGS/AD/Tech Note No. 486;BNL-104892-2014-IR

BNL MAD Programming Notes: Orbit Correction Systems - The Micado Command

J. Niederer

June 1999

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-AC02-98CH10886 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. 486

BNL MAD Programming Notes
Orbit Correction Systems – The Micado Command

J. Niederer, BNL

June 24, 1999

BNL MAD Programming Notes

Orbit Correction Systems - The Micado Command

J. Niederer

Brookhaven National Laboratory
June 24, 1999

1. Summary

Some features of orbit correction systems in the BNL MAD Program are discussed. Both the conventional "Micado" Corrector / Monitor influence technique and a BNL developed harmonic technique are considered. Both techniques are designed to work in a dynamic environment able to accept orbit signals from either a model or an operating accelerator, and to deliver appropriate responses. Command language features may be used to select which Correctors and Monitors are to be included in the correction schemes.

2. Introduction

Studies of spin tracking are often concerned with the effects of various orbit irregularities on the preservation of spin. Ideally, orbit tolerance criteria can be presented, and related to correction schemes which maintain such tolerances. Conventionally, correction schemes minimize some function of the orbit positions, such as least squares of offsets from some ideal orbit. The newer CERN versions of MAD also include dispersion minimization options, again relative to an ideal orbit.

In practice, there may be additional complications, such as artifacts from patterns of present or missing Correctors and Monitors, and operating orbits which are not necessarily the straight up the middle reference one of the computer models. Monitors placed according to the usual textbook criteria assume particular tunes and phase advances, which may differ in actual operation. Fitting schemes minimize values at the monitors and may miss deviant behavior away from them. Results will depend on which elements are working, and probably upon monitor spacing as well. Finer structures in the phase advances are usually missed or ignored, but may be relevant for high current machines where any beam losses are significant. Thus it may be useful to learn whether the usual treatments for controlling orbit deviations and their influence on parasitic effects may be extended or improved. To this end, various CERN and BNL MAD features have been combined within the somewhat more flexible BNL MAD environment for exploring such issues.

3. Micado Based Correction System Commands

3.1. Micado

The BNL implementation of the **Micado** command follows earlier BNL MAD (Vers 7) descriptions and coding, together with some BNL additions, along with some newer CERN (Vers 8) ones as well. **Micado** computes a closed orbit for the reference case. If this orbit exists, the command proceeds to compute Corrector settings that minimize the deviations of the actual closed orbit from the reference orbit at the Monitor positions. Monitor readings are taken from tracking within the orbit model. If no other guidance is given, the command assumes that all Correctors and Monitors in both planes are available for use. Alternatively lists denoting specific Monitors and Correctors may be supplied, and fitting limited to either the X or Y plane.

While a correction routine is active, it works from a lattice matrix table that has entries for each element at each occurrence. The correction commands, **Micado, Vcorrs, ..** , use linked lists among the active Monitors and Correctors for each plane to simplify various internal scans and calculations. These links persevere until the next correction scheme, or one of the **Getb, Getc, Putb,** & **Putc** data transfer commands noted below, is applied.

The BNL version has a Micado based option to constrain tunes in both planes. When active, it calibrates tune changes with respect to given tune parameters, and uses these calibrations along with tune errors at each step of the iterations to adjust tunes to given constraint values. As the corrector adjustments shift the tunes around during the fitting, it usually takes a few steps before the tunes begin to approach the constraint values.

As presently implemented, the Micado technique fits only to an unperturbed reference orbit, one assumed to have zero displacements at all of the Monitors. An influence matrix based upon phase advances among all active Correctors and Monitors for the reference orbit is used for the fitting. Thus the present Micado cannot be applied directly to produce an orbit with shifts at insertions or similar deliberate steering offsets. This restriction can presumably be overcome by program changes that construct the influence matrix from a prescribed orbit, apply orbit perturbations, and minimize the differences between the prescribed and the perturbed orbit at the Monitors.

Micado attributes are:

Error — The desired accuracy (tolerance) of the correction. Decimal. Normally the spread between the maximum and minimum residuals among the orbit at the Monitors is tested. Alternatively the tolerance test is applied instead to the rms residuals at Monitors if the *Rms* option is given. Default value is 1.d-5. As the approach to the minimum may differ somewhat among program versions, for larger tolerance values the delivered results may also differ.

Ncorr — Integer maximum number of active Correctors to be used in the fitting process. The minimization routine is a Householder transform that begins with the most effective Correctors, as found by comparing the elements of an influence matrix along with a function combining this matrix and residuals at the Monitors. Successive transformations work through the each of the active Correctors in the order of their relative "influence" rating. The *Ncorr* attribute limits this iteration to those Correctors most likely to be helpful for the fit. Reducing *Ncorr* often leads to better fits, and there is often a broadly optimal range over which it can be effective. Default is to compute using all active Correctors.

Iterate — Integer number of iterations to be tried before finishing. This limit may cut off the fitting before the specified tolerance is reached.

Plane — X or Y, corresponding to horizontal plane only or vertical plane only. Default is to do both planes.

M1list — Name option to enable printing of Monitors before fitting. This and the following three attributes affect the printing of tables. Choices are "NONE", "USED", "ALL", or nothing, corresponding to listing: (1) no table of individual Monitors (Correctors), (2) only those active, or (3) all, whether active or not. A short summary table is always printed as well.

C1list — Name option for printing of Correctors before fitting.

M2list — Name option for printing of Monitors after fitting.

C2list — Name option for printing of Correctors after fitting.

Twiss1 — Logical Flag: If True, a Twiss tracking pass and listing is carried out before the fitting. The listing is subject to the conditions set by previous **Print** commands.

Twiss2 — Logical Flag: If True, a Twiss tracking pass and listing is carried out after the fitting.

Additional features and attributes for the Micado Section of the BNL MAD Program include a *Rms* residual reading criterion for convergence, and use of lists to specify particular Monitors and Correctors to be applied to the corrector fitting process. The convenience commands **Usekick** and **Usemon** together with **Makeblist** and **Makeclist** noted later can be helpful for generating such lists of elements. The tune constraint feature involves an enabling flag, names of the two program parameters that regulate the tunes, and the values of the tunes.

Rms — Logical Flag: If True, causes the Micado iterations to be halted when the root mean square of the residual monitor readings becomes less than the value entered in the *Error* attribute field.

The default is to finish when the difference between the largest and least readings (signed) is less than the value of *Error*, as described in the *MAD User's Manual.*

Corlist — The name of a **List, Menu,** or **Xmenu** statement that holds a list of the names of the Correctors (Kickers) to be used by the **Micado** command. Optional.

The form of a **List** or **Menu** statement is:

"Corr_1" LIST = (c1, c2, c3, c4, c5, c6)

All occurrences of each Corrector named will be used, and any Correctors not in the list will be ignored. There is one such list, which can include both X and Y Correctors. Lists cannot be nested.

If specific occurrences of particular elements are wanted, the BNL **Xmenu** form of list should be used, along with the newer square brackets occurrence notation:

"Corr_X" Xmenu c1[1], c1[3], c2, c3, c4, c5, c6

Monlist — The name of a **List, Menu,** or **Xmenu** statement that holds a list of the names of the Monitors to be used by the **Micado** command. Optional. Default is to use all Monitors.

Example:

"Mon_1" LIST = (m1, m2, m3, m4, m5, m6, m7, m8)

All occurrences of each Monitor named will be used, and any Monitors not in the list will be ignored. Again, both X and Y Monitors can appear in this single list.

Clip — Decimal value for the maximum permitted Corrector angular kick strength, absolute. Optional. If given, settings are clipped (clamped) so as not to exceed this maximum. This feature will help avoid apparent fits that produce Corrector settings that exceed the capacities of the actual devices.

Csaves(2) — The name(s) of a BNL MAD **Vector** to receive the Corrector values computed by the Micado run. Separate names for X and for Y values are expected. Optional. The contents of these vector(s) may be further processed, or used as inputs for other runs, or be passed to accelerator control systems. These vectors may be written to the local file system via the **Archive** command, and read back as inputs for further runs by means of the **Retrieve** command. These vectors are suitable input for the **Vcorrs** harmonic based fitting command, which also outputs similar fitted setting data.

Tune — Logical Flag: If True, the tune constraint feature is enabled when all of the needed *Q* and *Tparam* attributes are provided.

Q(2) — Decimal values expected for both X and Y tunes. (Constraint values)

Tparam(2) — Names of variables in the program that regulate the two tunes. Normally the name of a MAD **Parameter** if a number of quads take the same value for their strength attribute, but *Element [Attribute (Index)]* format names can be used.

Examples:

```
Micado  Error = 1.e-5,  RMS,  Iterate = 5, C1list = All,          &
        C2list = All, M2list = All,                  &
        Corlist = "Corr_1", Monlist = "Mon_1"
```

Constrained Tunes:

```
Micado  Error = 1.e-6,  RMS,  Iterate = 7, C1list = All,          &
        C2list = All, M2list = All,                  &
        Corlist = "Corr_1", Monlist = "Mon_1",       &
        Tune,  Q = 28.19044,  29.17305,  Tparam = KQH, KQV
```

3.2. Correct Command

The BNL **Correct** command corresponds closely to the CERN CORRECT command. It is similar in function to **Micado,** with most of the same attributes, plus provision for dispersion. **Correct** explicitly uses the *Rms* of the orbit residuals at Monitors criteria for fitting. It uses only those Correctors and Monitors marked active, as determined by the initial building of the lattice table, and any subsequent use of **Usekick** and **Usemon** commands. *Twiss1* and *Twiss2* options have been dropped, as they are already provided by the ordinary **Twiss** command.

Additional **Correct** attributes are:

Dxweight — A decimal number for weighting the X dispersion part of the fits. As dispersion is typically of order of a meter, whereas orbit offsets are of order .001 meter, weights should be set accordingly. Giving no weights for dispersion produces the same closed orbit fit as the above **Micado** command.

Dyweight — Same for weighting Y dispersion.

The BNL version also has the optional *Corlist* and *Monlist* attributes for selecting elements from supplied lists, which over ride any previous selections. The BNL *Clip* and *Csaves* features have also been provided.

4. The Vcorrs Harmonic Fitting Command.

Vcorrs was developed as an harmonic based alternative to Micado fitting for smaller machines such as the AGS Booster, where the lattice structures are somewhat finer grained than the Monitor / Corrector sets. It makes use of both the lattice optical functions and harmonic patterns on Correctors as basis functions for matching models to real orbits, and for adjusting Correctors to obtain specified orbits, using ordinary least squares criteria. It follows the fine structure of measured orbits somewhat better than the Micado method, and can be less sensitive to missing elements. It obtains quite similar overall least squares results to the Micado method, both at the Monitors, and for the orbit as a whole as computed from "Twiss" tracking runs, for AGS and Booster trials. For inputs, **Vcorrs** expects lists of names of Correctors and Monitors, and initial conditions and orbit specification data, corresponding to the elements in these lists. The full details of its input information and output data are given in a separate manual. Some effort has been made to provide similar input and output formats for the two fitting commands. The main differences now are that **Vcorrs** expects included elements to be named explicitly in separate X and Y lists, and is able to fit to any orbit prescribed at the Monitors, not just the reference one.

In RHIC spin tracking studies, various perturbations in quad positions are expected to excite orbit harmonics near the tune, and presumably harmonics that are multiples of the six fold symmetry. Thus one might expect that harmonic based correction schemes on this larger machine might show some advantages over the conventional Micado method. Instead it turns out that the effects of a RHIC quad error distribution over the actual lattice for which Correctors lie adjacent to the displaced quads are easily corrected by the Micado method, indeed almost too well, if Corrector strength is not constrained appropriately. In early simulations, the apparent Micado orbit rms improvement was more than an order of magnitude better relative to the harmonic one, inspiring further study about this particular application of each of the two techniques. The harmonic scheme reduces the rms error by the more usual factor of 10 - 15, but does somewhat more poorly at the places where the larger quad (paper) displacements are located. However, the peak Corrector strength is considerably less for the harmonic case, and when this strength is constrained, the two methods lead to somewhat more comparable rms solutions. As the numbers of available correctors are reduced, the rms achieved by the harmonic method tends to fall off more slowly than the Micado one, so there may still be trade offs to be explored.

5. Selecting Monitors and Correctors

Early versions of the MAD Program were limited to using all Correctors and Monitors in the working beam line (lattice). In practice, not all of these elements are assigned to orbit management, some may be more effective than others, and some may not work at all. Hence we introduced the use of lists of named elements at BNL for NSLS applications. At CERN, where composing lists of hundreds of individually named elements for the big machines was more of a typing problem, selective features were based upon multiple occurrences of a common Corrector and Monitor. In this section we discuss BNL MAD element selection features that now include both BNL and CERN styles. The commands involved are presently static ones that are applied before the orbit correction commands are applied. Perhaps it might be useful to arrange for dynamic status changes as well when studying stability with the various BNL MAD orbit controller models.

Correctors include element types **KICKER, HKICKER** and **VKICKER.** Monitors include element types **MONITOR, HMONITOR,** and **VMONITOR.**

Monitor and Corrector data are usually handled in an internal table affiliated with expanded lattice maps, where repeated occurrences of a lattice element need language features to deal with the appropriate indexing. Such tables are built from the current beam line by one of the major commands such as **Twiss.** If there is no such table, then the **Usekick** and **Usemonitor** commands will build one.

5.1. Usekick and Usemonitor Commands

The CERN developed **Usekick** and **Usemonitor** commands are also included in the BNL version for activating or deactivating the corresponding elements involved in the fitting ensemble. The description here mostly repeats the CERN one.

Both commands have the same attributes:

X	Logical Flag: If True, X Plane Monitors are selected.
Y	Logical Flag: If True, Y Plane Monitors are selected. Both *X* and *Y* may be selected.
Status	Logical Flag: If True, (ON), the selected elements are set to "active". If False (OFF), the selected elements are set to "inactive". Inactive elements are ignored by the **Correct** and **Micado** orbit correction commands. *Status* is initially set to "ON" for all elements when the lattice table is built.
Mode	This name can take the value of "ALL", "USED" and "UNUSED" for both Monitors and Correctors, "USED" sets the status of all Correctors / Monitors which currently have status "ON". "UNUSED" sets the status of all Correctors / Monitors which currently have status "OFF". Default is to do "ALL". The Cern version accepts only "ALL" for Monitors.
Range(10)	The value of *Status* is set for all elements satisfying the range criteria. Up to 10 range specifiers can be given on a single command, but any number of such commands can be applied to a beam line. The *Range* attribute is only considered if *Mode* is not

given.

Example:

... Range = Quad31 [1 / 7], Quad73 / Quad 79, ...

An additional *List* attribute is provided in the BNL version.

List — The name of a **List, Menu,** or **Xmenu** list of names of Corrector or Monitor elements to be affected. For each name in the list, the *Mode* picks are satisfied. Unmatched names are not affected. If no *Mode* is given, ALL elements are considered.

The **Usekick** and **Usemon** commands can be highly selective. Some of the pick options are based on only the current status of an element, and may seem to give erratic results when intended to reset an entire configuration. It may be best to begin by resetting all elements using the ALL option, and then applying more detailed selections in a following command.

Example:

Usekick, X, Y, Status = .true., Mode = all

Usekick, X, Range = KICKH[13 / 18], KICKH[101 / 105], Status = .false.

The most recent BNL version of the **Micado** command recognizes these active / inactive status settings.

5.2. Makeblist and Makeclist

These BNL developed utility commands create a new list of the names and occurrences of all Monitors / Correctors for which their status is marked "ON", for a given plane (X or Y), or both planes. They provide a way to link the CERN range style of selecting elements with the BNL name list oriented style. The produced list is ordered by appearance in the lattice, the form expected by most commands that use these lists.

Newlist — The name of the intended new output list. (**Xmenu** form) If not given, a serialized default name based on the command will be generated for this list: *BpmList.001, CorrList.003, ...*

Plane — X, Y, or XY.

Anew — Internal name group giving actual name of output list, given, or default.

Example:

Usekick, X, Y, Status = .true., Mode = all

Usekick, X, Range = KICKH[13 / 18], Status = .false.

Usekick, Y, Range = KICKV[14 / 19], Status = .false.

Makeclist, "Cxylist", Plane = XY

The resulting pruned Corrector list is suitable for an ordinary **Micado** run:

Micado, ... Corlist = "Cxylist", ...

5.3. Sortblist and Sortclist Commands.

These BNL developed utility commands can be used for sorting lists of Monitor and Corrector names into the order in which the elements appear in the currently expanded beam line (lattice). Several kinds of BNL command, such as data gathering and storing, and the harmonic correction, expect all involved elements to be named, and elements in such lists to be in this proper order. These classes of commands work on the expanded lattice maps, allowing common elements to be addressed by their occurrence index. **Sortblist** deals with Monitors (Bpms) and **Sortclist** deals with Correctors.

Oldlist	The name of an unsorted list of names of Monitors or Correctors (**List, Menu,** or **Xmenu** form). If not given, a *Newlist* holding the names and occurrences of ALL Monitors / Correctors will be written.
Newlist	The name of an output list of names of Monitors or Correctors (**List, Menu,** or **Xmenu** form). If not given, a serialized default name will be generated for this list, typically: *BpmSort.001, CorrSort.004, ...*
Aold	Internal name group giving actual name of input list, if any.
Anew	Internal name group giving actual name of output list, given, or default.

6. Data Gathering and Storing Commands.

Four classes of similar command offer a way to insert or extract data into / from the various fitting procedures. All involve a sorted list of element names, an array of decimal data items, and the name of the common attribute involved in the operation. All can work with lattice descriptions in which a given element may have multiple occurrences. These are also useful commands for moving actual machine data into and out of model based computations.

Getbdata and **Getcdata** gather data from Monitor or Corrector sections of the lattice tables. **Putbdata** and **Putcdata** insert data into Monitor or Corrector sections of the lattice tables. All of these commands relink the lattice matrix table chains of Monitors or Correctors for the given plane(s). All four commands make use of related routines which also move data around for the harmonic fitting command.

Common attributes are:

List	The name of a list of names of the family of elements to be involved. (**List, Menu,** or **Xmenu**) form. Occurrence indexing has the form: Corr2(:3), Corr(:4), etc, or alternatively Corr[3], Corr[4], ..., and is restricted to the **Xmenu** form of list.
Data	The name of a BNL MAD **Vector** that holds or receives data from the operation. Needed for input on **Put** operations. For **Get** operations, if a name is given, its size is checked against the list length. If a **Get** name is not given, a new **Vector** is created, with its name serially defaulted *Getbdata.001, Getbdata.002, ...*
Plane	The name of the *Plane(s)* involved, X, Y, or XY. Some defaulting may be done, depending on the kind of element in the list.
Param	The name of the attribute common to all members of the list, such as "/X" for Monitor readings, or "/Xkick" for a Corrector setting.
Doerrors	Logical Flag: If True, apply field errors to Corrector **Putcdata** only.
Value	Decimal number to be set in given attribute for all listed Monitors / Correctors. Applies to **Putbdata** and **Putcdata** commands only. A convenience alternative to using a complete data array to set a common *Param* attribute.
Alist	Internal name group giving the name of the input list.
Adata	Internal name group giving the name of the output data array, given or defaulted.

Example

Retrieve, "Corr_X", "Corr_Y", "X_saves", "Y_saves"

Putcdata, List = "Corr_X", Plane = X, Data = "X_saves", Param = "/Xkick"

Putcdata, List = "Corr_Y", Plane = Y, Data = "Y_saves", Param = "/Ykick"

Here, lists of X and Y sets of Correctors, and results of a corresponding Micado run have been preserved in the local file system previously. Each of these four files is entered into the MAD data base by means of the **Retrieve** command. Using the lists of names of Correctors and corresponding arrays of values, the two **Putcdata** commands load these values into the current working lattice tables.

7. NMicado: Micado As a Neural Algorithm

The Micado technique has also been applied in the form of a controller neuron in the neural feedback model sections of BNL MAD. In one kind of study, time dependent perturbations are applied to some component(s) of an orbit. An ensemble of sensors and filters, organized as network of communicating neurons, directs processed beam position signals back to the controller at each of a series of time steps. Corrector settings are computed and delivered to the related model, the resulting orbits remeasured, and corrections recomputed. Actual machine readings and settings are readily introduced in place of the model generated ones.

There is a working set of demonstration files for both the AGS and for Rhic which track orbits for which time varying perturbations are corrected by a Micado controller, with workstation graphics and user guidance. For each time step, the model provides the perturbation, finds the closed orbit conventionally, reads the orbit, computes the corrections, adjusts the correctors, and repeats. These BNL MAD input files include the lattice description, the network description, the graphics setup, and a small control file. These setups can also be used just to watch the Micado procedure iterate a fit to an initial set of magnet and other error distortions.

Some of the **NMicado** attributes are similar to those of the ordinary **Micado** command, but the set is more complicated in that the neural object (data structure) is expected to hold more information about the correction process, as more than one such feedback controller may be operating at the same time. Various pointers involved in work arrays are also maintained in the object. An additional set of attributes deals with output signal details.

The tune constraint feature of the regular **Micado** command is not included, as this function in neural based configurations is better handled by a separate control loop. Instead tunes are readily regulated by attaching a pair of filter or discriminator neurons to the tunes, and feeding their outputs into a pair of adder neurons. The adder inputs are weighted to deliver appropriate responses to the variables which control the strengths of the relevent tune elements.

Sourcelist — The name of a BNL MAD **Varlist** type of list noting the names of input signals, nominally the outputs of filter neurons in this configuration. For Micado purposes, each signal should originate from a single Monitor. The procedure needs to know the positions of the corresponding Monitors in the lattice, so signals are traced backwards to orignating **Bpm** modules and hence to lattice **Monitor** elements by the program.

Vvvvv Varlist, FilterA["/output"], FilterB["/output"], ...

Readlist — The name of a BNL MAD **Varlist** type of list noting the names of Corrector reading cells in **Corrector** modules. For Micado purposes, each Corrector should receive only a single signal from the controller ensemble. The procedure needs to know the positions of the corresponding kickers in the lattice. The program traces signals forwards to accepting **Corrector** modules and hence to lattice **Kicker** elements.

Kkkkkk Varlist, CorrA["/cread"], CorrB["/cread"], ...

Outputs — The name of a BNL MAD **Vector** array to receive output signals from the correction computation. Each entry occupies 3 full words: (1) decimal signal value, (2) 2integer pointers, and (3) 2 integer status. This name will be used as given, and will be needed by neurons that use the Micado derived outputs.

Plane — The name of the coordinate plane being treated, X or Y. Default is X. Only one plane is treated by each **NMicado** controller loop.

Bpmlist — The name to be given to a BNL MAD internal **Xmenu** list of names of active Bpms (Monitors), derived from the members of the *Sourcelist*. Optional, will be defaulted. Mainly helps with debugging when a number of control loops may be active. A serial suffix will be added to the given name to avoid conflicts.

Corrlist	The name to be given to a BNL MAD internal **Xmenu** list of names of active Correctors (Kickers), derived from the members of the *Readlist.* Optional, will be defaulted. A serial suffix will be added to the given name to avoid conflicts.
Work	The name of an internal scratch array used in carrying out the computations. Optional, but may be helpful in debugging, especially if more than one such feedback loop is active. A serial suffix will be added to the given name to avoid conflicts.
Weights	Name of weights array, a BNL MAD **Vector** form. Optional. Should have same dimension as *Sourcelist.* Default is unit weights.
Bias	Decimal number to be added to each incoming signal. Optional. Default is 0.
Ncorr	Integer; number of most influential correctors to be treated in fit. Default is all active correctors.
Tolerance	Decimal number; value below which fitting calculation finishes. Should not be markedly smaller than the input noise level. Default is 1.d-5.
Rms	Not currently used. Intended as Logical Flag to enable an rms fitting criteria, as alternative to testing on extremes of residuals.
Rule	Integer giving rule for fitting. Currently ignored.
Ioform	Integer giving output signal treatment: (1); (2); (3) Sigmoid; (4) Linear.
Thresh1(4)	Decimal numbers giving limit (threshhold) below which output signals are clamped at threshhold value. Tends to suppress overshoots. If set to too low a fraction of peak signal, may interact with *Gain.* Keyed to signal form.
Thresh2(4)	Decimal numbers giving limit (threshhold) above which output signals are clamped at threshhold value. Keyed to signal form.
Gain	Decimal number multipling output signals. Applied before *Threshhold* and *Deadzone* tests. A critical number that affects the time response of the controller, If too large, it will cause instability. If too small, the system will be unresponsive. The value to be used is related to the lag in filters which precede the controller. For example, if filters take three time steps to respond fully to a changing error signal, the Micado signal gain should be under one third.
Step	
Scale	A decimal multiplier applied to signal before computing sigmoid function, for sigmoid output form only.
Deadzone(2)	Decimal numbers indicating range above and below which smaller signals are ignored. Essentially filters out low level noise.
Ablist	Internal name group marking internal list of lattice **Monitors** used.
Aclist	Internal name group marking internal list of lattice **Kickers** used.
Aodata	Internal name group marking actual array of output signals.
Arlist	Internal name group marking input list of names of Corrector read signals.
Aslist	Internal name group marking input list of names of Bpm sources of input signals.
Awdata	Internal name group marking input array of weights.
Awork	Internal name group marking actual internal Work array.
/iwa(2)	Internal integer start and end pointers to working influence matrix **A** in the **Work** array. Copied from **A0,** and written over in fitting.
/iwa0(2)	Internal integer pointers to influence matrix **A0** in the **Work** array.
/iwb(2)	Internal integer pointers to decimal array **B** of values of active Bpms in the **Work** array. Compacted from **Bi** array according to status of input signals.
/iwbi(2)	Internal integer pointers to decimal array **Bi** of present Bpm values in the **Work** array.

/iwb0(2)	Internal integer pointers to decimal array **B0** of original Bpm values in the **Work** array.
/iwbptrs(2)	Internal integer pointers to integer array of pointers to Bpm modules in the **Work** array.
/iwbstatus(2)	Internal integer pointers to integer array of Bpm status values in the **Work** array. Two values are carried for each member. Status values of 3 or larger mark satsifactory inputs.
/iwbpos(2)	Internal integer pointers to integer array of Bpm position indices in the **Work** array. These indices, ordered by position in the lattice, refer to the position of the corresponding member in the *Sourcelist.* This complication essentially sorts the input signal list according to lattice position, as tracking by position is embedded in Micado procedure.
/iwbave(2)	Internal integer pointers to decimal array of sums of Bpm values in the **Work** array. (Used for time averaging.)
/iwbnave(2)	Internal integer pointers to integer array of counters of Bpm values in the **Work** array.
/iwbbeta(2)	Internal integer pointers to decimal array of Bpm beta values in the **Work** array.
/iwbmu(2)	Internal integer pointers to decimal array of Bpm mu values in the **Work** array.
/iwc(2)	Internal integer pointers to decimal array **C** of reading values of active Correctors in the **Work** array. Compacted from **Ci** array according to on - off, other status.
/iwci(2)	Internal integer pointers to decimal array of Corrector reading values **Ci** in the **Work** array.
/iwc0(2)	Internal integer pointers to decimal array **C0** of original Corrector values in the **Work** array.
/iwcptrs(2)	Internal integer pointers to integer array of pointers to Corrector modules in the **Work** array.
/iwcpos(2)	Internal integer pointers to integer array of Corrector position indices in the **Work** array. Sorts *Readlist* entries, like Bpms above.
/iwcave(2)	Internal integer pointers to decimal array of sums of Corrector values in the **Work** array. (For time averaging.)
/iwcnave(2)	Internal integer pointers to integer array of counters of Corrector values in the **Work** array.
/iwcstatus(2)	Internal integer pointers to integer array of Corrector status values in the **Work** array.
/iwcbeta(2)	Internal integer pointers to decimal array of Corrector beta values in the **Work** array.
/iwcmu(2)	Internal integer pointers to decimal array of Corrector mu values in the **Work** array.
/iwcorder(2)	Internal integer pointers to integer array of Corrector ordering values in the **Work** array.
/iwr(2)	Internal integer pointers to decimal array **R** of Bpm residual values in the **Work** array.
/iwweights(2)	Internal integer pointers to decimal array **W** of Weight values in the **Work** array.
/iwhh(2)	Internal integer pointers to decimal array **HH** of correlation values in the **Work** array. Set from **HH0** array, and written over during fitting.
/iwhh0(2)	Internal integer pointers to decimal array **HH0** of **A** * **A** correlation values in the **Work** array.
/iwhg(2)	Internal integer pointers to decimal array **HG** of **A, B** correlation values in **Work** array.
/nbpms(2)	Internal integer numbers of Bpms: 1). All noted in *Sourcelist,* 2). All active Bpms of first group, as noted by status.

/ncorrs(2)	Internal integer numbers of Correctors: All noted in *Readlist,* 2). All active Correctors of first group, as noted by status.
/jxy	Internal integer index for plane - (1) X, (2) Y.
/wncorr	Internal integer for working number of most useful correctors to be treated in fit, as given by *Ncorr,* or default.
/wbias	Internal decimal value for input signal bias, given in *Bias,* or default of 0.
/woffset	Internal decimal value for output signal bias, given, or default of 0.
/wtol	Internal decimal number for tolerance, given in *Tolerance* or default of 1.e-5.
/count(2)	Internal integer counting time steps.
/nok	Internal integer counting time steps for which no correction was needed.
/nmiss	Internal integer counting time steps for which correction was needed.
/calon	Internal Flag: True if calibration is in progress.
/Open/(2)	