­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

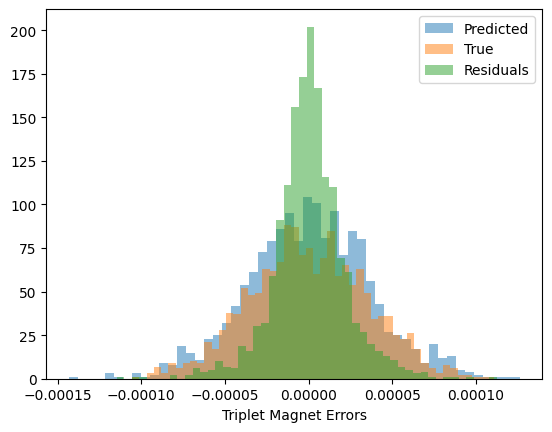
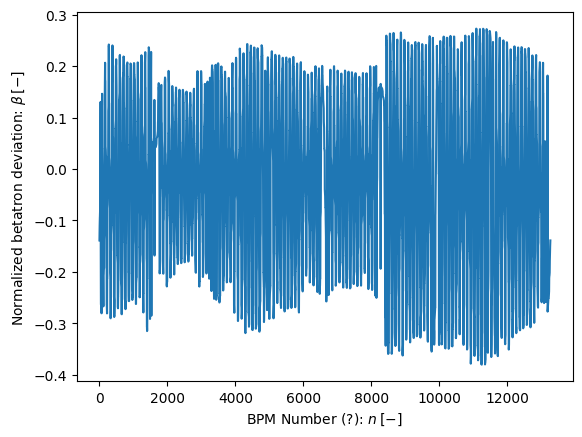
**State of the project:**

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!

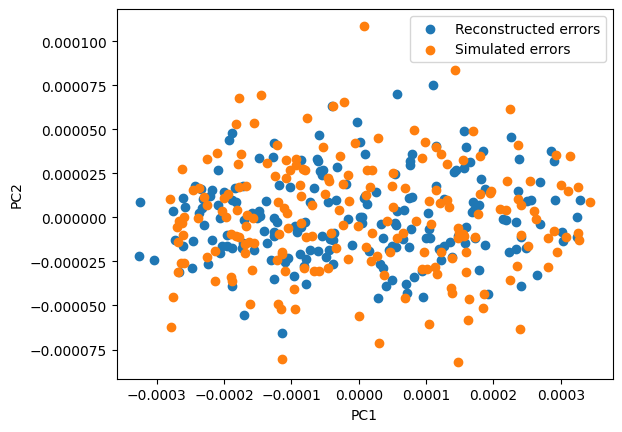
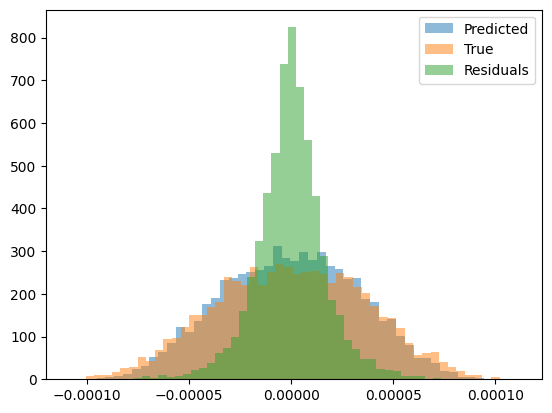


27/02/2023

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 | 1.48 s | 41 h |
| 80 | 1.4 s | 39 h |

1. **Coding**
   1. Extending for all previous magnets: Pending
   2. Updating to 2023 optic settings for 30 and 60 CM: Pending
   3. Simulation 100k data points: Pending
2. **Optics**
   1. Reread Elenas paper: Pending

16/03/2023

Fixing SSH problems definitively.

1. **Coding**
   1. Extending for all previous magnets: Done
   2. Updating to 2023 optic settings for 30 and 60 CM: Pending
   3. Simulation 100k data points: Pending
2. **Optics**
   1. Reread Elenas paper: Pending

21/03/2023

1. **Coding**
   1. Extending for all previous magnets: Done
   2. Updating to 2023 optic settings for 30 and 60 CM: Done
   3. Simulation 100k data points: Pending
2. **Optics**
   1. Reread Elenas paper: Pending

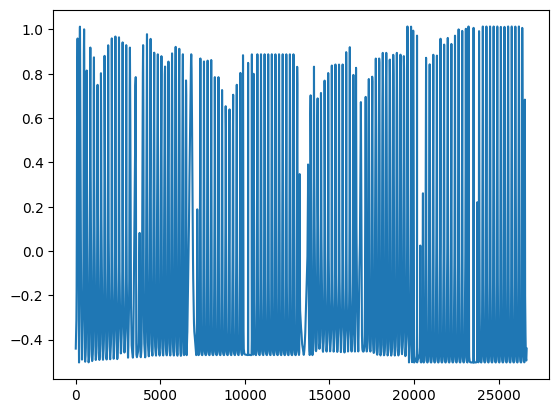
Still Have to reorganize and clean all the code.

27/03/2023

Lost some journal days. Results of simulation, bug fixing. Decision trees perform worse than ridge. Try svm.

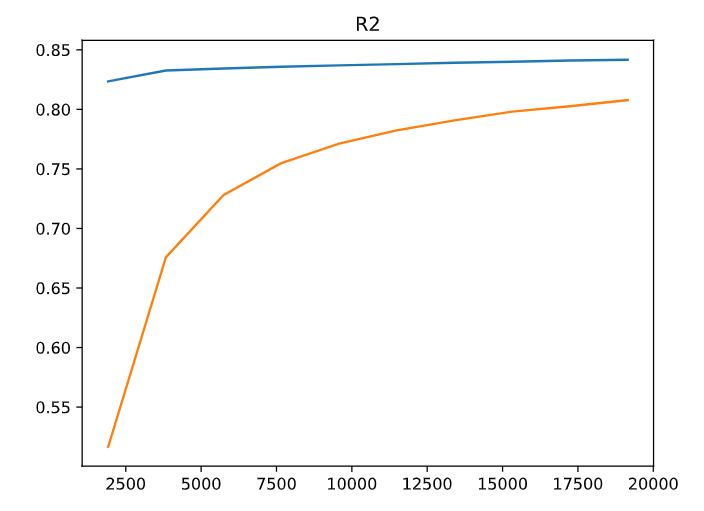
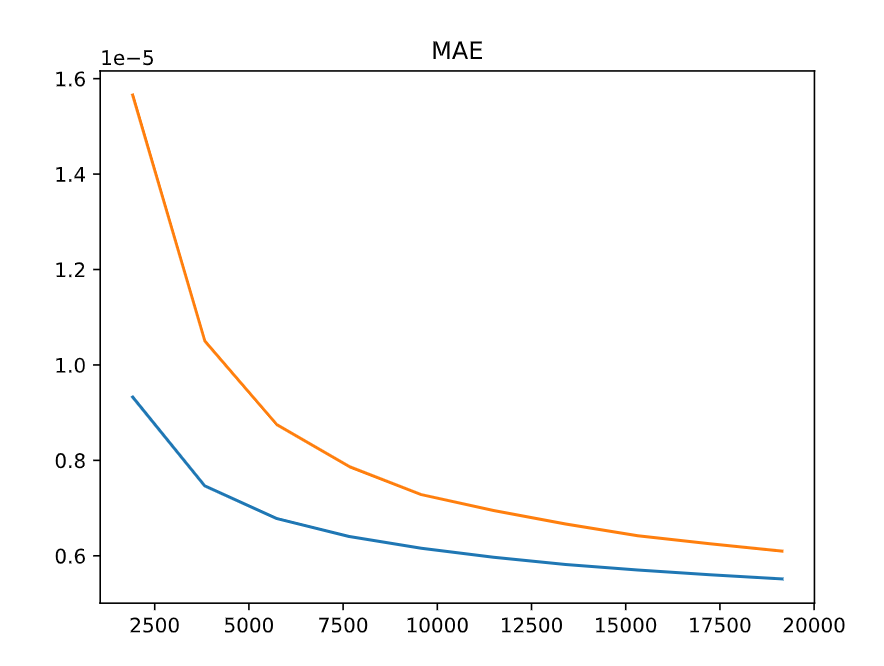
04/04/2023

Comparing before and after matching twiss



11/04/2023

Making a function to save predicted errors in a .tfs file. Played with pymadng, but first I want real results using madx. Training for 20K samples. Trying to use svr, no way.



Reasons why the sample generation seems to be failing.

* R2 too good for this amount of samples.
* 10-15% Of simulations fail
* Sometimes the MQT knob thing gives wrong outputs

All of this things can make sense.

12/04/2023

Presentation in 2 weeks.

**- Visualizaciones de la simulacion**

- Errores predecidos vs simulados para cada iman.

- Errores simulados vs usados en el algoritmo para comprobar que la simulacion esta bien y no es demasiado simple errores normalizados

**- Testing the model**

- Predecir twiss y meter las correciones de los datos reales y ver si funcionan

- Con los errores generados predecir twiss y comparar con el medido. Problema, puede generarse con distintos errores.

**- Problemas**

- R2 demasiado bueno, quizas pq el modelo es demasiado sencillo. Quizas porque ha disminuido la degenerancia del LHC.

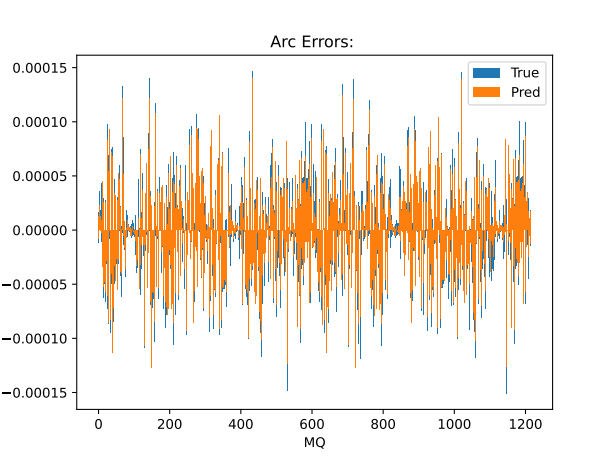
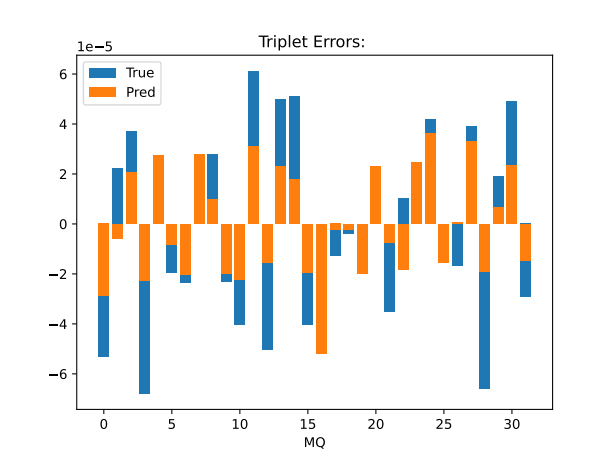
- 15% de las simulaciones fallan, quizas por usar opticas de 30cm y un beta mayor en los arcos

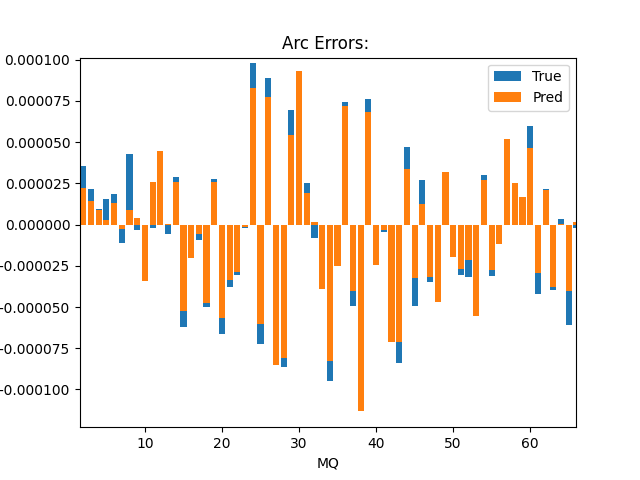
**- Ideas**

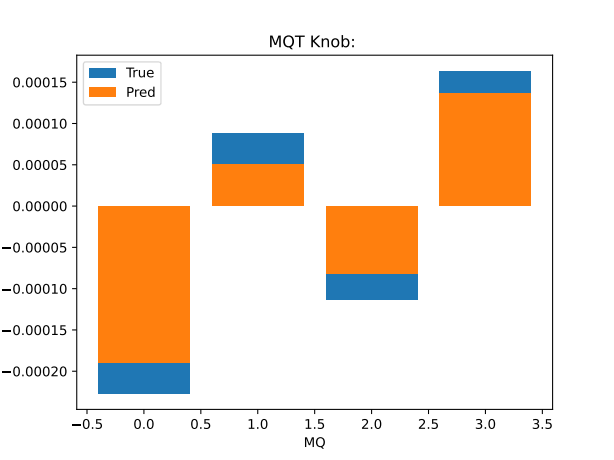
- El test con datos reales no parece optimo, quizas se podria hacer alguna forma de testearlo mas sistematicamente, predecir las correciones directamente

- madng, using gradients for training

- si estamos generando errores como se hacia en 2016, eso es lo mismo para los nuevos arcos??? quizas ha cambiado la tolerancia, ya que ha cambiado la beta

Simulated vs predicted errors for random sample:





At first glance the model performs worse for the triplet errors, this is not expected, maybe it is because the curse of dimensionality. A better way to visualize this is with the correlation plots by Elena. The simulated errors and the ones used for training seem to be the same. PROBABLY NEED MORE DATA!!!!

Testing in a random dataset, maybe not the best practice but still expected lower.

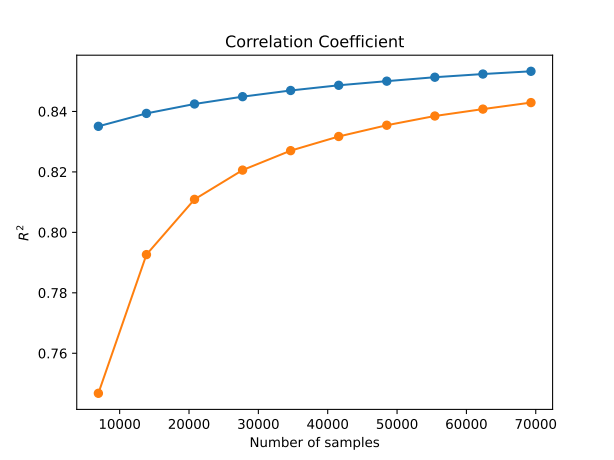
Train R2: 0.779 Test R2: 0.788

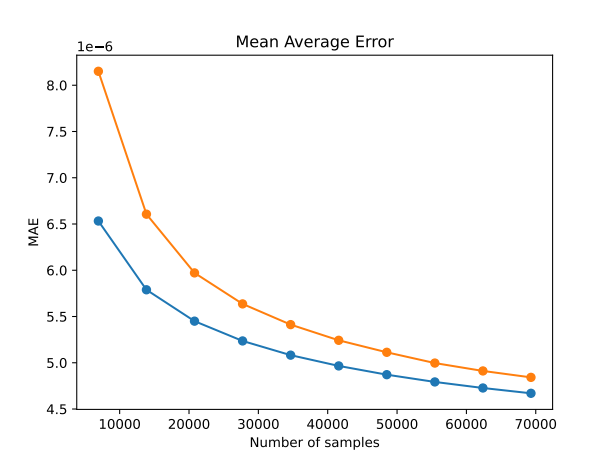
Train MAE: 8.41e-06 Test R2: 8.58e-06

Tomorrow train with more data, next days plot different results. Hope for the best about the shift, as long as its not on friday working hours I am good, lets hope.

13/04/2023

Training with 70 k samples 80/20 split for test, results are still improving, will make plots to understand the results better.





Figures I need to add to the presentation:

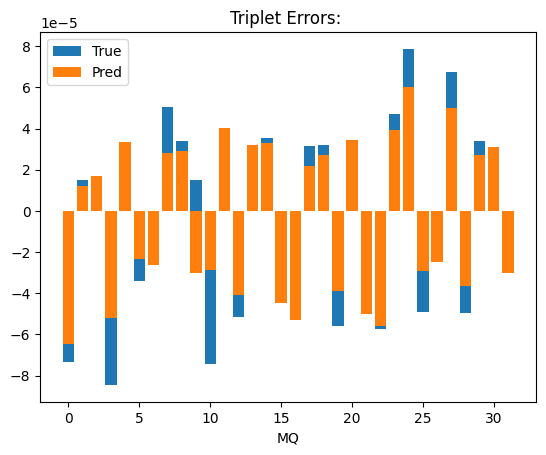
* Data pipeline, Done
* MAE and R2, Done
* Errors generated VS predicted histogram for different magnets, Done
* Twiss, Done
* Backup: Some error comparison and performance for different types of errors

Be careful if figures are obtained from test or train data, big difference.

The presentation should have scientific type communication and peer to peer communication. How its actually going etc.

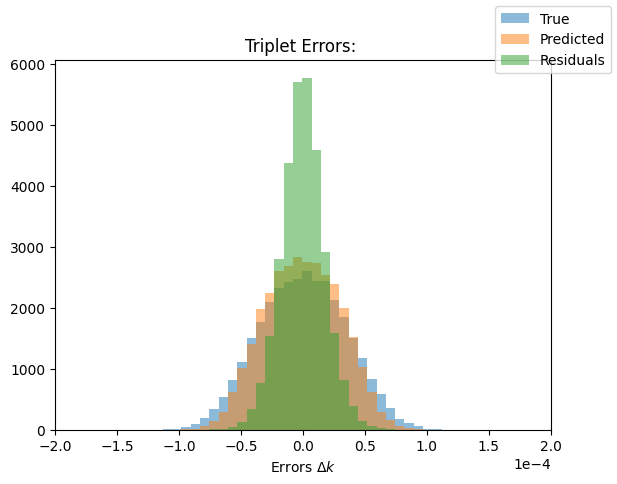
14/04/2023

Other example triplet error. This plots should take into account test vs train splits. In this example performance seems to be better, however maybe there is still worse performance on triplets.



14/04/2023

Understimation of triplet errrors especially, This might be because of ridge regression. As seen like in the next figure, still, results dont make as much sense as I would like them



IDEA

* Trying using linear regression instead of ridge.

Training: R2 = 0.8875, MAE = 3.45e-06

Test: R2 = 0.8715, MAE = 3.69e-06

Better results overall, compared to the 0.85 using ridge regression.

It seems like the understimation of the errors comes from using ridge regression.

**This last weeks summary:**

* Plotting errors, systematic understimation
* Beginning presentation
* Trying normal linear regression instead of ridge => No systematic error
* **RESULT: The systematic errors in elenas paper probably come from using ridge, ridge penalizes strong dependencies therefore making an estimator more robust to variance**
* Plotting example twiss! Some twiss are not centered in cero.

PLOT ONLY FOR TEST DATA. CHECK WEIRD TWISS

18/04/2023

RESULTS FOR RIDGE AND DIFFERENCE BETWEEN TRIPLET AND ARC PERFORMANCE

Training Triplet: R2 = 0.840, MAE = 4.752-06

Test Triplet: R2 = 0.836, MAE = 4.763e-06

Training: R2 = 0.853, MAE = 4.67e-06

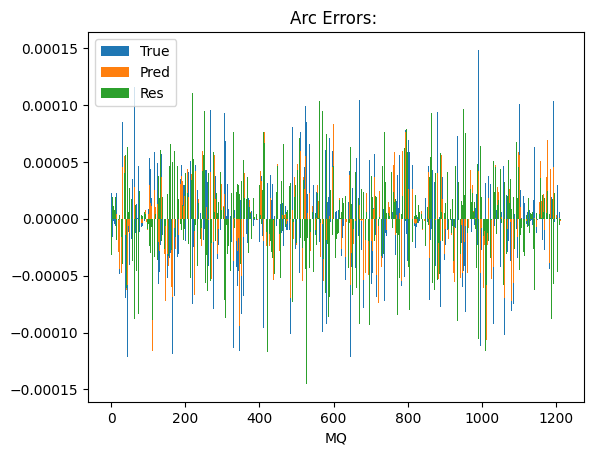
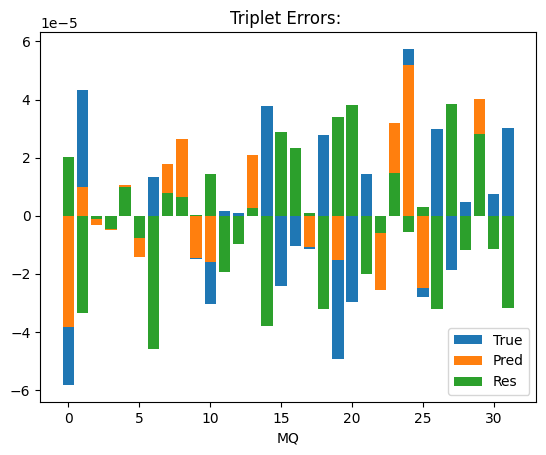
Test: R2 = 0.8430, MAE = 4.83e-06

Generated all figures for test data, obtaining BIG residuals according to histograms, maybe histograms are more centered!! In addition to the weird twiss data.

19/04/2023

RETRAIN AND TEST IN TEST DATA

Some results are much much worse, still for most samples the results are great, t**his might explain the thick histograms**. This means some samples are not being generalized by the model example sample 16. R2 and mae is great either way. Are these outliers??? **TRY USING RANSAC**



The fact that this works so well for some samples but others horribly suggest that some simulations enter a non stability region and are much harder to predict than others.

Decission trees work worse still.

21/04/23

With 45 cm optics the errors are still in the order of 1E-3

24/04/23

LINEAR 45 CM

Training: R2 = 0.8637265532726398, MAE = 4.13665951038814e-06

Test: R2 = 0.8562377544789292, MAE = 4.250593427395733e-06

RIDGE 45 CM

???

28/04/23

Ideas after presentation

1. Adding noise Done
2. Plotting beta beating histogram and training with more realistic data Done
3. Testing degeneracy training with and without increase in beta for ATS arcs Pending
4. Trying to predict \Delta p / p Chromaticity Pending
5. Recreate optics Pending
6. Introduce 2023 corrections Pending

IDEA!

Triplet performance is due to missing noise! Studying performance for different levels of noise! The parameters have not chaged!!!!

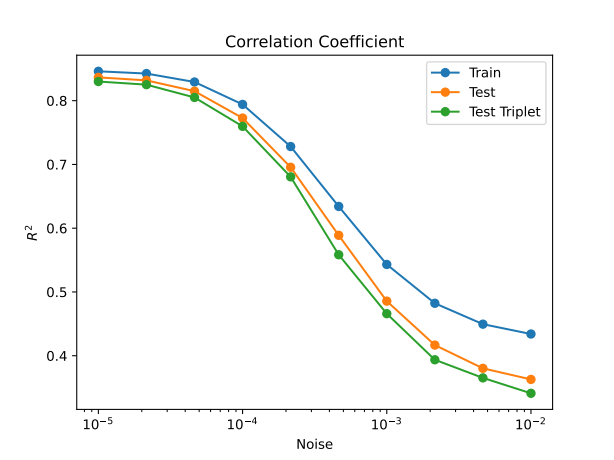
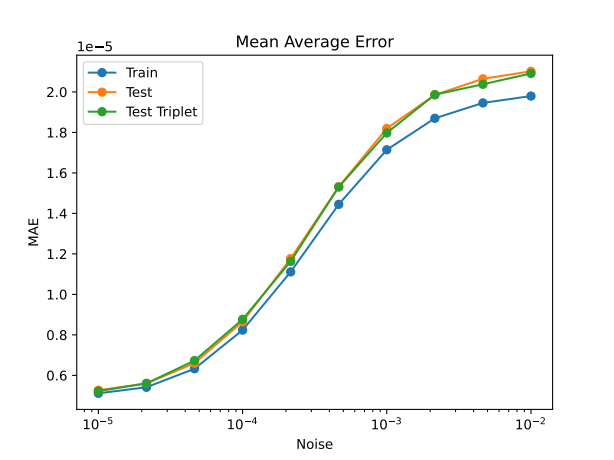
02/05/2023

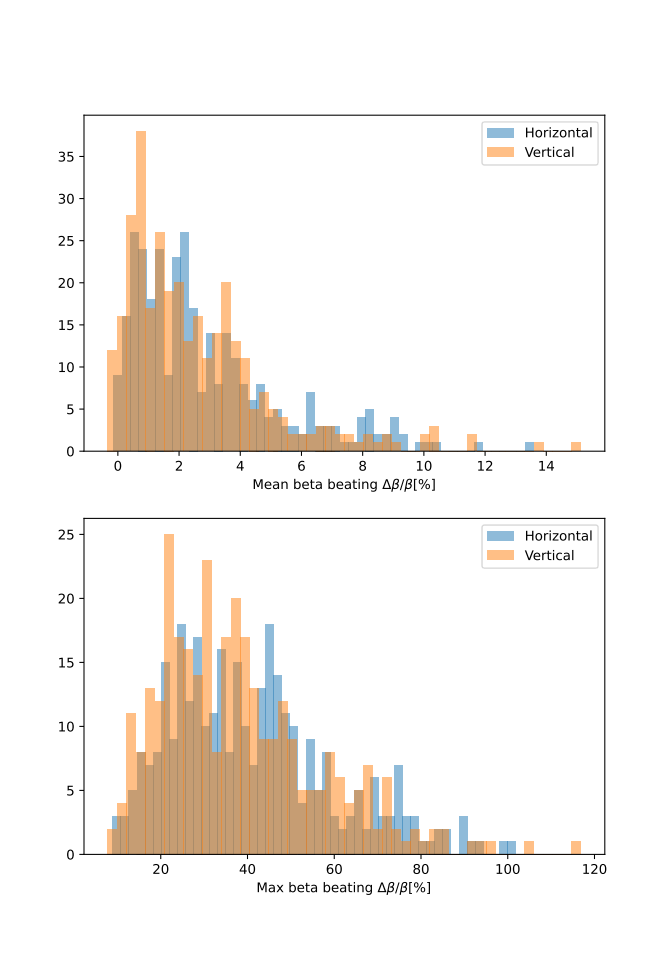
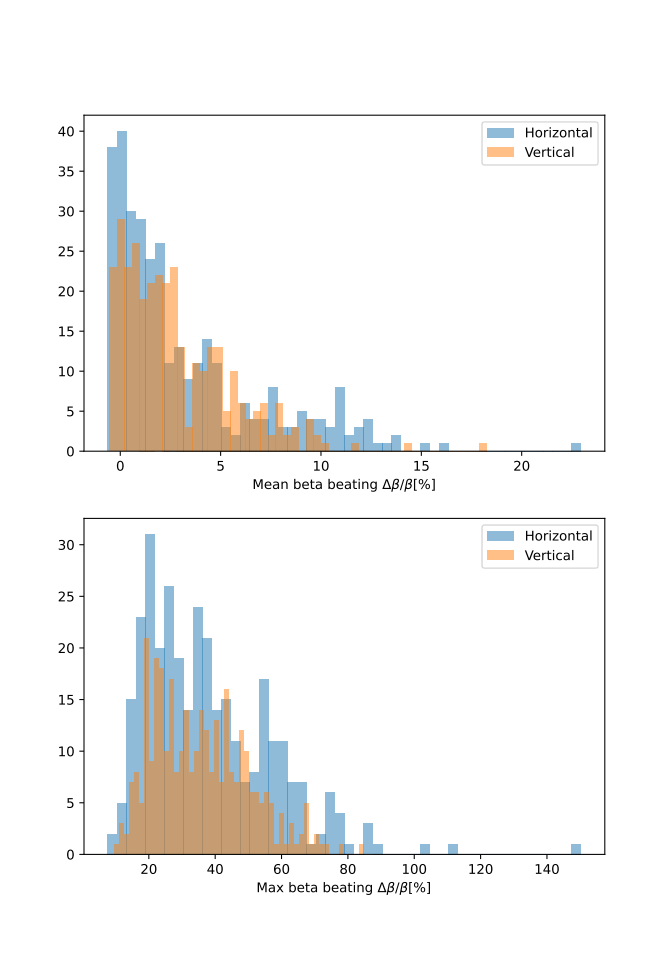
Data generation with a 30% cutoff

Effect of noise in ridge regression, used noise level is 10E-3 according to Elena.

Recreating optics is nearly complete!!!

e simulated normalized dispersion deviations,





03/05/2023

Optic recreation completed, still have to compare with a better example and with the original true errors.

**Results 30cm, ridge, 1e-3 noise for phase and disp**

Train Triplet: R2 = 0.5117993670334481, MAE = 1.7202585891941248e-05

Test Triplet: R2 = 0.4633846137835224, MAE = 1.8070333017981448e-05

Train: R2 = 0.5430435119648702, MAE = 1.7143778452768323e-05

Test: R2 = 0.4857963488172039, MAE = 1.8183475904515684e-05

**Results 30cm+bb30, ridge 10e-3 noise for phase and disp**

Train Triplet: R2 = 0.4832839275335945, MAE = 1.7812030836917395e-05

Test Triplet: R2 = 0.46073526707691526, MAE = 1.8195960713492678e-05

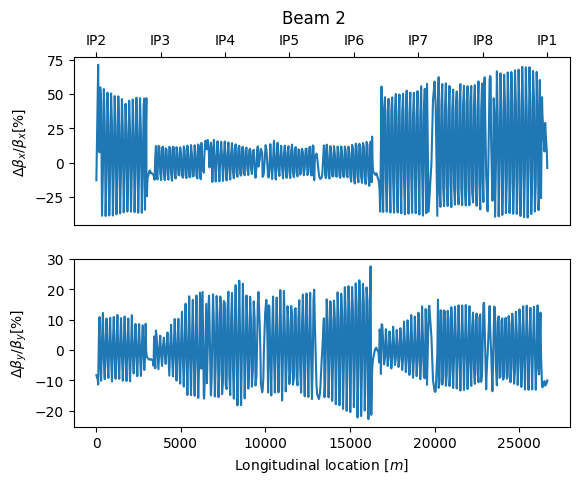
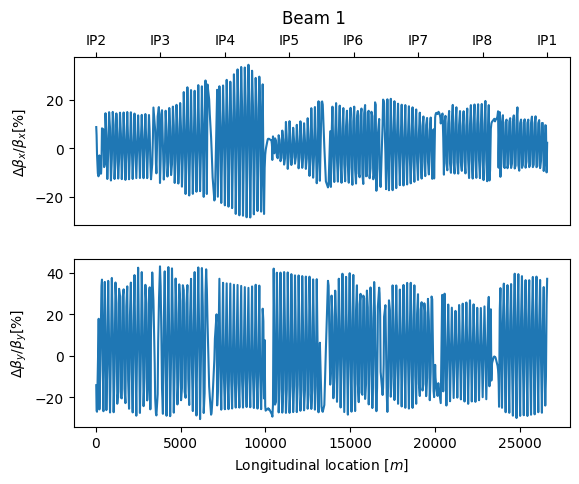
Train: R2 = 0.5216334719843492, MAE = 1.767973943729175e-05

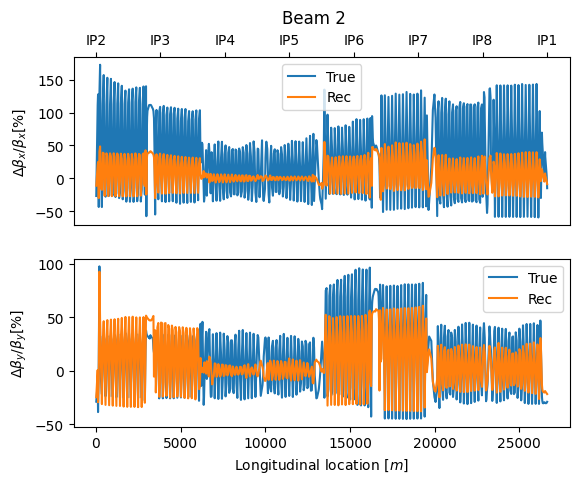
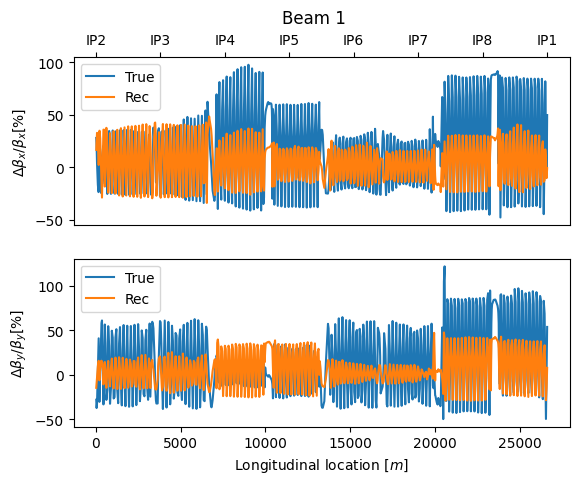
Test: R2 = 0.48848438816963746, MAE = 1.8284178935036706e-05

Worse results are due to the fact that the noise figures for Elena are different studies that the one we are using.

Ideas after presentation

1. Adding noise Done
2. Plotting beta beating histogram and training with more realistic data Done
3. Testing degeneracy training with and without increase in beta for ATS arcs Pending
4. Trying to predict \Delta p / p Chromaticity Pending
5. Recreate optics Done
6. Introduce 2023 corrections Pending

**Recreating optics sanity check:**

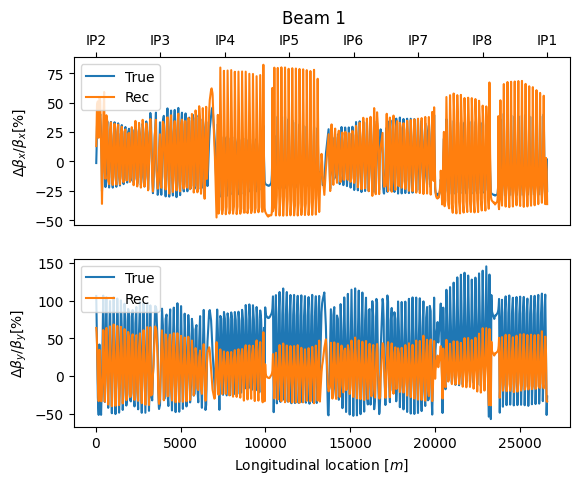
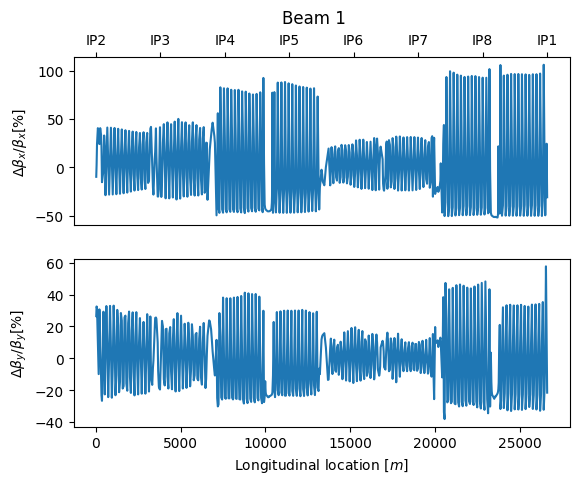
****

Something is completely wrong, a bug. The true betabeat should be the same!

- Not the noise

- Not the matching

Bug found, WRONG NOMINAL (using 45cm instead of 30cm) for BB30 data, testing with correct nominal, still no improvement

This is without adding the dipole errors in the reconstruction and without tuning an dipole

The problem is for the true errors not the predicted. Maybe mqts?

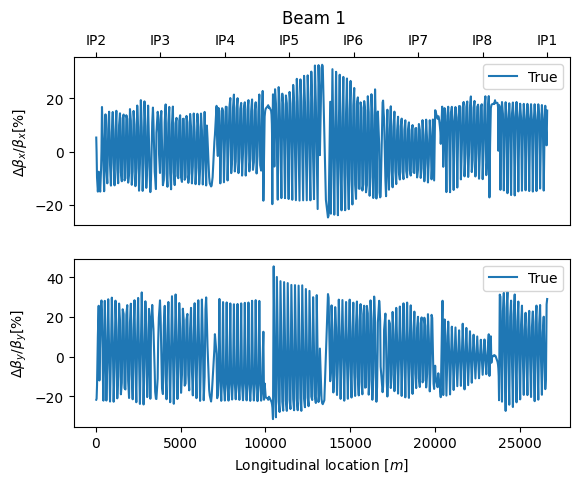
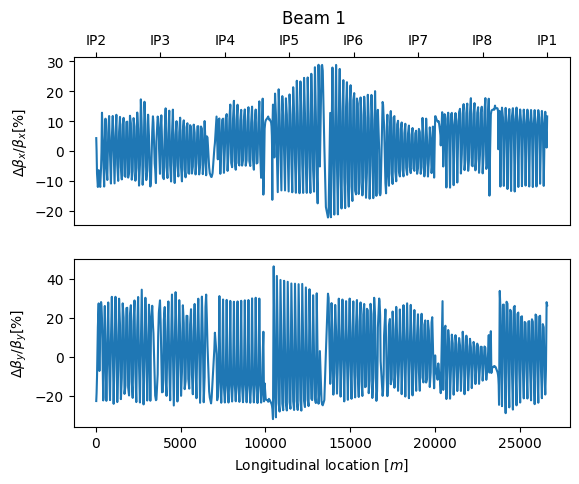
The mqts errors can not be cero!!

Introducing the mqt predicted by the ml algorithm, but we need the absolute value of the error, and also

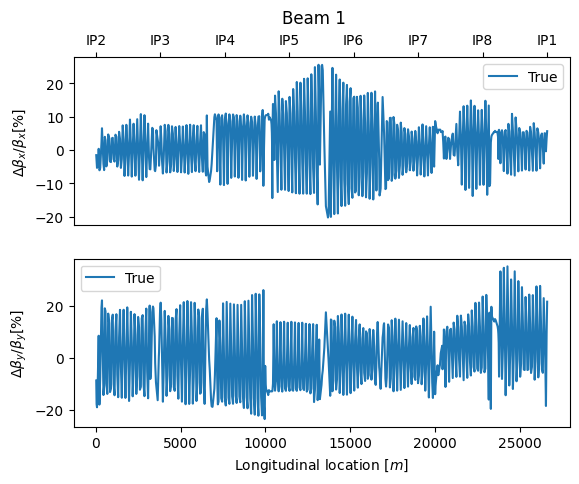
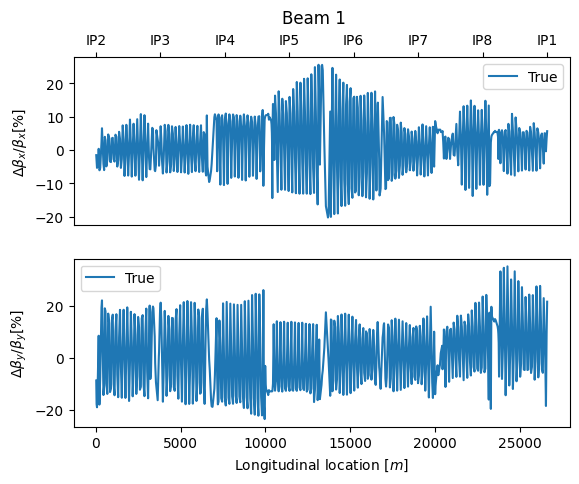
08/05/23

New test sample for reconstruction seed = 1 using the tfs data straight from data generation

**REAL RECONSTRUCTION**

****

**NOT MATCHING MATCHING**



Solution, not introducing the mqt values and matching to get them!!!

Not working!!! The matching is not really impacting, ploting different examples.

Probar nuevos algoritmos en vez de seguir por aquí!

09/05/23

Setting up training using LSTMs and CNNs. Searching for some data to use from the CERN logbook.

10/05/23

Trying to train using GPUs, HTCondor, lxplus gpu, and SWAN all give me issues. I think

- HTCondor is not as staight forward

-lxplus gpus is giving me some issues with the packages or cuda, maybe cuda not installed (???)

- Swan GPUs must be down today or something like that.

11/05/23

- Reinstalled tf in lxplus gpu,

-Training time much shorter 540 f/epoch to 25 s/epoch to for one of the models and 1/40 of the data around 50 times faster and maybe more.

12/05/23

-setting up CNN model!!!

15/05/23

-solving bugs and trying regression tree

24/05/23

Dec tree faster with MSE, choosing branch depth

Summary:

- Noise

- Twiss Recreating

- Error, mqt error generation, ask

- DL:

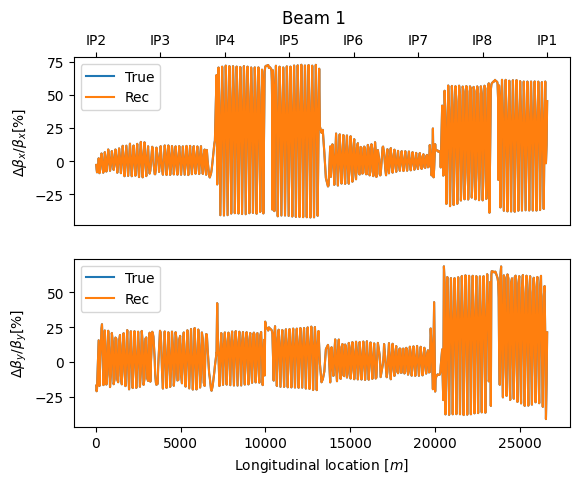
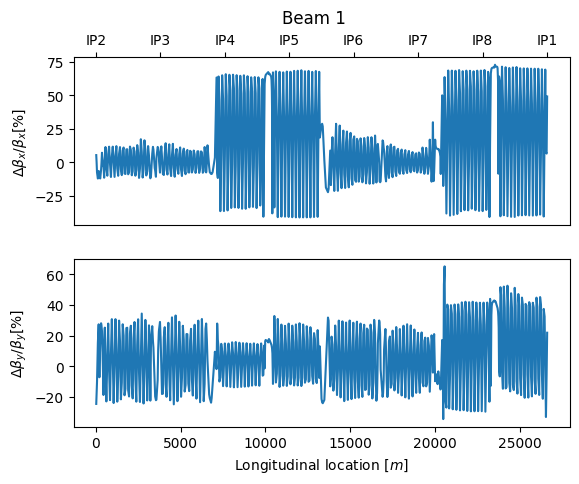
-Setting up GPU

- LSTM and CNN

Future: which measurement on the logbook would be better

Model training for beam 1  
- > assign arc errors as usual (triplet errors 1/10 of the usual strength to reflect the local corrections in + small residual errors)  
-> use the global corrections used in the machine  
  
Input: phase advance errors at every B1 BPM  
Output: arc errors B1 (including MQTs)  
  
Model training for beam 2:  
If we can out global corrections:  
-> usual errors in the arcs, 1/10 of usual triplet error values  
  
If not:  
-> same as for b2  
  
- maybe we can trim Q4 and measure triplet errors?  
  
Input: phase advance errors at every B2 BPM  
Output: arc errors B2 (including MQTs)  
  
For ‘virtual’ k-mod:  
Models trained on the data from same twisses:  
  
Input: phase advance errors everywhere  
Output: betas at BPMs: BPMSW.1L1.B1, BPMSW.1R1.B1, BPMSW.1L2.B1, BPMSW.1R2.B1, BPMSW.1L5.B, BPMSW.1R5.B1, BPMSW.1L8.B1, BPMSW.1R8.B1 (.B2 for the models trained for beam 2)

**CORRECT AND TRUE MATCH, ORIGINAL DATA GEN BUT NOT ORIGINAL, BUT CLOSE**

****

**The difference is the misalignment errors**

26/05/23

**Prepare for MD:**

+ Input correct errors: Done

+ Test and plot virgin model with different errors simulated: Misalign, mqt, triplets… Pending

+ Find correct optics: Done

+ Prepare and test models for MD:

- B1 Pending

- B2 Pending

+ Prepare measurement reading script and test on existing measurements Done

+ Create best knowledge model for twiss reconstruction

- Create the model Pending

- Plot difference between twisses Pending

+ Simulacrum Pending

Could try to predict misalignments

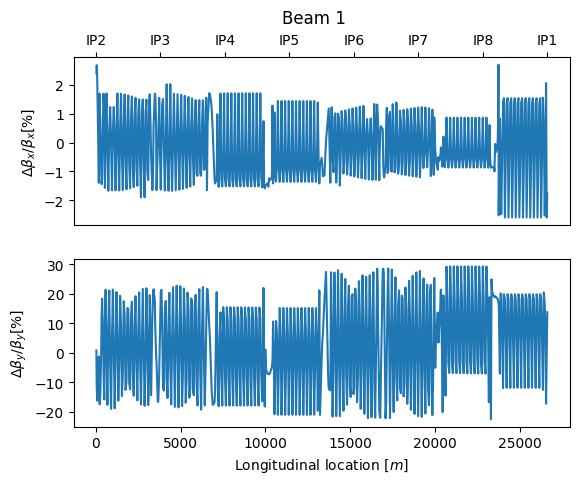
According to Elena all errors are relative errors, the macros work like that. “All quadrupoles in the lattice are assigned a random relative gradient error obtained from uniform distribution with the same rms error σ per magnets family”

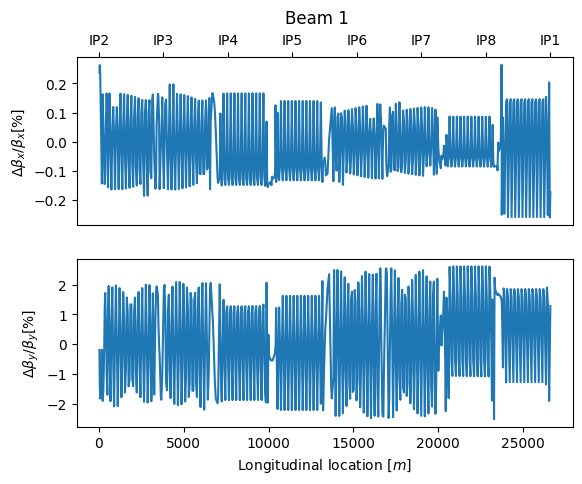
Which tune should I use??

30/05/23

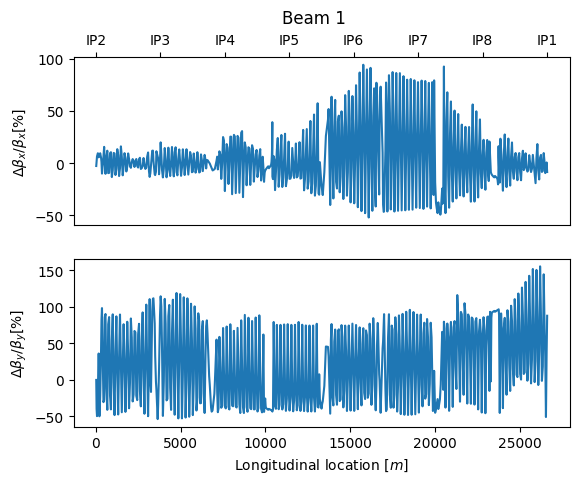
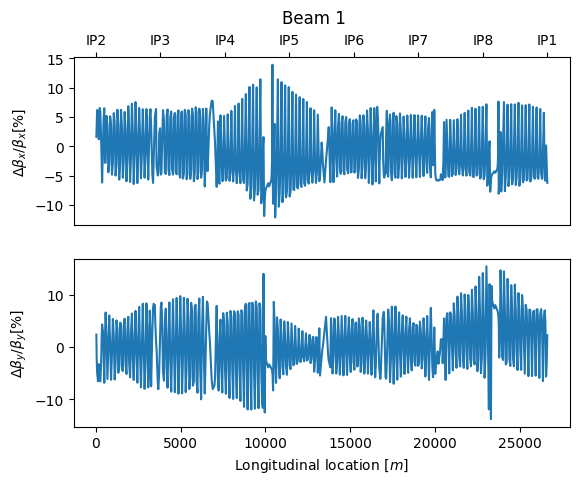
cs-ccr-dev3 crashed. Maybe my fault, more careful next time. Asking felix about the models.

**Some testing on the effect of different errors (SEED =3)**

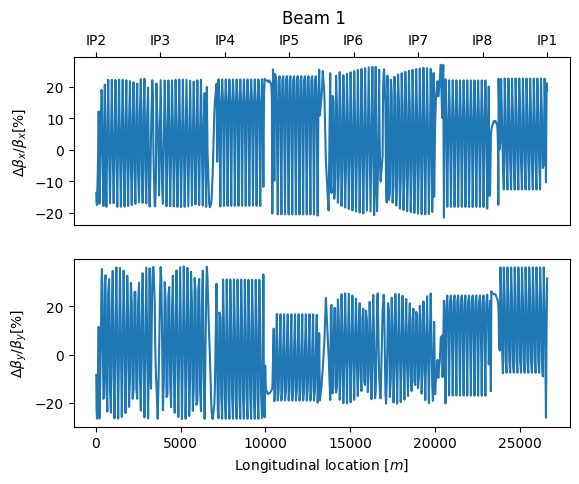
**Triplet errors only: 1/10 Triplet errors only:**

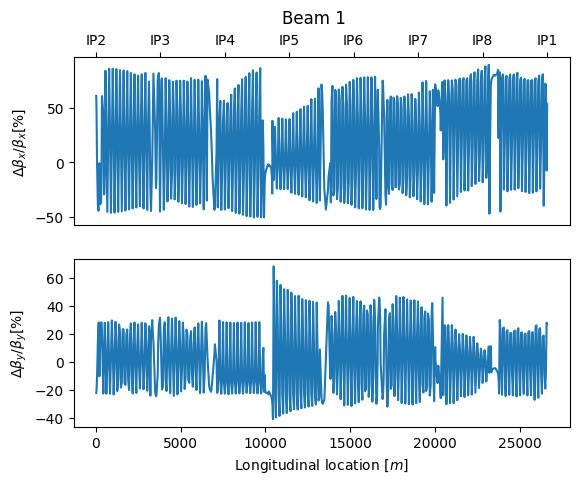
****

**1/10 Triplet+Dipole: 1/10 Triplet+Dipole+Arc:**

****

**1/10 Triplet + Misalign:**

****

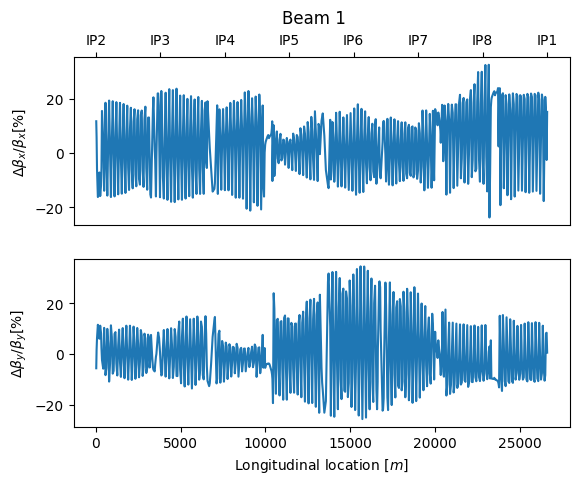
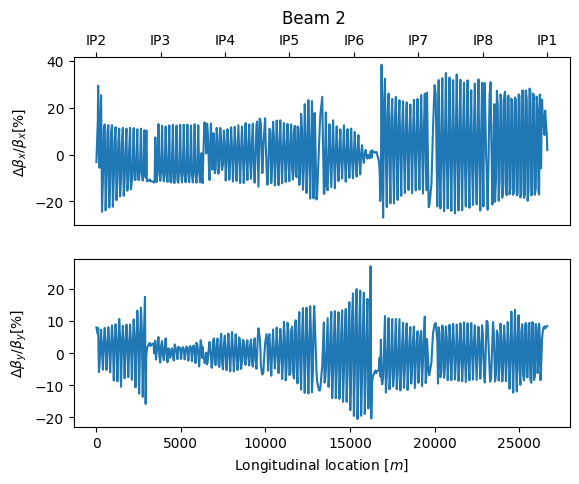
****

Check that the virgin data is okay, this is seed = 3, virgin model w everything

02/06/2023

LHCBEAM/ATS-2022-05-10\_B1\_globalcorr30cm

06/06/23



Questions:

* Is the 1E-3 noise adecuate, is it overshooting?
* How can I check corrections before making them
* Check error size for local\_corr\_md data, b2 should be 10 times bigger than b1

12/06/23

Finished the measurement to correction scripts. Checking if I can get the actual corrections and plot them to compare them, maybe even if they are different they still work due to degeneracy.

Checking all data, its fine!

16/06/2024

Lost 4 days of journaling, also the performance metrics

1:00 Problems in some Firmware in SPS or PS

1:50 Some advancements in the issue, it will be an extra hour

3:40 Kicking for reference measurements at 87cm

3:45 Beam dump, random dump

3:45 Ramp DownG

4:30 Finally Issue with a cooling cable

Generating Triplet errors only data. Discovered that global\_corr and best\_know data was generated wrongly, 100% misalignments no 10% misalignment!!!

19/06/2023

Finally generating only triplet and arc data, checking model performance in the next days, the best way to work is to generate all errors and save them, this way they can be used both for best knowledge and easier tasks.

Reading some papers on ML for accelerator science and non linear errors.

20/06/2023

Fixing the data and rearranging things in the code. Trying to plot RDTS using PTC MADX as a small test, memory overload.

Solo triplets sin mas errores ni nada funciona mucho mucho mejor.

No entiendo la idea de introducir misalignments y después no predecirlos.!!!!!

**Noise: 1E-3** **(maybe could use more data? maybe)**

Train: R2 = 0.8714997070015547, MAE = 9.175739274826868e-06

Test: R2 = 0.8571289786843876, MAE = 9.68252861570966e-06

**Noise: 1E-4**

Train: R2 = 0.9036335941189964, MAE = 7.619964959738119e-06

Test: R2 = 0.8975461744639082, MAE = 7.86826861514115e-06

What is the problema with delta(p)/p?? Rogelio said something about this.

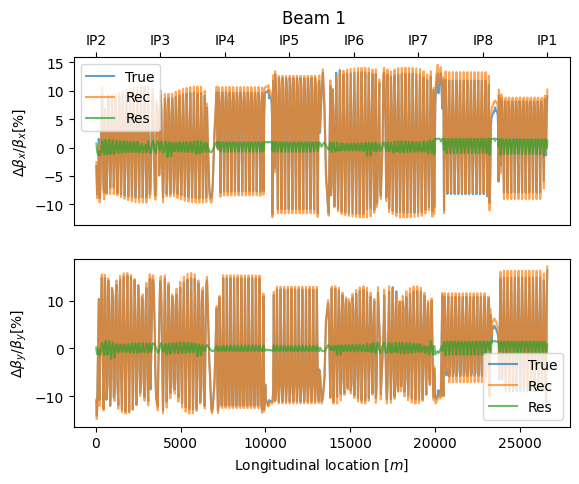
21/06/23

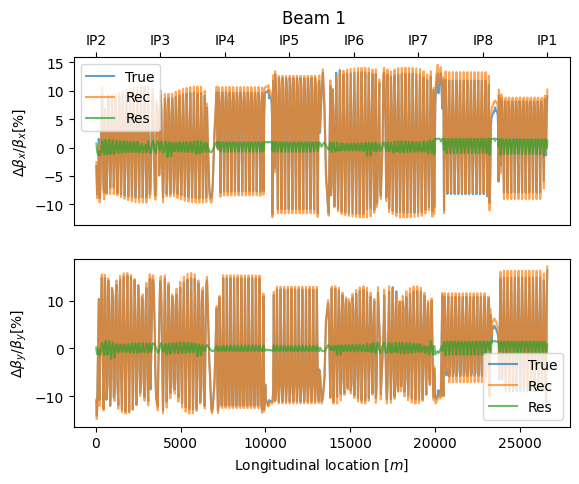
Triplet only phase advance

**Noise: 1E-4**

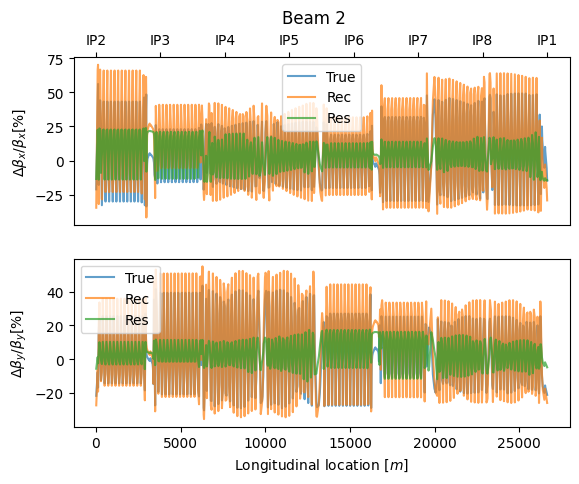
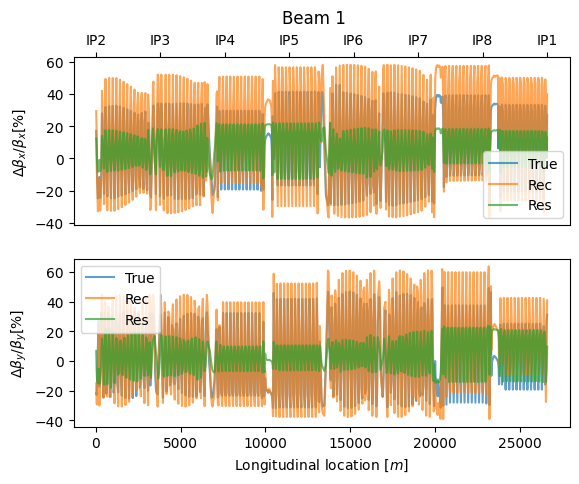
Train: R2 = 0.828539398590447, MAE = 1.1175135049490368e-05

Test: R2 = 0.8209933620285577, MAE = 1.137976453502223e-05

  
**SEED = 1**



**SEED = 2**



The correction seems to be much better for lower beta beating, maybe there is a systematic understimation because of ridge regression??

**TREE: 1E-3**

Train: R2 = 0.6864072693989036, MAE = 1.562993523351618e-05

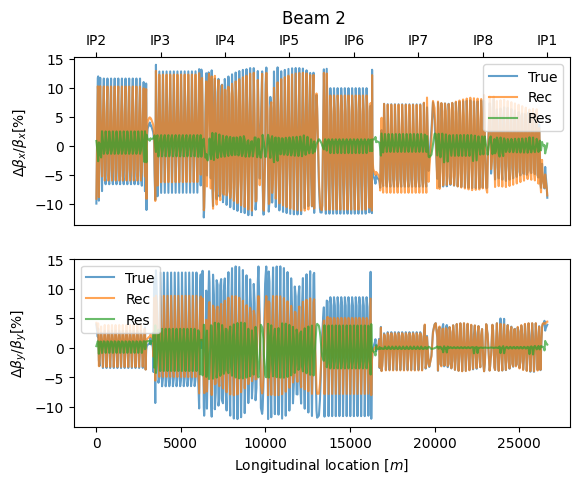
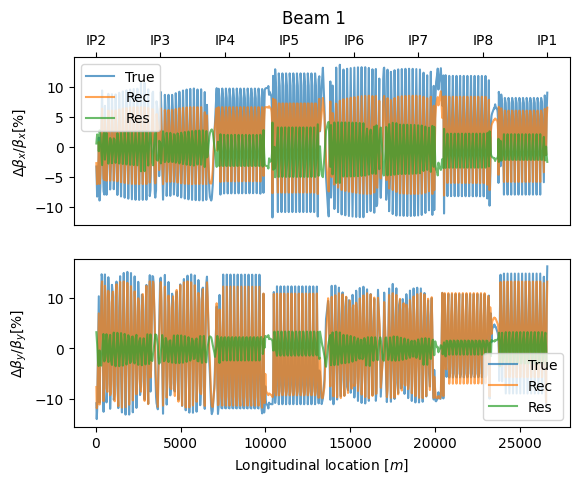
Test: R2 = 0.6593685129635116, MAE = 1.620958093943195e-05

**TREE: 1E-4**

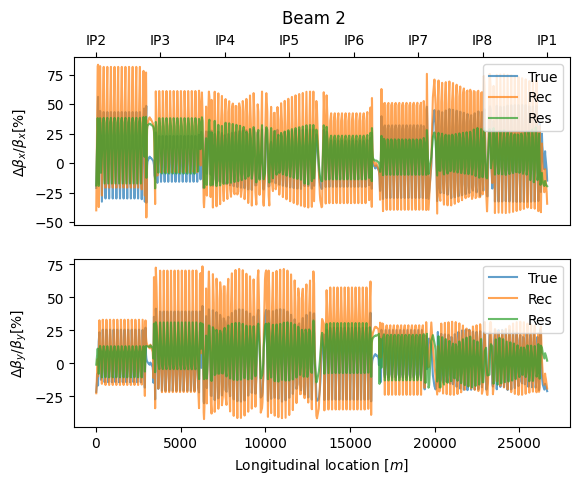
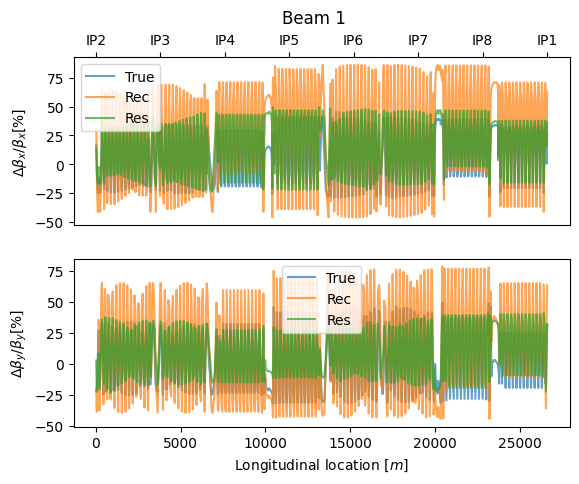
Train: R2 = 0.6895630910896087, MAE = 1.5552889995765836e-05

Test: R2 = 0.6652898014263584, MAE = 1.6072356326280285e-05

**SEED=1**



**SEED = 2**



Maybe models are just worse at predicting high beta beating??? Maybe low beta beating should be discarted for training as tobias said!!!! It might be normal, and it does not perform as well on higher errors.

22/06/23

**Starting non Linear errors:**

Starting Idea of the problem, simulating non-linear errors and trying to predict them from RDT data. Starting with the easiest case. The measured RDTs (driven) are not the same as the simulated RDTs (free) but this is a good first approximation, in the future maybe driven to error could be done.

**MADX-PTC:**

- mdx.ptc\_twiss(trackrdts=True, icase=5, no=5, file="b1\_monitors.out", closed\_orbit=True)

**- Execution time (s):** 2828.112677335739 => **47 min!!!**

Not feasible, need a way to make it a less heavy computation, many elements are not needed as well as many RDTs

**MADNG:**

- ????

**MADX-THIN LENS TRACKING:**

- Wont be faster

23/06/23

**MADX-PTC:**

- (MODEL=3, METHOD=2, NST=1) mdx.ptc\_twiss(icase=4, no=4,)

**- Execution time (s):** =>769.0243587493896 =>  **13 min!!!**

**MUCH FASTER, IS IT WORTH IT?**

It would take approximately **52 Hour to generate 10K** data points (best case scenario) This would be for all

26/06/23

Testing MADNG RDTs timing.

FCC Workshop.

Bayesian optimization. BoTorch. Markov Chain Montecarlo optimization for non linear errors. Jax?

27/06/23

Around 20-40 sec for rdt calculation with madng!!!! Depending on the integration order and other parameters.

29/06/23

**MADNG:**

For madng method = 6 the computation time is: **23.2 S**

For madng method = 2 the computation time is: **17.3 S**

RDTS seem too big, noisy, and sensible to nturns.

I am not so sure the optics are being properly loaded.

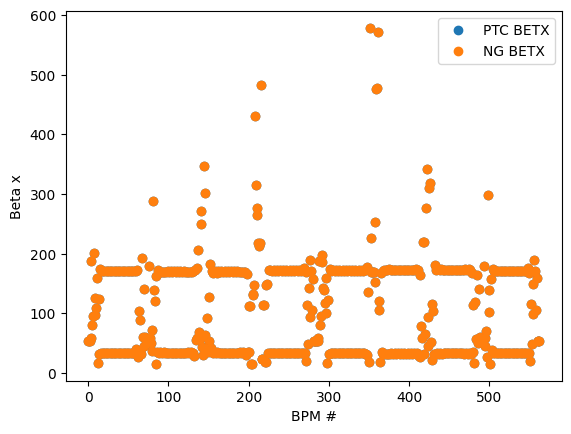
Tried both methods listed in examples for calculating RDTs, both are not working properly and don’t agree with PTC.

NEXT STEP, COMPARE AND PLOT WITH PTC

03/07/23

Not a single example script is working correctly with RDTs

Comparing the lineal parts of the computation, beta and orbits yields the same results.



Orbits are all cero! Same tunes

04/07/23

GNF[300000] (BPMSW.1R1) = 154.27

IN PTC it does not even show up!

Results MADNG 030000 = 154.271+176.784 i

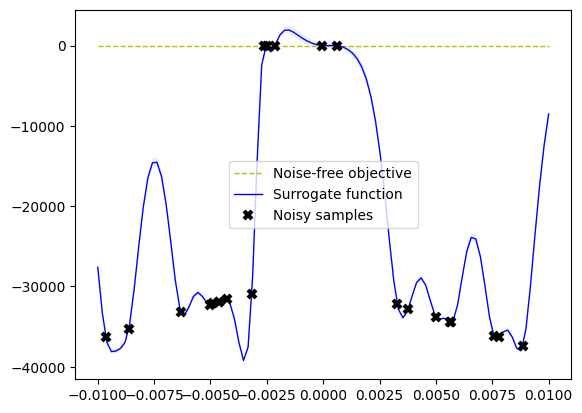
Result PTC 030000 = -1.395 + 0.574i

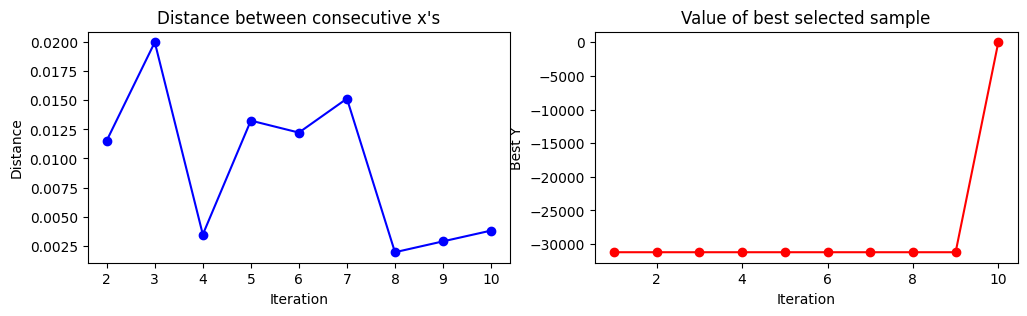
UNSTABLE ORBITS???

Testing Bayesian optimization with PTC, taking really long. Introduce a 1E-5 deviation to a sextupole knob, trying to obtain this knob value from the f(3000) absolute value

05/07/2023

Testing with a loss function calculated from beta beating. It seems to be working correctly, preliminary results!





**Introducing a big deviation of 1E-3 in the kqsx3.l2 circuit the model obtains a value of 1.04101E-3 within 10 iterations. Beta Beating after correction is minimal**

This is performed with bayesian optimization with a gaussian process regression surrogate function and an expected improvement acquisition function approach.

The question is, how will this generalize to:

* More errors
* More uncertainties in the model

The issue is that convergence must be fast, since the idea is to apply subsequent corrections to the machine and use actual measurements. In this case RDTS might help a lot, also MCMC methods.

**Tuning of the bounds, and hyperparameters is extremely important.**

In our case there is a very steep change on the slope when approaching the true value, this would make finding it with noise very hard.

Second simulated example, with 12 iterations:  
ERRORS

del kqsx3.l2 = + 1e-3

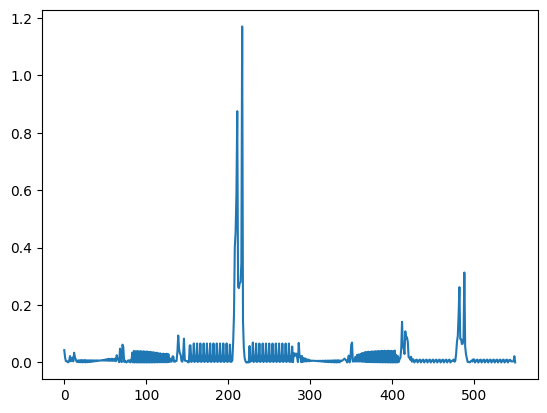
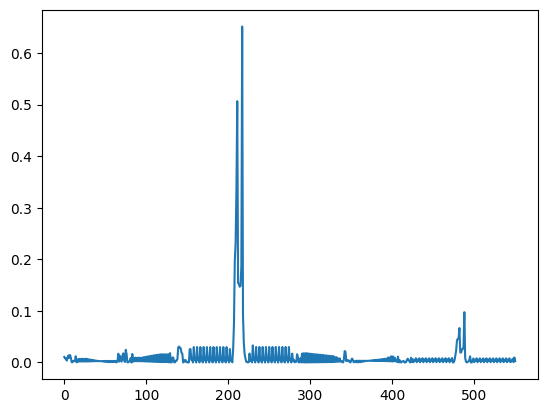
del kqsx3.r2 = 5e-4

Corrections: Not the actual simulated corrections but a 50% improvement

kqsx3.l2 = -5.75320008e-03

kqsx3.l2 = 3.75860602e-03

Uncorrected squared beta beating Corrected squared beta beating



Since rapid convergence should be important the exploration-exploitation parameter should reflect that!

Using RMS beta beating instead of sum of squares, even though is the same.

Probably using beta beating is not the safes option since it migh screw with other parameters such as phase or non linear parameters, however is a good start.

Using RMS instead of SUM OF SQUARES.

TEST WITH PTC leave it running.

06/07/2023

Reading papers and theory, conclusion:

- MCMC will probably converge faster.

- Scalability with higher dimensionality might be an issue

- What is the noise level for the GPR?? Alpha value, hyper parameter

- Two different sampling methods, variational inference and MCMC.

- Expected improvement is really hard to use multiple dimension

- For multiple factor optimization its calle MOBO and it minimizes the hypervolume, for example using BOTORCH using EHVI or (Expected HiperVolume Improvement)

-Existing implementation MOBO for accelerator physics: https://github.com/ChristopherMayes/Xopt

07/07/2023

Testin with proper error in Sextupole circuit, learn about how sextupoles are organized in the lattice.

10/07/2023

Testing with 11m optics, can not get it to work correctly.

**MADX-PTC:**  |gnf(310000)| at ip2 = 1178.54

**MAD:**  |gnf(310000)| at ip2 = 1401003.19

Talking with laurent it seems like it is not working, he will try to fix it in 2-3 days.

There is no WISE table for sextupole. It seems like the RDT errors come from the different small contributions from all sextupoles. Ask felix.

There are hundreds of sextupoles.

11/07/2023

HMC needs a analytic form to work.

Call UNI

Sextupoles used for LOCAL correction.

12/07/2023

Escrito al Jefe de Masteres.

13/07/2023

**MADNG Results:**

IP1: f200200r = -8.811992e+04

IP1: f200200i = 4.850031e+04

IP1: f400000r = 5.212319e+01

IP1: f400000i = -2.507562e+03

IP5: f400000r = -6.802671e+01

IP5: f400000i = 5.622129e+03

**PTC Results:**

IP1: f4000 = 52.16317533 -2507.563291

IP1: f2002 = -88119.82201 48499.94171

IP5: f4000 = -68.06480223 5622.150039

The cycle method is working properly

**Changing to a sextupole knob:**

**MADNG Results:**

IP1: q1 = 3.100000e-01

IP1: q2 = 3.200000e-01

IP1: q1j1 = -1.131681e+03

IP1: q2j2 = 1.873823e+03

IP1: f300000r = -5.114633e+00

IP1: f300000i = 5.894485e-01

IP1: f003000r = -6.480062e-17

IP1: f003000i = -1.824550e-17

IP5: f300000r = -6.329893e+00

IP5: f300000i = 2.827753e+00

IP5: f003000r = -4.660937e-17

IP5: f003000i = -6.497245e-17

**PTC Results:**

IP1: f300000 = -5.114626922 0.5894373025

For BPMS cycle can not be performed so its not getting the correct RDTS, but its calculating the one at the marker position. I can test the trk method to see if its the same.

17/07/23

Opening up a route about RDT prediction with autoencoder and SINDy modelling. Very Challenging, however denoising is a possible direct application.

Decision: Start with a denoising application without any real physical knowledge.

Read and ask about the dynamics and how the decomposition in normal forms is performed, since is not a real FFT, or is it???

18/07/23

TO DO FOR SVD DENOISING:

- 1 Simulating non Linear errors in MADNG lattice

- 2 Tracking with madng, obtaining tracking data and FFT with omc!

- 3 Play with simulated noise levels (?)

- 4 Training a VAE to denoise the signal

- 5 Try SINDY with raw data to obtain the RDTS

+ Other way:

- 1 Find actual kick measurements and FFT data, ask felix. See how to scrap for the data

- Problem, balanced data, is it general and not biased??? There is no way to tell, but it could be tested.

19/07/23

Choose, what should I priorice from this simplest examples:

* SVD Cleaning from tracking sim
* SVD Cleaning from actual kick data
* Knob prediction from RDT data using classical ML, or BO.

Since for SVD Cleaning from tracking sim and knob prediction are much faster in MAD NG, I will continue using it. Also they both need the same error generation!! So the problem of error generation is more important.

I have been looking into the idea of tune signal denoising, I think the first step should be using a normal autoencoder to try to denoise the signal, this creates two possibilities:

      - Use measured kick data: Pros. Real data, No need for data generation

                           Cons. Hard to scrape all data, no idea if the data is biased or will lead to not generalizable models!!

      - Use tracking data: Pros. Controlling the effect of noise, more generalizable, also useful for the optimization problem

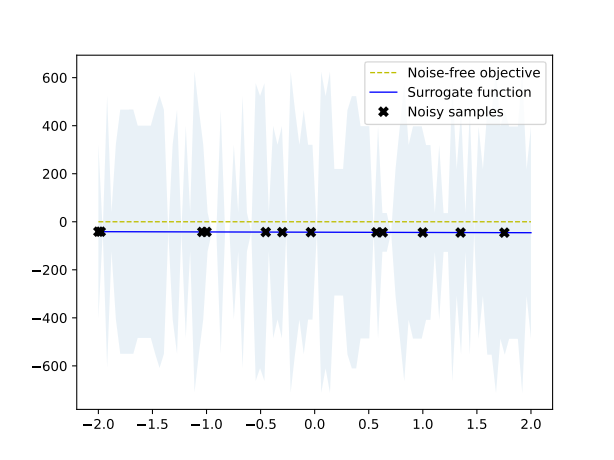
                      Cons. Costly simulations (MADNG), Not real data so it might be less aplicable to the actual machine.

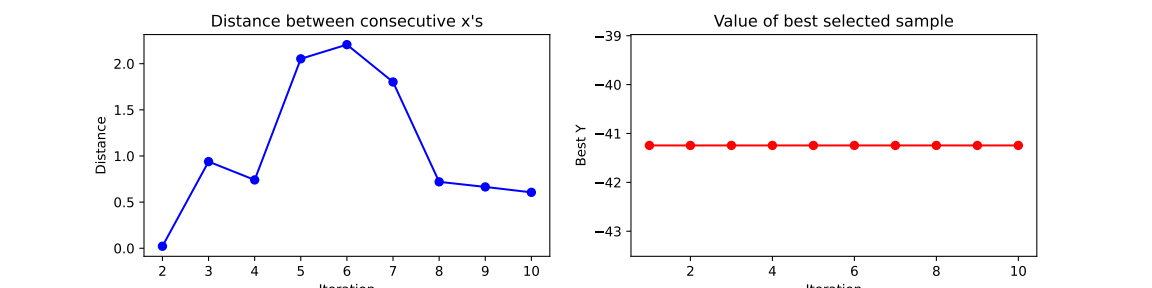
I think the tracking data approach is more likely to work, since there is no way to tell how good is the measurement data. Meaning that if the data is biased for example mostly measurements with corrected optics this would yield a biased model that means that is not generalizable.

With the tracking data approach we can make sure the model is trained with all possibilities meaning different errors in the lattice. This will yield a more generalizable model and it has more chances to work.

20/07/23

The results with PTC vs MADNG with different PTC algorithm settings are around 5%, pretty big. MADNG: 44... VS PTC: 43….





The lower the correction, the lower the rdt, makes no sense, maybe other rdts are increasing.

Other rdts such as 4000 are also changing, I dont understand why this would be the case.

However I am going to focus in the error generation portion for madx scripts, this is probably the most useful use of my time.

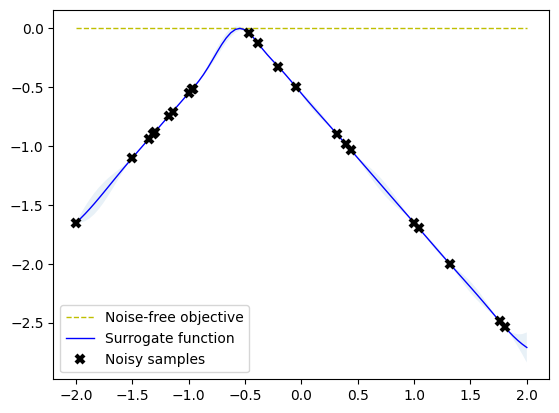
Calculated mean and standar deviation for the errors, now i need an error generation scripts as well as tracking and spectra simulation.

- Error generation

- Tracking

- Spectra calculation

Higher order RDTS are also changing, a lot, makes no sense wont pay attention to them, trying to measure de rdt deviation from nominal! Maybe this way it will work



21/07/23

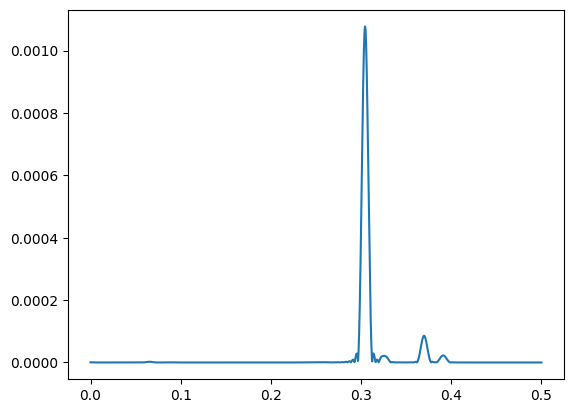
Error Generation with wise table variances for MQX.

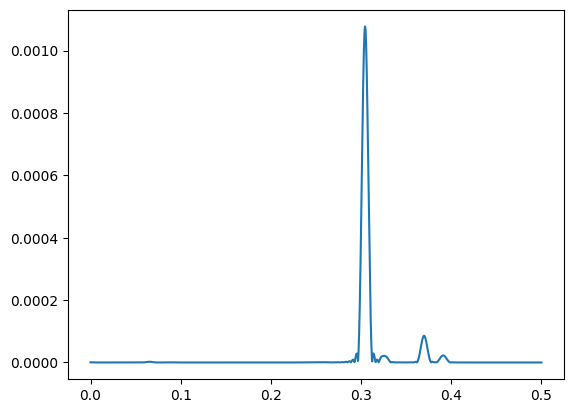
24/07/23

Done tracking and Spectra analysis using OMC hole\_in\_one script.

Time for tracking: 146 s

Time for analisis: 1.3 s





I still need to scale to the action for the tracking to make physical sense.

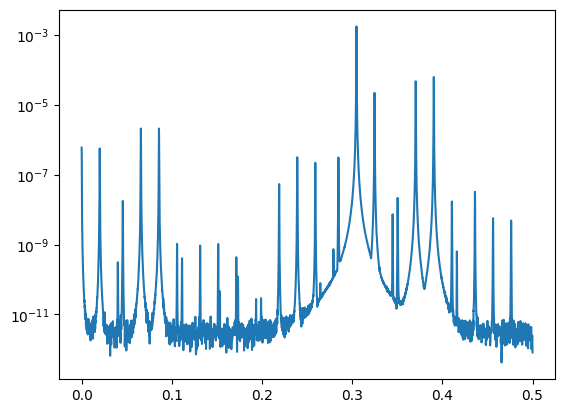
Also, mesurements are made wrt the orbit with errors, and not the nominal orbit, I think I should change this

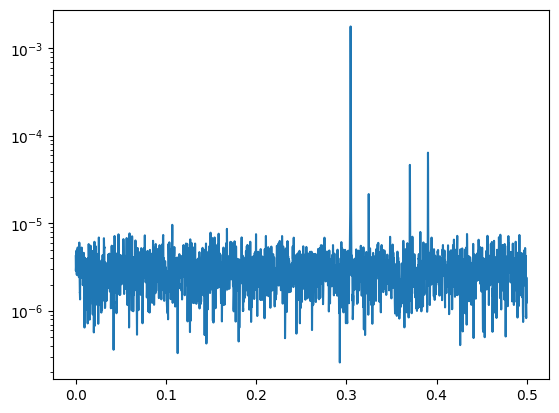
25/07/2023

Setup errors correctly, set noise levels for hole\_in\_one, setup correct (?) kick for tracking, still no main change in the tune signal. meeting with felix tomorrow would be nice.

26/07/2023

Noise = 1E-4 Noise = 0





If this is the real actual pattern of the data we want to reconstruct is not going to work, but maybe the peaks can be recovered. Lets test hole in one cleaning.

All the tracking and analysis is taking around **120 s per sample**, that means its pretty slow for data generation.

08/08/23

After vacation.

Summary of previous work:

- Spectra analysis seems very time consuming with the tracking, however further exploration must be done: scaling by the action correctly and speeding the process! Like using madng.

- RDT optimization still needs the scripts to be adapted and cleaned for correct data generation.

I would really want some results and launch some simulations, where should I focus?

10/08/23

The difference between madx and madng is because of the efcomp function, also the dkn are not as straight forward.

The fact that I wasnt getting any changes in the spectra is because the use statement after was deleting the errors.

I dont know how to figure out whats up with the field errors.

11/08/23

a,b are for relative errors, already figured out for K2 and madx, lets see for the skew is not as straight forward. Taking into consideration that madx outputs K1L, however dknl.

14/08/23

For all magnets

PTC. RDT 3000: 0.1451910

MADNG. RDT 3000: 0.1994092053695482

PYMADNG gets ahead of itself! It is very sensitive to changes in mad.recv

For knob Change

RDT 3000: 0.5461050591801495 MADX

RDT 3000: 0.5424300516445868 MADNG

0.6% error for a Knob Change

Only for MQXA.3L2

RDT 3000: 0.0003086477157836498 MADNG

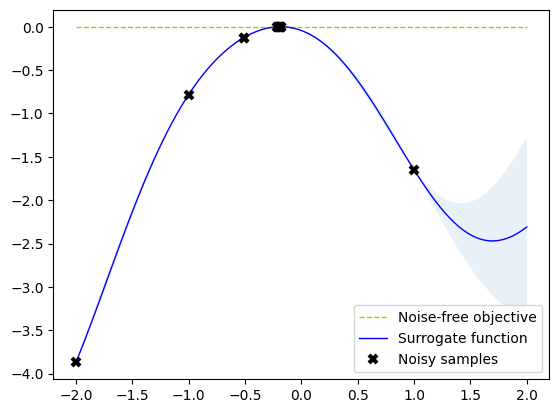
RDT 3000: -0.001589469475028693 MADX

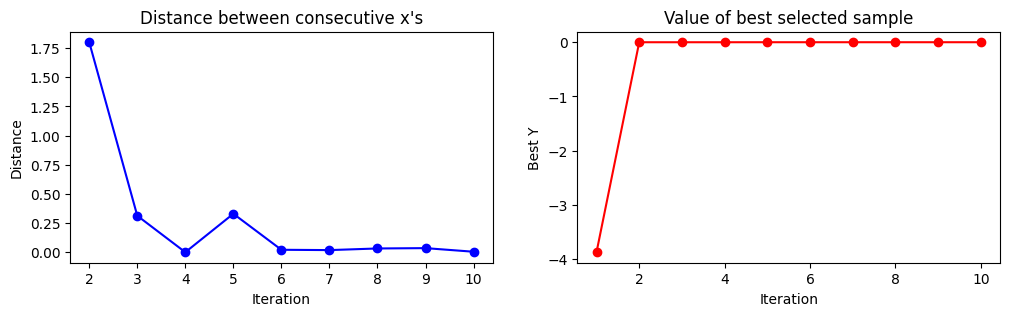
There are big differences when assigning particular errors. Look this up!!

Testing bayesian optimization with MADNG

Changing The kernel:

[x,y]: [-0.18300933] , -8.567347268835272e-07





Cant figure out what are the differences between MADNG and MADX

I feel like this time MADX is wrong and not MADNG

15/08/23

Trying to debug proves being really hard. Joshua told me to check model and slicing methods!

16/08/23

Trying the slicing from the toolkit instead of the makethin comment, however I took them from a script that was supposed to compare the same two things.

Since I am pretty stuck in MADNG i am trying to do something with the tracking code.

Meet with Felix!

With the different makethin

rdt = -0.0045 MADX ptc

rdt = 0.000308 MADNG

After meeting:

Input bigger error, observe the graphs, take RMS of the difference not difference of the RMS. Sensitive to noise?

Save the errors and load them after making use! Try making thin before error generation

For k2l Err = 1E-3 => K2L: 6.579728E-03

MADX PTC => 0.5890886979255896

MADNG => 0.5941

This seems more reasonable!!! Why then when applying to multiple triplets it does not work? Lets try again with the next ptc command!

For k2l Err = 1E-4 in all triplets!

MADX => RDT 3000: 14.661433415679504

MADNG => RDT 3000: 14.754079897921123

AGREE UP TO 0.6%

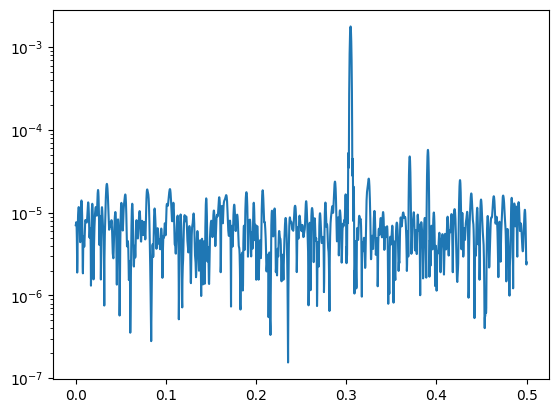
For k2l Err = Random in all triplets!

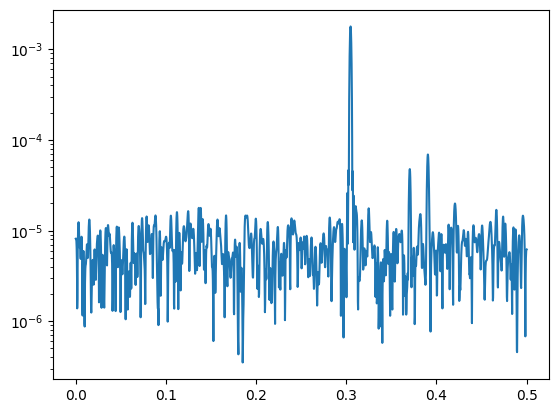
MADX =>RDT 3000: 0.19659191491385997

MADNG => RDT 3000: 0.1994092053695482

1.4% Error, as long as its sistematic it should be fine

Also fixed the spectra signal script with the loading of previous errors!

Noise level 1E-4 and error is 1E-5 Noise level 1E-4 and error is 0



This are different!! Finally they can be used! Can we use the fact that there is a noiseless and noisy version to recreate the noiseless version??? lets see!

17/08/2023

Comparison plots between PTC and MADNG. Adding K2SL and K3 errors. New loss function, RMS of the difference not diference of RMS. Check differences again.

21/08/2023

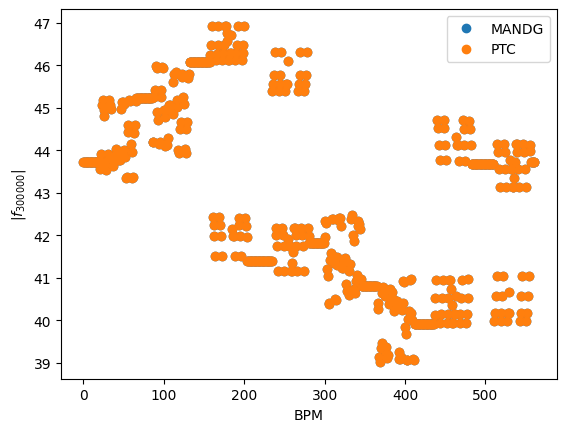
The resolution of the codes might be the reason for the difference of one code to another. Check the different results, with errors, with one error and with a distribution of errors.

22/08/2023

NOMINAL RESULTS

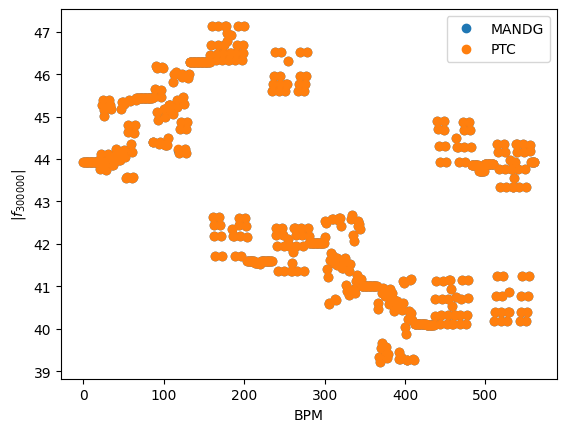
RDT 3000: 6.892872015906891e-13 PTC

RDT 3000: 0 MADNG



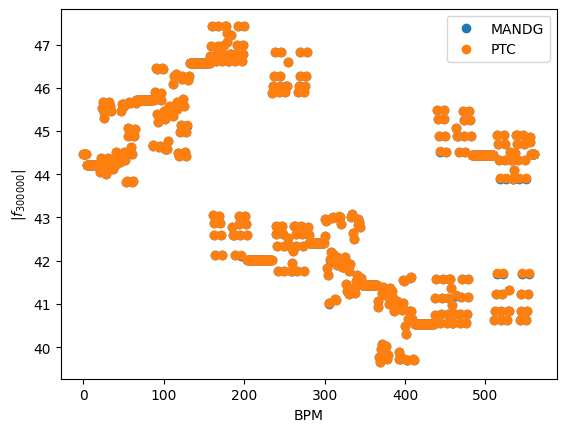
FOR A RANDOM SET OF ERRORS:

RDT 3000: 0.004693744527489033 PTC RDT 3000: MADNG



RDT 3000: 0.014170051911478281 MADNG

RDT 3000: 0.014197248486323057 PTC



23/08/23

Correcting with the actual corrector for the BO example! Finishing the presentation!

24/08/23

MADNG:

RDT3000: 43.654681362804894

RDT4000: 6110.388085114724

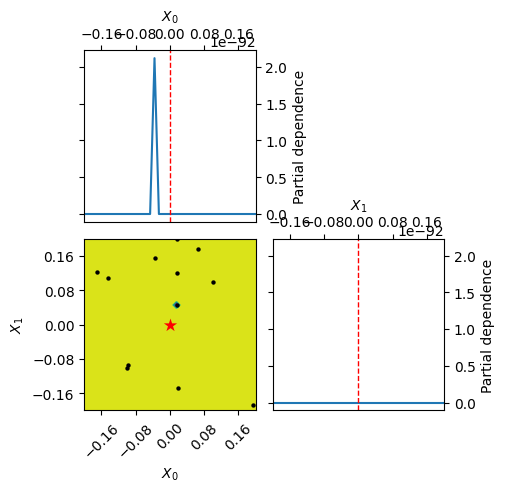
PTC:

RDT3000: 43.65541584440454

RDT4000: 6111.26325922064

25/08/23

Testing on multivariate bayesian optimization



28/08/23

Going into generating data for supervised learning. Introducing higher order PTC integrator to see if the precision improves, it does not, It would be interesting to see source of the error. **Check this out**

29/09/23

Made a script for multiple simulations at once. In the optics server its pretty slow, testing HTCONDOR.

78 s Per sample, slower than local??? => Parallel 40 s per sample, but not 20s as expected

11/09/23

**SUMMARY BEFORE VACATION**

Summary and observations of the work done before vacation. Creating a MADNG script with k1, k2, k2s, k3s errors, MADNG and PTC do not agree as good as expected, specially with higher order errors, although the errors are proven to be the same. This must be looked into before running more simulations. Just before going on vacation I was reading theory, setting up this jobs for HTcondor and using OMC tracking to learn relevant RDTs.

Pending work

For the ML applications to work with the simplest case (no misalignments or added errors) the precision issue must be looked at. Also all relevant RDTS saved, to do this I need to try the OMC rdts tracking. Once this is under control dataset generation can begin.

For this two weeks.

During this two weeks the expected things to do are:

* Trying to manage work related stress and adjust the way of working accordingly
* Prepare and perform the MD as good as possible and creating the necessary models with the **new optics!!! Urgent!!**
* Learning theory, reading the presentations and having the meetings that Felix recommended. Learn about scaling by the action for the tracking simulations.
* Legitimation and CERN cards!
* Lastly, try to solve which RDTs to save and the precision issue.

12/09/23

Talking with Jacqueline if there are new optics if I should create new data. Generating new models just in case, also there was maybe a mistake, using 2022 seq instead of 2023, although it might be the same, I think there cas no 2023 seq when I was doing my things.

13/09/2023

Check if the .seq file is the same for 2022 as for 2023. See how my MD scripts work and check if they are correct again in retrospect.

It seems like everything is working correctly.

Measurement 6 does not have a model and a correction script yet since I thought there would be no time and maybe losses would be too big.

Reading literature on non linear corrections, tomorrow will check with MADNG precision issues. The two .seq files are identical!

15/09/2023

MD On tuesday morning in principle. Setting up data generation for MADNG. Fixing my hole\_in\_one script.

Check action scaling before using the found RDTs. I think the problem is that the tracking is saving a lhcb exit point not only BPMS.

18/09/2023

Reading more theory, fixing tracking,

The relevant RDTS according to OMC optics analysis.

Normal quadrupole: (associated with normal quad errors)

F0020\_y

F2000\_x

Normal sextupole:

F0111\_y

F0120\_y

F1002\_x

F1011\_x

F1011\_y

F1020\_x

F1020\_y

F1200\_x

F2100\_x

F3000\_x

Normal Octupole:

f0013\_y

f0031\_y

f0040\_y

f0211\_y

f1102\_x

f0220\_y

f1120\_x

f1120\_y

f1300\_x

f2002\_x

f2011\_x

f2011\_y

f2020\_x

f2020\_y

f4000\_x

f3100\_x

Skew Quadrupole

f0110\_y

f1001\_x

f1010\_x

f1010\_y

Skew sextupole

f0012\_y

f0021\_y

f0030\_y

f0210\_y

f1101\_x

f1110\_x

f1110\_y

f2001\_x

f2010\_x

f2010\_y

Skew octupole

f0112\_y

f0121\_y

f0310\_y

f0130\_y

f1003\_x

f1012\_x

f1012\_y

f1021\_x

f1021\_y

f1030\_x

f1030\_y

f1201\_x

f1210\_x

f1210\_y

f2101\_x

f2110\_x

f2110\_y

f3001\_x

f3010\_y

f3010\_x

Many many more rtdts than expected, this is where ML excels, but maybe there is a practical or theoretical way to decrease this number, maybe take out redundancy. Maybe taking the rdt deviation is also a good idea in this case.

Theory, what is the difference between f3010\_x and y. for example, I guess is the resonance in the other plane, and the numbers only refer to the the frequency. Then what does MADNG return????

45 Different RDTs according to MADNG, that does not differentiate from \_x and \_y components.

Start generating new datasets with the optics servers since no one uses them.

19/09/23

Testing the efficiency right now.

10 s/sample with the parallelization (20 jobs)

Right now it will take 52 h to generate the 20k samples, this means 250h for 100k. 2 days for 20k samples, 10 days for the whole dataset, for only one beam. This can be optimized.

- Testing bottlenecks, maybe the way data is stored in .csv files. It had a sudden spike

- Using HTCondor

- More jobs in parallel

- Using all servers also office computer.

-The error assignment i remember it was pretty slow

I think the main issue is that parallelization is not working properly, and that pymadng is waiting for other scritps??

For one parallel sample in lxplus 56 s instead of 28 s in my computer for two samples it should take only a bit more but it takes 110 s. I think it might be using the same instance of madng?? Is this a problem???

For my computer

1 , 28s

2, 51 s

7, 35 s

10, 35 s

Easiest way to solve this is to use htcondor and send different jobs… Now to figure out how it actually works.

HTCONDOR cheatsheet

condor\_submit exercise01.sub

watch condor\_q

Parallel computing

[https://htcondor.readthedocs.io/en/latest/users-manual/parallel-applications.html#:~:text=HTCondor's%20parallel%20universe%20supports%20jobs,perhaps%20communicating%20with%20each%20other](https://htcondor.readthedocs.io/en/latest/users-manual/parallel-applications.html#:~:text=HTCondor's parallel universe supports jobs,perhaps communicating with each other).

[Monitoring](https://monit-grafana.cern.ch/d/000000032/home?orgId=1)

<https://monit-grafana.cern.ch/d/000000032/home?orgId=1>

20/09/23

-----------------------------------------------------------------------------------------------------

**Very fruitful meeting with Elena.**

- Discussing issues with bayesian optimization! Too many variables.

The proposed solution was to do the correction in an iterative maner, correcting for each IR and freezing for the next BO optimization, this decreases the number of variables optimized.

She has also observed that using a surrogate model based on supervised learning was really helpful, in our case it would be the model used for supervided ML, this would give us the initial conditions!

**The plan is the following:**

- Optimizing parallel computing

- SL only with K2

- Adding more errors

- Add more realistic simulations, NOISE and what actual rdts can be measured with NOISE and CLEANING artifacts a possible issue is modeling the CLEANING for the SL, i guess using less noisy samples is okay??

- Doing the iterative BO, try to compare local vs global RDT corr.

- Apply the previously trained SL surogate function to BO initial values

- Try model-less correction to compare model based, model guided and model less approach (boby qa)!

-----------------------------------------------------------------------------------------------------

Trying the pool option.

MADNG is much less RAM hungry than MADX so maybe using more processes is better?

For 20 processes 560 s => 28 s per sample

I see that with already 2 samples the CPU is overloading, I need more CPU?? lets try it out

21/09/23

Measurements and CAS presentation

22/09/23

Testing HTCondor

Generating the first datasets!!!! Sending to HT condor, a whole dataset is 50 GB of data, each time HTCondor is going to send 10 GB, this should be enough..

Preparing for neural network training!!

28/09/23

CAS:

Things done before:

* Setting up and testing different non linear error sim. With relevant RDTS according to OMC.
  + Noiseless Done
  + Noise Pending
* Measurements. Done
* Chat with Elena Done
* Setting up data generation, still in progress, problem with parallelizing but generation of the appropriate format data
  + Parallel Done
  + HTCondor Pending
* Setting up training, still need data Pending

30/09/23

Setting up a more efficient data loading and saving, also making sure I am reading the same error and sample file.

09/10/23

Finally setting up HTCondor? Testing on local HTCondor, everything works, testing on local.

Tracking and adding noise 10E-4 with OMC hole in one script, the same RDT outputs as before. **Question for Felix, how can I see which RDTS are actually relevant to this case? (Too much noisy measuring)**

See what is the x, and y about, read the Advanced CAS procedings for RDTS!

Pregunta para Felix, por que hay una componente H y V en los RDTs si en el formalismo hamiltoniano no se observan?

Question for Felix, why is there a H and V component for SOME RDTs if using the Hamiltonian formalism this does not appear, I guess it does appear I tought it was implicit in the jklm indices? Why is it only for some????