Preliminary results for PyAT-based loco code

Elaf Musa - PhD student (DESY)

Supervisors: Ilya Agapov, Tessa Charles

October 21, 2022

Table of contents

FCCee corrections

LOCO(Linear Optics From Closed Orbits)

Implementing LOCO in PYAT

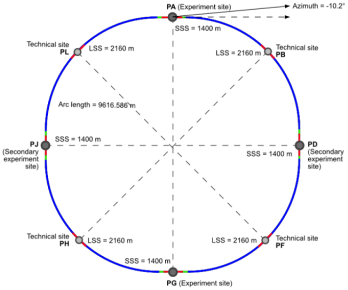
Simple lattice example

FCCee first test: fixed/random errors values

Speeding up the PyAT-based loco code

Summary

FCCee corrections



- A common challenge of the low emittance storage rings is their high sensitivity to errors and the need for precise control of beam parameters.
- Algorithms for orbit correction, linear optics correction are important to reliably run these facilities. (AC Dipole, LOCO, etc.)
- Their are limitations in the performance of these various techniques.
- Existing correction studies by Tessa Charles do not include simulation of the optics measurement procedures .
- We study the potential of LOCO for collider optics measurement and correction.

LOCO(Linear Optics From Closed Orbits)

The parameters in the used model is varied by LOCO to minimize the χ^2: deviation between the model and measured orbit response matrices C_{model} and $C_{measured}$.

$$\chi^2 = \sum_{ij}(\Delta C^{ij} - \sum_k \frac{\partial C^{ij}}{\partial g_k}\Delta g_k)^2$$

$$\Delta g_k = (\frac{\partial C^{ij}}{\partial g_k}^T \frac{\partial C^{ij}}{\partial g_k})^{-1}(\frac{\partial C^{ij}}{\partial g_k}^T \Delta C^{ij})$$

- We implemented LOCO in PyAT.
- First tests for the implemented code was done for the FCCee_t_v22 lattice (work in progress) .

Corrector-to-BPM response matrix

- An accelerator with m-BPMS and n-correctors produces an m x n dimensional response matrix.

$$C_{mn} = \frac{\sqrt{\beta_m \beta_n}}{2\sin(\pi\nu)} \cos(\pi\nu - \phi(s) + \phi(s_0)) + \frac{\eta_i \eta_j}{\alpha_c L_o}$$

- The aim of the closed orbit correction is to find a set of corrector kicks θ that satisfy the following relation:

$$\Delta x + C\Delta\theta = 0$$

- Closed orbit response is directly simulated.
- First tests in PETRA III and FCC-ee closed orbits were performed

Implementing LOCO in PYAT

```python
def ORM_x(dkick, ring, BPMs_random_noise, used_correctors_Names=None):
    cxx = []
    cxy = []

    for i in range(len(used_correctors_Names)):

        cor_index = get_refpts(ring, used_correctors_Names[i])
        cor_index = cor_index[0]

        ring[cor_index].KickAngle[0] = dkick

        closed_orbitx, closed_orbity = closed_orbit_bpm(ring, BPMs_random_noise)
        cxx.append(closed_orbitx)
        cxy.append(closed_orbity)

        ring[cor_index].KickAngle[0] = 0.00

    Cxx = np.squeeze(cxx) / dkick
    Cxy = np.squeeze(cxy) / dkick

    return Cxx, Cxy
```

Implementing LOCO in PYAT

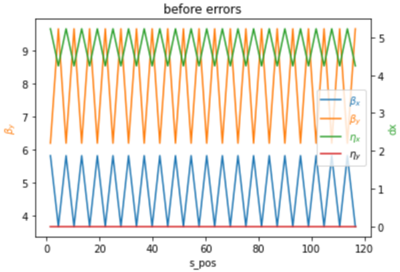
- Choosing quadrupoles families from in the lattice.
- Choosing orbit correctors equally distributed around the lattice.
- Calculating the quadrupoles families response matrices (Jacobian)
- Applying several correction loops

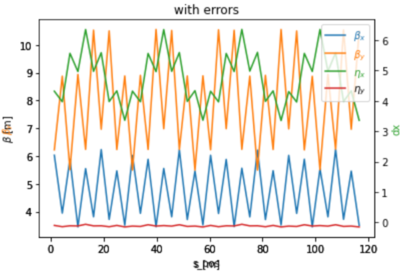
```
def QsensitivityMatrices(ring, qfamilyIndices, dk, BPMs_random_noise, used_correctors_Names):

    strength_before = []
    for i in qfamilyIndices:

        strength_before.append(ring[i].K)
        a = ring[i].K
        ring[i].K = a + dk

    qxx, qxy = ORM_x(dk, ring, BPMs_random_noise, used_correctors_Names)
    qyy, qyx = ORM_y(dk, ring, BPMs_random_noise, used_correctors_Names)

    for i in range(len(qfamilyIndices)):

        ring[qfamilyIndices[i]].K = strength_before[i]


    return  qxx, qxy, qyy, qyx
```

Simple lattice example

Parameters:

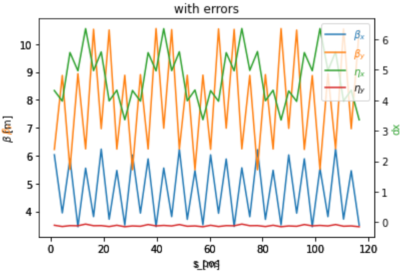
- 50 quadrpoles QF,QD,QS, 40 orbit correctors
- 1.e-3 rms gradient errors QF QD
- 1.e-5 m tilt errors QF QD
- skews strength = 0.05 m-2
- BPMs noise = 0.0
- Sextupoles turned off
- SVD cut off = 50
- 5 corrections loops

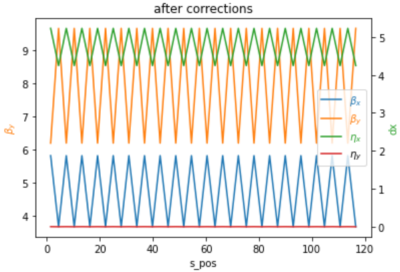




Results

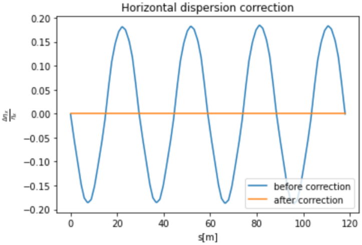
RMS Beta beating with errors:
x:4.94% y: 8.46%
RMS Beta beating after correction:
x:1.27e-08% y: 8.57e-09%

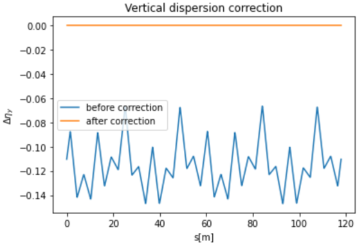


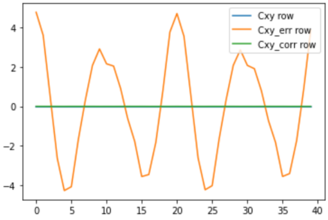


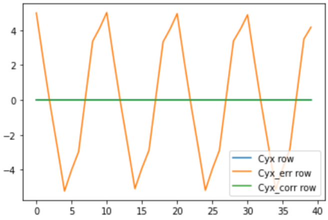
Results

Dispersion and coupling correction:





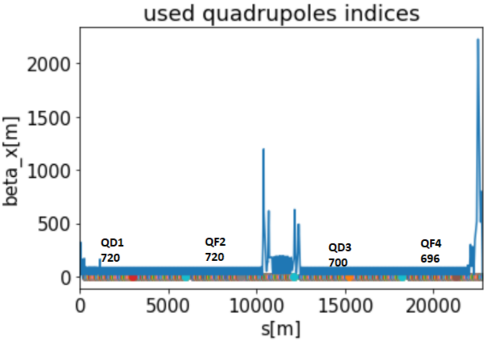




First tests: FCCee lattice with fixed errors

FCCee_t_v22 sequence file was converted to PyAT using the xsequense tool. (https://github.com/fscarlier/xsequence)
Number of correctors and BPMs were introduced to the FCCee_t_v22 lattice using PyAT

- Total number of quadrupoles 3324: 104 quadrpoles families

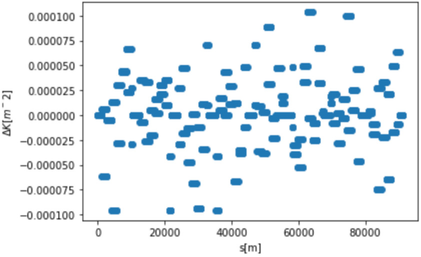


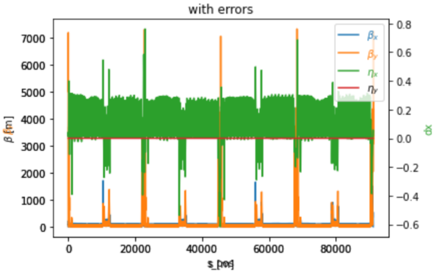
- Total number of correctors 3324: using 4 correctors
- Total number of BPMs 3324: using all BPMS

First tests: FCCee lattice with fixed errors

Parameters:

- 4.e-3 rms gradient errors (fixed through the family elements)
- BPMs noise = 0.0
- Sextupoles turned off
- 5 corrections loops





Results

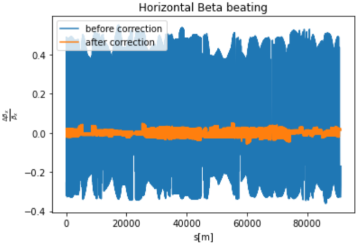
Execution time $\sim$ 25 min

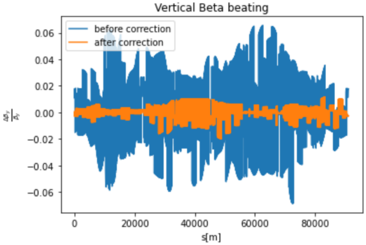
RMS Beta beating with errors:

x:30.97% y: 2.77%

RMS Beta beating after correction:

x:1.9% y: 0.50%

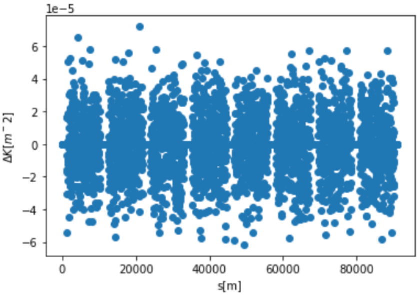




Random errors case

Parameters:

- Increasing the number of quadrpoles and correctors families
- BPMs noise = 0.0. Sextupoles turned off
- 400 quadrpoles families, 15 orbit correctors
- 10.e-4 rms gradient errors(random)
- 5 corrections loops



Random errors case

Result:

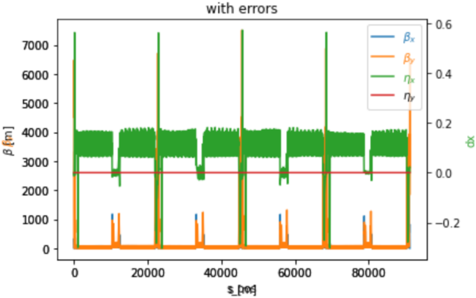
Execution time ~ 4.5 hours

RMS Beta beating with errors:

x:3.37% y: 5.88%

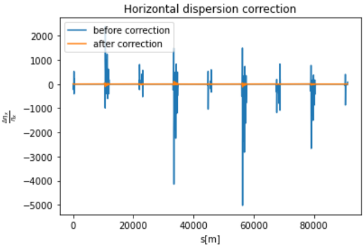
RMS Beta beating after correction:

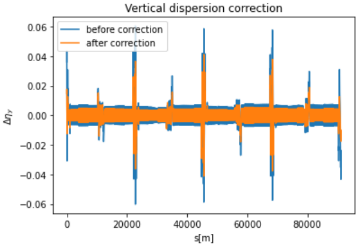
x:1.33% y: 3.71%



FCCee dispersion correction

- Inserting skews to the FCCee_t_v22 lattice.
- Use 24 skew quadrpoles families/80 arc quadrpoles families.
- 1.e-3 rms gradient errors(fixed through the family elements)
- 24 skews strength = 0.001 m-2





Speeding up the PyAT-based LOCO code (work in progress)

- Performing code linear profiling helped to reduce 9 hours $\longrightarrow \sim$4.5 hours
- We aim to perform code parallelization over several processor units
- Study how to reduce the problem such that the solution approximate the full solution well.
- We aim to use the Analytic formulas for rapid evaluation of the orbit response matrix by A. Franchi and S. Liuzzo.

Summary

- We made PyAT implementation of closed orbit and linear optics corrections github.com/iagapov/pyat-loco.
- Tests performed with simple FODO and FCCee_tt lattice with sextupoles off.
- Fine-tuning of the measurements and correction procedures are on going.

Outlook

- We aim to Improve the PyAT-based LOCO code to reduce the simulation consumed time
- Including BPM noise, sextupoles and radiation as next step.

Thank You