Web Interfaces for Accelerator Control

Jonathan Edelen

(On behalf of the RadiaSoft scientific and software staff and our collaborators at SLAC, ORNL, FNAL, LBNL, and BNL)

19 June 2024

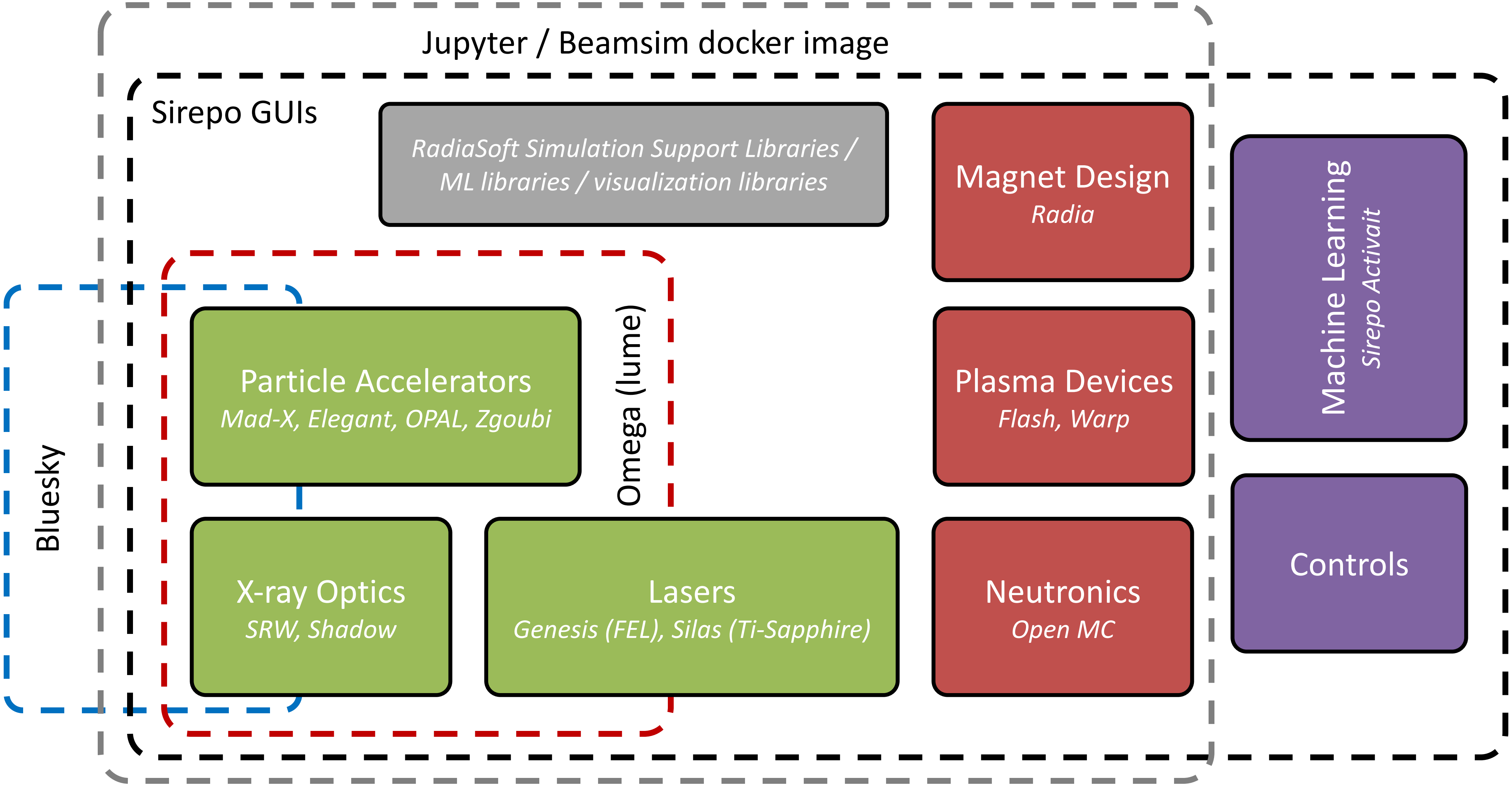
Accelerator Middle Layer Workshop: Hamburg Germany

Sirepo development has been supported by the U.S. Department of Energy Office of Science under multiple awards: by the office of High Energy Physics under Award Nos. DE-SC0011340, DE-SC0015897, DE-SC0018719, DE-SC0020931, DE-SC0022799, DE-SC0024244, and DE-SC0024243; by the office of Basic Energy Sciences under Award Nos. DE-SC0011237, DE-SC0015209, DE-SC0018556, DE-SC0018571, DE-SC0020593, and DE-SC0021551; by the office of Nuclear Physics under Award Nos. DE-SC0015212, DE-SC0017181, and DE-SC0019682; and by the Office of Advanced Scientific Computing Research under Award Nos. DE-SC0017162, DE-SC0017057, DE-SC0021552, and DE-SC0022386. Sirepo development has also been supported by the US Air Force AFWERX program, under Contract No. FA864922P0669. Sirepo development has also been supported by the National Cancer Institute, National Institutes of Health, Department of Health and Human Services, under Contract No. 75N91019C00053. Additional support has been provided by RadiaSoft LLC and from Sirepo customers. Portions of the work presented here are sponsored by research contracts from FNAL, LBNL, and ORNL.

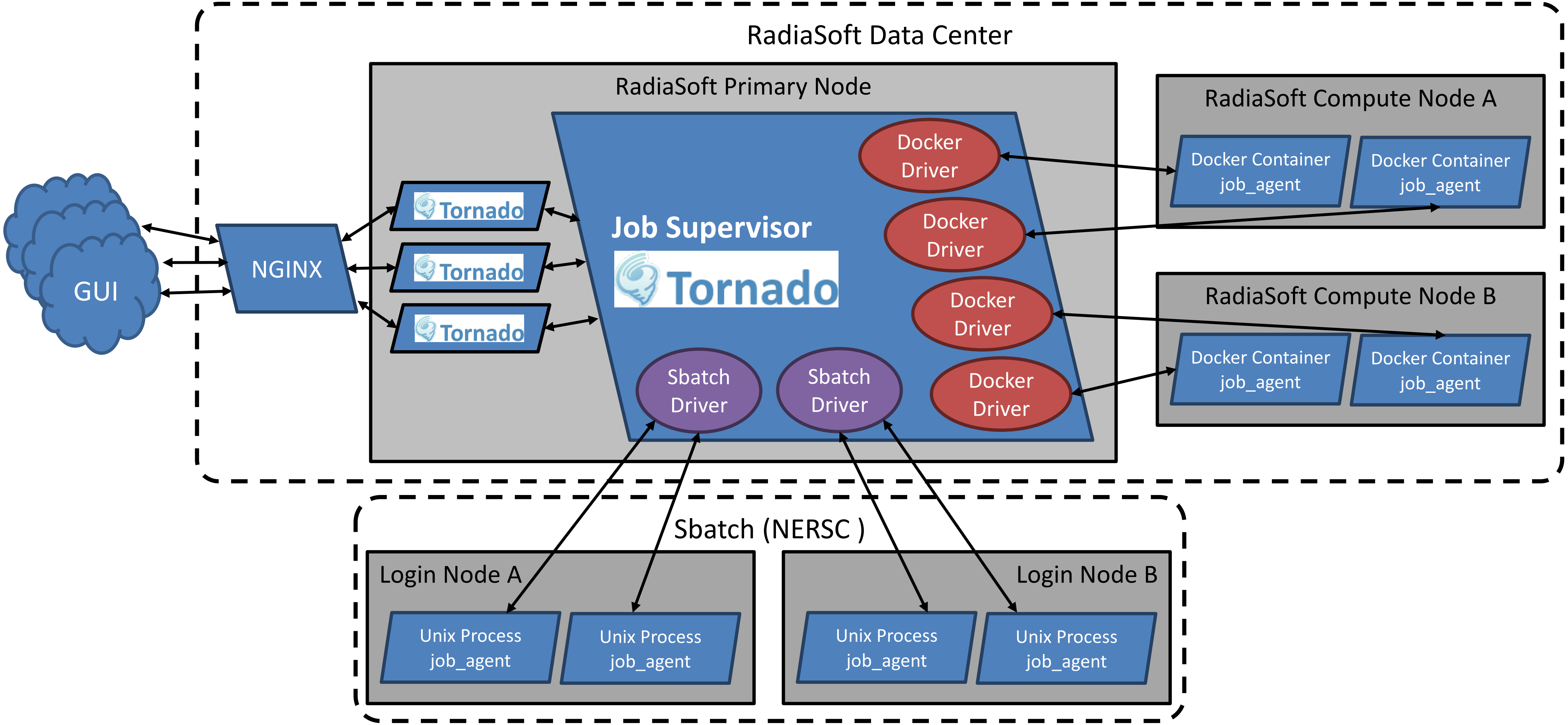
Web Interfaces for Accelerator Control

- Overview of Sirepo
  - Front end and back-end
  - Examples
- Bringing models to the machine
- Virtual experiments with Bluesky
- The Sirepo Controls application

Sirepo: an ecosystem for particle accelerator design and computation



Sirepo Job-System Architecture Overview

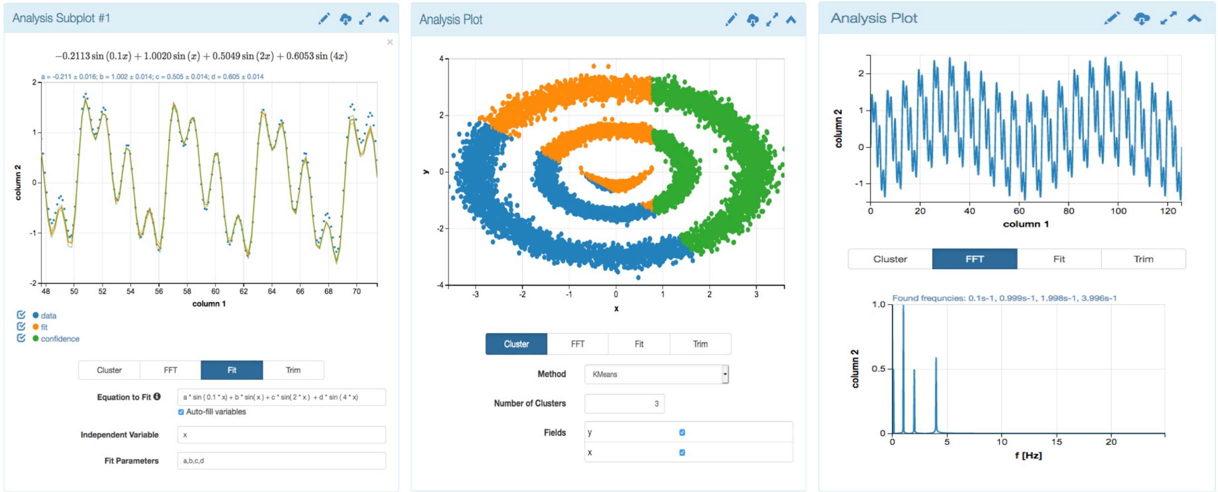


Sirepo provides a machine learning toolbox for accelerator studies

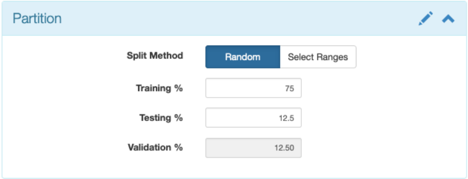
- Sirepo Activait is a web-based toolbox that support generic machine learning and data analysis workflows
  - Initial support for training classification and regression models
  - Export code to Python scripts leveraging TensorFlow and scikit-learn
- In-development workflows include image data, worked examples, and new model types

https://www.sirepo.com/en/apps/machine-learning/

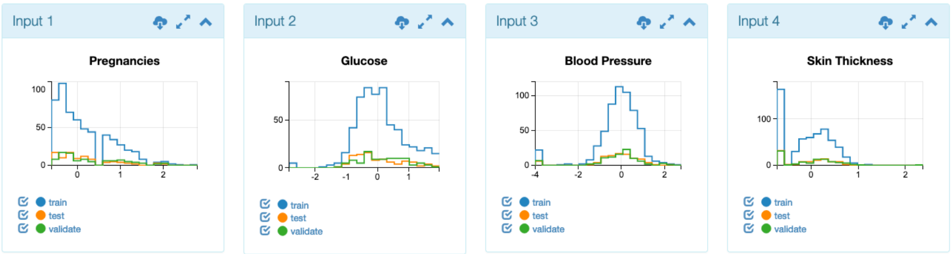
Analysis Toolbox – curve fitting, clustering, frequency analysis



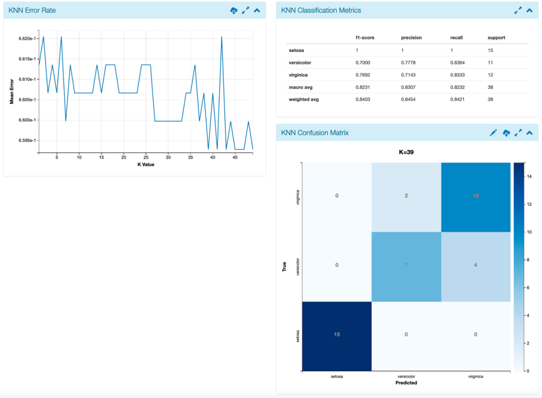
Machine Learning Workflow



Data import and splitting into training and validation and test. Splitting can be random or visual.



Classifier Tools



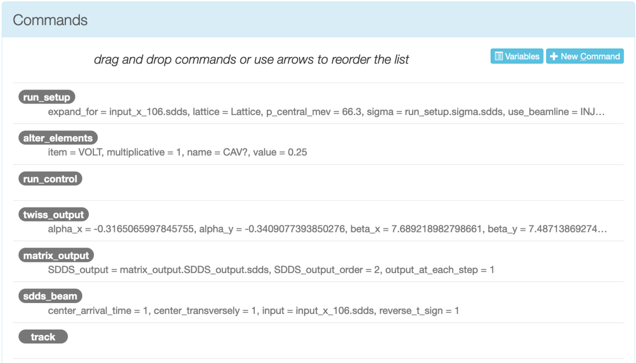
Neural Network Tools

Sirepo provides a workflow for modeling particle accelerators

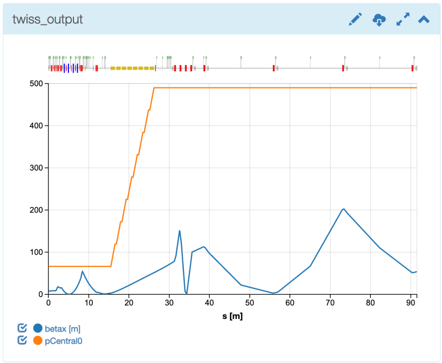
- Codes
  - Elegant – position based tracking including collective effects such as CSR
  - OPAL – time-based tracking with space-charge
  - Mad-X – position based tracking for linacs and storage rings
  - Zgoubi – integration of particle tracks through field maps including spin tracking
- Workflow
  - Beam initialization – lattice definitions – simulation control - visualizations

https://www.sirepo.com/en/apps/particle-accelerators/

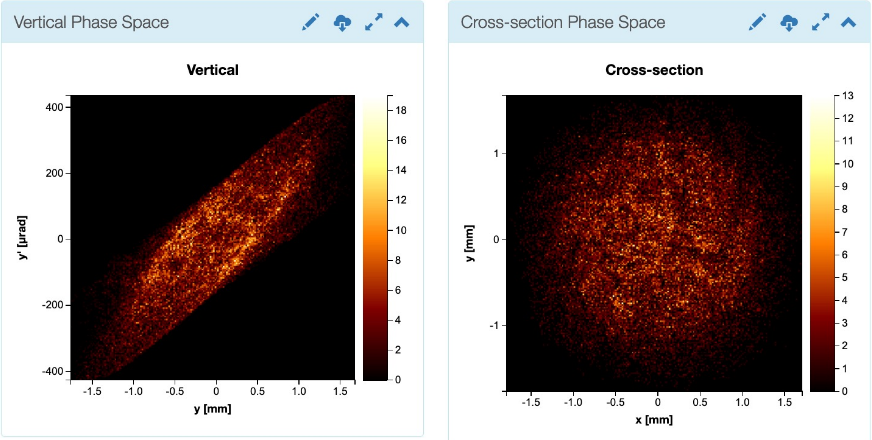
Simulation Control



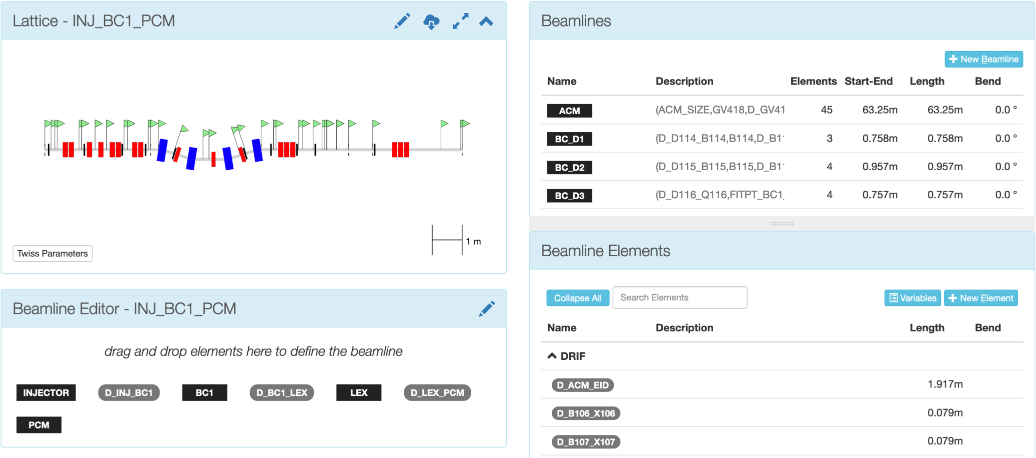
Simulation Output



Beam Initialization



Lattice Definition



Sirepo supports comprehensive modeling of X-ray beamlines

- SRW and Shadow for x-ray optics
  - SRW is a physical optics code that simulates partially coherent and fully coherent radiation
  - Shadow is a ray-tracing code
- Beam initialization interface provides initial diagnostics on undulator characteristics
- Beamline definition utilizes a drag and drop interface

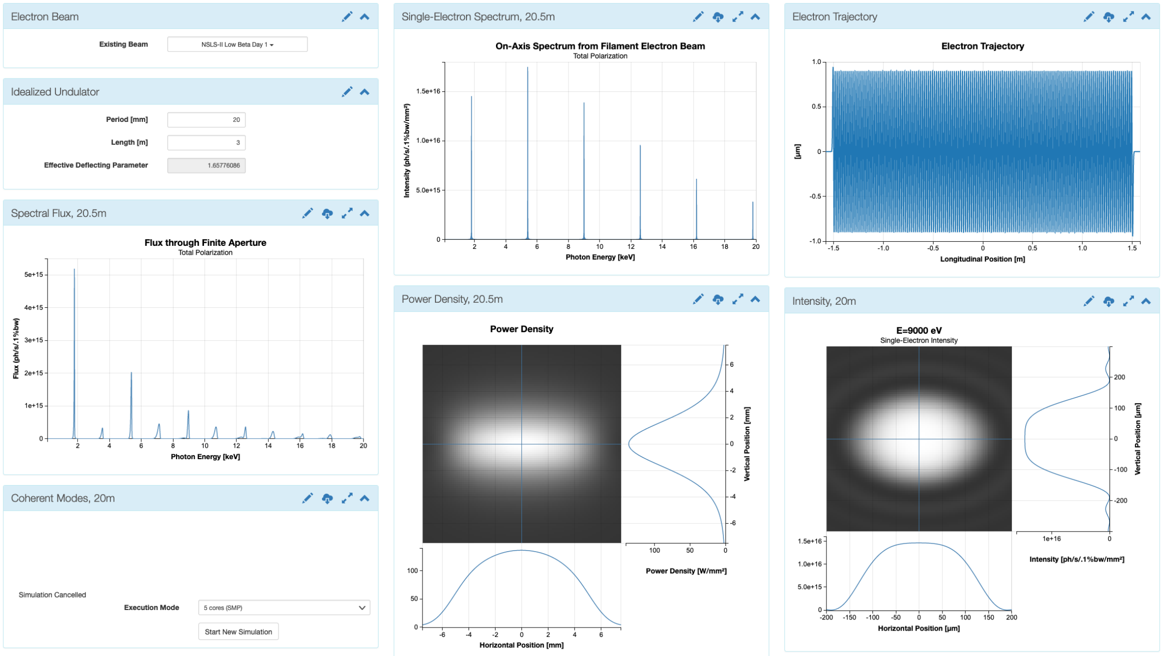
† O. Chubar et al. "Simulation of experiments with partially coherent x-rays using Synchrotron Radiation Workshop". In: Proc. SPIE. Vol. 10288. Aug. 2017

† L. Wiegart et al. "Towards the simulation of partially coherent x-ray scattering experiments". AIP Conf. Proc. **2054**, 060079 (2019).

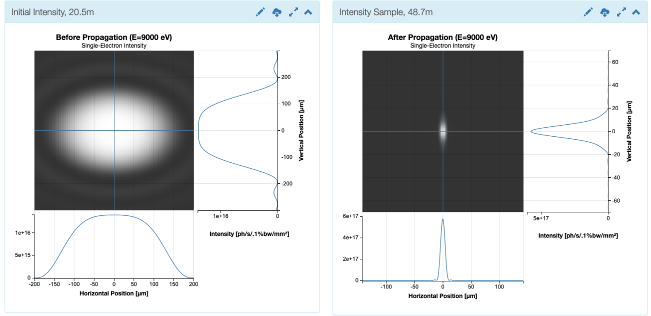
*M. S. Rakitin et al. "Introduction of the Sirepo-Bluesky interface and its application to the optimization problems". In: Proc. SPIE. Vol. 11493. Aug. 2020.

https://www.sirepo.com/en/apps/x-ray-beamlines/

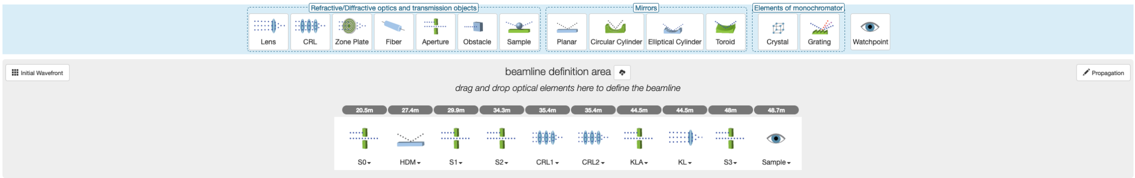
Initialization



Simulation Results

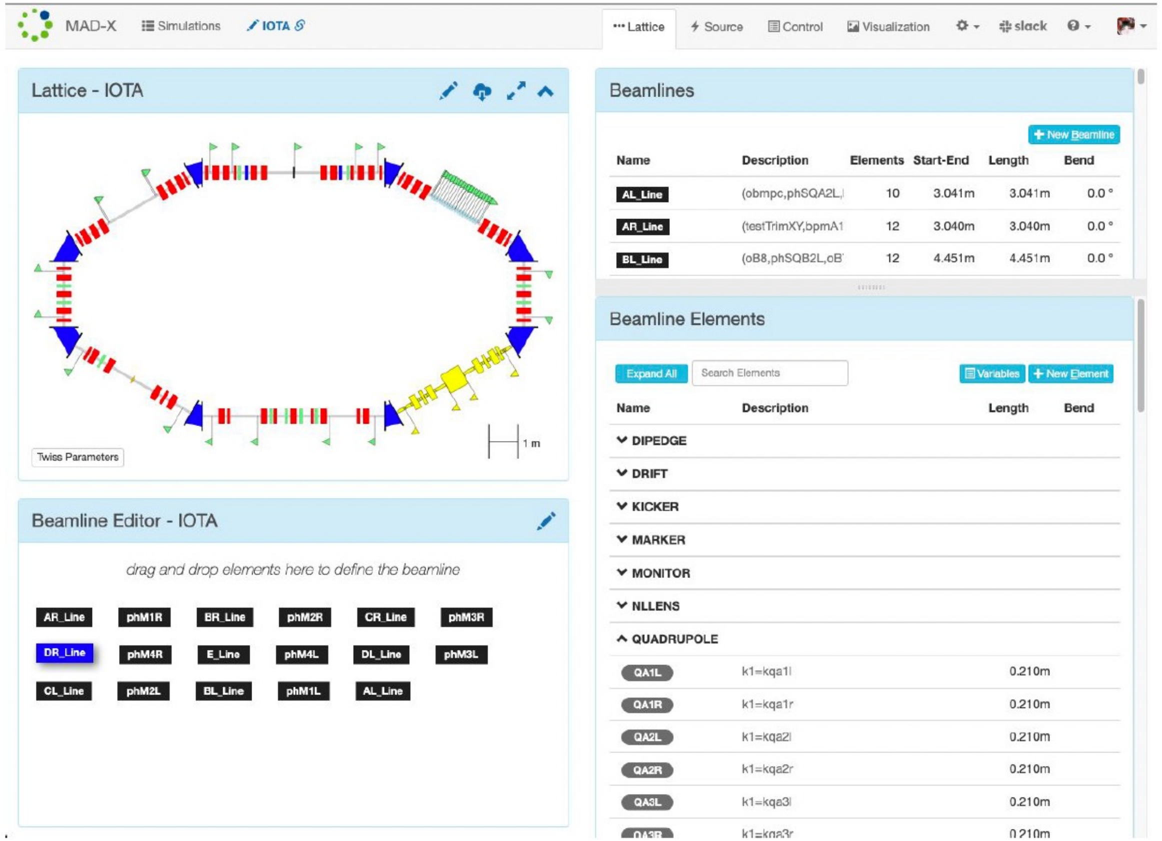


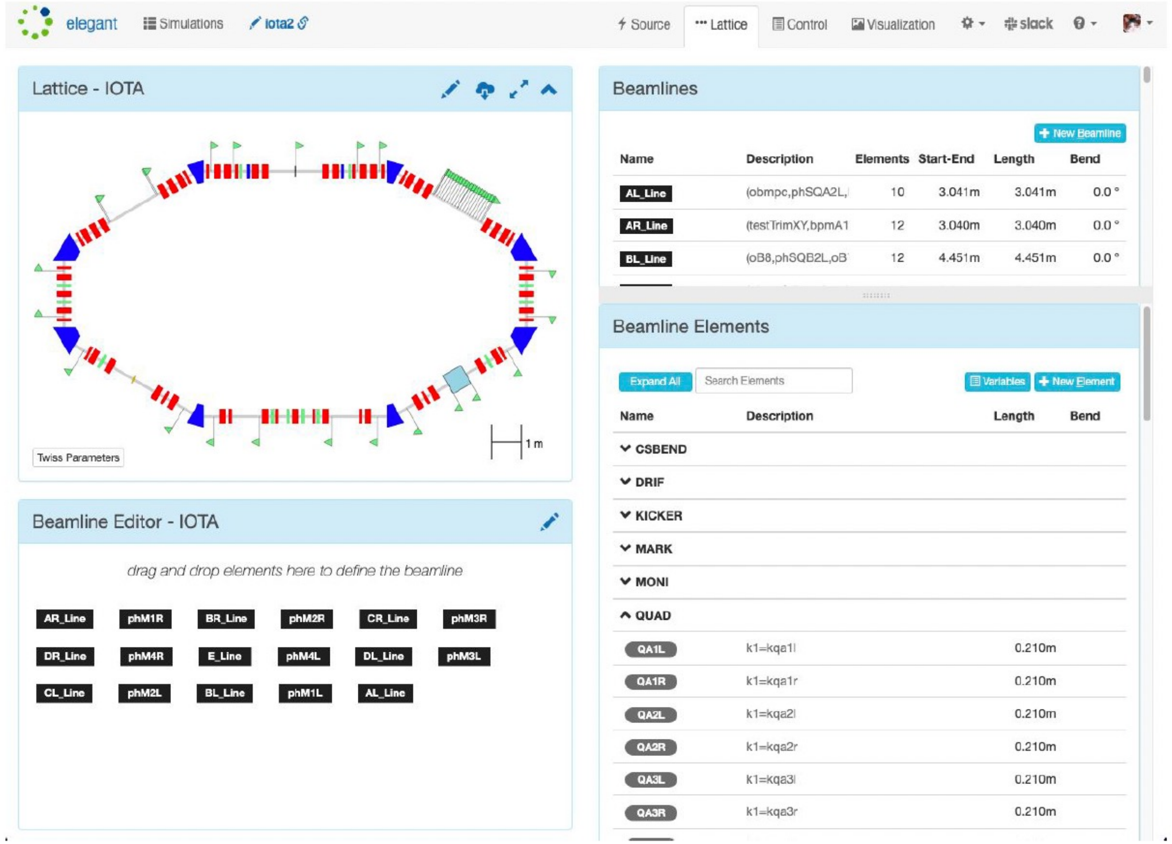
Beamline Definition



Common lattice editor across simulation codes

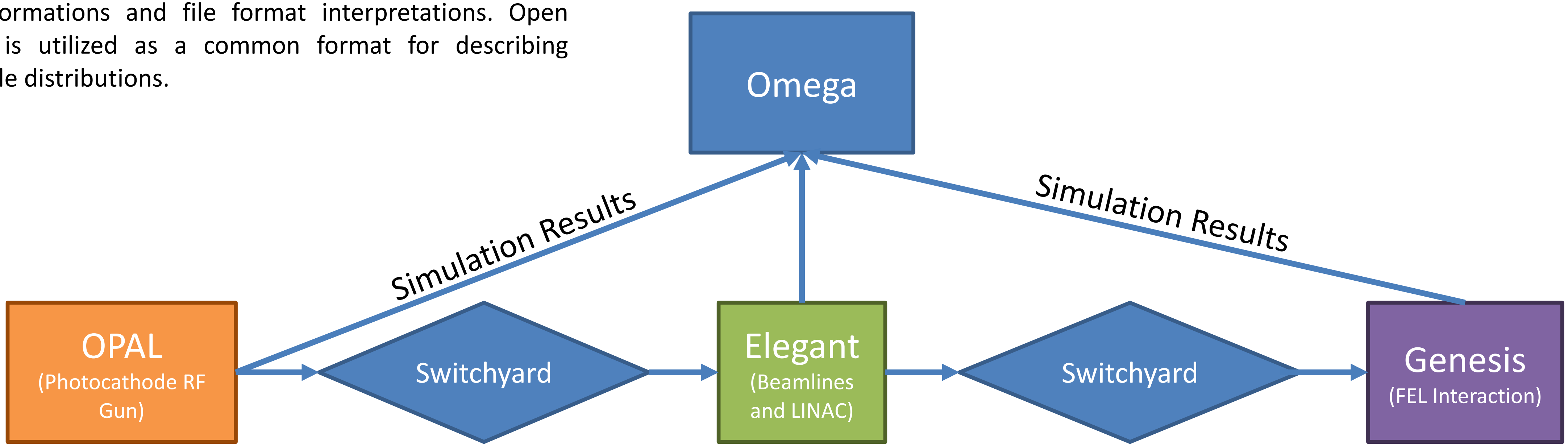
- Code agnostic lattice designer
  - Long term: abstraction of lattice elements
  - Seamless integration of different computation engines





Sirepo Omega: A workflow for end-to-end simulations

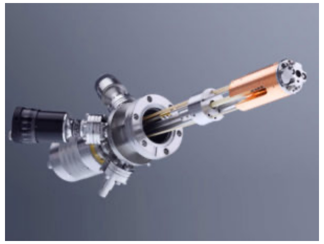
Switchyard implements the necessary coordinate transformations and file format interpretations. Open PMD is utilized as a common format for describing particle distributions.

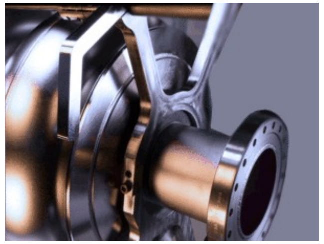


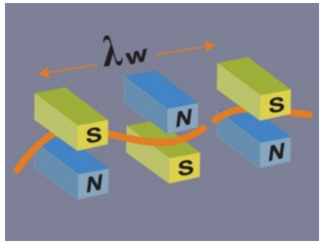
Time based particle tracking for the low energy regime (handles space-charge)

Position based particle tracking for the medium and "high" energy regimes (handles CSR and longitudinal space-charge)

Time dependent and time independent FEL simulations

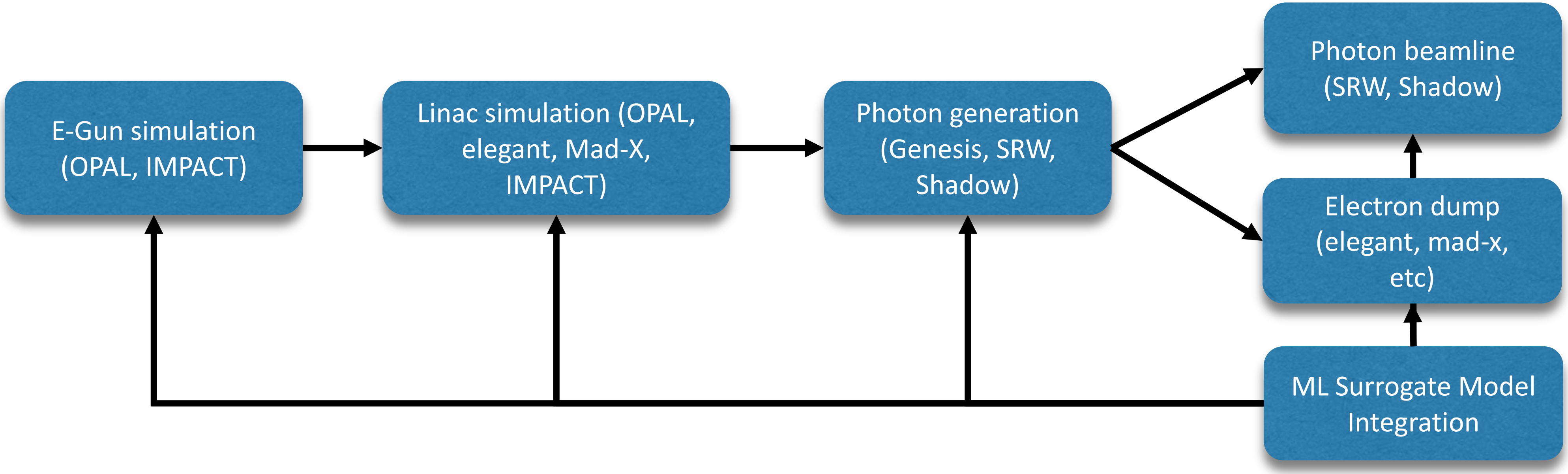






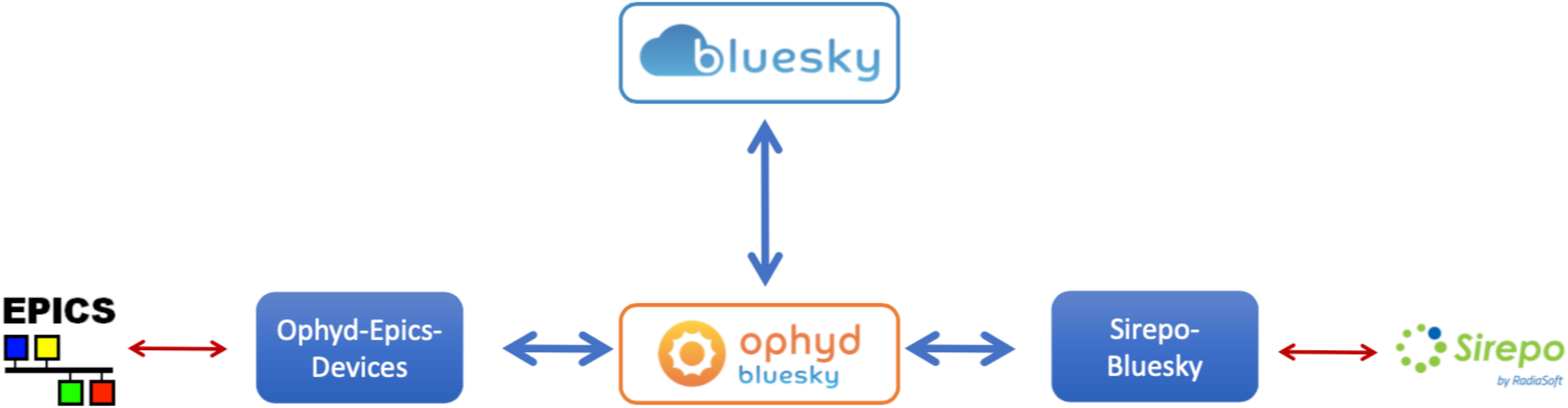
End-to-End Simulations and ML infrastructure for Light Sources

- LUME and Sirepo provide complementary integration tools
- Creating interoperability between these tools will improve the end-to-end simulation infrastructure
- Integrate machine-learning model infrastructure into Sirepo
- Demonstrate model deployment utilizing photon beamline test bench
- Provide integration with optimization workflows and controls



Chain simulations in Sirepo: Export to LUME: Deploy as digital twin

Connecting online models to experiments

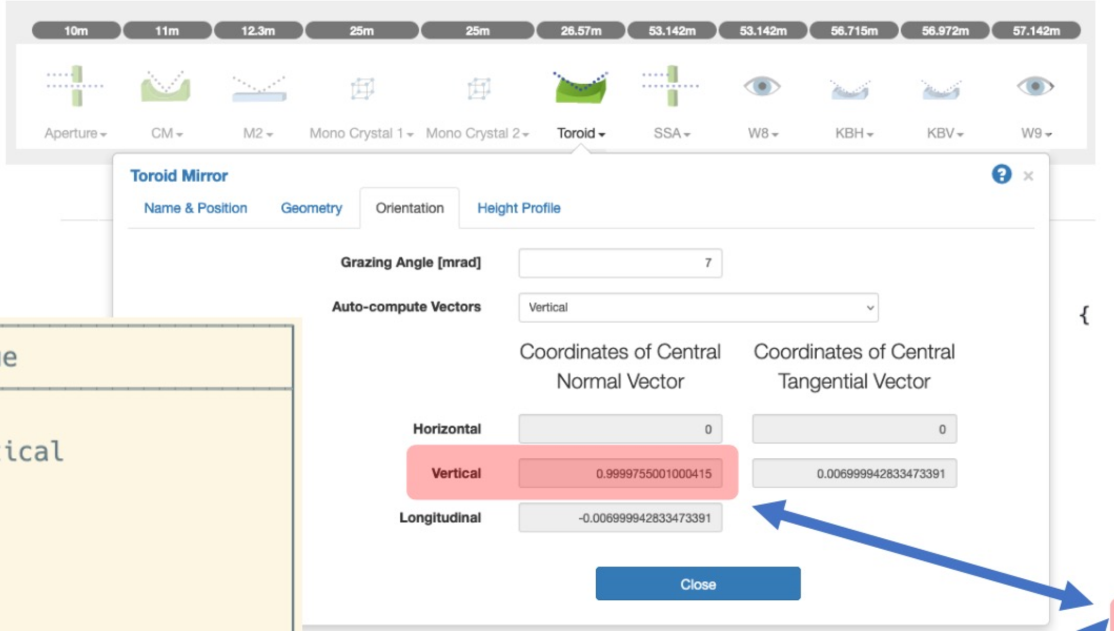


Operations
Experimentalists

SRW
SHADOW
MAD-X
Reduced models

Model result can be used to make choices on the beamline!

Sirepo-Bluesky: Linking Simulated Devices with Operations



Ophyd Device Representation

Ophyd object/component	Value
toroid_apertureShape	r
toroid_autocomputeVectors	vertical
toroid_grazingAngle	7
toroid_heightAmplification	1
toroid_heightProfileFile	
toroid_horizontalPosition	0
toroid_id	6
toroid_normalVectorX	0
toroid_normalVectorY	0.9999755001000415
toroid_normalVectorZ	-0.006999942833473391
toroid_orientation	y
toroid_sagittalRadius	0.186
toroid_sagittalSize	0.08
toroid_tangentialRadius	24500
toroid_tangentialSize	0.96
toroid_tangentialVectorX	0
toroid_tangentialVectorY	0.006999942833473391
toroid_title	Toroid
toroid_type	toroidalMirror
toroid_verticalPosition	0
toroid_element_position	26.57

Sirepo JSON


```
{
    "apertureShape": "r",
    "autocomputeVectors": "vertical",
    "grazingAngle": 7,
    "heightAmplification": 1,
    "heightProfileFile": "",
    "horizontalPosition": 0,
    "id": 6,
    "normalVectorX": 0,
    "normalVectorY": 0.9999755001000415,
    "normalVectorZ": -0.006999942833473391,
    "orientation": "y",
    "position": 26.57,
    "sagittalRadius": 0.186,
    "sagittalSize": 0.08,
    "tangentialRadius": 24500,
    "tangentialSize": 0.96,
    "tangentialVectorX": 0,
    "tangentialVectorY": 0.006999942833473391,
    "title": "Toroid",
    "type": "toroidalMirror",
    "verticalPosition": 0
},
```

Sirepo-Bluesky

https://github.com/NSLS-II/sirepo-bluesky
https://nsls-ii.github.io/sirepo-bluesky
https://doi.org/10.1117/12.2569000

DOI 10.5281/zenodo.8265981 | Tests passing | PyPI v0.7.2 | conda-forge v0.7.2

- Available on **conda-forge** and **PyPI**
- Support of the **SRW**, **Shadow3**, and **MAD-X** applications in Sirepo
- Simulations are performed on a Sirepo server (a VM, Docker container, or HPC resources)
- Communication is done over HTTP(s) with Sirepo REST API
- Watchpoints or other Sirepo "reports" are wrapped into dedicated "detector" Ophyd objects
- All other optical elements are wrapped into Ophyd's **`Device`**s with **`Signal`**s corresponding to individual parameters in Sirepo
- The exchange format is JSON
- List of predefined simulations in Sirepo: https://nsls-ii.github.io/sirepo-bluesky/simulations.html

List of predefined simulations in Sirepo

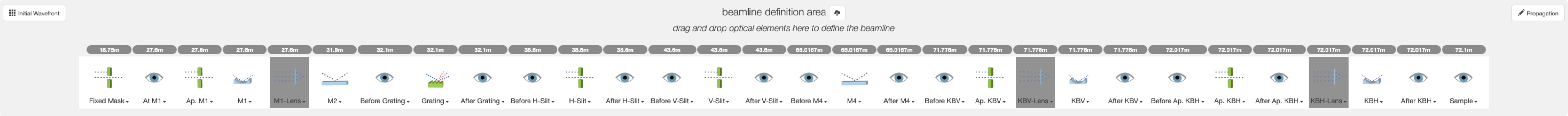
Below is a list of custom/predefined simulations available when one starts Sirepo following the Sirepo startup instructions, that are currently used for tests and demos.

SRW

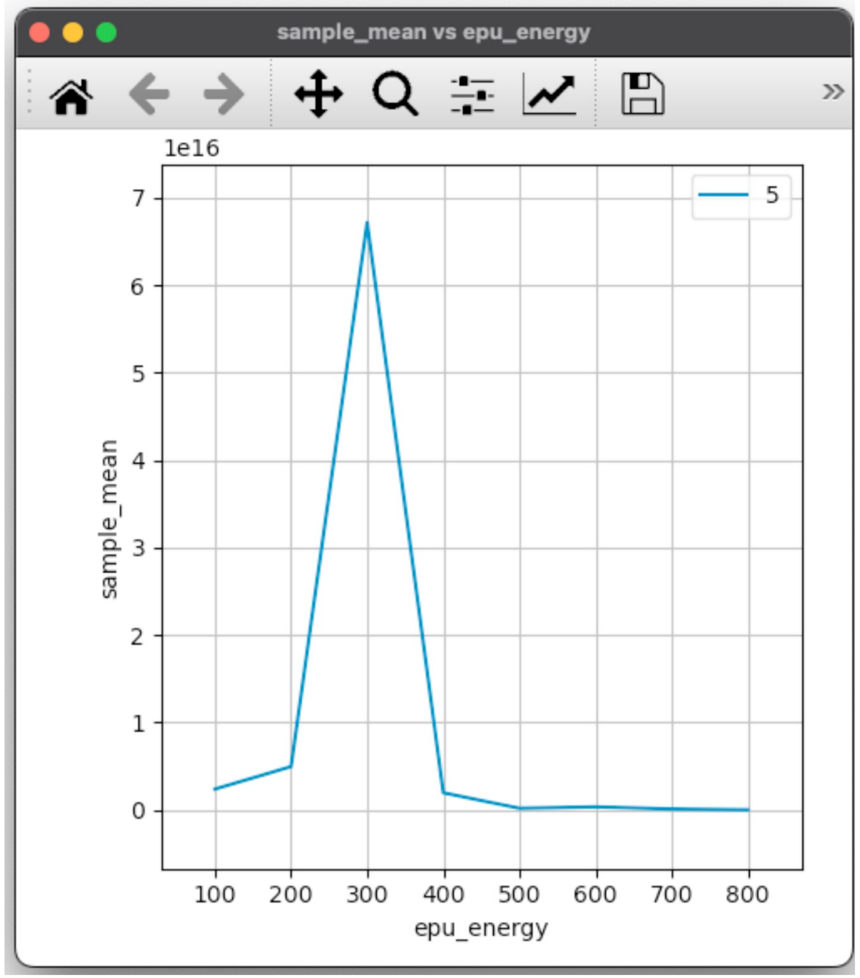
SRW simulations used for testing

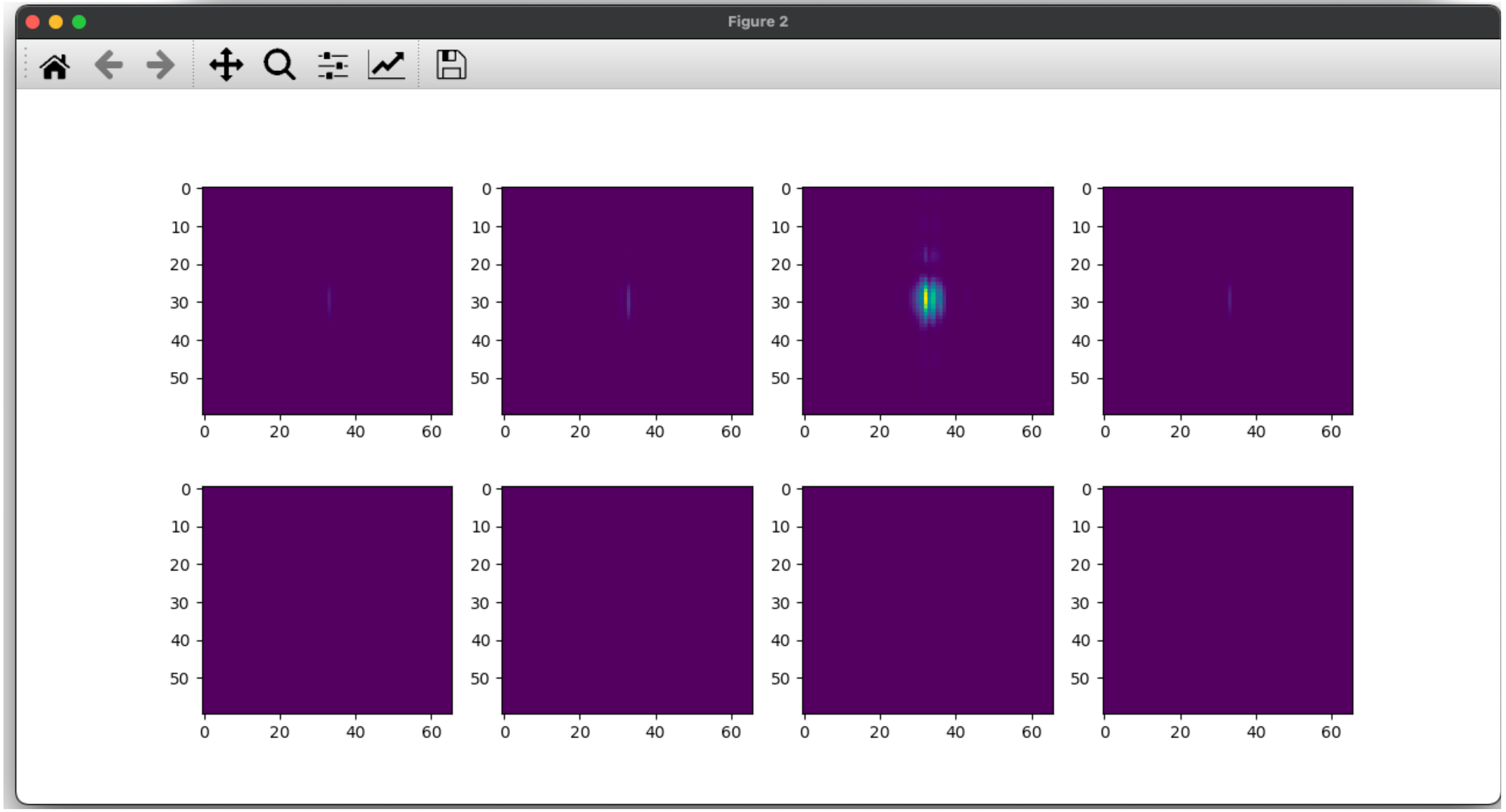
Simulation ID	Description
`00000000`	Young's Double Slit Experiment
`00000001`	basic
`00000002`	TES
`00000003`	PD ARI-RIXS 250eV_JulyReviewVersion oc
`00000004`	PD ARI-RIXS 400eV (tuned) oc
`00000005`	PD ARI-ARPES 150eV JulyReviewVersion oc
`00000006`	PD ARI-ARPES 250eV JulyReviewVersion oc
`00000007`	SXN_PD_lowE_250eV
`00000008`	SXN_PD_medE_1000eV
`00000009`	SXN_PD_highE_2000eV

NSLS-II ARI virtual beamline




```
RE(bp.scan([sample], epu.energy, 100, 800, 8))
```





Applications

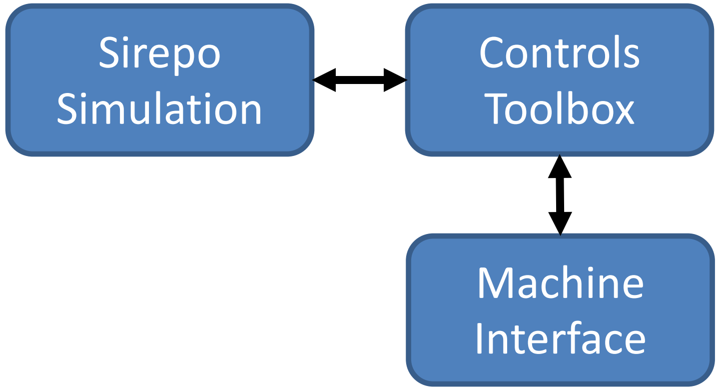
- Sirepo-bluesky supports three simulators: shadow, SRW, and MAD-X

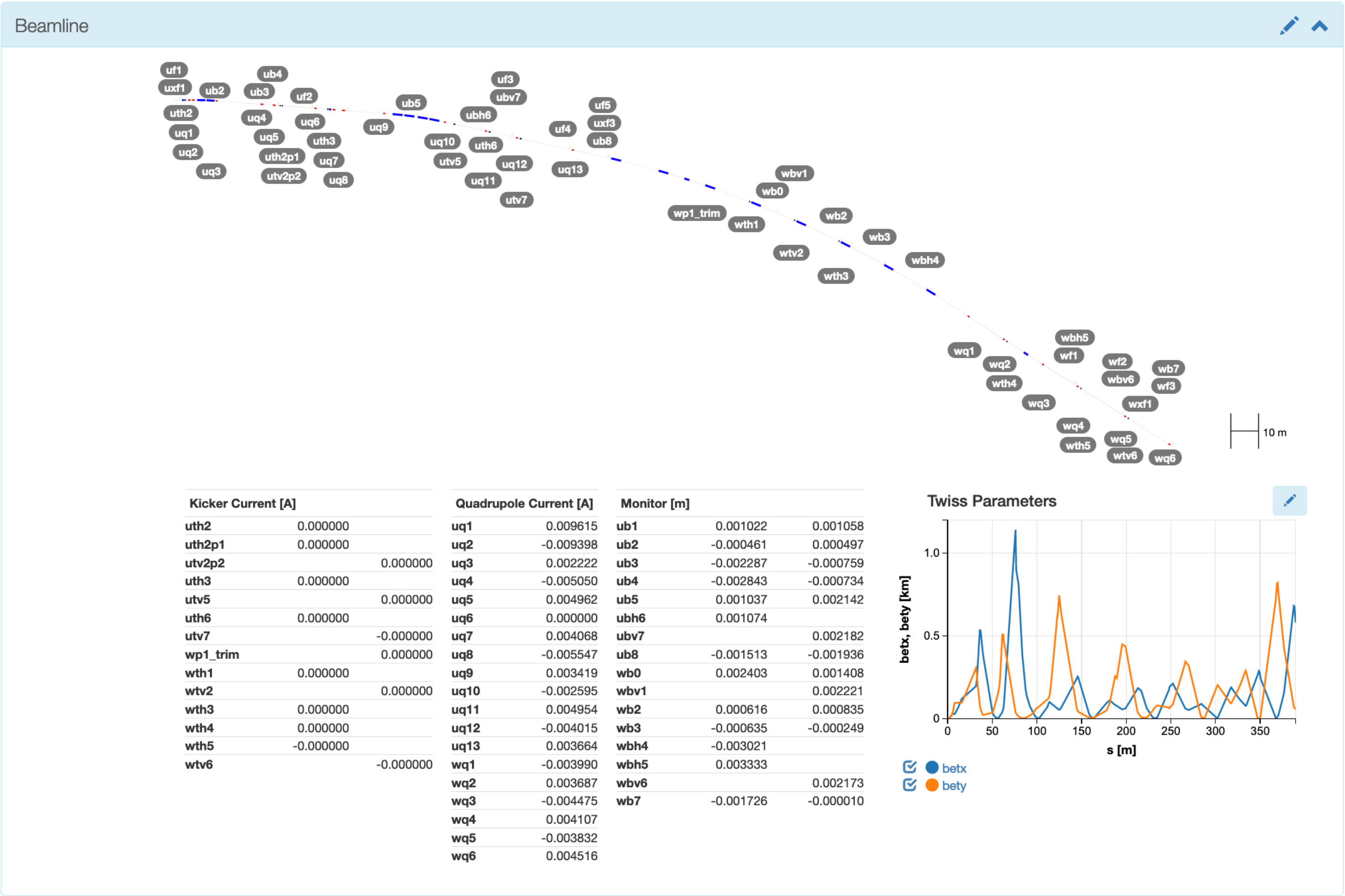
- The Sirepo-Bluesky library is used in the following projects:

  - https://github.com/NSLS-II/bloptools: beamline optimization tools
  - https://github.com/BNL-ATF/profile_atf: BNL ATF facility uses it for MAD-X simulations
  - https://github.com/NSLS-II-ARI/profile_sirepo_ari: used to prototype data acquisition plans for the future NSLS-II ARI beamline (currently under construction)

Sirepo as a toolbox for accelerator controls

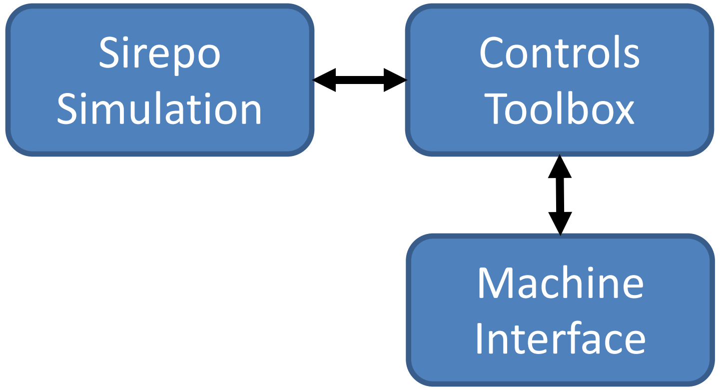
- The controls toolbox combines the simulation workflow of Sirepo with displays that provide visualizations for operators
  - Middleware can allow the users to examine the simulation output, machine settings, or logger data.
  - Machine settings can be loaded into the simulations for optimization
- User interface is generated automatically from existing simulations



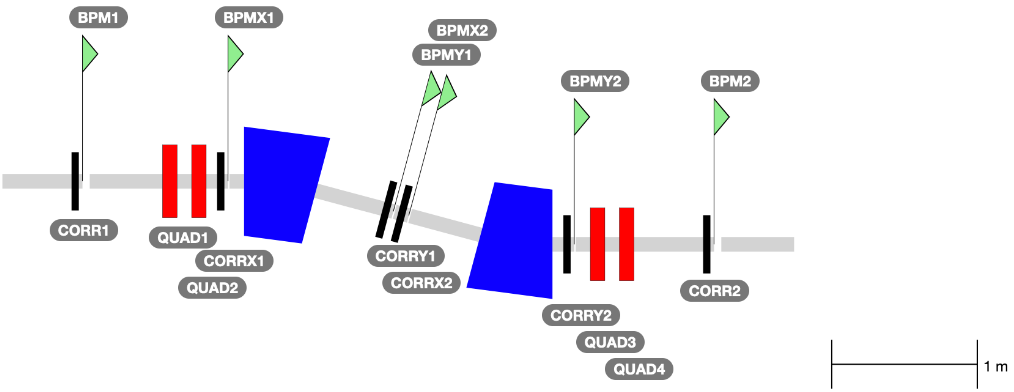


Sirepo as a toolbox for accelerator controls

- Conversion between current and magnet strength handled on back-end
  - Simplifies integration of online models with operations
  - Magnet excitation curves can be loaded for different magnet types
- Optimization routines for beam tuning are readily available



Beamline

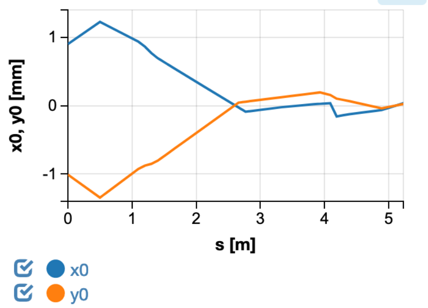


Kicker Current [A]		
CORR1	-0.000127	0.000174
CORRX1	0.000001	
CORRY1		-0.000136
CORRX2	0.000167	
CORRY2		-0.000085
CORR2	0.000042	0.000097

Quadrupole Current [A]	
QUAD1	1.216968
QUAD2	-1.216968
QUAD3	1.216968
QUAD4	-1.216968

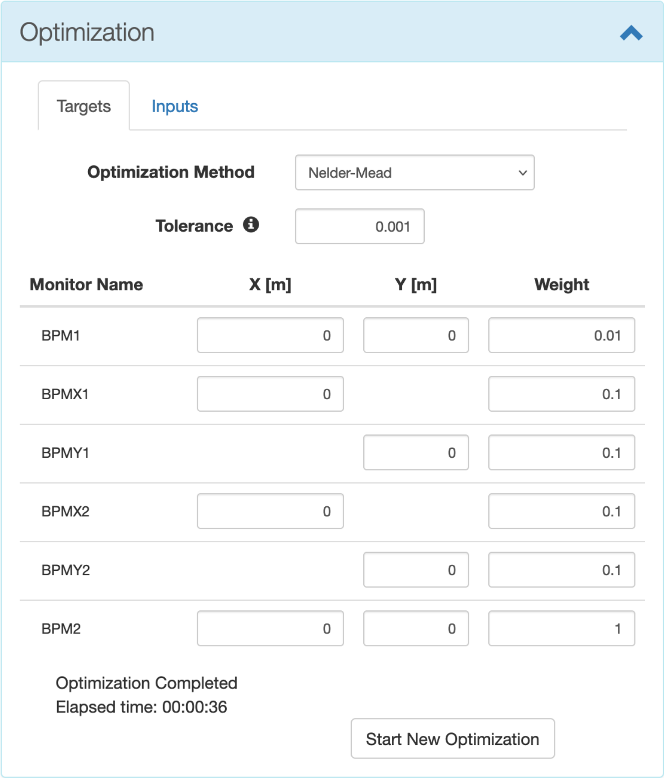
Monitor [m]		
BPM1	0.001176	-0.001274
BPMX1	0.000602	
BPMY1		0.000052
BPMX2	-0.000082	
BPMY2		0.000180
BPM2	-0.000035	-0.000021

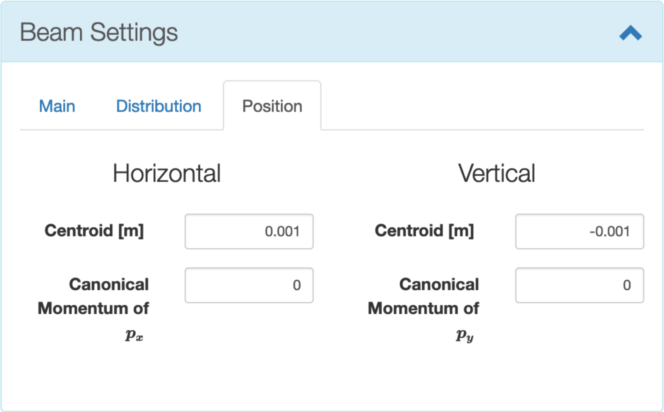
Twiss Parameters



Sirepo as a toolbox for accelerator controls

- Optimization targets for a given beamline are specified along with initial conditions
  - Optimization methods can be accessed via Python or rsopt
  - Currently Nelder-Mead and Linear Response Matrix methods implemented for demonstration purposes

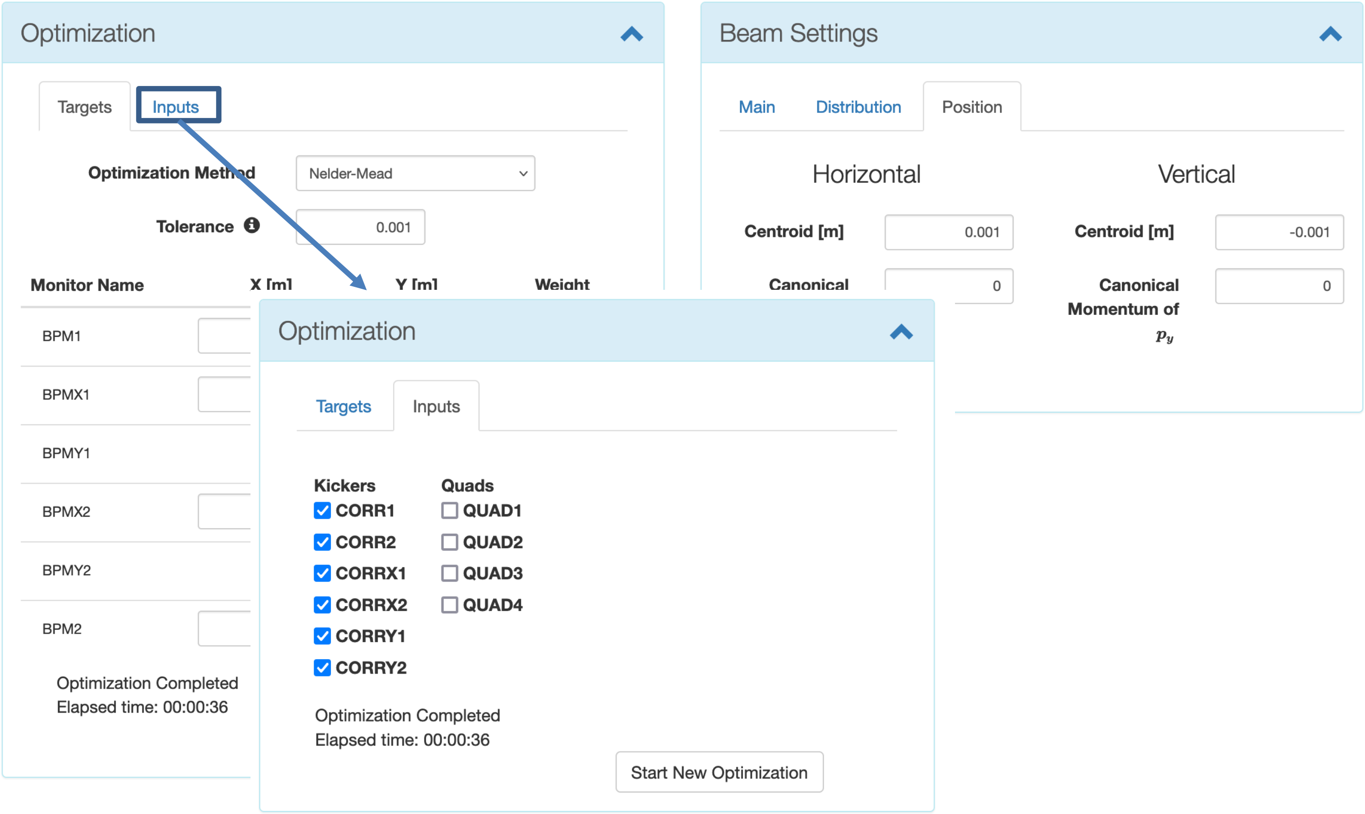




Sirepo as a toolbox for accelerator controls

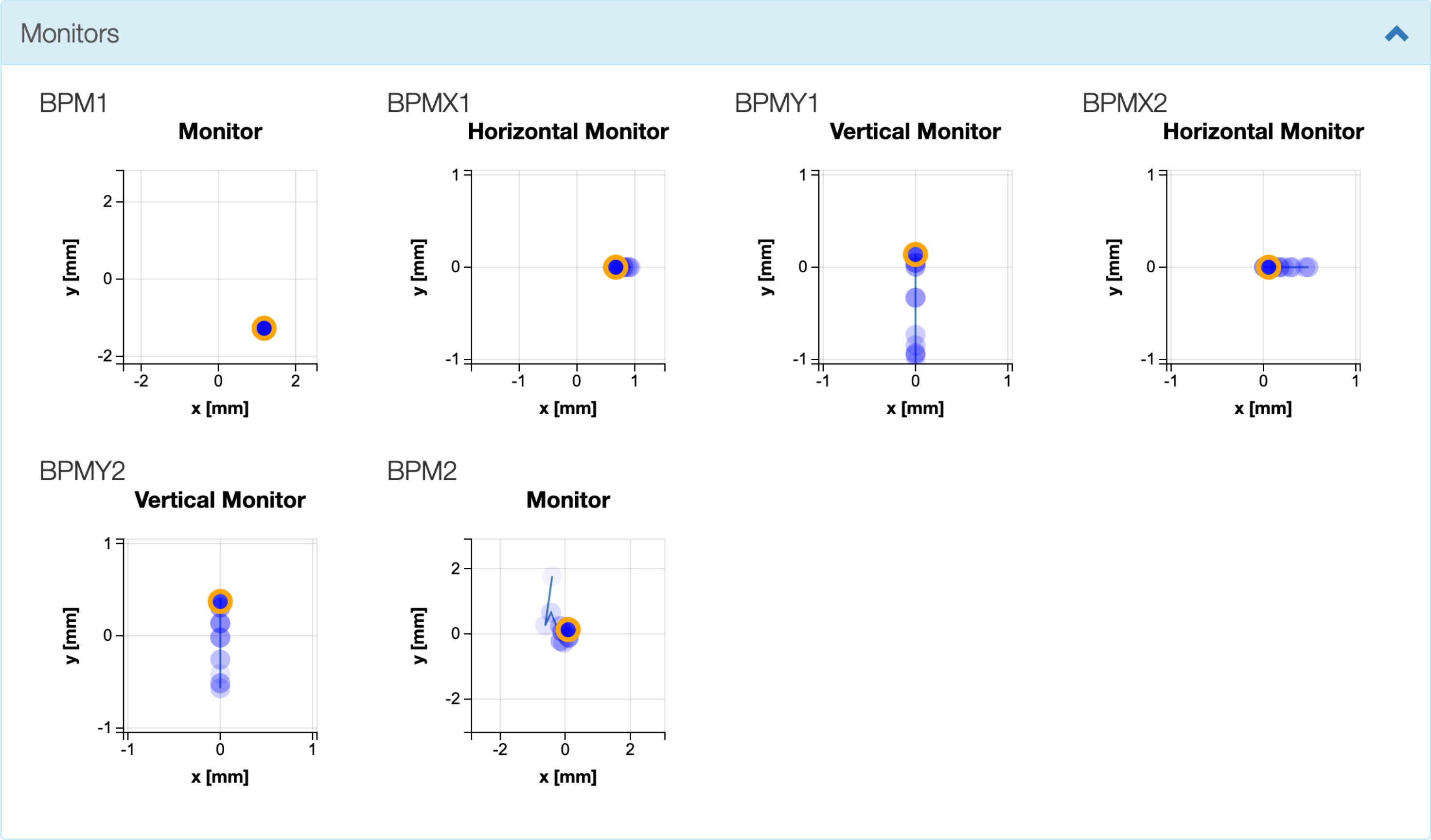
- Optimization targets for a given beamline are specified along with initial conditions
  - Optimization methods can be accessed via Python or rsopt
  - Currently Nelder-Mead and Linear Response Matrix methods implemented for demonstration purposes

- Optimization inputs are selectable
  - Tunable parameters are automatically populated by the application



Sirepo as a toolbox for accelerator controls

- Visualizations update when simulation data (or machine data) become available
- Data is colored based on timestamp
  - More recent data is darker
  - Most recent datapoint is outlined in orange
- Diagnostics are populated automatically based on the lattice
  - Customization is possible



Conclusions

- Why do we want browser-based interfaces?
  - Platform agnostic – Can run on different operating systems or hardware (tablets and phones for example)
  - Ease of access – interfaces are shared via a web-link

- Sirepo framework supports accelerator simulations and controls
  - Accelerator design – end to end modeling and optimization
  - Online modeling – virtual experiments with Sirepo Bluesky
  - Controls – direct link between simulations and controls for machine optimization

Contact Information

Jonathan Edelen, President

RadiaSoft LLC

jedelen@radiasoft.net

Disclaimer