Modeling and optimization of accelerators with OCELOT

Sergey Tomin

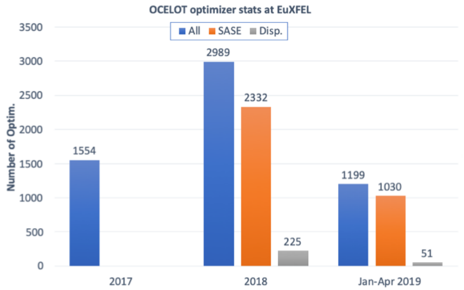
Joint DESY and University of Hamburg Accelerator Physics Seminar
24 September 2019

Introduction and OCELOT overview

- Started as simulation project (spontaneous radiation, FEL) at European XFEL at 2012
  - *I. Agapov, G.Geloni, S.Tomin, I.Zagorodnov, NIM A. 768, 2014*
- Beam dynamics module was developed (linear optics, collective effects, second order effects).
  - *S.Tomin, I. Agapov, M.Dohlus, I.Zagorodnov.* doi:10.18429/JACoW-IPAC2017-WEPAB031, 2017
- Advanced FEL simulations including the electron beam dynamics effects
  - *S. Serkez et al, Journal of Optics, Volume 20, Number 2, 024005, 2018*
- Turned into more on-line control-oriented development
  - *Orbit correction tool, Optimizer, adaptive feedback*
    - ► S. Tomin and A. Valentinov, On-Line Beam Control With OCELOT at Siberia-2, *IPAC2014*
    - ► *I. Agapov, G.Geloni, S.Tomin, I.Zagorodnov*, *arXiv:1704.02335, 2017*
    - ► S. Tomin et al., doi:10.18429/JACoW-ICALEPCS2017-WEAPL07, 2018

OCELOT structure

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

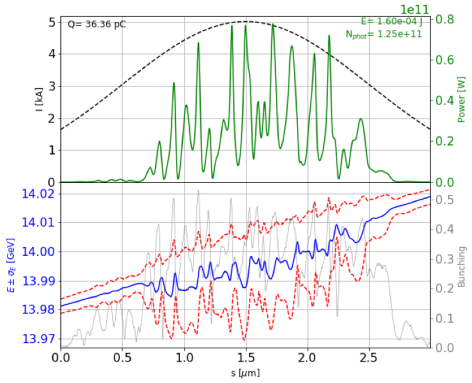


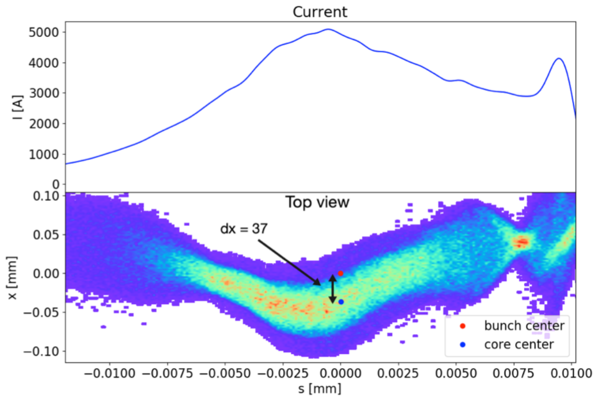
Data sets

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

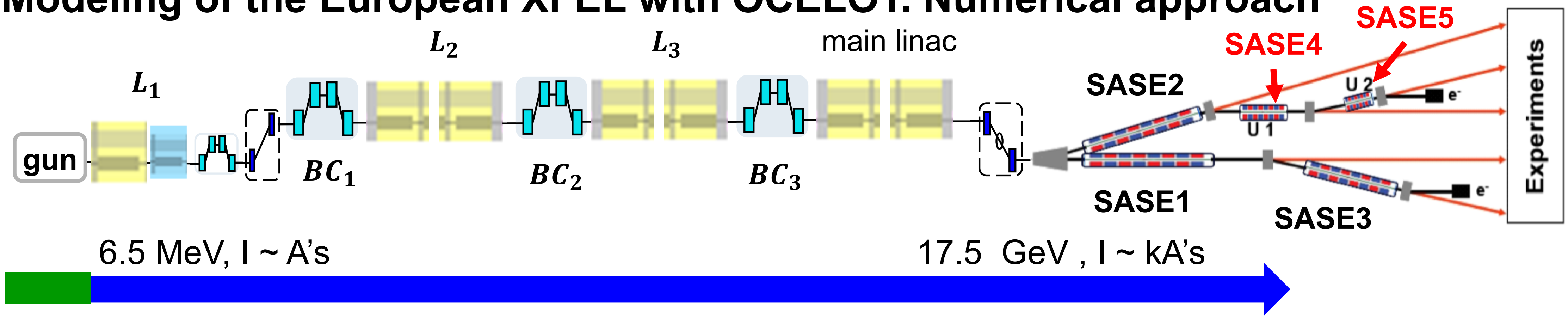
Charged Particle Beam Dynamics (CPBD) module (linacs, rings)

Online beam control
- Orbit correction
- Adaptive FB
- Optimizer

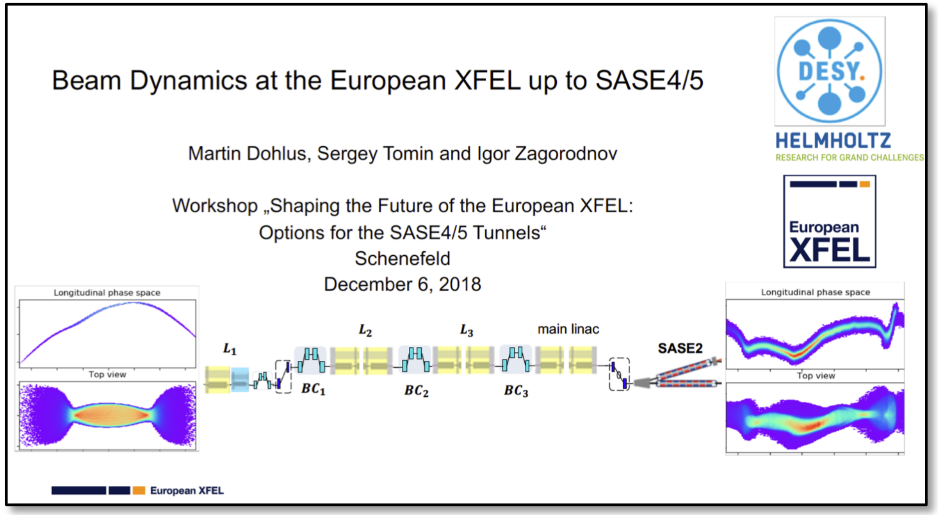




Modeling of the European XFEL with OCELOT. Numerical approach



Krack3 / ASTRA



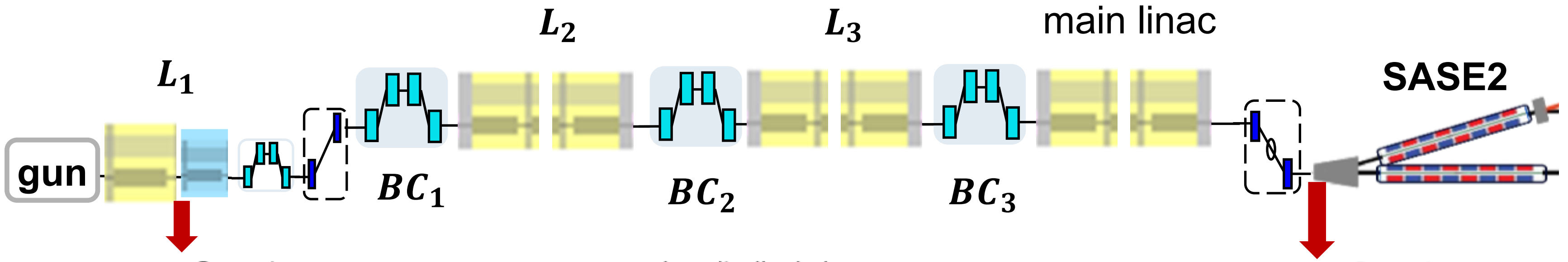
OCELOT

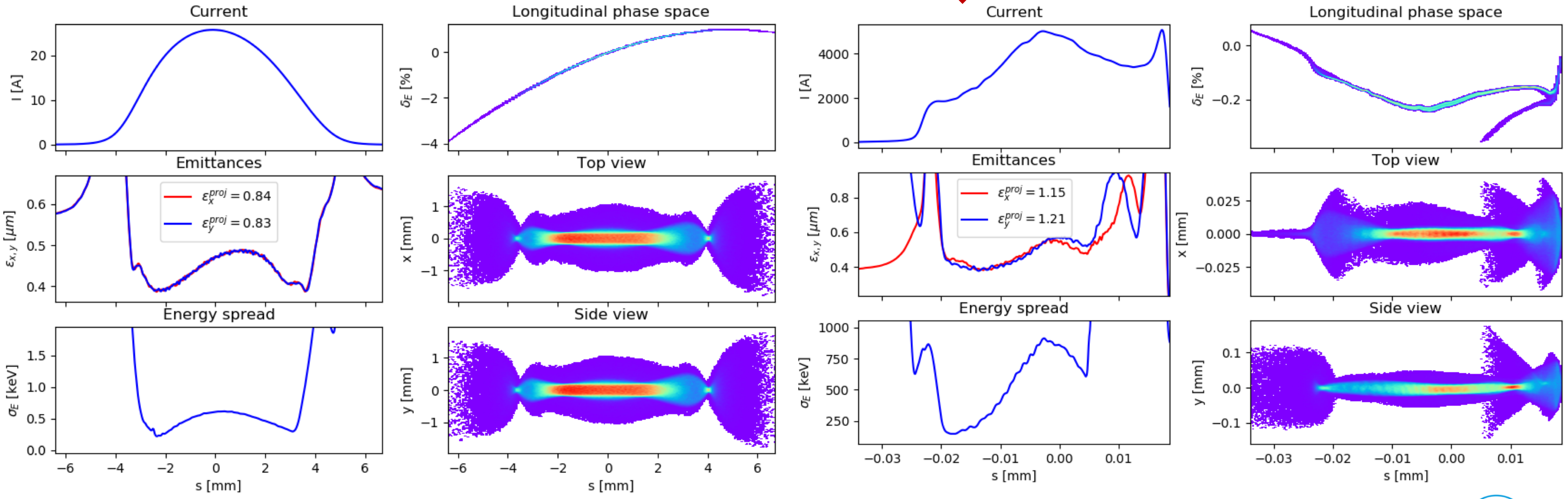
- 3D Space Charge
- 3D wakefields
- 1D CSR
- particle motion with transport matrices of second order
- RF cavity with Rosenzweig-Serafini model
- Chamber with wakes
- ISR and quantum fluctuations

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

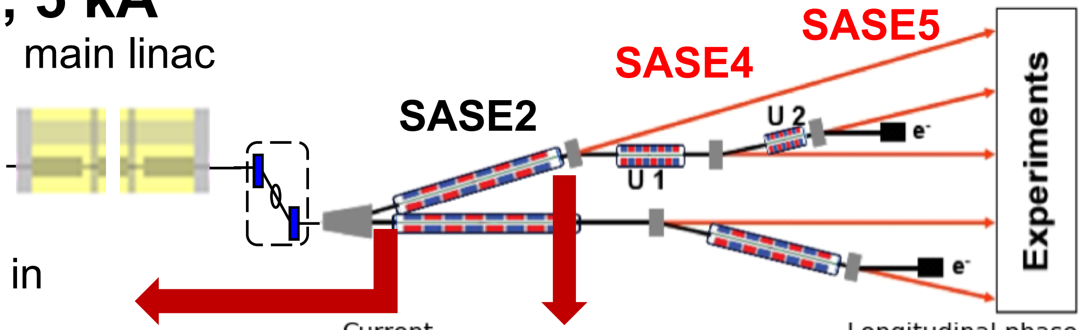
- projected x-emittance growth by 37%
- projected y-emittance growth by 44%

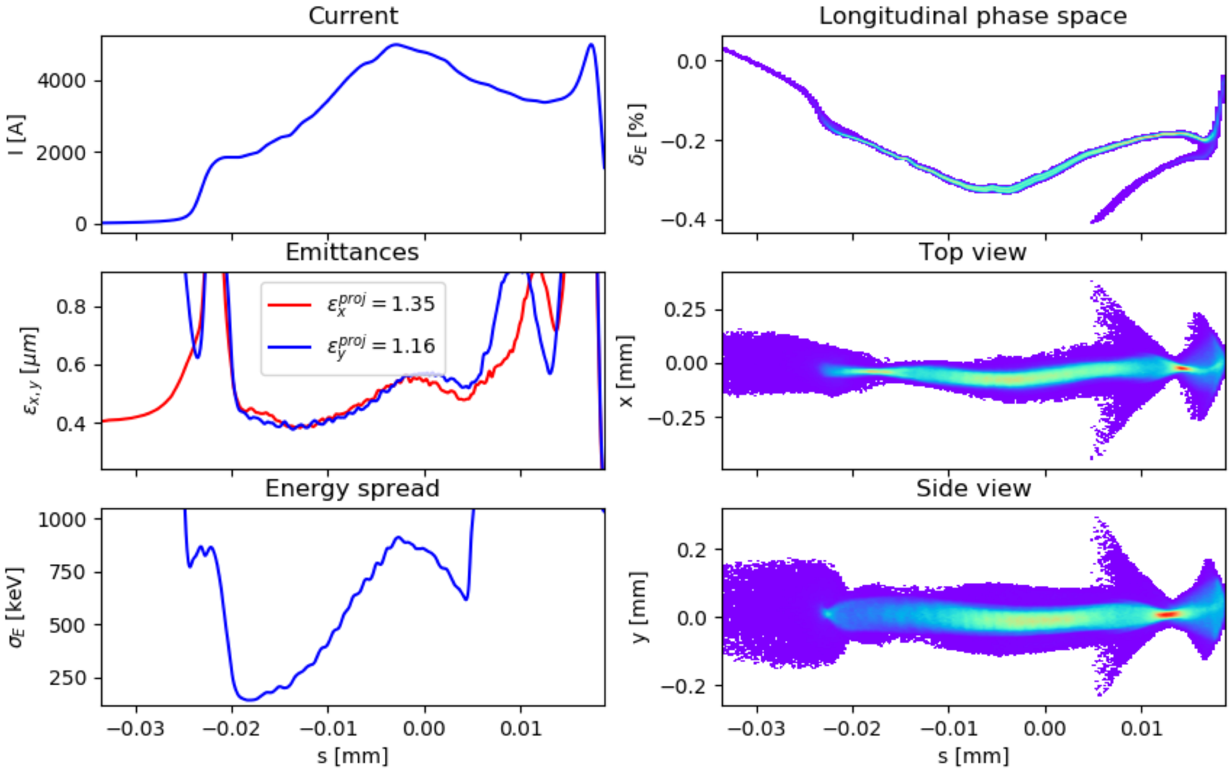


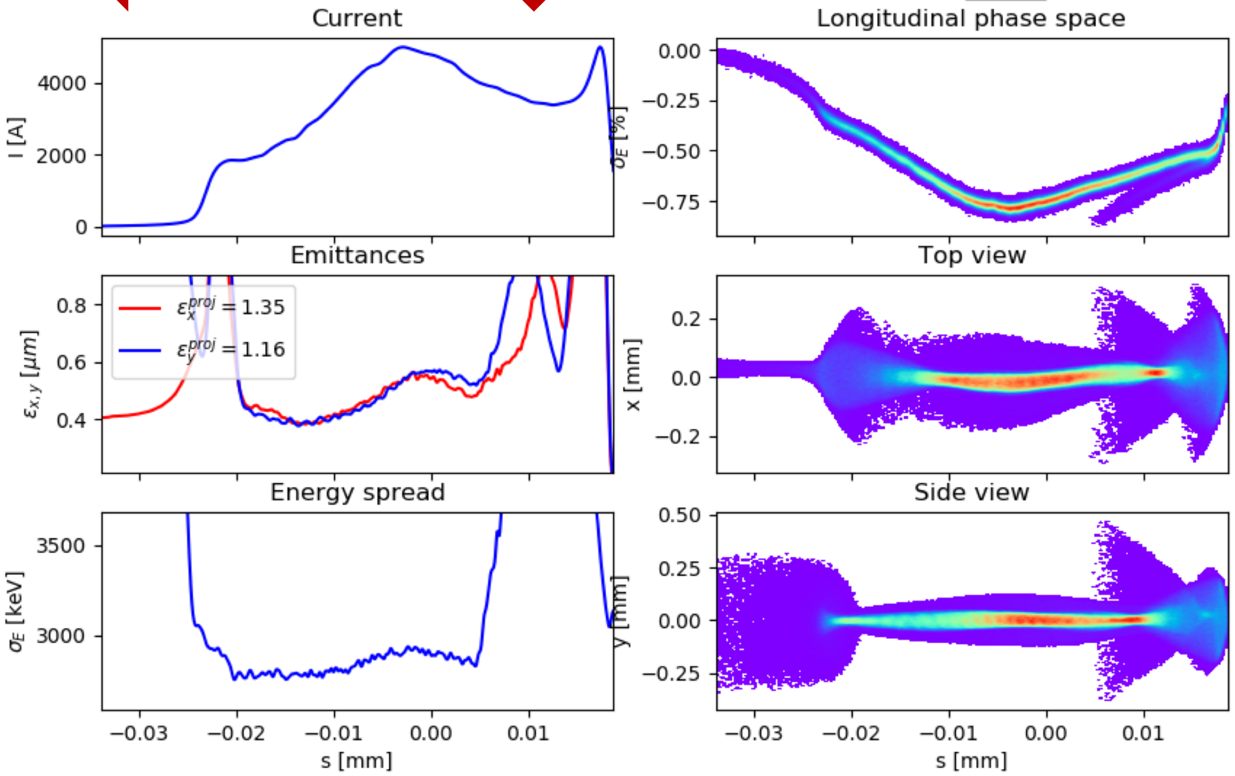


Modeling #1. Beam dynamics for 500 pC, 5 kA

- 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



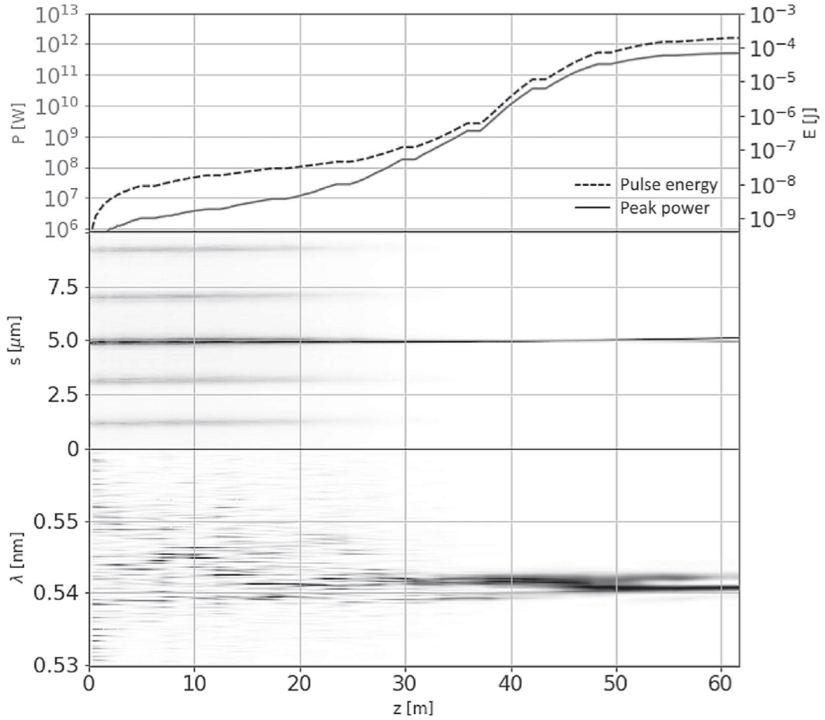


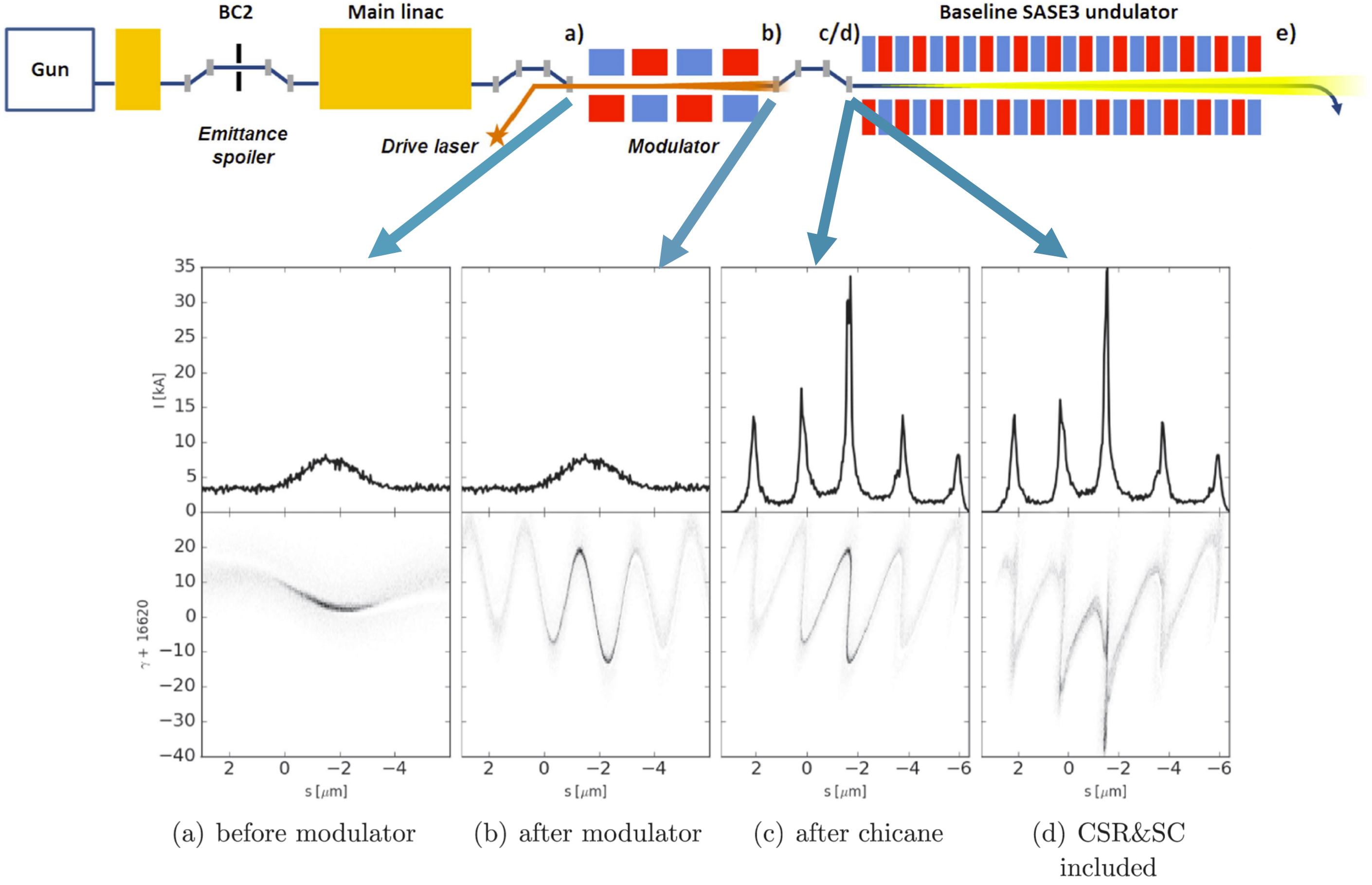


Overview of options for generating high-brightness attosecond x-ray pulses

PFS ⟷ **CPBD**

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





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*

JOURNAL OF SYNCHROTRON RADIATION
ISSN 1600-5775

Dynamical effects on superradiant THz emission from an undulator

Gianluca Geloni, Takanori Tanikawa and Sergey Tomin*

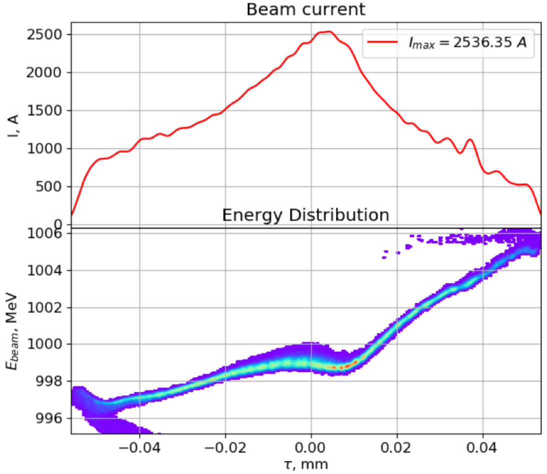
PFS ⟷ CPBD

- Motivation
  - In some cases, from experimental results follows that intensity of THz radiation is underestimated by theory.

- One of the explanations can be a presence of the energy chirp in the electron beam which is often a case for accelerator based THz sources.

Electron beam current and energy distribution [DESY S2E simulation web page].

FLASH

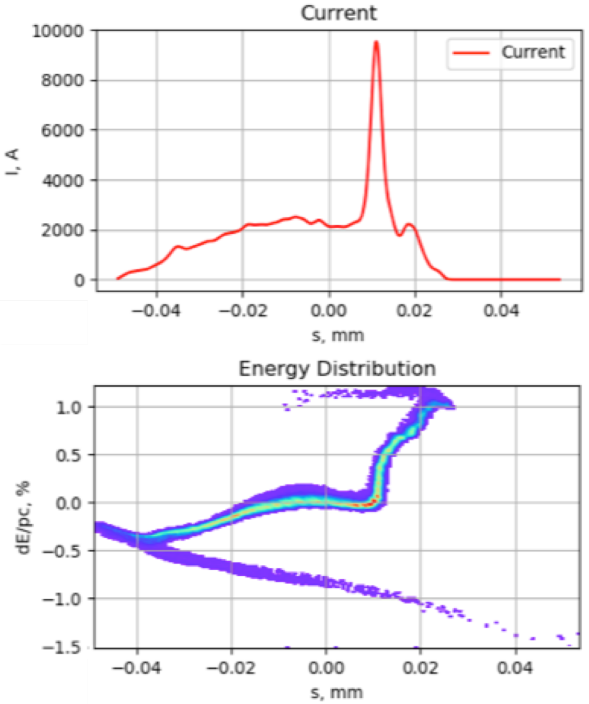


THz Undulator

Eb = 600 MeV

$\lambda_{ph} = 145\ \mu m$

$R_{56} \approx -2.7\ mm$



JSR JOURNAL OF SYNCHROTRON RADIATION

ISSN 1600-5775

Dynamical effects on superradiant THz emission from an undulator

Gianluca Geloni, Takanori Tanikawa and Sergey Tomin*

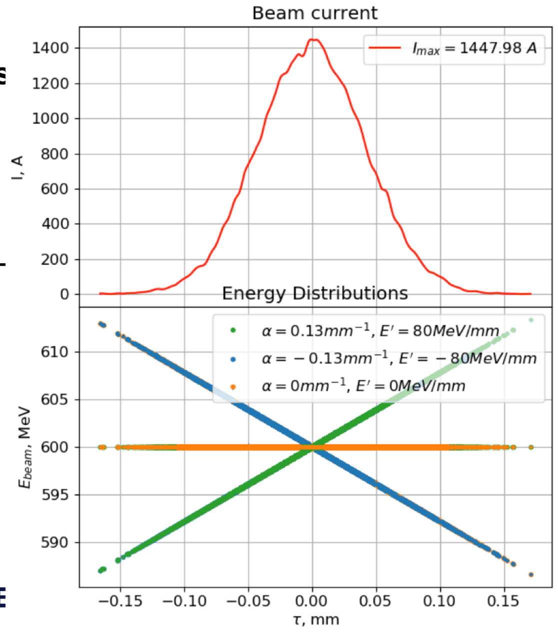
PFS ⟷ CPBD

- Motivation
  - In some cases, from experimental results follows that intensity of THz radiation is underestimated by theory.

- One of the explanations can be a presence of the energy chirp in the electron beam which is often a case for accelerator based THz sources.

dis ... page].

- FLASH

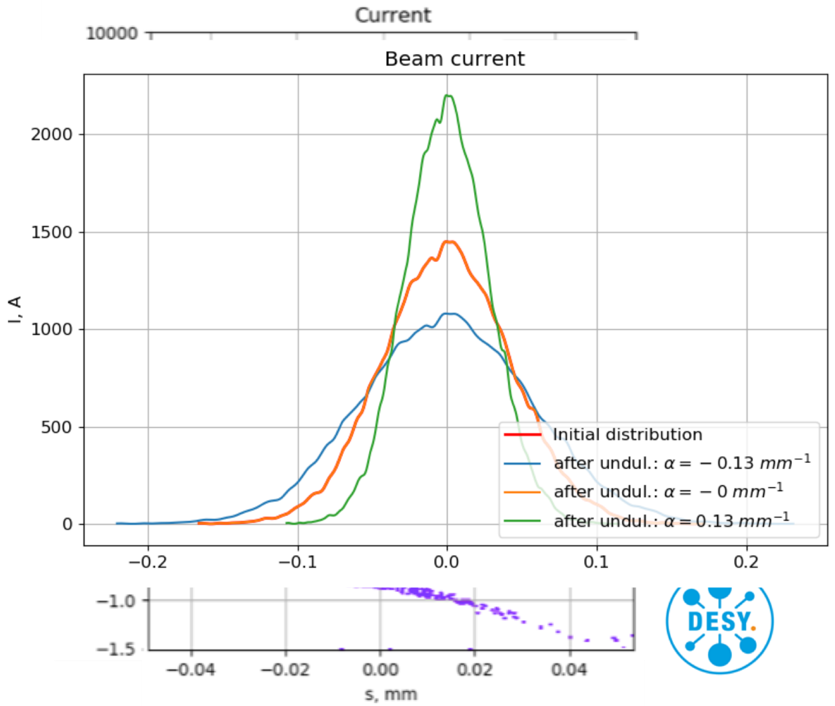


- THz Undulator

Eb = 600 MeV

$\lambda_{ph} = 145\ \mu m$

$R_{56} \approx -2.7\ mm$



JOURNAL OF SYNCHROTRON RADIATION
ISSN 1600-5775

Dynamical effects on superradiant THz emission from an undulator

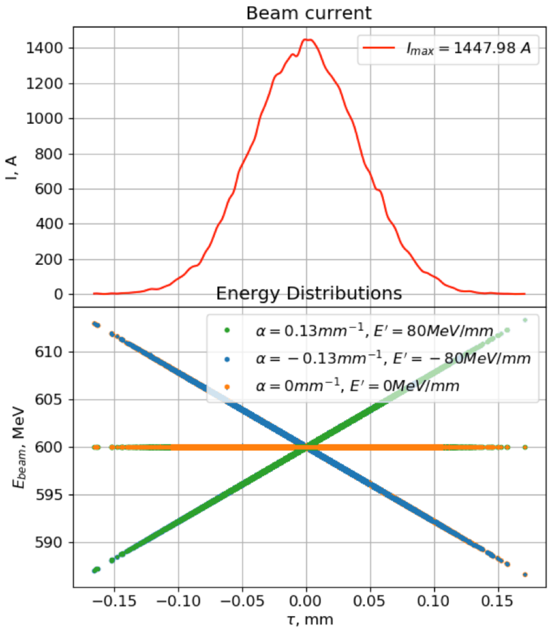
Gianluca Geloni, Takanori Tanikawa and Sergey Tomin*

PFS ⟷ CPBD

- Motivation
  - In some cases, from experimental results follows that intensity of THz radiation is underestimated by theory.
- One of the explanations can be a presence of the energy chirp in the electron beam which is often a case for accelerator based THz sources.

dis ... page].

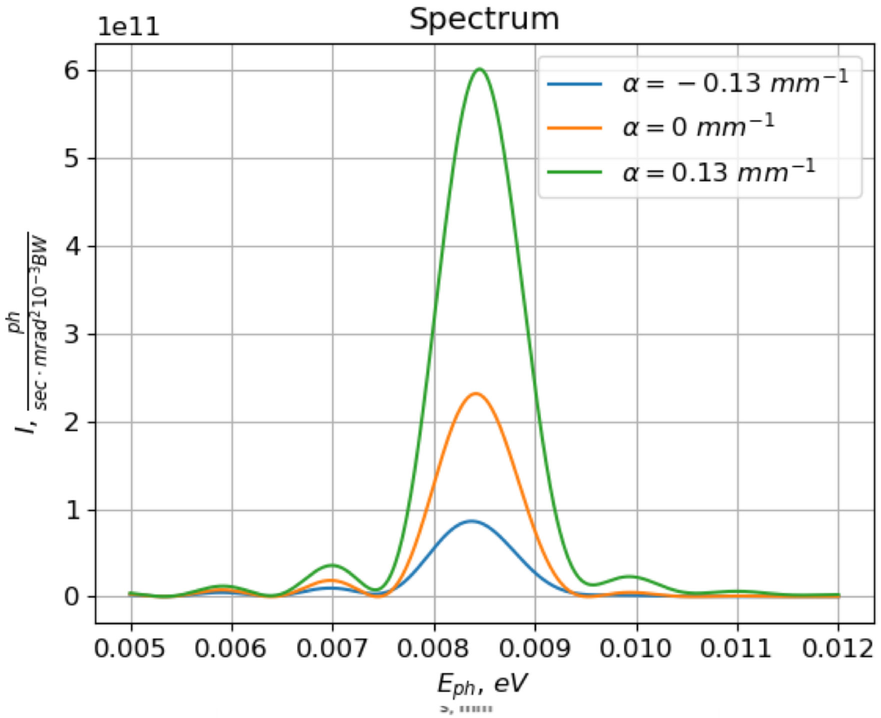
- FLASH
- THz Undulator



Eb = 600 MeV

$\lambda_{ph} = 145\ \mu m$

$R_{56} \approx -2.7\ mm$



OCELOT in plasma physics

nature COMMUNICATIONS

Corrected: Publisher correction

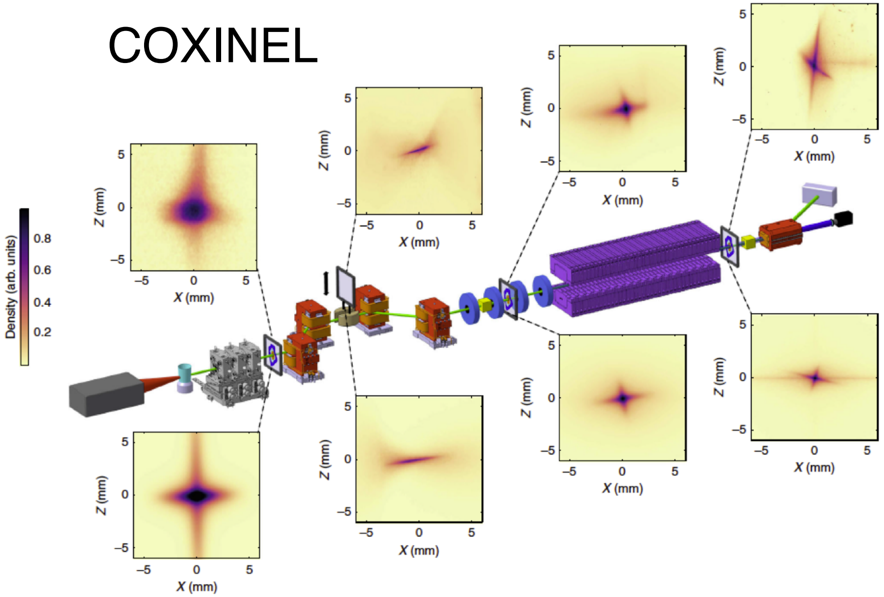
ARTICLE

DOI: 10.1038/s41467-018-03776-x OPEN

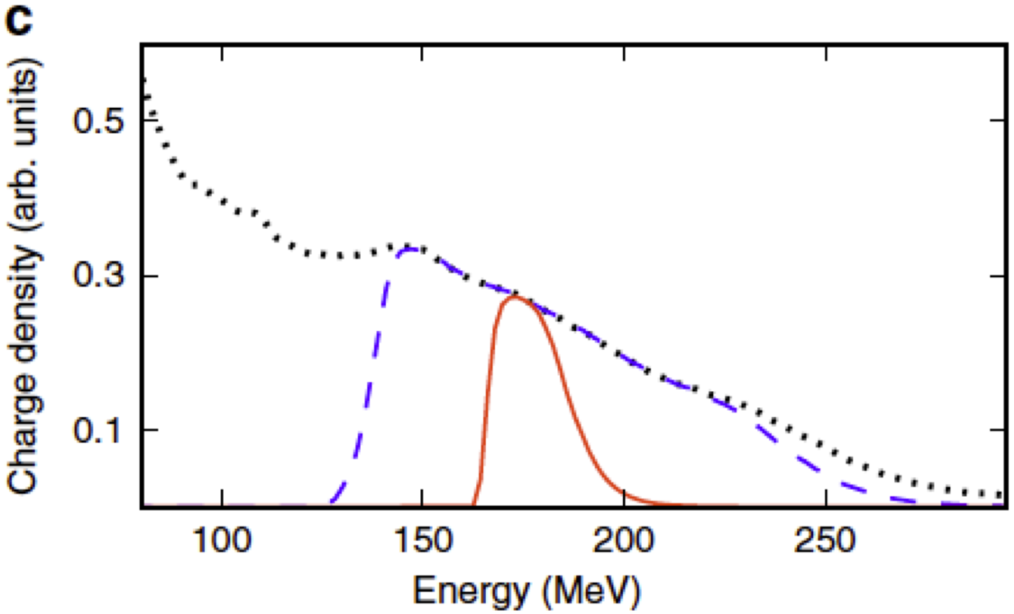
Control of laser plasma accelerated electrons for light sources

T. André[1,2], I.A. Andriyash[1], A. Loulergue[1], M. Labat[1], E. Roussel[1,3], A. Ghaith[1,2], M. Khojoyan[1], C. Thaury[4], M. Valléau[1], F. Briquez[1], F. Marteau[1], K. Tavakoli[1], P. N'Gotta[1], Y. Dietrich[1], G. Lambert[4], V. Malka[4,5], C. Benabderrahmane[1], J. Vétéran[1], L. Chapuis[1], T. El Ajjouri[1], M. Sebdaoui[1], N. Hubert[1], O. Marcouillé[1], P. Berteaud[1], N. Leclercq[1], M. El Ajjouri[1], P. Rommeluère[1], F. Bouvet[1], J.-P. Duval[1], C. Kitegi[1], F. Blache[1], B. Mahieu[4], S. Corde[4], J. Gautier[4], K. Ta Phuoc[4], J.P. Goddet[4], A. Lestrade[1], C. Herbeaux[1], C. Évain[3], C. Szwaj[3], S. Bielawski[3], A. Tafzi[4], P. Rousseau[4], S. Smartsev[4,5], F. Polack[1], D. Dennetière[1], C. Bourassin-Bouchet[1], C. De Oliveira[1] & M.-E. Couprie[1,2]

COXINEL



"… Beam transport until the undulator is simulated with the OCELOT tracking code, which was modified to enable the large spectra beam modelling…"



c Spectrum measured at the exit of the electron source (dotted), and simulated at the entrance of the undulator after transport in the line (dashed), with the 4mm slit (solid curve)

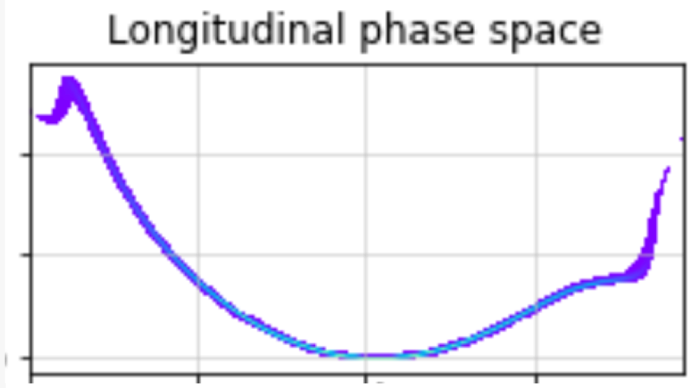
Interface for implementation of new methods

Tutorial N8. Physics process addition. Laser heater

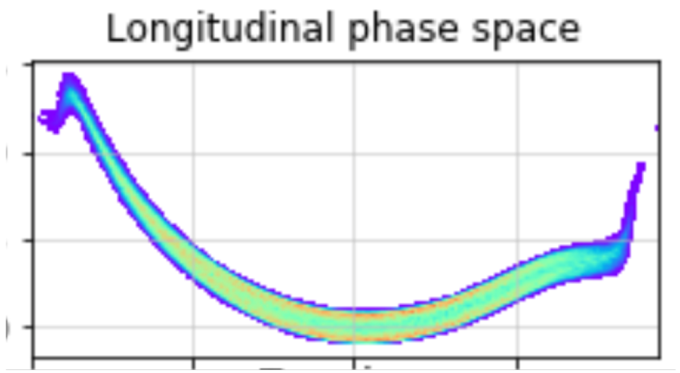
The OCELOT Charged Particle Beam Dynamics (CPBD) module includes various physics processes like CSR, SpaceCharge, Wake3D, BeamTransform. There are other "physics processes", which do not have anything in common with physics but use the same interface (or parent class "PhysProc") e.g. SaveBeam. Using this interface, one can implement own "physics process". In this tutorial we will show how to do it on the example of the laser heater.

```
class LaserHeater(PhysProc):
    def __init__(self, step=1):
        PhysProc.__init__(self, step=1)
        # amplitude of energy modulation on axis
        self.dE = 12500e-9 # GeV
        self.Ku = 1.294 # undulator parameter
        self.Lu = 0.8 # [m] - undulator length
        self.lperiod = 0.074 # [m] - undulator period length
        self.sigma_l = 300e-6 # [m]
        self.sigma_x = self.sigma_l
        self.sigma_y = self.sigma_l
        self.x_mean = 0
        self.y_mean = 0

    def apply(self, p_array, dz):
        gamma = p_array.E/m_e_GeV
        lbda_ph = self.lperiod/(2*gamma**2)*(1+self.Ku**2/2)
        k_ph = 2*np.pi/lbda_ph
        pc = np.sqrt(p_array.E**2 - m_e_GeV**2)

        A = self.dE/(pc)*dz/self.Lu
        dx = p_array.x()[:] - self.x_mean
        dy = p_array.y()[:] - self.y_mean
        p_array.p()[:] += A*np.cos(k_ph*p_array.tau()[:])*np.exp(
                    -0.25*dx**2/self.sigma_x**2
                    -0.25*dy**2/self.sigma_y**2)
```

Before Laser heater



After Laser heater



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

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

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
```

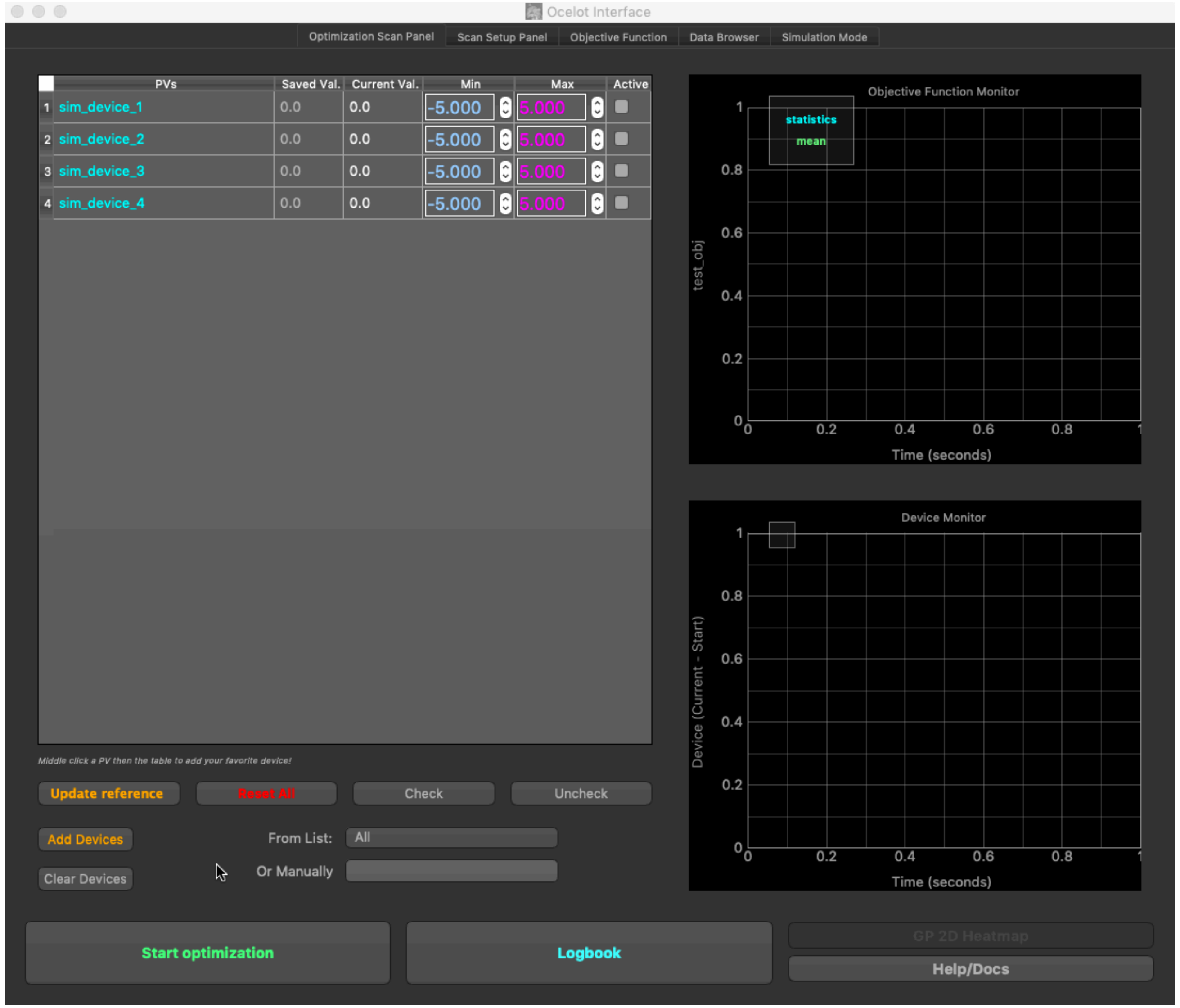
Accuracy of the results

- Beam dynamics without collective effects
  - the second-order optics cross-checked with MAD
  - the chromatic effects in dispersive sections are cross-checked with CSRtrack
- Beam dynamics with collective effects
  - the space charge is cross-checked with ASTRA and with analytical estimations
  - CSR in dispersive sections is cross-checked with CSRtrack
  - the results are checked by increasing of macroparticle number 200k -> 2M
  - the results are checked by varying of sampling/mesh parameters in wakes, CSR, space-charge
- Photon field simulation module
  - Spontaneous radiation module is cross-checked with Spectra and SRW
  - Wavefront propagation functions is cross-checked with SRW

Optimization of accelerators

OCELOT Optimizer

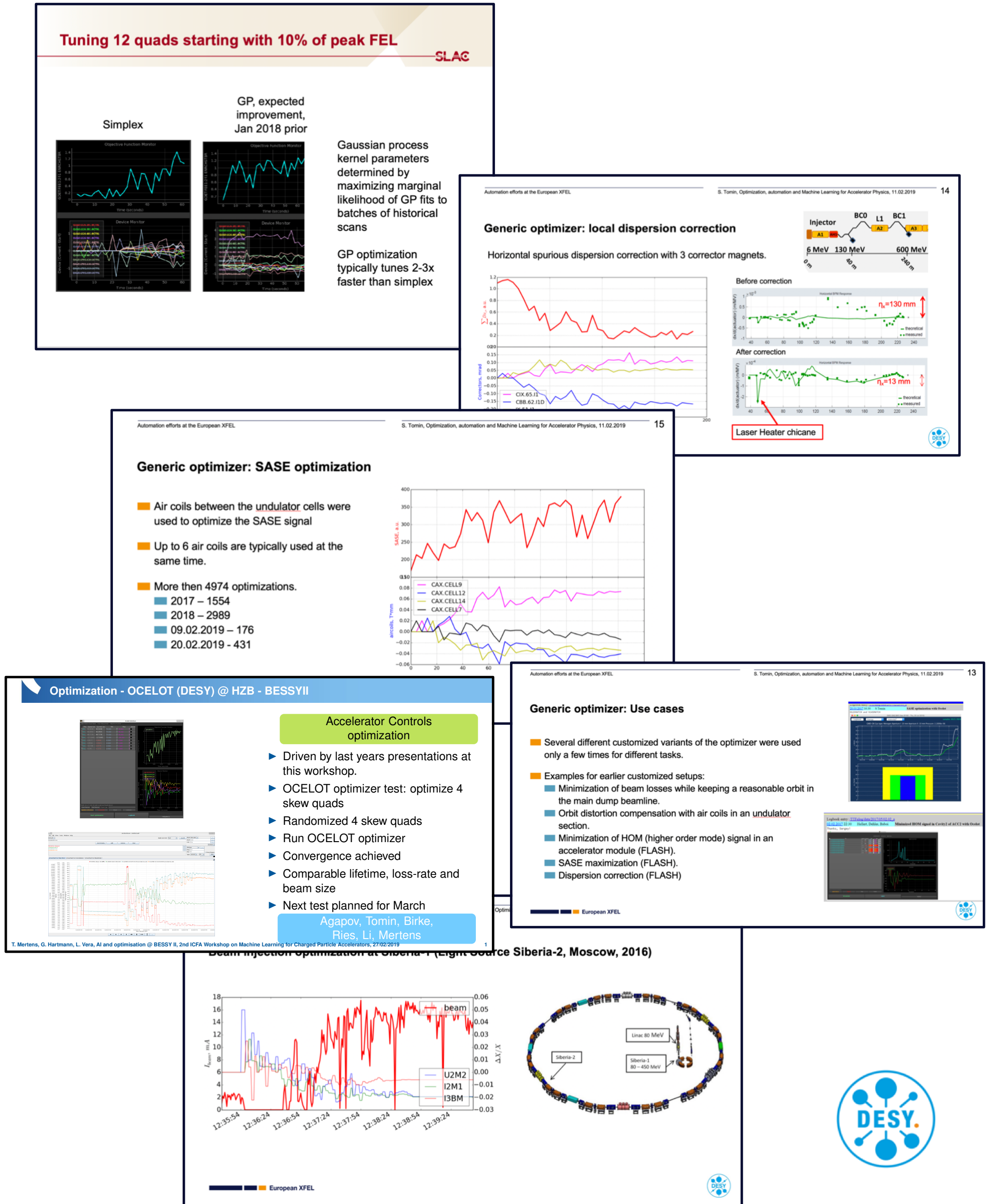
- Optimization algorithms are faster than scanning
- OCELOT optimizer is a flexible platform for optimization:
  - Interchangeable optimization methods
  - GUI
    - ► Add/select device or group of devices
    - ► Craft/modify target function
  - Infrastructure for testing new methods
  - Save/load configs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



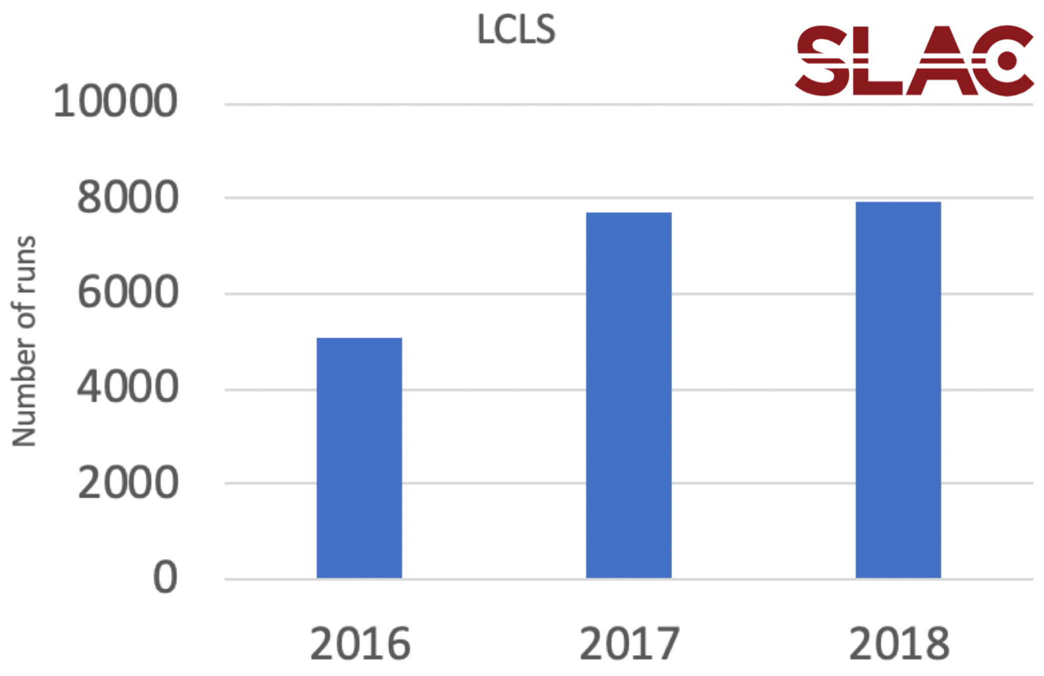
- I. Agapov et al, arXiv:1704.02335
- S. Tomin et al, https://doi.org/10.18429/JACoW-IPAC2017-WEPAB031
- M.W. McIntire et al, DOI:10.18429/JACoW-IPAC2016-WEPOW055

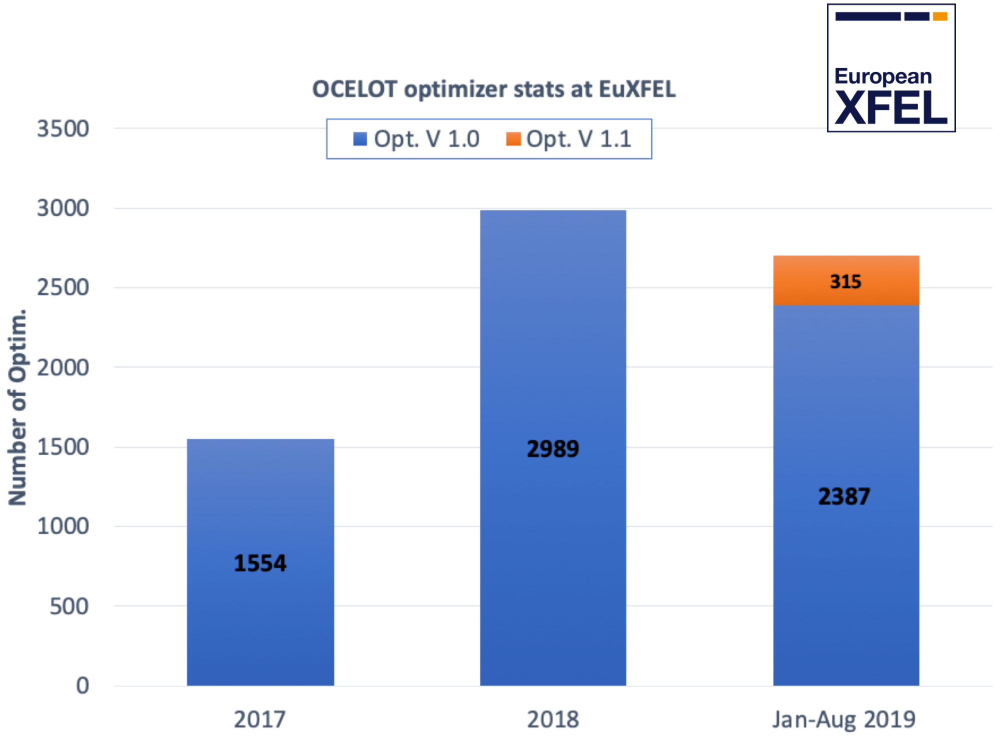
Use cases

- FEL facilities
  - FEL pulse energy maximization:
    - ► Launch orbit and orbit inside an undulator (EuXFEL)
    - ► Phase-shifters (EuXFEL)
    - ► Orbit in low energy sections (EuXFEL)
    - ► Matching quads (LCLS & EuXFEL)
    - ► RF settings (EuXFEL)
  - Local dispersion correction (EuXFEL & FLASH)
  - HOM signal minimization in cavities (FLASH)
- Storage rings:
  - Injection efficiency optimization (Kurchatov Institute)
  - Beam life time (test at BESSY-II)



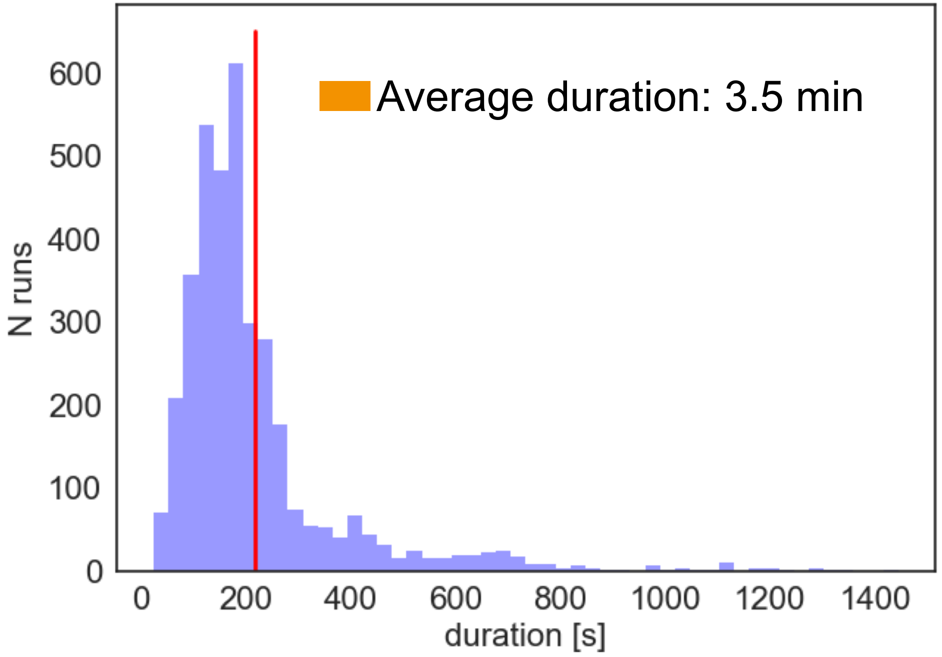
OCELOT Optimizer: Statistics

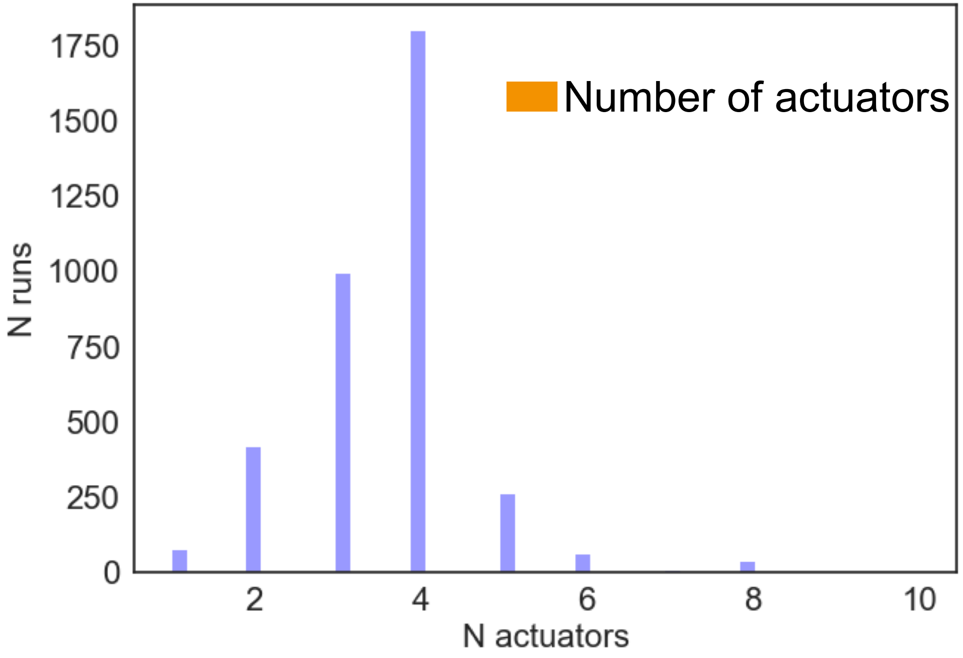




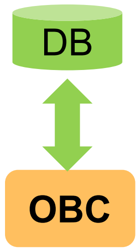
SLAC: Ocelot with Nelder-Mead simplex reduced tuning times on average by 25% to 50% compared to hand tuning

OCELOT Optimizer: Statistics



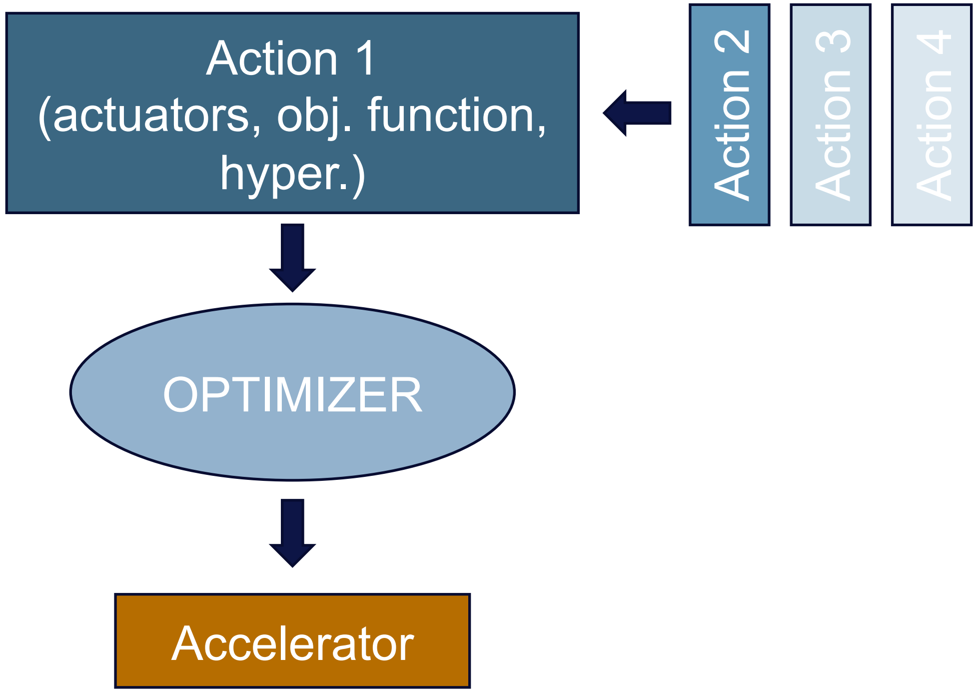


In most optimizations 4 devices are used and the average time duration of a single optimization is 3.5 minutes

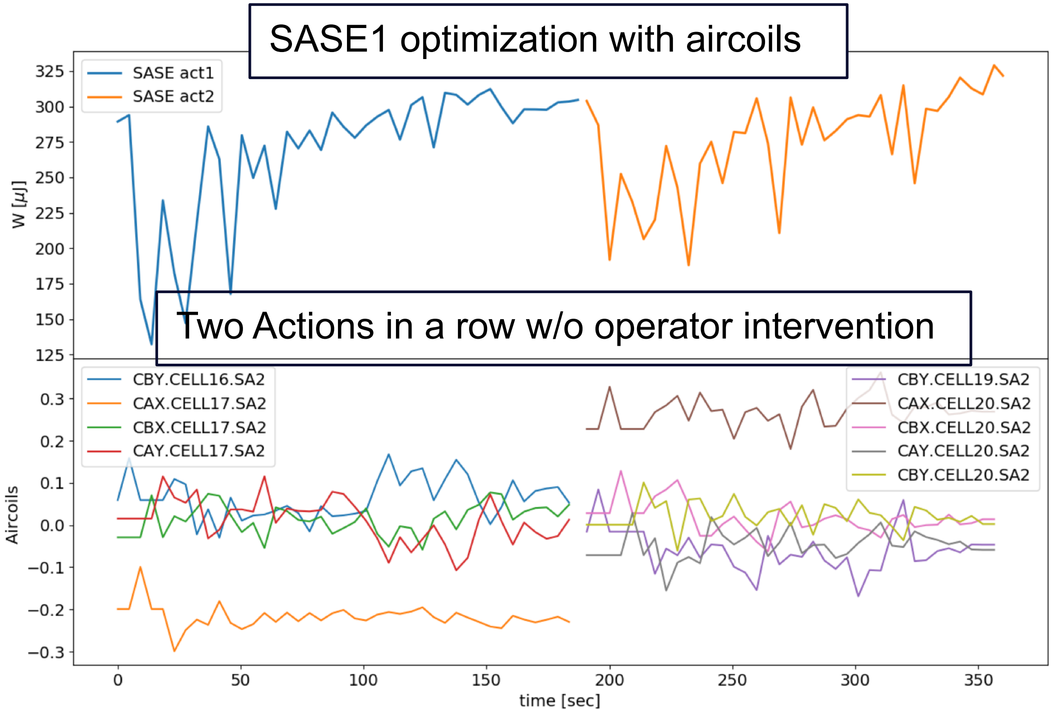


Sequence of optimizations: automatic optimization

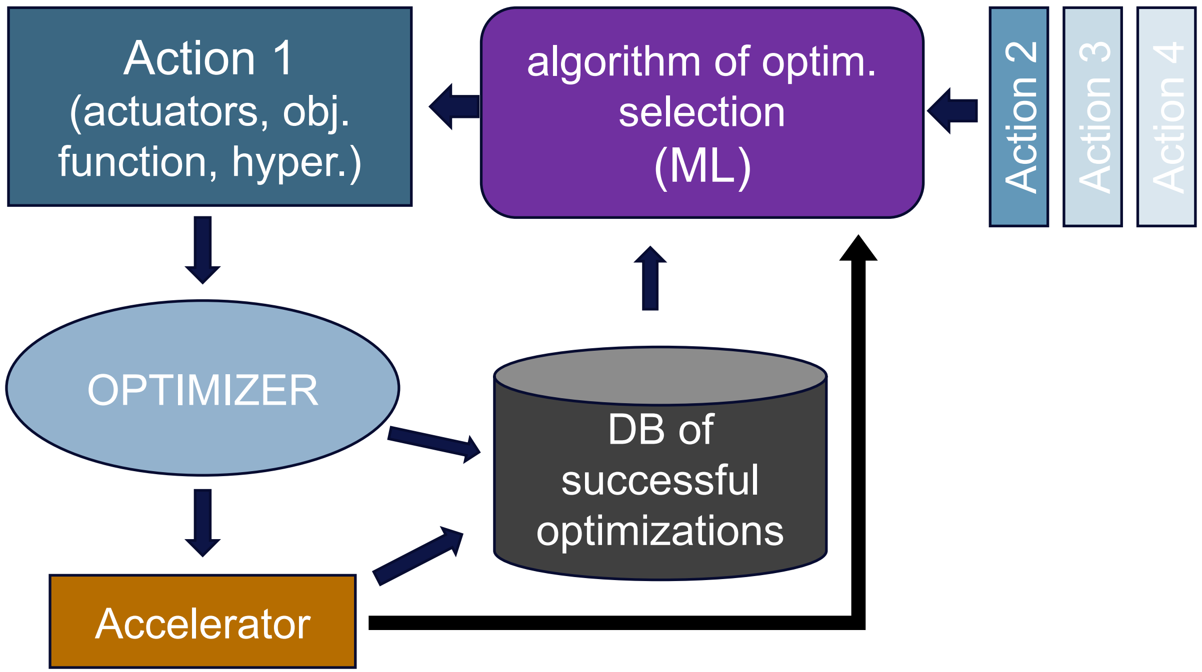
- Optimization with small number of actuators (4-6) is more efficient than with many due to noise and slow drifts



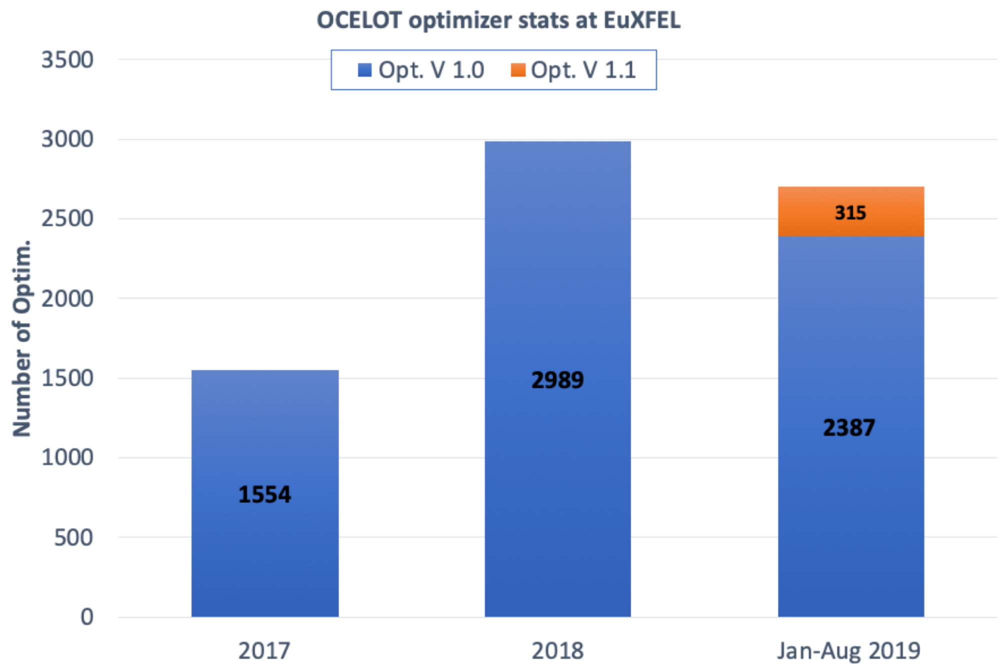
- New version (v1.1) was deployed
- Predefined sequence of optimization
  - without operator intervention
  - Optimizer monitors machine state – paused optimization if necessary.



Sequence of optimizations: even more automation with ML?



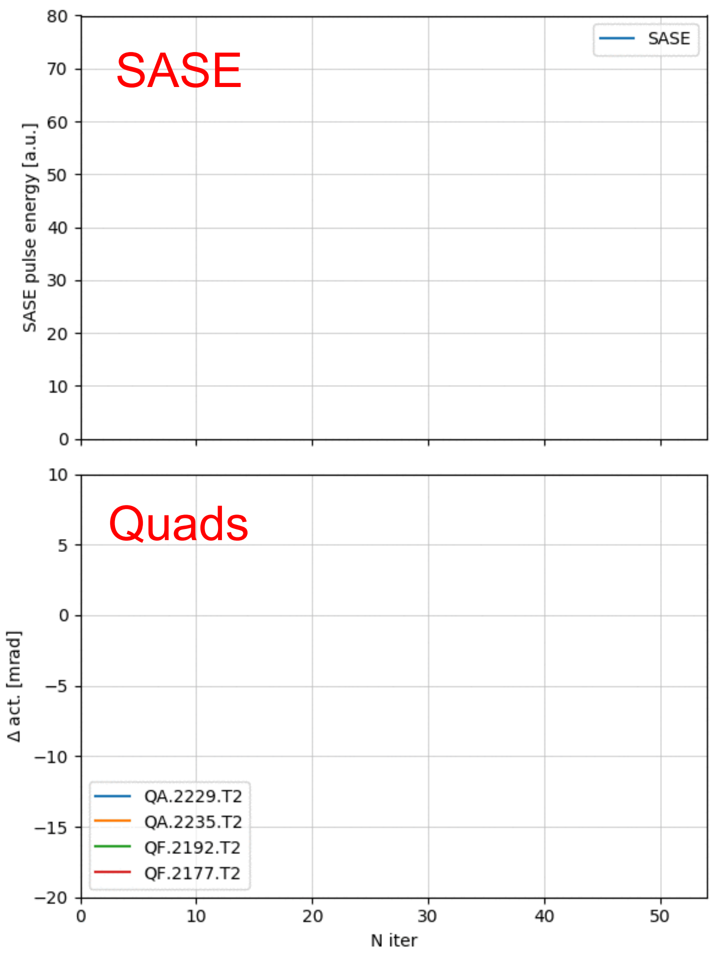
More logged DOOCS channels about machine state



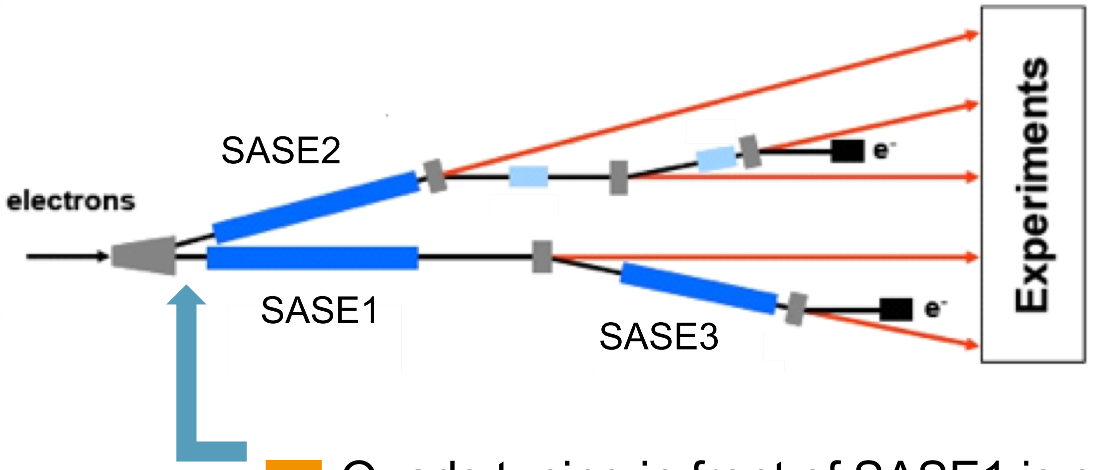
I. Agapov, G. Geloni, S. Tomin, I. Zagorodnov. arXiv:1704.02335, 2017

Hyperparameters. Beam matching

SASE optimization with 4 quads

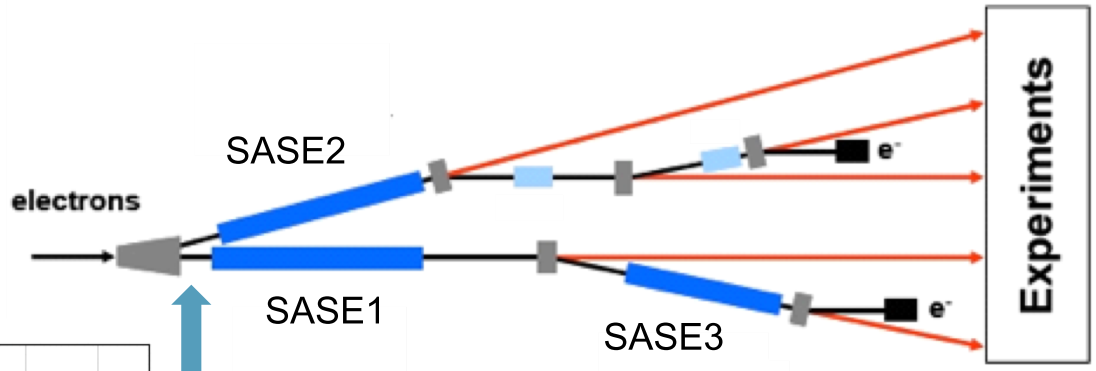


- Initial quad step
- Group of quads

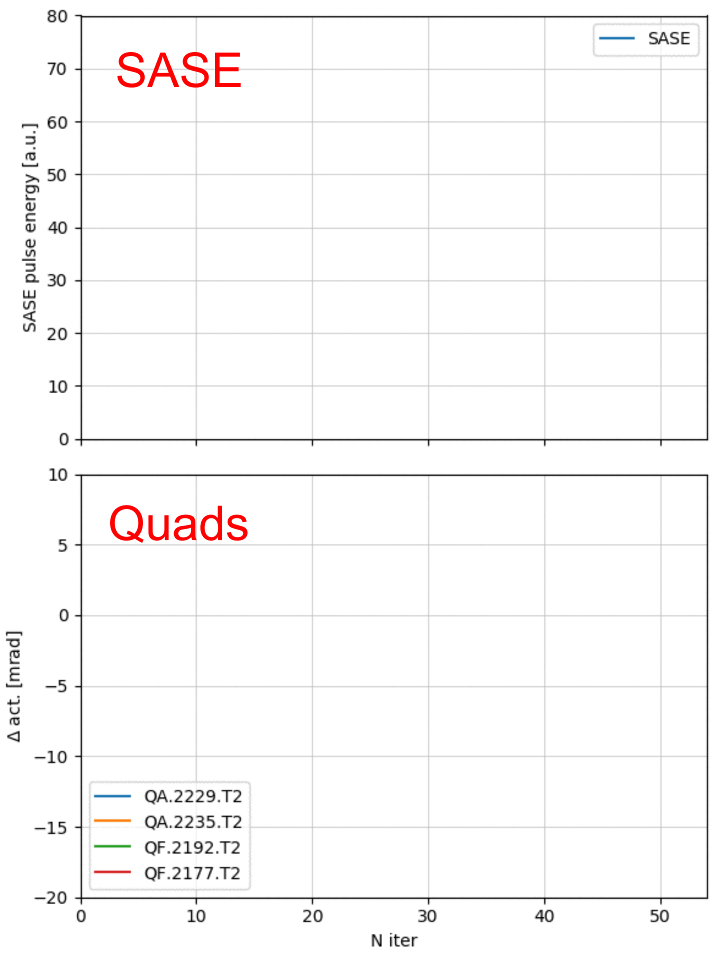


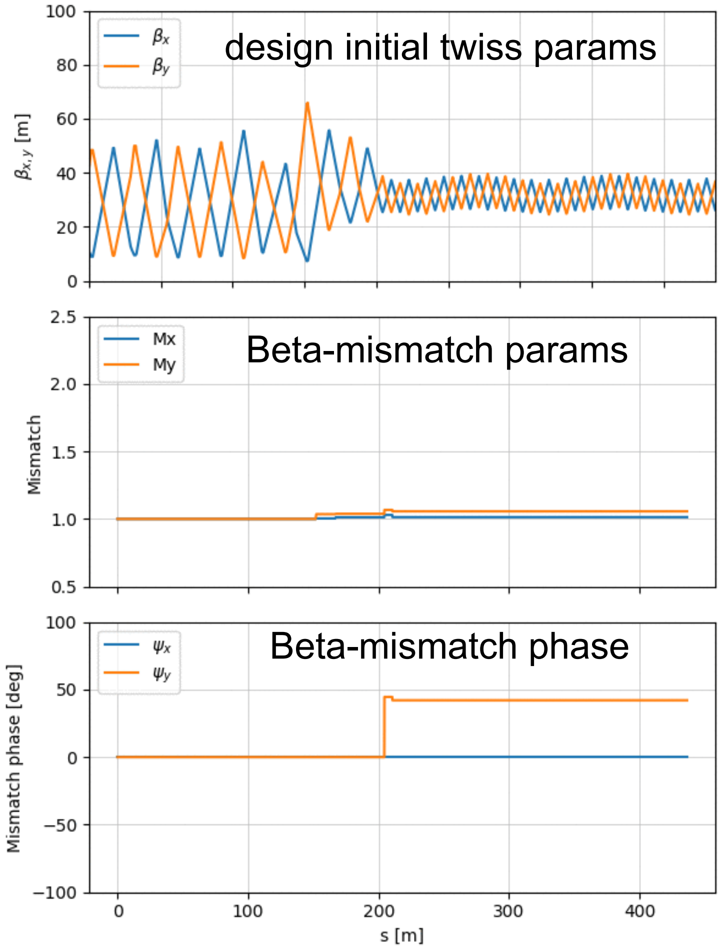
- Quads tuning in front of SASE1 is one of the steps of standard optimization procedure with Optimizer.
- Hyperparameters, such as initial steps, number of iterations, were corrected as experience increased.

Hyperparameters. Beam matching



SASE optimization with 4 quads

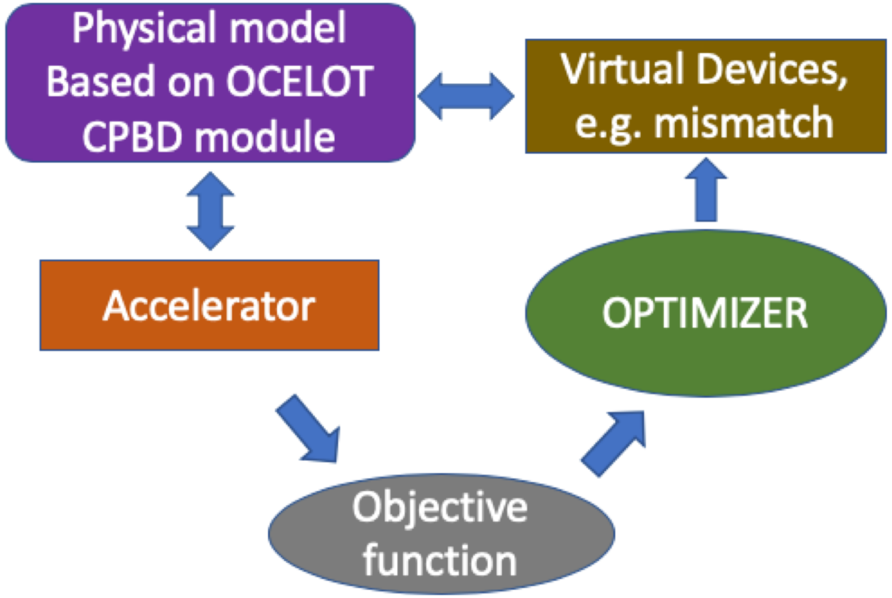




- Quads tuning in front of SASE1 is one of the steps of standard optimization procedure with Optimizer.
- Hyperparameters, such as initial steps, number of iterations, were corrected as experience increased.
- Tweaking beta-mismatch and beta-mismatch phase instead of quadrupoles directly.

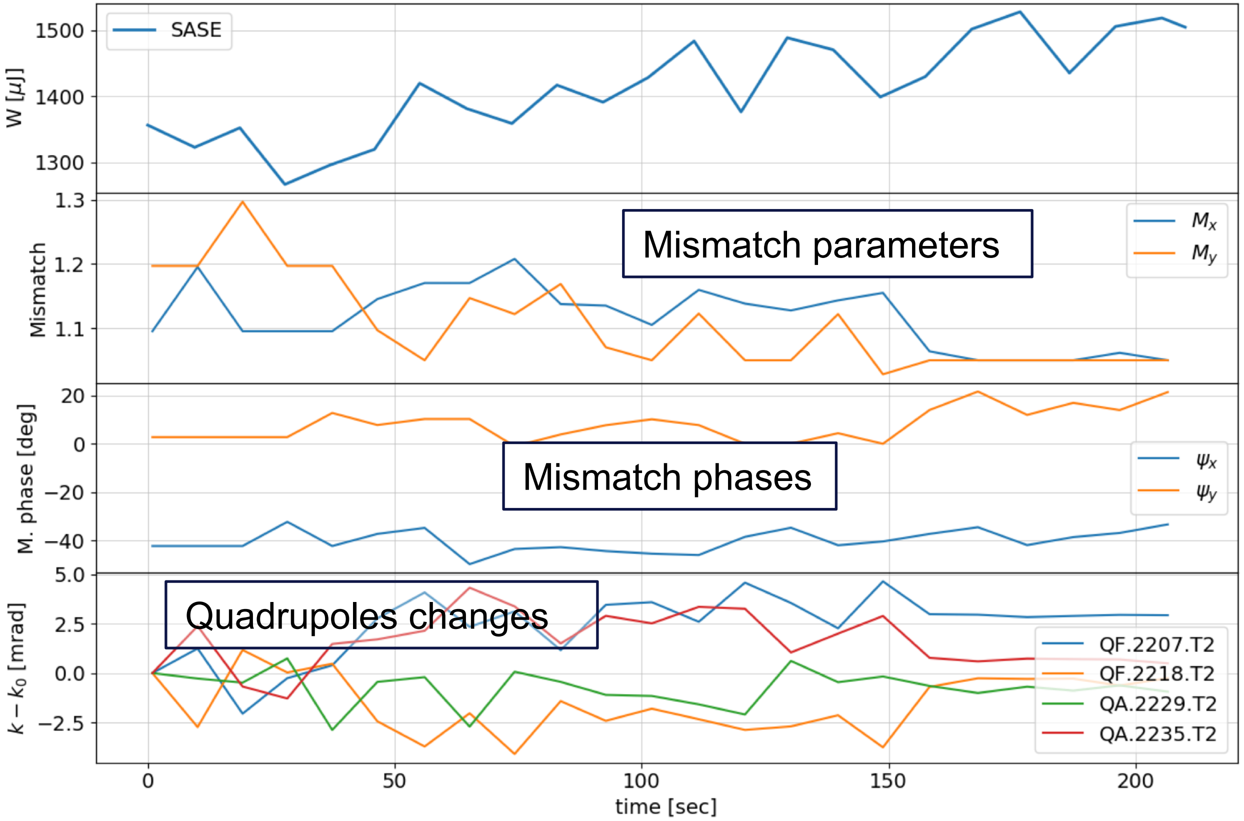
CPBD ⟷ OBC

Beta mismatch parameter* optimization



- 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

- SASE1 pulse energy opt. 25 iterations.



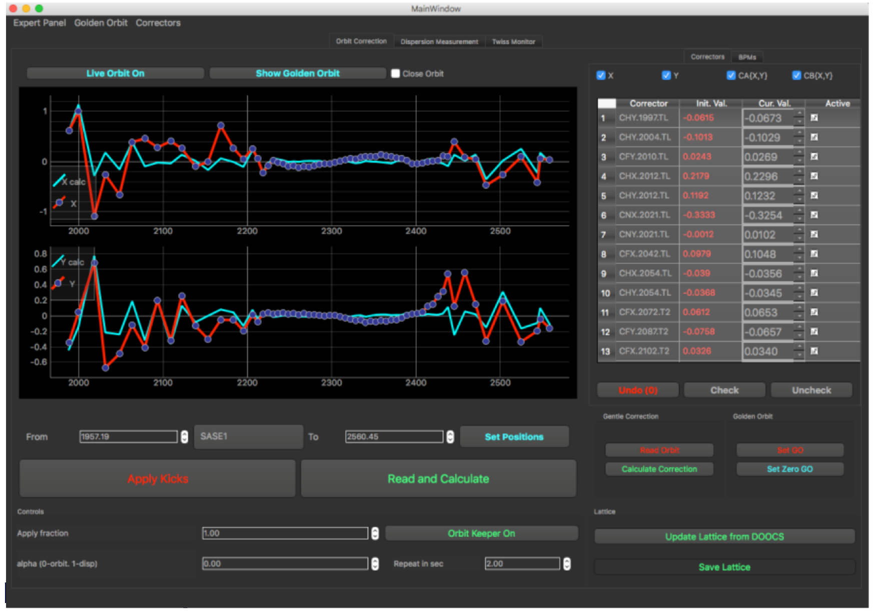
* M. Sands, SLAC-AP-85 1991

CPBD ↔ OBC

Orbit correction tool with adaptive feedback.

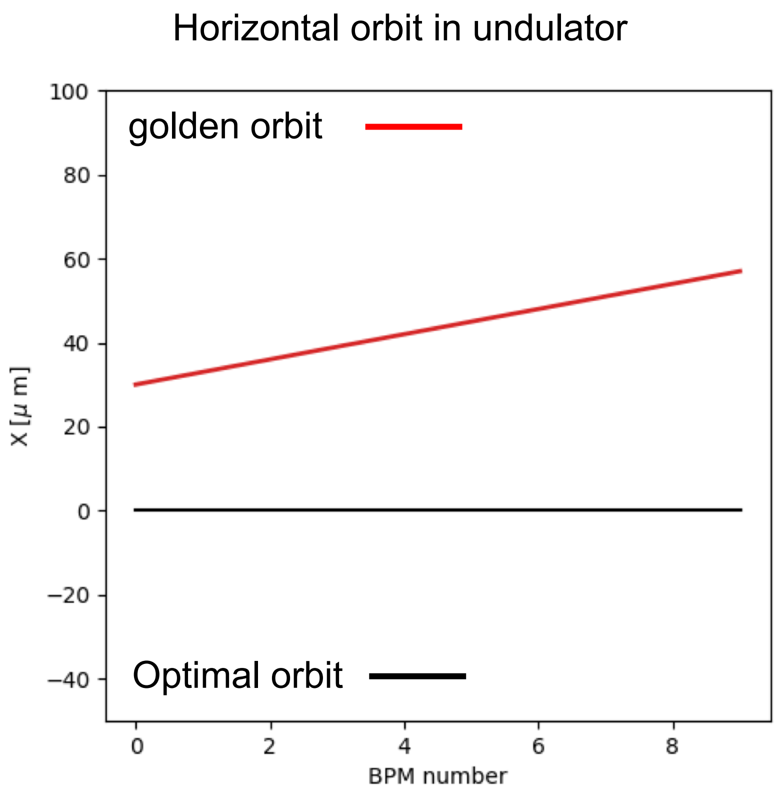
- OCELOT orbit correction is the standard tool for orbit correction (using SVD algorithm).
- The Adaptive Feedback is a statistical optimizer exploiting the orbit jitter and its correlation with a fast FEL intensity signal (shot-to-shot resolution) to optimize the undulator launch orbit

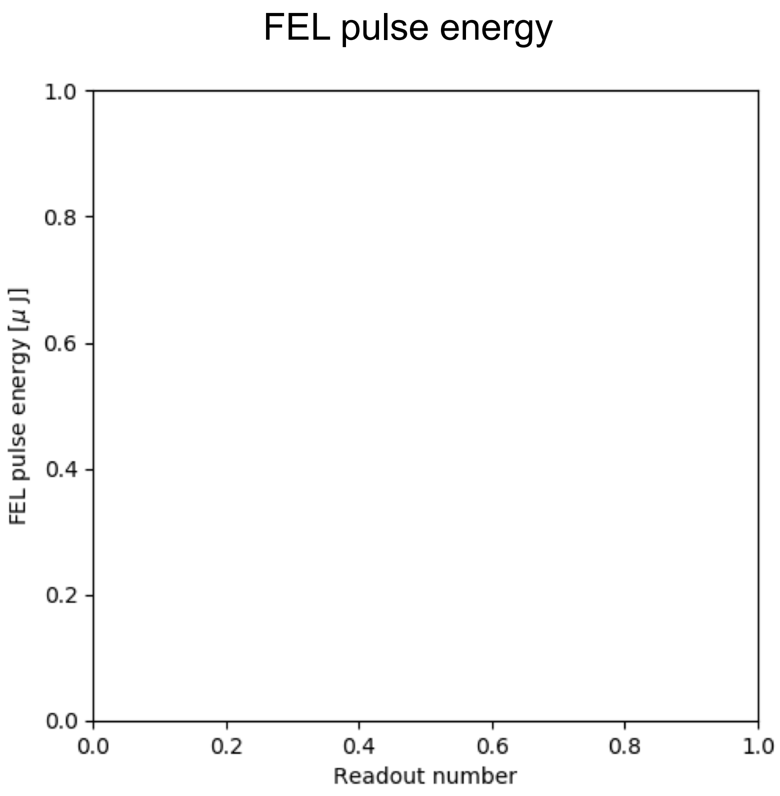
Orbit correction tool GUI



Adaptive Feedback GUI

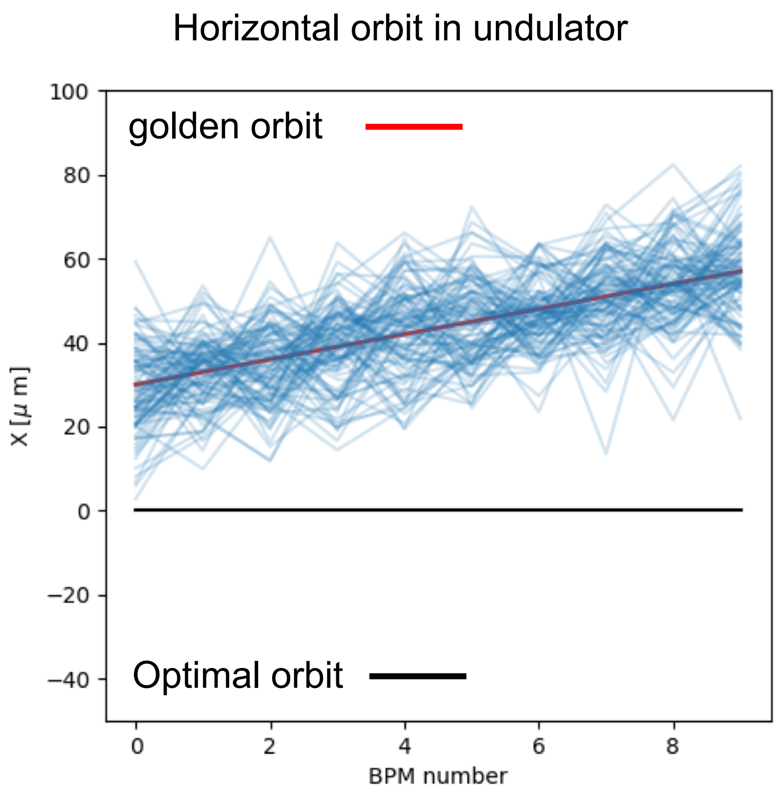
Adaptive Feedback: how it works

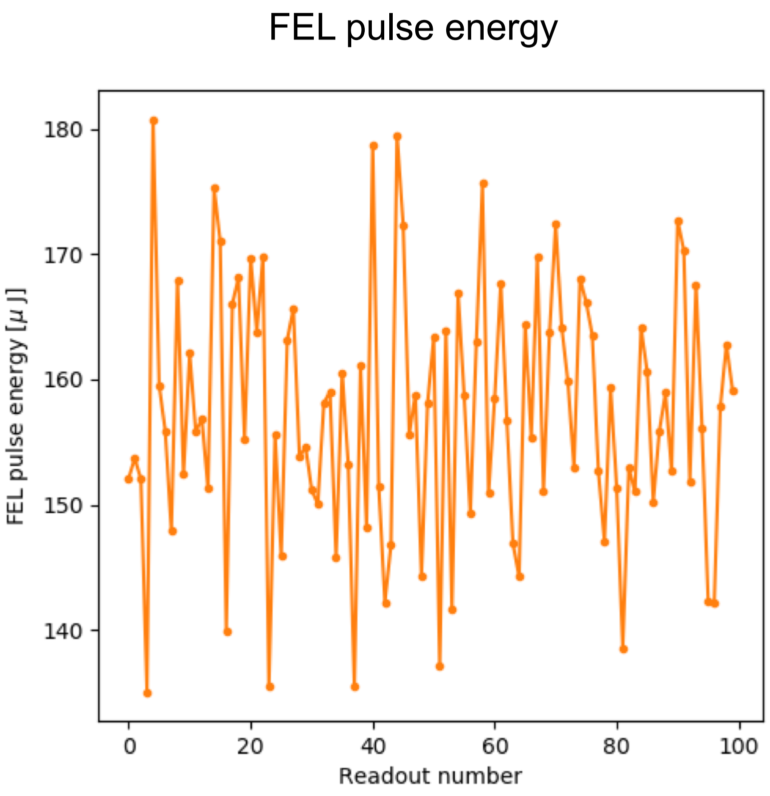




Iteration #1

Adaptive Feedback: how it works

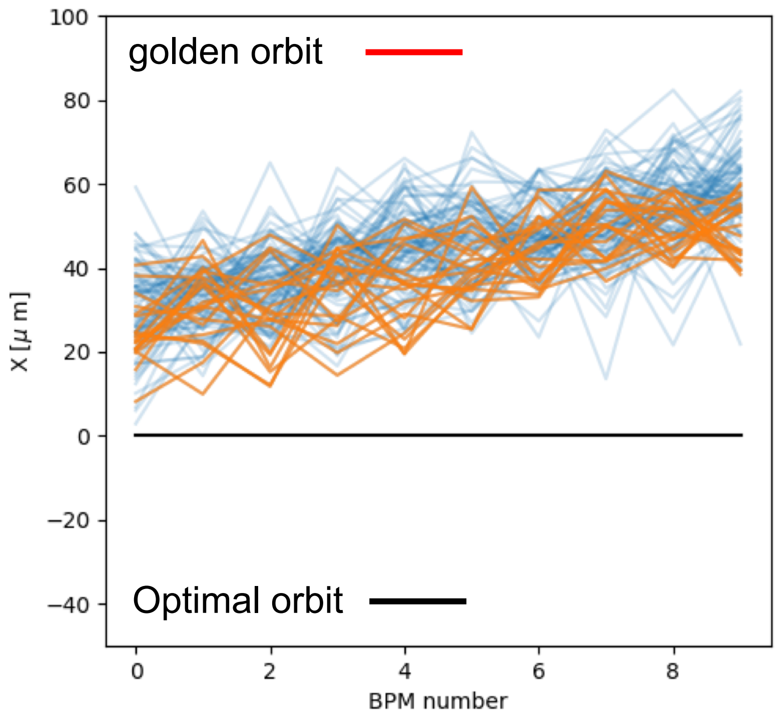




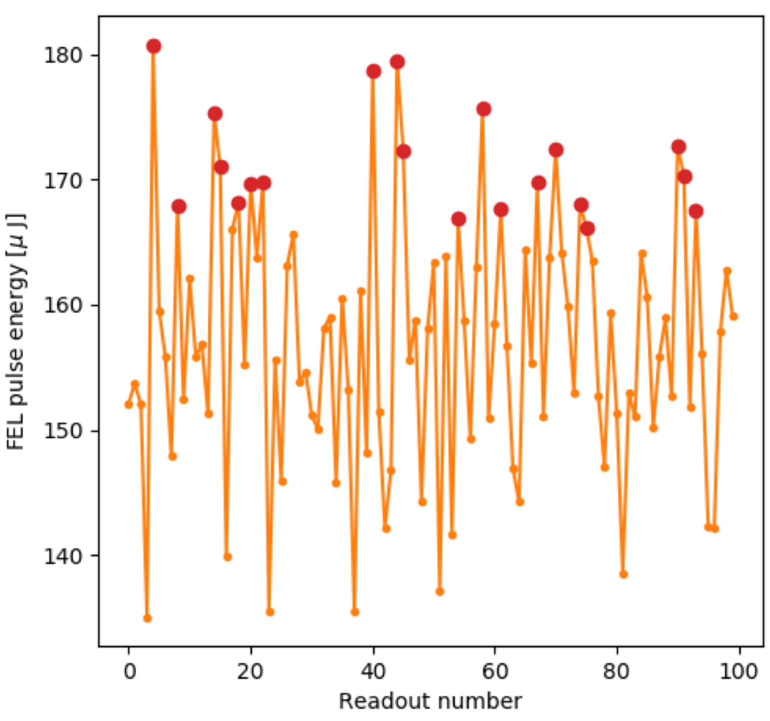
Iteration #1

Adaptive Feedback: how it works

Horizontal orbit in undulator



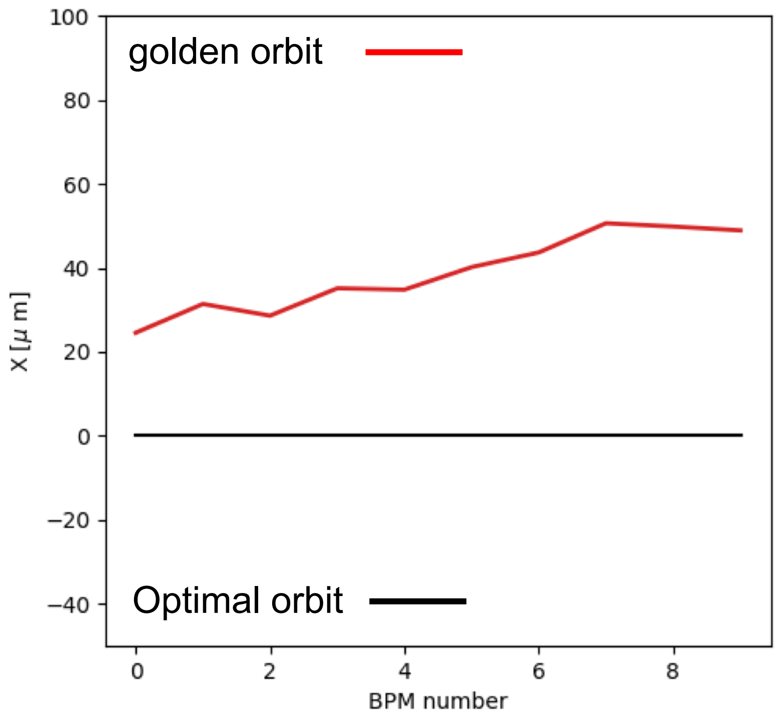
FEL pulse energy



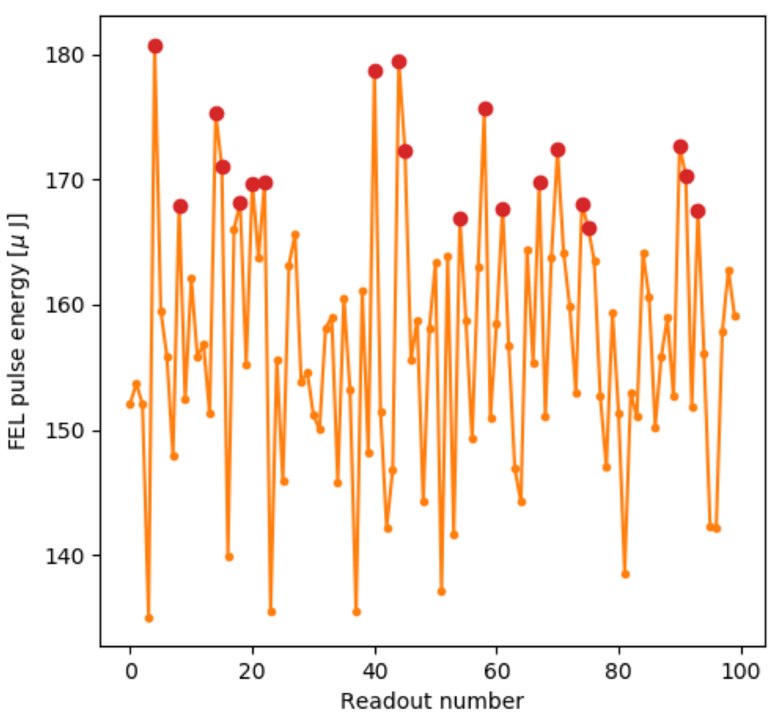
Iteration #1

Adaptive Feedback: how it works

Horizontal orbit in undulator



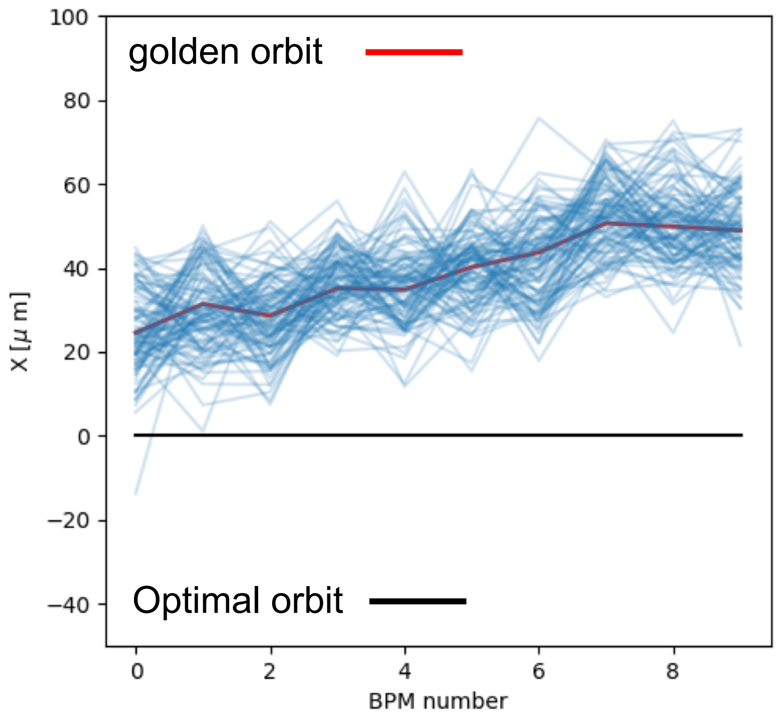
FEL pulse energy



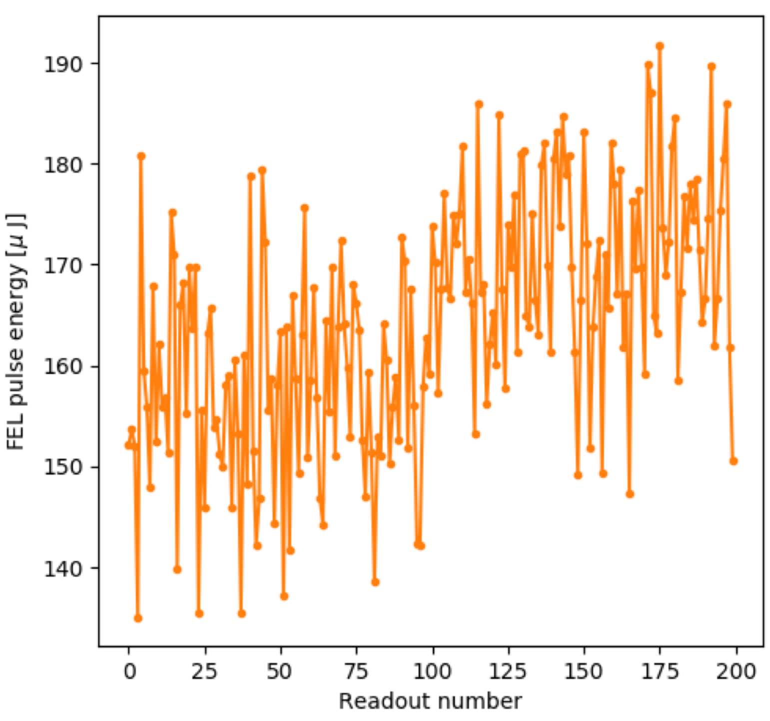
Iteration #1

Adaptive Feedback: how it works

Horizontal orbit in undulator



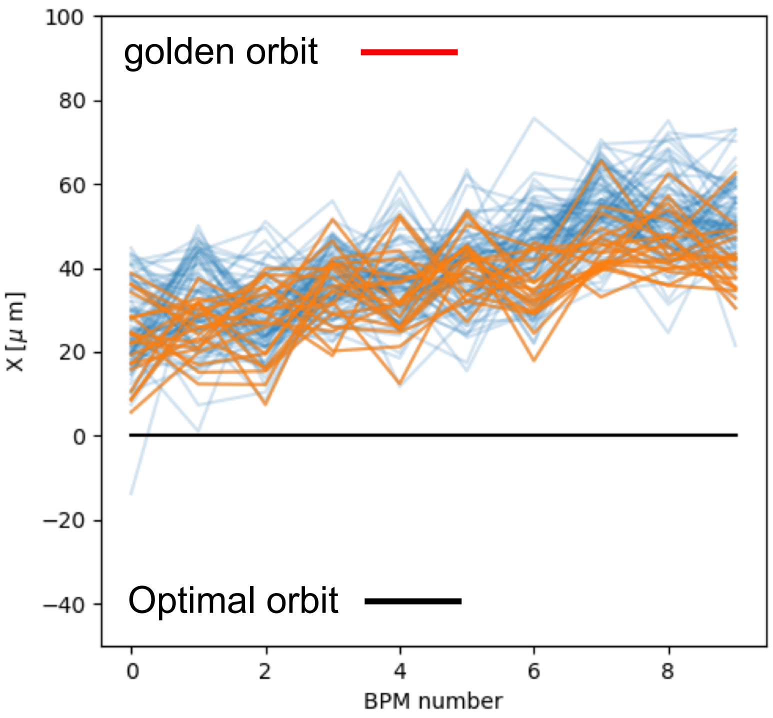
FEL pulse energy



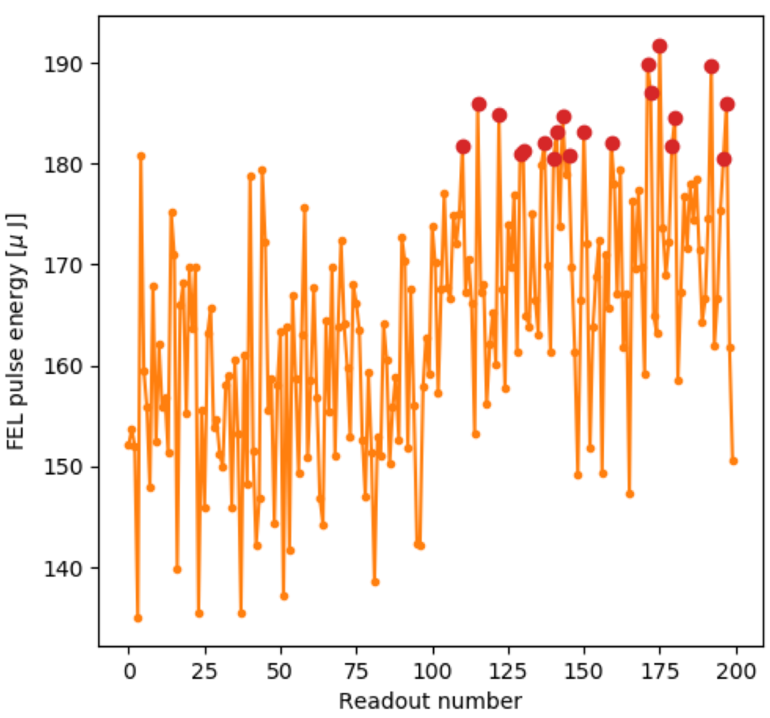
Iteration #2

Adaptive Feedback: how it works

Horizontal orbit in undulator

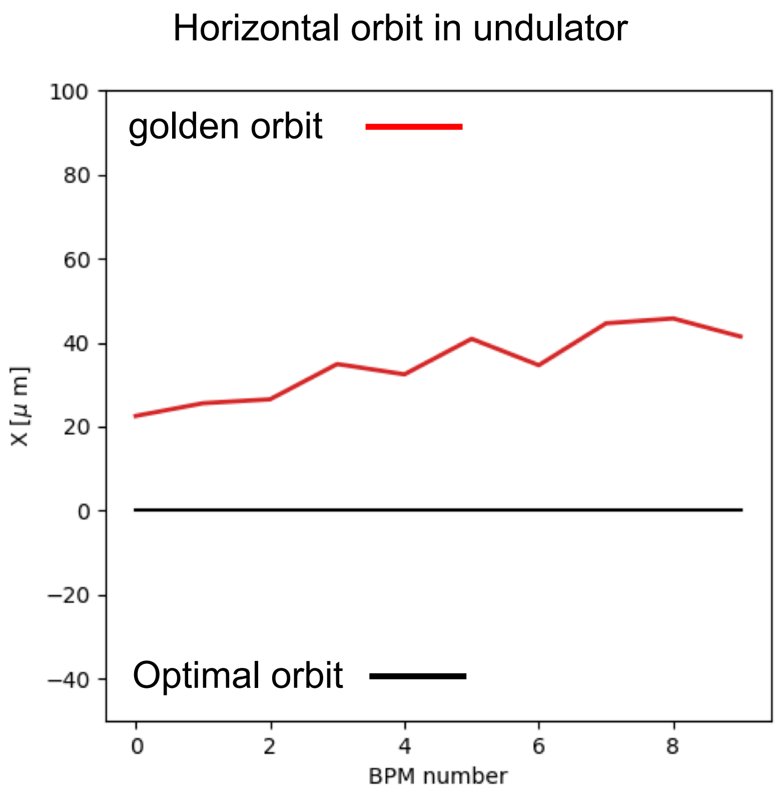


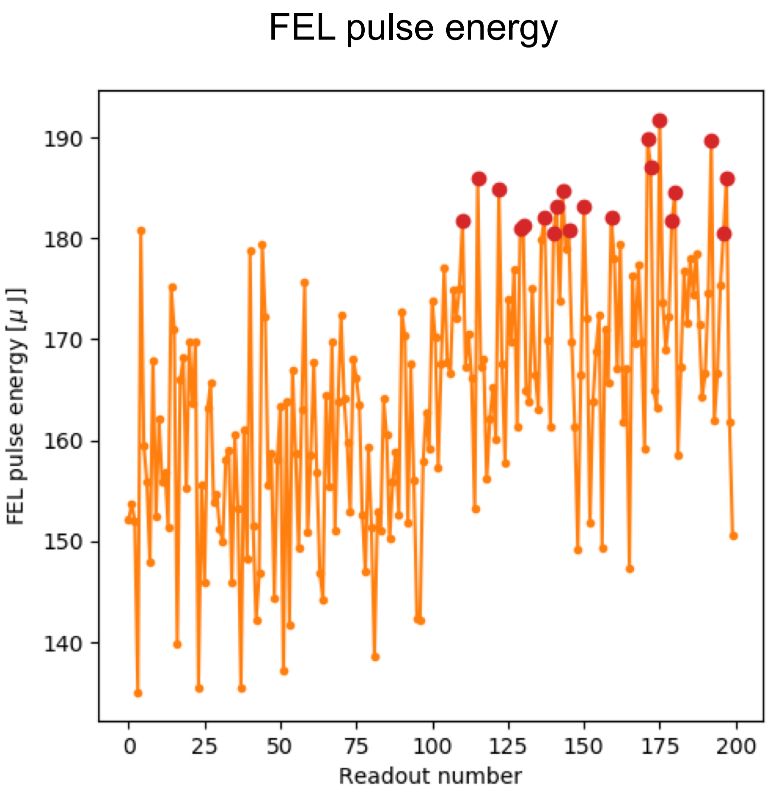
FEL pulse energy



Iteration #2

Adaptive Feedback: how it works

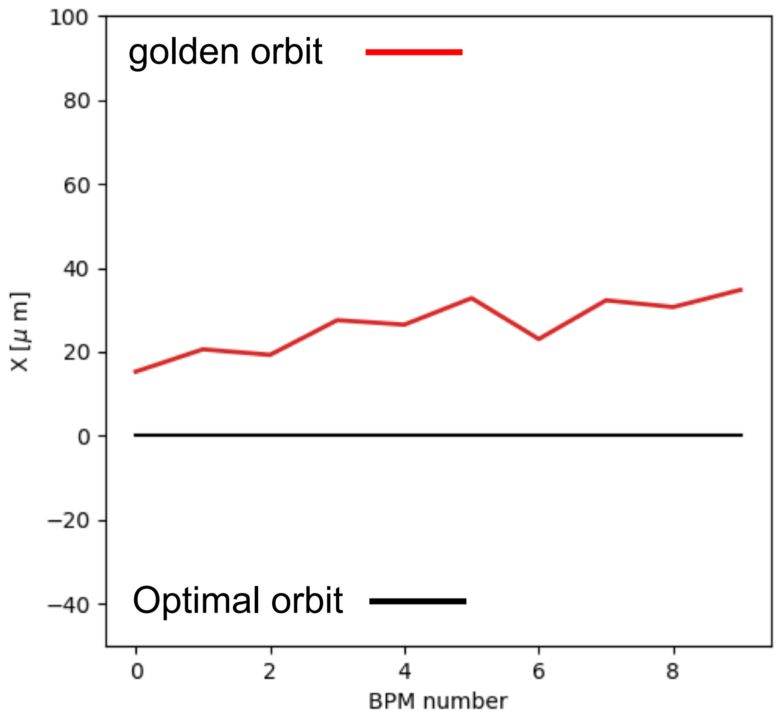




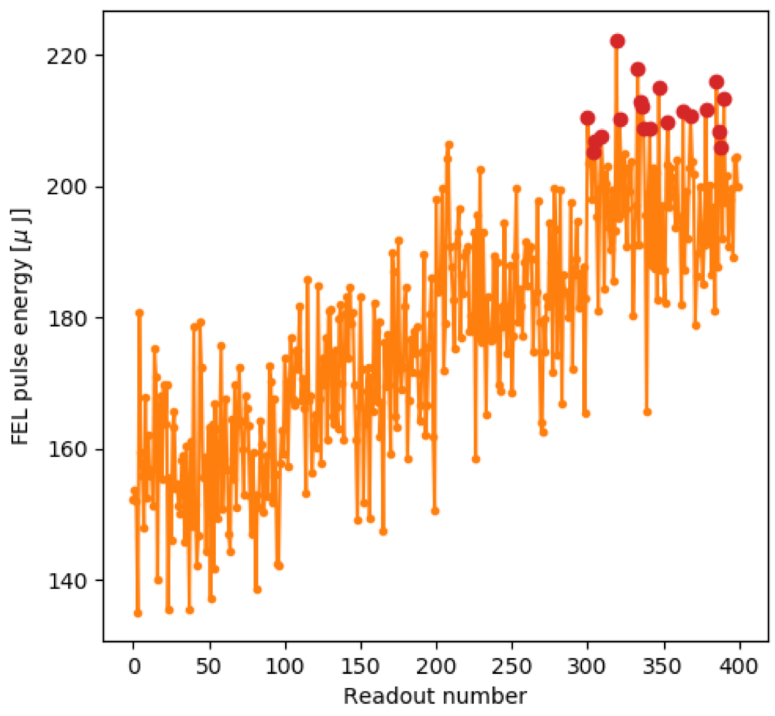
Iteration #2

Adaptive Feedback: how it works

Horizontal orbit in undulator

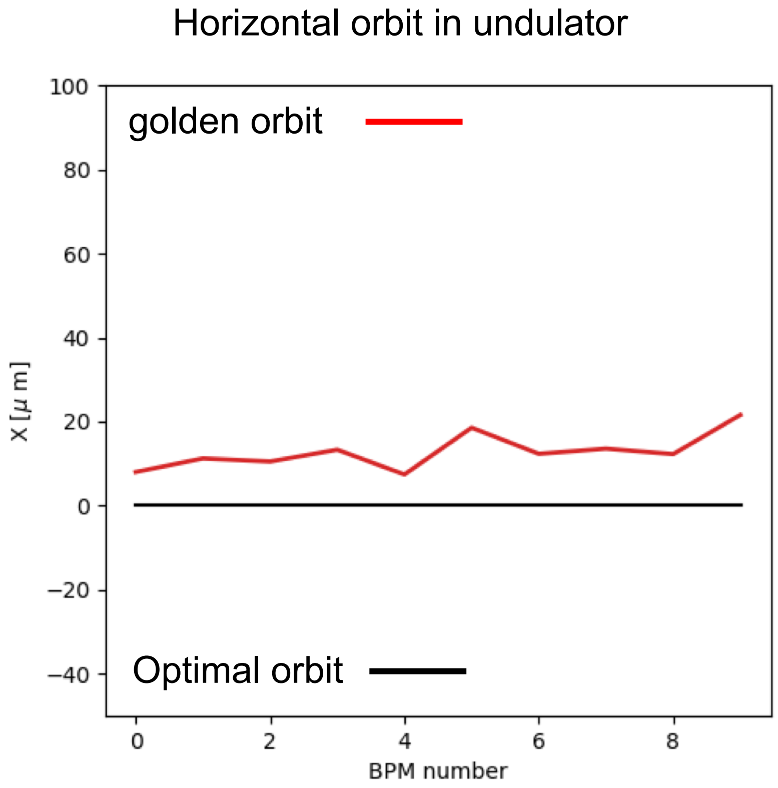


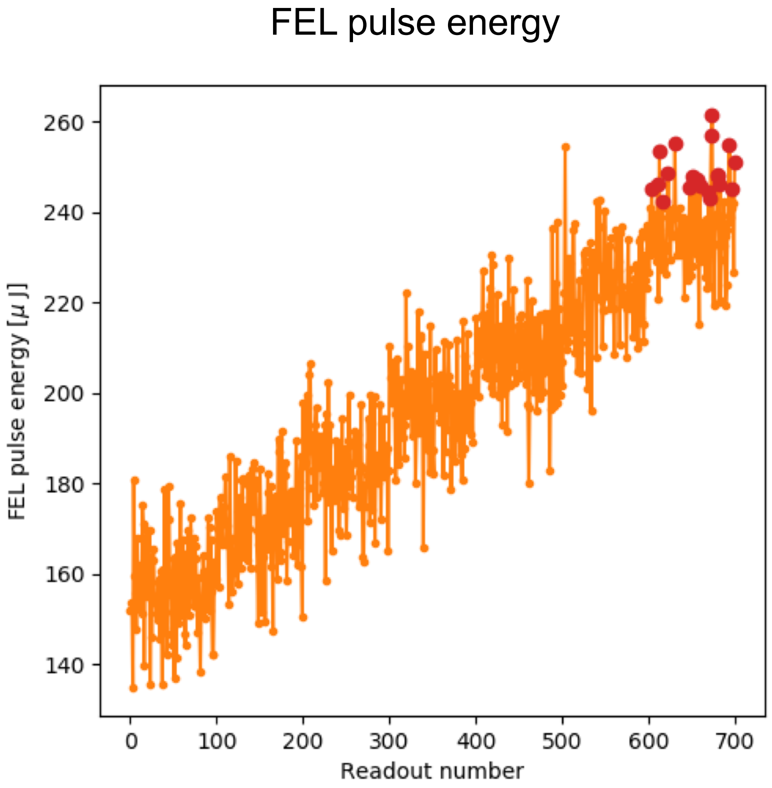
FEL pulse energy



Iteration #4

Adaptive Feedback: how it works

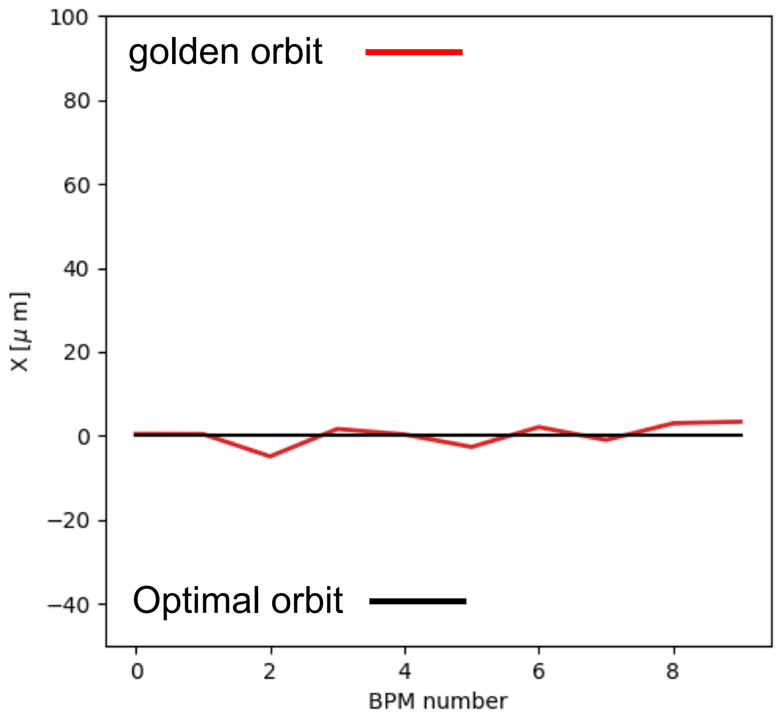




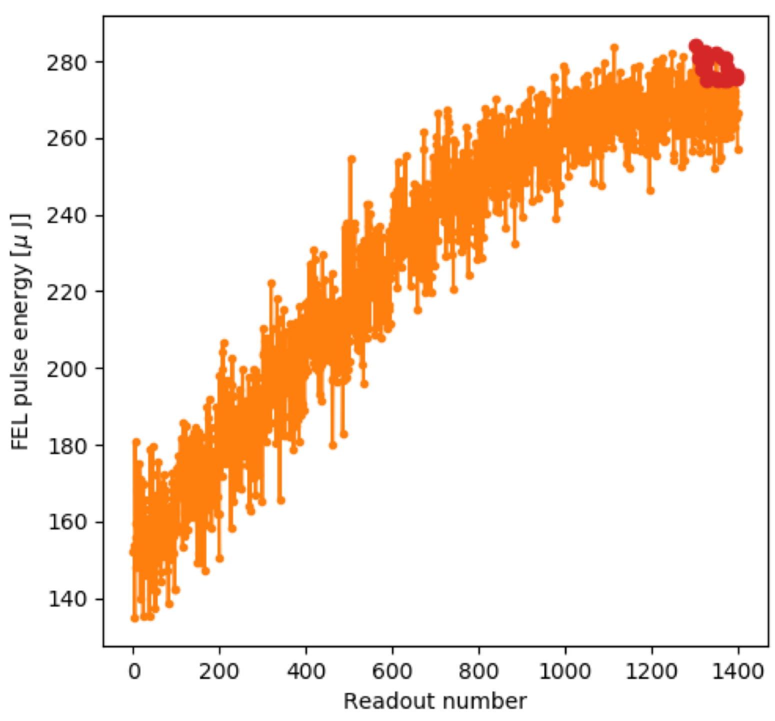
Iteration #7

Adaptive Feedback: how it works

Horizontal orbit in undulator



FEL pulse energy

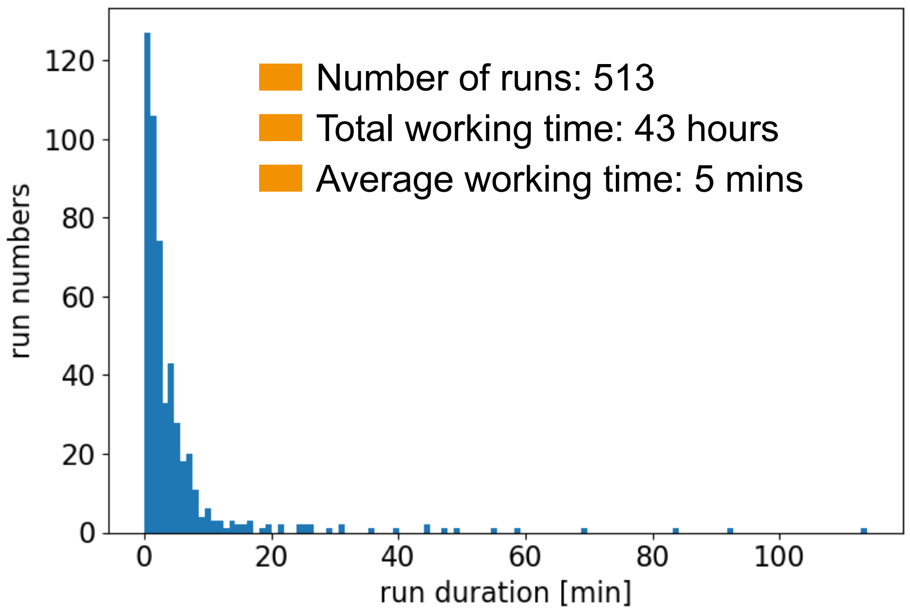


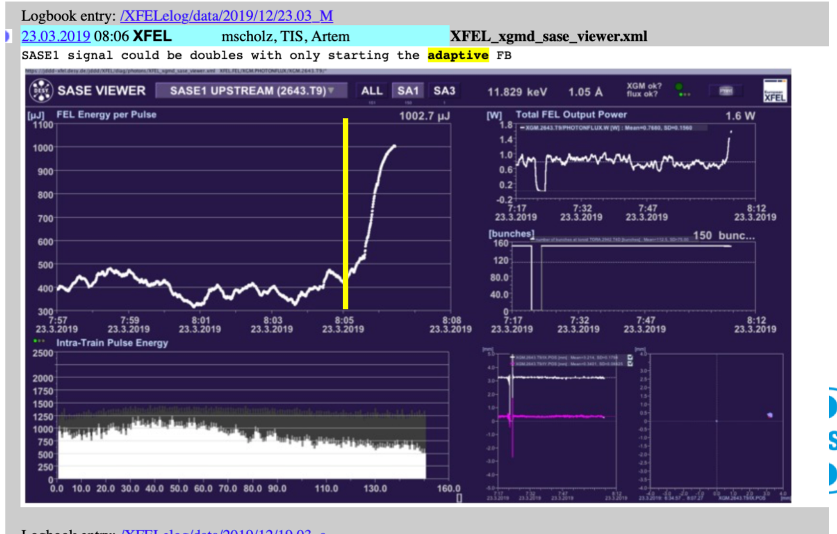
Iteration #14

Adaptive Feedback statistics

- Adaptive Feedback has become one of the main tools for SASE tuning
- In some cases the adaptive feedback is used as an orbit feedback
- The soft X-Ray FEL pulse energy signal is not sensitive to the orbit jitter in the SASE3 undulator.
  - artificially induced orbit changes have to be used to catch correlations
- Deployed to FLASH2
- Active search?

Statistic of the Adaptive Feedback runs from March 11 to April 7, 2019





Conclusion

OCELOT modules and functions
for accelerator modeling and optimizing

+

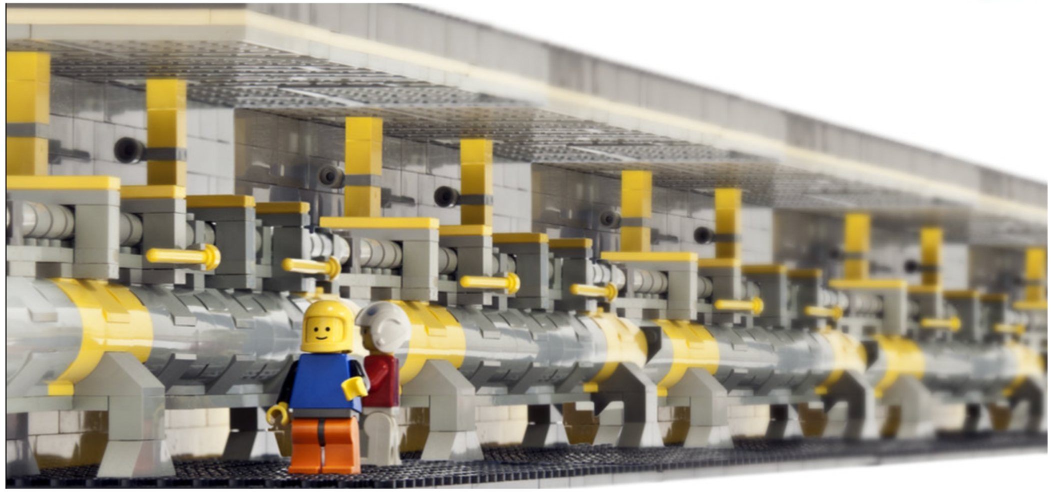
Available Python modules:
Math., data analysis, ML, plotting,
optimization algorithms etc

Conclusion

OCELOT modules and functions
for accelerator modeling and optimizing

+

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



Documentations

- Tutorials
- Description of functions and classes in source files **OR** use *help()* in python/ipython/Jupyter

```
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.

Synchrotron radiation module

- Tutorial N9. Synchrotron radiation module. Web version.
  - Simple examples how to calculate synchrotron radiation with OCELOT.
- Tutorial N10. Simple accelerator based THz source. Web version.
  - A simple accelerator with the electron beam formation system and an undulator to generate THz radiation.

Wavefront propagation

- Tutorial N11. Coherent radiation module and RadiationField object. Web version.
- Tutorial N12. Reflection from imperfect highly polished mirror. Web version.
- Tutorial N13. Converting synchrotron radiation Screen object to RadiationField object for viewing and propagation. Web version.

Unit-tests

```
=============================== test session starts ================================
platform darwin -- Python 3.6.9, pytest-5.0.1, py-1.8.0, pluggy-0.12.0
rootdir: [redacted]/ocelot
collected 202 items

unit_tests/adaptors_test/elegant_lattice/elegant_lattice_test.py ....s            [  2%]
unit_tests/ebeam_test/csr_ex/csr_ex_test.py .............s                        [  9%]
unit_tests/ebeam_test/dba/dba_test.py ....s                                       [ 12%]
unit_tests/ebeam_test/dba_track_ellipse/dba_track_ellipse_test.py ...s            [ 14%]
unit_tests/ebeam_test/dba_tracking/dba_tracking_test.py .....s                    [ 17%]
unit_tests/ebeam_test/dogleg/dogleg_test.py .......s                              [ 21%]
unit_tests/ebeam_test/io/io_test.py ...s                                          [ 23%]
unit_tests/ebeam_test/io_lattice/io_lattice_test.py ....s                         [ 25%]
unit_tests/ebeam_test/laser_heater/laser_heater_test.py .........s                [ 30%]
unit_tests/ebeam_test/linac_orb_correct/linac_orb_correct_test.py .......s        [ 34%]
unit_tests/ebeam_test/matching/match_test.py ......s                              [ 38%]
unit_tests/ebeam_test/multipoles/multipoles_test.py .....s                        [ 41%]
unit_tests/ebeam_test/navi/navi_test.py ...........s                              [ 47%]
unit_tests/ebeam_test/non_disp_offset/non_disp_offset_test.py ......s             [ 50%]
unit_tests/ebeam_test/phys_proc/phys_proc_test.py .........s                      [ 55%]
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            [ 60%]
unit_tests/ebeam_test/rk_track/rk_track_test.py ....s                             [ 62%]
unit_tests/ebeam_test/space_charge/space_charge_test.py .......s                  [ 66%]
unit_tests/ebeam_test/storage_ring/storage_ring_test.py ......s                   [ 70%]
unit_tests/ebeam_test/storage_ring_da/storage_ring_da_test.py .......s            [ 74%]
unit_tests/ebeam_test/triplet/triplet_test.py ......s                             [ 77%]
unit_tests/ebeam_test/tune_shift/tune_shift_test.py ....s                         [ 80%]
unit_tests/ebeam_test/twiss_sase3/twiss_sase3_test.py ...s                        [ 82%]
unit_tests/ebeam_test/undulator/undulator_test.py ....s                           [ 84%]
unit_tests/sr_test/rad_beam/rad_beam_test.py ..........s                          [ 90%]
unit_tests/sr_test/spatial/spatial_test.py .......s                               [ 94%]
unit_tests/sr_test/spect_mag_file/spect_mag_file_test.py .s                       [ 95%]
unit_tests/sr_test/spectrum/spectrum_test.py ....s                                [ 98%]
unit_tests/wave_test/generate_gaussian_dfl/generate_gaussian_dfl_test.py ...s     [100%]

=============== 172 passed, 30 skipped, 2 warnings in 457.04 seconds ===============
(base) exflqr29269:ocelot tomins$
```

- 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 and collaborators (the order is random ☺):

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

I. Agapov, M. Dohlus, W. Kuropka, M, Scholz, I. Zagorodnov – *DESY*

J. Duris, D. Ratner, H. Slepicka, J. Shtalenkova - *SLAC*

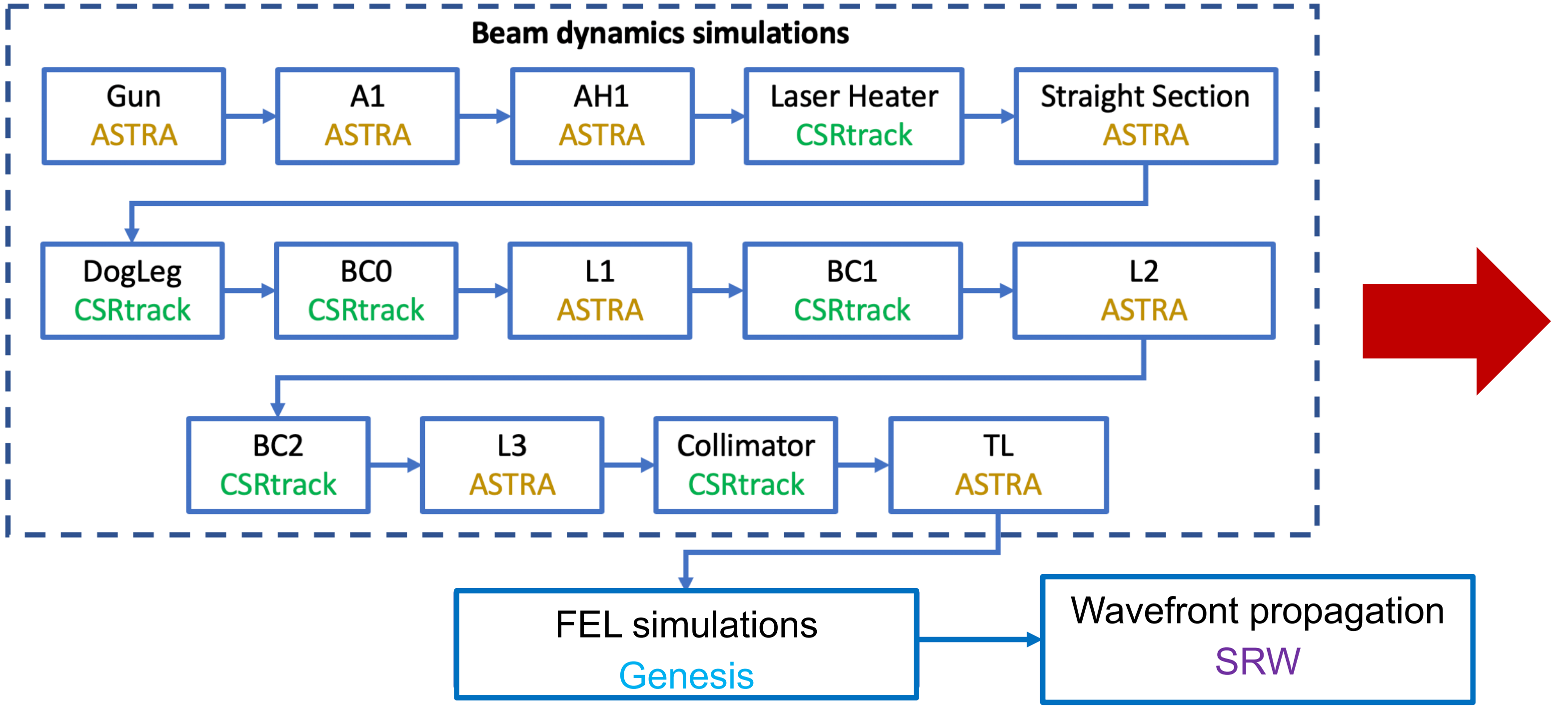
E. Fomin – *Kurchatov Institute*

A. Trebushinin – *BINP (Novosibirsk)*

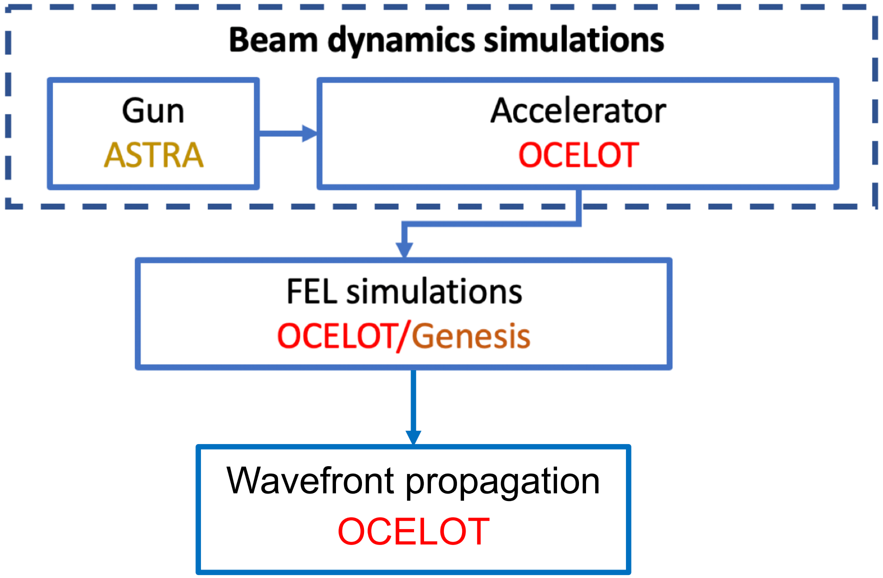
M. Veremchuk – *Kyiv National University*

Backup slides

Modeling of accelerators with OCELOT

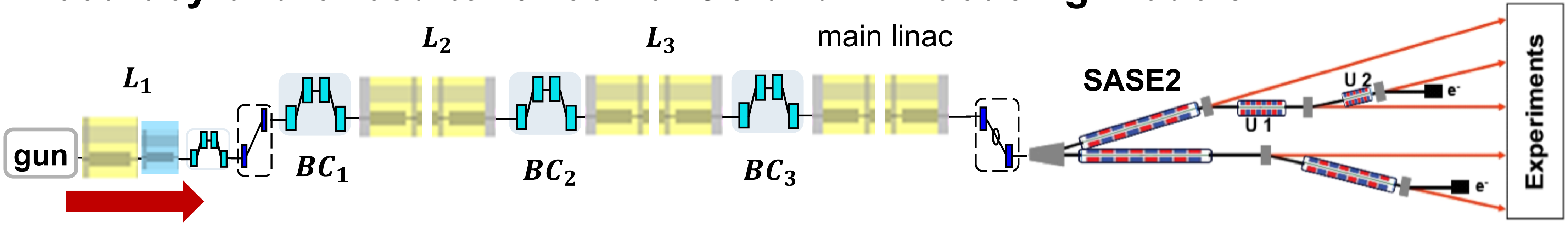


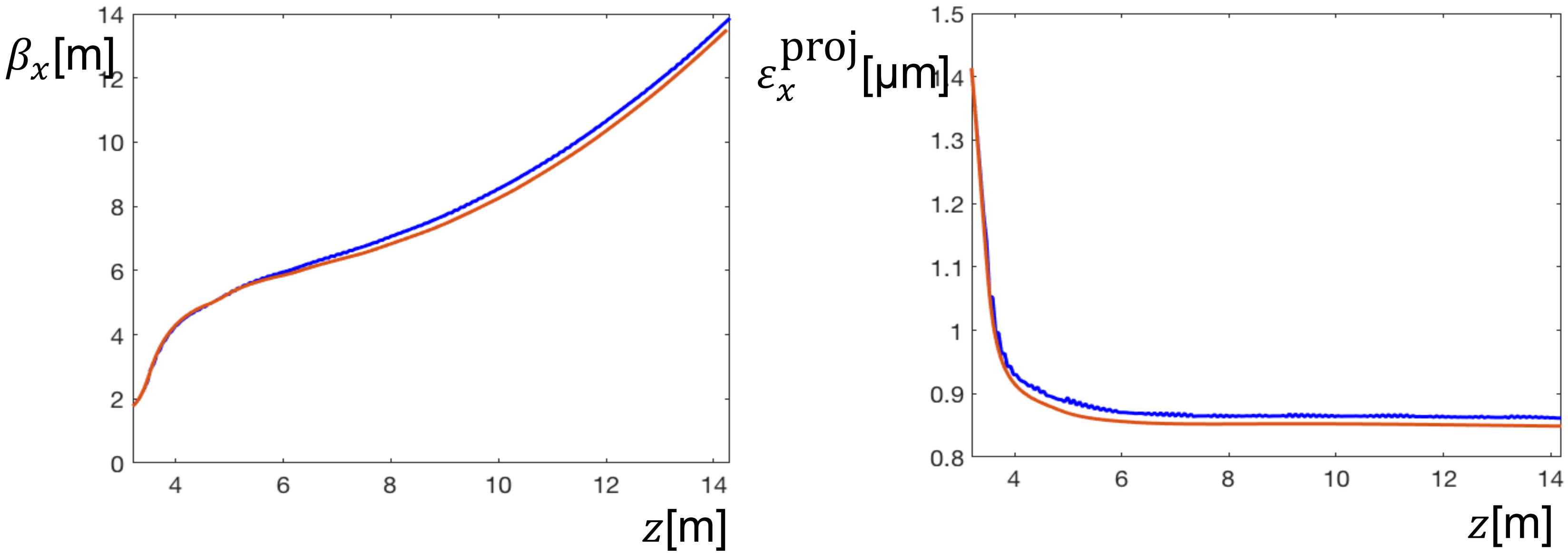
2014



2018

Accuracy of the results. Check of SC and RF focusing models

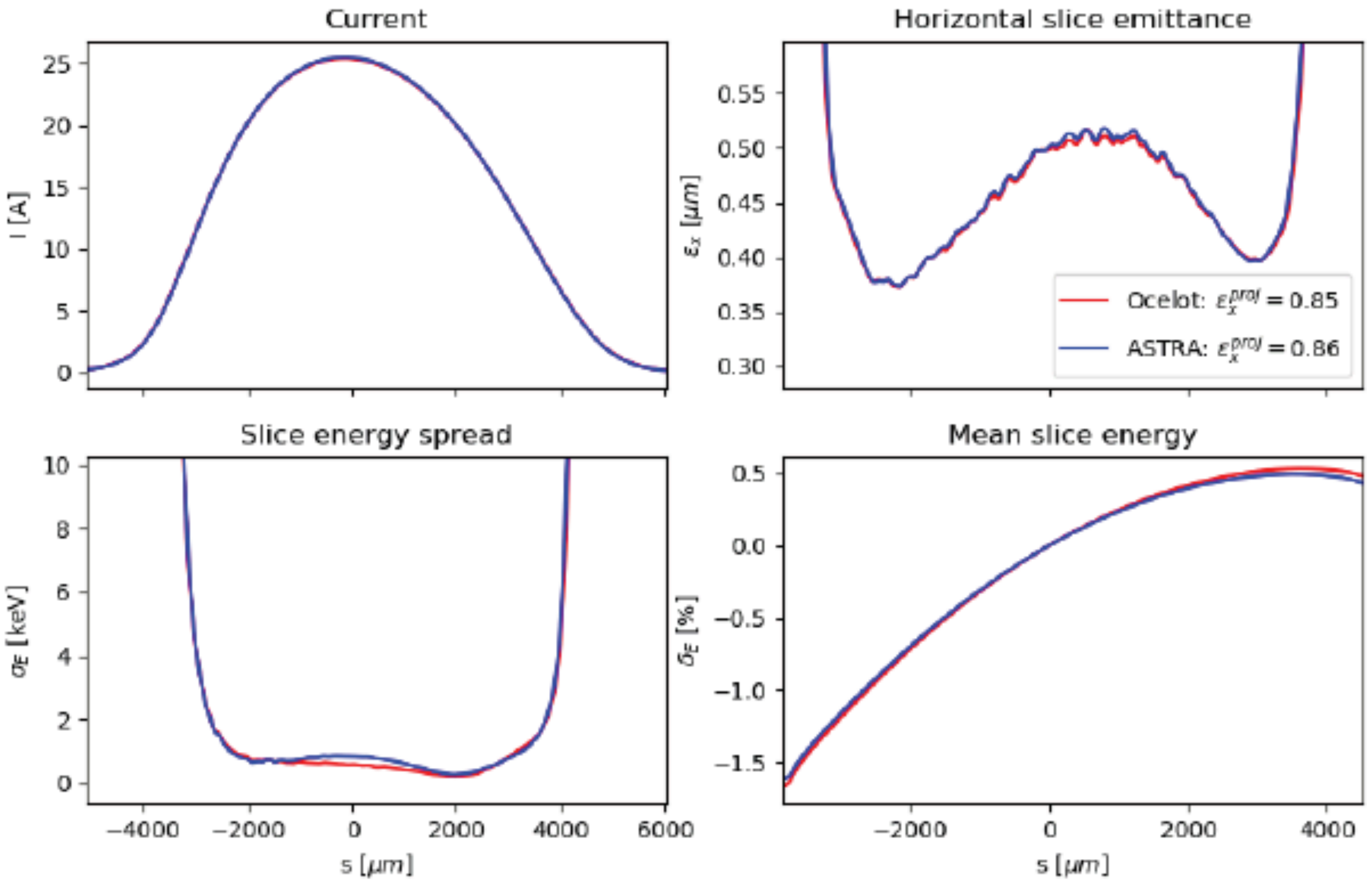




- Comparison of $\boldsymbol{\beta}$-function and projected emittance in the booster calculated by ASTRA (in blue) and Ocelot (in red)

I. Zagorodnov, M. Dohlus, and S. Tomin, Phys. Rev. Accel. Beams 22, 024401, 2019.

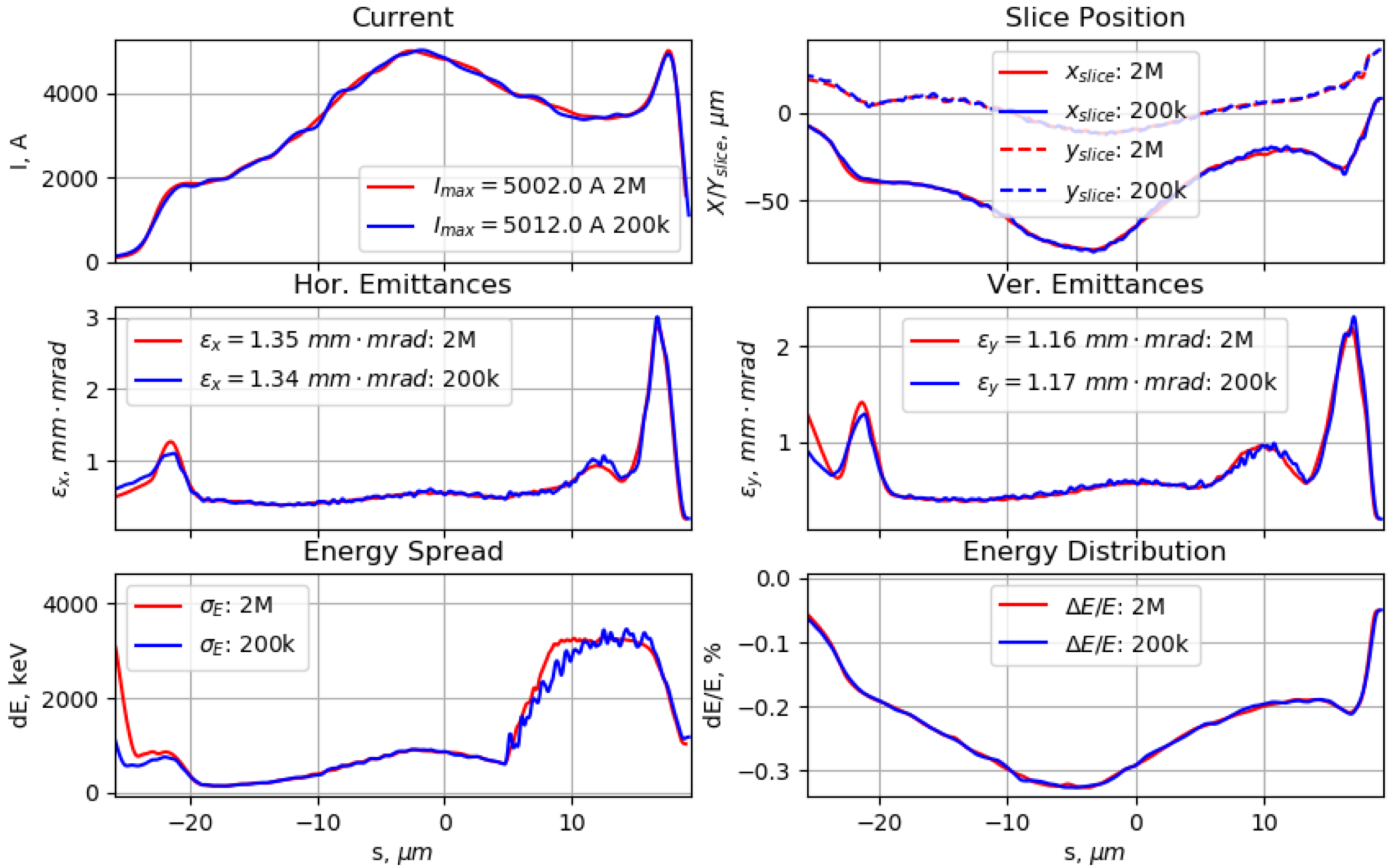
Accuracy of the results. Check of SC and RF focusing models



- Comparison of the slice parameters after the booster calculated by ASTRA (in blue) and Ocelot (in red)

I. Zagorodnov, M. Dohlus, and S. Tomin, Phys. Rev. Accel. Beams 22, 024401, 2019.

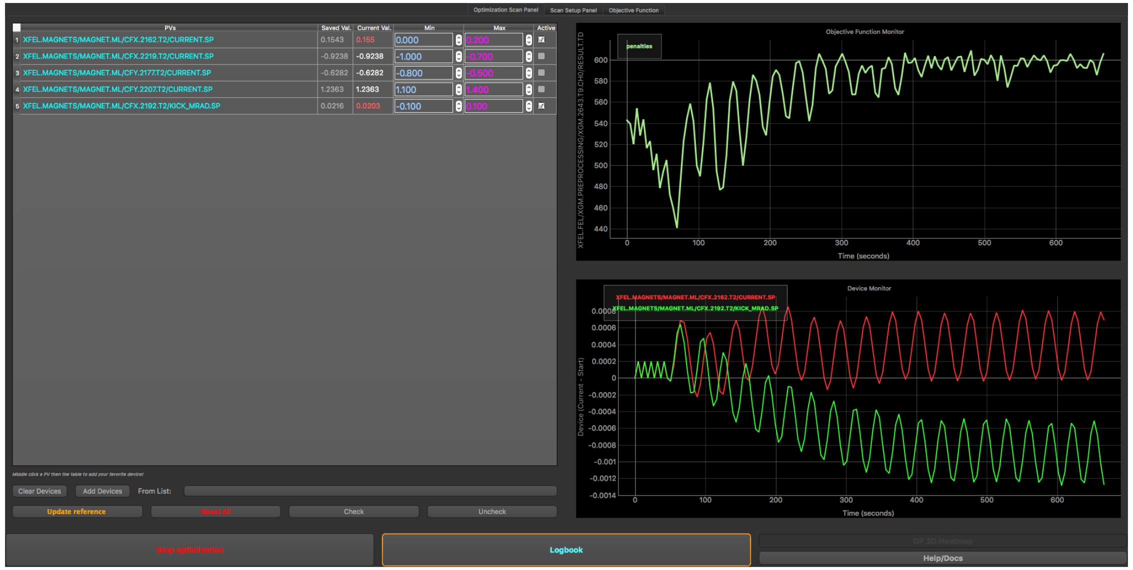
Accuracy of the results. Increase of number of macroparticles



- increase of the macroparticle number by factor 10: from 200k to 2M
- the slice and projected parameters do not change

I. Zagorodnov, M. Dohlus, and S. Tomin, Phys. Rev. Accel. Beams 22, 024401, 2019.

Extremum seeking in Optimizer



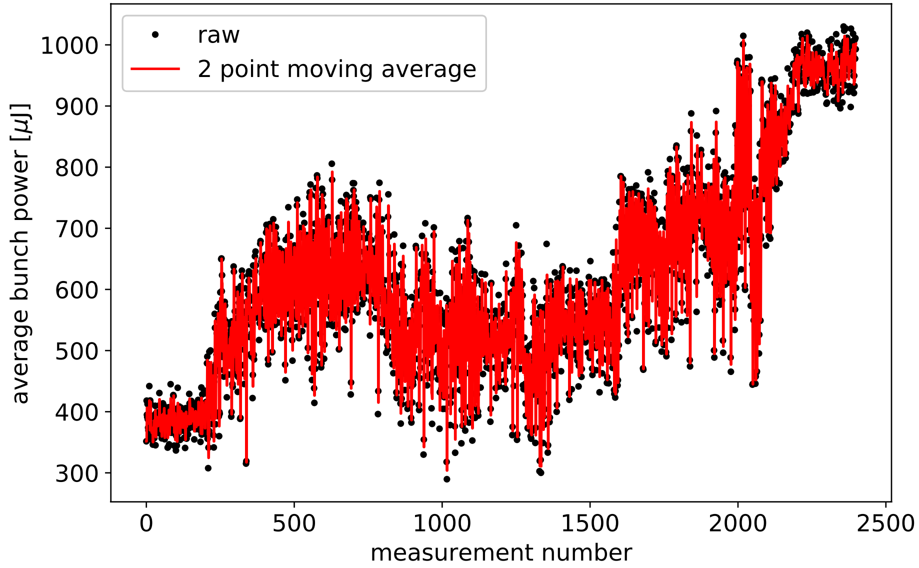
■ Launch orbit optimization

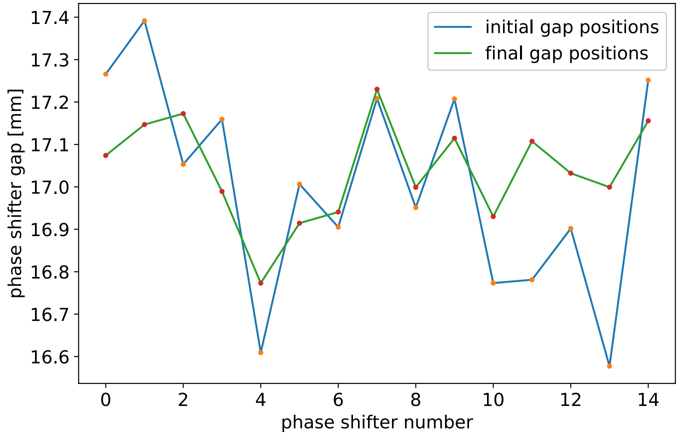
PHYSICAL REVIEW ACCELERATORS AND BEAMS **22,** 082802 (2019)

Model-independent tuning for maximizing free electron laser pulse energy

Alexander Scheinker,[1,*] Dorian Bohler,[2] Sergey Tomin,[3] Raimund Kammering,[4] Igor Zagorodnov,[4] Holger Schlarb,[4] Matthias Scholz,[4] Bolko Beutner,[4] and Winfried Decking[4]

■ 16 phase-shifters optimization





Bayesian optimization method in OCELOT

Joseph Duris et al, Bayesian optimization for FEL tuning, 2[nd] ICFA Workshop on Machine Learning at PSI