

# **Implementation of new closed orbit correction procedures in the MAD-X program**

**W. Herr, SL Division, CERN, 1211-Geneva 23**

## **Abstract**

The possibility to correct a closed orbit is a basic functionality in the MAD program. During the rewrite leading to the MAD-X version, the closed orbit correction procedures have been completely re-written and substantially extended. Additional algorithms have been added and problems with existing algorithms have been solved. Furthermore, new options allow more sophisticated studies and better control of the correction procedures. In this note I shall describe these new features and their *raison d'être*. Most new features are the result of the immediate needs emerging from the extensive use of the program in realistic simulations. Some features which were practically never used have been dropped.

Geneva, Switzerland

17. October 2002

# 1 Introduction

The possibility to correct closed orbit distortions was implemented in the MAD8 program. The underlying algorithm was the MICADO [1] procedure. Although satisfactory for many applications, the implementation in MAD8 had several drawbacks:

- Only MICADO as algorithm available.
- Practically impossible to extend to other algorithms.
- Works only for circular machines, i.e. not for trajectories or beam lines.
- Limited features for extensive and realistic closed orbit studies.
- Correction may give unphysical results.

The last mentioned effect is a direct consequence of the straightforward use of the MICADO algorithm. While MICADO is the preferred algorithm when a limited number of correctors is desired, it can encounter problems when a large number or all available correctors are wanted. I shall treat this in more detail in a later section.

The purpose of this rewrite was to (in this order of priority):

- reproduce the existing functionality.
- allow the easy implementation of other algorithms.
- extend the algorithm to work also for trajectories.
- provide additional features that allow realistic closed orbit studies.
- improve the algorithms to avoid unphysical results.

The implementation of the above was largely driven by immediate needs during closed orbit studies for the LHC [2] and the CNGS beam line [3]. Some improvements were inspired by the procedures used in the COCU program [4, 5] which was used e.g. for the control of the SPS and LEP closed orbits and in various other laboratories.

## 2 Definition of the problem

The purpose of most closed orbit correction algorithms is to minimize the function:

$$S = \| b - b_t - A \cdot x \| \quad (1)$$

where  $b$  is a vector of length  $[m]$  containing the measured orbit,  $A$  the response matrix of size  $[m \times n]$  and  $x$  the vector containing the  $[n]$  correctors strengths. The vector  $b_t$  is the wanted or target orbit and most of the time the zero orbit. The norm  $\| \dots \|$  can be either a least squares norm  $l_2$  (e.g. for MICADO [1]) or the norm  $l_1$  (e.g. for SIMPLEX, optionally available in [4, 5]).

I have implemented three algorithms in MAD-X, all minimizing the least squares norm  $l_2$ . The first is the well-known MICADO algorithm, which is a best kick algorithm and best suited when a small number of correctors is desired. The second algorithm is a complete minimization using a pseudo inversion. A third algorithm is based on a Singular Value Decomposition (SVD) which can also be used to condition the problem for other algorithms. For that purpose, a new option MODE was introduced to define the algorithm and it can presently take the values MICADO, SVD or LSQ. The default is MICADO.

## 3 MICADO algorithm

### 3.1 Basic algorithm

The MICADO algorithm is well known and fully described in [1]. It is a best kick algorithm and in each step (iteration) of the correction it adds a kick to the correctors list and recomputes the needed strengths with the present list.

### 3.2 Parameters

The number of desired correctors (i.e. iterations, NCORR) has to be specified, otherwise all available correctors are used. The correction proceeds until all iterations are done or else when the desired quality of correction is reached, which is specified as a target r.m.s. orbit value (ERROR).

### 3.3 Extension to beam trajectories

The MICADO algorithm was extended to handle beam lines, i.e. trajectories. This was extensively used to study the performance of the CNGS proton beam line [3]. This requires a different setup of the problem, in particular of the response matrix to reflect the properties of a trajectory. It is controlled by a new option FLAG which can take the values *ring* or *line*. The default is *ring*.

### 3.4 Advantages and disadvantages

MICADO has the advantage that a good correction can be obtained with a limited number of correctors very efficiently. When more correctors are added, those already in the list are not dropped but new strengths are calculated for them. This can have two consequences. First, a corrector found in an early stage of a correction may be found inefficient when a larger number of correctors is considered. Secondly, and more important, linearly dependent pairs of correctors have the tendency to form local bumps or compensate each other when both are in the active list. Such defects make the correction useless. An example is given in appendix A. Two things have happened: the algorithm encountered a pair of correctors which are redundant and the procedure is aborted. In the list of correctors already found we see corrector strengths which are beyond any reasonable value. Missing monitors can cause such problems: if the phase advance between two correctors is a multiple of  $180^\circ$ , two columns (or rows) in matrix A (1) become linearly dependent. In other words, the effect of two correctors on the global closed orbit is identical except for a constant factor. The algorithms cannot deal with such a problem and the matrix has to be changed (i.e. re-conditioned) to avoid numerical problems during the minimization process. A division by zero and a program crash would be the consequences and the correction process is aborted.

These problems were encountered in abundance when LEP was first operated with a lattice with a phase advance of  $90^\circ$  per cell in both planes.

A solution would be to condition the response matrix before the MICADO minimization is performed.

## 4 Least squares algorithm

### 4.1 Basic algorithm

Our least squares problem

$$S = \| b - b_t - A \cdot x \| \quad (2)$$

can easily be solved by a pseudo inversion of the type

$$b - b_t = A \cdot x \longrightarrow (A^T \cdot A)^{-1} \cdot A^T \cdot (b - b_t) = x \quad (3)$$

## 4.2 Advantages

The main advantage of this straightforward least squares minimization is its simplicity. Matrix inversion algorithms are easily available and fast enough even for large machines.

## 4.3 Problems

The least squares minimization always uses all available correctors and is therefore more sensitive to wrong measurements than a correction with a few correctors. Furthermore it suffers from the same problem as MICADO: an ill-conditioned matrix leads to malfunctioning and crazy results. Obviously, if everything works correctly, it gives the same result as a correction with MICADO using all correctors. The correction results of a least squares minimization is shown in appendix B. The same input was used as for appendix A. We observe again the appearance of local bumps and large correctors strengths. Since this method **always** uses **all** correctors it is almost impossible to get a good correction when monitors are missing or some correctors are badly positioned. It is therefore mandatory to condition the response matrix before such a minimization is performed. Such a conditioning procedure is implemented and described below. It was already used successfully for the correction of LEP closed orbits [6, 7].

## 5 Singular Value Decomposition (SVD)

The two methods described above can only work as long as matrix A (1) is not rank deficient, in which case an infinite number of solutions exists. We also have to consider the cases where A is ‘very close’ to be rank deficient (i.e. A is close to a matrix of the next lower rank) or ‘numerically’ rank deficient. Such cases may not lead to an infinite number of solutions and therefore to an ill-defined problem, but may lead to unphysical results. Both cases have to be treated. In practice, in particular when large machines like the LHC are handled, rounding errors can make this a non-trivial problem. Fortunately, with the Singular Value Decomposition (SVD) we have a tool available that efficiently provides all the information we need: the rank of the matrix, a measure how close it is to a matrix of lower rank, and the singular values themselves.

### 5.1 Basic principle

The principle of a SVD is the following: for any rectangular matrix A[m×n] with rank r, there exist unitary matrices U[m×m] and V[n×n] such that

$$A = U \cdot \Sigma \cdot V^T \quad \text{with} \quad \Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdot & 0 & 0 \\ 0 & \sigma_2 & \cdot & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \sigma_r & 0 \\ 0 & 0 & \cdot & 0 & 0 \\ 0 & 0 & \cdot & 0 & 0 \\ 0 & 0 & \cdot & 0 & 0 \end{bmatrix} \quad (4)$$

i.e.  $\Sigma$  is a  $m \times n$  matrix whose elements are all zero except for the diagonal elements  $\sigma_1 \dots \sigma_r$ , which are the singular values of the matrix A and are ordered according to their values:  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$ .

The algorithm used here is adapted from Golub and Reinsch in [9].

### 5.2 How does it help ?

It can be shown, that the singular values  $\sigma_i$  provide a measure for the ‘singularity’ of the matrix A. In particular, the smallest singular value  $\sigma_r$  is a measure how close A is to be rank deficient [8].

The procedure is now as follows: find the singular values of the response matrix A. Identify the correctors which correspond to the zero values or too small  $\sigma_k$  and remove them from the available corrector set.

### 5.3 Strategy and implementation

Since both the least squares minimization as well as MICADO may require a pre-conditioning of the response matrix, the SVD is implemented as a separate module which can be activated by either of the two algorithms (or any other algorithm added in the future).

The SVD is used to identify correctors that should be omitted from the solution. The necessary information is contained in the matrices  $U$  and  $V$ . A small complication is due to the algorithm that is implemented. My implementation of the SVD decomposition requires that the number of rows is larger than (or equal to) the number of columns. Since we are interested in handling all cases, i.e. when the number of correctors is either larger or smaller than the number of monitors, we have to condition either the matrix  $A$  or its transposed matrix. The analysis of the  $U$  or  $V$  is required, depending on the case. For the  $k_{th}$  singular value, the  $k_{th}$  column of either  $U$  or  $V$  is examined to identify the corresponding corrector.<sup>1</sup>

The SVD conditioning is activated by adding the option **COND=1** in the CORRECT command. The default value is **COND=0**, i.e. no decomposition and conditioning is done.

### 5.4 Some subtleties

To avoid an ill-conditioned matrix we have to identify singular values which are "zero" or "too small". While the first is (rather) straightforward, the latter is not obvious and depends on the problem under study. As an illustration I show in Fig.1 the distribution of the singular values  $\sigma_i$  for an LHC study. They

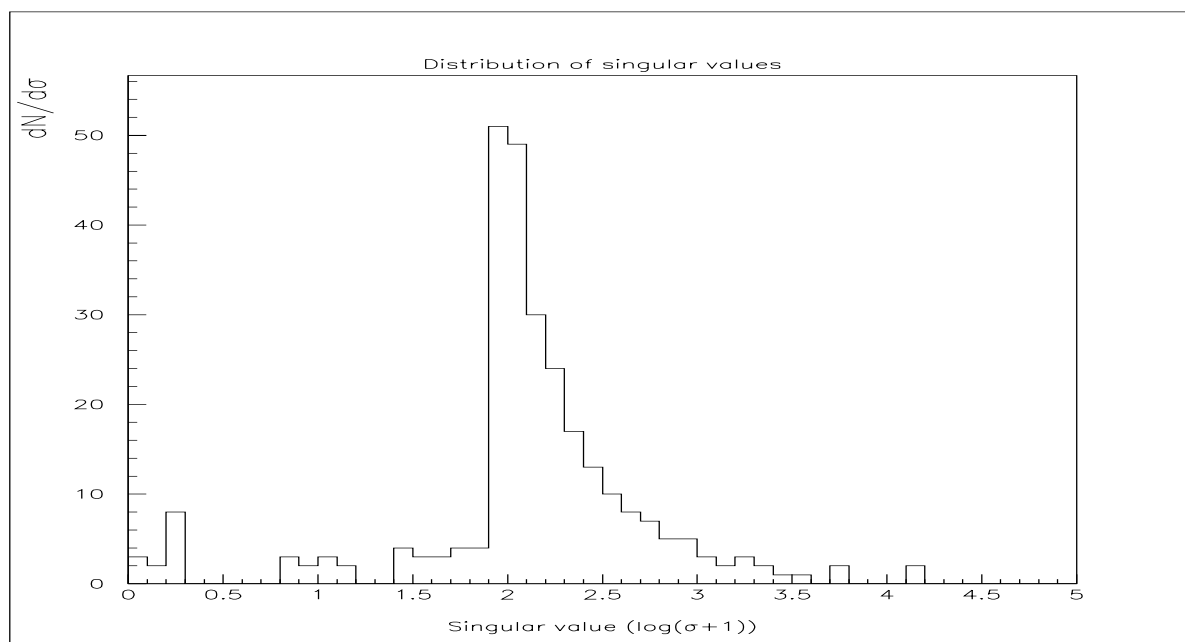


Figure 1: *Distribution of singular values  $\sigma_i$  for a LHC closed orbit correction.*

range from 15600 to  $10^{-14}$ . For better illustration I show the distribution of  $\log_{10}(\sigma_i + 1.0)$ . While the "zero" values and the finite values are clearly separated, some intermediate values may indicate potential problems. It is necessary to introduce empirical parameters such as a cutoff value and the number of singular values to consider. This is not the same as a numerical tolerance in the mathematical sense for which some good estimates can be defined (usually using the precision of the calculation and the largest singular value) and which work well to define the "zeroes". During the analysis of the  $U$  or  $V$  matrices a further parameter is needed. Unfortunately these parameters are not unique constants and for smaller

<sup>1</sup>The implementation in COCU required the number of monitors to be larger than the number of correctors [6].

machines such as ELETTRA (which was used as an example for a small machine during the program development) they can be very different. A change of parameters may cause either too large or too small a number of correctors being discarded. The present set of parameters is optimized for the LHC and may be refined with increasing experience.

## 5.5 Results

The results of a MICADO correction (from appendix A) after a SVD conditioning are shown in appendix C. A list of the correctors which are removed is given. In total 9 correctors are discarded. The comparison with appendices A and B shows that these are the previously offending correctors which proves that the SVD did what was expected. The maximum required corrector strength is now small ( $\approx 30 \mu\text{rad}$ ) and the correction result is excellent: the r.m.s. was improved from 6.2 mm to 0.08 mm and the peak-to-peak orbit from above 40 mm to 0.6 mm.

## 6 SVD as correction mode

The results of the singular value decomposition can directly be used to solve the least squares problem. It is trivial to show that:

$$x = V \cdot \Sigma^{-1} \cdot U^T \cdot b \quad \text{with} \quad \Sigma^{-1} = \text{diag}\left(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \frac{1}{\sigma_3}, \dots\right) \quad \text{of size } [n \times m] \quad (5)$$

is a solution to the problem, using all available correctors. A matrix inversion algorithm like for (3) is not necessary because  $\Sigma^{-1}$  is trivial. The solution above has the smallest  $l_2$  norm of all minimizers. Therefore a particular feature of this mode of correction is that the calculated strengths are distributed and shared in case of redundant correctors, thus reducing the maximum required strengths. This may be used with advantage to correct the orbits of real machines where the strength of correctors is limited. This feature is illustrated in appendices F and G. In appendix F the correction results are shown for a simulated orbit in ELETTRA using the MICADO method, preceded by a SVD conditioning. The appendix G shows the same input orbit corrected with the SVD method and it is clearly demonstrated that now the strengths are shared. For a good illustration I have installed always doublets of correctors side by side to make it obvious. Although this feature may have some advantages, it is not always the desired solution since it can inhibit strictly local corrections and produce small, closed bumps, however not of the type seen before.

## 7 Additional or modified features

Some additional features and commands have been added to either control the correction algorithms or to allow more extensive and realistic orbit studies [2]. In particular when the orbit correction is run inside a loop to test a large number of cases (possibly many thousands) it is important to:

- catch an abnormal situation that might normally lead to an abortion.
- provide the possibility to simulate erroneous or missing measurements.
- store relevant information for later analysis in computer accessible form.

### MONERROR:

When **MONERROR=1** is specified, the monitor alignment errors assigned with a EALIGN command are taken into account, otherwise they are ignored. (In EALIGN: MREX and MREY). The default value is 0.

### MONON:

This option is used as **MONON=r** and a fraction 1-r of all selected monitors is randomly disabled. An example with and without SVD is shown in appendices D and E. It is clear that only with a proper conditioning a correction can succeed when a large number of monitors is unavailable (30% in the example). The conditioning has removed correctors which became redundant when monitors were discarded. The optics and layout is most important in that case. The present layout of the LHC with beam position monitors at every quadrupole for both planes is far more robust than the LEP layout and missing monitors can be handled much easier.

#### **MONSCALE:**

This option allows to use scaling errors assigned to the monitors with the error command **EALIGN** when it is set to 1. The default is 0. (In **EALIGN**: **MSCALX** and **MSCALY**).

#### **MONCUT:**

When **MONCUT** is specified, all beam positions are normalized to their respective  $\beta$ -function and all monitors with values above the number of r.m.s. beam sizes given as the argument are discarded. Default is 0.0 (no cut).

#### **CORRLIM:**

A limit on the maximum corrector strength can be given and a **WARNING** is issued if it is exceeded by one or more correctors. Default is 1.0 mrad.

A variable **CORRMAX** is always set and available which contains the maximum strength of all correctors found during a **MICADO** correction.

#### **RESOUT:**

This command outputs the results for all monitors and all correctors in a computer readable format if its integer argument is larger than 0. The argument is added at the beginning of each line of the output (e.g. allows to tag the output with a loop variable). It is very useful to analyze runs with loops to produce large statistics.

#### **TWISSUM:**

If the argument of **twissum** is larger than 0, it prints maximum orbit and r.m.s. for both planes taken from the Twiss summary table in computer readable form. Allows to study orbit excursions at elements which are not monitors or correctors and therefore are not visible in the orbit correction program. The argument is added to the output. Only for output: no correction is made, all other commands are ignored.

#### **PLANE:**

If the attribute of **PLANE** is x, the horizontal orbit is corrected. If y is specified then the vertical orbit is corrected. To correct the orbit in both planes, the correction has to be called twice, once with x and once with the y attribute. This is different from the **MAD8** implementation. The default value is x.

#### **COPTION,SEED=integer,PRINT=integer:**

The **SEED** option selects a particular sequence of random values. It is an integer in the range 0 to 999999999 (default: 123456789). **SEED** without an argument continues with the current sequence of random values. **SEED** may be an expression. Random numbers are used by the command **MONON**. Every time **COPTION** is called with the same parameter, the seed is set to *integer*, i.e. subsequent commands **COPTION** and **MONON** with the same parameters will discard the same monitors.

The **PRINT** flag can take integer values and controls the printout to the screen. In general: the higher its value the more printout is produced. For **PRINT=0** no output is produced. The default value is 1 (Correction summary is given).



## 8 Abandoned features and commands

Some commands and options have been abandoned because they are practically never used or the functionality is provided by other (often simpler) means. These are:

- **ITERATE**: can be done using loops or macros which allow a better control.
- **ITERMAX**: related to the command **ITERATE**.
- **C1LIST, C2LIST**: are replaced by **CLIST**.
- **M1LIST, M2LIST**: are replaced by **MLIST**.
- **THREADER, THRTOL, WRORBIT**: the beam threader is not anymore part of the orbit correction module and will be provided as a separate module in the future.
- All commands related to correction of dispersion.

## 9 Timing considerations

In Tab.1 I have summarized some measurements of the execution time needed for different types of algorithms and number of correctors wanted (MICADO only). At the start I had 536 monitors and 270

	Total time (s)	Time per corrector (ms)
MICADO, 270 correctors	0.960	3.56
MICADO, 200 correctors	0.690	3.45
MICADO, 100 correctors	0.370	3.69
LSQ, 270 correctors,no SVD	2.200	8.15
LSQ, 270 correctors,after SVD	1.200	4.45
SVD conditioning	6.10	NA
SVD correction	5.90	21.9

Table 1: *Time needed for orbit correction for different modes and number of correctors desired (MICADO). Numbers are average of 10 runs.*

correctors. The numbers in Tab.1 correspond to runs on a HP KAYAK, Pentium 3 with 666 MHz, and 256 Mb RAM. The compiler used was g77 using the flags: **-O3 -fno-second-underscore -funroll-loops**.

In all cases the time needed for the orbit correction is not significant compared to the setup time and Twiss calculation. With the coding of the algorithms I tried to minimize (or avoid) any overhead.



## References

- [1] B. Autin and Y. Marti; *Closed Orbit Correction of A.G. Machines using a limited Number of Magnets*; CERN ISR MA/73-17(1973)
- [2] B. Salvant; *Simulation of Large Hadron Collider closed orbit with faulty Beam Position Monitors*, unpublished, (2002).
- [3] W. Herr and M. Meddahi; *Trajectory correction studies for the CNGS proton beam line*, SL-Note-2002-015 AP.
- [4] W. Herr, T. Limberg, J. Miles, R. Schmidt; *A new Closed Orbit Correction procedure for the CERN-SPS*;  
1989 Particle Accelerator Conference, Chicago, USA, March 20-23
- [5] D. Brandt, W. Herr, J. Miles and R. Schmidt; *A New Closed Orbit Correction Procedure for the CERN SPS and LEP*;  
Nucl. Instr. and Meth., **A293** (1990) 305. and CERN/SPS/89-48 (AMS).
- [6] W. Herr; *Algorithms and procedures used in the orbit correction packages COCU*;  
CERN/SL/95-07 (AP).
- [7] W. Herr; *How do we deal with missing monitors and redundant correctors in COCU ?*;  
SL/Note 95-29 (AP).
- [8] S. Barnett; "Matrices, Methods and Applications", Clarendon Press, Oxford (1990)
- [9] J.H. Wilkinson and C. Reinsch, eds.; "Handbook for Automatic Computation", Vol.2, Springer Verlag, New York (1971).

## 10 Appendix A

```
COPTION,PRINT=2;  
CORRECT, FLAG=ring,MODE=micado,cond=0,NCORR=300,ERROR= 1E-6,  
MONERROR=1,MONON=1.0000, PLANE=x, RESOUT=1;
```

```
536 monitors and 279 correctors found in input  
536 monitors and 276 correctors enabled  
enter MICADO correction ...  
Fewer correctors available than requested by ncorr  
you want 300, you get 276  
ncorr reset to 276
```

```
start MICADO correction with 276 correctors
```

```
Correction process aborted  
during 117th iteration  
Last r.m.s.: 2.16694164  
Last p-t-p.: 18.8062973  
Division by zero expected  
Probably two kickers too close  
SUSPECTED KICKER: 240  
mcbyh.a4r8.b1:1  
MICADO correction completed with error code -1
```

```
CORRECTION SUMMARY:
```

```
rms before correction: 6.196049 mm  
rms after correction: 2.166743 mm
```

```
ptp before correction: 40.200778 mm  
ptp after correction: 18.806296 mm
```

```
Corrector: Before: After: Difference:  
            (mrad) (mrad) (mrad)  
mcbx.1r1:1 0.000000 -0.000126 -0.000126  
mcbx.2r1:1 0.000000 0.000165 0.000165  
mcbxa.3r1:1 0.000000 0.000235 0.000235  
mcbyh.4r1.b1:1 0.000000 -0.005319 -0.005319  
mcbch.5r1.b1:1 0.000000 0.008037 0.008037  
mcbch.7r1.b1:1 0.000000 -0.025337 -0.025337  
mcbch.9r1.b1:1 0.000000 -0.007688 -0.007688  
mcbh.11r1.b1:1 0.000000 0.000000 0.000000  
mcbh.13r1.b1:1 0.000000 0.004950 0.004950  
mcbh.15r1.b1:1 0.000000 0.000000 0.000000  
mcbh.17r1.b1:1 0.000000 -0.010094 -0.010094  
. . .  
mcbch.712.b1:1 0.000000 0.032420 0.032420
```

mcbyh.b5l2.b1:1 0.000000 -1.566963 -1.566963  
mcbyh.a5l2.b1:1 0.000000 1.519788 1.519788  
mcbyh.4l2.b1:1 0.000000 -0.006876 -0.006876  
mcbxa.3l2:1 0.000000 0.063558 0.063558  
mcbx.2l2:1 0.000000 -0.190818 -0.190818  
mcbx.1l2:1 0.000000 0.083215 0.083215  
mcbx.1r2:1 0.000000 -0.565183 -0.565183  
mcbx.2r2:1 0.000000 1982.424561 1982.424561  
mcbxa.3r2:1 0.000000 -4783.375977 -4783.375977  
mcbyh.a4r2.b1:1 0.000000 47591.078125 47591.078125  
mcbyh.b4r2.b1:1 0.000000 -45852.394531 -45852.394531  
mcbyh.5r2.b1:1 0.000000 0.021354 0.021354  
mcbch.6r2.b1:1 0.000000 -0.001374 -0.001374  
mcbch.8r2.b1:1 0.000000 -0.001643 -0.001643  
mcbch.10r2.b1:1 0.000000 0.002070 0.002070

.  
. .  
.

stop;

++++  
+ MAD-X 1.03 finished normally +  
++++

## 11 Appendix B

```
CORRECT, FLAG=ring,MODE=lsq,cond=0,NCORR=300,ERROR= 1E-6,  
MONERROR=1,MONON=1.0000, PLANE=x, RESOUT=1;
```

```
536 monitors and 279 correctors found in input  
536 monitors and 276 correctors enabled
```

```
start LEAST SQUARES correction with all correctors
```

```
end LEAST SQUARES correction with all correctors
```

```
CORRECTION SUMMARY:
```

```
rms before correction: 6.196049 mm
```

```
rms after correction: 62.691529 mm
```

```
ptp before correction: 40.200778 mm
```

```
ptp after correction: 331.895102 mm
```

```
Corrector:  Before:      After:      Difference:  
            (mrad)      (mrad)      (mrad)  
mcbx.1r1:1  0.000000  0.000000  0.000000  
mcbx.2r1:1  0.000000  0.000000  0.000000  
mcbxa.3r1:1  0.000000  -0.000001  -0.000001  
mcbyh.4r1.b1:1  0.000000  0.000017  0.000017  
mcbch.5r1.b1:1  0.000000  -0.000026  -0.000026  
mcbch.7r1.b1:1  0.000000  0.000081  0.000081  
.  
.  
mcbch.815.b1:1  0.000000  -0.000829  -0.000829  
mcbch.615.b1:1  0.000000  0.000151  0.000151  
mcbyh.b415.b1:1  0.000000  26.537599  26.537599  
mcbyh.a415.b1:1  0.000000  -27.177032  -27.177032  
mcbxa.315:1  0.000000  0.749252  0.749252  
mcbx.215:1  0.000000  -0.199658  -0.199658  
mcbx.115:1  0.000000  -0.000000  -0.000000  
.  
.  
stop;
```

```
++++  
+ MAD-X 1.03 finished normally +  
++++
```

## 12 Appendix C

```
COPTION,PRINT=1;  
CORRECT, FLAG=ring,MODE=micado,cond=1,NCORR=300,ERROR= 1E-6,  
MONERROR=1,MONON=1.0000, PLANE=x, RESOUT=1;
```

```
536 monitors and 279 correctors found in input  
536 monitors and 277 correctors enabled
```

Removed after SVD:

=====

```
Removed: 33 mcbyh.b5l2.b1:1  
Removed: 40 mcbx.2r2:1  
Removed: 41 mcbxa.3r2:1  
Removed: 42 mcbyh.a4r2.b1:1  
Removed: 43 mcbyh.b4r2.b1:1  
Removed: 134 mcbyh.b4l5.b1:1  
Removed: 242 mcbyh.a4r8.b1:1  
Removed: 243 mcbyh.b4r8.b1:1  
Removed: 274 mcbyh.b4l1.b1:1
```

After SVD conditioning:

```
536 monitors and 268 correctors enabled  
Fewer correctors available than requested by ncorr  
you want 300, you get 268  
ncorr reset to 268
```

```
start MICADO correction with 268 correctors
```

CORRECTION SUMMARY:

```
rms before correction: 6.196049 mm  
rms after correction: 0.078594 mm
```

```
ptp before correction: 40.200778 mm  
ptp after correction: 0.604464 mm
```

```
Max strength: 2.984711e-02 should be less than 1.000000e+00
```

stop;

```
++++  
+ MAD-X 1.03 finished normally +  
++++
```

### 13 Appendix D

```
COPTION,PRINT=2;  
CORRECT, FLAG=ring,MODE=micado,cond=0,NCORR=300,ERROR= 1E-6,  
MONERROR=1,MONON=0.7000, PLANE=x, RESOUT=1;
```

```
536 monitors and 255 correctors found in input  
165 monitors disabled on request  
371 monitors and 255 correctors enabled
```

```
enter MICADO correction ...  
Fewer correctors available than requested by ncorr  
you want 300, you get 255  
ncorr reset to 255
```

```
start MICADO correction with 255 correctors
```

#### CORRECTION SUMMARY:

```
rms before correction: 6.050892 mm  
rms after correction: 2.748951 mm
```

```
ptp before correction: 32.260361 mm  
ptp after correction: 23.302443 mm
```

Corrector:	Before:	After:	Difference:
	(mrad)	(mrad)	(mrad)
mcbx.1r1:1	0.000000	0.000000	0.000000
mcbx.2r1:1	0.000000	0.000000	0.000000
mcbxa.3r1:1	0.000000	0.000000	0.000000
mcbch.5r1.b1:1	0.000000	0.000000	0.000000
mcbch.7r1.b1:1	0.000000	0.000000	0.000000
mcbch.9r1.b1:1	0.000000	0.004403	0.004403
mcbh.11r1.b1:1	0.000000	0.000000	0.000000
mcbh.13r1.b1:1	0.000000	0.021030	0.021030
mcbh.15r1.b1:1	0.000000	0.000000	0.000000
mcbh.17r1.b1:1	0.000000	-0.026929	-0.026929
mcbh.19r1.b1:1	0.000000	0.000000	0.000000
mcbh.21r1.b1:1	0.000000	0.007789	0.007789
mcbh.23r1.b1:1	0.000000	0.000000	0.000000
mcbh.25r1.b1:1	0.000000	-0.015173	-0.015173
mcbh.27r1.b1:1	0.000000	0.000000	0.000000
mcbh.29r1.b1:1	0.000000	-0.038423	-0.038423
mcbh.31r1.b1:1	0.000000	-0.006896	-0.006896
mcbh.33r1.b1:1	0.000000	-0.016648	-0.016648
mcbh.33l2.b1:1	0.000000	-0.015414	-0.015414
mcbh.31l2.b1:1	0.000000	0.014058	0.014058
mcbh.29l2.b1:1	0.000000	-0.035511	-0.035511
mcbh.27l2.b1:1	0.000000	-0.027616	-0.027616
mcbh.25l2.b1:1	0.000000	-0.004509	-0.004509

```
mcbh.2312.b1:1 0.000000 -0.019408 -0.019408
mcbh.2112.b1:1 0.000000 0.003559 0.003559
mcbh.1912.b1:1 0.000000 -102557.906250 -102557.906250
mcbh.1712.b1:1 0.000000 -13553.422852 -13553.422852
mcbh.1512.b1:1 0.000000 -102974.234375 -102974.234375
mcbh.1312.b1:1 0.000000 -0.040209 -0.040209
mcbh.1112.b1:1 0.000000 0.005863 0.005863
mcbch.912.b1:1 0.000000 0.000000 0.000000
mcbch.712.b1:1 0.000000 0.000000 0.000000
mcbxa.312:1 0.000000 0.488316 0.488316
mcbx.212:1 0.000000 -1.646360 -1.646360
mcbx.112:1 0.000000 0.738529 0.738529
mcbx.1r2:1 0.000000 -71230.867188 -71230.867188
mcbx.2r2:1 0.000000 69824.281250 69824.281250
mcbxa.3r2:1 0.000000 -32636.052734 -32636.052734
mcbch.6r2.b1:1 0.000000 -2695.031006 -2695.031006
mcbch.8r2.b1:1 0.000000 3119.203125 3119.203125
mcbch.10r2.b1:1 0.000000 -2368.102539 -2368.102539
mcbh.12r2.b1:1 0.000000 83.013321 83.013321
mcbh.14r2.b1:1 0.000000 -110.816986 -110.816986
mcbh.16r2.b1:1 0.000000 0.015726 0.015726
mcbh.18r2.b1:1 0.000000 -0.014348 -0.014348
mcbh.20r2.b1:1 0.000000 0.040710 0.040710
mcbh.22r2.b1:1 0.000000 -0.018064 -0.018064
mcbh.24r2.b1:1 0.000000 0.001933 0.001933
mcbh.26r2.b1:1 0.000000 0.001103 0.001103
mcbh.28r2.b1:1 0.000000 0.011478 0.011478
mcbh.30r2.b1:1 0.000000 0.000000 0.000000
mcbh.32r2.b1:1 0.000000 0.008402 0.008402
mcbh.3413.b1:1 0.000000 0.000000 0.000000
mcbh.3213.b1:1 0.000000 -0.010138 -0.010138
mcbh.3013.b1:1 0.000000 0.000000 0.000000
mcbh.2813.b1:1 0.000000 0.004596 0.004596
```

```
stop;
```

```
+++++
+ MAD-X 1.03 finished normally +
+++++
```



## 14 Appendix E

```
CORRECT, FLAG=ring,MODE=micado,cond=1,NCORR=300,ERROR= 1E-6,  
MONERROR=1,MONON=0.7000, PLANE=x, RESOUT=1;
```

536 monitors and 257 correctors found in input

165 monitors disabled on request

371 monitors and 255 correctors enabled

```
Removed: 6 mcbh.11r1.b1:1  
Removed: 25 mcbh.1912.b1:1  
Removed: 27 mcbh.1512.b1:1  
Removed: 35 mcbx.1r2:1  
Removed: 36 mcbx.2r2:1  
Removed: 37 mcbxa.3r2:1  
Removed: 39 mcbch.8r2.b1:1  
Removed: 77 mcbh.23r3.b1:1  
Removed: 80 mcbh.29r3.b1:1  
Removed: 81 mcbh.31r3.b1:1  
Removed: 125 mcbxa.315:1
```

After SVD conditioning:

371 monitors and 244 correctors enabled

enter MICADO correction ...

```
Fewer correctors available than requested by ncorr  
you want 300, you get 244  
ncorr reset to 244
```

start MICADO correction with 244 correctors

CORRECTION SUMMARY:

```
rms before correction: 6.050892 mm  
rms after correction: 0.102153 mm
```

```
ptp before correction: 32.260361 mm  
ptp after correction: 1.472243 mm
```

stop;

```
++++  
+ MAD-X 1.03 finished normally +  
++++
```

## 15 Appendix F

```
COPTION,PRINT=2;  
CORRECT, FLAG=ring,MODE=micado,cond=1,NCORR=300,ERROR= 1E-6,  
MONERROR=1,MONON=0.7000, PLANE=x, RESOUT=1;
```

10 monitors and 10 correctors enabled

Removed: 0 cbh.10:1

Removed: 2 cbh.20:1

Removed: 4 cbh.30:1

Removed: 6 cbh.40:1

Removed: 8 cbh.50:1

After SVD conditioning:

10 monitors and 5 correctors enabled

start MICADO correction with 5 correctors

CORRECTION SUMMARY:

rms before correction: 7.264963 mm

rms after correction: 0.000692 mm

ptp before correction: 25.307761 mm

ptp after correction: 0.002615 mm

Monitor:	Before:	After:	Difference:
	(mm)	(mm)	(mm)
bpha:1	4.368	-0.000	-4.368
bpv1:1	4.323	-0.000	-4.323
bph2:1	1.813	0.000	-1.813
bph3:1	-12.667	-0.000	12.667
bpv3:1	-4.772	0.000	4.772
bph4:1	-2.027	-0.000	2.027
bpv4:1	2.995	-0.000	-2.995
bph5:1	12.641	-0.001	-12.641
bpv5:1	7.534	0.002	-7.532
bph1:1	11.924	-0.000	-11.924

Corrector:	Before:	After:	Difference:
	(mrad)	(mrad)	(mrad)
cbh.11:1	0.000000	-0.082002	-0.082002
cbh.21:1	0.000000	0.007105	0.007105
cbh.31:1	0.000000	-0.000073	-0.000073
cbh.41:1	0.000000	0.000011	0.000011
cbh.51:1	0.000000	-0.057114	-0.057114

```
++++  
+ MAD-X 1.03 finished normally +  
++++
```

## 16 Appendix G

```
COPTION,PRINT=2;  
CORRECT, FLAG=ring,MODE=svd,cond=0,NCORR=300,ERROR= 1E-6,  
MONERROR=1,MONON=1.0000, PLANE=x, RESOUT=1;
```

8 monitors and 10 correctors enabled

SVD CORRECTION STARTED

CORRECTION SUMMARY:

```
rms before correction: 7.634667 mm  
rms after correction: 0.000466 mm
```

```
ptp before correction: 25.307761 mm  
ptp after correction: 0.001642 mm
```

Monitor:	Before:	After:	Difference:
	(mm)	(mm)	(mm)
bpha:1	4.368	-0.001	-4.368
bph2:1	1.813	0.000	-1.813
bph3:1	-12.667	0.000	12.667
bph4:1	-2.027	-0.000	2.027
bpv4:1	2.995	0.000	-2.995
bph5:1	12.641	-0.000	-12.641
bpv5:1	7.534	0.001	-7.533
bph1:1	11.924	-0.000	-11.924

Corrector:	Before:	After:	Difference:
	(mrad)	(mrad)	(mrad)
cbh.10:1	0.000000	-0.041009	-0.041009
cbh.11:1	0.000000	-0.041003	-0.041003
cbh.20:1	0.000000	0.003552	0.003552
cbh.21:1	0.000000	0.003502	0.003502
cbh.30:1	0.000000	0.000028	0.000028
cbh.31:1	0.000000	0.000023	0.000023
cbh.40:1	0.000000	-0.000012	-0.000012
cbh.41:1	0.000000	0.000001	0.000001
cbh.50:1	0.000000	-0.028546	-0.028546
cbh.51:1	0.000000	-0.028527	-0.028527

```
++++  
+ MAD-X 1.03 finished normally +  
++++
```