Machine Learning for Optic Correction in the LHC

Alejandro Börjesson Carazo

Thanks to Elena Fol and Felix Carlier

26/04/2023

Summary

Introduction

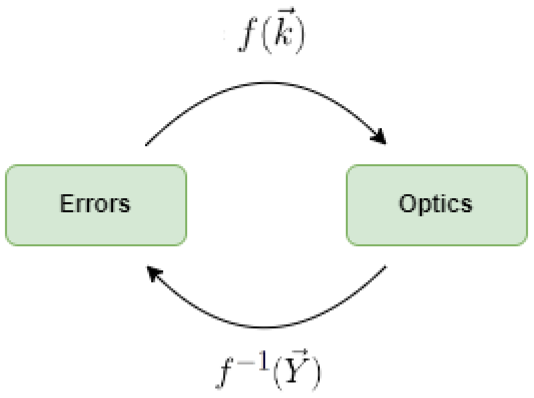
Elena's work continuation **"Supervised learning-based reconstruction of magnet errors in circular accelerators"** [1]

Actual methods for quadrupole magnetic error correction consist on knob tuning, response matrix …

- Problems, no information on the actual magnet errors
- Correcting the optics, but not the actual magnet errors!

Machine learning promises multiple new ways to manage quadrupole errors

- The effect of magnet errors on optics can be calculated using simulation software
- ML can be used to model the relation between optics and errors regardless on how complex



$$\vec{Y} = f(\vec{k})$$

$$\vec{k} = f^{-1}(\vec{Y})$$

$$\vec{Y} = (\beta_x, \beta_y, \mu_x \ldots)$$

Methods: Data pipeline

Generating random quadrupole strength errors and calculating the corresponding twiss parameters for data generation

Creating a ML model with:

- Input: Optic measurements, 3346 features
- Output: Quadrupole Magnet strength error

Possible improvements include:

- Using **MAD-NG** for non linear errors or model training
- Adding noise to simulated optic data, more realistic training data

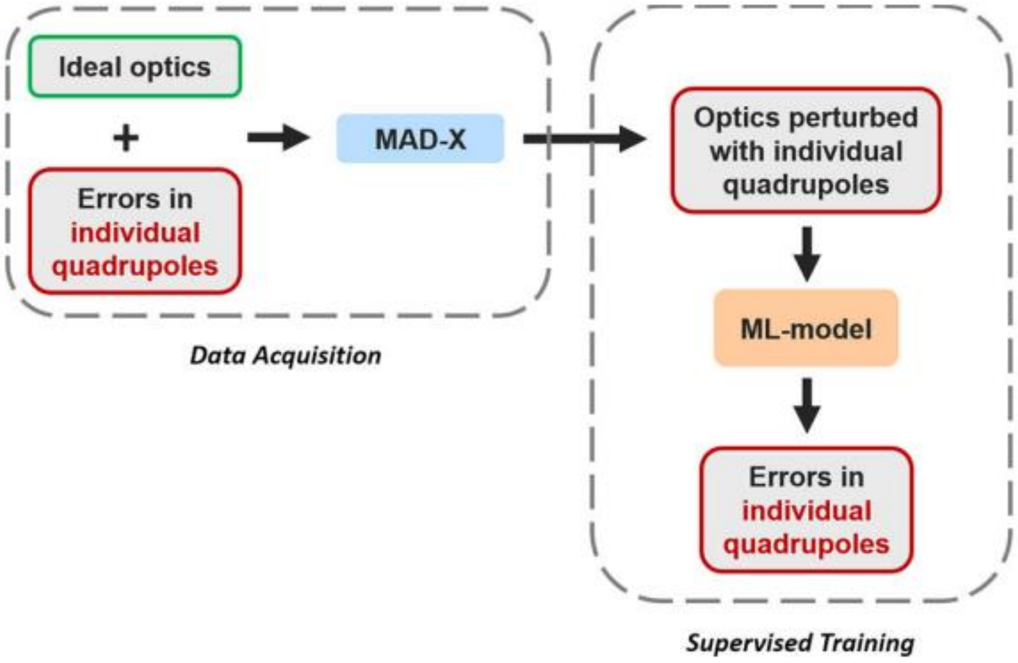


Fig 1. Data pipeline [1]

Methods: ML Model

Ridge and Linear Regression:

$$Loss = Error(Y - \widehat{Y}) + \lambda \sum_{1}^{n} w_i^2$$

Least squares regression with L2 regularization

Bagging:

Training using ten different subsets of data and averaging the results.

This methods decreases variance of the model and overfitting

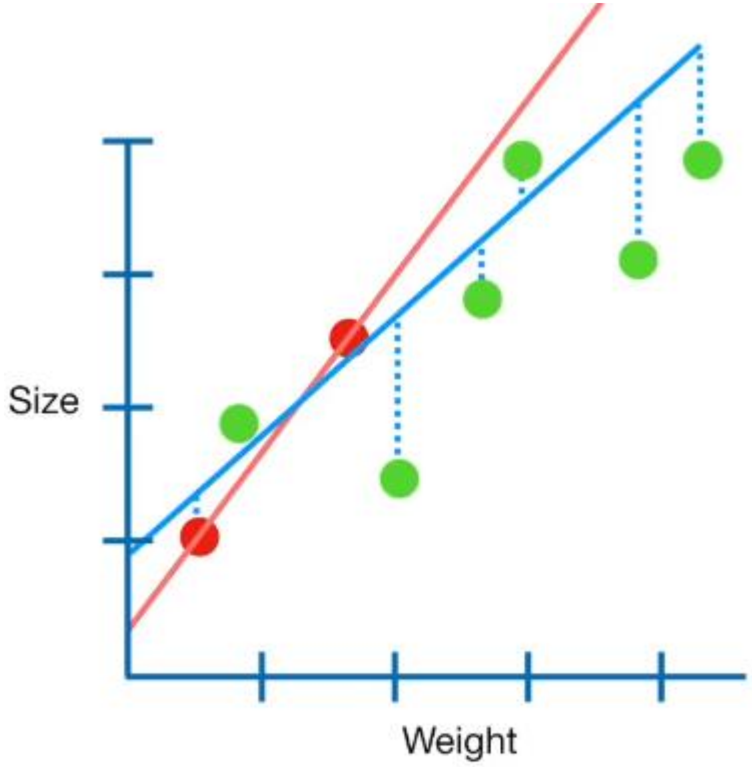


Fig 2. Example Ridge regression [2]

Results: Error Generation and Simulation

Previous setup, Elenas:

- 2016 40 CM optics
- MADX script with older python MADX wrapper
- Error generation Tab 1. and dipole errors according to best knowledge model but not used as input
- Matching tunes

New setup:

- 2023 45CM and 30CM optics
- CPYMAD instead of MADX
- Same error generation
- Matching tunes

Magnet	σ_K / K_1 $[10^{-4}]$	σ_s [mm]
MQ	19	...
MQX	4	6
MQY	11	...
MQM	12	...
MQW	15	...
MQT	75	...

Tab 1. Error Generation parameters obtained from WISE [1] [3]

Results: Error Generation and Simulation

- **70k Samples using 2023 30 CM optics**

  Errors seem to be too big for this optics 15% of twiss failed!

- **70k Samples with 2023 45 CM optics**

  In this case 1-2% of twiss failed, indicating that errors are probably too big for 30CM optics also with the “new” ATS arcs

- **All results shown are for 2023 30 CM optics**

Results: Error Simulation

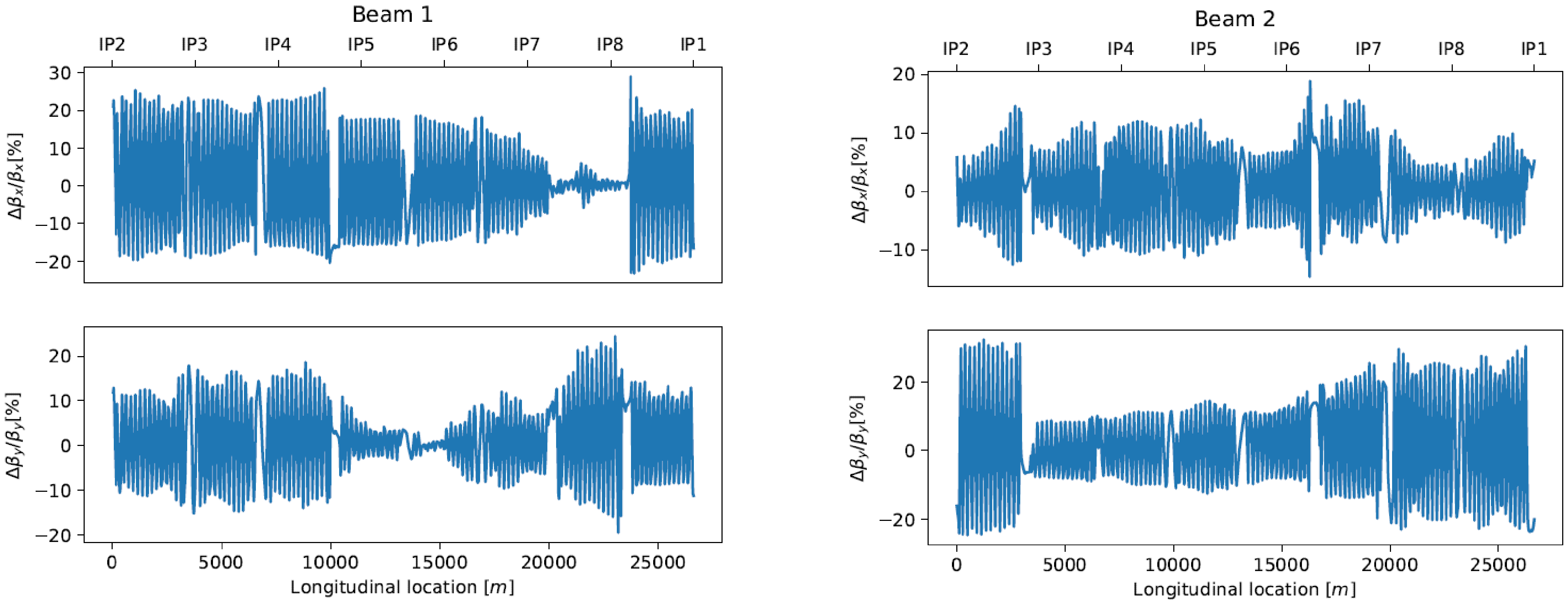


Fig 3. Example Beta Beating

Results: Model Training

Algorithm	Correlation Coefficient: R2	Mean Absolute Error: MAE
Ridge Regression	Train: 0.853 Test: 0.843	Train: 4.67e-06 Test: 4.83e-06
Linear Regression	Train: 0.888 **Test: 0.872**	Train: 3.45e-06 **Test: 3.69e-06**

Decision tree regression: Worse results

For 45CM optics results are similar

$$R^2(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{\mathrm{Var}\{\mathbf{y} - \hat{\mathbf{y}}\}}{\mathrm{Var}\{\mathbf{y}\}}$$

$$\mathrm{MAE}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

Results: Model Training

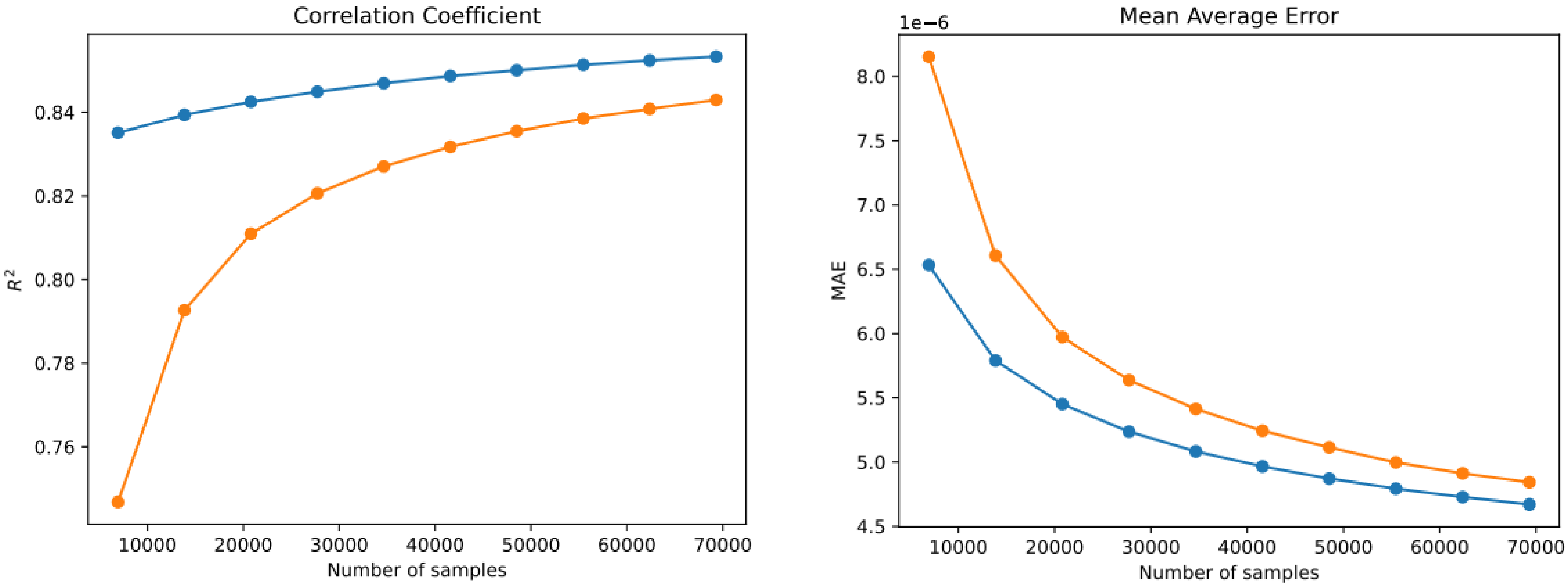


Fig 4. Ridge regression training for different number of samples

Results: Model Training

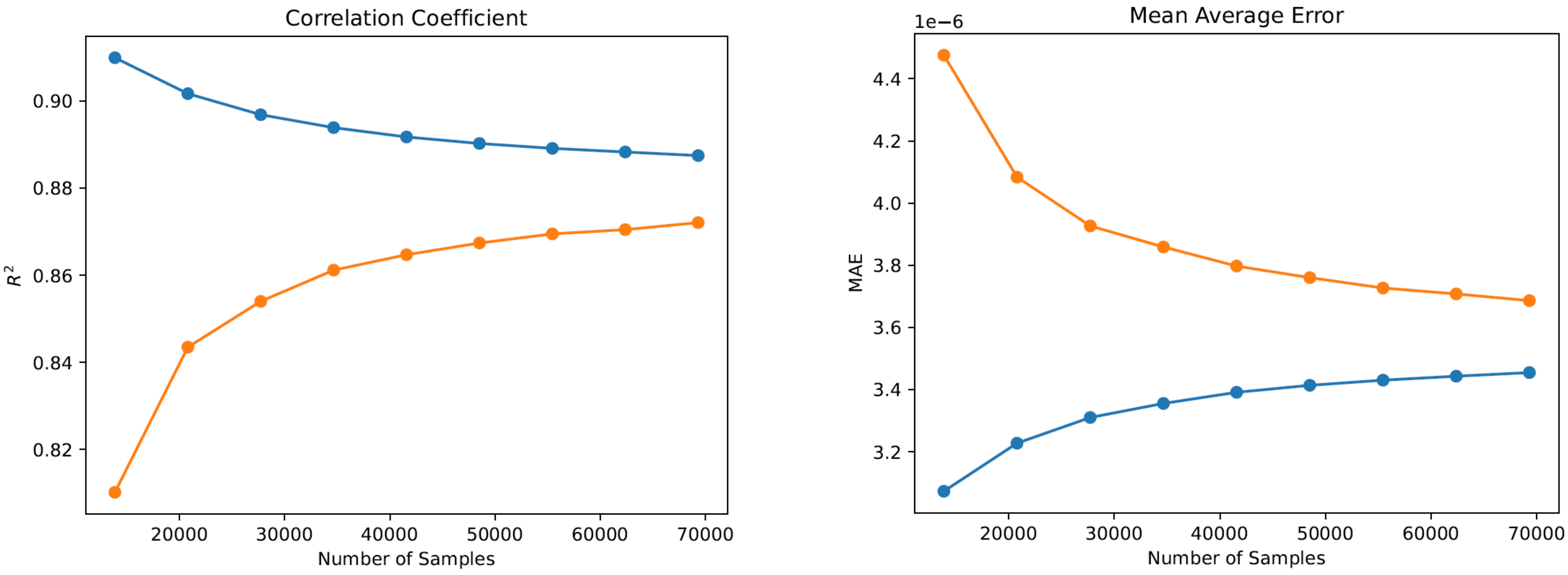


Fig 5. Linear regression training for different number of samples

Results: Example prediction

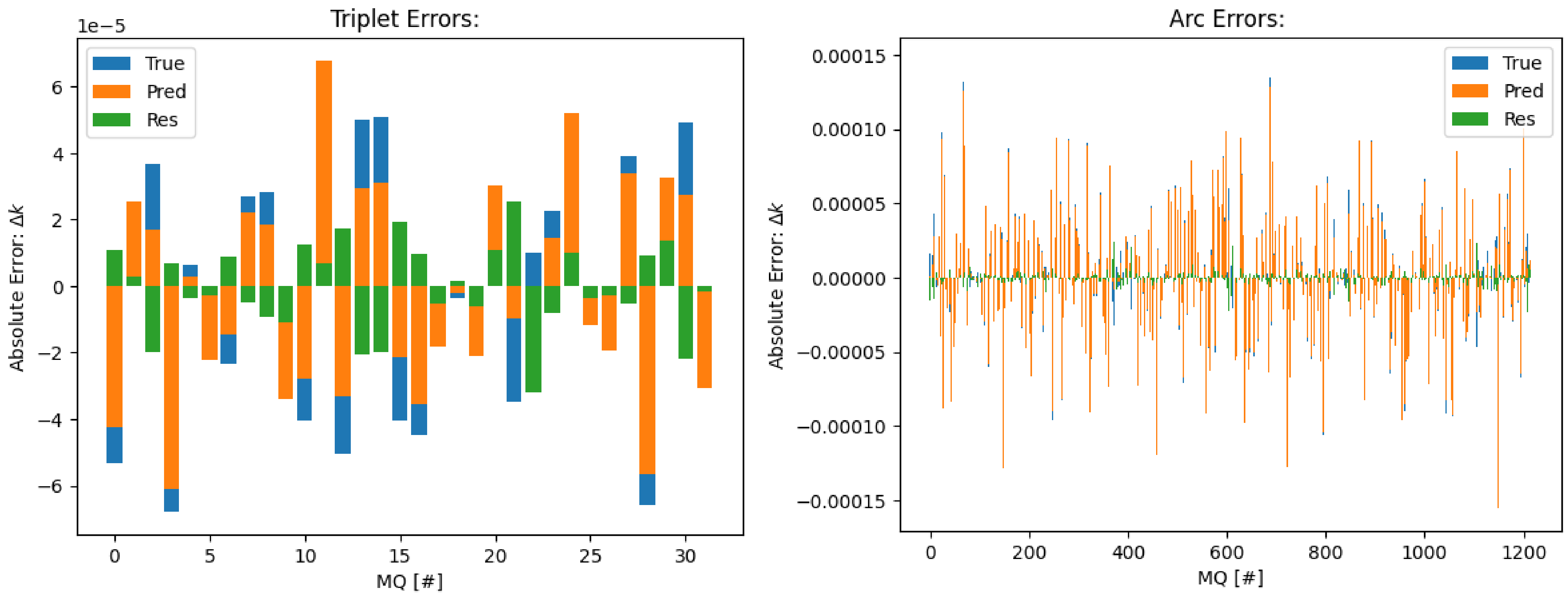
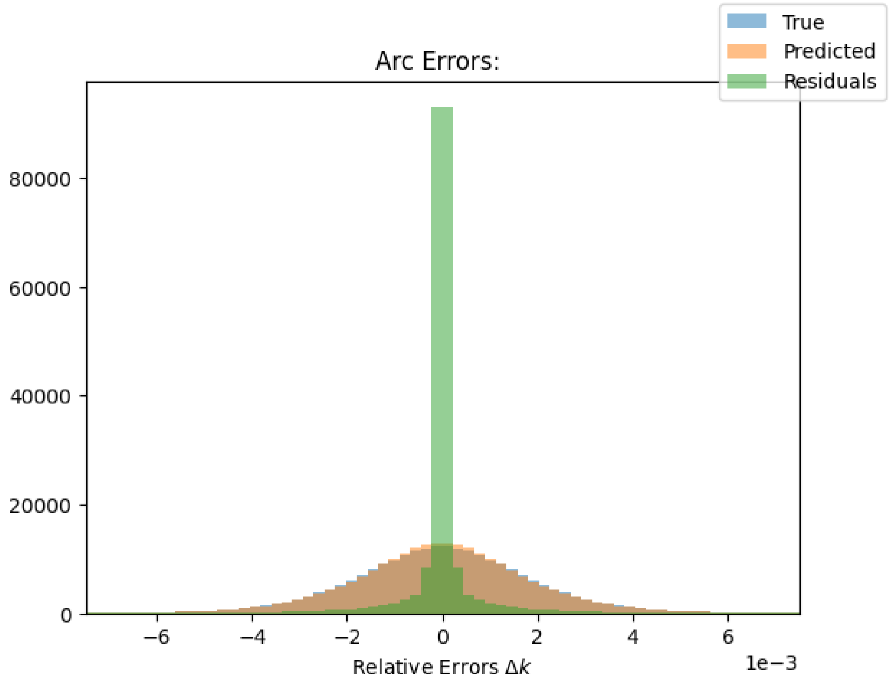


Fig 6. Quadrupole error prediction using linear reg. (random sample)

Results: Error Histogram Prediction



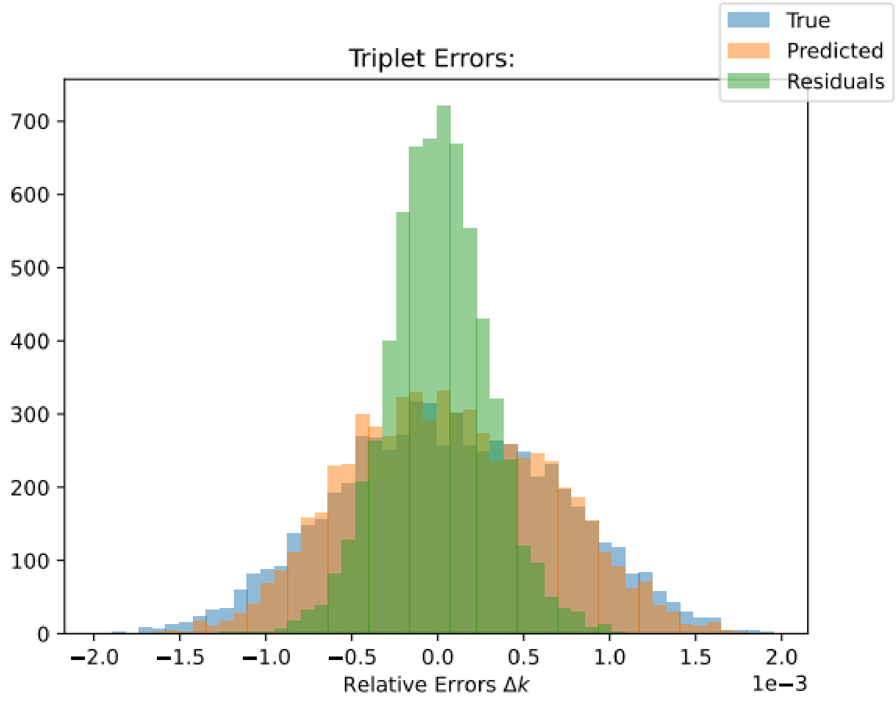


Fig 7. Relative error histograms for 200 test samples

Results: Worse performing samples

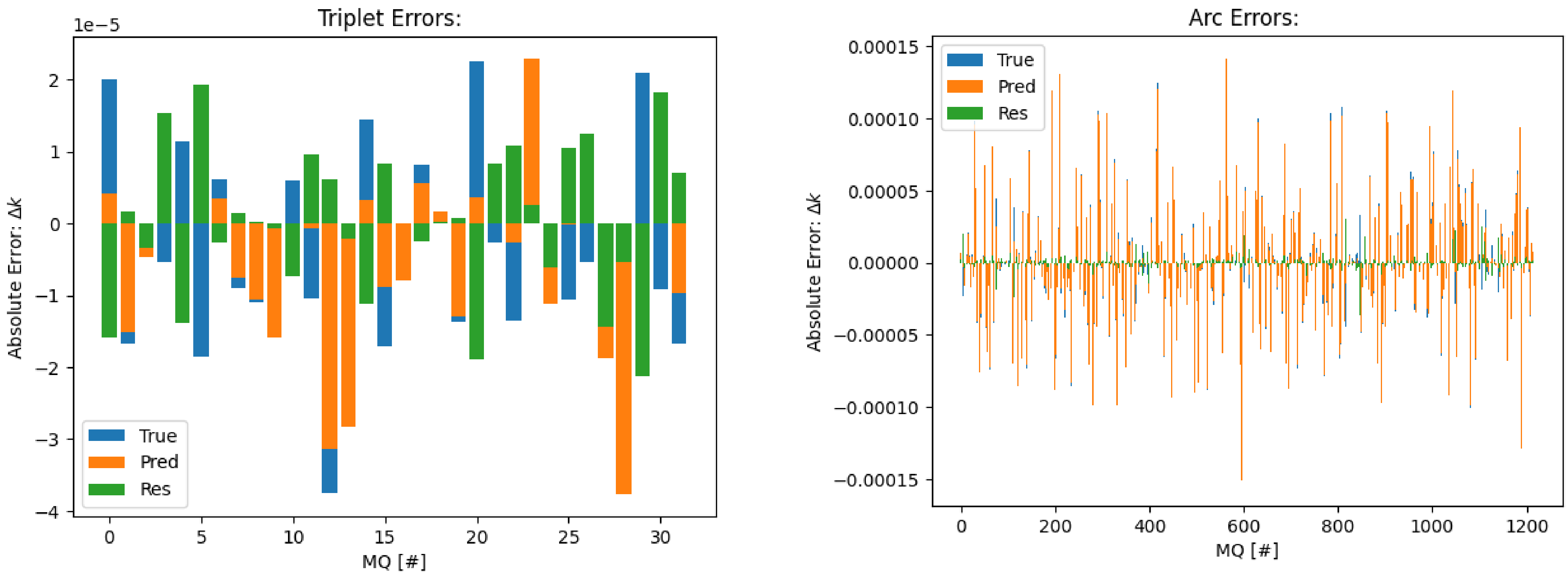


Fig 8. Example of worse performing sample (Triplets!)

Results: Performance for different magnets

Magnets	Correlation Coefficient: R2	Mean Absolute Error: MAE
Triplet	Train: 0.853 Test: 0.843	Train: 4.67e-06 Test: 4.83e-06
Arc + Triplet	Train: 0.888 **Test: 0.872**	Train: 3.45e-06 **Test: 3.69e-06**

Unexpected! R2 slightly worse or similar for triplet errors, expected better R2

MAE Is not a great indicator since error generation is different for triplet and arc

Results: 45CM Optics

Triplet quadrupole performance is worse than in arc. Maybe the errors generated for 40CM in 2016 are too big for 30CM. Testing for 45 CM optics.

Train: R2 = 0.895 MAE = 3.118e-06
Test: R2 = 0.884 MAE = 3.292e-06

R2 Test 0.872 for 30CM

Similar results, hypothesis is wrong.

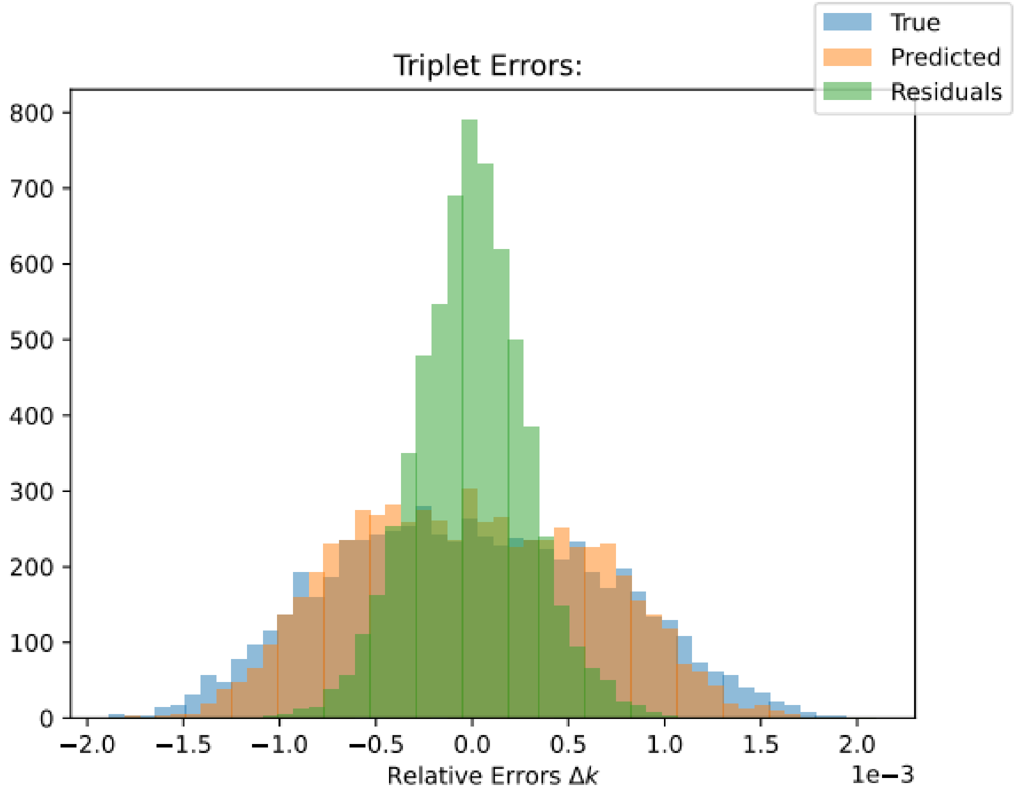


Fig 9. Quadrupole error histograms for 200 test samples

Conclusion

- Improvement in simulation with respect to 2016, maybe because of the decrease in degeneracy in the arc magnets due to change to ATS optics

- Linear regression shows better results than ridge

- Triplet errors prediction is more challenging than arc magnet error prediction in our case obtaining worse samples and worse R2, this might be because of the “new” ATS optics, bigger effect and less degeneracy for arc magnets could explain better performance

- Testing the model on real world data is the next step and most important!

References

- [1] *Fig 1. Data pipeline.* **"Supervised learning-based reconstruction of magnet errors in circular accelerators"** ***by E. Fol****, 2021,* *https://doi.org/10.1140/epjp/s13360-021-01348-5*
- [2] *Fig 2. Example Ridge regression. "***Regularization Part 1: Ridge (L2) Regression***" by J. Stamer, 2018,* *https://www.youtube.com/watch?v=Q81RR3yKn30*
- [3] *Tab 1.* P.Hagen, M.Giovannozzi, J.-P.Koutchouk, T.Risselada, F.Schmidt, E.Todesco, and E.Wildner, **"WISE: A Simulation of the LHC Optics including Magnet Geometrical Data, LHC-Project-Report-1123",** *2008,* https://cds.cern.ch/record/1123714

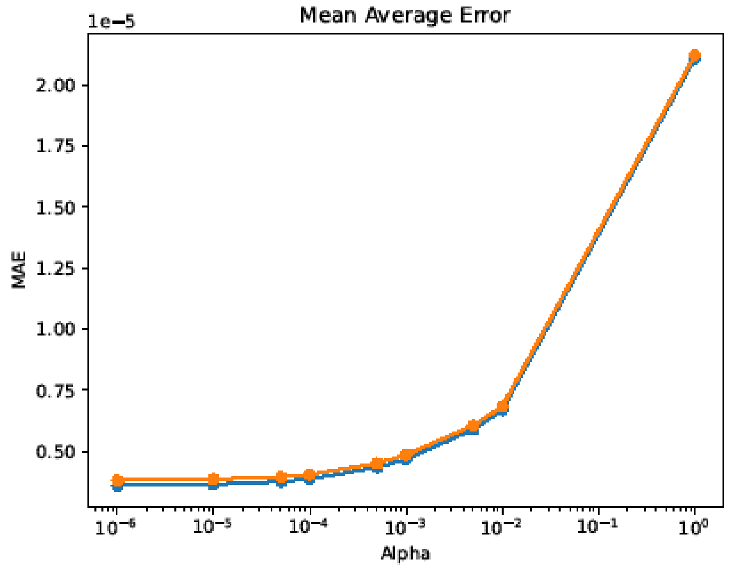
Backup Slides: Error generation macros

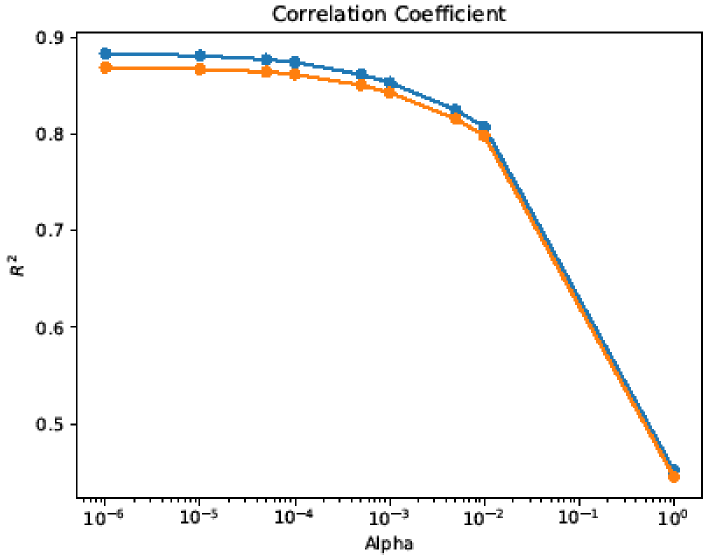
"All quadrupoles in the lattice are assigned a random relative gradient error obtained from uniform distribution with the same rms error σ per magnets family."

Rr = 0.017
GCUTR = 3
B2r = Tab. 1

```
SetEfcomp_Q: macro = {
   Efcomp,  radius = Rr, order= 1,
        dknr:={0,
        1E-4*( B2s  *ON_B2S  + B2r  *ON_B2R  * TGAUSS(GCUTR)),
        1E-4*(-B3s  *ON_B3S  + B3r  *ON_B3R  * TGAUSS(GCUTR)),
        1E-4*( B4s  *ON_B4S  + B4r  *ON_B4R  * TGAUSS(GCUTR)),
        1E-4*(-B5s  *ON_B5S  + B5r  *ON_B5R  * TGAUSS(GCUTR)),
        1E-4*( B6s  *ON_B6S  + B6r  *ON_B6R  * TGAUSS(GCUTR)),
        1E-4*(-B7s  *ON_B7S  + B7r  *ON_B7R  * TGAUSS(GCUTR)),
        1E-4*( B8s  *ON_B8S  + B8r  *ON_B8R  * TGAUSS(GCUTR)),
        1E-4*(-B9s  *ON_B9S  + B9r  *ON_B9R  * TGAUSS(GCUTR)),
        1E-4*( B10s *ON_B10S + B10r *ON_B10R * TGAUSS(GCUTR)),
        1E-4*(-B11s *ON_B11S + B11r *ON_B11R * TGAUSS(GCUTR))},
        dksr:={0,
        1E-4*(-A2s  *ON_A2S  + A2r  *ON_A2R  * TGAUSS(GCUTR)),
        1E-4*( A3s  *ON_A3S  + A3r  *ON_A3R  * TGAUSS(GCUTR)),
        1E-4*(-A4s  *ON_A4S  + A4r  *ON_A4R  * TGAUSS(GCUTR)),
        1E-4*( A5s  *ON_A5S  + A5r  *ON_A5R  * TGAUSS(GCUTR)),
        1E-4*(-A6s  *ON_A6S  + A6r  *ON_A6R  * TGAUSS(GCUTR)),
        1E-4*( A7s  *ON_A7S  + A7r  *ON_A7R  * TGAUSS(GCUTR)),
        1E-4*(-A8s  *ON_A8S  + A8r  *ON_A8R  * TGAUSS(GCUTR)),
        1E-4*( A9s  *ON_A9S  + A9r  *ON_A9R  * TGAUSS(GCUTR)),
        1E-4*(-A10s *ON_A10S + A10r *ON_A10R * TGAUSS(GCUTR)),
        1E-4*( A11s *ON_A11S + A11r *ON_A11R * TGAUSS(GCUTR))};
        }
```

Backup Slides: Tuning for Ridge reg





CERN