

On-line Optimization of European XFEL with OCELOT



S. Tomin, [I. Agapov](#), W. Decking, G. Geloni,
M. Scholz, I. Zagorodnov
ICALEPCS 2017, Barcelona
8-13 October 2017

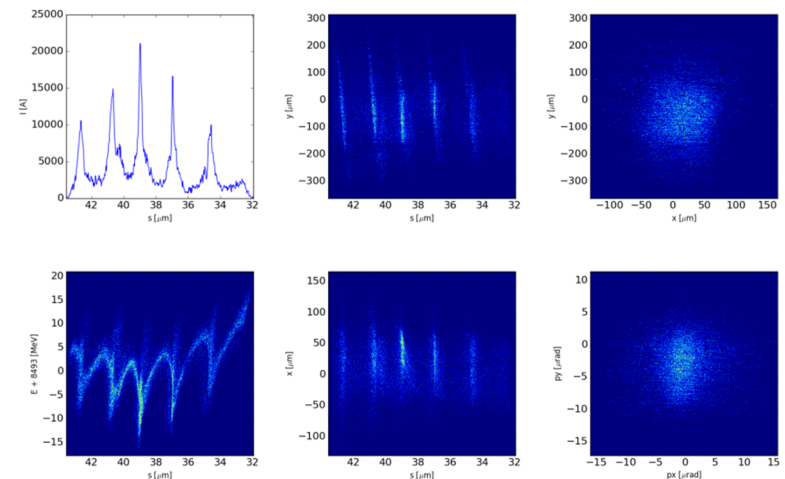
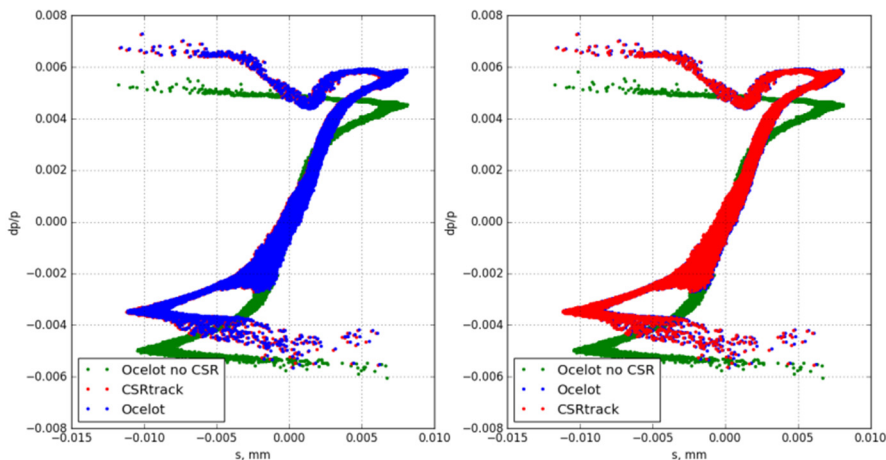
OCELOT concept

- Started as simulation project (Undulator radiation, FEL) at European XFEL
- FELs are notorious for difficulty in model-to-measurement comparison (short pulse length, small emittances, synchronization - diagnostics non-trivial) and need of manual retuning. This puts a limit on the merit of simulations (such as taper optimization).
- Branched into more on-line control-oriented development
- Resulting concept of generic simulation/controls framework/library, featuring

Electron beam physics models X-ray physics models On-line Controls

Advanced GUI

Data-centric (statistics, ML)



OCELOT technical concept

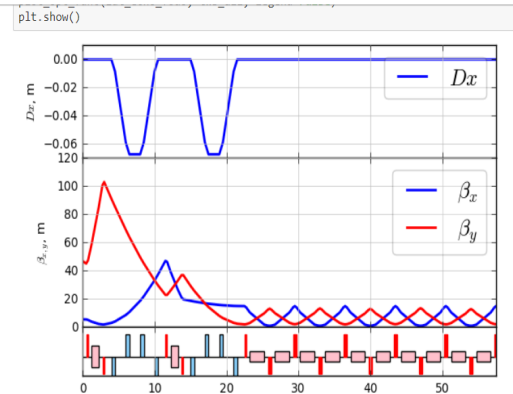
- Everything in python. Focus on simplicity
 - No API to other languages
 - Everything is a python object, no (additional) input language, no parsing, no data standards, free serialization. Implement only physics
 - Parsers however easily implementable (included ELEGANT, MAD-X, GENESIS)
- Abundant numerical libraries
 - *Numpy, scipy, matplotlib, mpi4py*
 - *Scikit-learn* for machine learning
- Not a standalone application but a python library + a set of tools
- Code optimized when possible, but simplicity and usability has priority over speed
- Modular physics process architecture (concept similar to *Geant4*)
- Open source

OCELOT details

- Single-particle electron optics
- Beam dynamics in linacs
 - Wakes, CSR, Space charge
 - 2nd order tracking
- X-ray optics
 - Fourier optics
 - Dynamic diffraction (Bragg)
- Synchrotron radiation
 - UR/SR solvers
- FEL calculations
 - Integrates Genesis
- Controls interface and optimization tools

- Tutorials on <https://github.com/ocelot-collab/ocelot>:
 - [Tutorial N1. Linear optics. Web version.](#)
 - [Tutorial N2. Tracking. Web version.](#)
 - [Tutorial N3. Space Charge. Web version.](#)
 - [Tutorial N4. Wakefields. Web version.](#)
 - [Tutorial N5. CSR. Web version.](#)
 - [Tutorial N6. RF Coupler Kick. Web version.](#)
 - [Tutorial N7. Lattice design. Web version.](#)

jupyter
nbviewer



Step 3. Matching section

```
In [9]: Q1 = Quadrupole(l=0.3, k1=1)
Q2 = Quadrupole(l=0.3, k1=1)
Q3 = Quadrupole(l=0.3, k1=1)
Q4 = Quadrupole(l=0.3, k1=1)

m1 = Marker()
m2 = Marker()
dm = Drift(l=1.5)
match_sec = (m1, dm, Q1, dm, Q2, dm, Q3, dm, Q4, dm, m2)

lat_m = MagneticLattice(match_sec[::-1])
```

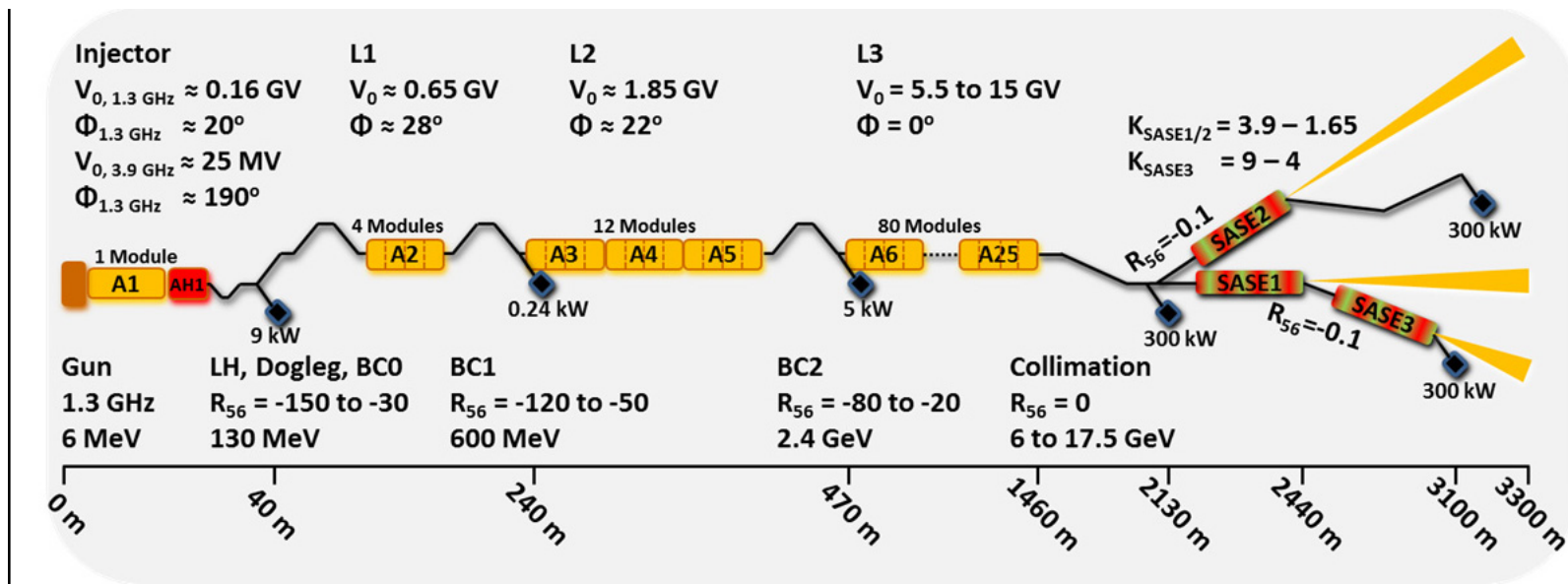
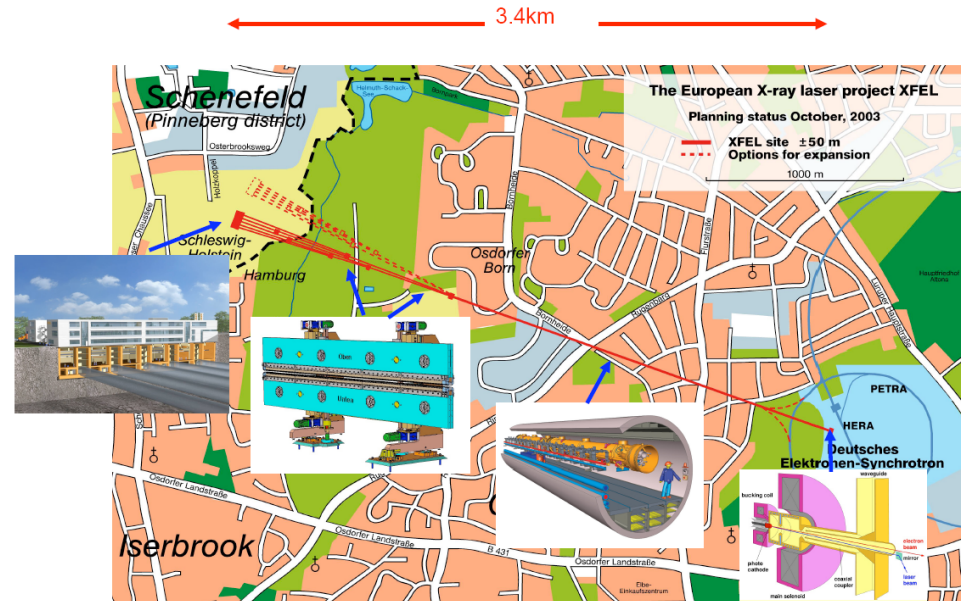
On GitHub <https://github.com/ocelot-collab/ocelot>

I. Agapov et al., NIM A. 768 2014

The European XFEL in a nutshell

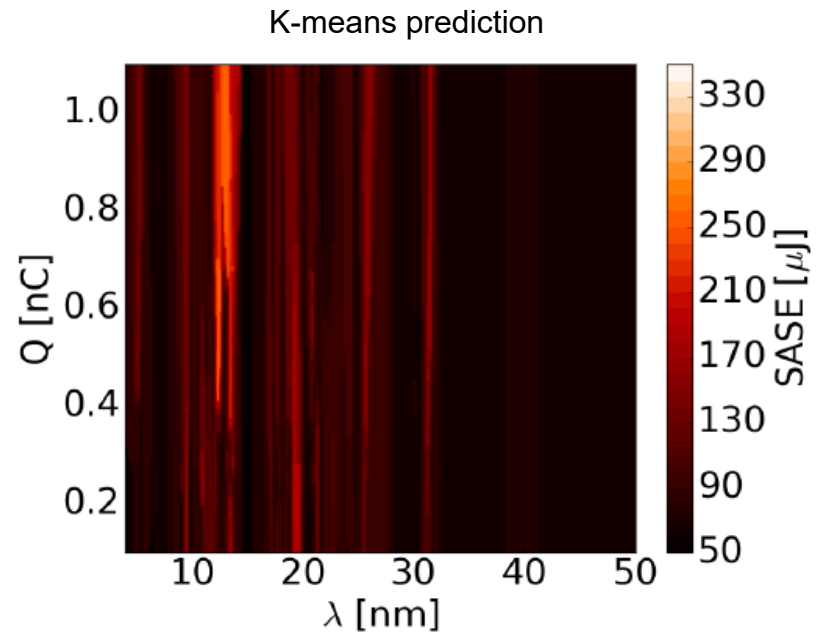
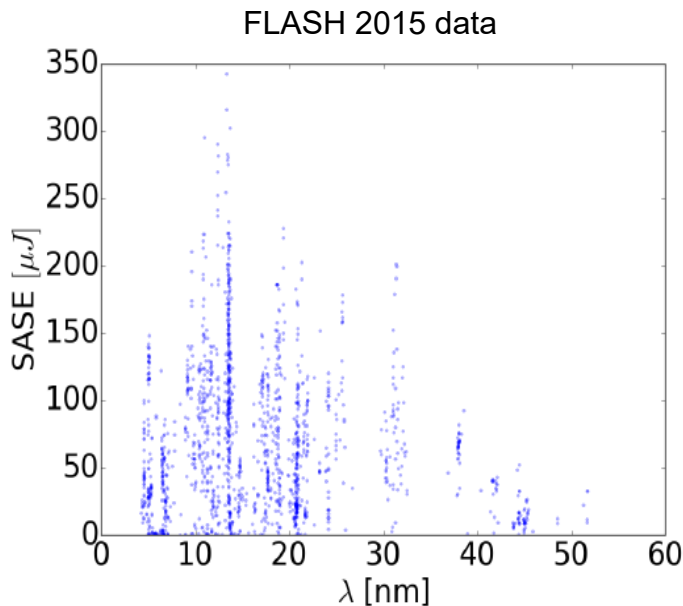
Driven by superconducting linac

- Pulse repetition rate 10 Hz
- Pulse length 600 ms
- #bunches per pulse 2700 -
- Bunch length (compressed) 2-180 fs (FWHM)
- Bunch charge 0.02-1 nC
- Slice emittance 0.4-1.0 mm mrad
- Slice energy spread 4-2 MeV



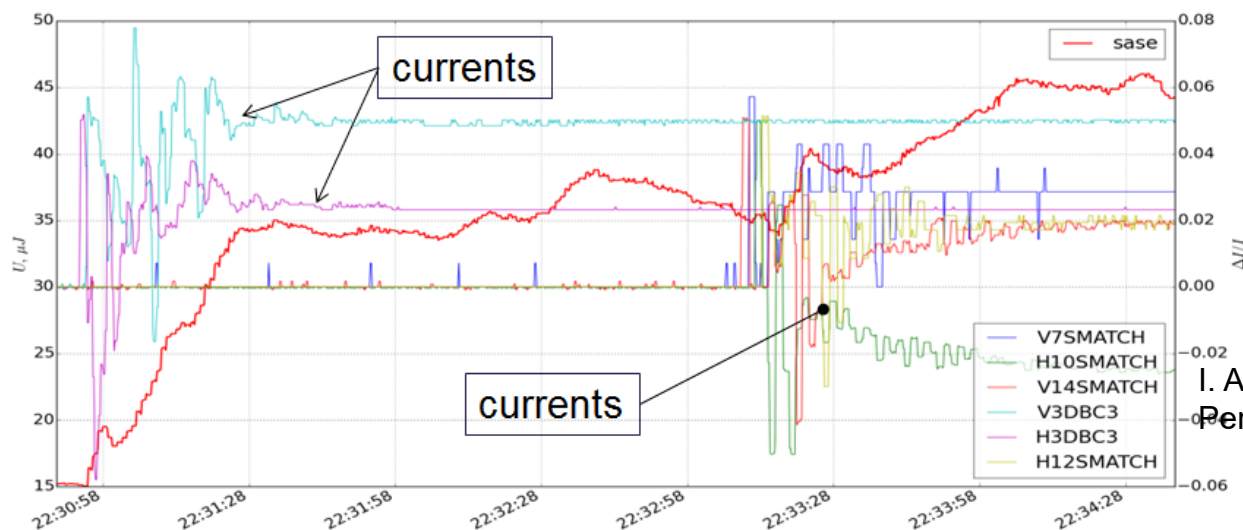
Optimization and tuning

- Even when major technical systems work well, getting close-to-design photon pulse parameters requires manual fine-tuning
- Tuning done with the gun, linac RF voltages and phases, orbit, optics, undulator gaps, phase shifters,
- Hundreds of free tuning parameters
- Requires time and expertise, and the results varies depending on operator, time investment and machine conditions
- With increasing complexity and pressure on availability, more automation needed



Optimizer - FLASH

- Undulator orbit, optics matching and compression tweaked empirically at FLASH for SASE tuning
- Implemented a tool that optimizes SASE with a set of actuators with a functional minimization instead of manually
- Major difficulty: parameter setup (signal averaging, device boundaries) and error handling (hardware and software hiccups)

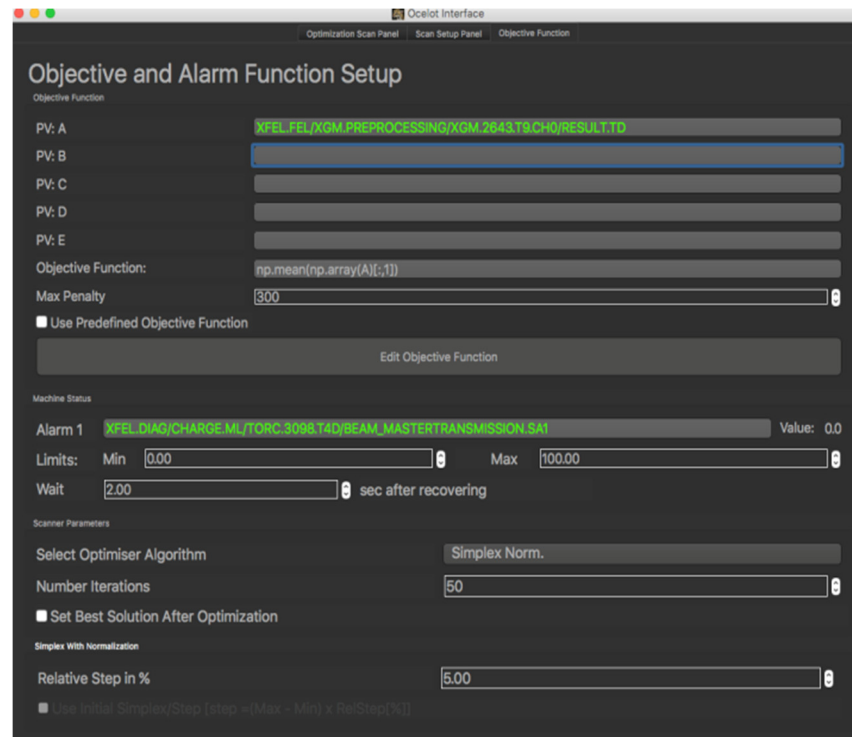
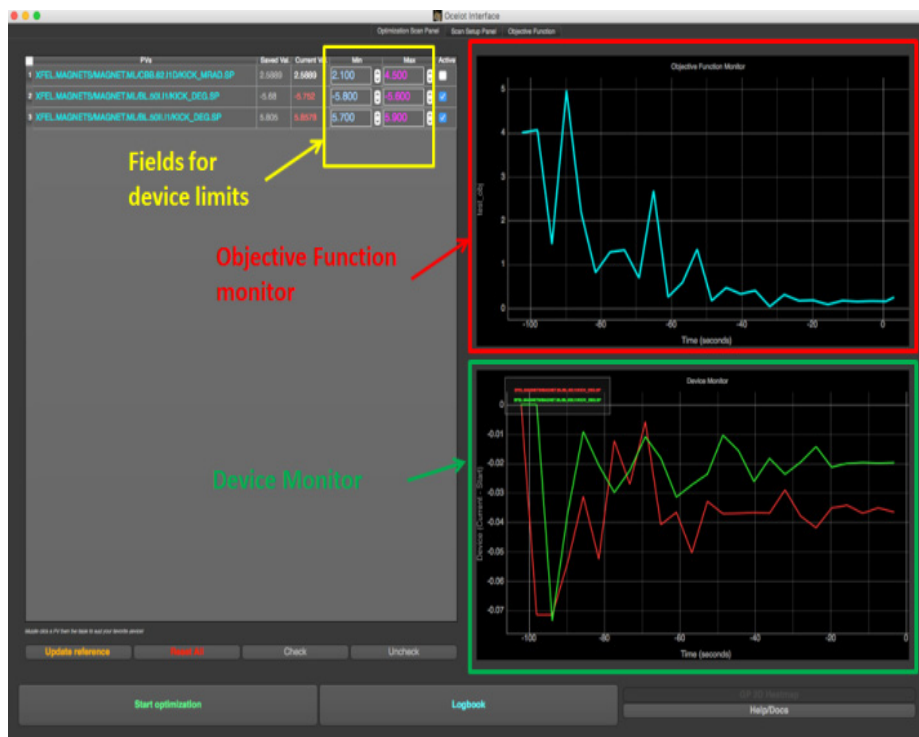


Example with 4 correctors
 $\lambda = 10.4 \text{ nm}$

I. Agapov et al. Statistical Optimization of FEL Performance. IPAC15.

Generic optimizer - XFEL.EU

- Sequence of actions / decision tree implemented initially
- However development shifted towards making the tool more universal and useful for ad-hoc tasks (important for commissioning)
- Concentrated on single actions and developed generic GUI
- Deployed for European XFEL



Dispersion correction - XFEL.EU

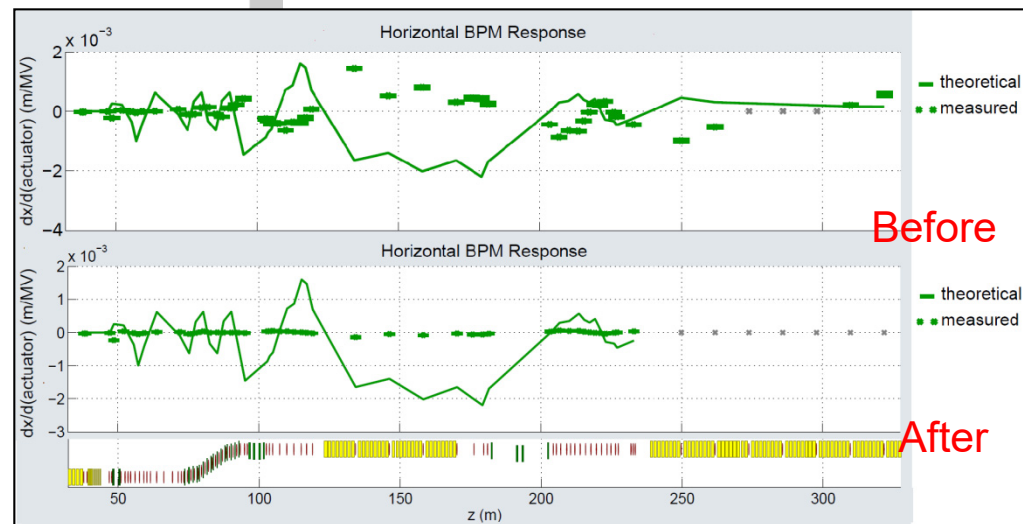
Alternative to response-matrix-based correction

European XFEL

Status: Preparation for undulator BBA: Transport beam to T4D at 6/8/10/12 GeV and p
News: First Lasing in SASE1 in the night from May 2nd to May 3rd

Logbook entry: /XFELelog/data/2017/04/27.01_a
27.01.2017 20:22 S.Tomin Horiz. dispersion correction.
Was used only one actuator CBB.62.I1D.
Objective function was respond BPMs signal on RF voltage changing (BPMA.92.I1, BPMF.95.I1, BPMF.95.I1):

OCELOT params :
obj func: A : predefined
CBB.62.I1D : KICK_MKAD.SP 3.07003888221 --> 2.73253888221
iterations : 12
delay : 1.6000000000000003
START-->STOP : -0.83154440072 --> -0.11268520107



Dispersion compensation from I1D dump magnet
(compensation with one actuator)

Beam losses and orbit - XFEL.EU

Minimization of the beam losses and keeping reasonable orbit in the TLD

European XFEL
News: [First Lasing in SASE1 in the night from May 2nd to May 3rd](#)

2017
▲

52
▼

19
▼

09.05_1
▼

08.05_1
▼

08.05_1
▼

18
▼

17
▼

16
▼

15
▼

14
▼

13
▼

12
▼

11
▼

10
▼

09
▼

08
▼

<
>

Logbook entry: [/XFELelog/data/2017/13/29.03_n](#)

30.03.2017 01:56

after hour of the steering the beam in Dump to rid of BLM alarms we used Optimizer with BLMs signals as a target function (see below) and got transmission without BLM alarms

```

obj func: A : BLM.2112.TLD/SIGNAL.TD
obj func: B : BLM.2117.TLD/SIGNAL.TD
obj func: C : BHM.U.2122.TLD/SIGNAL.TD
obj func: D : BHM.R.2122.TLD/SIGNAL.TD
obj func: E : BHM.L.2122.TLD/SIGNAL.TD
obj func: expr: -(np.max(np.array(A)[:1]) + np.max(np.array(B)[:1]) + np.max(np.array(C)[:1]) +
np.max(np.array(D)[:1]) + np.max(np.array(E)[:1]))
dev : CFX.1894.TL/KICK_MRAD.SP 0.128220755283 --> 0.138468939487
dev : CFY.1910.TL/KICK_MRAD.SP -0.015373917147 --> -0.0287091451133
iterations : 29
delay : 1.9000000000000004
START-->STOP : -1.77062509954 --> -1.8136251159
                    
```

OCLOT Optimization

View Current
Hide Untagged

All Errors
All ToDo's

Safety
Schedule

Access Requests
Beam Time Request

XFELBau Logbook
Logbook Search

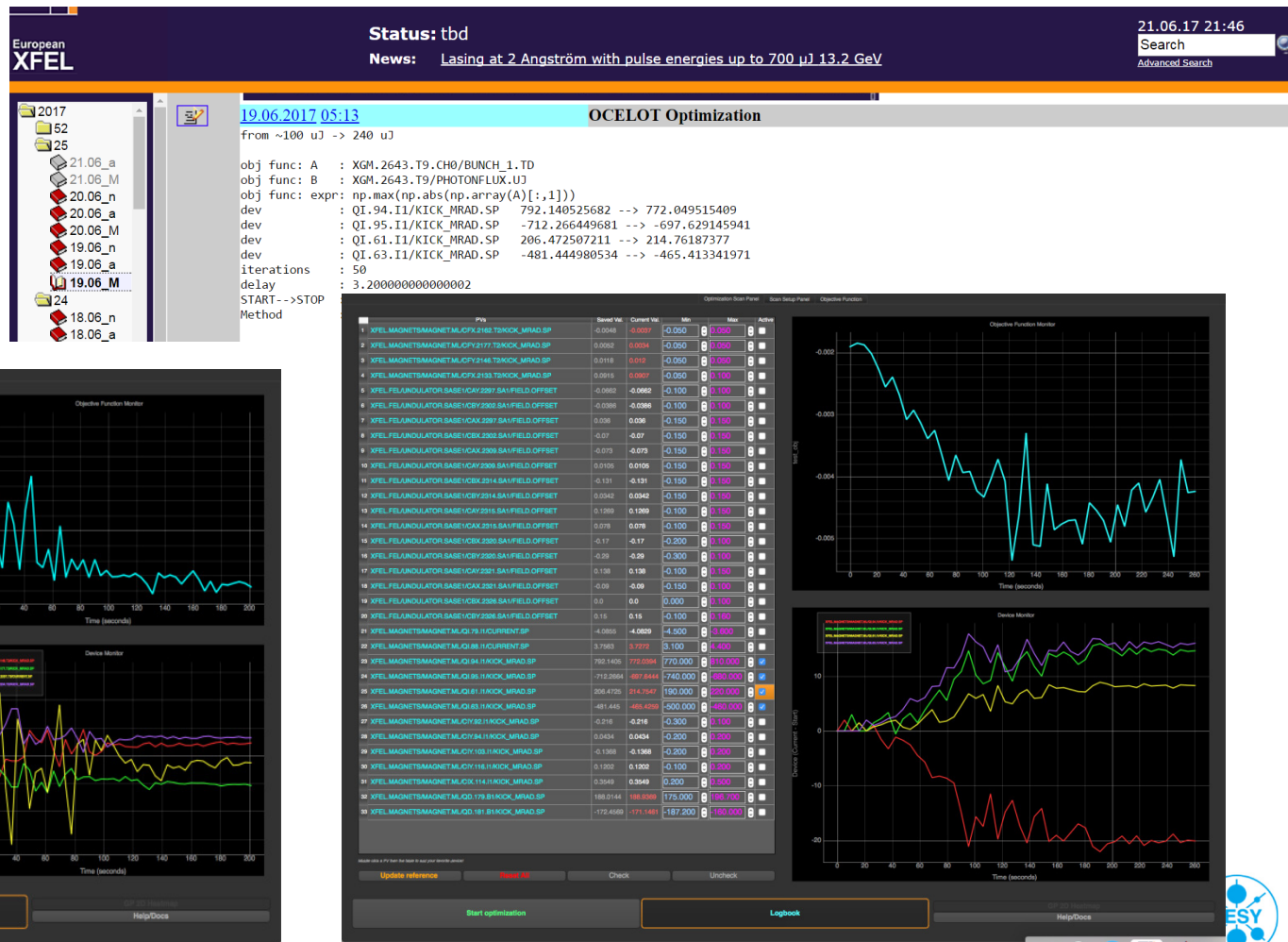
Logbook Help
XFELeu

XFELdesy
XFELML2



SASE optimization - XFEL.EU

- Example: 4 quads in the Injector DogLeg SASE level was increased from 100 uJ up to 240 uJ
- Routinely used with 4 launch steerers



OCELOT orbit correction tool - XFEL.EU

- OCELOT electron optics model for XFEL in place, tools using response matrices can be implemented in the same framework
- Response-matrix (SVD) tool for orbit and dispersion correction implemented and in operation



Adaptive feedback - XFEL.EU

- Optimizer cannot be effectively used during beam delivery
- Implemented “adaptive feedback”: Average orbits with best SASE and correct orbit to that value (SVD)
- Works great during user operation



Idea from: G. Gaio, M. Lonza, Automatic FEL Optimization at FERMI, Proc. of ICALEPCS2015



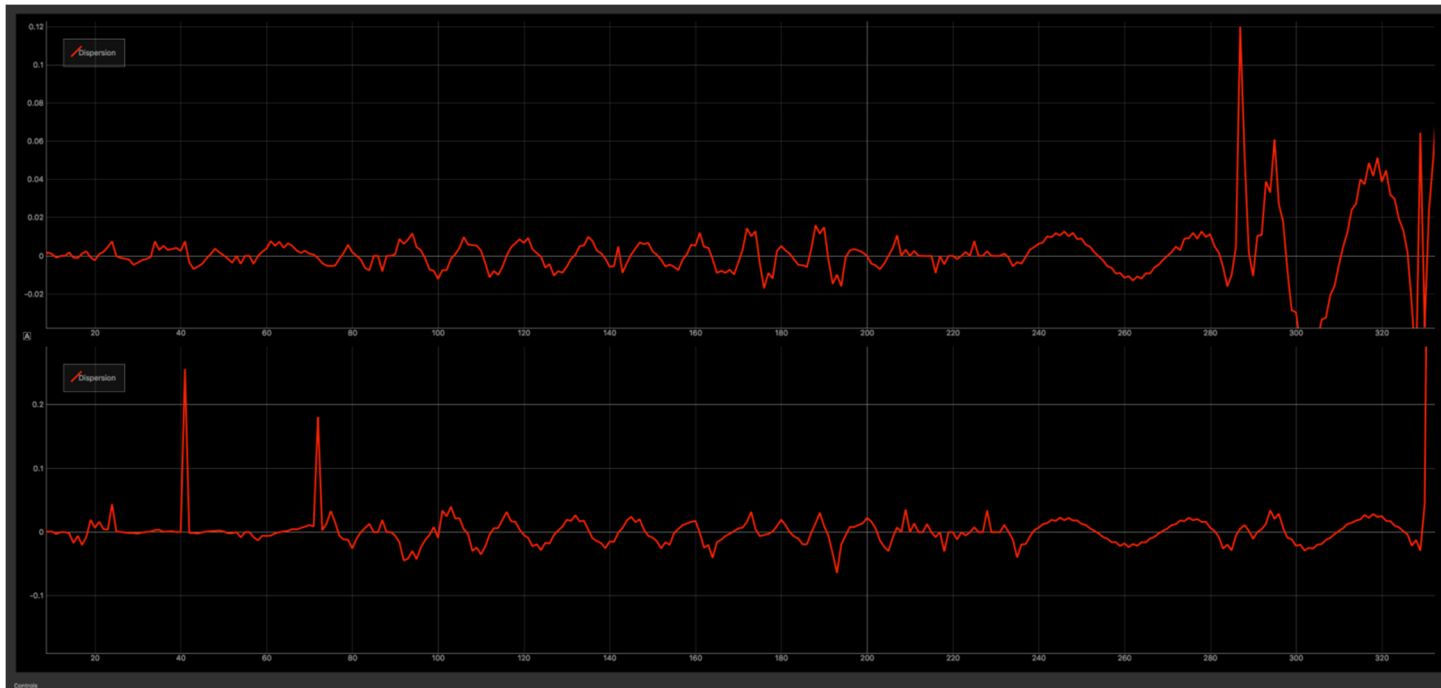
OCELOT on-line model - XFEL.EU

- Read machine parameters (quadrupoles and cavities) and measured beta-functions at a screen (I1, BC1, BC2), compute optics
- Change settings in flight simulator mode
- Possibilities of using such tool for tuning being explored



Correlation tool – XFEL.EU

- In the spirit of data-centric software suite
- Real-time correlations give machine parameters such as dispersion, coupling etc.

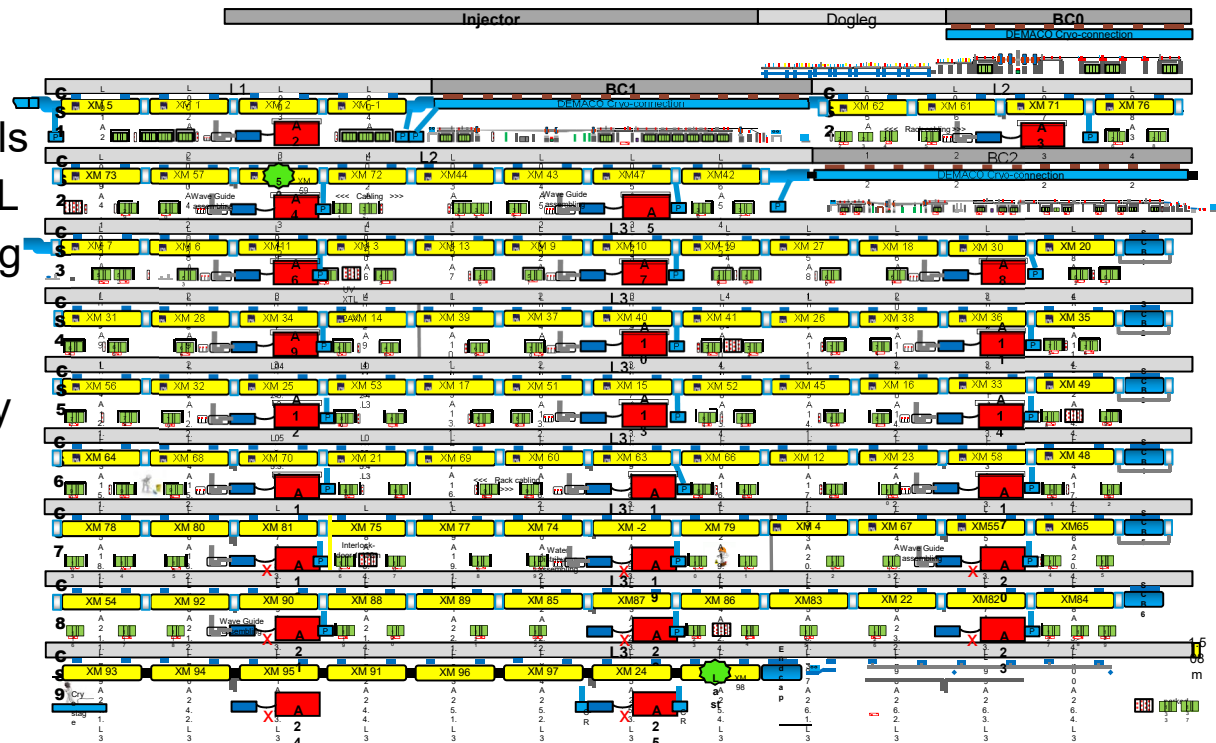
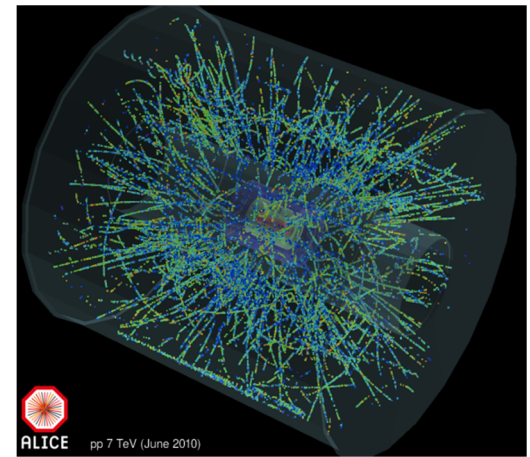


Horizontal
dispersion

Vertical
dispersion

Dealing with data complexity

- ML tools have been long used in astronomy and HEP for pattern recognition/classification problems (data rates prohibitive for visual inspection)
- Boosting advanced data handling/ML techniques tools in accelerator controls for machines like the XFEL is important due to growing complexity
- Failure prediction probably most natural application
- Possibilities of using such methods for tuning being explored



Conclusion and Outlook

- Empirical optimization methods successfully used for European XFEL commissioning and operation
- Corresponding software tools in place, open source and available through OCELOT
- Challenges to reach full tuning automation include
 - Exploiting beam physics models to restrict search space
 - Exploiting statistics and machine learning

<https://github.com/ocelot-collab/ocelot>

