­02/02/2023

**Paper:** [**https://doi.org/10.1140/epjp/s13360-021-01348-5**](https://doi.org/10.1140/epjp/s13360-021-01348-5)

Problem, usually correction is made few times a year, multiple magnets at a time. Goal, predict individual magnet errors as misalignment, sextupoles…

**INPUT:** deviation of the optic measurement from design \Delta(x)

**OUTPUT:** In this case, effective quadrupole field errors

source of the problem (change in intensity of field? position of magnet? Intensity of current of electromagnet?)

* Correction of the problem compensating predicted errors

How does mad-X work

**Pregunta**: The paper does not talk about the correlation between change in field and an applicable correction ie repositioning of magnet, intensity change

03/02/2023

**Paper: Thesis Tobias Persson**

Summary on beam optics theory.

Summary on beam measurements.

* Exciting the beam
* Phase
* beta
* K modulation
* Dispersion

Summary on beam corrections and procedures.

**Expected workflow:**

1 Generation of possible magnet errors using MAD-X (ARTIFICIAL Y)

2 Simulation using MAD-X => OPTIC MEASUREMENTS (X)

3 Algorithm training and selection Y\_pred = f(X)

4 Validation using model data and new EXPERIMENTAL DATA

Y => Deviation in magnetic field

X => beta(?)

Today I read and understood most of the theoretical part of the thesis, set up my github and played with MADX simulation data an OMC3 software.

06/02/2023

Ideas:

1. Non-linear optic errors

2. Measurement=>optic functions

08/02/2023

Running first MADX script.

Dont request many files from afs => DDOS.

Understand how to generate data from MADX.

Reading documentation for OMC python package

09/02/2023

This project is highly dependant on accelerator physics knowledge. I have to study and understand the problem. Lots of reading.

Trying to read .npy data from Elena does not work. Trying to run her script doesnt work either because of library dependencies.

13/02/2023

Library dependencies solved, VS code SSH setup, trying to understand elenas python script and MADX.

**Questions for Elena:**

1. **Data shape. 10 simulations (different seeds for the random distributions of errors I guess), 610 Measurements of (beta, mu, n) 8 possible arc magnet errors 2 possible sextupole errors for two beams (1,2) and two planes (x, y)**
2. **The data is for one turn or multiple turns, does this matter?**
3. **Weird learning curves, is this normal? (Not important)**

14/02/2023

**beta\*=> IMPORTANT PARAMETER** for focusing near triplet magnets, (collision)

**mu =>** Allows us to calculate actual beta parameter

**n =>** dispersion parameter (beam quality)

**Triplet error =>** Error in near colision triplet magnets (independent)

**Arc error =>** Error in circuit of magnets

**MQT error =>** ??

**Misaligment errors =>** Cant be fixed simulated to make data more realistic

**Possible improvements:** If you take your data as a matrix instead of a vector you have additional information on the mixed effects of magnets therefore deep learning with a multidimensional input would be interesting this is something I know how to do and would not be extra hard, maybe I have more idea than elena in this topic and can improve.

If everything is a lineal relationship with noise then it does not make much sense, I would like to know how important are the non linearities and how we can model then in MADX this is a field where ML can improve the existing tech.

**Possible improvements:**

* **CPYMAD allows for simulation and processing at the same time, no TWISS FILE**
* **Non linearities => Deep learning and comparing with Linear optics simulations**
* **FPD Metric for model performance (ML Group meeting) => K1L K2L K3L…**
* **Up to date optics, models, errors(?)...**

Opening the original .npy can be useful, I dont have to simulate everything

16/02/2023

Three files for sim errors =>

* Before matching
* After matching
* Common errors

How to obtain this measurements using cpymad instead of older madx wrapper

20/02/2023

Translation to cpymad complete, only for triplet errors. Still saving files because you need to read the errors from B1 when using B2. Not reading from python. Program still crashing and slow as hell. **Expected time to generate 10K samples, 7 hours aprox, manageable with 5 parallel simulations.** Trying to manage crashing, and do sanity checks!!!! Try higher number of processes as possible,

21/02/2023

Computer crash: Computer crashes after 5 iterations even without parallel computing.

PROBLEM: **SWAPING**, memory gets filled and everything stats running slow until it crashes

SOLUTION: Optimize, increase performance… All of them

10K samples => 138 hours, without parallel simulations, not manageable.

Optimization, what parameters get saved????

Debugging script and watch what variables get saved.

23/02/2023

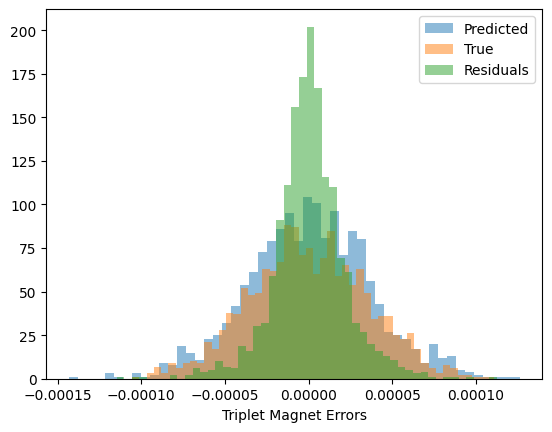
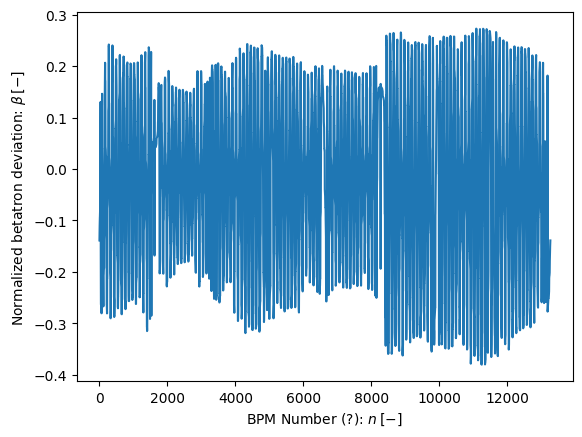
**State of the project:**

1. **Coding**
   1. Running Elenas code on updated packages: Done
   2. Changing from omc madx wrapper to cpymad: Done
   3. Memory allocation issues: Fixed
   4. Obtaining more data points: Pending
   5. cpymad running without saving files: Pending
   6. Expanding to all errors: Pending
   7. Running simulation on cs-ccr-dev3 server: Pending
2. **Optics, sanity checks**
   1. Comparing optics saved by python script with the saved files: Done
   2. Plotting relative beta ideal optics w.r.t. B2: Done
   3. Plotting error histograms for more data points: Done

Results, changing from the omc3 madx wrapper to cpymad has allowed me to learn the basics of madx, understand elenas code better and will allow for faster simulation times. Updating the code not to save files has proven being harder than expected, thinking of running everything from python, to make a cleaner code, however since running times are not an issue right now I will focus elsewhere. Right now I think it would be best to focus on the optics and theoretical part of the project, understand the simulations and the data better ie matching, errors, madx saved files, sanity checks...

RAM problem: created a madx instance for each iteration => RAM explodes even if deleting this instance 14 s/data point => 20-30 Hours per 10k points not ideal 17,8 s/data point

Plotting relative beta seems fine!

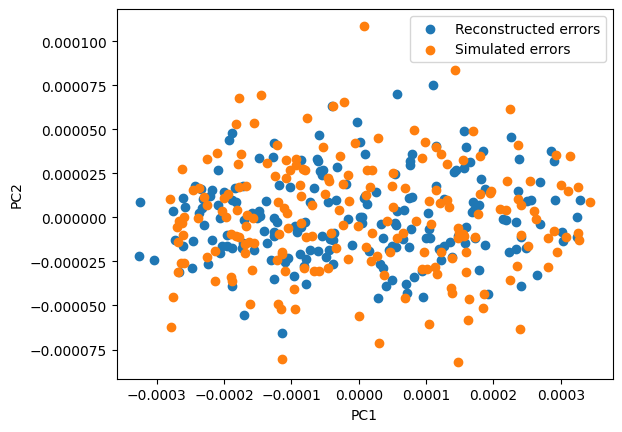
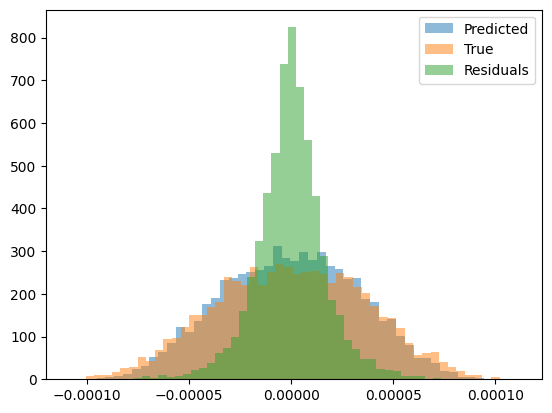


27/02/2023

No meeting, creating more data points. Trying to fix remote teleworking.

28/02/2023

200 Data points no crash, will leave simulation running on server today

03/03/2023

Creating only one madx instance and multiprocessing is really hard to implement. Going to run different instances of the program. Anyway it is faster to create multiple instances. Not going to focus in this really.

1. **Coding**
   1. cpymad running without saving files: Pending
   2. Expanding to all errors: Pending
   3. Running simulation on cs-ccr-dev3 server: Pending
   4. Running parallel code with only one instance: Pending
   5. Updating to 2020 optic settings etc: Pending
   6. Using pymadng(?): Ask Felix
2. **Optics**
   1. Reading possible problems and application: Pending

06/03/2023

Trouble connecting to remote, possible issues.

System suspension of CERN desktop due to memory issues (Most probable), change in IP address (Maybe), github ssh permission expired (I don’t think so)

Possible solutions:

Remote desktop Ubuntu-windows. Familiar.

VS Code remote development. Issues connecting.

Installing Ubuntu partition, bad idea

Using swan, easy to use different servers

TO-DO: Check if I can connect before and after suspension!

07/03/2023

Reading about MADNG: Questions, non linearities are a problem most of the time or only when the beam enters in an unpredictable caotic motion?

Reading about non linear optics:

Implemented cpymad in python class code:

13/03/2023

1. **Coding**
   1. cpymad running without saving files: Done
   2. Expanding to all errors: Pending
   3. Running simulation on cs-ccr-dev3 server: Done
   4. Running parallel code with only one instance, running on multiple instances: Problem (madx crashes when using pool)
   5. Updating to 2020 optic settings etc: Pending
2. **Optics**
   1. **Using pymadng(?): Ask Felix, hard to use**
   2. Reading possible problems and application: Pending

14/03/2023

Paralellisation done, time of execution 433 s for 10 scripts simulation 10 samples each. => 4.33 S per sample => 100 K samples is 5 days,Probably less since the longer the script runs the better.

Lets try with more scripts and see how it performes. Make sure that the data does not get overwritten.With 20 simulations 472 s => 2.36 S per sample => 66 Hours

|  |  |  |
| --- | --- | --- |
| N\_Parallel | Time Per sample | Time 100 K samples (extrapolated) |
| 10 | 4.33 s | 120 h |
| 20 | 2.36 s | 66 h |
| 30 |  |  |