’INTELLIGENT’ AUTOMATIC BEAM STEERING AND SHAPING

A. Jansson[*][†] and M. Lindroos, CERN, Geneva, Switzerland

Abstract

The strategy for Automated Beam Steering and Shaping (ABS) in the PS complex is to use theoretical response matrices calculated from an optics database. The main reason for this is that it enforces a certain understanding of the machine optics. A drawback is that the validation of such a matrix can be a lengthy process. However, every time a correction is made using an ABS program, a partial measurement of the response matrix is effectively performed. Since the ABS programs are very frequently used, the full matrices could thus be measured on an almost daily basis, provided this information is retained. The information can be used in two ways. Either the program passively logs the data to be analysed off-line, or the information is directly fed back to the matrix, which makes the program ’learn’ as it executes. The data logging provides a powerful machine debugging tool, since deviations between the measured and theoretical matrices can be traced back to incorrect optical parameters. The ’learning’ mode ensures that the correction will always converge. Implementation details and simulation results are discussed.

1 INTRODUCTION

Due to the complexity of large accelerators, computer support is needed for efficient operation. In the PS complex, software is developed for this purpose within the context of Automated Beam Steering and Shaping[1] (ABS).

The approach taken is that every relation between machine parameters and measured quantities can, if sufficiently small corrections are considered, be treated within the framework of linear perturbation theory. The first order relation between changes in the parameters c_i and changes in the measured quantities m_i are given by the response matrix A (also called the correction matrix), defined as

$$A \cdot c = m. \tag{1}$$

This response matrix is in general a function of the parameters. If this matrix is known, optimal corrections can be calculated from the errors in the measured quantities. Rather than inverting the matrix, which can be numerically unstable, a special correction algorithm called Micado[2] is used to calculate c from the measured errors m. This algorithm solves the matrix equation approximately, using only a few of the most effective correctors available.

The matrix can be obtained in two ways: either by direct measurement, or as a linearization of a theoretical machine model. In the PS complex, the second method has been chosen, since the creation and maintenance of an accurate theoretical model enforces a thorough understanding of the machines. However, the way to verify that the theoretical model correctly describes the machine is usually to measure the response matrix and compare it to the theoretical one. This can be a long and cumbersome process.

2 LEARNING CORRECTION SYSTEMS

2.1 ’Passive learning’ - Data Collection

When the ABS programs are used, measurements and subsequent corrections are iterated until the optimum is found. In the process, a lot of measurement data passes through the program. Presently, this data is lost once it has been used to calculate a correction. This is a waste of valuable information. If the information was retained, enough measured data could be collected in a typical ABS program session to reconstruct the entire response matrix. This can be implemented by a simple log file. Since the program collects information about the machine, but does not use it, this could be called a ’passive learning’ mode.

2.2 ’Active learning’ - Direct Feedback

In the current implementation, the ABS programs are static. Given a certain error, they will always propose the same correction. This is not a problem as long the iterative correction converge, but this may not always be the case, for example in closed orbit corrections if the tune of the machine is different from the model tune.

The obvious next step from the passive data collection is to enable the program to analyse the data online and modify the response matrix accordingly, if it should be found to be incorrect.

A program in ’active learning’ mode should identify and retain useful information and use it to reconstruct as much as possible of the response matrix. The reconstructed parts should then be substituted into the current response matrix. A calculated matrix could be used in the beginning of an ABS program session, but it would be gradually transformed into an up-to-date measured matrix. At the end of the session, the two matrices could be compared in order to identify differences between the model and the real machine. Thus, the learning approach is not contrary to the underlying idea of using theoretical matrices, but rather provides a simple and powerful tool to verify their correctness.

[*] CERN and Manne Siegbahn Laboratory, Stockholm, Sweden

[†] Email: Andreas.Jansson@cern.ch

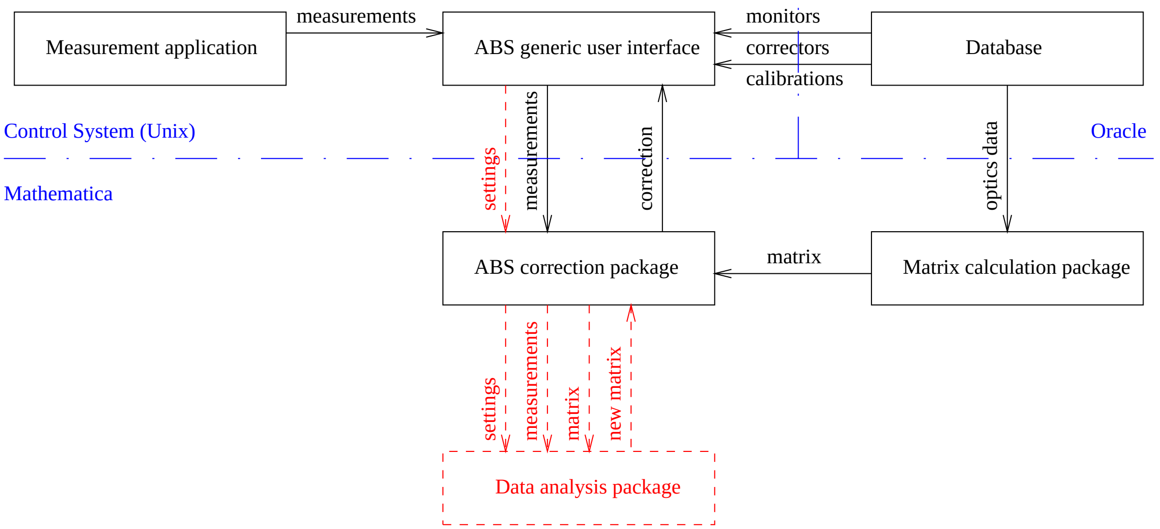

Figure 1: Schematic structure of the ABS application programs. The dashed line show where modifications are needed. Since the modifications are almost fully localised to the Mathematica part, they could be implemented easily.

3 PRACTICAL IMPLEMENTATION

The ABS programs have several components[3] (see Fig. 1). The application program is configured from a database[4], that also contains the theoretical model of the machine. The actual computation of corrections, based on the measured data and the database model, is made in a Mathematica kernel running as a sub-process to the application program. This modular structure makes it very easy to implement the ideas presented above, since in principle only the Mathematica code needs to be modified. Data logging can be enabled simply by dumping all measured data to a flat file for further analysis.

The code for the analysis of the logged data have been written in Mathematica and could therefore be incorporated easily in the correction module in order to make it analyse the data on-line. The data from several corrections can be summarised as

$$A \cdot C = M \tag{2}$$

where each column of the matrices C and M corresponds to one applied correction. The crucial difference from (1) is that now the matrix A is the unknown. While the equation is often under-determined because certain correctors have not been changed, there is often a sub-system which is solvable.

The analysis module performs the analysis of this data in several steps. It first suppresses data that are too noisy. That is, corrections where the changes made are either zero or too small to give an effect above the noise level. This is important for the stability of the solution.

In the second stage it, analyses the remaining data to see which parts of the response matrix that can be reconstructed. This is done by studying the properties of the matrix C. First, the matrix is reduced by removing linearly dependent columns. Then, by permutating rows and columns, the program identifies the largest possible square sub-matrix C_1, such that

$$\tilde{C} = \left(\begin{array}{c|c} C_1 & C_2 \\ 0 & C_3 \end{array} \right) . \tag{3}$$

The corresponding permutated matrix equation

$$\left(\begin{array}{c|c} A_1 & A_2 \end{array} \right) \cdot \left(\begin{array}{c|c} C_1 & C_2 \\ 0 & C_3 \end{array} \right) = \left(\begin{array}{c|c} M_1 & M_2 \end{array} \right) \tag{4}$$

can then be solved for

$$A_1 \cdot C_1 = M_1 \tag{5}$$

since the columns of the reduced matrix are linearly independent and thus the inverse exists.

Having identified the solvable correctors, the program goes back to the full data set, and selects all measurements where only these correctors have been used. The resulting matrix equation is similar to (5), but in general overdetermined, and can be solved using a least squares fit.

Finally, the calculated columns are substituted into the previous response matrix.

4 TESTS

The performance of the learning algorithm, as compared to the standard ABS algorithm based on a static matrix, has been evaluated using simulations. In each simulation, initial errors in both the matrix and the corrector settings were randomly generated, and the performance of the two algorithms studied as a function of the number of iterative corrections made. Measurement noise was also included. Two examples of such simulations are shown in Figs. 2 and 3. It

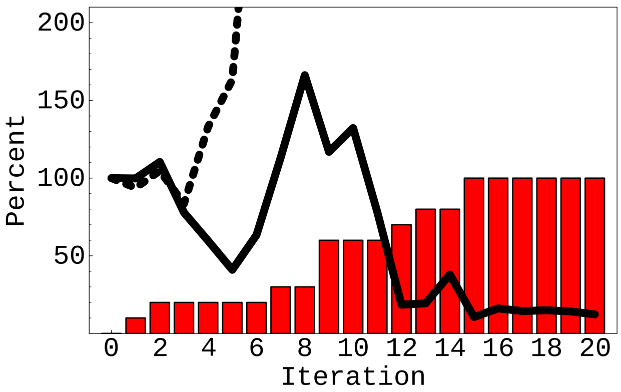

Figure 2: Simulation results for a system of 10 correctors and 10 monitors. The solid line shows the residual error in percent as a function of number of corrections performed, when using the 'learning' algorithm. The dashed line shows the same curve using a static matrix. The bar chart indicates how many percent of the matrix entries that have been updated. In this case the initial matrix had large errors. After an initial excursion, the learning algorithm converges, whereas the static matrix version does not.

was found that the 'learning' algorithm always converged eventually, whereas with a static matrix it often diverged.

It was noted, however, that if the initial matrix error is very large, the residual error can initially grow significantly before the learning algorithm starts to converge. In a real machine this could lead to beam loss, and is thus not acceptable. This problem, appears mainly when corrections are made using many correctors. In such a situation, the operator would cancel the correction and make a new attempt with fewer correctors.

Another experience from the simulations is that the average number of iterative corrections needed to reconstruct the full matrix is not much greater than the theoretical minimum (which is given by the number of columns in the matrix). Since the corrections converge very rapidly once the correct matrix is obtained, this gives an estimate of the maximum number of corrections needed for convergence.

5 EXTRACTING THE OPTICS INFORMATION

Just as the response matrix can be calculated from optical parameters, the optical parameters can under certain circumstances be deduced from the matrix. For example, the matrix elements for beam steering in a transfer line are given by

$$\Delta x_i = \begin{cases} \sqrt{\beta_i\beta_j}\,\sin(\mu_i - \mu_j) & \text{if } \mu_i > \mu_j, \\ 0 & \text{if } \mu_i < \mu_j. \end{cases} \qquad (6)$$

For such a response matrix of n correctors and m monitors, the matrix has $n \cdot m$ entries, where roughly half are zero in case monitors and correctors are evenly distributed along the line. The matrix can be parametrised in terms of the value of the beta function at $n + m$ locations, as well as $n + m - 1$ positions in betatron phase (the phase of the first element can be arbitrarily chosen).

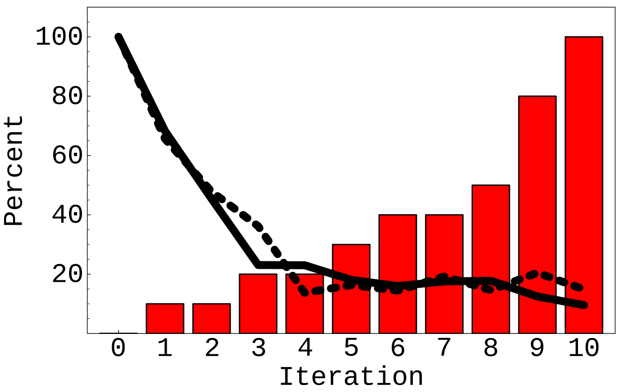

Figure 3: Simulation performed and presented as in Fig. 2. In this case, the initial matrix errors were small. Both algorithms converges, but in the process the learning algorithm measures the matrix, which can then be compared to the calculated one.

Since the two numbers scale very differently, when the matrices are sufficiently large, the resulting system of equations is over-determined and thus the optical parameters can be determined.

Therefore, if the measured matrices resulting from each ABS session are logged, effectively a history of the machine optics is obtained. This could be useful for example to study the long term stability of the machine.

6 CONCLUSIONS

By making ABS programs learn from previous corrections, a faster and more secure convergence can be obtained.

An implementation of such a learning system have been made in Mathematica, as an extension to the current ABS Mathematica package. and tested by simulations with very good results.

By logging the information collected by the program, statistics on the machine optics, and its stability in time, is automatically obtained.

REFERENCES

[1] B. Autin et al, "Emittance Preservation in the PS Complex", PAC'97, Vancouver, 1997.

[2] B. Autin and Y. Marti, "Closed Orbit Correction Using a Small Number of Magnets", CERN internal note, ISR-MA/73-17, 1973.

[3] M. Arruat et al, "Generic Automated Beam Steering and Shaping Programs with an Object Oriented Approach", ICALEPCS'97, Beijing, 1997.

[4] B. Autin et al, "Database for Accelerator Optics", ICALEPCS'97, Beijing, 1997.