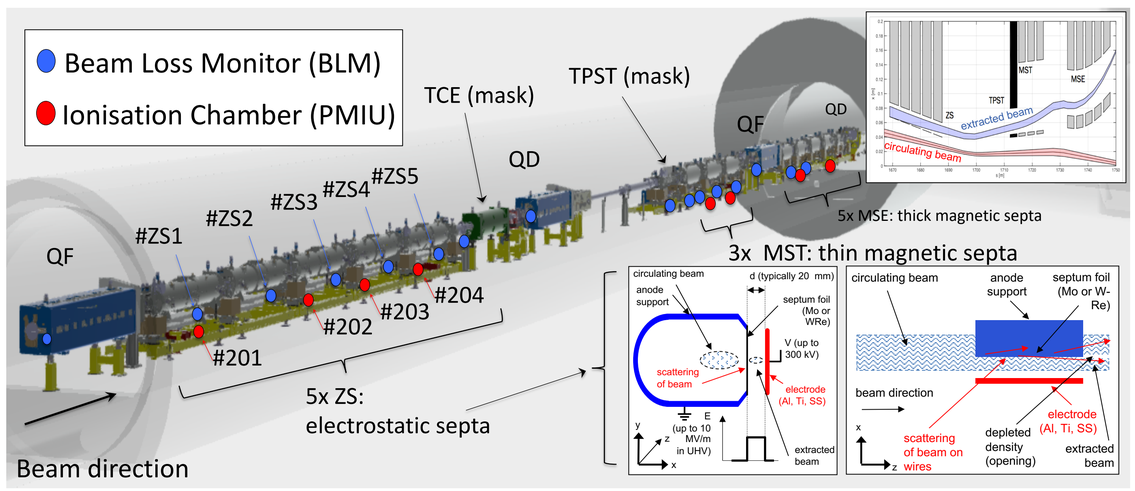
Electrostatic septum alignment studies at the CERN SPS

M.A. Fraser#, Y. Dutheil, B. Goddard, S. Hirlaender, V. Kain, K. Li,
J. Prieto, M. Szakaly, L. Stoel, F. Velotti, CERN & M. Kagan, SLAC

Work of the Automatic ZS Alignment Working Group (AZSAWG) [1]

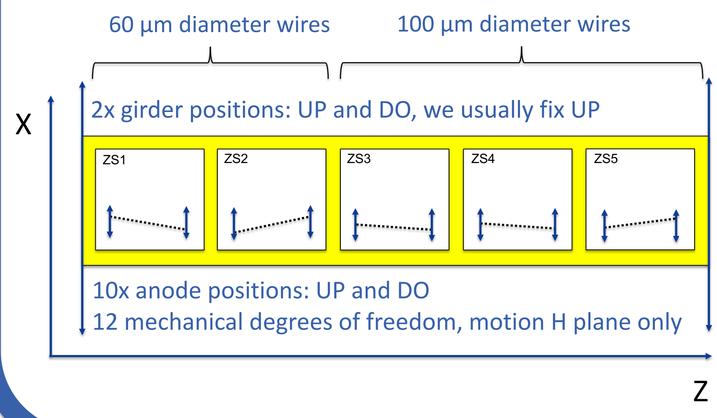
2nd ICFA Workshop on Machine Learning for Charged Particle Accelerators

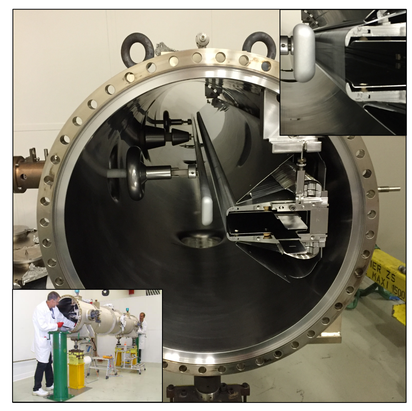
400 GeV slow extraction system



Electrostatic septum (ZS)

ZS is composed of 5 independent tanks mounted on a single girder, which spans almost 20 m along the extraction straight:





Simulation tool

pyTrack: a simplified tracking model written fully in Python [2]:

- Linear machine + thin sextupole kicks + scattering modelled in pycollimate [3]
- Source term at septa upstream is generated in MAD-X and resampled every time
- Only the last 3 turns before extraction are simulated: huge speed up
- Custom or off-the-shelf optimisers can be easily plugged into the library. Collaborations welcome!

Septa geometry:

- More detailed than MAD-X: independent tanks with field
- Less detailed than FLUKA: "ribbon" anodes instead of individual wires

Alignment procedures/algorithms

Current (operational) alignment procedure:

1. Scan girder and fix at location of minimum loss
2. Scan every anode motor sequentially from upstream to downstream, fixing at minimum of loss each time
3. Repeat until no further benefit

Pros: simple; Cons: time-consuming, poor worst-case convergence

Gradient descent:

1. Scan girder and fix at location of minimum loss
2. Estimate gradient:
   - Probe each degree of freedom, left and right: 2 shots * 9 degrees of freedom = 18 shots
3. Move all anodes together in direction of decreasing gradient with exponentially decreasing step size.

Bayesian Optimisation [4]: maximise a target function f: $X \to Y$

1. Define a probability measured over a suitable function space with a Gaussian process (infinite-dimensional generalisation of a multivariate Gaussian), $f(x) \sim \mathcal{GP}(\mu(x), k(x, x'))$
2. Probe point x* maximising expected improvement w.r.t. that probability measure, $x^* = \underset{x \in X}{\operatorname{argmax}} \mathbb{E}[f(x) \mid \mu(x), k(x, x')]$
3. Update the probability with each function evaluation.
4. Repeat and stop after given number of steps.

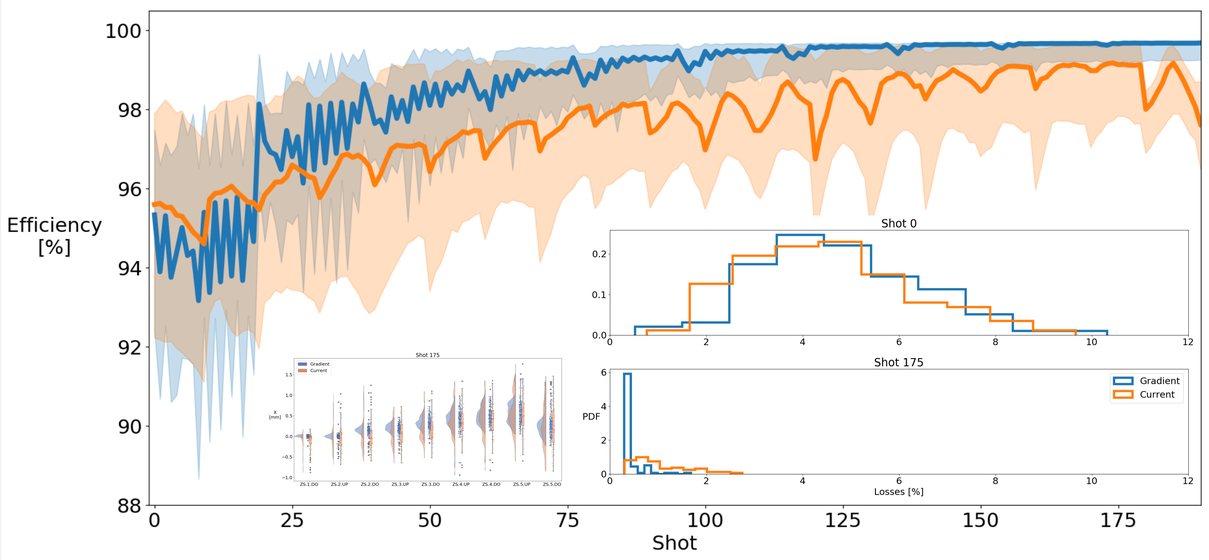
Simulation results

Error studies:

1. Position girder at minimum losses
2. Randomly misalign all tanks except ZS1 upstream, yielding 9 DOF
3. Simulate alignment procedure over a few hundred shots
4. Repeat for many different initial misalignment seeds N(0,σ=500 µm)

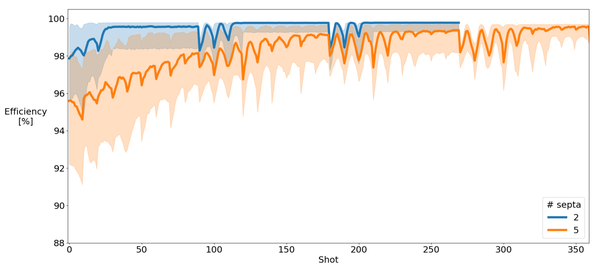
Comparison of algorithms:

- Current (operational) vs. gradient descent:

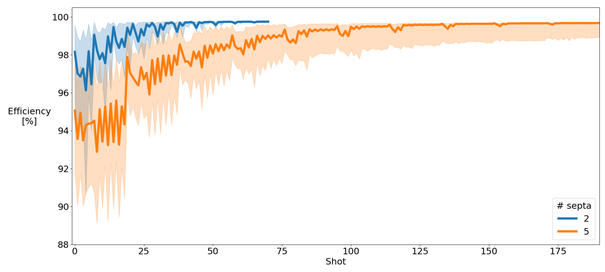


- Number of septa: changing the number of DOF

Current (operational) procedure:

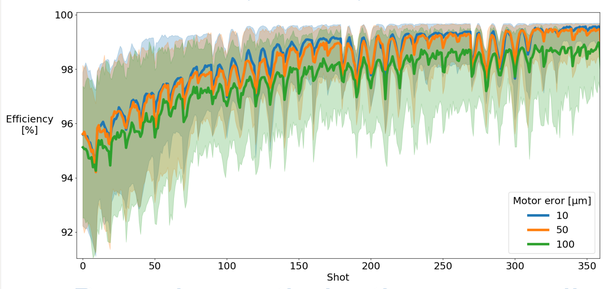


Gradient descent:

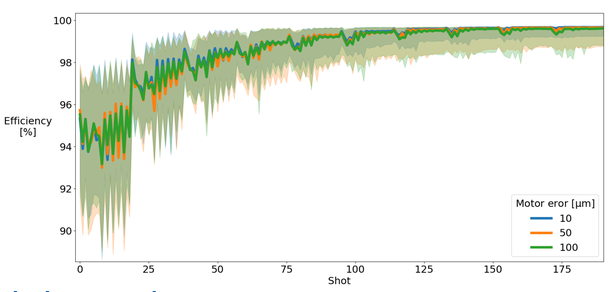


- Motor precision errors:

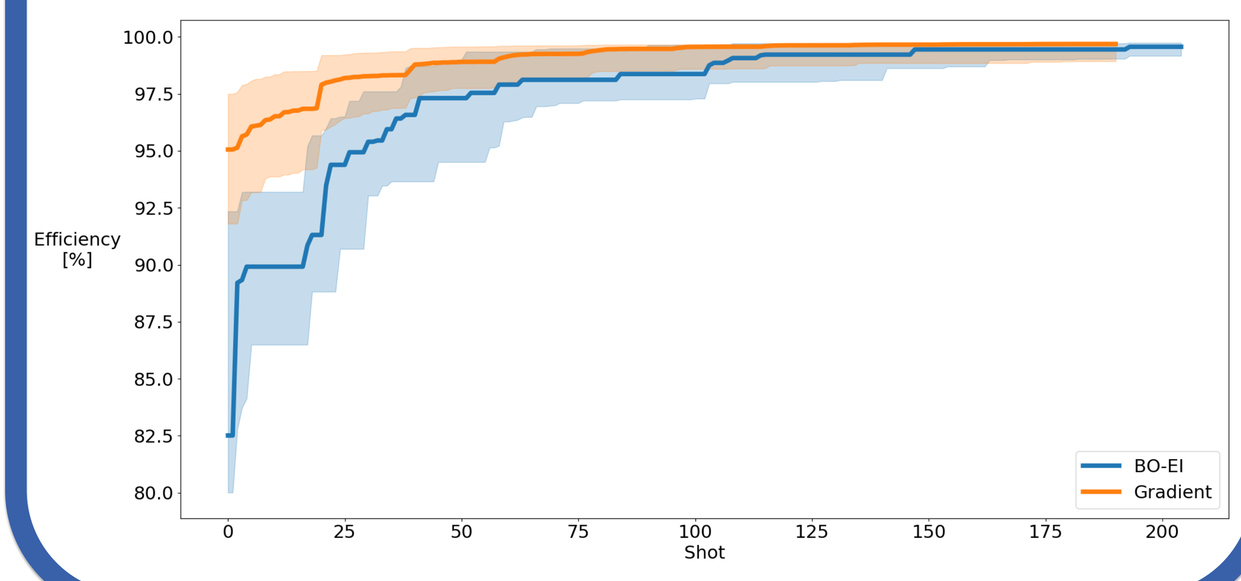
Current (operational) procedure:



Gradient descent:



- Bayesian optimisation vs. gradient descent:



References

#Mail to: mfraser@cern.ch

[1] AZSAWG: https://indico.cern.ch/category/9695/
[2] J. Prieto, https://gitlab.cern.ch/jprietop/pyTrack CERN's Gitlab.
[3] F.M. Velotti, PyCollimate: francesco.maria.velotti@cern.ch
[4] Bayesian Optimization,https://github.com/fmfn/BayesianOptimization