THE NOVEL IMPLEMENTATION OF THE ORBIT CORRECTION ALGORITHM FOR SOLARIS STORAGE RING

P. Sagalo[*], L. J. Dudek, A. Kisiel, G. W. Kowalski, A. I. Wawrzyniak,
Solaris National Synchrotron Radiation Centre, Krakow, Poland
P. P. Goryl, 3controls, Krakow, Poland

Abstract

The storage ring which is located in the National Synchrotron Radiation Center SOLARIS works under the TANGO control system. So far the correction of an electron beam orbit has been performed with an algorithm implemented in the Matlab Middle Layer (MML). To ensure consistency of the correction process with the entire control system, a new implementation of this algorithm has been developed. The algorithm of orbit correction based on SVD has been implemented as a TANGO Device, which is one of the fundamental blocks used in the Tango control system. The entire code has been written in the Python.

GENERAL DESCRIPTION OF THE SYNCHROTRON

The Solaris storage ring has circumference equal to 96 m. The maximum current which can be accumulated is 500 mA at the maximum energy of 1.5 GeV. The accelerator consists of a lattice of twelve DBAs (double-bend achromat), connected with 3.5 m straight sections. Ten of them are reserved for insertion devices and the other two are intended for beam injection and RF systems. The main Solaris storage ring parametes are presented in Table 1.

Table 1: The Main Storage Ring Parameters [1]

Energy	1.5 GeV
Nominal current	500 mA
Circumference	96 m
RF frequency	99.931 MHz
Natural emittance	5.598 nmrad
Energy spread	$0.745 \cdot 10^{-3}$
Radiation losses/turn	114.1 keV
Betatron tunes (H/V)	11.22/3.15
Corrected chromaticities (H/V)	+1/+1
Momentum compaction factor	$3.055 \cdot 10^{-3}$
Total lifetime	13 h

BEAM POSITION MONITORS AND CORRECTOR MAGNETS

Beam Position Monitors

Each of DBAs consist of three beam position monitors (BPM), whose active elements are buttons. Two of them are diagonal type BPMs and they are mounted at the end of

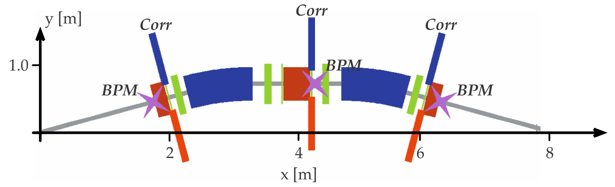

Figure 1: Positions of BPMs and correctors. The magenta crosses indicate locations of BPMs. The blue and red strips indicate the position of dipole correctors [2].

a DBA. In the center of a DBA, BPM is placed, which has buttons set horizontally (see Fig. 1). Matlab code for buttons simulation has been done in [3], using Matlab boundary element solving routines [4].

Signals from BPMs are registered by Libera Briliance+ modules, which are an expanded analogue-digital converter with FPGA-based digital signal processing system supporting the Tango interface.

Corrector Magnets

Corrector sextupole magnets (SCi, SCo) include extra windings (see Figs. 2 and 3), each of this windings is powered independently so that, this magnet can work as horizontal/vertical correction dipol. Thanks to that correction magnets do not take extra space.

The corrector strength is changed by changing the current on the power supply. Therefore the operator is setting the current which corresponds to certain corrector strength.

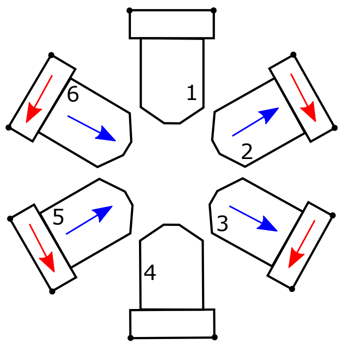

Figure 2: Implementation of the vertical corrector [2].

The correctors and BPMs are located close to each other in order to minimize the phase advance in between.

[*] przemyslaw.sagalo@uj.edu.pl

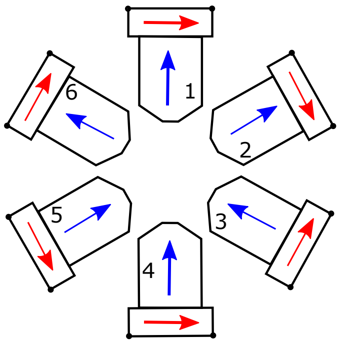

Figure 3: Implementation of the horizontal corrector [2].

THE IDEA OF BEAM CORRECTION BASED ON SVD

The principle mathematical "tool" used in orbit correction is a corrector-to-BPM response matrix. This matrix determines dependence of beam position change according to angular kick change of corrector magnets.

$$\Delta\vec{x} = R\Delta\vec{\theta}, \tag{1}$$

where: R - response matrix, $\Delta\vec{x}$ - vector of changing beam position in specific measurement points, $\Delta\vec{\theta}$ - vector of changing specific correctors strength.

Columns of a response matrix are created through measure a difference between a beam position before and after current impulse given to the corresponding corrector[1]. Then this column is divided by the value of this current impulse.

Transforming the Eq. (1) to form

$$\Delta\vec{\theta} = R^{-1}\Delta\vec{x}, \tag{2}$$

we get the equation by which we can determine $\Delta\vec{\theta}$, ie values that correctors strengths have to be change to make the beam orbit change of $\Delta\vec{x}$.

A main problem in the direct use of this equation is in the "nature" of a response matrix. Very often a number of correctors is not equal to the number of BPMs, therefore this matrix is not square or can be a non-invertible matrix.

However, the SVD (singular value decomposition) theorem is helpful. For each matrix $A \in \mathbb{R}^{M\times N}$, $M \geqslant N$, factorization exist

$$A = U(diag(w_i))V^T, \tag{3}$$

where: $U \in \mathbb{R}^{M\times N}$ - columnar orthogonal, $V \in \mathbb{R}^{N\times N}$ - orthogonal and $w_i \in \mathbb{R} \ : \ i = 1,\ldots,N$. If matrix A is non-invertible matrix ($det(A) = 0$) then equation $A\vec{x} = \vec{b}$ has a solution, if $\vec{b} \in Range(A)$. This solution takes the following form:

$$\vec{x} = \tilde{A}^{-1}b + \vec{x_0} \tag{4}$$

where:

$$\tilde{A}^{-1} = V\left(diag(\tilde{w}_i^{-1})\right)U^T, \tag{5a}$$

[1] The column number corresponds to the number of the corrector.

$$\vec{x_0} \in Ker(A), \tag{5b}$$

$$\tilde{w}_i^{-1} = \begin{cases} w_i^{-1} & \text{gdy } w_i \neq 0, \\ 0 & \text{gdy } w_i = 0 \end{cases} \tag{5c}$$

The solution with $\vec{x_0} = 0$ has the smallest form.

Thanks to this theorem we can, in a relatively easy way, find $\Delta\vec{\theta}$ in case, when a response matrix R is non-invertible.

The orbital correction process starts with the measurement of the orbit that exists at a given moment. It is also necessary to determine which orbit we want to achieve (the so-called "golden orbit"). Based on this data we can determine vector $\Delta\vec{x}$. By inserting this vector into the equation (2), we get a vector which components are determined by values of correctors strength change [5, 6].

A PROGRAMMATIC DESCRIPTION OF THE NEW SYSTEM

The whole implementation of the orbit correction algorithm has been done in Python as a TANGO class and fits into the idea of the Tango control system. By means of this class is defining Tango Device instance, which is directly responsible for doing correction. We can divide the whole algorithm into two blocks.

The first of them is responsible for determining a response matrix. An operator defining the value of the corrector strength and numbers of repetition for a positive and a negative impulse. For each repetition a response matrix is determined. Next, these matrixes are averaged to one final matrix. An inverse response matrix is determining by means of the SVD algorithm, which is provided by Numpy package as the linalg.pinv function. This function performs the Moore-Penrose pseudo-inverse with automatic consideration of the number of singular values specified by the operator. It should be noted that the determining of the inverse matrix is not strictly assigned to the process of determining the response matrix and can also be performed during the operation.

The second block is directly responsible for the correction execution. The main parameters set by the operator are: the percentage by which the correction has to be performed, the number of singular values and the list of correctors to be excluded. The operator also defines the number of iteration in the loop, which will be interrupted in the event of an incident.

SIMPLE TEST MEASUREMENTS

The example response matrix (see Figs. 4 and 5) has been determined for the current pulse, which value was 0.75 A. This pulse was both positive and negative.

RMS values for each BPM and for new and old implementation were determined. The results of these measurements are shown in the Fig. 6, 7.

As one can see most of the measuring points in the new implementation are below the value of 0.1 mm.

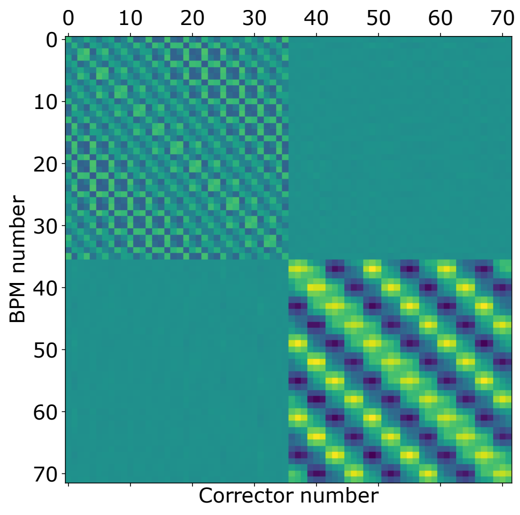

Figure 4: Response matrix.

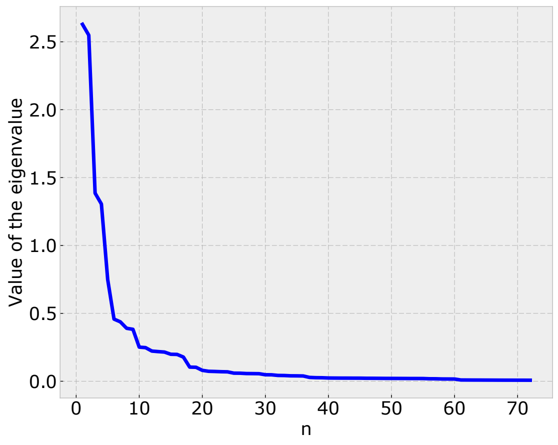

Figure 5: Eigenvalues of the response matrix.

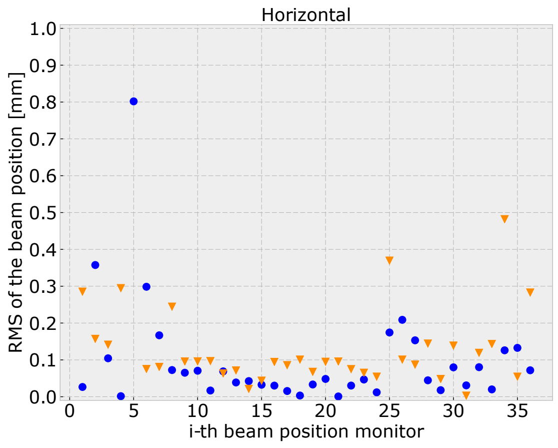

Figure 6: RMS for horizontal BPMs; blue — Tango implementation, orange — MML implementation.

CONCLUSION

For the synchrotron working in the Solaris National Synchrotron Radiation Center the new implementation of an

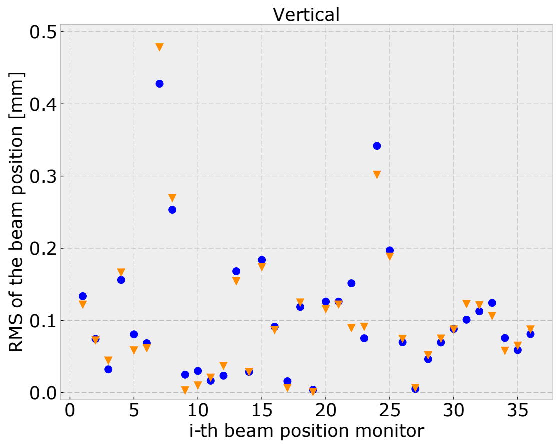

Figure 7: RMS for vertical BPMs; blue — Tango implementation, orange — MML implementation.

orbit correction algorithm has been done. This novel algorithm is based on SVD and its purpose is to achieve full integrity with the TANGO control system. Therefore the whole program code has been written in Python as a Tango Class, thus totally being adjusted to the use of the TANGO control system, giving the possibility of replacing the currently working system implemented in MML. Most of the measurement points of beam position RMS in particular BPMs for the new implementation are located near the measurement points obtained by using similar code in MML. The new system needs some additional tests but the recently obtained results show of its potential.

REFERENCES

[1] A. I. Wawrzyniak *et al.*, "Solaris storage ring commissioning," in *Proc. 7th Int. Particle Accelerator Conf. (IPAC'16)*, Busan, Korea, May 2016, paper WEPOW029, pp. 2895–2897.

[2] The MAX IV Detailed Design Report, 3.4. Lattice Errors and Correction, `https://www.maxiv.lu.se/wp-content/plugins/alfresco-plugin/ajax/downloadFile.php?object_id=a919525a-74b7-4118-b7a3-f46860d256af`

[3] A. Kisiel *et al.*, "Performance of the beam position monitor system in Solaris synchrotron," in *Proc. 7th Int. Particle Accelerator Conf. (IPAC'16)*, Busan, Korea, May 2016, paper MOPMW017, pp. 432–434.

[4] A. Olmos, F. Perez, G. Rehm, "Matlab code for BPM button geometry computation," in *Proc. of DIPAC 2007*, Venice, Italy, 2007, paper TUPC19, p.186.

[5] USPAS January 2005, `http://http://uspas.fnal.gov/materials/05UCB/UCB-Resp-Matrix-Measure.shtml`

[6] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. New York, NY, USA: Cambridge University Press, 2007.