Modeling and optimization of the European XFEL with OCELOT

Sergey Tomin
DESY

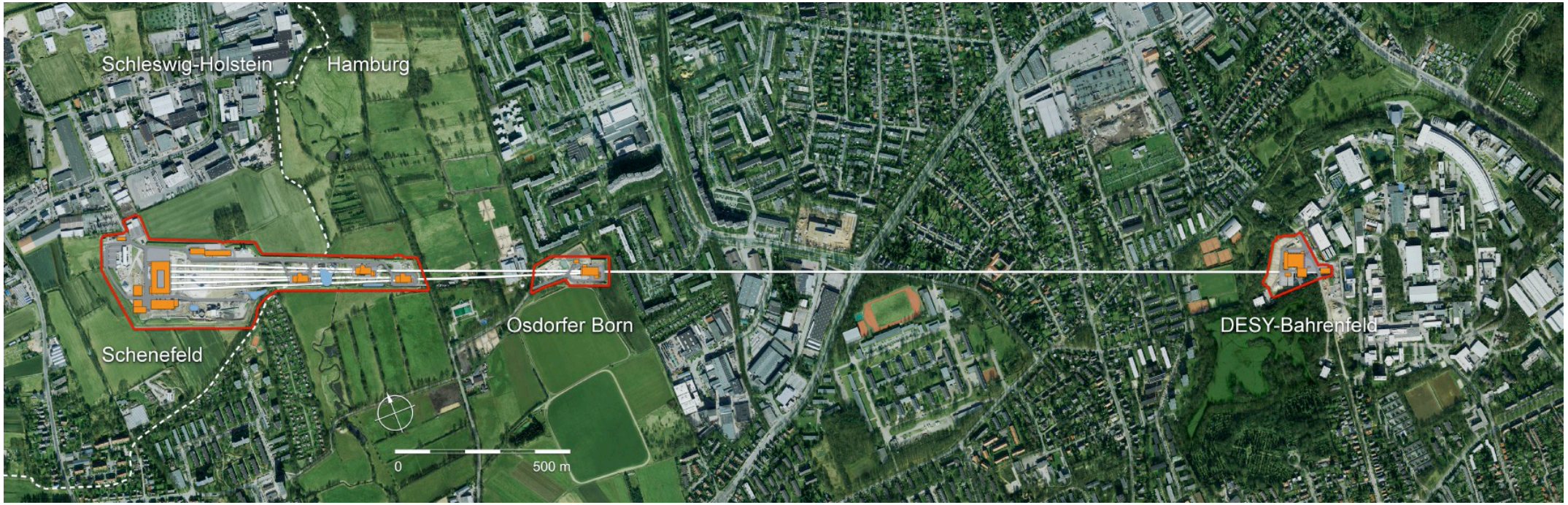
CERN BE Seminar, 28 August 2020

Outline

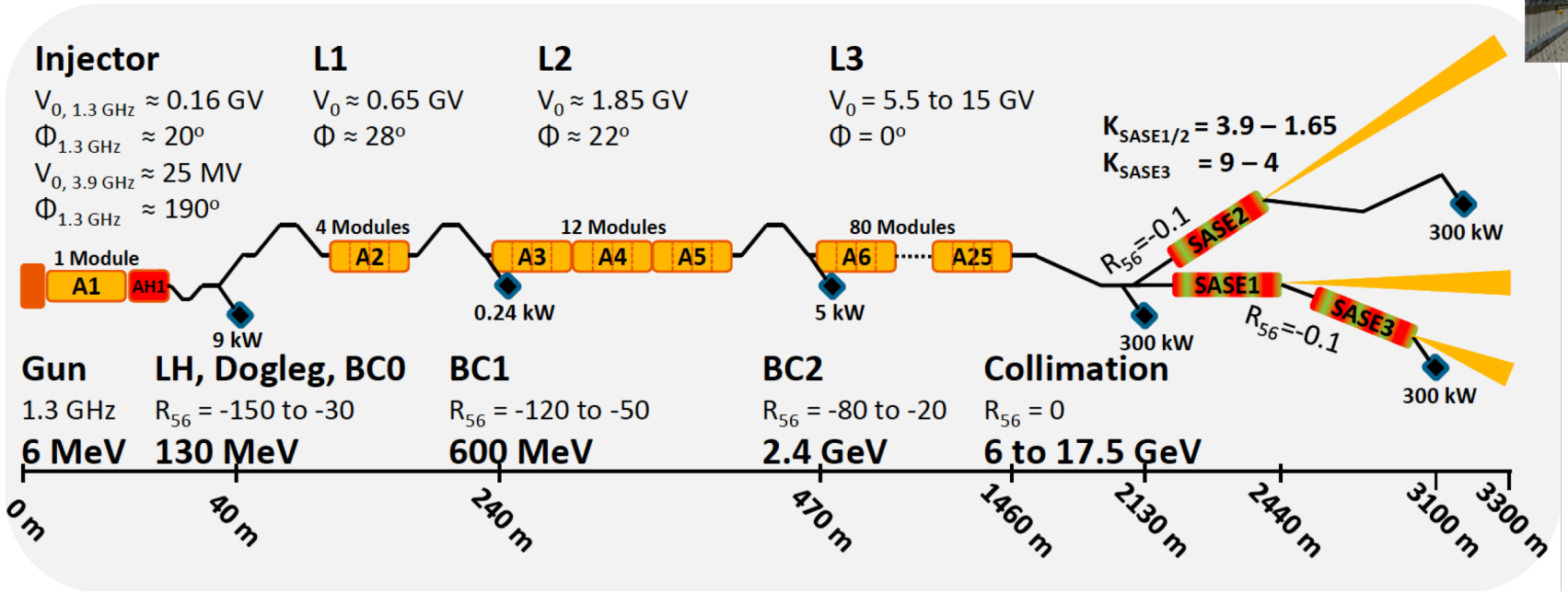
- Introduction and motivation
- OCELOT structure
- Modeling with OCELOT
- Optimization with OCELOT
- Conclusion

The European XFEL

- The 3.4 kilometre-long facility runs (in underground tunnels) from the DESY campus in Hamburg to the town of Schenefeld



Accelerator Overview



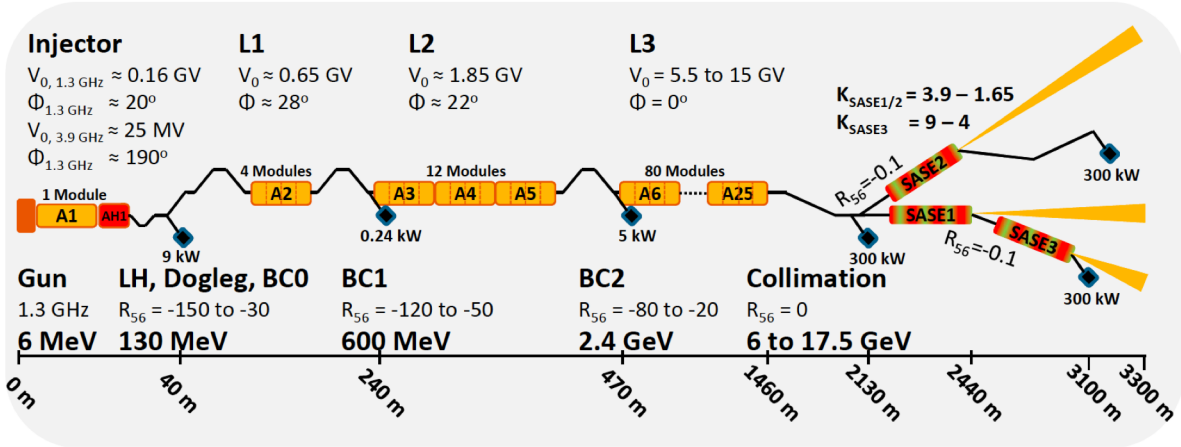
Nominal parameters

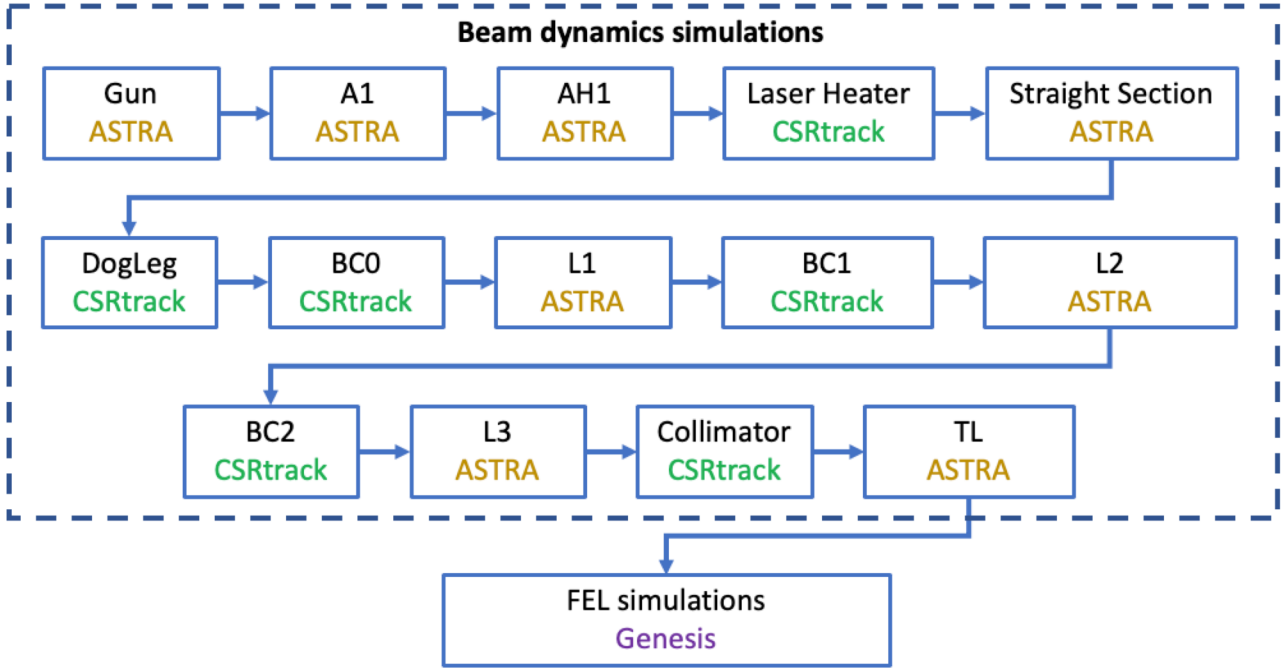
- Electron beam energy up to 17.5 GeV
- Pulse rep. rate 10 Hz
- Bunches per pulse 2700
- Intratrain rep. rate 4.5 MHz
- Bunch charge 0.02 – 1 nC

- SC accelerator is able to produce up to 27000 electron bunces per second in burst mode with 10 Hz
- Electron bunches in a single pulse are distributed by a fast kicker system to three SASE undulators

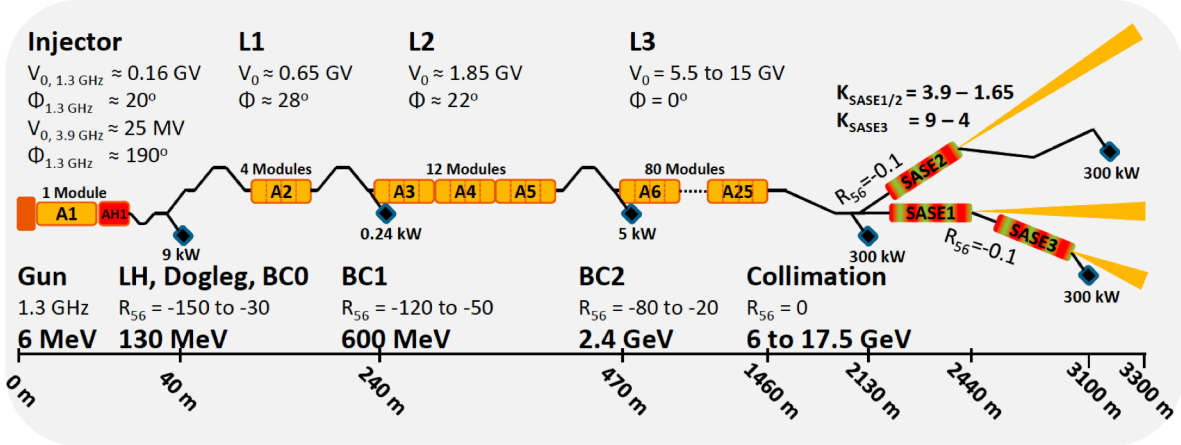
W. Decking et al, "Commissioning of the European XFEL", https://doi.org/10.18429/JACoW-IPAC2017-MOXAA1

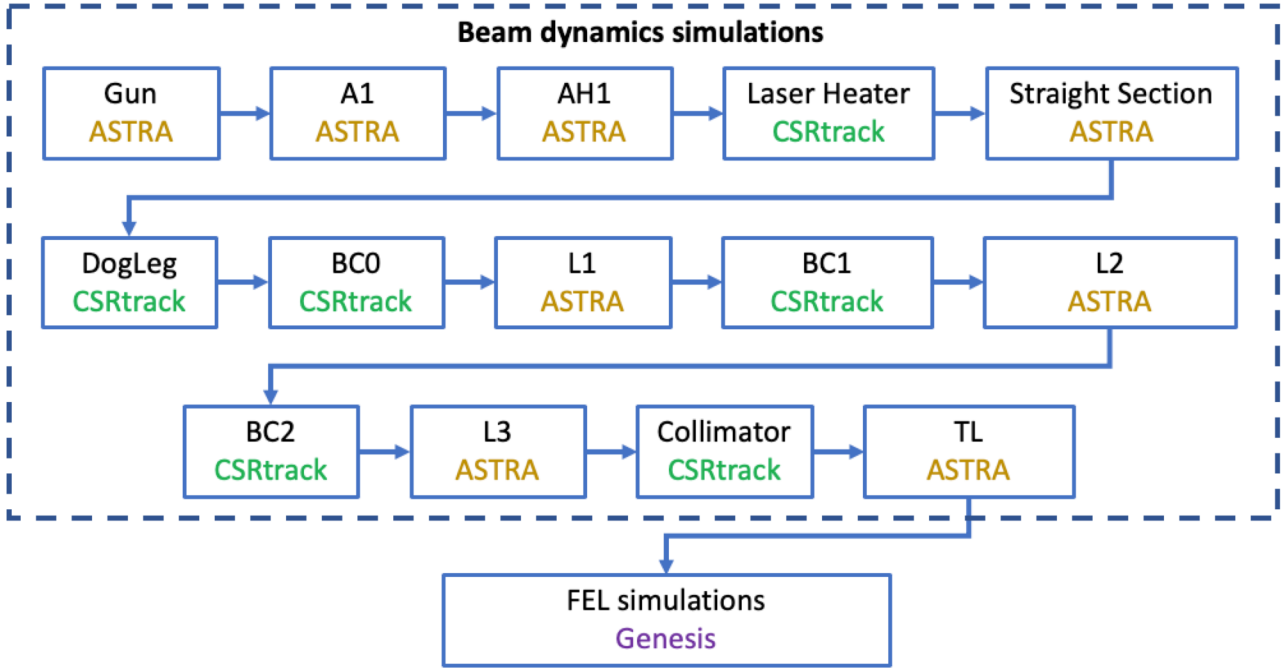
Motivation

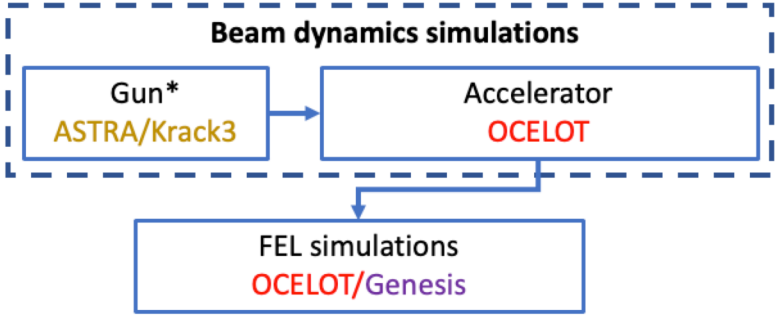

Injector
$V_{0,\,1.3\,GHz} \approx 0.16$ GV
$\Phi_{1.3\,GHz} \approx 20^o$
$V_{0,\,3.9\,GHz} \approx 25$ MV
$\Phi_{1.3\,GHz} \approx 190^o$
L1
$V_0 \approx 0.65$ GV
$\Phi \approx 28^o$
L2
$V_0 \approx 1.85$ GV
$\Phi \approx 22^o$
L3
$V_0 = 5.5$ to 15 GV
$\Phi = 0^o$
$K_{SASE1/2} = 3.9 - 1.65$
$K_{SASE3} = 9 - 4$
1 Module
A1
AH1
4 Modules
A2
12 Modules
A3
A4
A5
80 Modules
A6
A25
SASE2
SASE1
SASE3
R_{56}=-0.1
R_{56}=-0.1
9 kW
0.24 kW
5 kW
300 kW
300 kW
300 kW
Gun
1.3 GHz
6 MeV
LH, Dogleg, BC0
R_{56} = -150 to -30
130 MeV
BC1
R_{56} = -120 to -50
600 MeV
BC2
R_{56} = -80 to -20
2.4 GeV
Collimation
R_{56} = 0
6 to 17.5 GeV
0 m
40 m
240 m
470 m
1460 m
2130 m
2440 m
3100 m
3300 m


Beam dynamics simulations
Gun
ASTRA
A1
ASTRA
AH1
ASTRA
Laser Heater
CSRtrack
Straight Section
ASTRA
DogLeg
CSRtrack
BC0
CSRtrack
L1
ASTRA
BC1
CSRtrack
L2
ASTRA
BC2
CSRtrack
L3
ASTRA
Collimator
CSRtrack
TL
ASTRA
FEL simulations
Genesis

Motivation







2017

The current setup:

- Faster
- More easy to modify
- Can be run by non experts

2014

OCELOT structure

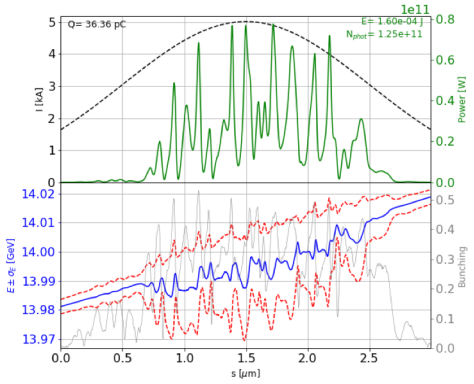
- Everything in **Python**. Focus on simplicity. Implement only physics
- Open source (On GitHub **https://github.com/ocelot-collab/ocelot**)

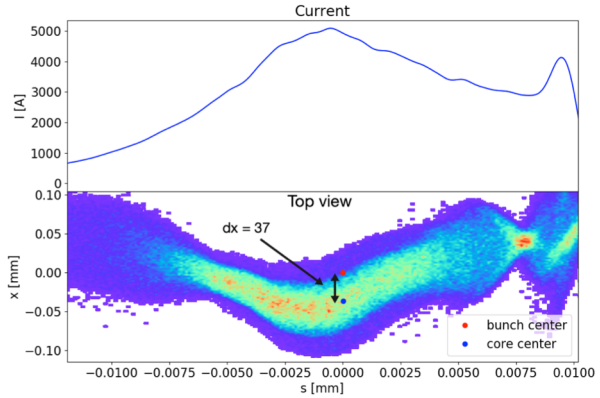
Data sets

Photon field simulation
- FEL simulations (genesis)
- Spontaneous radiation
- Wavefront propagation
- FEL estimator

Charged Particle Beam Dynamics (CPBD) module (linacs, rings)

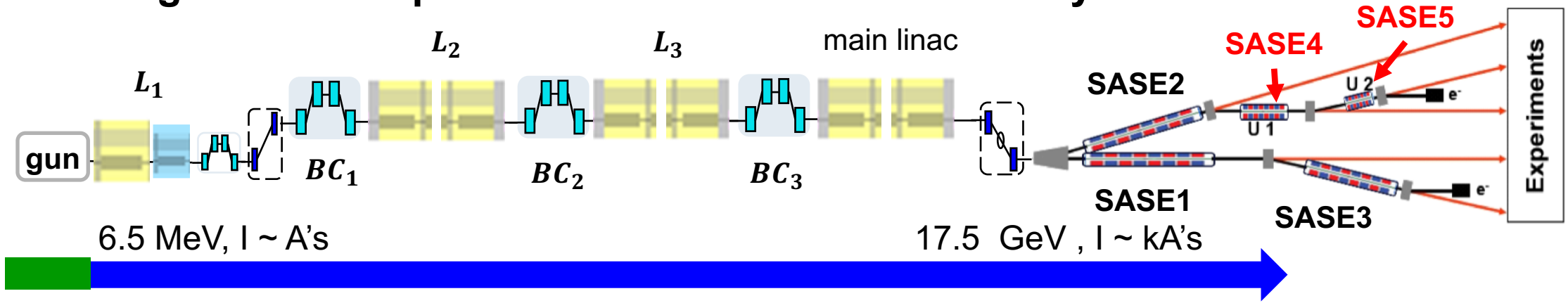
Online beam control
- Orbit correction
- Adaptive FB
- Optimizer



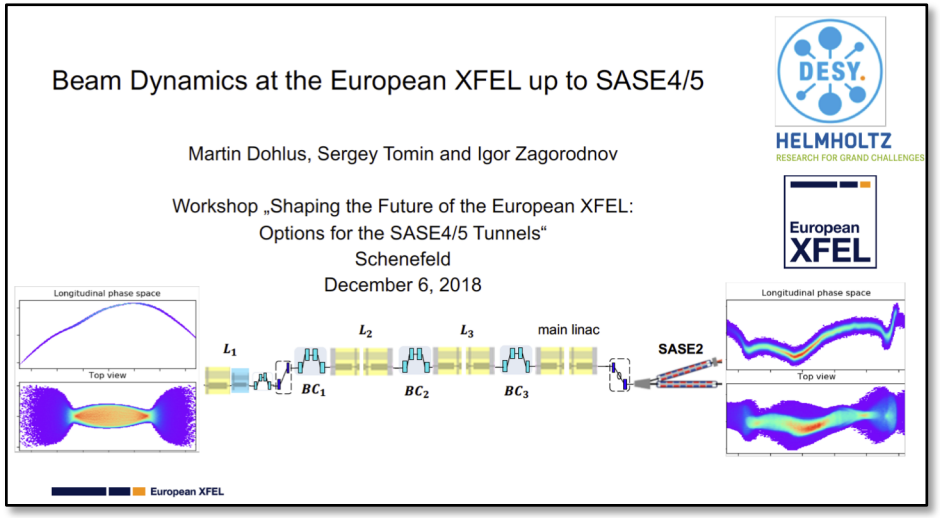


Modeling of the European XFEL with OCELOT

Modeling of the European XFEL with OCELOT. Beam dynamics simulations



Krack3 / ASTRA

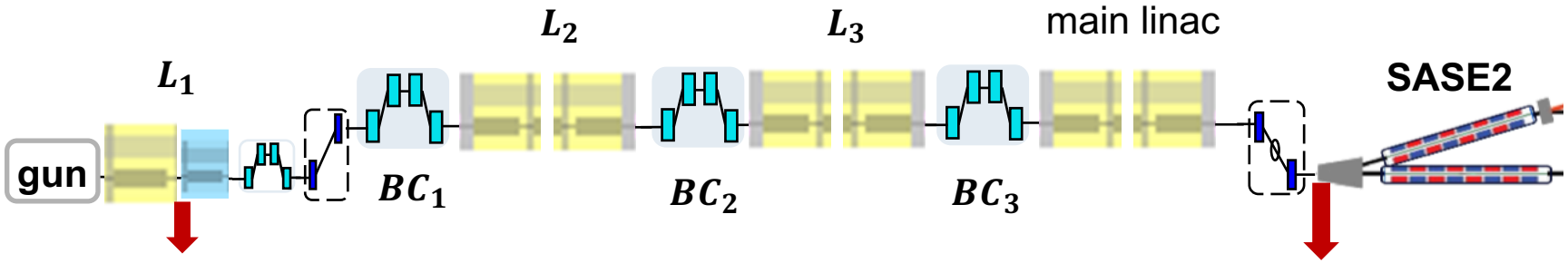


OCELOT

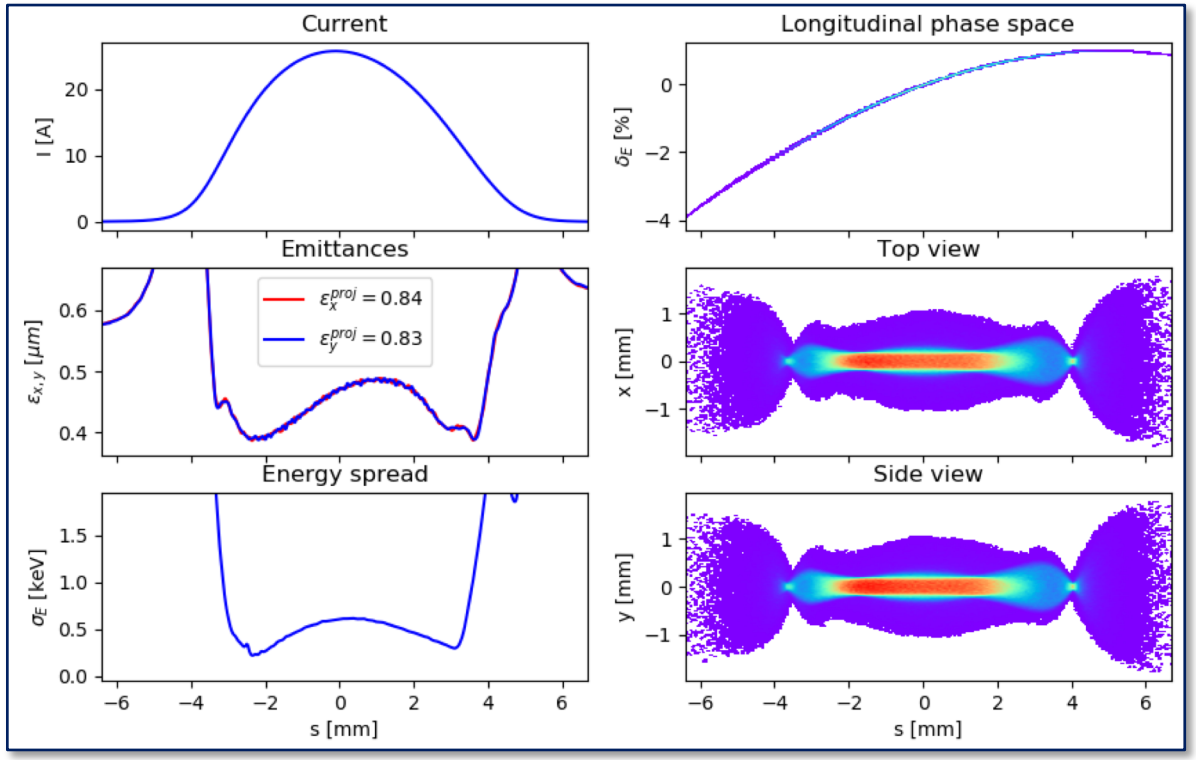
- 3D Space Charge
- 3D wakefields
- 1D CSR
- particle motion with transport matrices of second order
- RF cavity with Rosenzweig-Serafini model
- Chamber with wakes
- ISR and quantum fluctuations

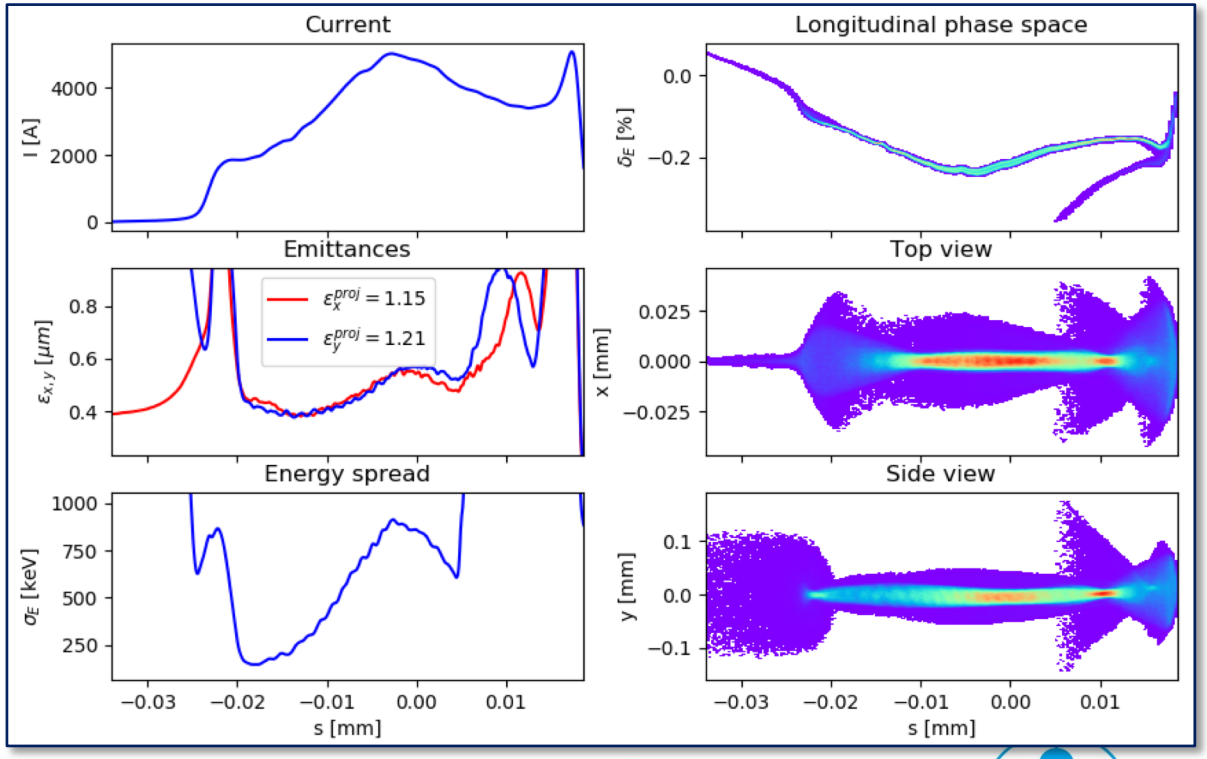
S.Tomin, I.Agapov, M.Dohlus, I.Zagorodnov, *Ocelot as framework for beam dynamics simulations of x-ray sources, in Proceedings of IPAC 2017, WEPAB031*

Modeling #1. Beam dynamics for 500 pC, 5 kA



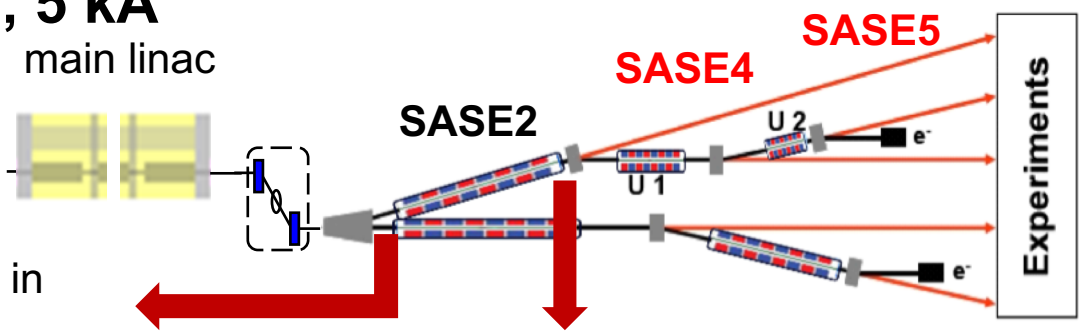
- projected x-emittance growth by 37%
- projected y-emittance growth by 44%

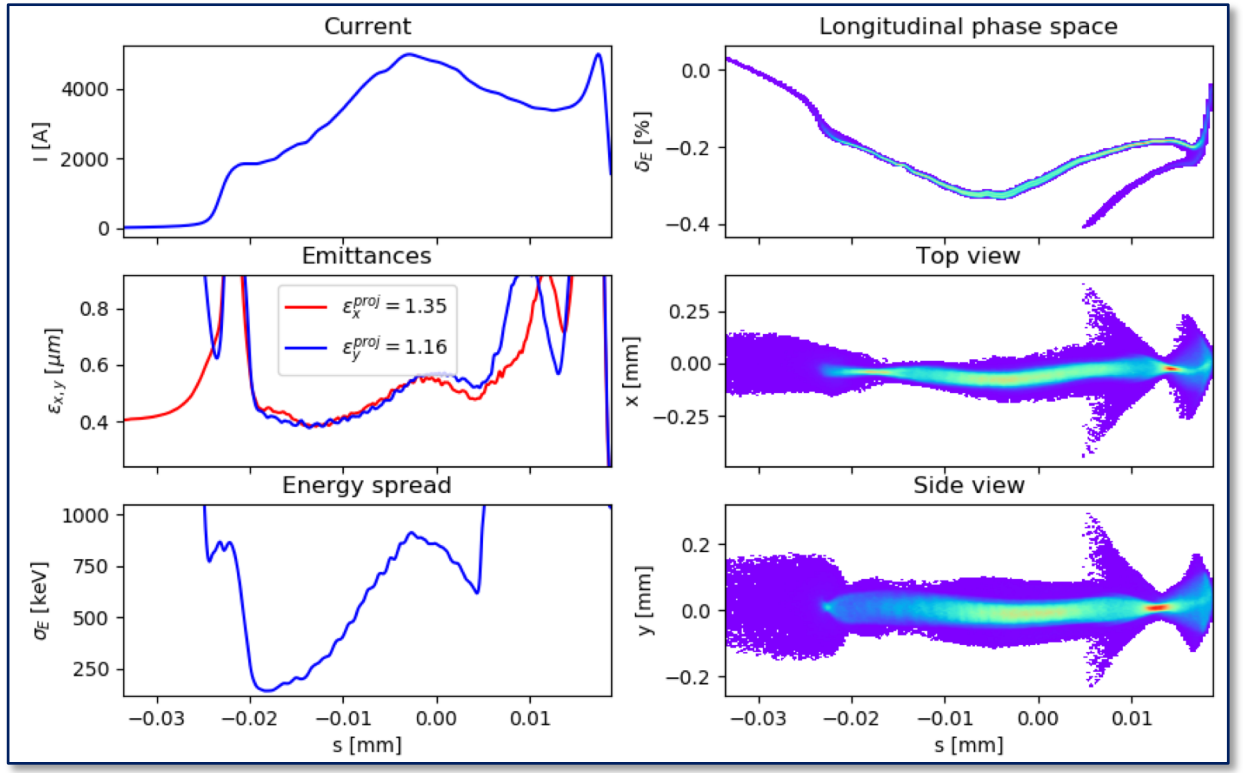


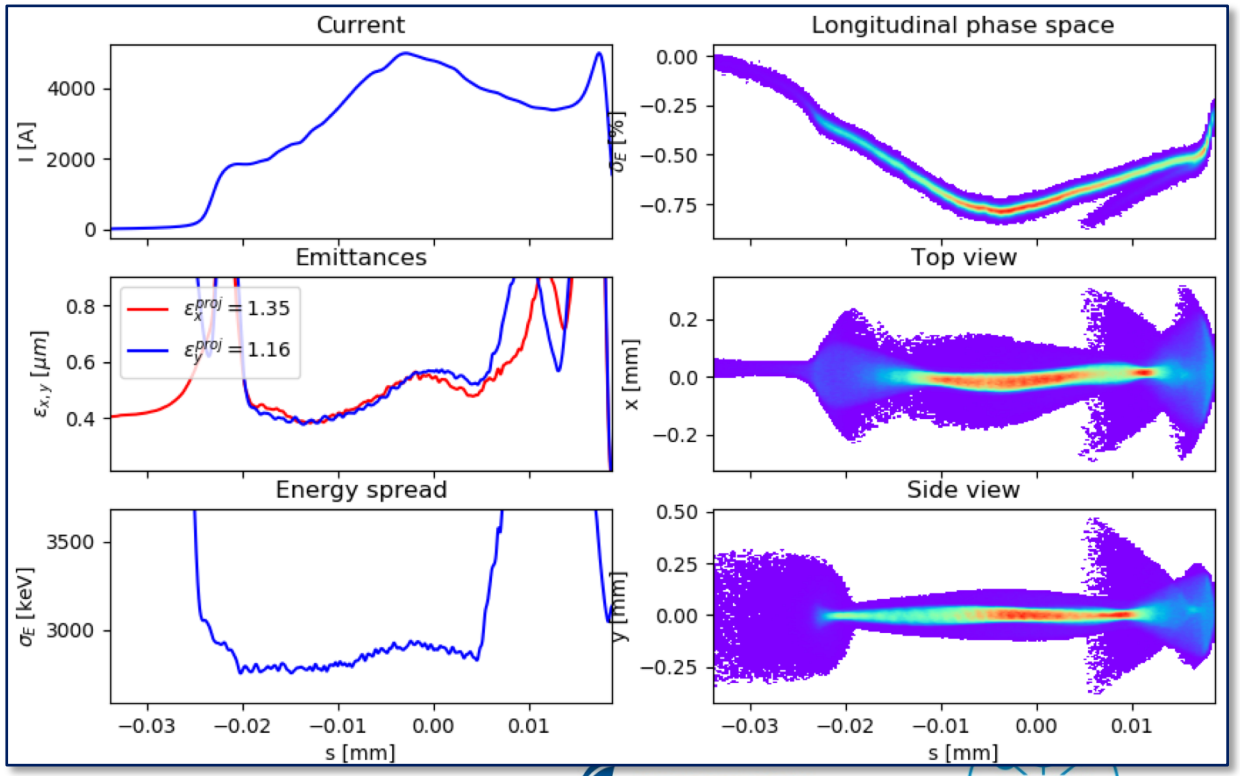


Modeling #1. Beam dynamics for 500 pC, 5 kA

- projected x-emittance growth by 60%
- slice energy spread growth due to quantum fluctuations in SASE2 (K=3.9)
- correlated energy spread growth due to strong wakefields in SASE2



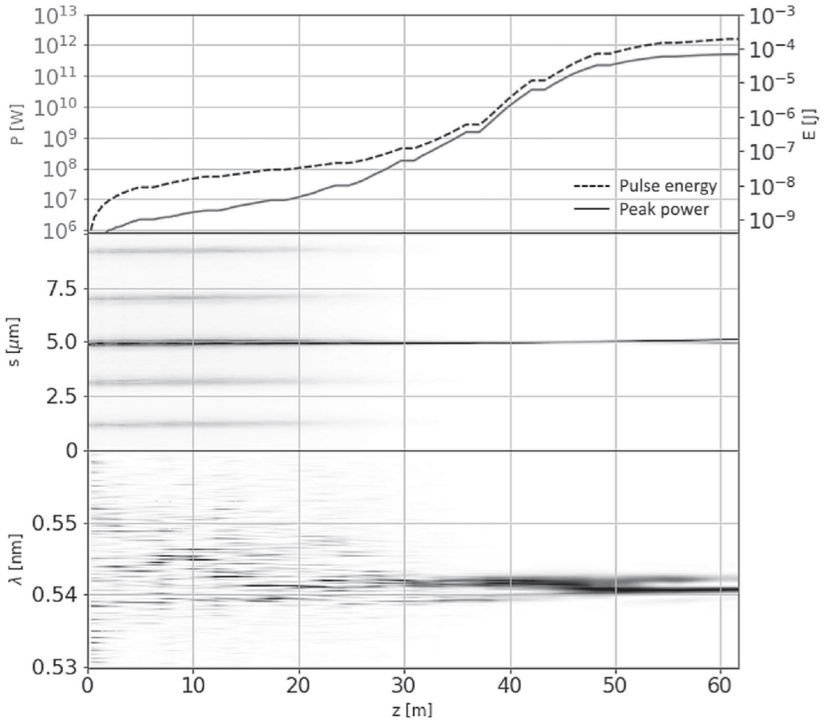


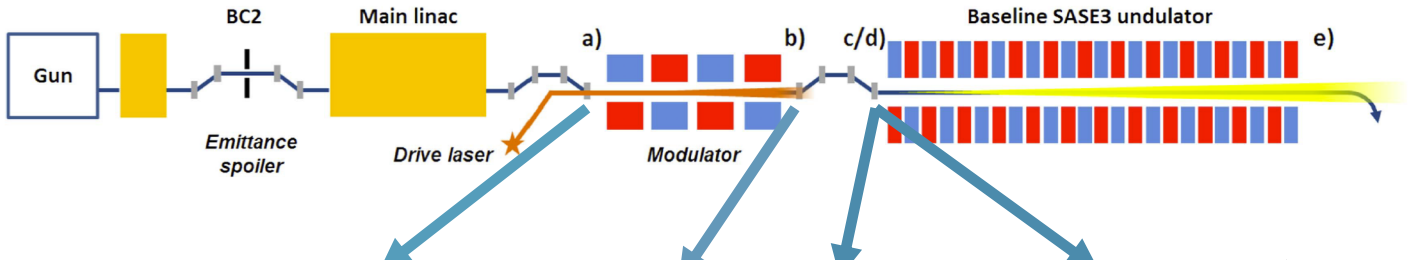


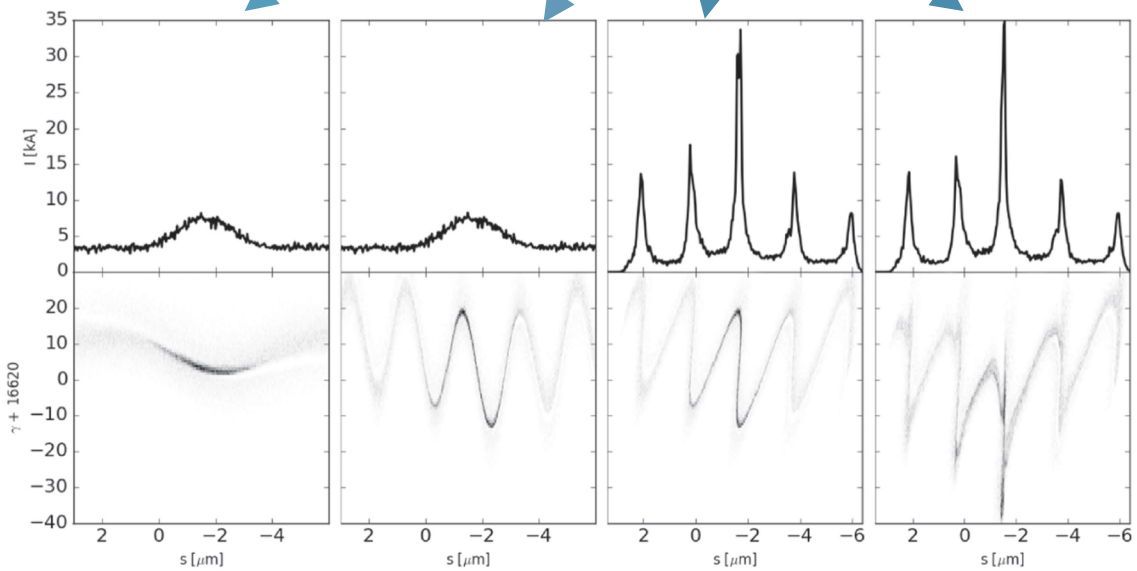
Overview of options for generating high-brightness attosecond x-ray pulses

PFS ⟷ CPBD

Evolution of the 2.3 keV radiation emitted by the modulated electron beam.





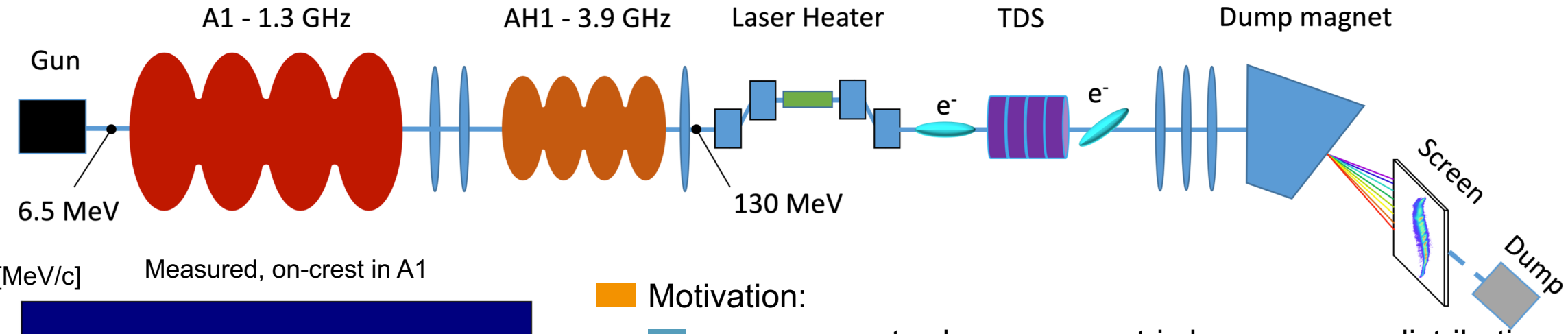


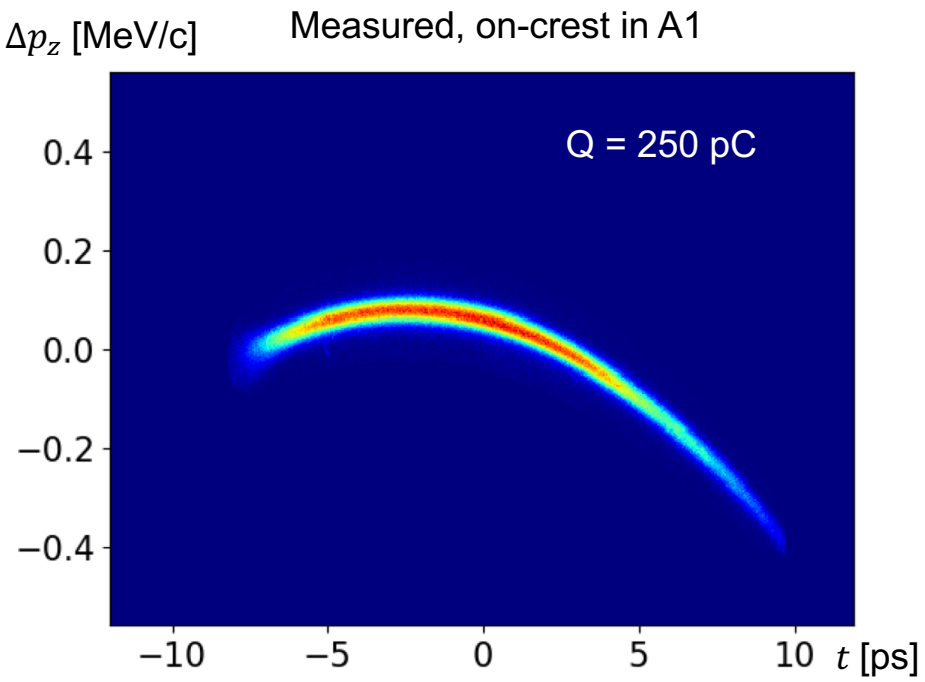
(a) before modulator (b) after modulator (c) after chicane (d) CSR&SC included

Current and longitudinal phase space of the 500 pC electron beam at different stages of the XLEAP scheme

S. Serkez, G Geloni, S Tomin *et al, Journal of Optics, Volume 20, Number 2, 024005, 2018*

Modeling and measurement of collective effects in the EuXFEL Injector.

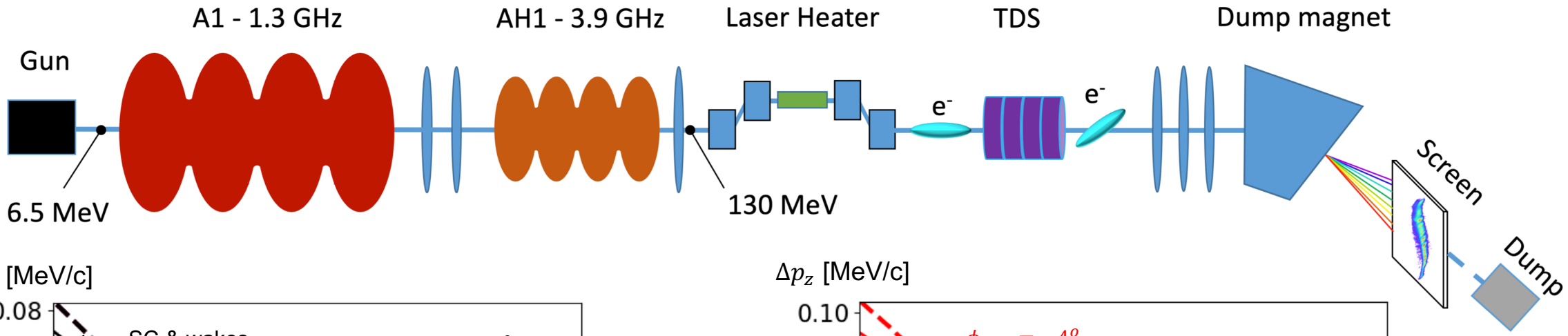


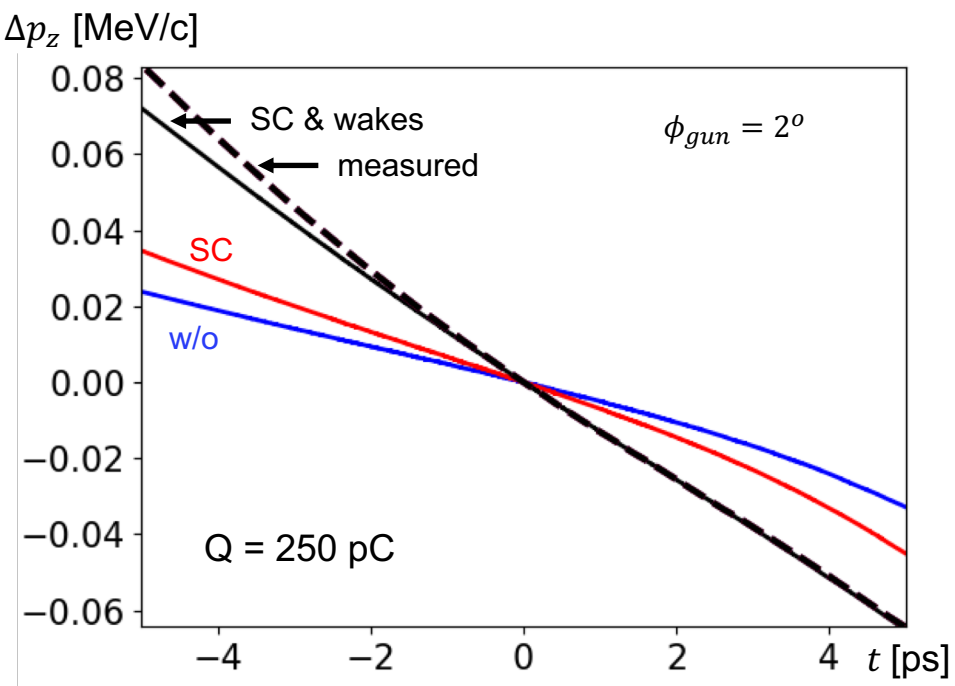


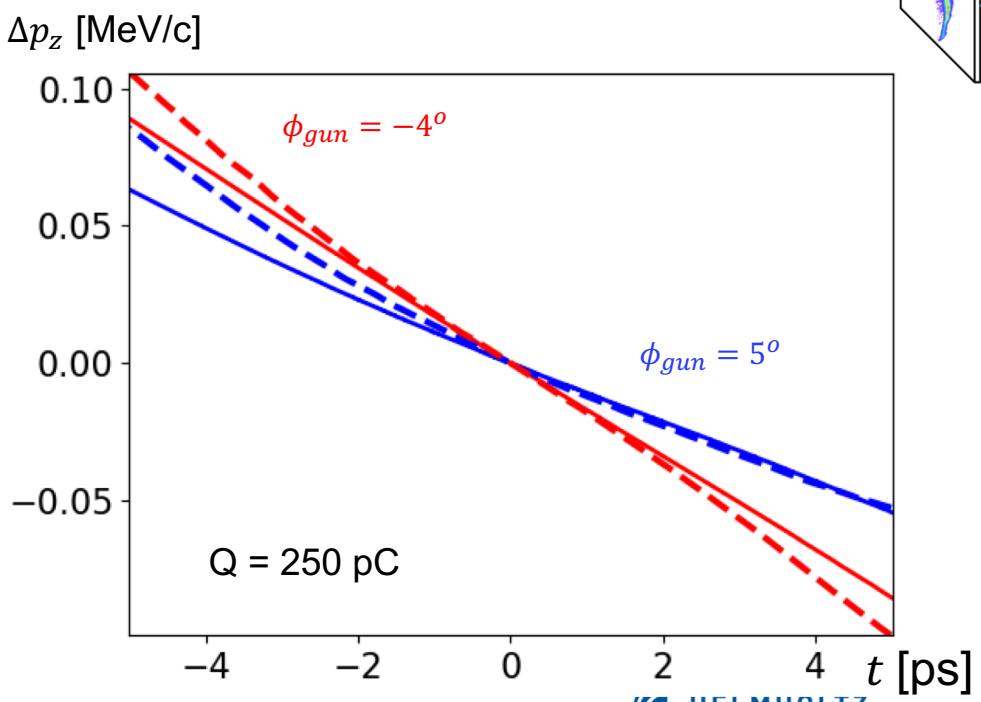
- Motivation:
  - measurements show asymmetric beam energy distribution with $\phi_{A1} = 0^o$
- Simulations:
  - Length of the simulated setup (up to the screen) ~ 40 m
  - 3D wakefields from A1-, AH1-, TDS-cavities.
  - 3D SC along the whole beam line
  - Tracking up to 2nd order
  - Gun simulations with ASTRA (up to 3.2 m from the cathode)

Coauthors: I. Zagorodnov, Y. Chen, S. Tomin

Modeling and measurement of collective effects in the EuXFEL Injector





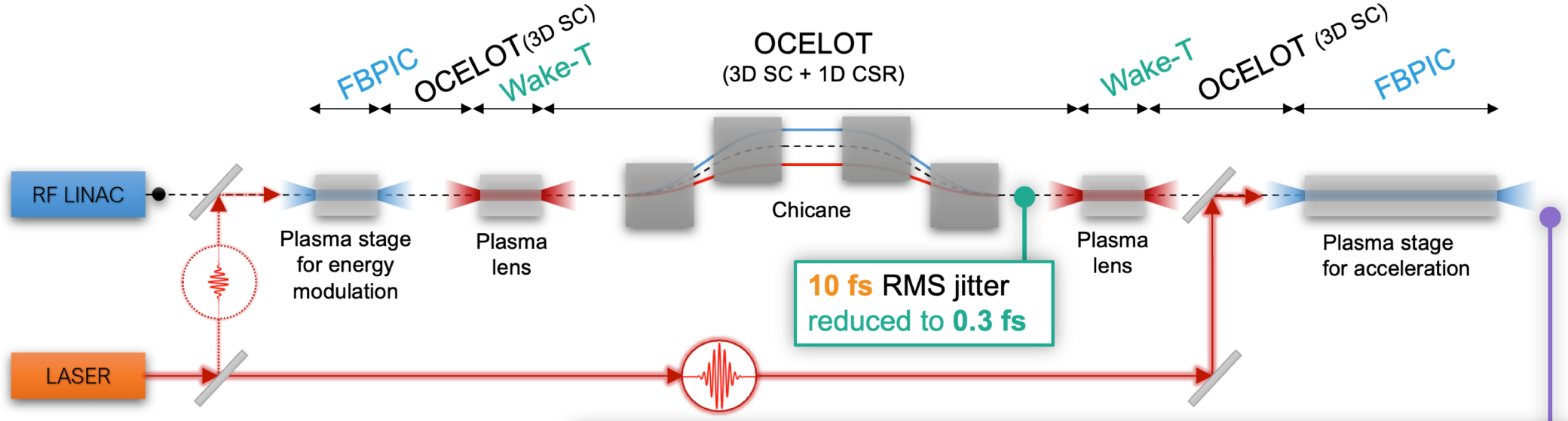


Coauthors: I. Zagorodnov, Y. Chen, S. Tomin

What else can you do with OCELOT?

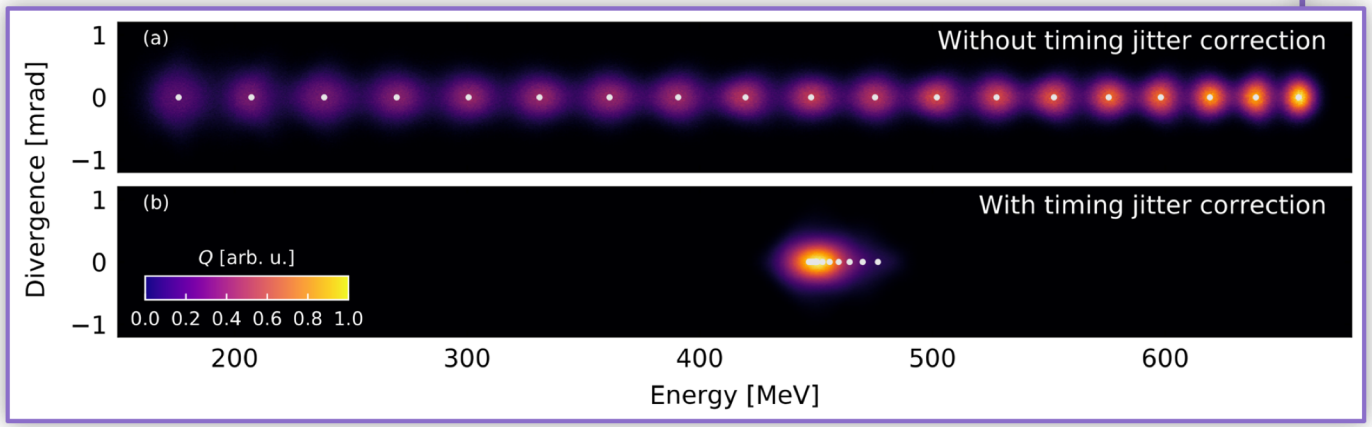
Timing jitter correction for plasma acceleration

Beamline simulations require combination of Ocelot with plasma codes (FBPIC, Wake-T)



Conceptual design for achieving **sub-fs timing jitter** between laser pulse and RF electron beam for injection into plasma accelerator.
[A Ferran Pousa *et al* 2017 *J. Phys.: Conf. Ser.* **874** 012032]

18% RMS energy jitter reduced to 0.7%



Reference:
A. Ferran Pousa PhD thesis (to be published)

Courtesy of Ángel Ferran Pousa (DESY)

Accelerator toy for ML studies

- GitHub: **Python** is most popular language for Machine Learning
- GitHub: python libraries that used in ML are:
  - Numpy
  - Scipy
  - Scikit-learn
  - Theano
  - TensorFlow
  - Keras
  - PyTorch
  - Pandas
  - Matplotlib

I. Agapov, "*Some (possible) ML applications for low-emittance storage rings*". 2nd ICFA Workshop on ML for Particle Accelerators PSI, 27 Feb 2019

Sanity check — FODO

In [2]:
```
# try to learn FODO stability
import os,sys
from ocelot import *
from pylab import *
QF = Quadrupole(l=0.1, k1=0.1)
QD = Quadrupole(l=0.1, k1=-0.14)
D = Drift(l=1.0)
fodo = (QF,D,QD,QD,D,QF)
lat = MagneticLattice(fodo)
```

Script uses OCELOT python module

```
n_train = 10000

x_train = np.zeros([n_train,3])
y_train = np.zeros([n_train,1])


for i in range(n_train):
    QF.k1 = np.random.rand()
    QD.k1 = -np.random.rand()
    D.l = 2.0 * np.random.rand()

    x_train[i,0] = QF.k1
    x_train[i,1] = QD.k1
    x_train[i,2] = D.l
    lat.update_transfer_maps()
    tws = twiss(lat, Twiss())
    if tws is None:
        y_train[i,0] = 0
    else:
        y_train[i,0] = 1
```

```
import keras
from keras.models import Sequential
from keras.layers import Input, Dense, Dropout
from keras.utils import to_categorical
from pylab import *

#fixing a duplicate openmp dylib on mac???
import os
os.environ['KMP_DUPLICATE_LIB_OK']='True'

model = Sequential()
model.add(Dense(64, input_dim=3, activation='relu'))
#model.add(Dropout(0.5))
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='sgd',metrics=['accura
model.fit(x_train,y_train, epochs=30, batch_size=16)
```

```
Using TensorFlow backend.

Epoch 1/30
10000/10000 [==============================] - 2s 211us/step - loss: 0.325
Epoch 2/30
10000/10000 [==============================] - 2s 153us/step - loss: 0.2748 - acc: 0.9313
Epoch 3/30
10000/10000 [==============================] - 2s 152us/step - loss: 0.2614 - acc: 0.9313
```

```
n_test = 5000

x_test = np.zeros([n_test,2])
y_test = np.zeros([n_test,1])

D.l = 3.9

for i in range(n_test):
    QF.k1 = 5*np.random.rand()
    QD.k1 = -5*np.random.rand()

    x_test[i,0] = QF.k1
    x_test[i,1] = QD.k1
    lat.update_transfer_maps()
    tws = twiss(lat, Twiss())
    if tws is None:
        y_test[i,0] = 0
    else:
        y_test[i,0] = 1
```

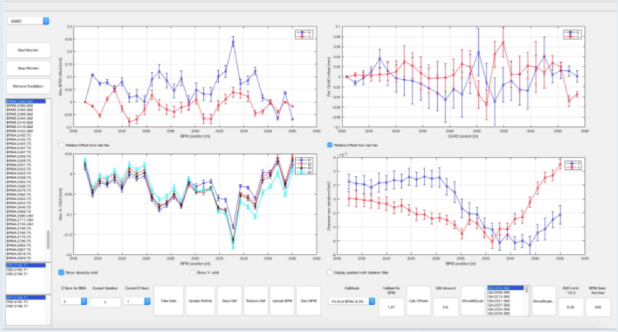
Optimization of the European XFEL with OCELOT

Motivation

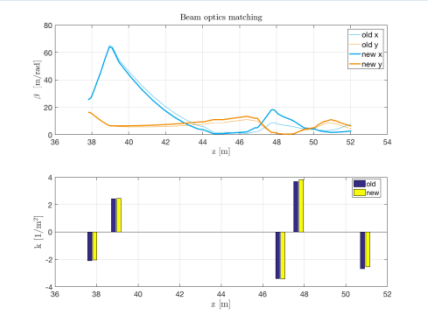
- Modern Free Electron Lasers are complex facilities with hundreds of free tuning parameters
  - Bunch compression, orbit, beam optics, gun optimization, undulator gaps, phase-shifters, etc
- Even when the main accelerator systems work well, manual fine-tuning is necessary to get the best performance and this is time expensive

Zoo of High Level Control tools

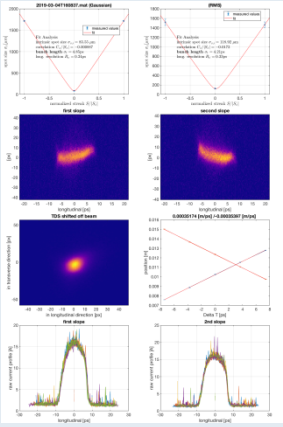
Beam Based Alignment



Beam matching

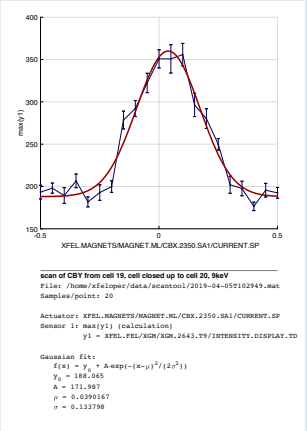


Longitudinal profile measurement



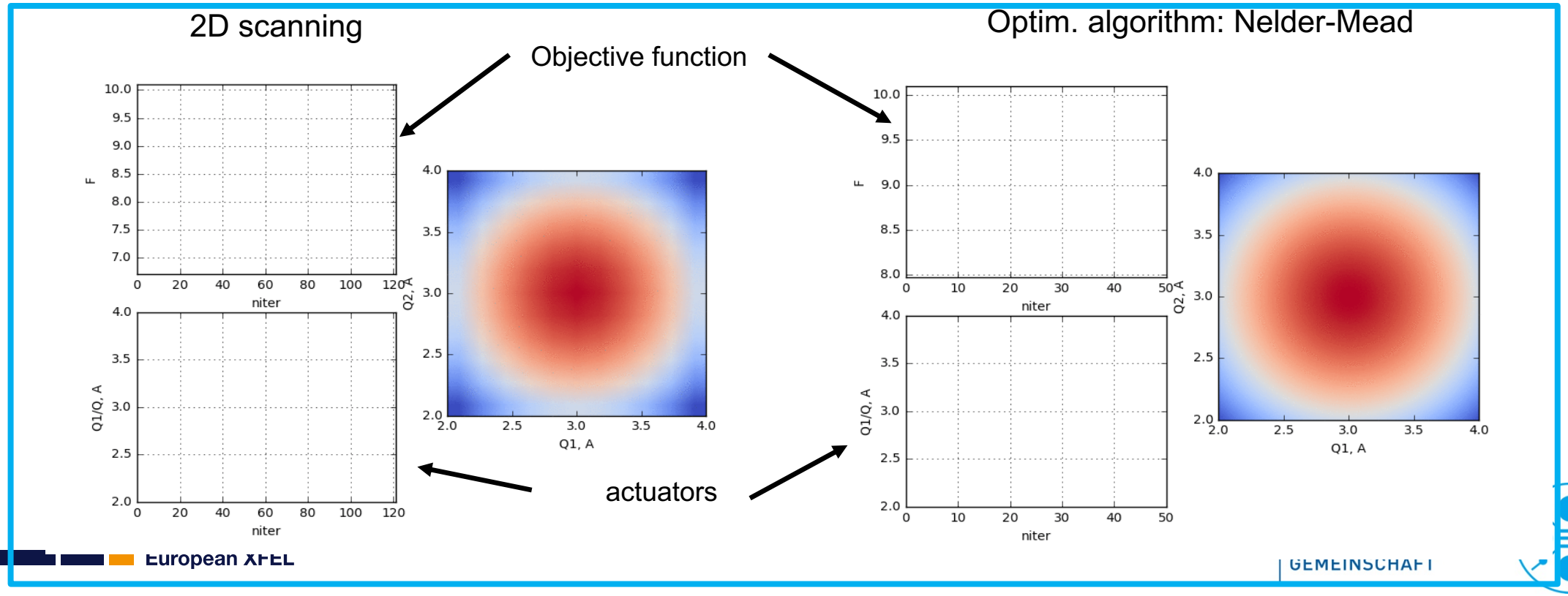
Orbit correction

Scan tool



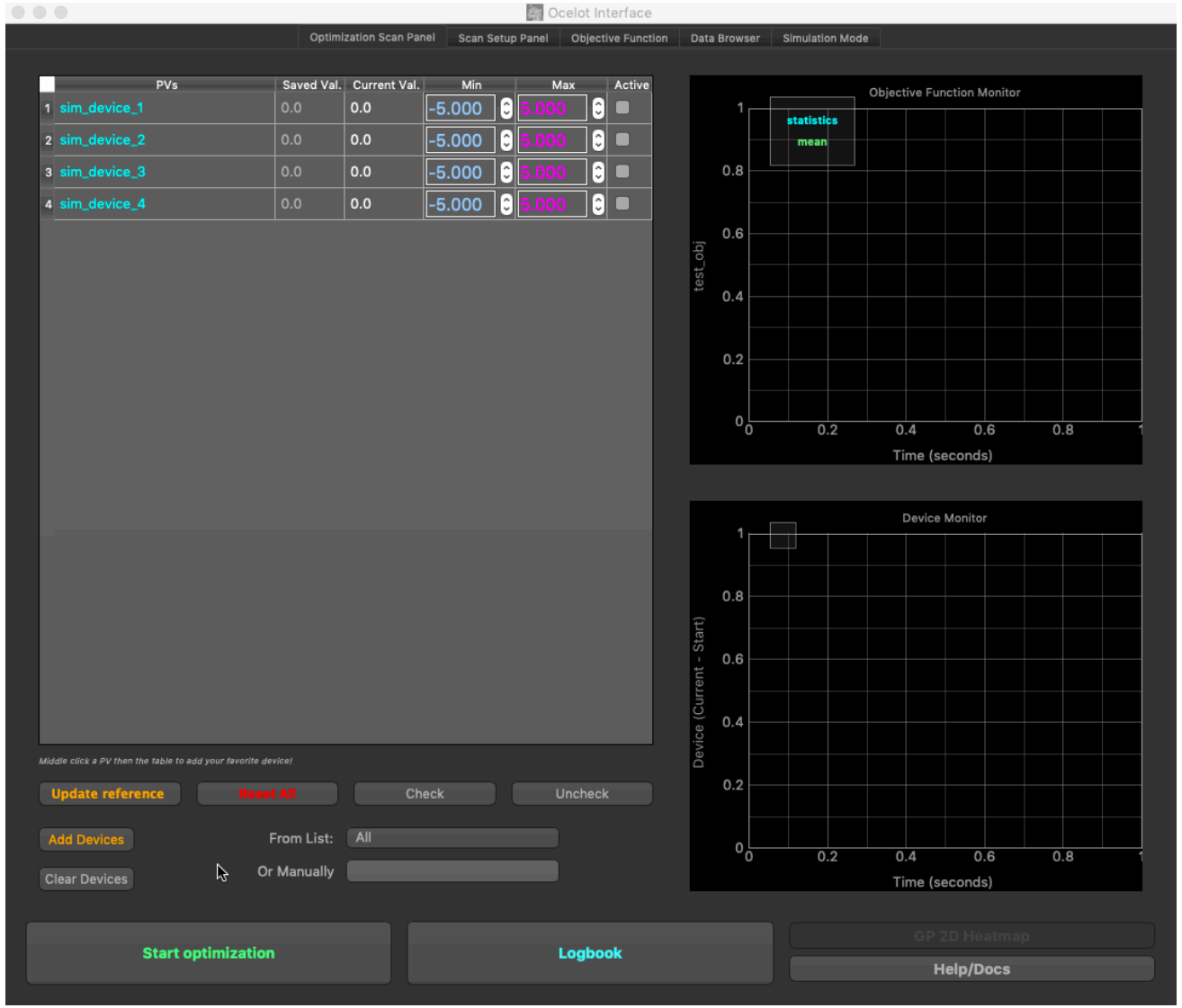
Optimization algorithm

- Optimization algorithms are faster than scanning
- Optimization methods can be model-independent or model-dependent



OCELOT Optimizer

- OCELOT optimizer is a flexible platform for optimization:
  - Interchangeable optimization methods
  - GUI
    - ► Add/select device or group of devices
    - ► Craft/modify target function
  - Infrastructure for testing new methods
  - Save/load configs
  - Logging
- Collaboration

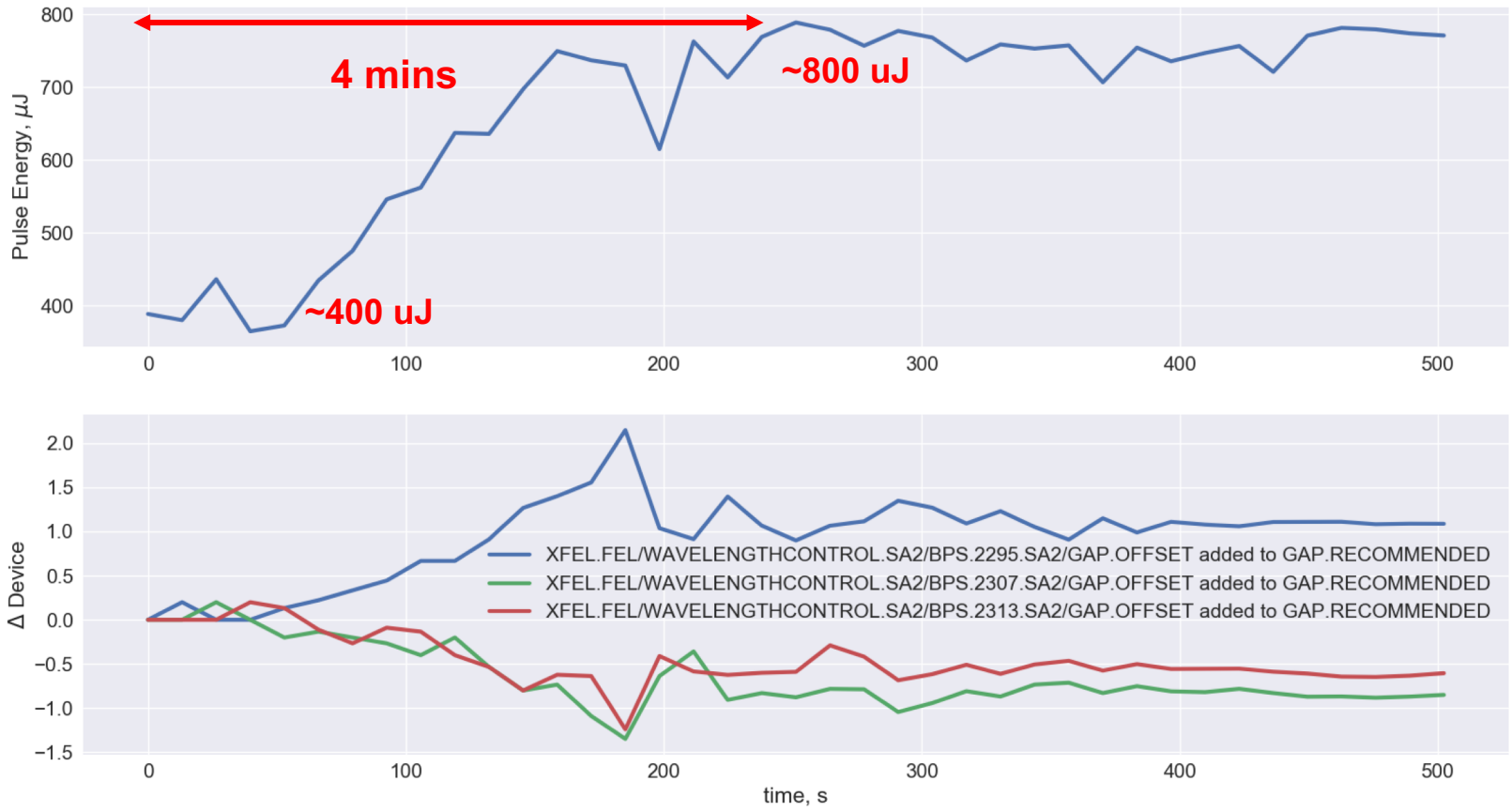


- *I. Agapov et al, arXiv:1704.02335*
- *S. Tomin et al, https://doi.org/10.18429/JACoW-IPAC2017-WEPAB031*
- *M.W. McIntire et al, DOI:10.18429/JACoW-IPAC2016-WEPOW055*

OCELOT Optimizer: Use cases

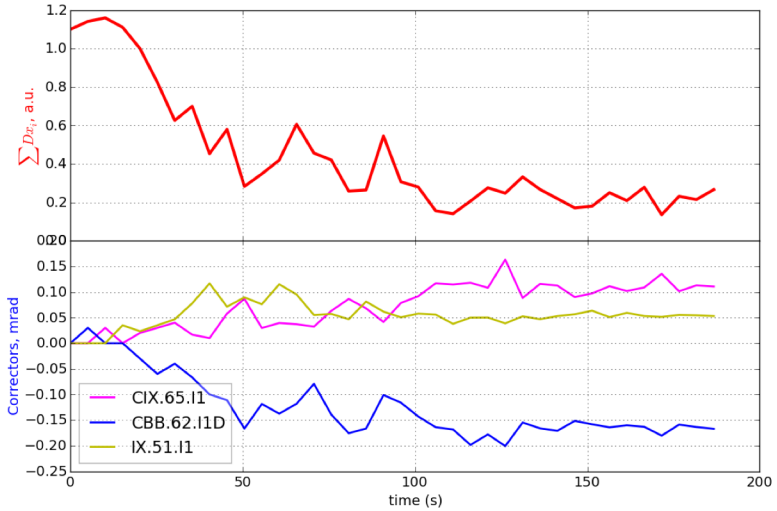
- European XFEL
  - FEL pulse energy maximization:
    - ► Orbit inside an undulator
    - ► Phase-shifters
    - ► Orbit in injector
    - ► Matching quads
    - ► RF settings

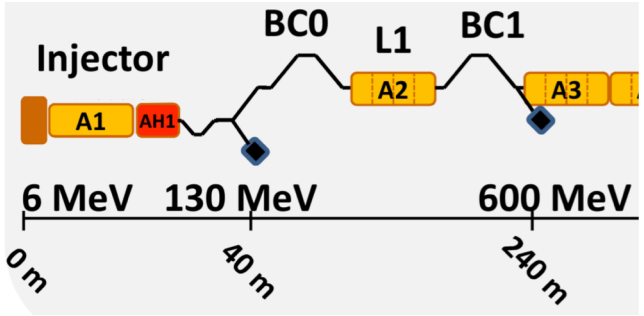
Phase-shifters



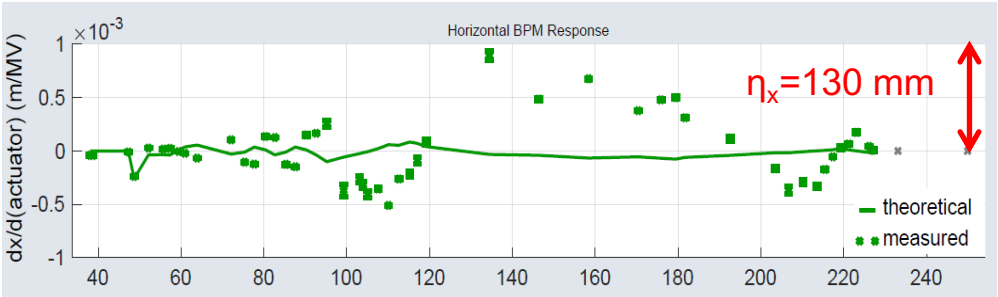
OCELOT Optimizer: Use cases

- European XFEL
  - FEL pulse energy maximization:
    - ▶ Orbit inside an undulator
    - ▶ Phase-shifters
    - ▶ Orbit in injector
    - ▶ Matching quads
    - ▶ RF settings
  - Local dispersion correction in injector

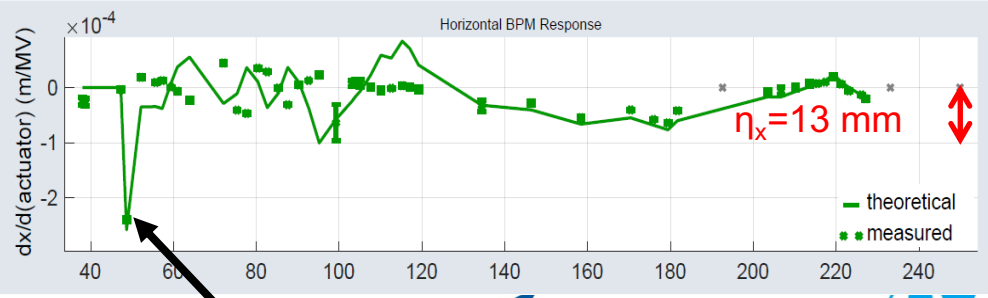




Before correction

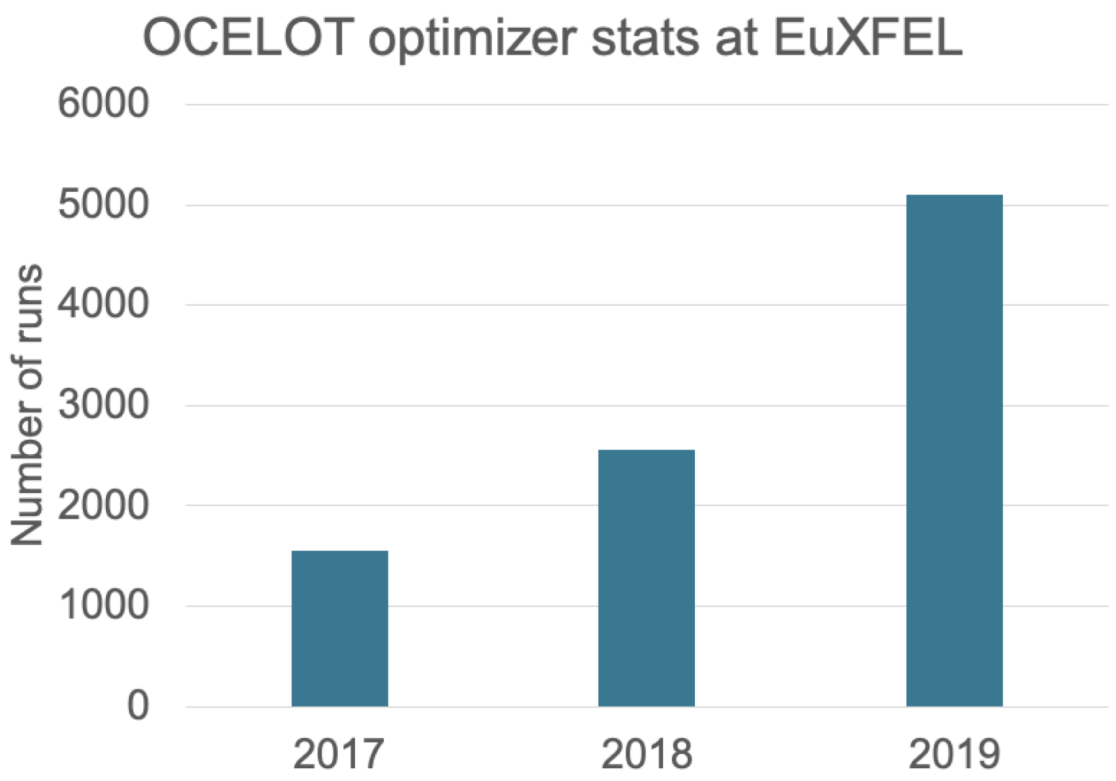


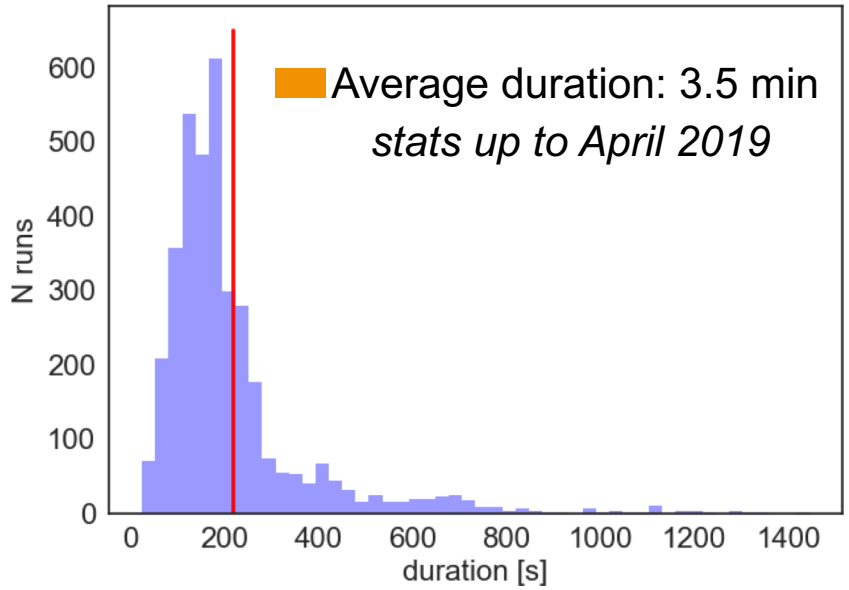
After correction

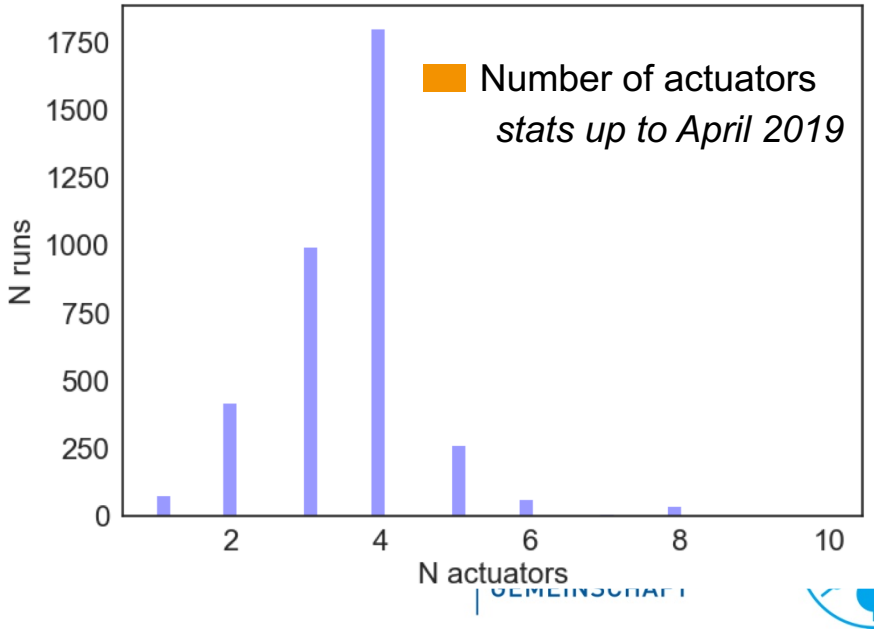


Laser Heater chicane

OCELOT Optimizer: Statistics



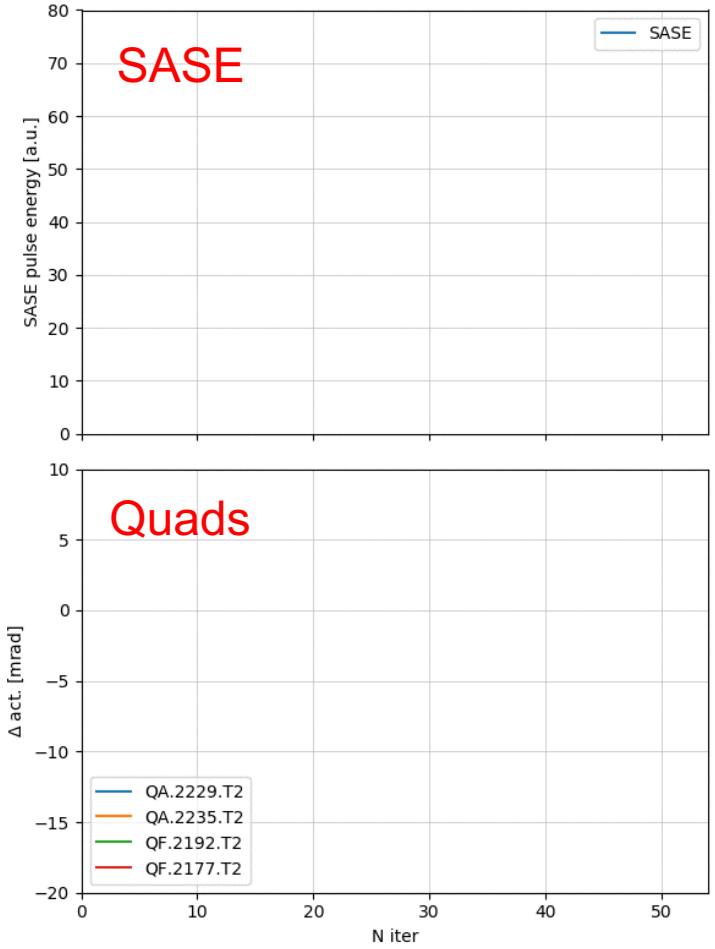




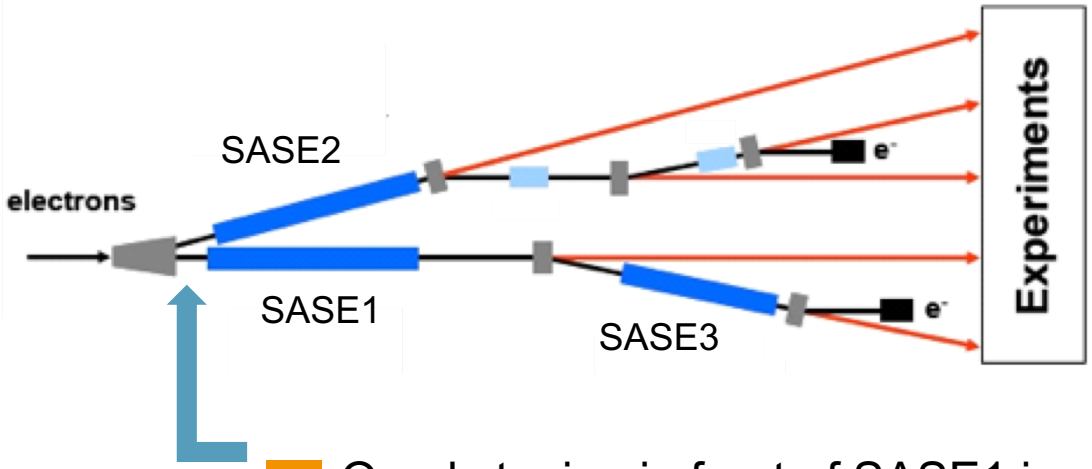
In most optimizations 4 devices are used and the average time duration of a single optimization is 3.5 minutes

Hyperparameters. Beam matching

SASE optimization with 4 quads



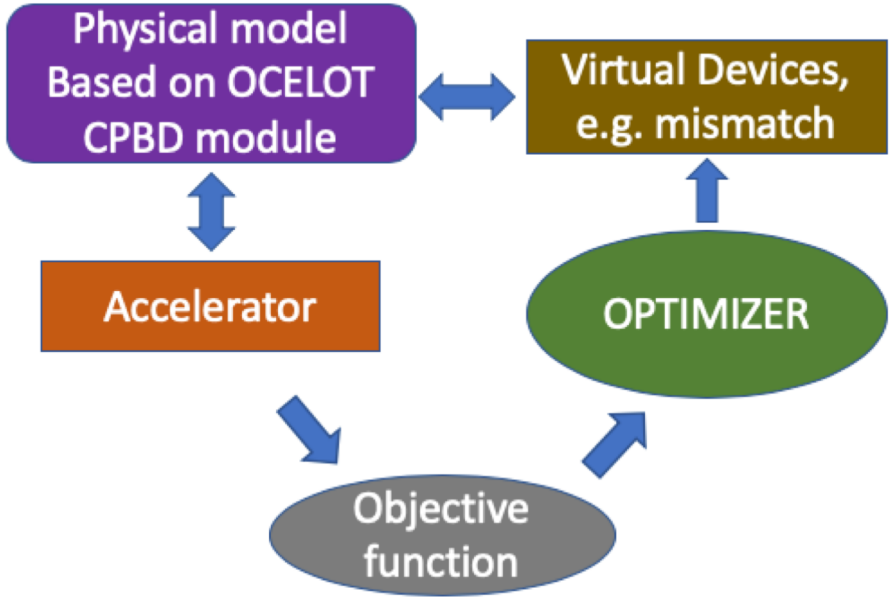
- Initial quad step
- Group of quads



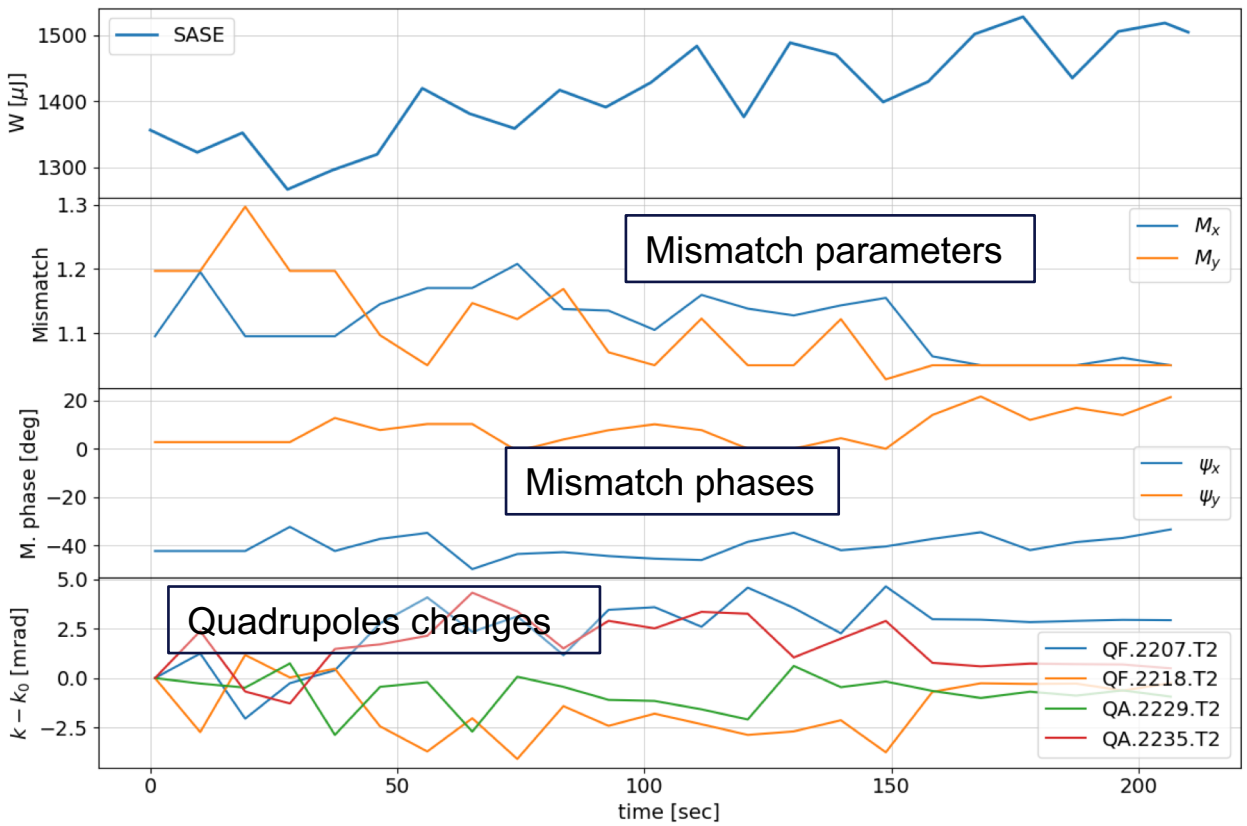
- Quads tuning in front of SASE1 is one of the steps of standard optimization procedure with Optimizer.
- Hyperparameters, such as initial steps, number of iterations, were corrected as experience increased.

CPBD ⟷ OBC

Beta mismatch parameter* optimization



SASE pulse energy opt. 25 iterations



- OCELOT beam dynamics module is already used for orbit correction
- Bounds and hyperparameters are defined for all matching sections
- Only 4 actuators are used while number of quads can be more - reduction of dimensionality in some cases

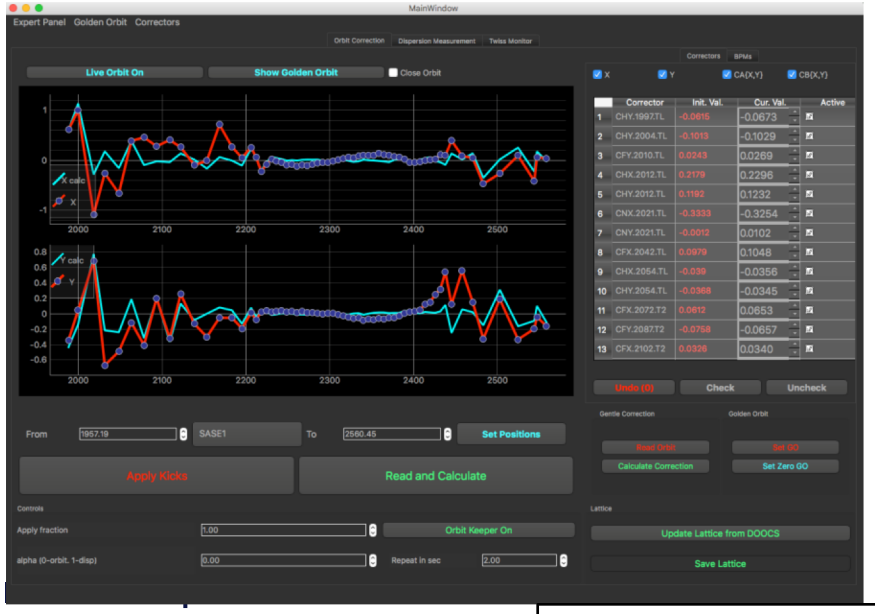
* M. Sands, SLAC-AP-85 1991

CPBD ⟷ OBC

Orbit correction tool with adaptive feedback.

- OCELOT orbit correction is the standard tool for orbit correction (using SVD/MICADO algorithms).
- The Adaptive Feedback is a statistical optimizer[1] exploiting the orbit jitter and its correlation with a fast FEL intensity signal (shot-to-shot resolution) to optimize the undulator launch orbit

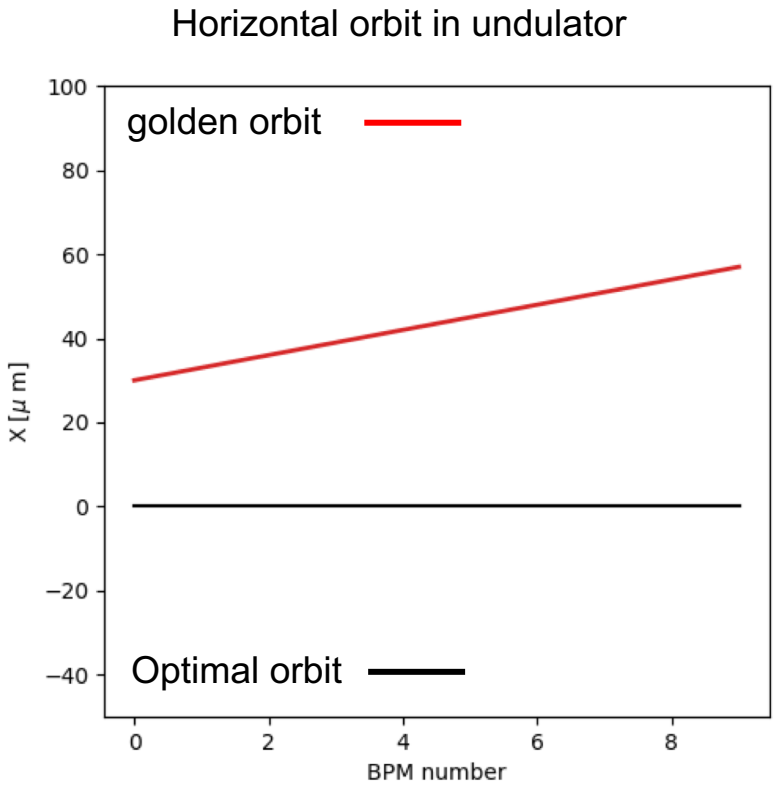
Orbit correction tool GUI

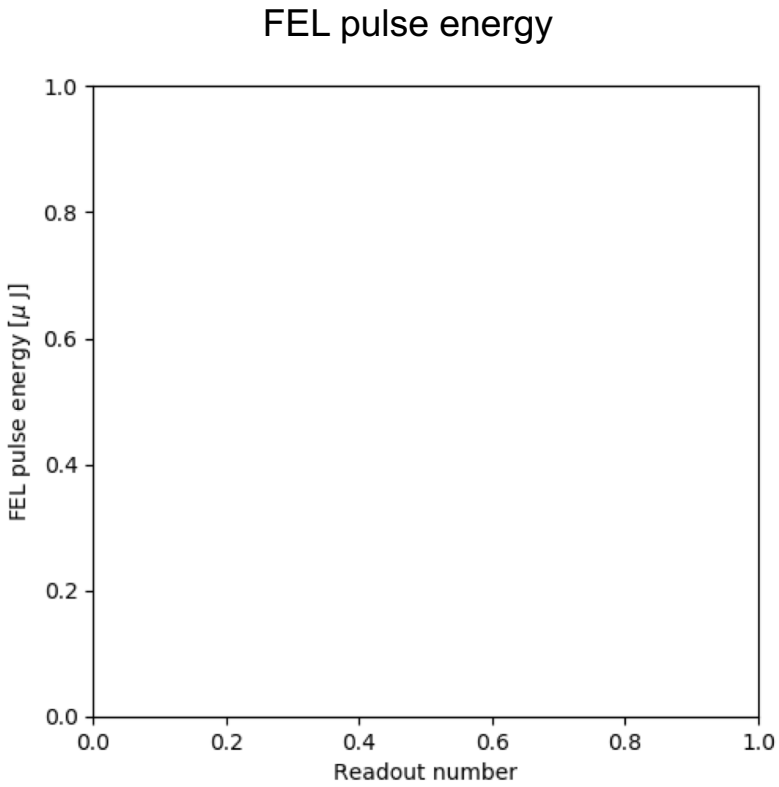


Adaptive Feedback GUI

[1] G. Gaio, M. Lonza, Automatic FEL Optimization at FERMI, Proc. of ICALEPCS2015

Adaptive Feedback: how it works

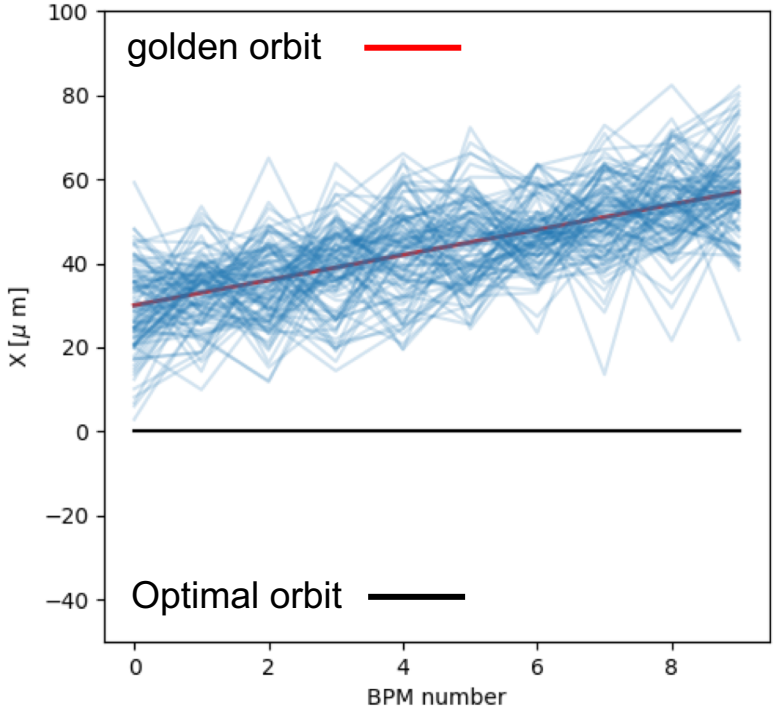




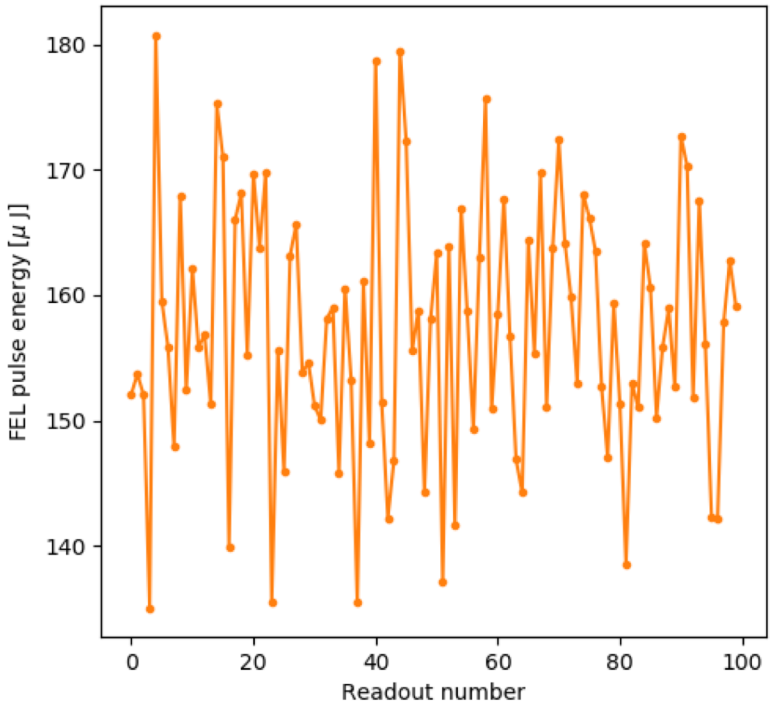
Iteration #1

Adaptive Feedback: how it works

Horizontal orbit in undulator



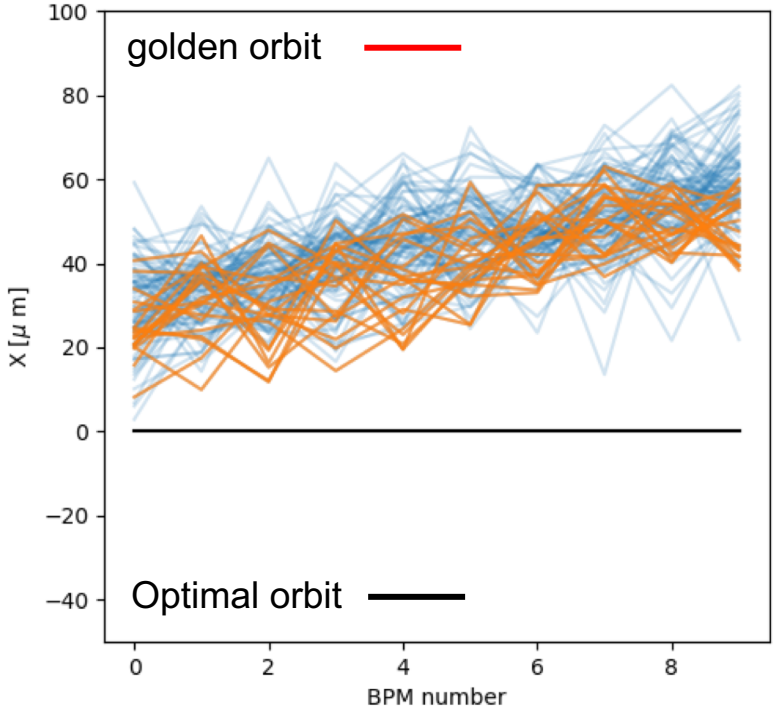
FEL pulse energy



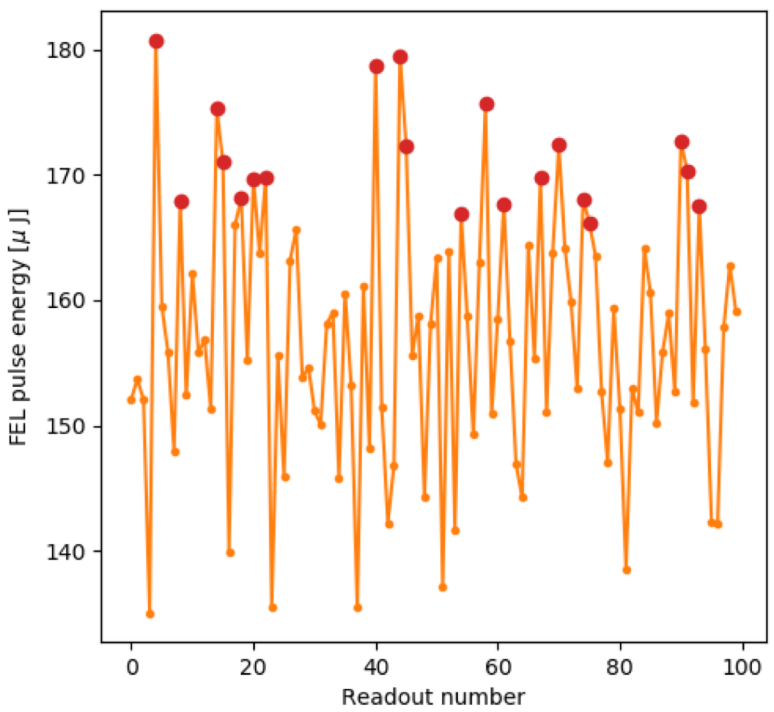
Iteration #1

Adaptive Feedback: how it works

Horizontal orbit in undulator



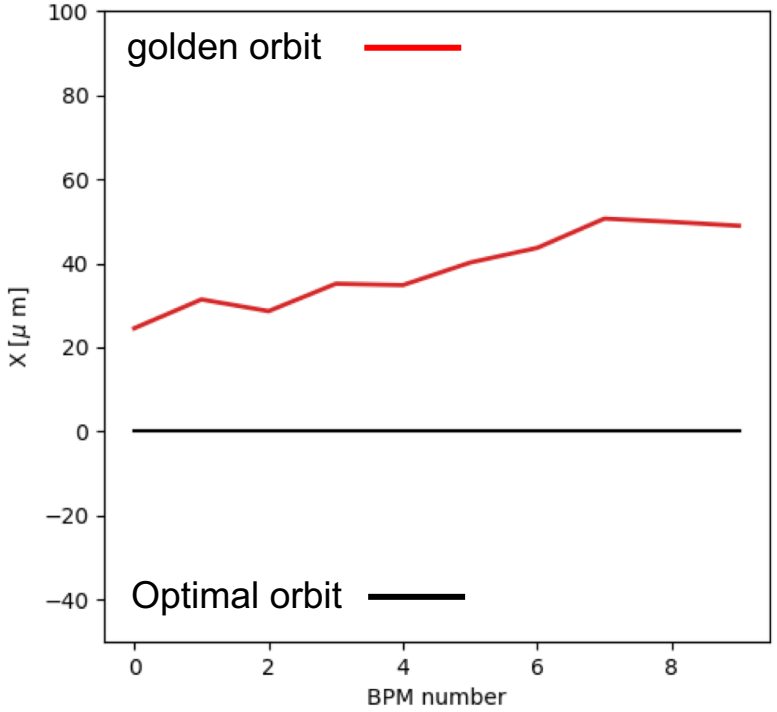
FEL pulse energy



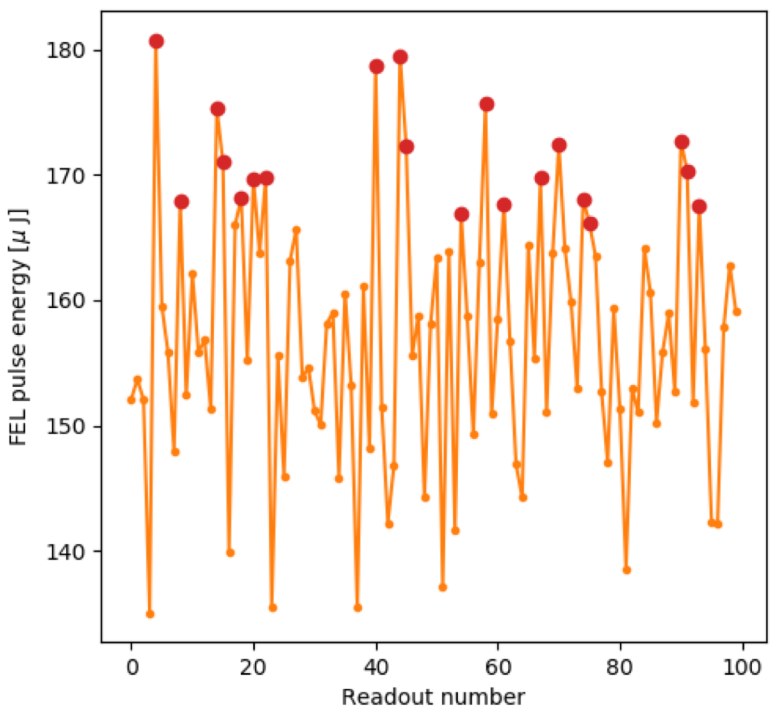
Iteration #1

Adaptive Feedback: how it works

Horizontal orbit in undulator



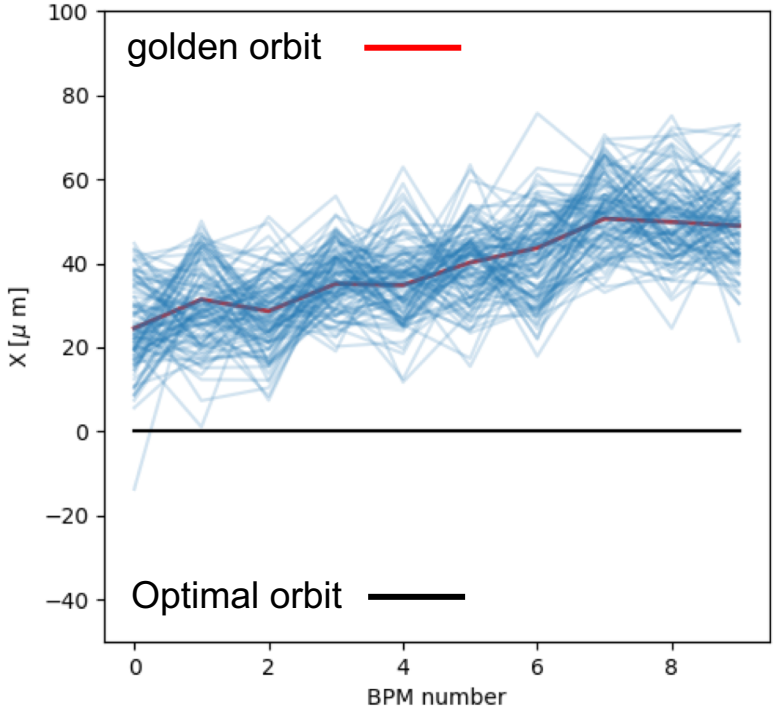
FEL pulse energy



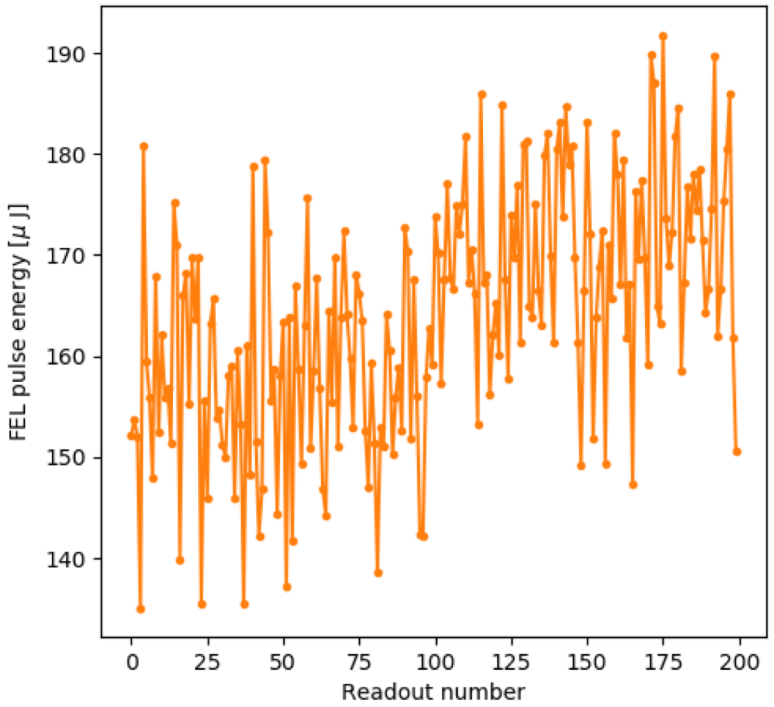
Iteration #1

Adaptive Feedback: how it works

Horizontal orbit in undulator



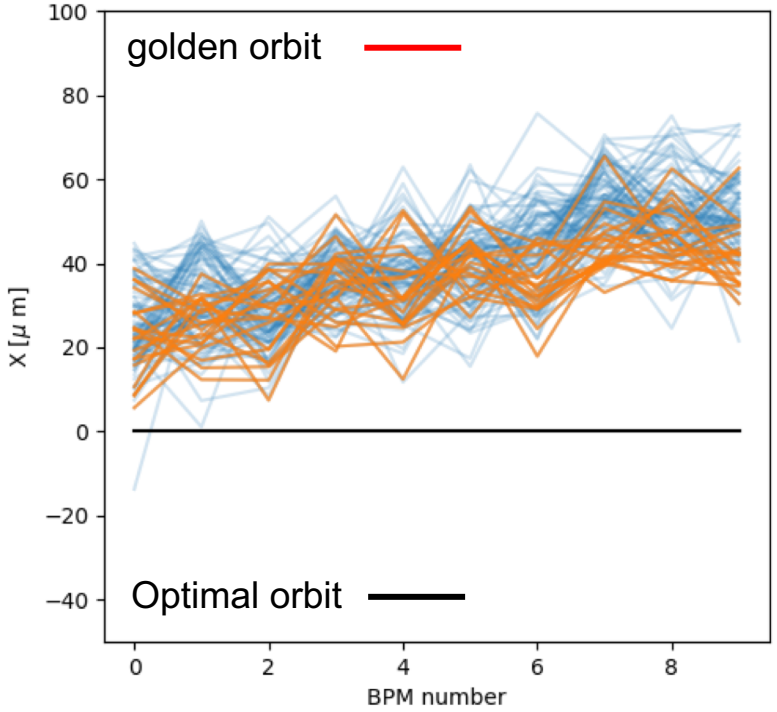
FEL pulse energy



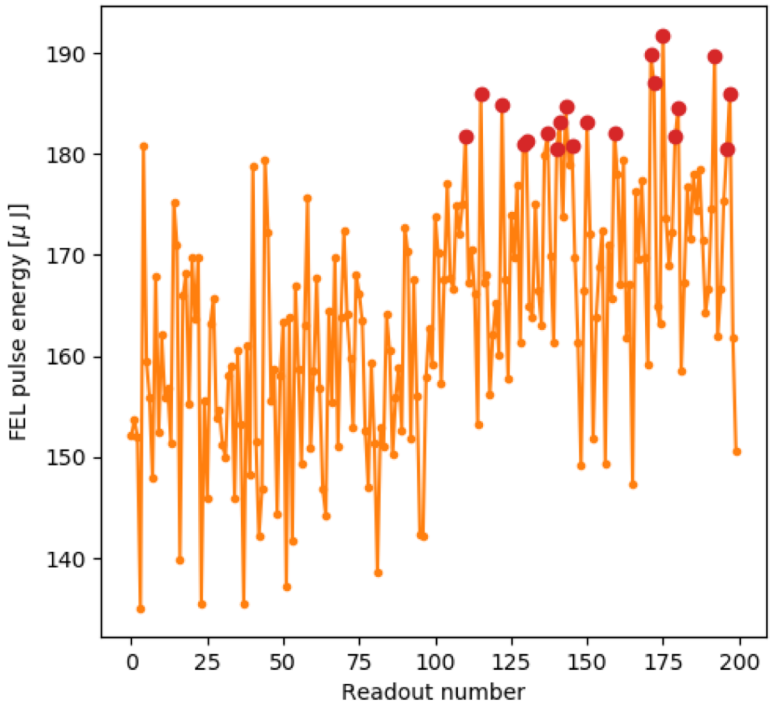
Iteration #2

Adaptive Feedback: how it works

Horizontal orbit in undulator



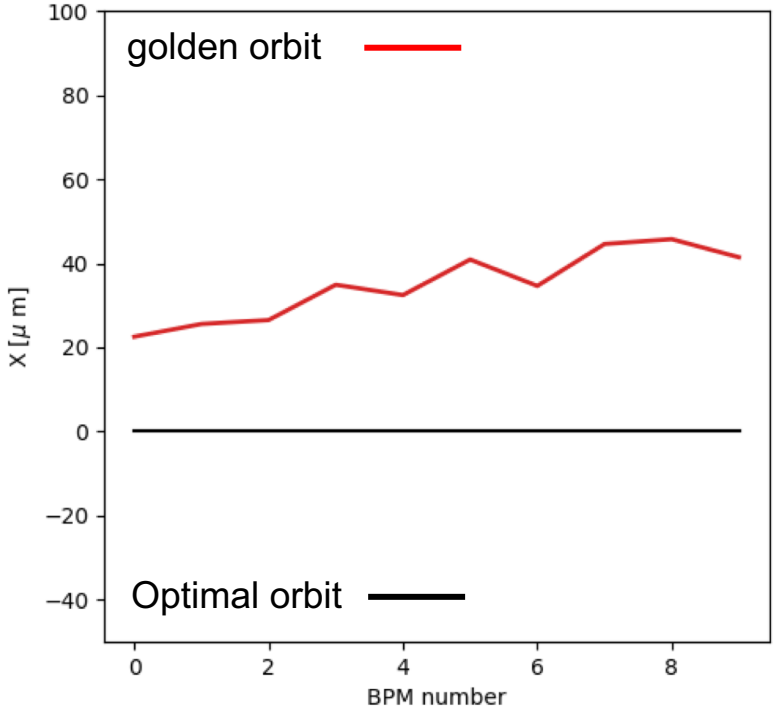
FEL pulse energy



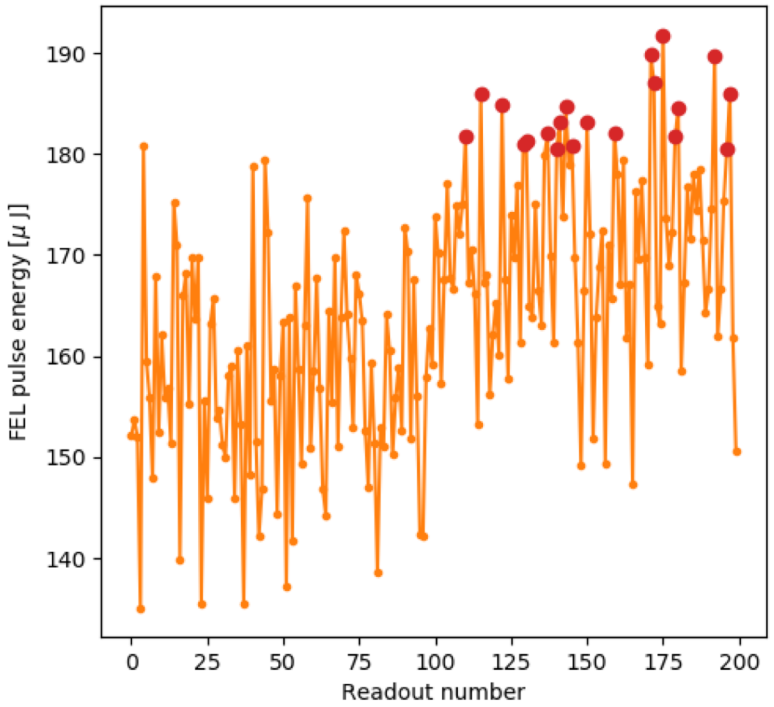
Iteration #2

Adaptive Feedback: how it works

Horizontal orbit in undulator



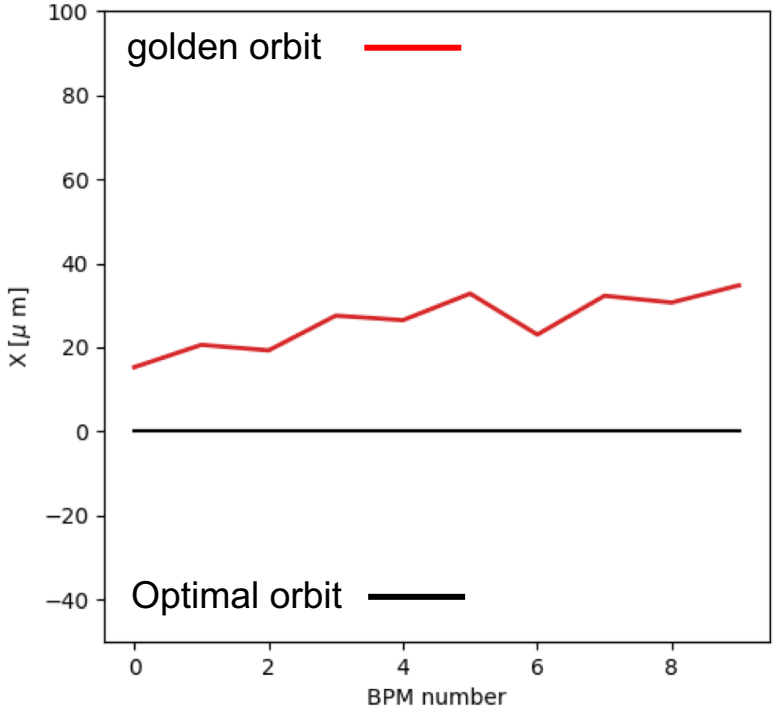
FEL pulse energy



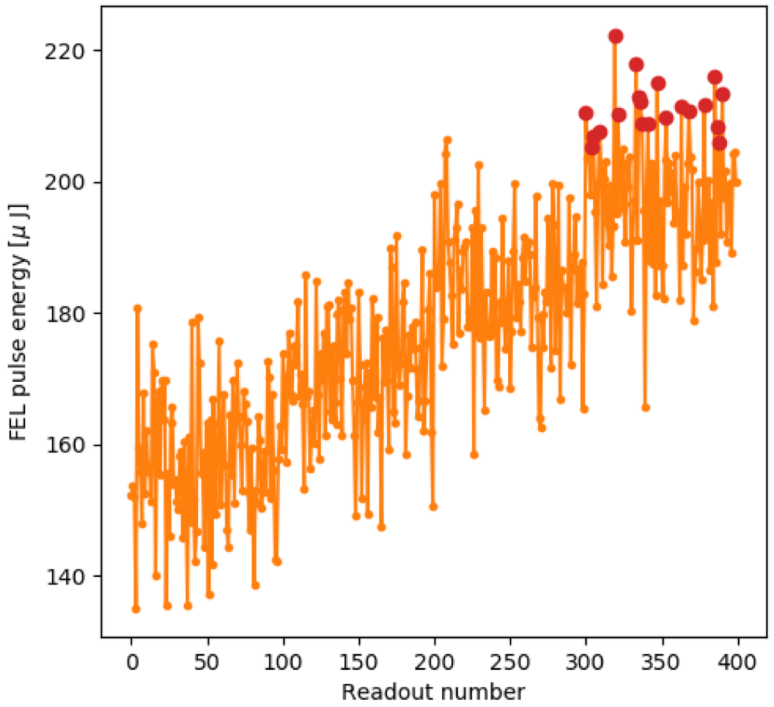
Iteration #2

Adaptive Feedback: how it works

Horizontal orbit in undulator



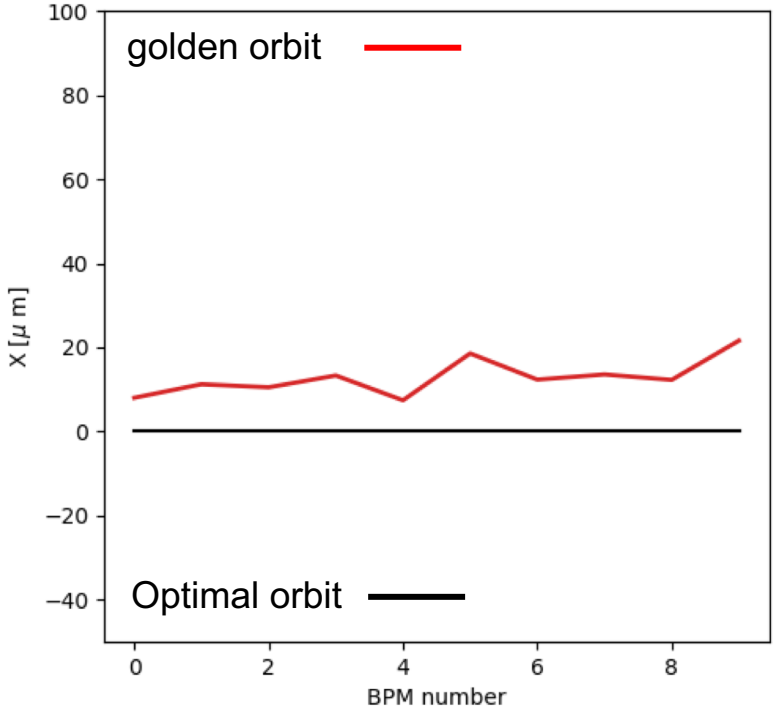
FEL pulse energy



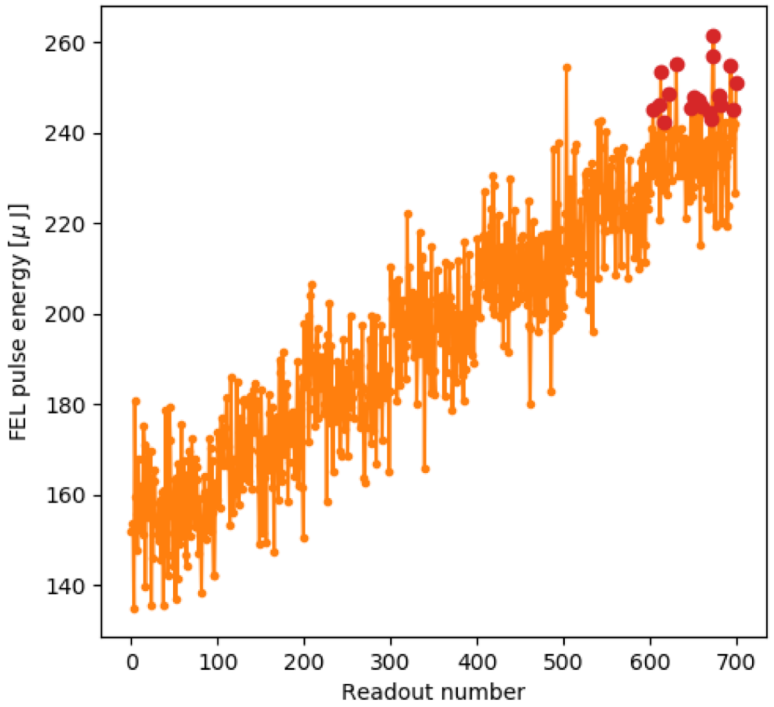
Iteration #4

Adaptive Feedback: how it works

Horizontal orbit in undulator



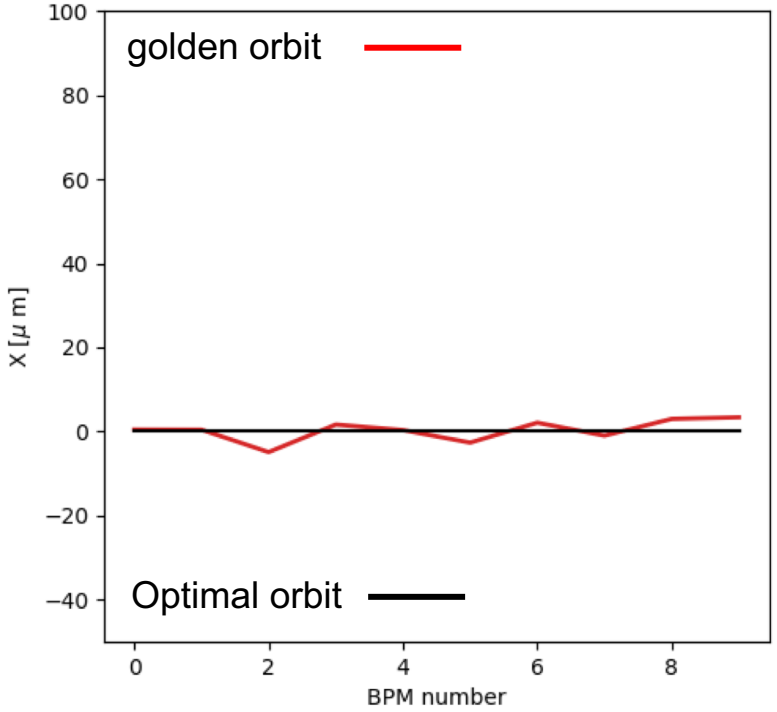
FEL pulse energy



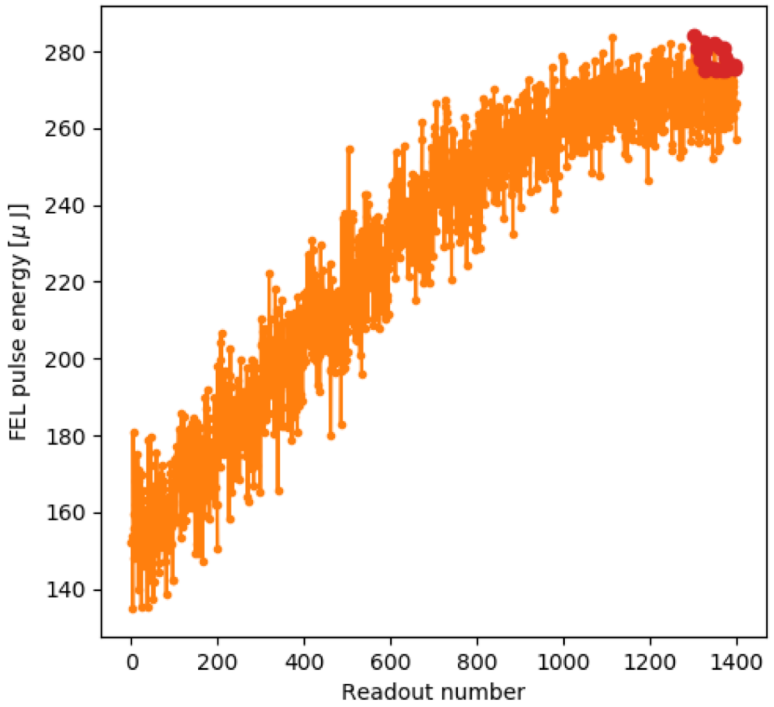
Iteration #7

Adaptive Feedback: how it works

Horizontal orbit in undulator



FEL pulse energy



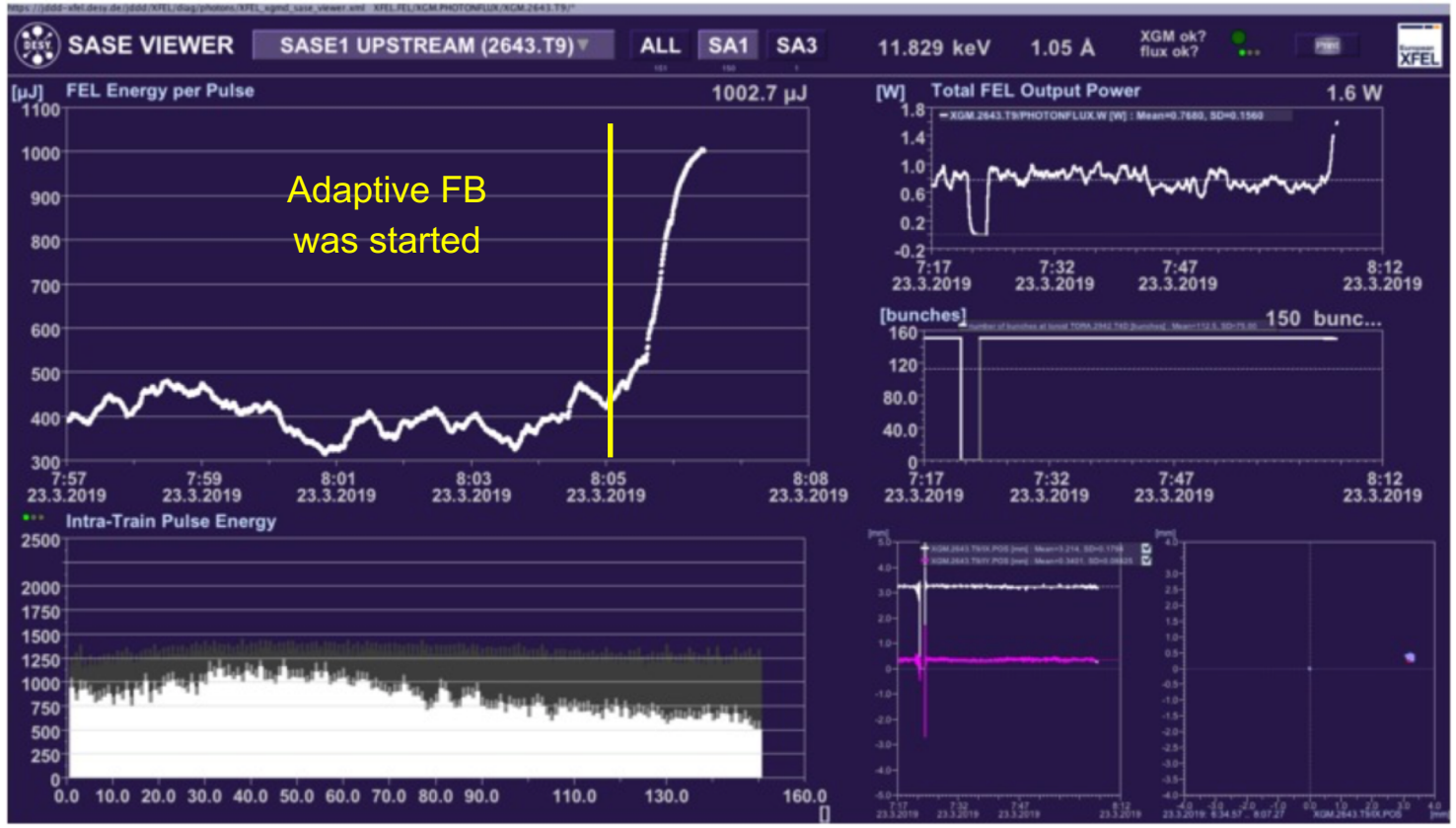
Iteration #14

Adaptive Feedback

Logbook entry: /XFELelog/data/2019/12/23.03_M

23.03.2019 08:06 **XFEL** mscholz, TIS, Artem **XFEL_xgmd_sase_viewer.xml**

SASE1 signal could be doubles with only starting the **adaptive** FB



Conclusion

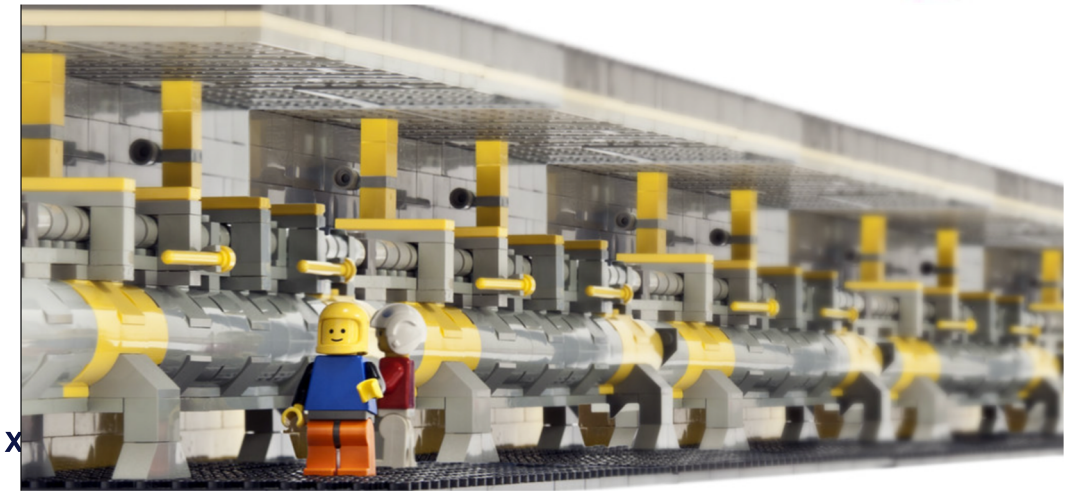
OCELOT modules and functions
for accelerator modeling and optimizing

Conclusion

OCELOT modules and functions
for accelerator modeling and optimizing

+

Available Python modules:
Math., data analysis, ML, plotting,
optimization algorithms etc



Documentations

- Tutorials. https://github.com/ocelot-collab/ocelot
- Description of functions and classes in source files **OR** use *help()* in python/ipython/Jupyter
- Read code

```
In [6]: help(Quadrupole)

Help on class Quadrupole in module ocelot.cpbd.elements:

class Quadrupole(Element)
 |  quadrupole
 |  l - length of lens in [m],
 |  k1 - strength of quadrupole lens in [1/m^2],
 |  k2 - strength of sextupole lens in [1/m^3],
 |  tilt - tilt of lens in [rad].
 |
 |  Method resolution order:
 |      Quadrupole
 |      Element
 |      builtins.object
 |
 |  Methods defined here:
 |
 |  __init__(self, l=0.0, k1=0, k2=0.0, tilt=0.0, eid=None)
 |      Initialize self.  See help(type(self)) for accurate signature.
 |
 |  ----------------------------------------------------------------------
 |  Methods inherited from Element:
 |
 |  __eq__(self, other)
 |      Return self==value.
 |
 |  __hash__(self)
 |      Return hash(self).
 |
 |  ----------------------------------------------------------------------
 |  Data descriptors inherited from Element:
 |
 |  __dict__
 |      dictionary for instance variables (if defined)
 |
 |  __weakref__
 |      list of weak references to the object (if defined)
```

Tutorials

- Preliminaries: Setup & introduction

Beam dynamics

- Introduction. Tutorial N1. Linear optics.. Web version.
  - Linear optics. Double Bend Achromat (DBA). Simple example of usage OCELOT functions to get periodic solution for a storage ring cell.
- Tutorial N2. Tracking.. Web version.
  - Linear optics of the European XFEL Injector.
  - Tracking. First and second order.
  - Artificial beam matching - BeamTransform
- Tutorial N3. Space Charge.. Web version.
  - Tracking through RF cavities with SC effects and RF focusing.
- Tutorial N4. Wakefields.. Web version.
  - Tracking through corrugated structure (energy chirper) with Wakefields
- Tutorial N5. CSR.. Web version.
  - Tracking trough bunch compressor with CSR effect.
- Tutorial N6. RF Coupler Kick.. Web version.
  - Coupler Kick. Example of RF coupler kick influence on trajjectory and optics.
- Tutorial N7. Lattice design.. Web version.
  - Lattice design, twiss matching, twiss backtracking
- Tutorial N8. Physics process addition. Laser heater. Web version.
  - Theory of Laser Heater, implementation of new Physics Process, track particles w/o laser heater effect.
- Tutorial N9. Simple accelerator based THz source. Web version.
  - A simple accelerator with the electron beam formation system and an undulator to generate THz radiation.

Photon field simulation

- PFS tutorial N1. Synchrotron radiation module. Web version.
  - Simple examples how to calculate synchrotron radiation with OCELOT Synchrotron Radiation Module.
- PFS tutorial N2. Coherent radiation module and RadiationField object. Web version.
- PFS tutorial N3. Reflection from imperfect highly polished mirror. Web version.
- PFS tutorial N4. Converting synchrotron radiation Screen object to RadiationField object for viewing and propagation. Web version.
- PFS tutorial N5: SASE estimation and imitation. Web version.

Appendixes

- Undulator matching. Web version.
  - brief theory and example in OCELOT

HELMHOLTZ GEMEINSCHAFT
DESY

Unit-tests

```
unit_tests/adaptors_test/elegant_lattice/elegant_lattice_test.py ....s                    [  1%]
unit_tests/ebeam_test/acc_utils/acc_utils_test.py ....s                                   [  3%]
unit_tests/ebeam_test/csr_ex/csr_ex_test.py ..............s                               [  9%]
unit_tests/ebeam_test/dba/dba_test.py ....s                                               [ 11%]
unit_tests/ebeam_test/dba_track_ellipse/dba_track_ellipse_test.py ...s                    [ 12%]
unit_tests/ebeam_test/dba_tracking/dba_tracking_test.py .....s                            [ 14%]
unit_tests/ebeam_test/dogleg/dogleg_test.py .......s                                      [ 17%]
unit_tests/ebeam_test/io/io_test.py .....s                                                [ 19%]
unit_tests/ebeam_test/io_lattice/io_lattice_test.py ................s                  [ 26%]
unit_tests/ebeam_test/laser_heater/laser_heater_test.py .........s                        [ 29%]
unit_tests/ebeam_test/linac_orb_correct/linac_orb_correct_test.py ........s               [ 33%]
unit_tests/ebeam_test/matching/match_test.py .........s                                   [ 36%]
unit_tests/ebeam_test/multipoles/multipoles_test.py .....s                                [ 39%]
unit_tests/ebeam_test/navi/navi_test.py ...............s                                   [ 45%]
unit_tests/ebeam_test/non_disp_offset/non_disp_offset_test.py ......s                     [ 47%]
unit_tests/ebeam_test/phys_proc/phys_proc_test.py ...................s                   [ 54%]
unit_tests/ebeam_test/rf_twiss/rf_twiss_test.py ......s                                   [ 57%]
unit_tests/ebeam_test/ring_orb_correct/ring_orb_correct_test.py .....s                    [ 59%]
unit_tests/ebeam_test/rk_track/rk_track_test.py ....s                                     [ 61%]
unit_tests/ebeam_test/section_track/section_track_test.py ......................s         [ 69%]
unit_tests/ebeam_test/space_charge/space_charge_test.py .......s                          [ 72%]
unit_tests/ebeam_test/storage_ring/storage_ring_test.py ......s                           [ 74%]
unit_tests/ebeam_test/storage_ring_da/storage_ring_da_test.py .......s                    [ 77%]
unit_tests/ebeam_test/triplet/triplet_test.py ......s                                     [ 80%]
unit_tests/ebeam_test/tune_shift/tune_shift_test.py ....s                                 [ 82%]
unit_tests/ebeam_test/twiss_sase3/twiss_sase3_test.py ...s                                [ 83%]
unit_tests/ebeam_test/undulator/undulator_test.py ....s                                   [ 85%]
unit_tests/ebeam_test/wake/wake_test.py .s                                                [ 86%]
unit_tests/sr_test/rad_beam/rad_beam_test.py ..............s                              [ 91%]
unit_tests/sr_test/spatial/spatial_test.py .......s                                       [ 94%]
unit_tests/sr_test/spect_mag_file/spect_mag_file_test.py ...s                             [ 96%]
unit_tests/sr_test/spectrum/spectrum_test.py .....s                                       [ 98%]
unit_tests/wave_test/generate_gaussian_dfl/generate_gaussian_dfl_test.py ...s             [100%]
====================== 238 passed, 33 skipped, 33 warnings in 538.50s (0:08:58) ======================
```

- Main functions and modules are covered with unit-tests
- But bugs are unavoidable, if found any please report it:

https://github.com/ocelot-collab/ocelot/issues

Thank you for your attention

And thanks to all contributors (the order is random ☺):

G. Geloni, S. Serkez, T. Tanikawa – *the European XFEL*

I. Agapov, M. Dohlus, W. Kuropka, A. Ivanov, Á. Ferran Pousa, I. Zagorodnov – *DESY*

J. Duris, H. Slepicka - *SLAC*

E. Fomin – *Kurchatov Institute*

A. Trebushinin – *BINP (Novosibirsk)*

M. Veremchuk – *Kyiv National University*

….