Online Modelling and Optimization of Nonlinear Integrable Systems

Nikita Kuklev, on behalf of IOTA collaboration

NAPAC 2019

September 3rd, 2019

In partnership with:

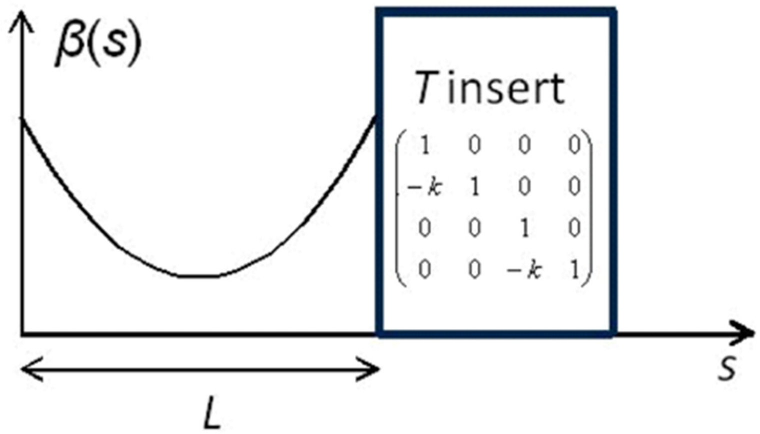
Why NIO?

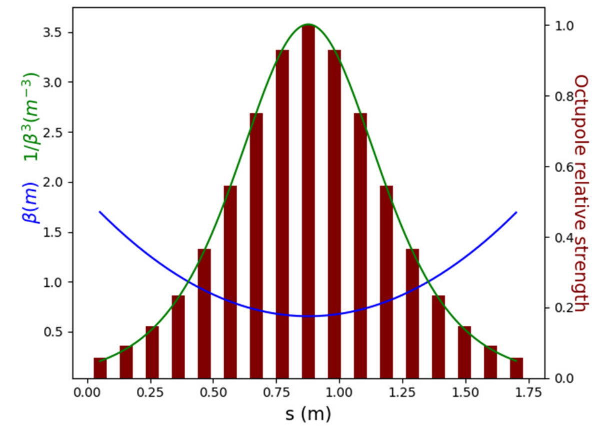
- Fast collective instabilities limit intensity
  - Suppressed with Landau damping
  - Requires tune spread → need nonlinear elements → lower DA
    - i.e. 336 octupoles @ LHC

- Nonlinear integrable optics (NIO)
  - Still gives strong tune spread
  - And maintains DA!
  - Has special requirements

How to build NIO?

- The ‘T-insert’ condition





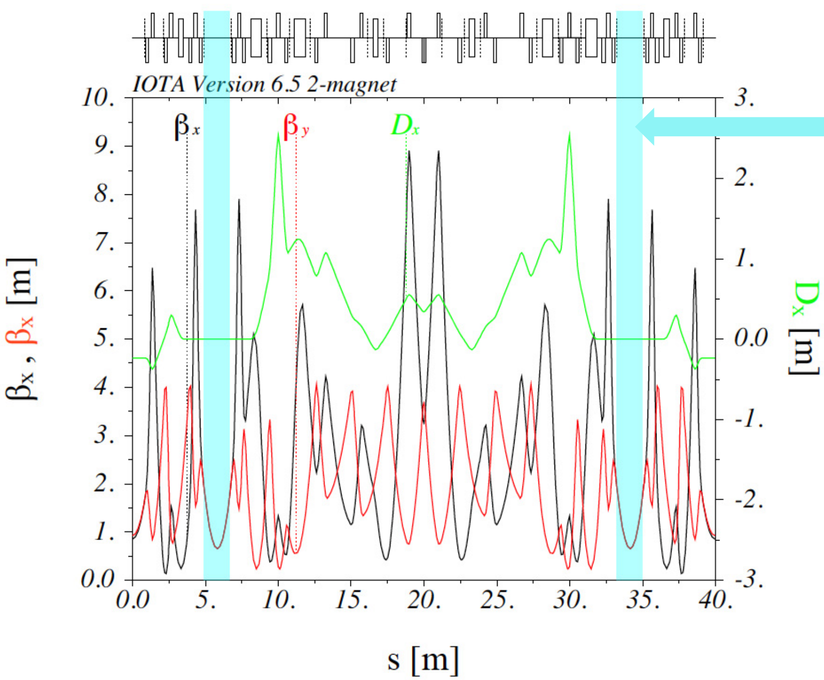
Practical requirements:

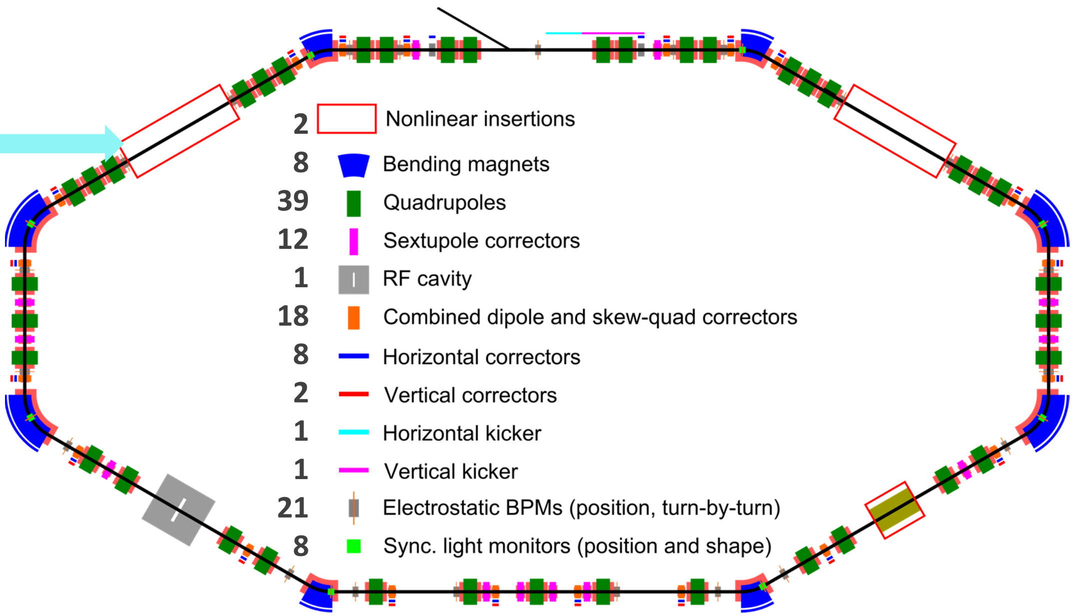
- Round axially-symmetric linear lattice (FOFO)
  - 2π x n phase advance
- Drift with $\beta_x=\beta_y$, no dispersion – put special V(x,y,s) here

Phys. Rev. ST Accel. Beams 13, 084002 (2010)

Experimental implementation - IOTA

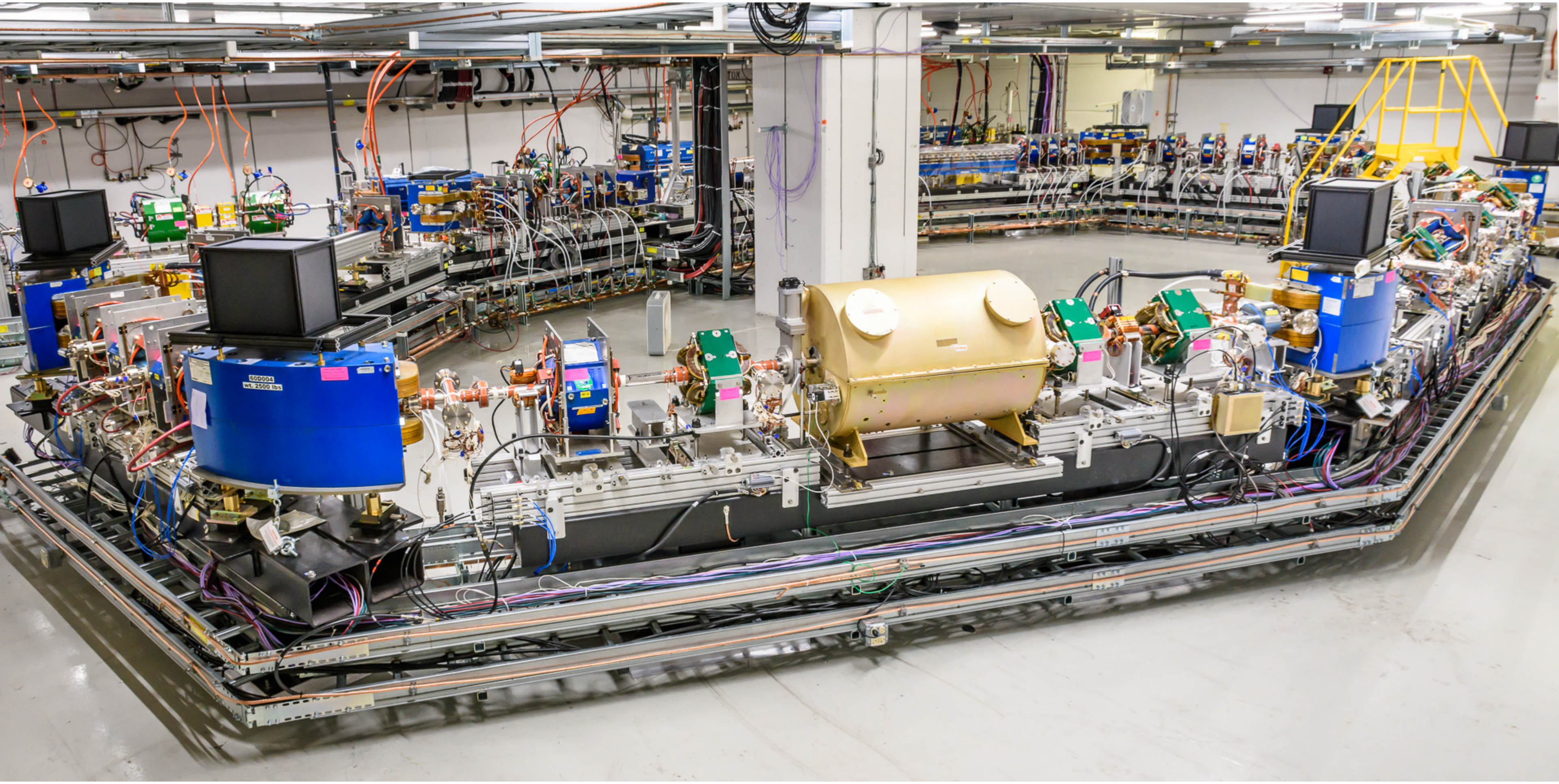
- Extreme flexibility, every magnet individually powered
- Satisfies all NIO contraints





Experimental implementation - IOTA

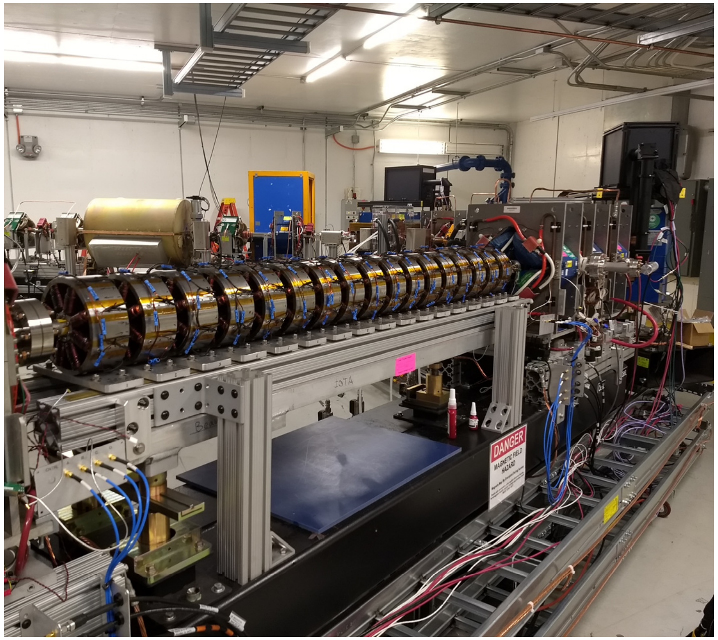
- It is already built!



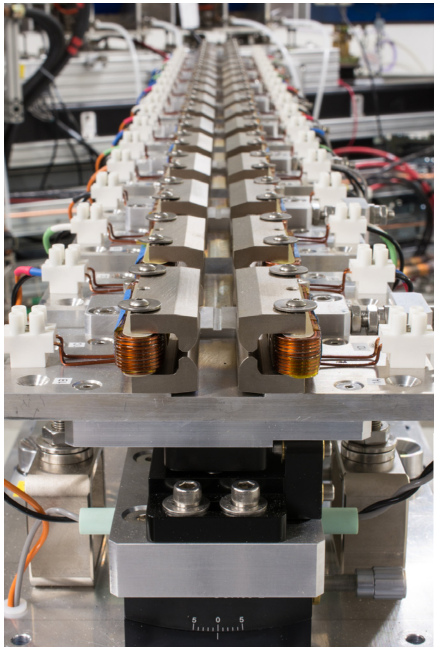
Experimental implementation - IOTA

- Two nonlinear inserts in the ring

Octupoles



DN magnet



- See our talks/posters (TUZBB6, WEXBA2, TUPLM08, etc.)

Why online modelling/correction?

Ideal world:

- Linear lattice is exact - set optimal sextupoles, get data

Reality:

- Limited LOCO accuracy – uncertainty on optics
- Unaccounted field errors/misalignments
- Small ring problems - path lengthening, geom. nonlinearities

Solutions:

- More LOCO, fancier correction (RDTs, detuning terms)
- Tweak the knobs and see what happens = “online correction”

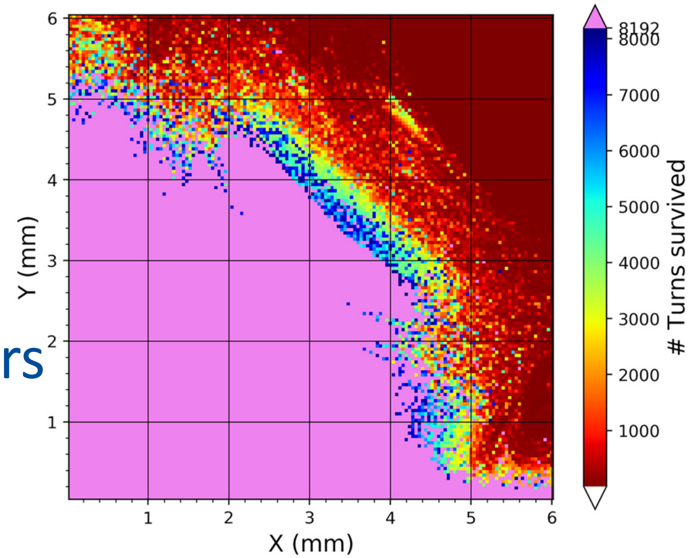
Online modelling/correction

- Typically deal with a ‘few’ knobs
  - i.e. 3-6 sextupole families + several parameters

- IOTA has significantly more knobs
  - Linear optics essential to performance, can’t be held ‘fixed’
  - No families – all magnets controllable, including 18 in insert

- Complications
  - Integrability/DA slow to measure (>10s/pt)
  - Costly simulations
  - Nonlinear ‘objective function’

~30 CPU-hours



Online modelling/correction

Proposed solution – 2 parts

- An online, ‘surrogate’ model (heuristic)
  - Predict results at close to optimal settings quickly

- Smart knob turning algorithm
  - That can sample sparsely and robustly

Online modelling

- Method of choice: neural network
  - Simple universal approximator
  - Easy to brute-force with more data
  - Shown to work in accelerators

- No goal of full accuracy,
  or to capture all physical properties
  - Just needs to be good enough

Online modelling

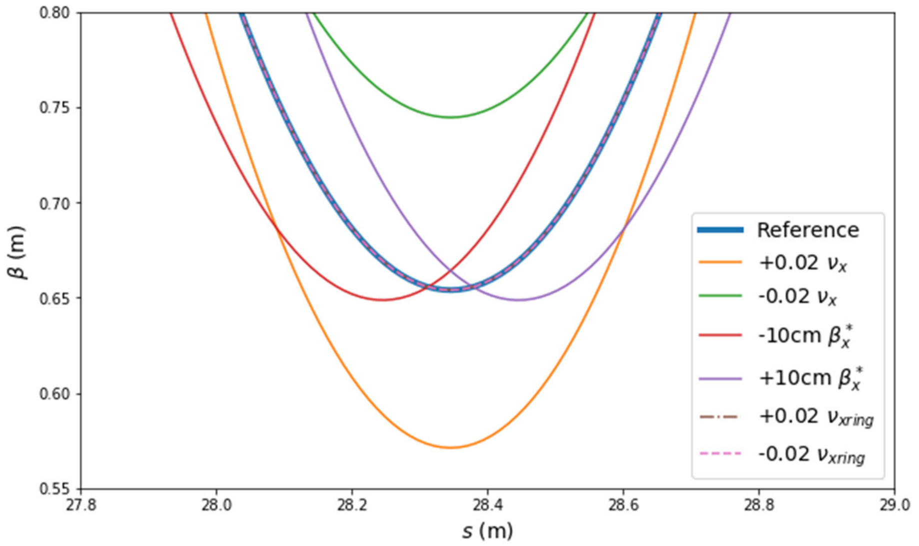
Knob selection

Typically, can use response matrices to simplify knobs (via SVD)

Except…we can't!

- Linear optics – need to satisfy special constraints
- Insert current – not useful to reduce at cost of tune shift

Parameter	Range	Knob count
$\mu_{x,y}$ (insert)	0.3 ± 0.02	2
$\mu_{x,y}$ (ring)	5.0 ± 0.02	2
β^* (z)	0 ± 10 cm	2
$\xi_{x,y}$	0 ± 0.1	2-6
κ (skew quads)	$0 - 0.01$	2-9
I_{NL}	$\pm 5\% (\sim 0.1A)$	9-18
H,V (correctors)	n/a	28



Online modelling

Implementation

- Simulated 10k DA scans + 10k sparse tracking grids
  - Easiest measurable features, but need a lot of storage space

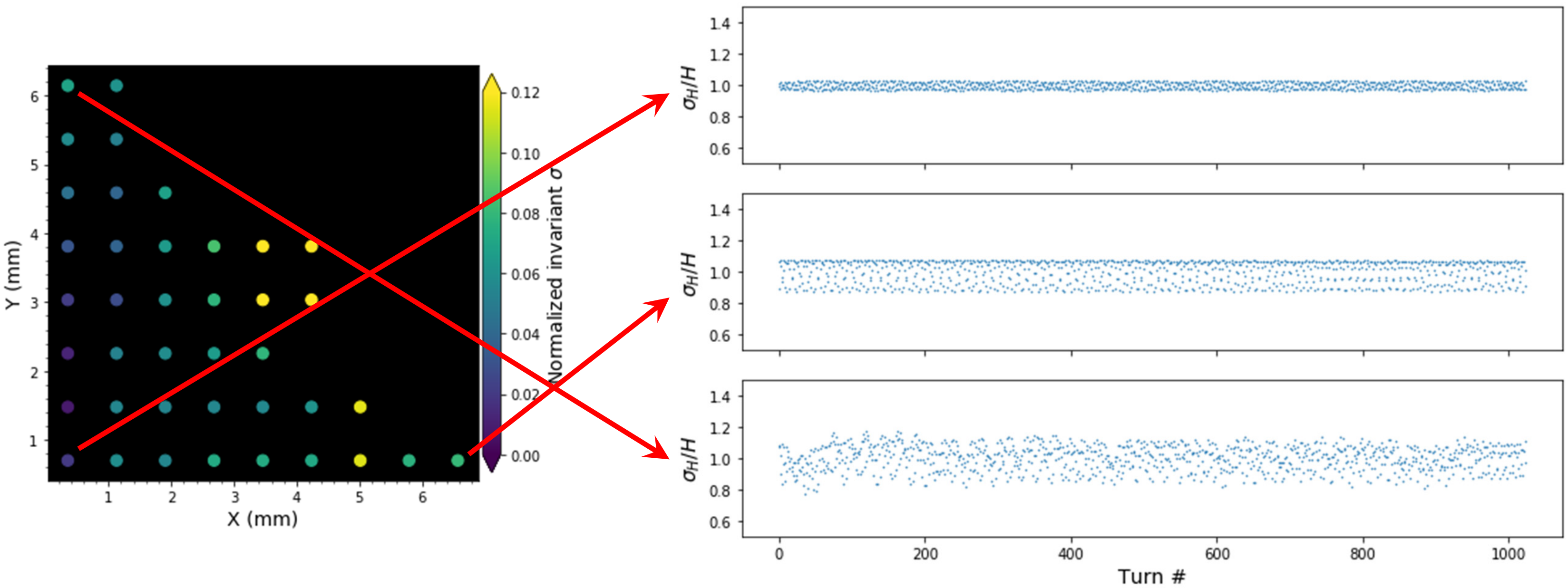
```
>>> Capacity Filesystem: project2 (GPFS)
---------------- ---------------- --------- ---------
youngkee         blocks (group)     447.55G   500.00G
                 files  (group)      505896    575000
---------------------------------------------------------
```

- Fed into standard MLP (fully-connected) network
  - Adam optimizer, ReLU activation, ramped schedule, etc.
  - 5 hidden layers x 25 nodes

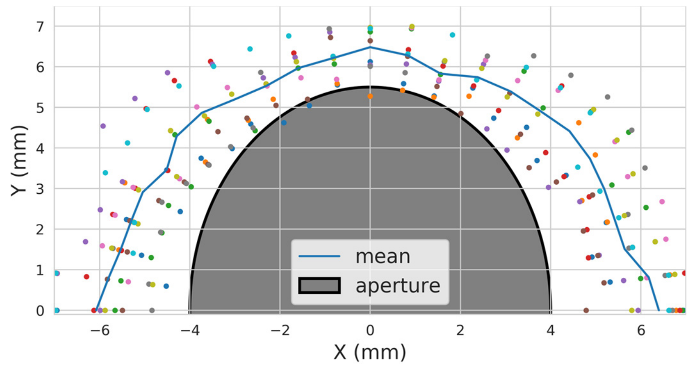
Online modelling

Data samples

Tracking



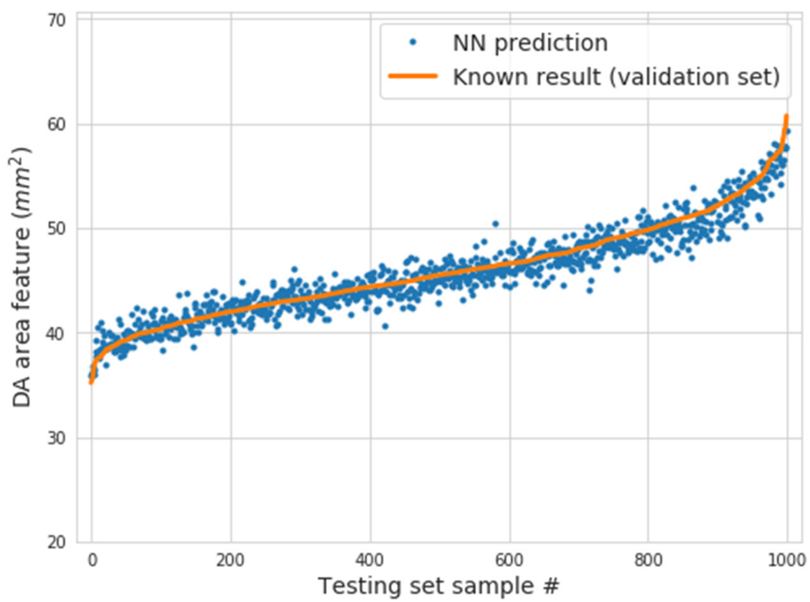
DA

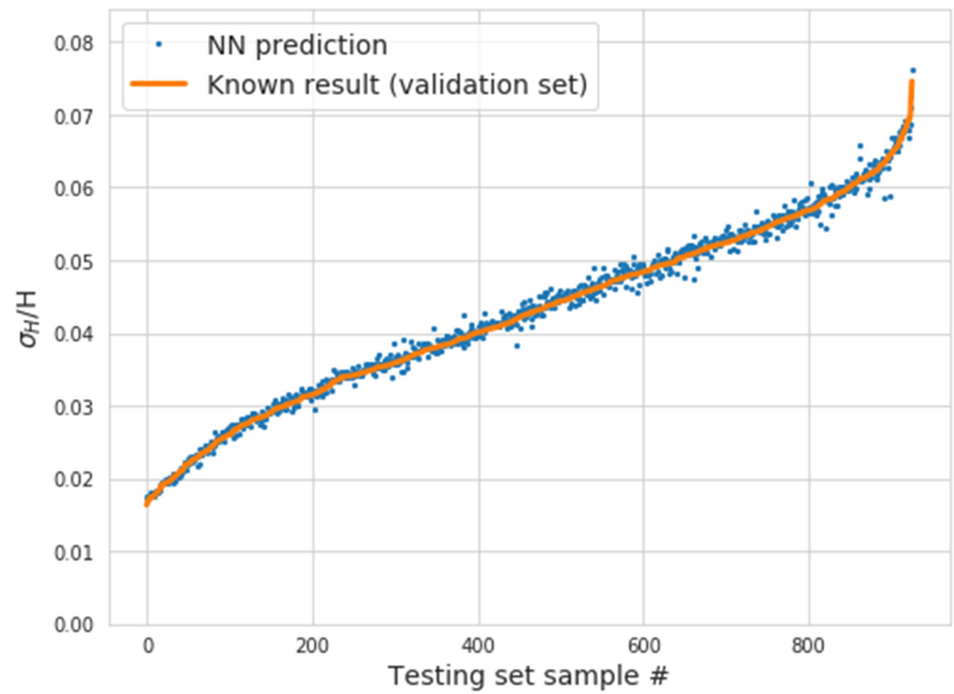


Online modelling

Performance

- Test set (10%) predictions





- Random misalignments/stochastic losses create noise for DA
- Doesn’t affect invariant jitter

Online correction

- Many available methods
  - Simplex
  - Genetic algorithms (MOGA)
  - Directed searches (Powell's, gradient descent)

- Goal: minimize number of evaluations
  - Cost is >10s/point

- Use Robust Conjugate Direction Search (RCDS) as basis
  - By X. Huang, SLAC
  - Simple algorithm – brackets search interval, fits for minimum along each 'conjugate' direction. Experimentally verified.

Online correction

- Naïve RCDS
  - Start with unit knob vectors as directions
- Better
  - Pre-calculate initial optimal directions
- Proposed
  - Dynamic heuristic to ‘nudge’ searches, based on past steps

Online correction

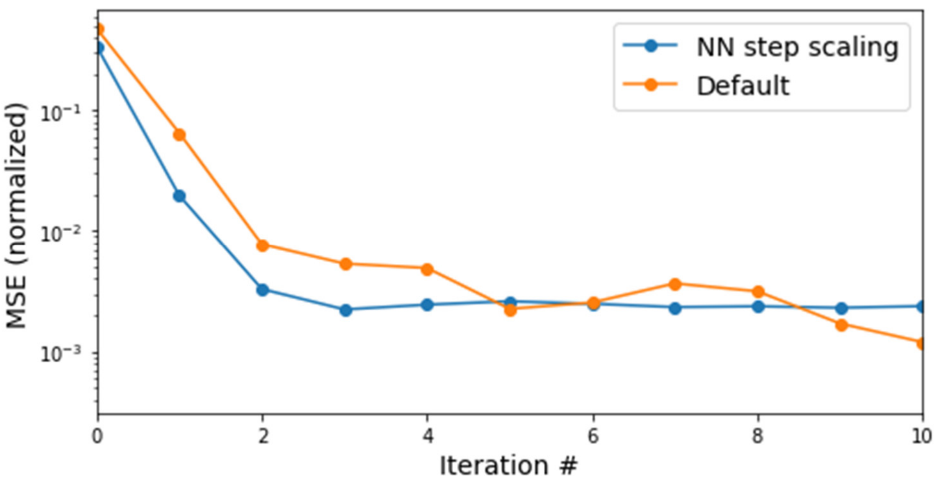
Implementation

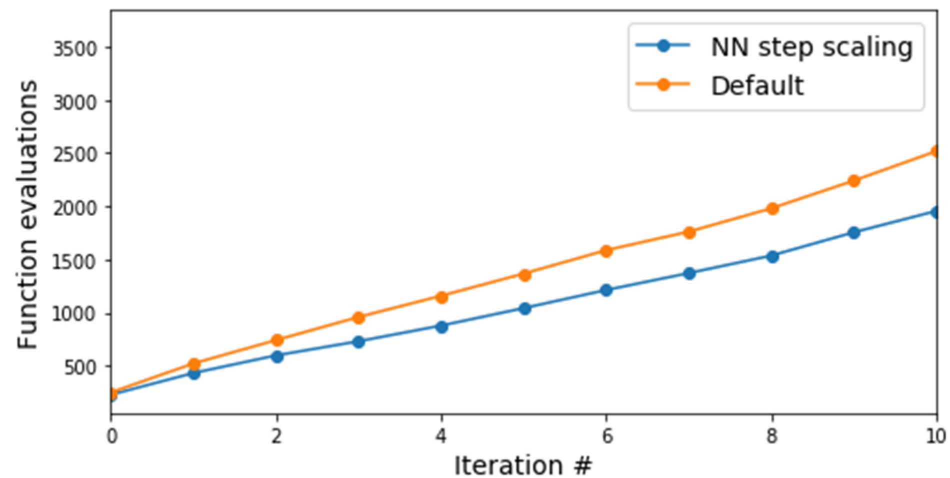
- Record naïve RCDS runs with different initial conditions
  - Can use online model vs expensive simulations!!!
- Train MLP or recurrent neural network (RNN)
  - Step sizes, directions, minima locations
- Run NN-RCDS
  - First few (1-3) steps same
  - Afterwards, modify search direction/step closer to next predicted vector
  - Fallback: modify directions based on parameter-space position

Online correction

Performance

- Test with 9-D space (3x online model) + simulated noise
- Compute parameter distance to optimum as ‘error’





- RNNs hard to get going, MLP reliably better
- Still too many evaluations…WIP

Experimental tests

- Had base RCDS test in run 1
  - Only DA objective (= survival after kick)
  - Octupoles got optimized away…but it technically worked

- Many operational hurdles
  - Hard to measure small DA changes reliably
  - Beam loss required auto-reinjection/state switch code
  - Live and automated BPM data analysis

Plans and summary

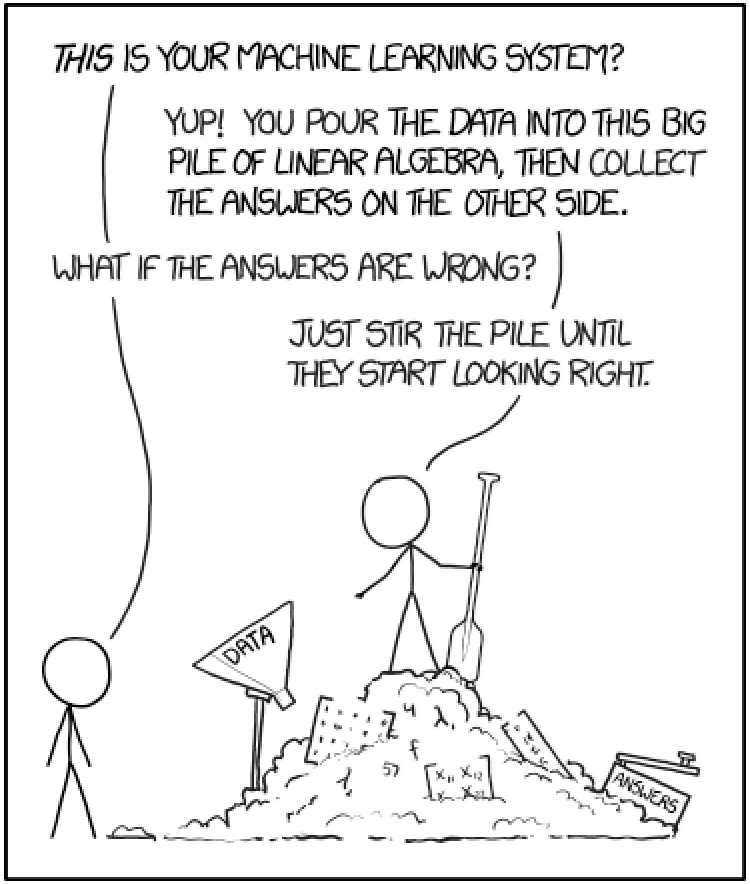
Studied NN-based online modelling and correction schemes to improve performance of NIO lattices. Promising simulation data.

Plans:

- (better) Experimental implementation
- Generative convolutional networks for 2D parameters (FMA)
- Transfer learning – can we only train last bit with expr. data?

Run 2 starting in September 2019 – expect results soon!

Thank you!
Questions?



xkcd.com/1838