CLASSE
Cornell Laboratory for Accelerator-based ScienceS & Education

Brookhaven National Laboratory

The Center for BRIGHT BEAMS
A National Science Foundation Science & Technology Center

U.S. DEPARTMENT OF ENERGY

# Simulation Studies and Machine Learning Applications for Orbit Correction at AGS


Weijian (Lucy) Lin, Bohong Huang, Weining Dai, Vincent Schoefer, Kevin Brown, Georg Hoffstaetter

Cornell ERL/EIC Group; BNL Collider-Accelerator Department; Stony Brook University

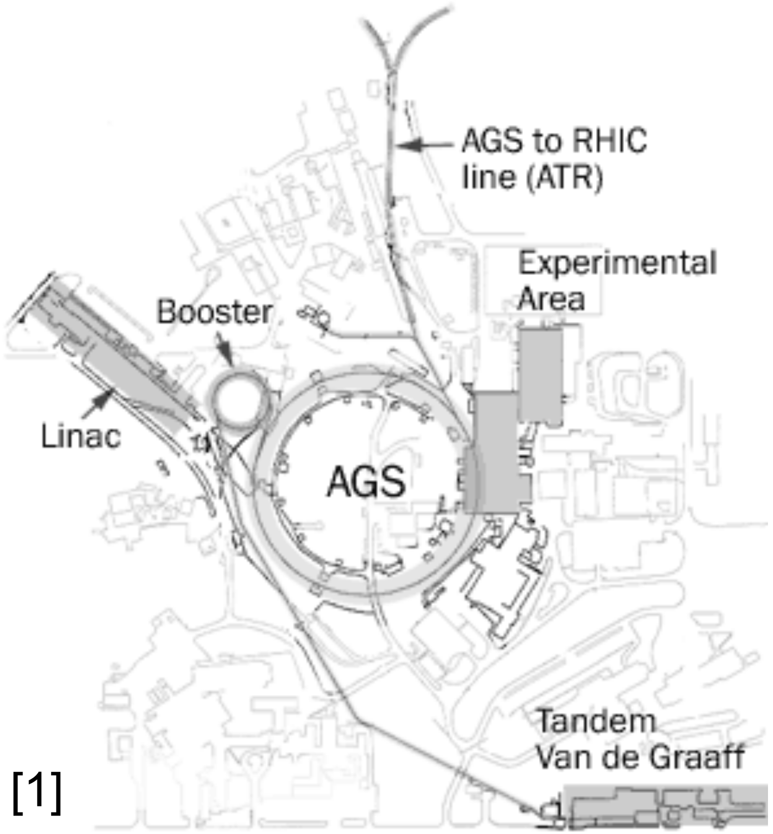


3rd ICFA Beam Dynamics Mini-Workshop on Machine Learning Applications for Particle Accelerators

Chicago, IL, USA    Nov 3rd, 2022


@BrookhavenLab

# Brightness control at the Alternating Gradient Synchrotron (AGS)




[1]

- Alternating gradient / strong focusing principle: achieve strong vertical and horizontal focusing of charged particle beam at the same time
- Accelerates proton to 33 GeV in 1960
- 12 super-periods (A to L), 240 main magnets, 810 m circumference
- Now serves as injector for Relativistic Heavy Ion Collider (RHIC)

Brookhaven National Laboratory

CLASSE Cornell Laboratory for Accelerator-based ScienceS & Education

# Motivation: support for EIC Cooler

- Electron cooling for the EIC requires small incoming emittances from the AGS
- Necessary pre-cooler at RHIC injection energy (AGS extraction energy)
- Current AGS lacks systematic tuning routine, mostly hand tuned by operators
- Algorithm to better control beam in AGS will be helpful for future EIC cooler

# Orbit Response Matrix (ORM)

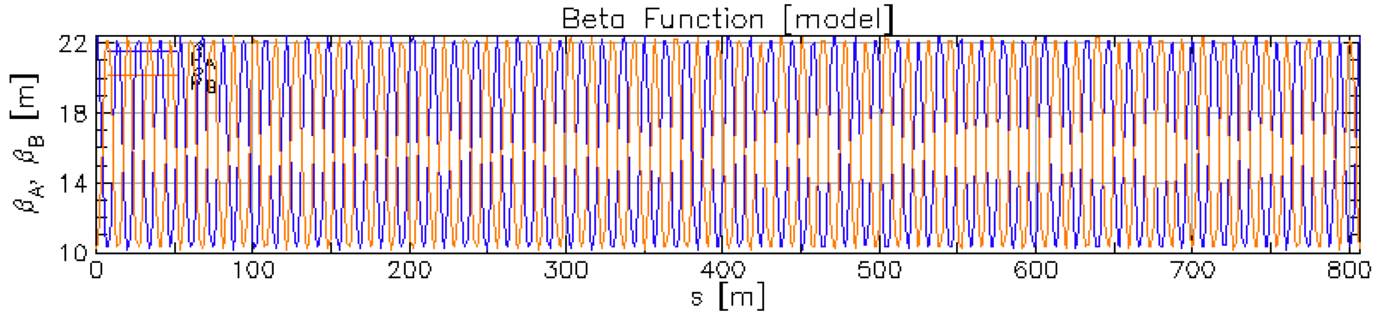
- Mapping $\underline{R}$ between closed orbit measurements and corrector settings
- 72 pick-up electrodes (PUE), 48 horizontal and vertical corrector pairs
- Linear orbit response to corrector change: calculate $\underline{R}$ matrix by changing each corrector pair separately
- Corrector current $I \rightarrow$ angle $\theta$ by calibration factor
- Traditional orbit correction: $\Delta\vec{\theta} = \underline{R}^{-1}\,\Delta\vec{y}$

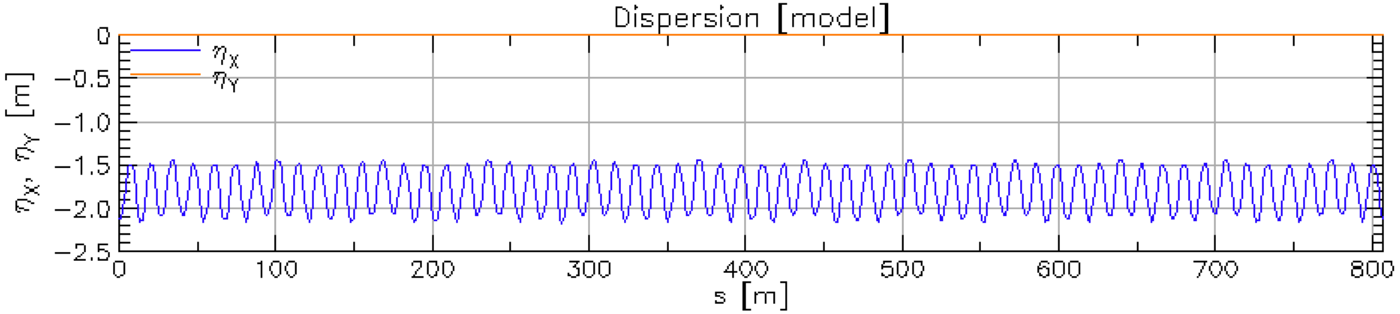
$$\begin{pmatrix} \Delta\vec{x} \\ \Delta\vec{y} \end{pmatrix} = \underline{R} \begin{pmatrix} \Delta\vec{\theta}_x \\ \Delta\vec{\theta}_y \end{pmatrix}$$

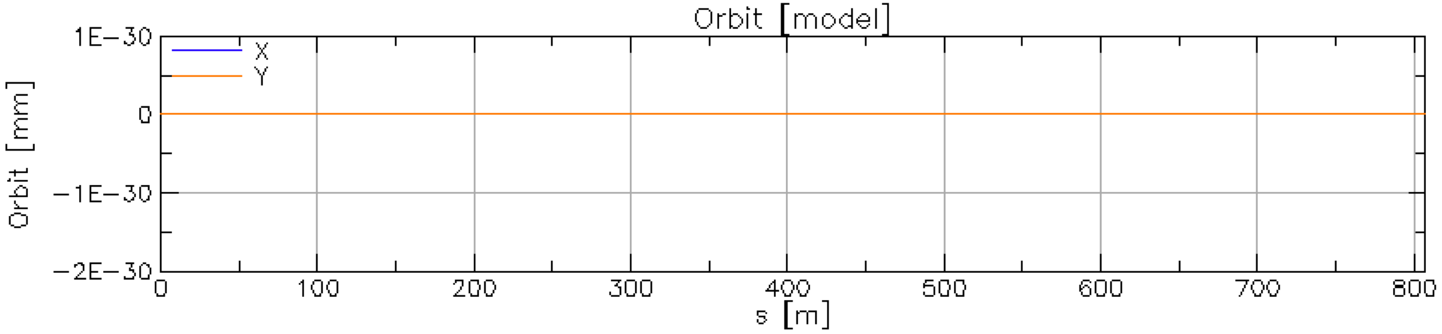
$$\frac{\Delta x_i}{\Delta\theta_j} = R_{ij}$$

# MAD-X to BMAD translation

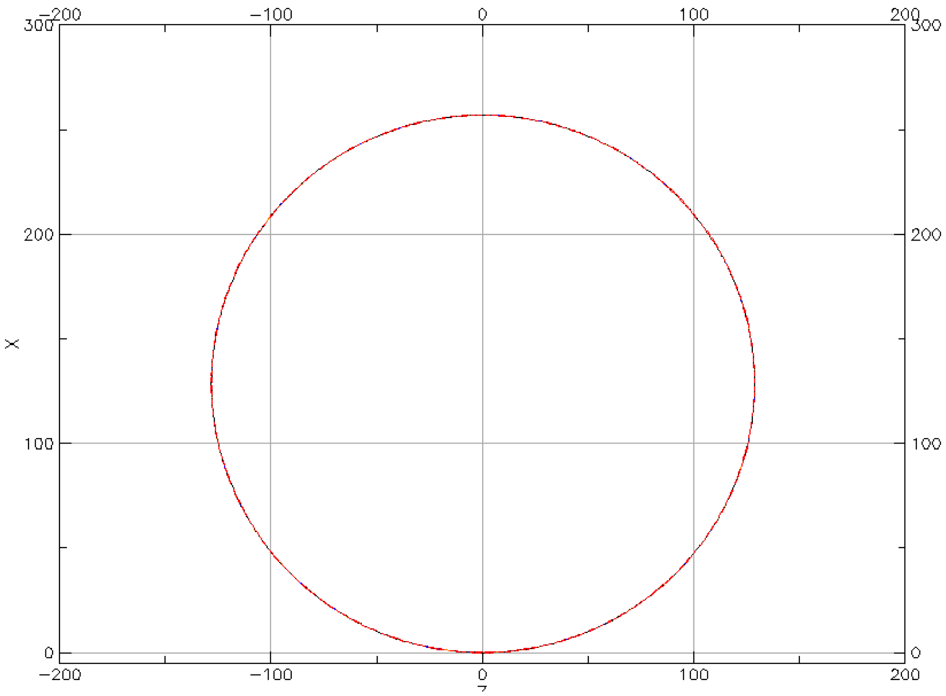
- Successfully translated bare machine to BMAD: ramping in progress
- Can use Python interface (PyTao) to run simulations much easier



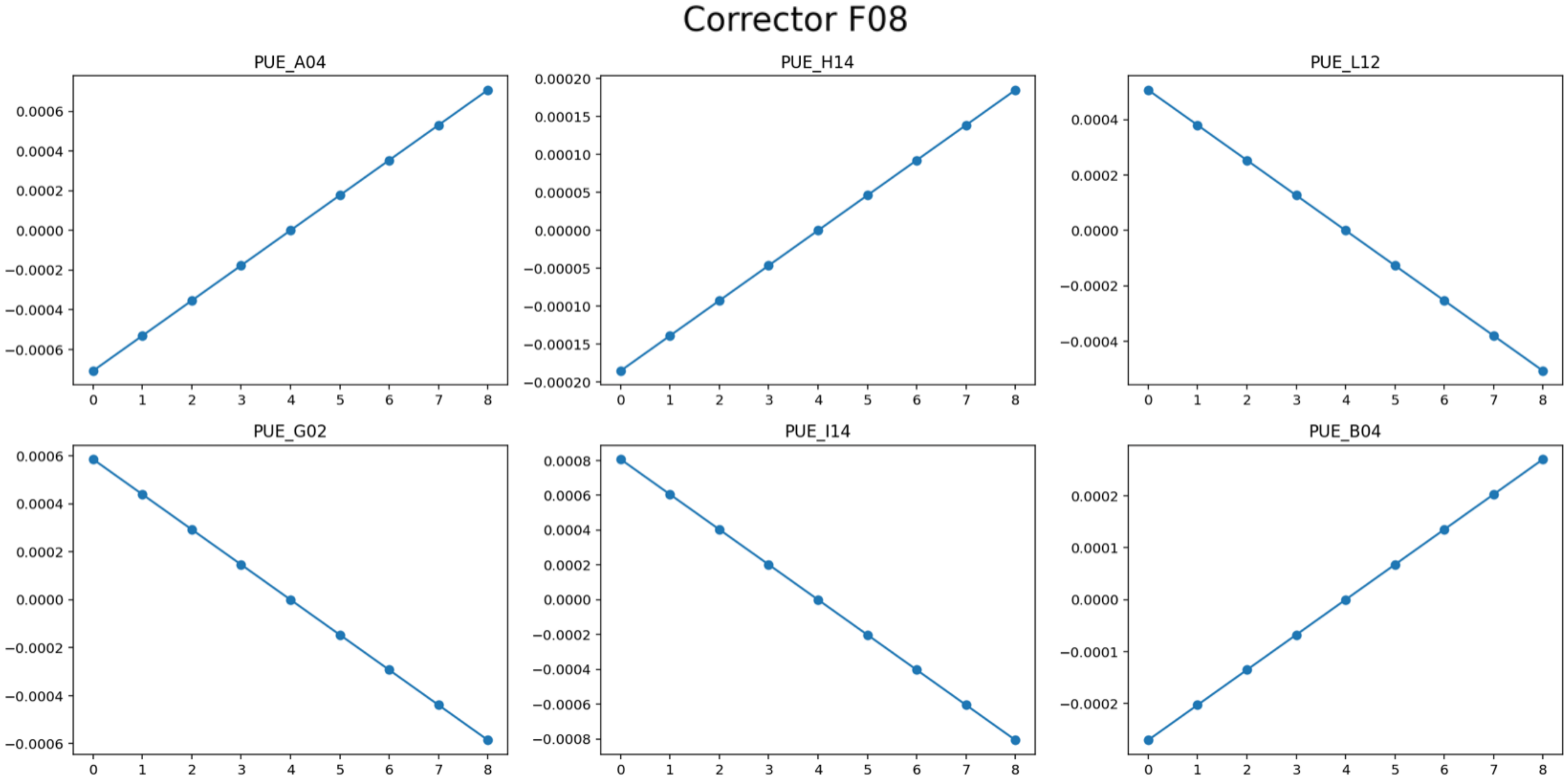




Floor plan



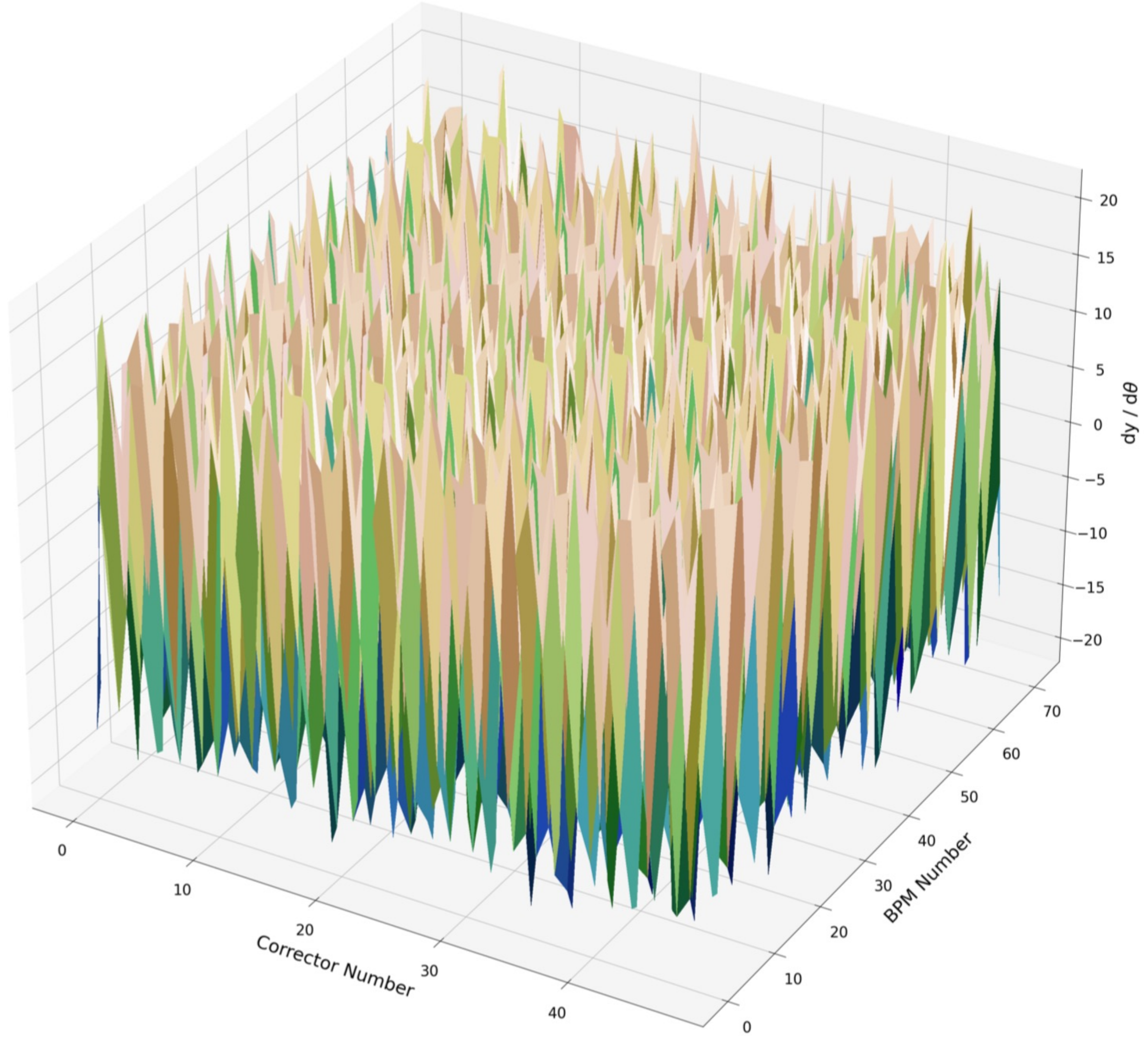
# Orbit Response vs. One Corrector (Sim.)

- PUE = pick-up electrode = BPM
- Vertical axes = vertical orbit in meters



# Reference $\underline{R}_y$ matrix

- Reference = bare machine (only main magnets turned on), no error
- Change vertical correctors, observe change in vertical orbit



# Use ORM to identify machine errors

- Actual machine with errors (e.g. quadrupole gradient errors, corrector calibration errors, etc.) produce different $\underline{R}_{measured}$ from model/reference machine $\underline{R}_{model}$

$$\Delta R_{ij} = R_{ij}^{model} - R_{ij}^{measured}$$

- Considering all possible sources of errors as a vector $\vec{\nu}$, build response error model $\underline{J}_{model}$

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

- Reconstruct any $\vec{\nu}$ given known $\Delta\vec{R}$ and $\underline{J}_{model}$

# Reconstruct errors using SVD

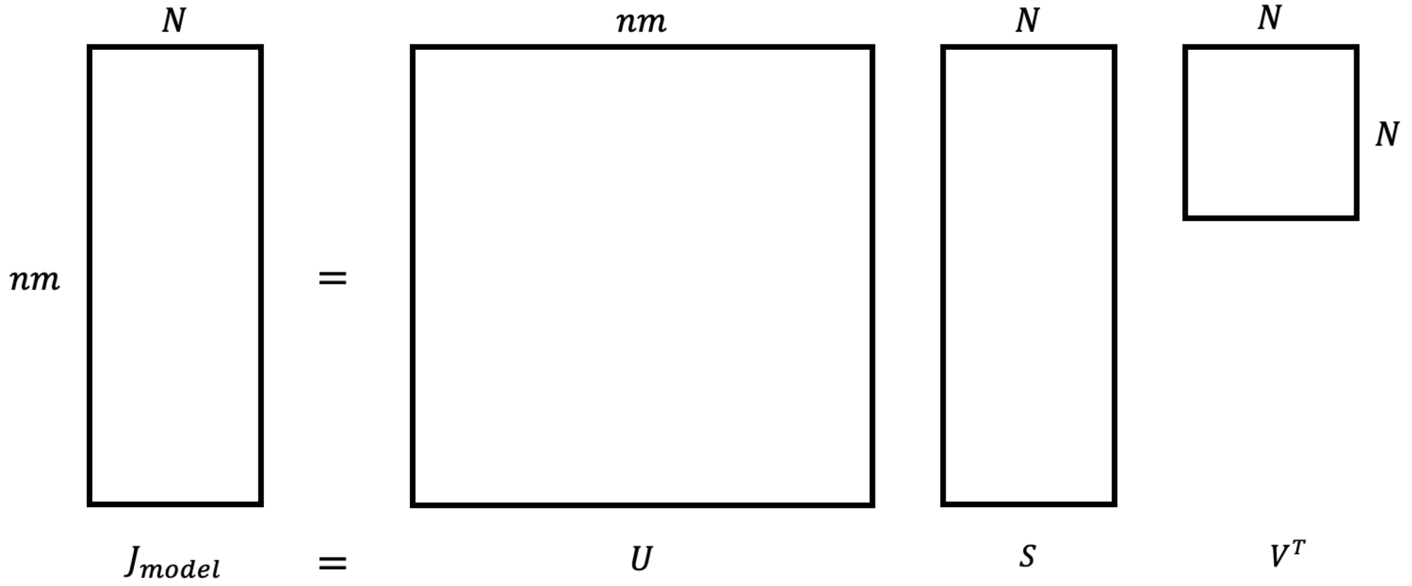
- Traditional tuning routine: perform singular value decomposition (SVD) directly on $\underline{R}$
- Machine error detection: perform SVD on $\underline{J}_{model}$
- Solve for $\Delta\vec{v}$ using $\Delta\vec{R} = \underline{J}_{model}\,\Delta\vec{v}$, where $\underline{J}_{model}$ is not a square matrix

$$J_{model} = USV^T$$

$$n = N_{corr}, m = N_{BPM}$$

$$\Delta\vec{R}: (48 \times 72, 1)$$

$$\underline{J}_{model}: (3456, N_{error})$$



# Test case: quadrupole strength error

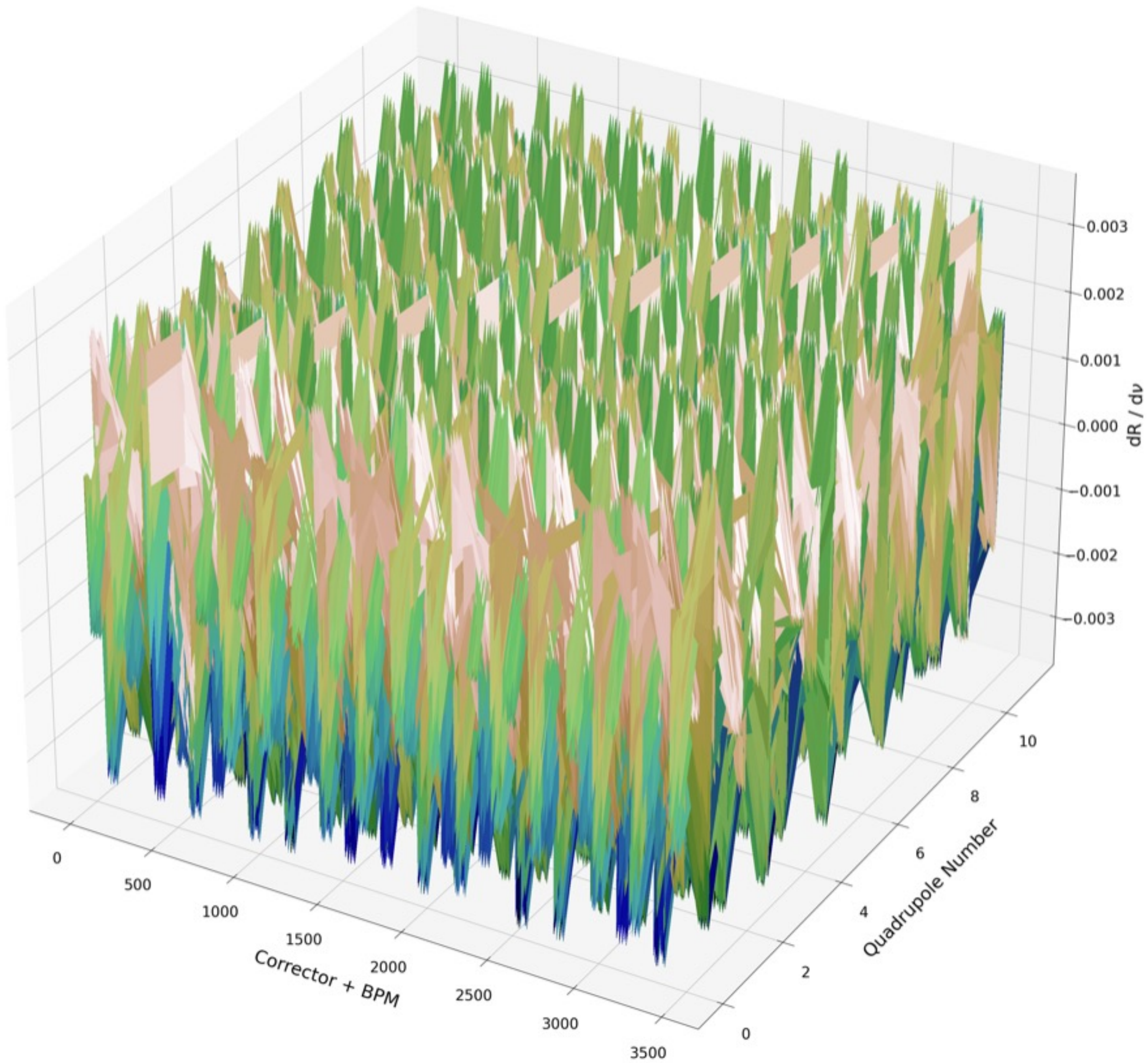
- 24 quadrupoles (12 horizontal, 12 vertical), 1 in each super-period
- Linear orbit response to quadrupole kick change: calculate $\Delta\vec{R} = \underline{R}_{measured} - \underline{R}_{ref}$ by changing each quadrupole separately $\rightarrow J_{ijk} = \frac{\Delta R_{ij}}{\Delta v_k}$
- Quad kick defined with one variable KQH/KQV in MAD-X → variables in BMAD allow separate change of quad kicks

```
tao.cmd('show var quads.x')
```

```
['   Variable                      Slave Parameters            Meas        Model        Design  Useit_opt',
 '   quads.x[1]                    QH_F17[K1]            0.0000E+00  -6.5349E-05  -6.5349E-05        T',
 '   quads.x[2]                    QH_G17[K1]            0.0000E+00  -6.5349E-05  -6.5349E-05        T',
 '   quads.x[3]                    QH_H17[K1]            0.0000E+00  -6.5349E-05  -6.5349E-05        T',
 '   quads.x[4]                    QH_I17[K1]            0.0000E+00  -6.5349E-05  -6.5349E-05        T',
 '   quads.x[5]                    QH_J17[K1]            0.0000E+00  -6.5349E-05  -6.5349E-05        T',
 '   quads.x[6]                    QH_K17[K1]            0.0000E+00  -6.5349E-05  -6.5349E-05        T',
 '   quads.x[7]                    QH_L17[K1]            0.0000E+00  -6.5349E-05  -6.5349E-05        T',
 '   quads.x[8]                    QH_A17[K1]            0.0000E+00  -6.5349E-05  -6.5349E-05        T',
 '   quads.x[9]                    QH_B17[K1]            0.0000E+00  -6.5349E-05  -6.5349E-05        T',
 '   quads.x[10]                   QH_C17[K1]            0.0000E+00  -6.5349E-05  -6.5349E-05        T',
 '   quads.x[11]                   QH_D17[K1]            0.0000E+00  -6.5349E-05  -6.5349E-05        T',
 '   quads.x[12]                   QH_E17[K1]            0.0000E+00  -6.5349E-05  -6.5349E-05        T',
 '   Variable                      Slave Parameters            Meas        Model        Design  Useit_opt']
```

# Test case $J_{model}$ matrix (horizontal)

- Calculated using $\Delta\nu = 40$ A in power supply current for each quadrupole ($\pm 10\%$ in k1 value, later reproduced using $\pm 1\%$ in k1)
- Agreement with MAD-X model (redefined every quad individually) was obtained



# Reconstruct errors using SVD

- $\underline{U}$ and $\underline{V}$ are square orthogonal matrices: $UU^T = VV^T = I$
- $\underline{S}$ is an $nm \times N$ matrix whose first $N$ diagonal elements are singular values $\sigma$ of $\underline{J}_{model}$

$$S = \begin{pmatrix} S_N \\ 0 \end{pmatrix} \in \mathbb{R}^{nm \times N}, \;\; S_N := diag(\sigma_1, \ldots, \sigma_N, 0, \ldots, 0) \in \mathbb{R}^{N \times N}$$

- $\underline{S}^+$ is pseudoinverse of $\underline{S}$ whose first $N$ diagonal elements are $\frac{1}{\sigma}$

$$S^+ = \begin{pmatrix} S_N^+ \\ 0 \end{pmatrix} \in \mathbb{R}^{N \times nm}, \;\; S_N^+ := diag(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_N}, 0, \ldots, 0) \in \mathbb{R}^{N \times N}$$

$$\begin{pmatrix} \Delta\nu_1 \\ \Delta\nu_2 \\ \cdots \\ \Delta\nu_{N-1} \\ \Delta\nu_N \end{pmatrix} = J^+_{model} \begin{pmatrix} \Delta R_{11} \\ \Delta R_{12} \\ \cdots \\ \Delta R_{n(m-1)} \\ \Delta R_{nm} \end{pmatrix} = V S^+ U^T \begin{pmatrix} \Delta R_{11} \\ \Delta R_{12} \\ \cdots \\ \Delta R_{n(m-1)} \\ \Delta R_{nm} \end{pmatrix}$$

# Test case: reconstruct errors with $J\_model$

- Reconstructed error = quadrupole power supply current

**Case 1: One quadrupole 1% (4A) error**

```
# Quad A17 +4 Amp
np.dot(V, np.dot(S_inv, np.dot(UT, dr1)))
```

```
array([ 4.04152292e+00, -4.15488269e-05,  2.17313140e-05,  6.45374239e-05,
        4.03913733e-05,  3.09693635e-05,  2.76558248e-05, -4.31669566e-05,
       -1.36249941e-05,  4.91338661e-05, -6.14294896e-05,  3.19703471e-05])
```

**Case 2: Two quadrupoles 0.5% (2A), 0.18% (0.7A) error**

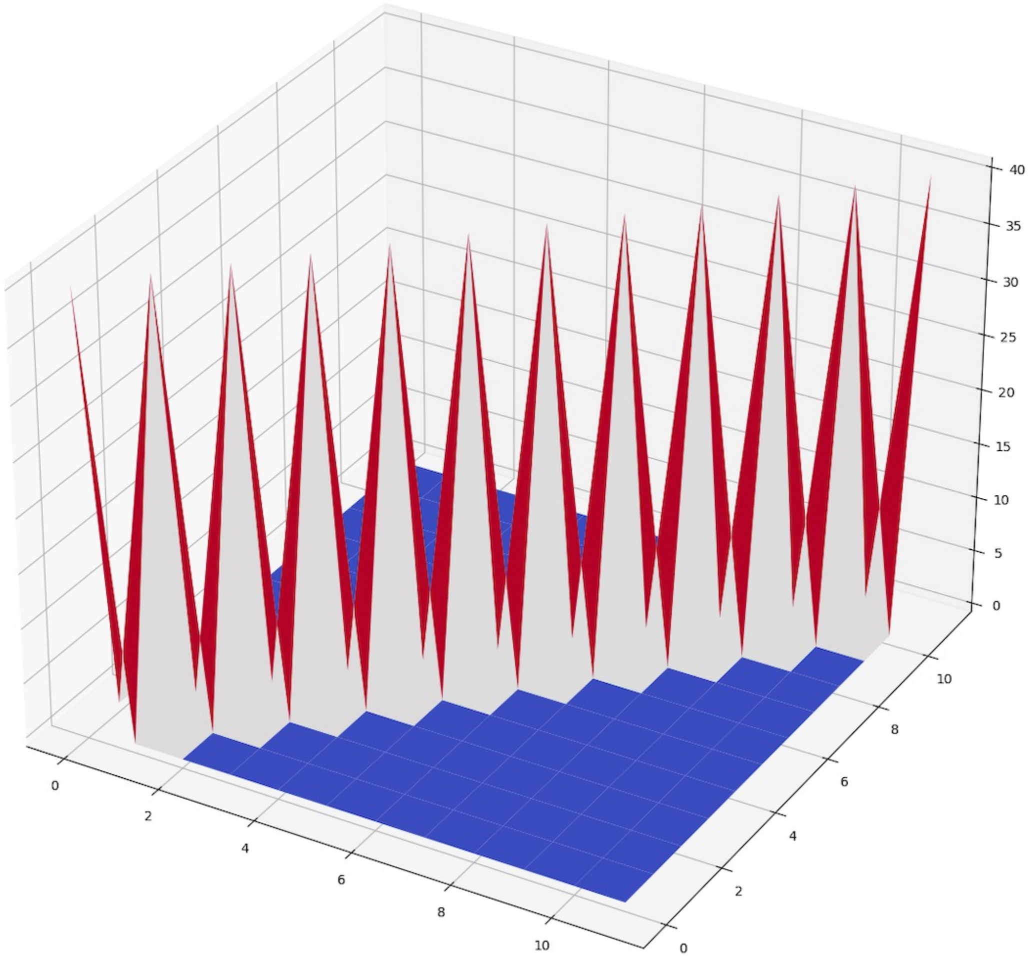
```
# Quad C17 +2 Amp, H17 +0.7 Amp
np.dot(V, np.dot(S_inv, np.dot(UT, dr2)))
```

```
array([ 3.50482558e-05, -5.54479409e-05,  2.02147800e+00,  7.69381741e-05,
        5.06832047e-05,  4.13148646e-05,  4.02598848e-05,  7.07636616e-01,
       -2.78341654e-05,  4.27531143e-05, -6.90270247e-05,  2.50657000e-05])
```

**Case 3: Three quadrupoles 0.75% (3A), 0.02% (0.08A), 0.25% (1A) error**

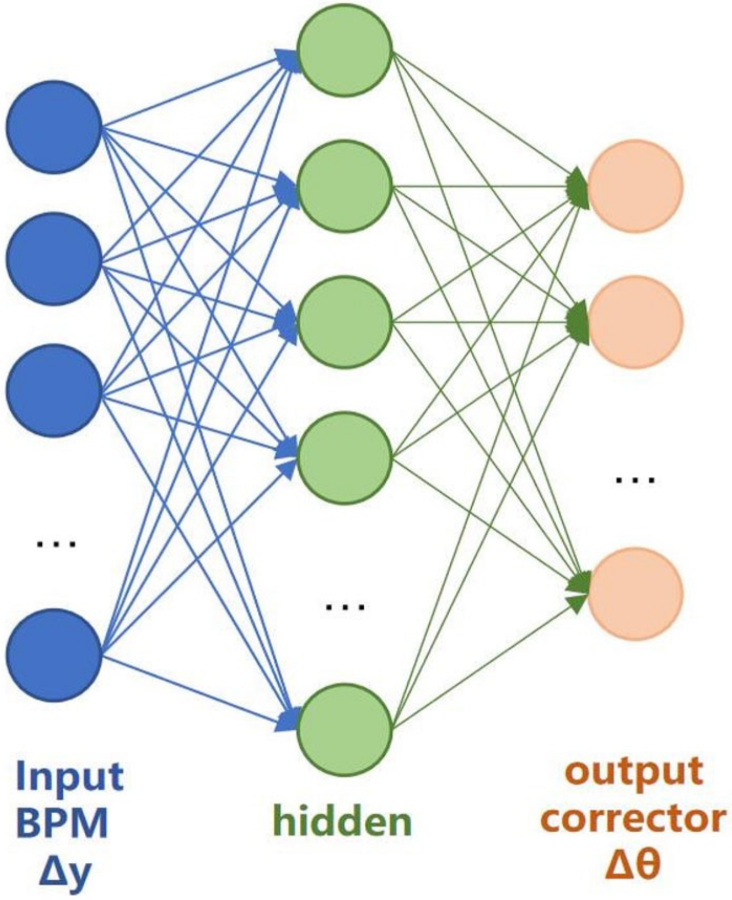
```
# Quad B17 +3 Amp, F17 +0.08 Amp, J17 +1 Amp
np.dot(V, np.dot(S_inv, np.dot(UT, dr3)))
```

```
array([ 6.97595445e-05,  3.03074518e+00, -1.42673230e-05,  8.18292016e-06,
        6.05175589e-05,  8.07700864e-02,  4.40237777e-05, -8.92267806e-05,
       -4.99647748e-05,  1.01013295e+00, -2.99336376e-05, -2.01460387e-04])
```




Satisfactory reconstruction results

# Neural Network for real-time ORM

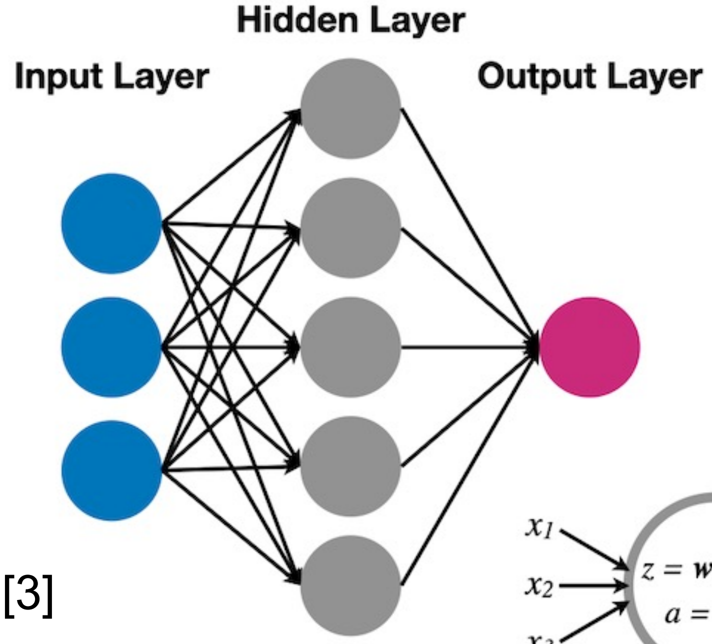


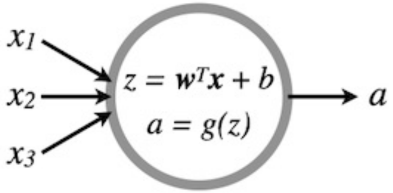

[2]

- Need dedicated machine time to measure ORM $\underline{R}_{measured}$: at least 30 min
- Pre-measured $\underline{R}_{measured}$ gets less accurate with time → orbit drift / brightness drop
- Update ORM with real-time data: build neural network model for $\underline{R}_{measured}$ or $\underline{R}_{measured}^{-1}$
- Can be used to calculate $\Delta\vec{R}$ for machine error reconstruction

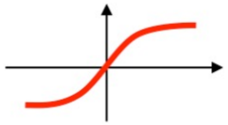
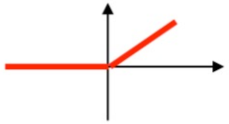
# Method: Feed Forward Neural Network

- Neural Network (NN) built with PyTorch library
- Fully connected layers: output = activation(dot(input, weight) + bias)
- Activation function: Hyperbolic Tangent (Tanh) and Rectified Linear Unit (ReLU)
- Feed forward neural network (FFNN): most common, no feedback route



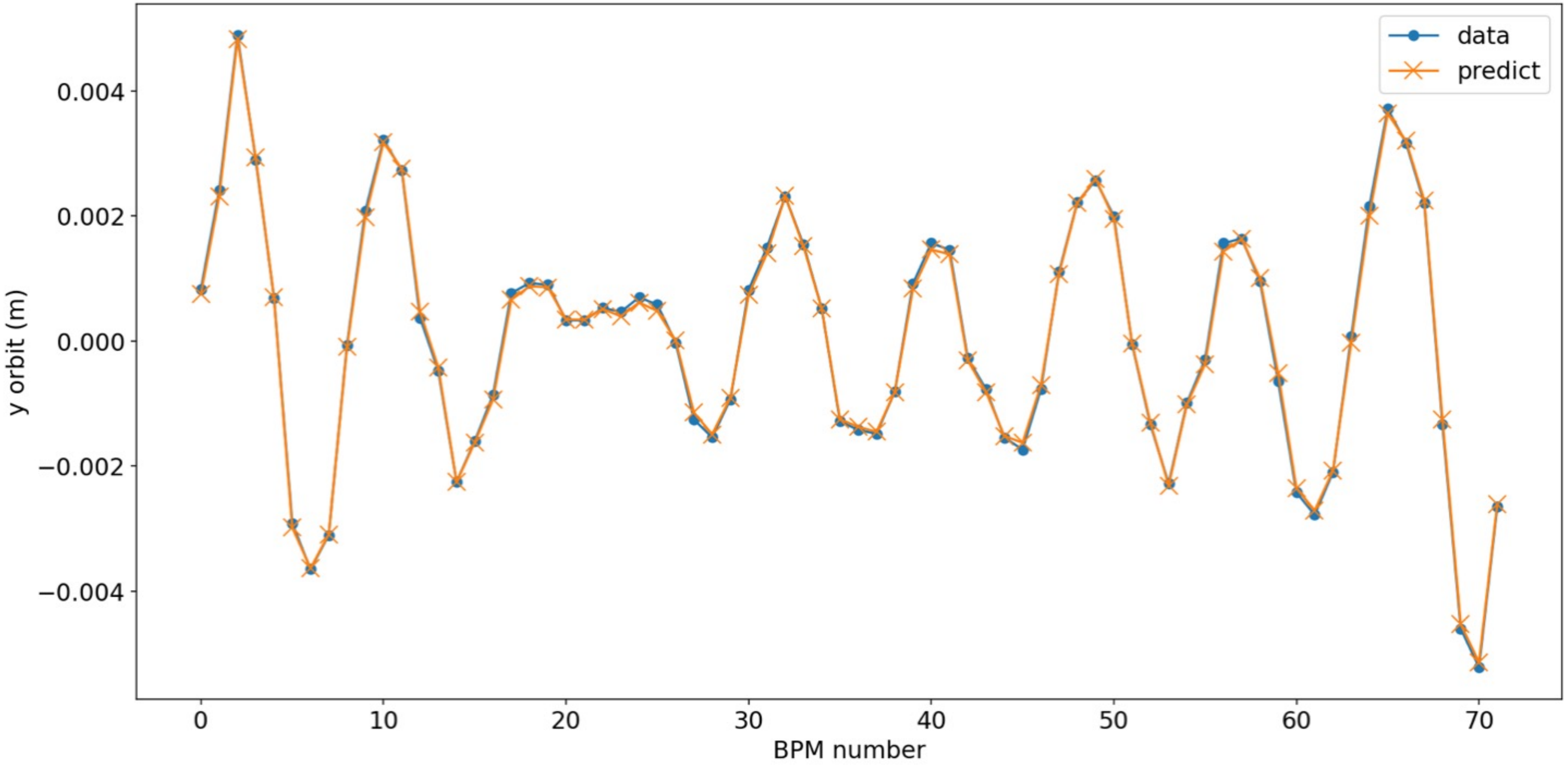

[3]

| | | | |
|---|---|---|---|
| Hyperbolic tangent | $\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ | Multi-layer Neural Networks |  |
| Rectifier, ReLU (Rectified Linear Unit) | $\phi(z) = max(0, z)$ | Multi-layer Neural Networks |  |

# ORM NN model: training results

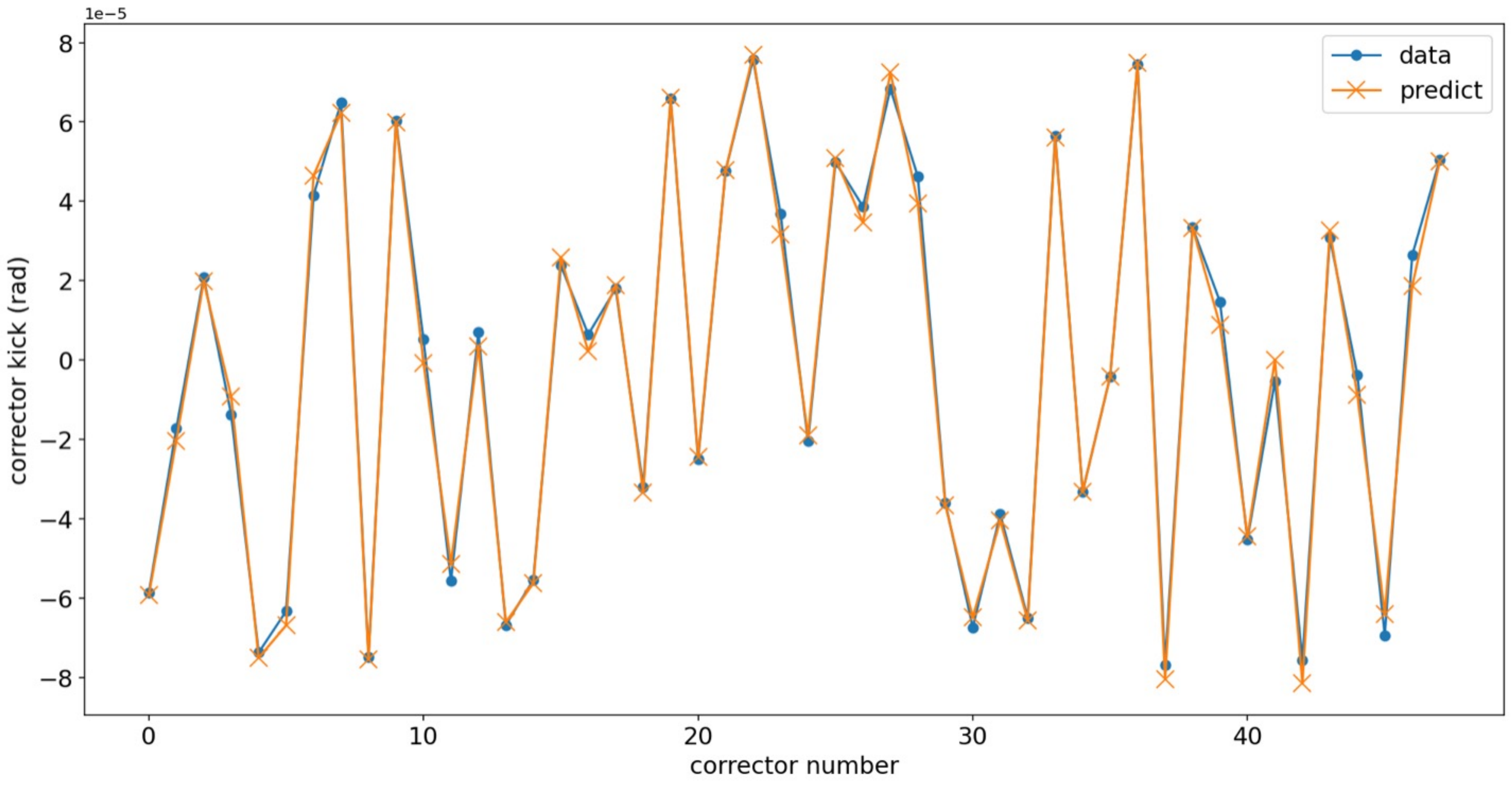
- Input 48 vertical corrector kick → Output 72 y orbit measured at BPM
- FFNN with one hidden layer and Tanh activation
- Trained on 800 data pairs, tested on 200 data pairs: $R^2$ score = 0.998

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \overline{y}_i)^2}$$



# Inverse ORM NN model: training results

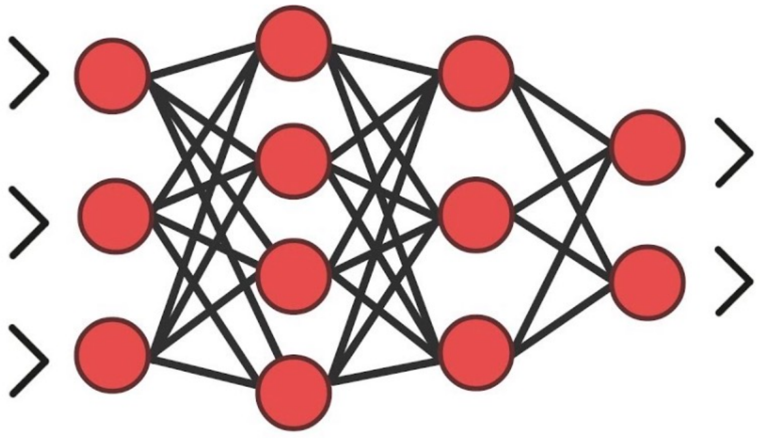
- Input 72 y orbit measured at BPM → Output 48 vertical corrector kick
- FFNN with one hidden layer and Tanh activation
- Trained on 800 data pairs, tested on 200 data pairs: $R^2$ score = 0.993



# Sensitivity studies for ORM

- Scan through some common sources of error to see how much ORM changes
- Find relevant parameters to include for building error-detecting model
- **Goal**: establish a neural network that identify error source given a measured ORM

$$\begin{pmatrix} \Delta\nu_1 \\ \Delta\nu_2 \\ \cdots \\ \Delta\nu_{N-1} \\ \Delta\nu_N \end{pmatrix} = J^{+}_{model} \begin{pmatrix} \Delta R_{11} \\ \Delta R_{12} \\ \cdots \\ \Delta R_{n(m-1)} \\ \Delta R_{nm} \end{pmatrix}$$



# Sensitivity studies: error sources

- Sources or error and ranges come from past survey data

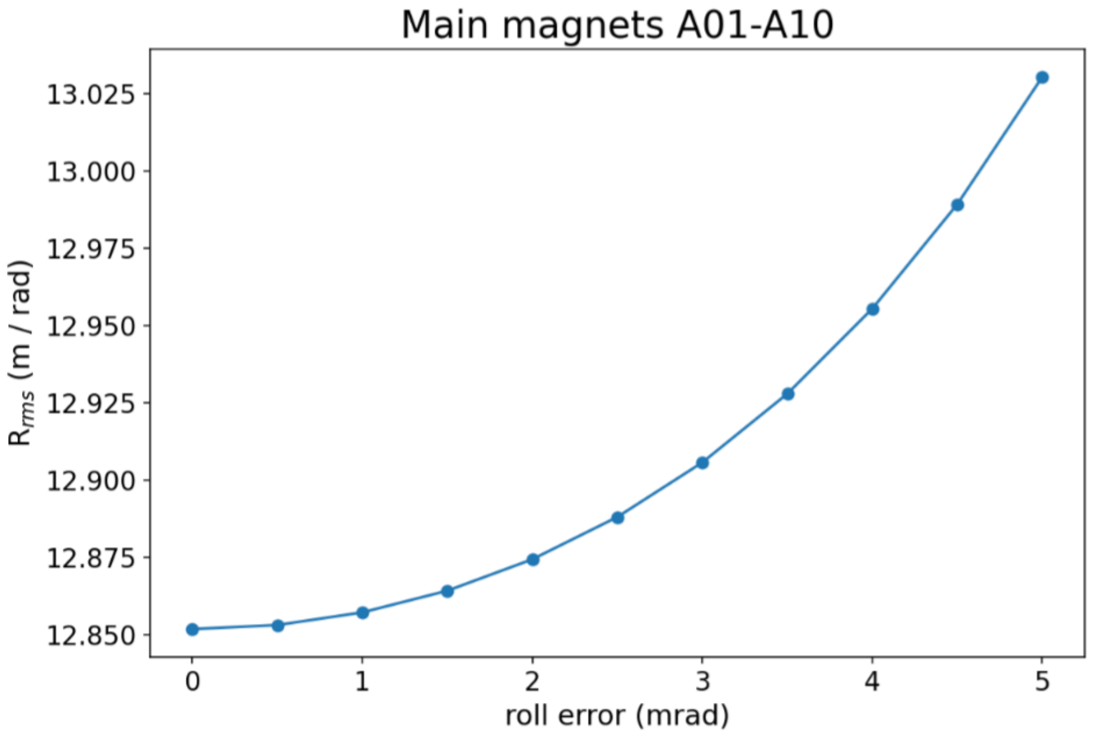
- Criteria to quantify & visualize sensitivity:
    - RMS of ORM matrix
    - Beta-beating (vertical & horizontal)

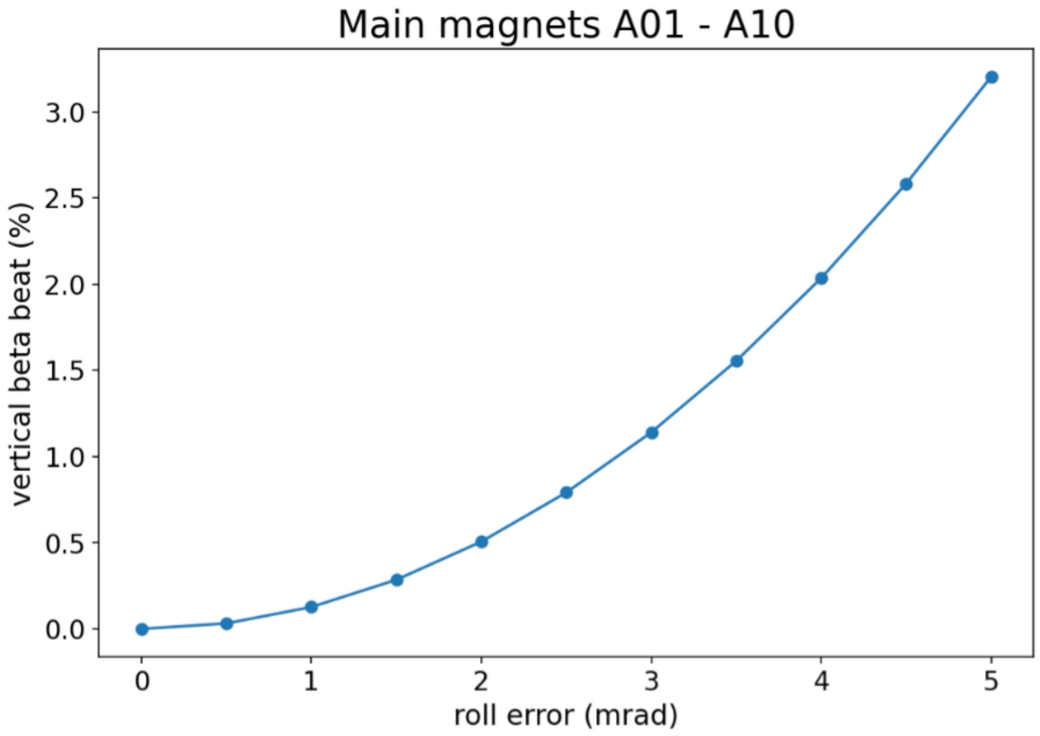
$$\frac{\Delta\beta}{\beta} = \frac{\beta_{measured} - \beta_{model}}{\beta_{model}}$$

| Name | Unit | Range |
|---|---|---|
| Main magnet roll error | mrad | [-0.5, 0.5] |
| Main magnet gradient error | $m^{-2}$ | $\pm 0.1\%$ |
| Quadrupole gradient error | $m^{-2}$ | $\pm 0.2\%$ |
| Sextupole offset error | mm | [-8, 8] |
| Snake magnet roll error | mrad | [-1.5, 1.5] |

# Main magnet roll error

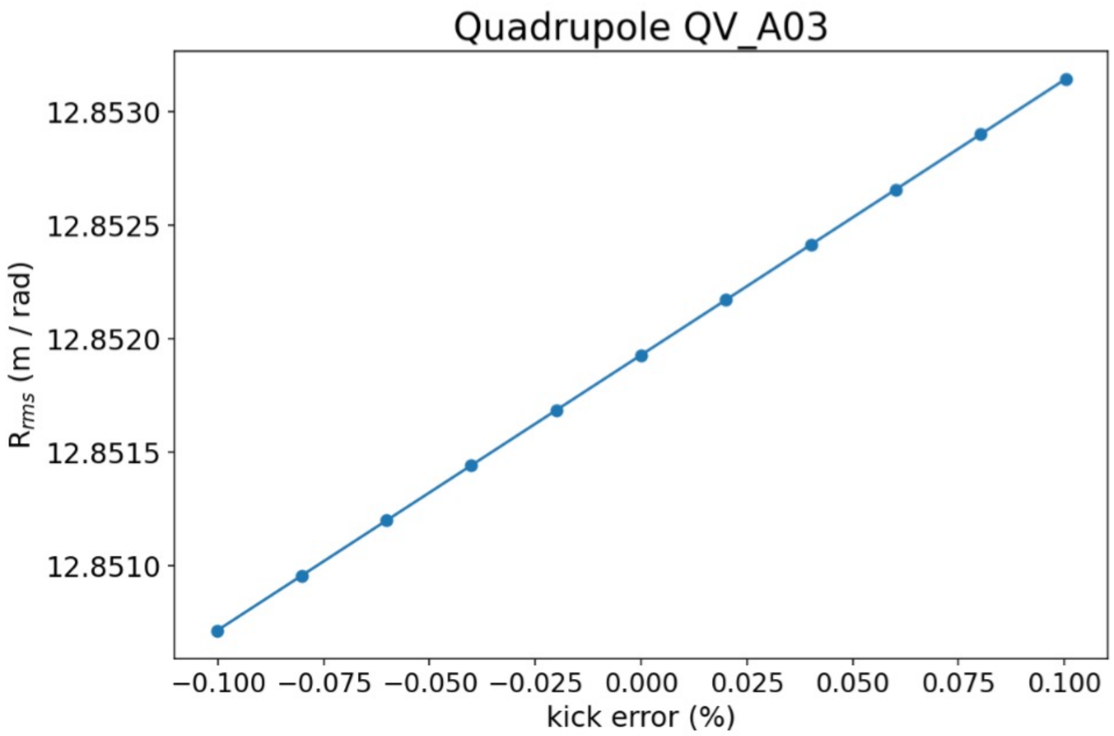
- 240 main magnets, 20 magnets (01 to 20) in each super-period (A to L)
- Combined function magnets: dipole (Rbend) with non-zero k1, k2
- Scan range: $\pm 5$ mrad with strong systematic super-periodicity (01 to 10 rolls one way, 11 to 20 rolls another way)

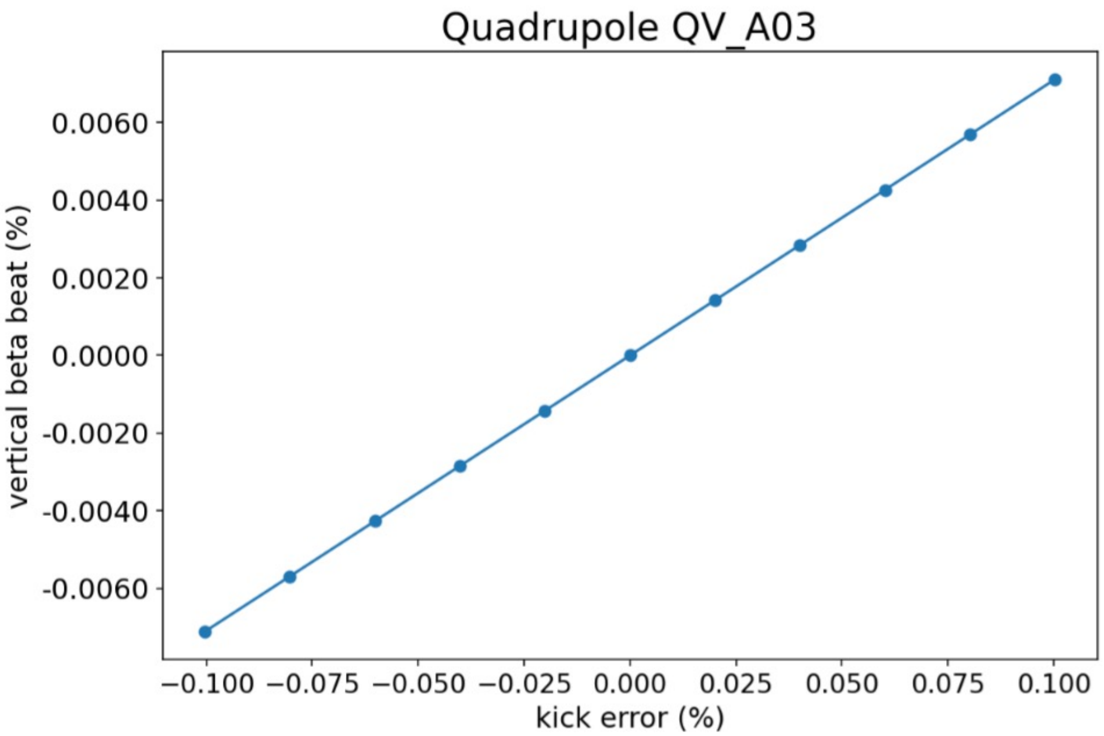


# Quadrupole kick error

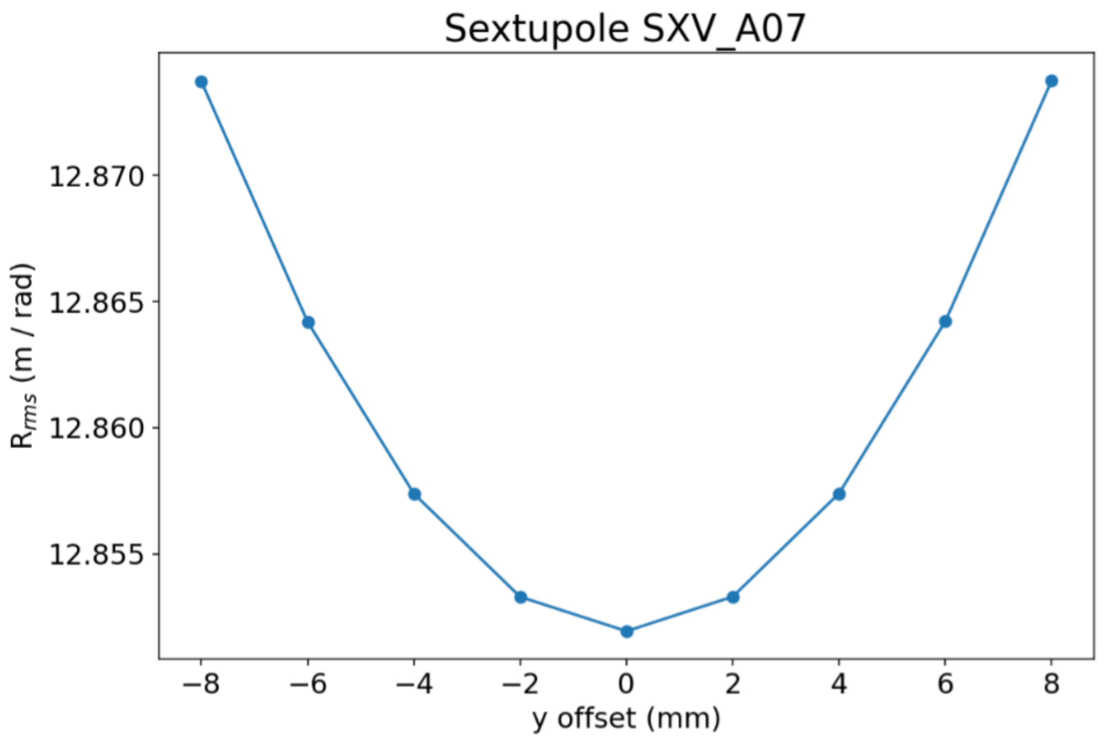
- 24 quadrupole magnets (12 horizontal, 12 vertical), one (17 for QH, 03 for QV) in each super-period
- Scan range: $\pm 0.1\%$ in k1 values

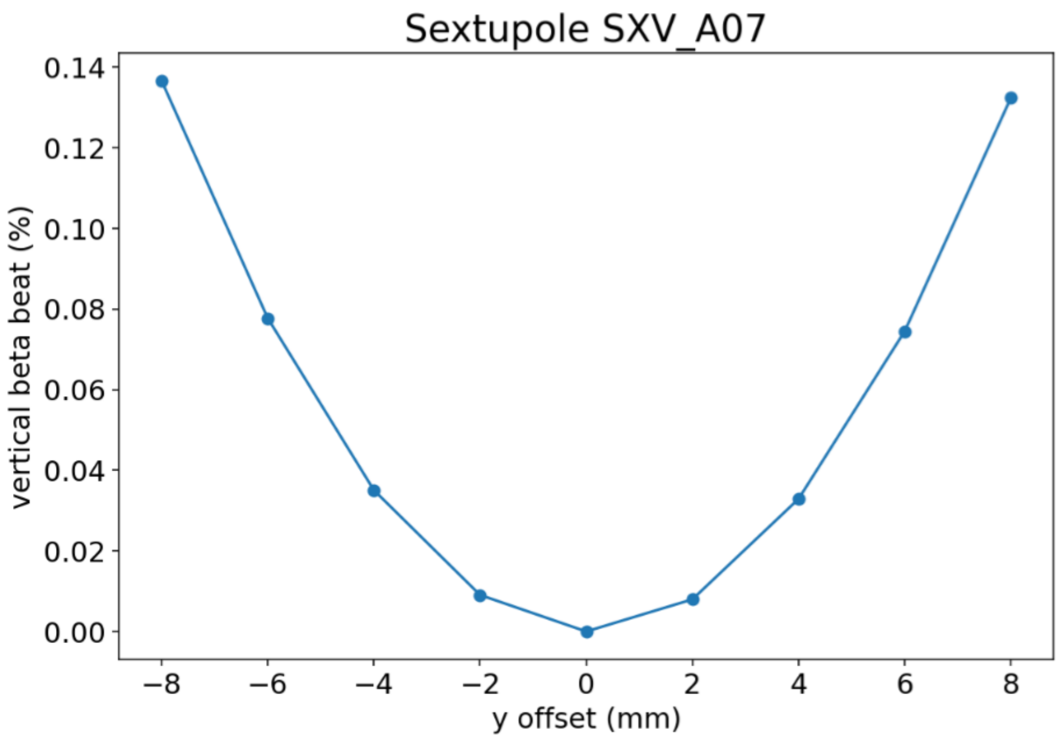


# Sextupole offset error

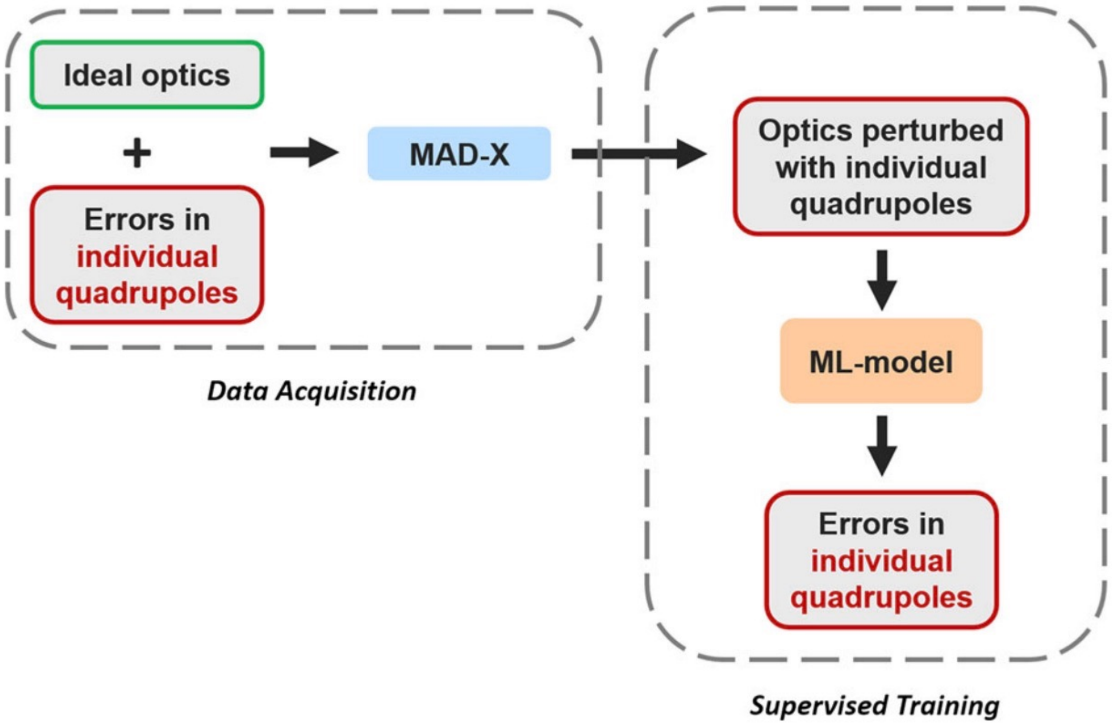
- 28 sextupole magnets (14 horizontal, 14 vertical), 2 chromaticity sextupoles (13 for SXH, 07 for SXV) per super-period
- Scan range: $\pm 8$ mm in x, y offset



# Future work

- Finish sensitivity scan to determine relevant error sources: snake magnet incorporation to Bmad using field maps in progress
- Make simulation more realistic: add Gaussian noises to both magnets and BPMs
- Establish a dynamic retraining routine to keep model updated during operation




[4]

# References


- [1] Alternating Gradient Synchrotron, https://www.bnl.gov/rhic/ags.php, Accessed on Sep. 6 2022.
- [2] Y. Bai et al., “Research on the slow orbit feedback of BEPCII using machine learning”, Rad. Det. Tech. Meth. 6, 179-186 (2022).
- [3] A shallow neural network for simple nonlinear classification, https://scipython.com/blog/a-shallow-neural-network-for-simple-nonlinear-classification/, Accessed on May 14 2022.
- [4] Fol, E., Tomás, R. & Franchetti, G., “Supervised learning-based reconstruction of magnet errors in circular accelerators”, Eur. Phys. J. Plus 136, 365 (2021).

CLASSE
Cornell Laboratory for Accelerator-based ScienceS & Education

The Center for BRIGHT BEAMS
A National Science Foundation Science & Technology Center

Brookhaven National Laboratory

U.S. DEPARTMENT OF ENERGY

# Thank you!

@BrookhavenLab