# ORBIT CORRECTION WITH FREQUENCY IN JAVA

Anže Zupanc*, Igor Križnar, Mark Pleško, Urban Simončič
Jožef Stefan Institute, Ljubljana, Slovenia

*Abstract*

In this article we represent an integration of a new orbit correction algorithm into the OrbitCorrection Java application [1]. A new closed orbit correction method for the ANKA storage ring optimizes both correctors strength and RF frequency. The OrbitCorrection application has grown from a simple application preforming closed orbit correction to a platform for different orbit manipulation tools. It is daily used at the synchrotron light source ANKA in Karlsruhe [2]. So far several orbit manipulation routines, such as closed orbit correction and local bumps, have been implemented and included into the OrbitCorrection application. Modular construction of the OrbitCorrection application offers physicists an opportunity to easily write their own orbit manipulation methods and focus on algorithms, with a minimum concern about control system and general application details.

## 1 INTRODUCTION

The closed orbit correction method with RF frequency was developed for ANKA synchrotron light source. The method uses beside correctors also the frequency of the storage rings RF cavity as another parameter which can be adjusted to minimize the orbit in horizontal plane. This method was recently developed and has been successfully tested only under the simulation mode while machine tests are yet to be done at ANKA in a short time.

## 2 CORRECTION METHOD

The displacement of the orbit is connected with dispersion function by relation [3]

$$x_g(s) = x(s) + x_D(s) = x(s) + D(s)\frac{\delta p}{p} \qquad (1)$$

which describes the orbit displacement at point $s$ along the storage ring. First term $x(s)$ represents the orbit displacement due to the correctors and the second term $x_D(s)$ represents the orbit displacement due to an electron energy deviation from the ideal energy defined by the electon path. It is related to the dispersion function $D(s)$ at point $s$ and the energy deviation from the ideal value $\frac{\delta p}{p}$. Our goal is to alter the orbit which can be done by changing the correctors strength and the RF frequency. Our task is now to find out such corrector strength and RF frequency that will bring the orbit as closely to zero as possible.

First we use the identity [4]

$$\frac{\delta p}{p} = \frac{1}{\eta_c}\frac{\delta f}{f} \qquad (2)$$

which determines the relationship between the energy deviation $\frac{\delta p}{p}$ and the frequency deviation $\frac{\delta f}{f}$. This quotient represents the difference between the RF frequency, which is actually set to the machine, and the ideal RF frequency and is devided by the ideal RF frequency.

Applying Eq. 2 to Eq. 1 we get

$$x_g(s) = x(s) + x_D(s) = x(s) + D(s)\frac{1}{\eta_c}\frac{\delta f}{f} \qquad (3)$$

where $\eta_c$ is momentum compaction and is given by the combination of the total particle energy in units of the particle rest energy $\gamma$ and momentum compaction factor which describes variation of the path length with momentum

$$\eta_c = \gamma^{-2} - \alpha_c \text{ and } \alpha_c = \frac{\Delta L / L_0}{\delta p / p_0}.$$

Displacement of the orbit can be measured by beam position monitors (BPMs) which are distributed around the storage ring. The beam position can be represented by a M-dimensional vector $\vec{x}$, where the components of this vector are values of the orbit displacement at BPMs. The correctors strength (kicks) can be represented by a N-imensional vector $\vec{\theta}$. M is the number of BPMs and N the number of correctors. The first term in Eq.3 can now be written as

$$A\vec{\theta} = \vec{x}. \qquad (4)$$

Linear response matrix A ( $N \times M$ dimensional matrix) describes the relation between corrector kicks and beam position changes at BPMs. The element $A_{ij}$ of the response matrix corresponds to the orbit shift at the $i$ th monitor due to a unit kick from the $j$ th corrector and it is equal

$$A_{ij} = \frac{\sqrt{\beta_i \beta_j}}{2\sin \pi\upsilon}\cos(\left|\Phi_i - \Phi_j\right| - \pi\upsilon).$$

The second term can be also represented by an N-dimensional vector which $i$ th component is

$$D(s_i)\frac{1}{\eta_c}\frac{\delta f}{f} \qquad (5)$$

*anze.zupanc@cosylab.com

where $D(s_i)$ is the value of the dispersion function at $i$ th BPM. With Eq. 4 and Eq.5 eqution 3 may now be written in matrix form as

$$\begin{bmatrix} A_{11} & A_{12} & . & . & . & A_{1N} & \frac{1}{\eta_c}D(s_1) \\ A_{21} & & & & & . & \frac{1}{\eta_c}D(s_2) \\ . & & & & & . & . \\ . & & & & & . & . \\ . & & & & & . & . \\ . & & & & & . & . \\ . & & & & & . & . \\ A_{M1} & A_{M2} & . & . & . & A_{MN} & \frac{1}{\eta_c}D(s_M) \end{bmatrix} \begin{bmatrix} \vartheta_1 \\ \vartheta_2 \\ . \\ . \\ . \\ \vartheta_N \\ \frac{\delta f}{f} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_M \end{bmatrix}$$

To solve this linear system we invert this combined response matrix with SVD [5] method, which is quite robust and it has been successfully used with default closed orbit correction. The result is then applied to the machine.

## 3 ORBIT CORRECTION APPLICATION

The modular architecture of the application allows the programme developers an easy maintaince and the development of new orbit manipulation methods in the application. The OrbitCorrectionModels in form of plug-ins use DataBush [6] for calculating the correction with semi-real-time control system properties. The use of the DataBush library gives programmers an intuitive access to devices and elements which are relevant to machine physics, while the access to the control system is provided by Abeans [7].
The OrbitCorrectionModel must implement the interface OrbitCorrectionModel and provide methods for :

- Receiving by user selected correctors, BPMs and other control system channels
- Calculating corrections
- Returning new values for these parameters

The module can also provide simple GUI panel, where a user can directly customize the module. The Orbit Correction application takes care of the rest:

- the selection of correctors and BPMs
- scaling the results from the model
- sending the results of the model with fast stepwise sets to the control system
- handling possible errors

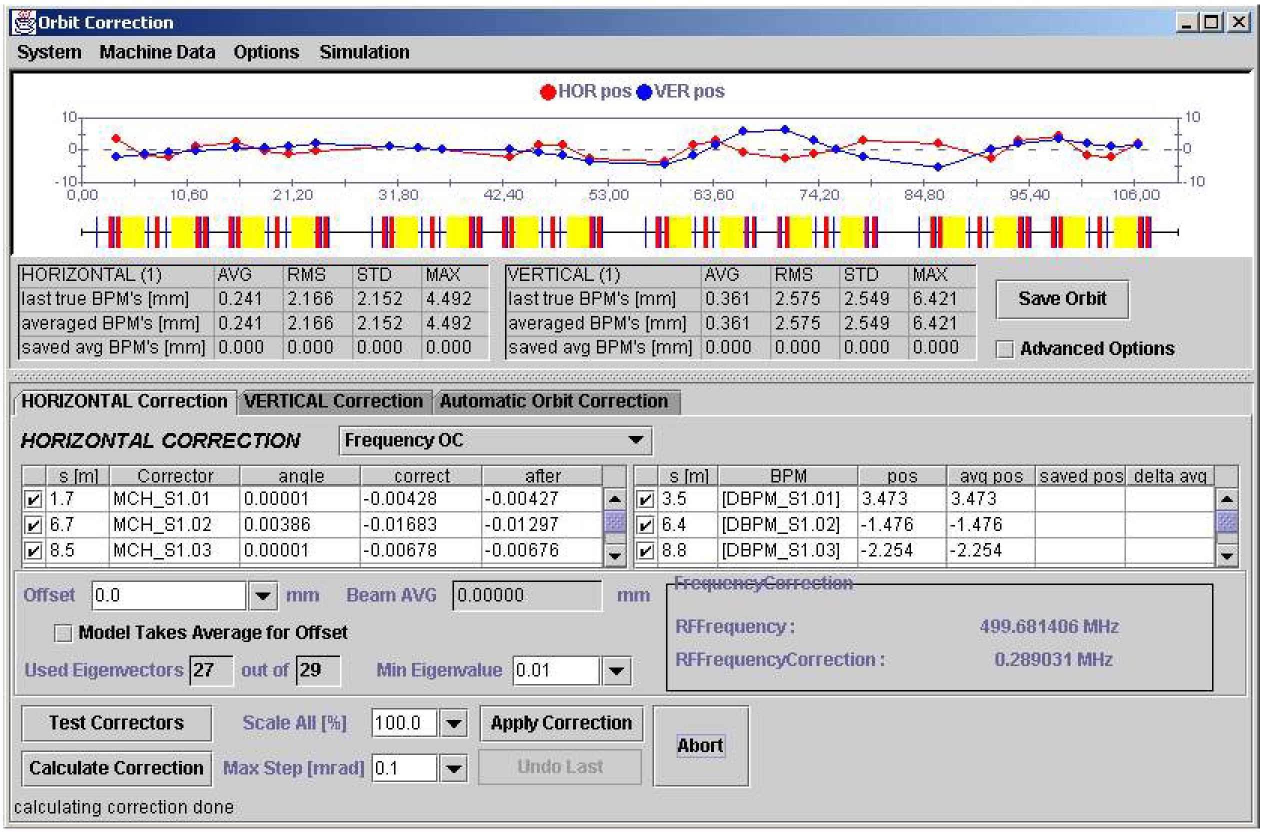


Figure 1: OrbitCorrection application

The Orbit Correction application was developed for ANKA storage ring with the intention of providing tools for the orbit manipulation.

- handling control system problems.

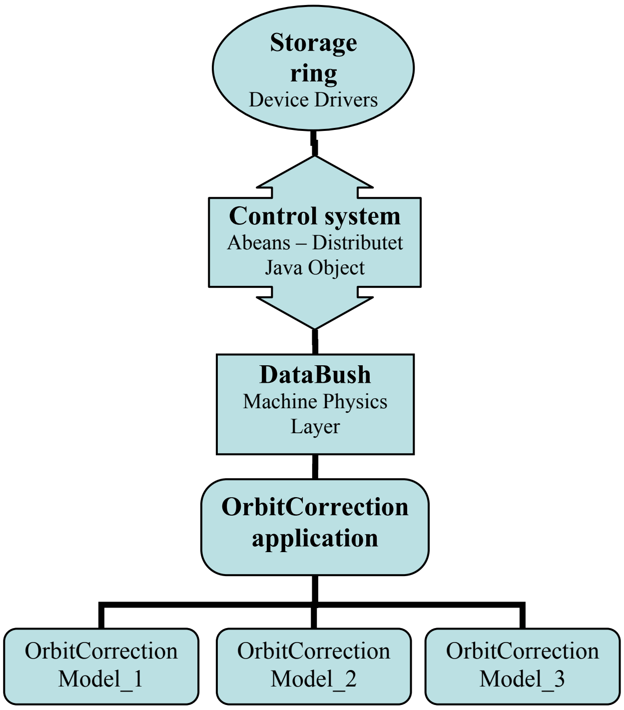


Figure 1: To implement new orbit correction method in Orbit Correction application new OrbitCorrectionModel must be implemented and simply connected to application itself.

So far the Orbit Correction application has the following orbit correction and manipulation methods, which are implemented in the form of the OrbitCorrectionModels :

- **Default Orbit Correction** – the closed orbit correction method with the theoretical response matrix based on the SVD method
- **Empiric Orbit Correction** – the closed orbit correction with the measured response matrix based on the SVD method
- **Orbit Correction with frequency** – closed orbit correction with the theoretical response matrix with the RF frequency as new parameter, based on the SVD method
- **Correction Strength Correction -** reduction of correctors strength with the theoretical response matrix, based on the SVD method
- **Local Bump -** makes local bump with three or four correctors
- **Response Matrix Bump** - makes local bump with the theoretical or user specified response matrix, based on the SVD method

## 4 TESTS

The orbit correction method with frequency has only been tested under the simulation. In order to test this method a special physical model was developed, which calculates the form of the orbit. Test results indicate that orbit correction method with frequency is a bit more efficient than the default orbit correction method.

## 4 ACKNOWLEDGEMENT

We would like to thank ANKA for their support and for placing the initial contract to us as well as for their hospitality and for the opportunity to develop the OrbitCorrection application, from which we gained a lot of expirences for the future.

## 5 REFERENCES

[1] I. Križnar, *Databush Clients user manual* http://kgb.ijs.si/KGB/Docs/Part%20I.%20User%27s%20 Manual/3.6.%20Databush%20Clients/Databush_Clients.htm#_Toc524271373
[2] http://www.fzk.de/anka/
[3] K. Wille, *Physik der Teilchenbescleuniger und Synchrotronstrahlungsquellen* (Teubner Studienbücher Physik, 1992)
[4] H. Wiedemann, *Particle Accelerator Physics – Basic Principles and Linear Beam Dynamics* (Springer-Verlag, 1993)
[5] A. Wu Chao, M. Tigner, *Handbook of Accelerator Physics and Engineering* (World Scientific, 1999)
[6] I. Križnar, *User manual* http://kgb.ijs.si/databush I.Križnar, *DataBush: Machine Physics Programming Package in Java* (http://kgb.ijs.si/KGB/articles)
[7]G. Tkacik *Java Beans of Accelerator Devices for Rapid Application Development* http://kgb.ijs.si/KGB/articles/