MACHINE LEARNING APPLICATIONS FOR ORBIT AND OPTICS CORRECTION AT THE ALTERNATING GRADIENT SYNCHROTRON*

W. Lin[1]†, G.H. Hoffstaetter[1,3], D. Sagan[1]
B. Huang[2], W. Dai[2], T. Robertazzi[2], V. Schoefer[3], K.A. Brown[2,3]
[1]CLASSE, Cornell University, Ithaca, New York, USA
[2]Stony Brook University, New York, USA
[3]Collider-Accelerator Department, Brookhaven National Laboratory, Upton, New York, USA

Abstract

The Alternating Gradient Synchrotron (AGS) is a particle accelerator at Brookhaven National Laboratory (BNL) that accelerates protons and heavy ions using the strong focusing principle. In this work, we perform simulation studies on the AGS ring of a machine error detection method by comparing simulated and measured orbit response matrices (ORMs). We also present preliminary results of building two neural network (NN) surrogate models of the AGS system. The first NN model is a surrogate model for the ORM, which describes mapping between orbit distortions and corrector settings. Building a self-adaptive model of ORM eliminates the need to re-measure ORM using the traditional time-consuming procedure. The second NN model is an error identification model, which maps the correlation between measurement errors (differences between measurement and model) and sources of such errors. The most relevant error sources for the error model are determined by performing sensitivity studies of the ORM.

INTRODUCTION

The Alternating Gradient Synchrotron (AGS) is a strong focusing synchrotron ring that currently serves as the injector for Relativistic Heavy Ion Collider (RHIC) at Brookhaven National Laboratory (BNL). It will continue to serve as the injector for the future Electron Ion Collider (EIC). The AGS accelerates polarized protons and heavy ions up to a typical maximum rigidity of 81 T-m (e.g., 23 GeV protons and 10 GeV/N Au beams) [1]. Orbit correction and brightness control at the AGS are important to the EIC project because electron cooling for the EIC requires small incoming emittances from the AGS. The current AGS is mostly hand tuned by operators, without systematic tuning routine. EIC also requires an extra cooler at AGS extraction energy, so better beam control in the AGS will benefit EIC cooler operation.

In this work, we present machine learning (ML) based orbit correction method by studying the orbit response matrix (ORM). The traditional orbit correction method used by most accelerator facilities calculates the inverse of a pre-measured ORM matrix $\underline{R}^{-1}$ using singular value decomposition (SVD), and solves the matrix equation to get required corrector settings [2]. The AGS ring is comprised of 12 super-periods, each containing 4 dual plane orbit correctors and 6 dual plane beam position monitors (BPMs). By establishing surrogate models for both the ORM itself and its change due to machine errors, we can achieve accurate real-time orbit tuning at the AGS.

ORBIT RESPONSE MATRIX

The orbit response matrix (ORM) of an accelerator maps the relationship between orbit measurements $(\vec{x}, \vec{y})$ and corrector settings $(\vec{\theta}_x, \vec{\theta}_y)$ via [3]:

$$\begin{pmatrix}\Delta\vec{x}\\ \Delta\vec{y}\end{pmatrix} = \underline{R}\begin{pmatrix}\Delta\vec{\theta}_x\\ \Delta\vec{\theta}_y\end{pmatrix} \tag{1}$$

The measured $\underline{R}^{meas}$ can be obtained by changing each corrector successively and observe the change in orbit measurements. If the $\underline{R}$ is an accurate representation of the machine, then we can easily correct any orbit offsets $\Delta\vec{y}$ with corrector settings $\Delta\vec{\theta} = \underline{R}^{-1}\Delta\vec{y}$.

Simulated data inquiry using Bmad and PyTao

The AGS ring has a total of 72 BPMs and 48 correctors, so $\underline{R}$ has a dimension of (72, 48). The optical model of AGS is traditionally built with MAD-X [4], but simulating a full ORM will require running multiple times of MAD-X simulation and then dealing with all the output files. In order to streamline this process, we built the AGS optical model using Bmad [5]. Bmad has a Python interface PyTao for its simulation program Tao, which allows accelerator simulations to run in a Python environment. Therefore, we developed a Python routine using for loops to automatically run simulations with different corrector settings, combine all the output orbits easily to an array, and calculate each element of $\underline{R}$ using $R_{ij} = \frac{\Delta y_i}{\Delta \theta_j}$.

Machine error detection with SVD

An actual machine with errors (i.e. quadrupole gradient errors, corrector calibration errors, etc.) will produce a different $\underline{R}^{meas}$ than that of a model/reference machine with no error $\underline{R}^{mod}$:

$$\Delta R_{ij} = R_{ij}^{meas} - R_{ij}^{mod} \tag{2}$$

The amount of the difference $\Delta\underline{R}$ is determined by the machine errors $\vec{v}$. Therefore, we can introduce another mapping matrix $\underline{J}$, which relates the change of orbit response matrix with machine error via:

* Work supported by Brookhaven Science Associates, LLC under Contract No. DE-SC0012704 with the U.S. Department of Energy and by the U.S. National Science Foundation under Award PHY-1549132.

† wl674@cornell.edu

$$\begin{pmatrix} \Delta R_{11} \\ \Delta R_{12} \\ \ldots \\ \Delta R_{n(m-1)} \\ \Delta R_{nm} \end{pmatrix} = \underline{J} \begin{pmatrix} \Delta \nu_1 \\ \Delta \nu_2 \\ \ldots \\ \Delta \nu_{N-1} \\ \Delta \nu_N \end{pmatrix} \tag{3}$$

Since $\underline{J}$ is not a square matrix, the solution to Eq. 3 can be found using singular value decomposition (SVD):

$$\underline{J} = \underline{U}\,\underline{S}\,\underline{V}^T \tag{4}$$

where $\underline{U}$ is an $nm \times nm$ orthogonal matrix, $\underline{V}$ is an $N \times N$ orthogonal square matrix, such that $UU^T = VV^T = 1$. $\underline{S}$ is an $nm \times N$ diagonal matrix whose elements are the singular values of $\underline{J}$.

The pseudo-inverse of $\underline{J}$ is then given by:

$$\underline{J}^+ = \underline{V}\,\underline{S}^+\,\underline{U}^T \tag{5}$$

Therefore, if we can establish a $\underline{J}$ by testing how $\Delta\underline{R}$ changes with some commonly known machine errors, any future error of the same sources can be reconstructed simply by measuring $\Delta\underline{R}$ and solving Eq. 6:

$$\begin{pmatrix} \Delta \nu_1 \\ \Delta \nu_2 \\ \ldots \\ \Delta \nu_{N-1} \\ \Delta \nu_N \end{pmatrix} = \underline{V}\,\underline{S}^{-1}\,\underline{U}^T \begin{pmatrix} \Delta R_{11} \\ \Delta R_{12} \\ \ldots \\ \Delta R_{n(m-1)} \\ \Delta R_{nm} \end{pmatrix} \tag{6}$$

We tested this error detection method on quadrupole errors in the AGS. There are 24 quadrupoles (12 horizontal, 12 vertical) in the AGS. All quadrupoles are wired in series in one plane, but in Bmad we can add kick error $\Delta\nu_k$ to individual quadrupole and observe how $\underline{R}^{meas}$ differs from $\underline{R}^{mod}$, therefore calculating $\underline{J}$ by $J_{ijk} = \frac{\Delta R_{ij}}{\Delta \nu_k}$.

Figure 1 shows the values of $\underline{J}$ calculated for the horizontal quadrupoles. As mentioned before for the AGS $\underline{R}$ has a dimension of (72,48), so the flattened $\Delta\underline{R}$ has a dimension of $(72 \times 48, 1) = (3456, 1)$, and $\underline{J}$ in this case has a dimension of (3456,12). A $\Delta\nu$ of 4 Amp in power supply current, which corresponds to ± 1% in k1 value, is added to each horizontal quadrupole to calculate $\underline{J}$.

Once we obtained $\underline{J}$, we are able to reconstruct any horizontal quadrupole setting by getting an ORM measurement and plugging the values into Eq. 6. Figure 2 presents an example of successful reconstruction of a randomly chosen current combination for all 12 quadrupoles.

Neural Network for real-time ORM

Despite the importance of ORM to orbit correction, in reality measured $\underline{R}$ is not updated often at most accelerators. It takes a time-consuming measuring process that takes at least 30 minutes and interrupts normal operations. Pre-measured $\underline{R}^{meas}$ will inevitably get less accurate with time due to changes in the machine elements. This usually results in an orbit shift or brightness drop even when the traditional orbit correction method is still running.

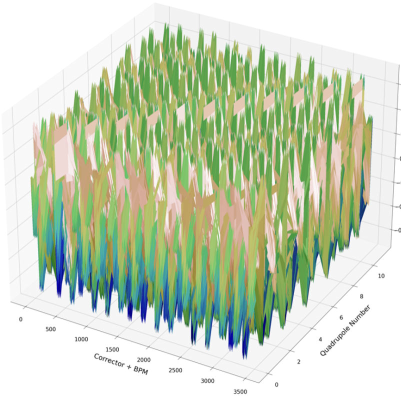

Figure 1: Simulated $\underline{J}$ matrix for horizontal quadrupoles.

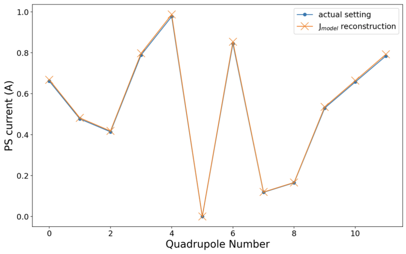

Figure 2: Horizontal quadrupole power supply current settings reconstructed from ORM measurement.

One way to solve this problem is by building a surrogate model for the real-time $\underline{R}^{meas}$ and retrain the model constantly with new orbit data. Most accelerator facilities has feedback systems set up so that the orbit data and the corrector settings are automatically saved after a predetermined time interval during daily operation. Such abundance of data is very fitting for machine learning applications. An example study is done at BEPCII as detailed in [6].

Here we present the training results of two neural network (NN) models: one for $\underline{R}$ (Fig. 3) and the other for $\underline{R}^{-1}$ (Fig. 4). Both are fully connected feed-forward neural networks (FFNN) with three layers and Hyperbolic Tangent activation function [7]. They are trained on 800 pairs of simulated orbit data, and both models managed to reach accuracy above 99% on 200 pairs of testing data.

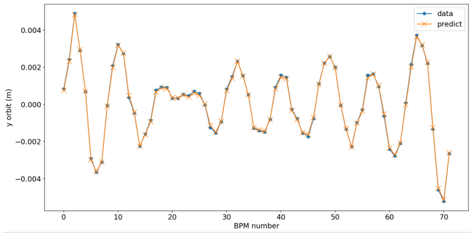

Figure 3: Vertical orbit predictions produced by a $\underline{R}$ model.

The benefit of building a NN model for $\underline{R}$ and $\underline{R}^{-1}$ is multi-faceted. First, the model can be retrained in seconds with new data to adapt to a new machine configuration, thus eliminating the need to spend more than 30 minutes to re-

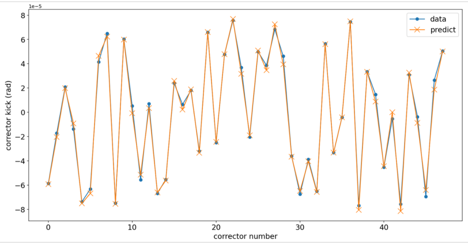

Figure 4: Corrector setting predictions produced by a $\underline{R}^{-1}$ model.

measure $\underline{R}$. Second, SVD calculation is not needed for orbit correction since the NN can model $\underline{R}^{-1}$ directly. Finally, the $\underline{R}$ model can be used to calculate $\Delta\underline{R}$ in real-time for machine error detection method described in the previous section.

SENSITIVITY STUDIES OF ORBIT RESPONSE MATRIX

The end goal of the studies is to develop an error-detection model using neural networks that can identify the kind and amplitude of machine error given a measured ORM $\underline{R}^{meas}$. In order to build such a model, we have to determine which error sources are important to change in $\underline{R}^{meas}$, so that they can be included as outputs of the error-detection model.

Table 1 lists the parameters considered in this study, and the value ranges within which they were changed. In this work, sensitivity is quantified with two criteria. One is the effect a control parameter has on the root mean square (RMS) of $\underline{R}^{meas}$ as percentage change, the other is the effect on beta-beating, which is defined as:

$$\frac{\Delta\beta}{\beta} = \frac{\beta_{meas} - \beta_{model}}{\beta_{model}} \tag{7}$$

Table 1: AGS Error Sources Scan Range

Name	Unit	Range
Main magnet roll error	mrad	[-0.5, 0.5]
Main magnet gradient error	m^{-2}	± 0.1%
Quadrupole gradient error	m^{-2}	± 0.2%
Sextupole offset error	mm	[-8, 8]

The AGS has 240 main magnets, with 20 magnets in each super-period. They are combined function magnets which are defined in simulation as Rbend dipoles with non-zero quadrupole (k1) and sextupole (k2) kicks. According to past survey data, there is a strong systematic super-periodicity in the main magnet roll errors. In each super-period, magnets 01 - 10 tend to have positive roll errors, while magnets 11 to 20 tend to have negative roll errors.

Table 2 shows the sensitivity results to main magnet roll errors by scanning roll errors of [0, 0.5] mrad for magnets 01 - 10 and [-0.5, 0] mrad for magnets 11 - 20. The machine is much more sensitive to magnets 11 - 20, which means the direction of roll plays an important role. We also observe that roll errors in super-periods A, C, E, G, I, K cause more changes than other super-periods. RMS value changes are 0.25% greater and beta-beatings are 0.2% greater.

Table 2: ORM Sensitivity to Main Magnet Roll Error

Magnet	$\Delta\underline{R}^{rms}$ (%)	$\Delta\beta_x$ (%)	$\Delta\beta_y$ (%)
01 - 10	[-0.13, 0]	[-2.5, 4.5]	[-4.5, 4.7]
11 - 20	[-0.1, 0.52]	[-5.7, 5.6]	[-8.5, 9.3]

For the main magnet gradient errors, the main magnets are divided into six families: AD, AF, BD, BF, CD, CF. Their quadrupole and sextupole kick values are calculated with different polynomials in simulation. Furthermore, magnets in the A and C families have length of 94 inches, while magnets in the B family have length of 79 inches. For each 20 magnets in one super-period, there are two AD, two AF, four BD, four BF, four CD, and four CF magnets. Table 3 shows the sensitivity results to main magnet gradient errors.

Table 3: ORM Sensitivity to Main Magnet Gradient Error

Family	$\Delta\underline{R}^{rms}$ (%)	$\Delta\beta_x$ (%)	$\Delta\beta_y$ (%)
AD	[-1.6, 1.8]	± 0.08	± 0.1
AF	[-0.01, 0.11]	± 0.12	± 0.09
BD	[-2.34, 2.87]	± 0.06	± 0.1
BF	[-0.14, 0.46]	± 0.1	± 0.06
CD	[-2.11, 2.72]	± 0.23	± 0.29
CF	[-0.73, 1.18]	± 0.34	± 0.23

Table 4 shows the sensitivity results to quadrupole gradient errors and sextupole offset errors. For future work, we aim to survey more error sources such as roll errors for the partial Siberian Snakes [1] and BPM errors.

Table 4: ORM Sensitivity to Quadrupoles & Sextupoles

Source	$\Delta\underline{R}^{rms}$ (%)	$\Delta\beta_x$ (%)	$\Delta\beta_y$ (%)
QH k1	± 0.0048	± 0.0015	± 0.007
QV k1	± 0.00037	± 0.0049	± 0.0044
SXH x-off	[-0.39, 0.6]	[-1.04, 1.5]	[-1.29, 1.55]
SXV x-off	[-1.4, 2]	[-0.9, 0.8]	[-2.46, 3.04]
SXH y-off	[0, 0.11]	[-0.017, 0.005]	[0, 0.07]
SXV y-off	[0, 0.15]	[-0.005, 0.025]	[0, 0.14]

CONCLUSIONS

In this work, we tested an ORM-based machine error detection method on the AGS, and built satisfactory NN models for the AGS ORM and its inverse using AGS lattice built in Bmad. We also conducted sensitivity studies on the AGS ORM. This work demonstrate the possibility and benefit of applying machine learning techniques to improve the efficiency of orbit correction and error detection in accelerator control systems.

REFERENCES

[1] H. Huang *et al.*, "Overcoming depolarizing resonances with dual helical partial siberian snakes," *Phys. Rev. Lett.*, vol. 99, no. 15, p. 154 801, 2007. doi:10.1103/PhysRevLett.99.154801

[2] Y. Chung, G. Decker, and K. Evans, "Closed orbit correction using singular value decomposition of the response matrix," in *Proc. PAC'93*, Washington, DC, USA, 1993, pp. 2263–2265. doi:10.1109/PAC.1993.309289

[3] G. H. Hoffstaetter, J. Keil, and A. Xiao, "Orbit-response matrix analysis at HERA," in *Proc. EPAC'02*, Paris, France, June 2002, pp. 407–409.

[4] H. Grote and F. Schmidt, "MAD-X: An upgrade from MAD8," *Conf. Proc. C*, vol. 030512, p. 3497, 2003.

[5] D. Sagan, "Bmad: A relativistic charged particle simulation library," *Nucl. Instrum. Methods Phys. Res., Sect. A*, vol. 558, no. 1, pp. 356–359, 2006. doi:10.1016/j.nima.2005.11.001

[6] Y. Bai, Y. Wei, W. Liu, G. Xu, and J. Wang, "Research on the slow orbit feedback of BEPCII using machine learning," *Rad. Det. Tech. Meth.*, vol. 6, no. 2, pp. 179–186, 2022. doi: 10.1007/s41605-022-00318-4

[7] N. Ketkar, "Feed forward neural networks," in *Deep Learning with Python: A Hands-on Introduction*. 2017, pp. 17–33. doi: 10.1007/978-1-4842-2766-4_3