MATLAB-BASED LOCO

J. Safranek[*], G. Portmann, A. Terebilo, SSRL/SLAC, Stanford, CA 94309, USA
C. Steier, LBNL, Berkeley, CA 94720, USA

Abstract

The storage ring linear optics debugging code LOCO (Linear Optics from Closed Orbits)[1] has been rewritten in MATLAB and linked to the accelerator modeling code AT [2]. LOCO uses the measured orbit response matrix to determine normal and skew quadrupole gradients. A MATLAB GUI provides a greatly improved user interface with graphical display of the fitting results. The option of including the shift in orbit with rf-frequency in the orbit response matrix has been added so that the model is adjusted to match the measured dispersion. This facilitates control of the horizontal dispersion, which is important for achieving small horizontal emittance. Also included are error bar calculation, outlier data rejection, accommodation of single-view BPMs (beam position monitors), and the option of including coupling in the fit. The code was written to allow the flexibility of linking it to other accelerator modeling codes.

1 INTRODUCTION

LOCO is an algorithm for debugging the optics of storage rings. LOCO can be used to calibrate the normal and skew gradients in each quadrupole in a storage ring, as well as calibrating the gains and cross-plane coupling in BPMs and strength and rotational alignment of steering magnets. LOCO can also be used to find the changes in quadrupole gradients that best restore the design optics in a storage ring, and to find the settings for skew quadrupoles to correct horizontal to vertical betatron coupling. The algorithm has been successfully applied at many laboratories. (See [1,3-5] for examples.)

The main goal in porting to MATLAB has been to make the application easier to use. The LOCO code was originally written in FORTRAN. The FORTRAN code has a somewhat awkward user interface. The FORTRAN LOCO code itself does not calculate the model response matrix. It requires a separate modeling code such as MAD [6] to calculate the model matrix. The linking between the FORTRAN LOCO code and MAD was cumbersome. The MATLAB LOCO code has a cleaner link to the MATLAB-based AT accelerator modeling code as well as a greatly improved GUI-driven user interface. The MATLAB code also incorporates several new features that were not available with the FORTRAN code. Although MATLAB LOCO was written with AT as the tracking code, a mechanism to connect to other modeling codes has been provided.

2 LOCO ALGORITHM

See reference [1] for more details on the LOCO algorithm.

2.1 Characterizing Linear Optics

The orbit response matrix, M, is the shift at each BPM for a change in strength of each steering magnet,

$$\begin{pmatrix} \vec{x} \\ \vec{y} \end{pmatrix} = M \begin{pmatrix} \vec{\theta}_x \\ \vec{\theta}_y \end{pmatrix}. \quad (1)$$

The storage ring quadrupole gradients are determined by varying the gradients in a computer model of the storage ring to minimize the χ^2 difference between the model and measured response matrices.

$$\chi^2 = \sum_{i,j} \frac{(M_{\text{mod},ij} - M_{meas,ij})^2}{\sigma_i^2}, \quad (2)$$

where σ_i is the measured noise level on the ith BPM. Differences between the diagonal sub-matrices (e.g. the response at horizontal BPMs from horizontal steerers) arise from quadrupole gradient differences between the model and real storage ring as well as differences in the BPM gains and steering magnet calibrations. Therefore the gains and steerer calibrations are also fit in the χ^2 minimization. The result is a model that best reproduces the measured response matrix. The model also accurately reproduces the beta functions, magnet gradients, BPM gains, and steering magnet calibrations of the real storage ring.

In order to calibrate the storage ring coupling, the off-diagonal sub-matrices (vertical orbit shift with horizontal steerers and visa-versa) can be included in χ^2. Skew gradients, BPM coupling, and steering magnet tilt all contribute to the off-diagonal sub-matrices. When including coupling, all three are included in the χ^2 minimization.

2.2 Optics and Coupling Correction

Section 2.1 described how to use LOCO to calibrate individual gradient errors. Alternatively, the LOCO algorithm can be used to find the best gradient changes to restore the design optics. For example, one can find the quadrupole gradient changes that best correct for focusing in a wiggler or undulator magnet, or one can find the skew gradient strengths for correcting coupling.
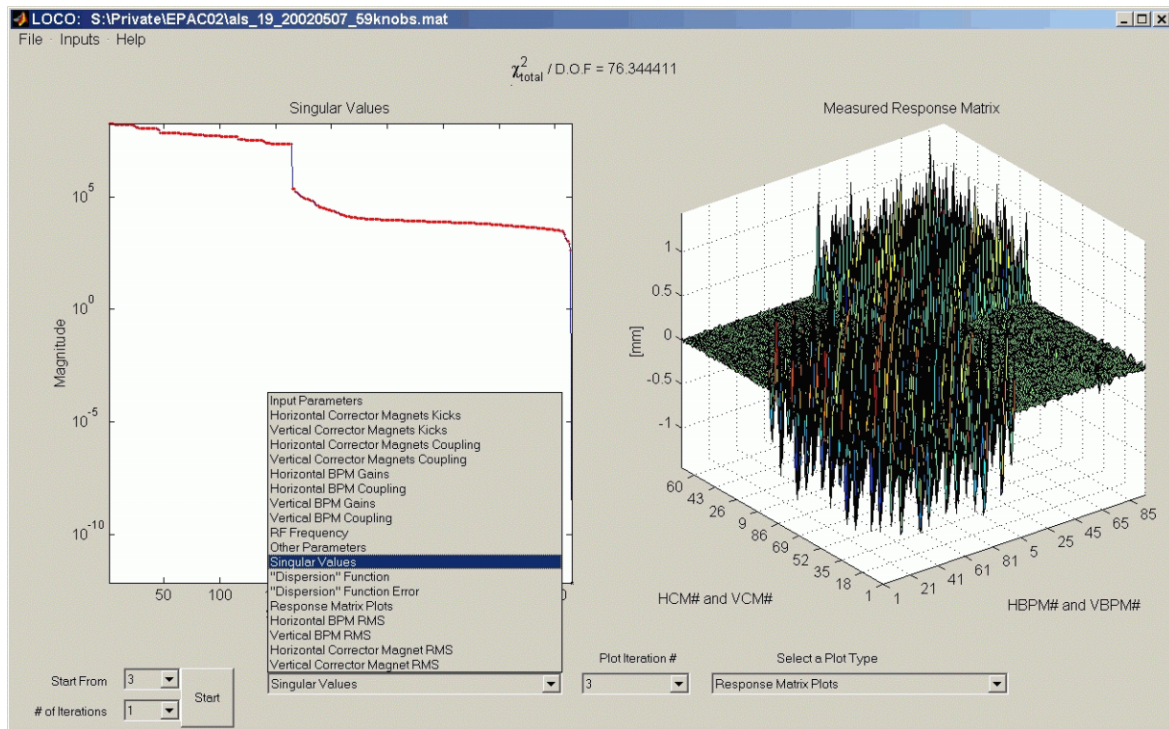
Figure 1. MATLAB LOCO GUI.

3 CODE FEATURES

3.1 Graphical User Interface

Figure 1 shows the MATLAB-LOCO GUI. The information displayed on each plot is selectable from the plot selection menu shown expanded in the figure.

LOCO uses an iterative method to converge to minimum χ^2. The iteration number used in the two plots is selectable. The "Start" button in the bottom left of Fig. 1 is used to start LOCO fitting for a user-specified number of iterations. The iteration number from which to start can also be selected.

The "Inputs" pull down menu is used to make choices concerning the fitting in each iteration. Many of the options in the "Inputs" menu are discussed below.

3.2 Orbit Response Matrix Calculation

There are two options in MATLAB LOCO to calculate the response matrix.

In the linear approximation the orbit response matrix is calculated from the numerically obtained 4-by-4 transfer matrixes at each corrector and BPM, the model dispersion function, and model momentum compaction factor. This option is useful on first few iterations of the LOCO algorithm.

The iterative search for a closed orbit in the presence of the corrector magnet kick is slower but includes the non-linear effects due to sextupoles and other non-linear elements. In this case LOCO MATLAB makes a call to a modeling code (AT) to find the solution constrained by the requirement of a constant revolution frequency set by the RF.

3.3 Dispersion Fitting

An option to include dispersion fitting has been added to MATLAB LOCO. The shift in orbit with rf-frequency is included as an additional column in the response matrix – much like including another horizontal steering magnet.

Fitting the response to steering magnets alone usually gives a model with the correct tunes and beta functions, but does not always reproduce the dispersion as accurately. This can happen because the beta functions do not vary much with dipole magnet errors around the ring, while the dispersion does. Or it can happen when LOCO is used to find the best correction for some gradient error as described in section 2.2. The non-local gradient corrections can correct beta function distortion associated with the gradient error, but not necessarily the dispersion distortion. Including dispersion explicitly as a column of the response matrix forces LOCO to generate a model that accurately reflects both the beta functions and the dispersion of the real storage ring.

The dispersion fitting option can be useful for controlling dispersion to achieve low emittance.

3.4 BPM Calibration

BPM coupling and gain correction is treated somewhat differently in MATLAB LOCO than FORTRAN LOCO. In the FORTRAN version of LOCO, BPM corrections were applied to the measured orbit response:

$$\begin{pmatrix} x_{meas} \\ y_{meas} \end{pmatrix} = \frac{1}{\sqrt{1-C^2}} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & C \\ C & 1 \end{pmatrix} \begin{pmatrix} g_x x_{meas0} \\ g_y y_{meas0} \end{pmatrix}. \quad (3)$$

In the MATLAB version, BPM corrections are applied to the model response matrix:

$$\begin{pmatrix} x_{\text{mod}} \\ y_{\text{mod}} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_{\text{mod,0}} \\ y_{\text{mod,0}} \end{pmatrix}. \qquad (4)$$

In the MATLAB GUI, a and d are referred to as horizontal and vertical BPM gains, which are not exactly the same as the FORTRAN gain correction factors, g_x and g_y.

Putting the BPM correction factors on the model rather than the measured response matrix gives two benefits:

1.) BPM gains make more intuitive sense in MATLAB LOCO. A BPM gain of 1.10 in the MATLAB LOCO means the BPM gain is 10% high, while g_x=1.10 in the FORTRAN LOCO meant the BPM gain was 10% low.

2.) MATLAB LOCO has the capability to do a complete coupling analysis of storage rings with single-view BPMs (BPMs that measure horizontal or vertical orbit, but not both). With single-view BPMs, FORTRAN LOCO could only do a partial coupling analysis – BPM coupling could not be included in the fit. Equation 3 cannot be applied with single-view BPMs, because both x_{meas} and y_{meas} are required.

3.5 Error Bar Propagation

MATLAB LOCO includes an analytical calculation of the error bars on the fit parameters associated with BPM random measurement noise. This calculation gives a lower bound on the error bars, because is does not include systematic errors. The calculated error bars, however, are still useful as a guide to the relative accuracy with which the quadrupole gradients can be fit.

Fit parameter error bars can also be determined by analyzing multiple response matrices and calculating the rms variation in the fit parameters over the different fit models.

3.6 Outlier Data Rejection

MATLAB LOCO also has the option of outlier data rejection. The user can input a data rejection threshold in terms of number of standard deviations. Before each iteration, the rms difference between the model and measured response matrices is calculated. Data in the measured response matrix is rejected if it differs from the model by more than this threshold.

3.7 Numerical Derivative Calculation

LOCO requires the numerical derivatives of the model response matrix with respect to each of the quadrupole gradients as input to the χ^2 minimization routine. In the FORTRAN code the gradient step sizes for calculating the derivatives were required user input. The user had to determine the appropriate step sizes by trial and error. In the MATLAB code this requirement has been eliminated. The code will automatically determine the appropriate step sizes or accept a user input.

3.8 Memory Requirements and Speed

The memory requirement is usually driven by the size the response matrix and the number of parameters to be estimated. MATLAB uses 8-byte precision for matrices, so the numerical derivative of the model response matrices requires 8 * # of parameters * #BPMs * #steerers bytes of memory for the fully coupled case. A rough estimate of the memory required for the LOCO subroutine is 2.5 times this number.

The MATLAB LOCO code converges on the order of three times faster than the FORTRAN code. The time required for a particular storage ring depends strongly on the size of the response matrix and number of fit parameters. To give an example, it takes about 15 minutes to converge to a solution for the Advanced Light Source without coupling, when running on a cpu with speed equivalent to about 2 GHz.

4 CODE AVAILABILITY

MATLAB LOCO will be available on the web soon. For details contact safranek@slac.stanford.edu.

ACKNOWLEDGEMENTS

REFERENCES

[1] J. Safranek, 'Experimental determination of storage ring optics using orbit response measurements', Nucl. Inst. And Meth. A388, 27 (1997).

[2] http://www-ssrl.slac.stanford.edu/at/

[3] D. Robin, J. Safranek, G. Portmann and H. Nishimura, "Model calibration and symmetry restoration of the Advanced Light Source," EPAC'96.

[4] H.P. Chang, C.C. Kuo, and J. Safranek, "Correction of lattice optics in the presence of strong wiggler magnet at SRRC," APAC'98.

[5] F. Wang et al., "Linear optics calibration at the MIT-Bates south hall ring," PAC'99.

[6] H. Grote and F.C. Iselin, "The MAD program, version 8..1," CERN/SL/90-13, June 17, 1991.