ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-WEPV024

X-RAY BEAMLINE CONTROL WITH MACHINE LEARNING AND AN ONLINE MODEL*

Boaz Nash, Dan Tyler Abell, David Leslie Bruhwiler, Evan Carlin, Jonathan Edelen, Michael Keilman, Paul Moeller, Robert Nagler, Ilya V. Pogorelov, Stephen Davis Webb
RadiaSoft LLC, Boulder, Colorado, USA
Maksim Rakitin, Yonghua Du, Abigail Giles, Joshua Lynch, Jennefer Maldonado, Andrew Walter
Brookhaven National Laboratory, Upton, New York, USA

Abstract

We present recent developments on control of x-ray beamlines for synchrotron light sources. Effective models of the x-ray transport are updated based on diagnostics data, and take the form of simplified physics models as well as learned models from scanning over mirror and slit configurations. We are developing this approach to beamline control in collaboration with several beamlines at the NSLS-II. By connecting our online models to the Blue-Sky framework, we enable a convenient interface between the operating machine and the model that may be applied to beamlines at multiple facilities involved in this collaborative software development.

INTRODUCTION

Electron storage ring based light sources and x-ray FELs provide radiation for a vast array of scientific research. The radiation is created in either undulator or bending magnet sources and then passes down an optical beamline, transporting the radiation to the sample for use in a scientific study.

Although modeling of beamline components and x-ray transport plays an important role in the design and commisioning of x-ray beamlines, these methods are largely unused during day to day operations. In comparison to the electron beam storage ring, in which reduced models combined with diagnostics play a crucial role in electron beam control, little has been done on the x-ray beamline side regarding such methods.

Modeling codes for x-ray beamlines can generally be broken into two classes, either wavefront propagation or ray tracing. The two most widely used codes of these types are Synchrotron Radiation Workshop (SRW) and Shadow. Whereas in principle, either of these codes could be suitable for an online model, there are some deficiencies which we look to improve. In particular, ray tracing methods, though fast and accurate, do not take into account diffraction effects or the effects of partial coherence. Wavefront propagation, on the other hand, while including diffraction effects, is substantially more computationally intensive, particularly when partially coherent computations are required.

We propose two types of simplified models that provide additional tools with which to build fast online models for x-ray beamlines. The first is a physics based model based on what we call a *Matrix-Aperture Beamline*[1]. And the second type of model is a surrogate model using the methods of machine learning on either simulated or measured training data. We describe our progress on the development of each of these types of models and their application to x-ray beamlines at NSLS-II at Brookhaven National Laboratory.

Next, in order to create an accurate online model that ties the beamline and photon beam status to a software model, we must make measurements at beamline diagnostics and take into account beamline component positions using the existing beamline control software. For this purpose, we use the BlueSky software, developed at NSLS-II, but with a goal for wider adoption across the world of synchrotron light sources in the US and beyond.

We document here our continuing efforts to improve this situation by creation of online models in synchrotron light sources that may be used for automated beamline control. We are working with the TES bending magnet beamline at NSLS-II [2], and thus draw our simulations and control examples from this beamline.

REDUCED PHYSICS MODELS

Our reduced physics models are based on the concept of a *Matrix-Aperture-Beamline*. We call the code we are building to implement this concept `MABTrack`, and a diagram for the case of two apertures is shown in Fig. 1. The `MABTrack` code is being developed on GitHub in the `rslight` repository[1] and will be available as an open source package.

The `MABTrack` code starts with a model in SHADOW to compute reference orbit $\vec{Z}_0(s)$, ABCD matrices $M_j(s)$ and physical apertures $t_j(\vec{x})$. Three levels of sophistication in modeling are being developed within `MABTrack`. At the simplest level, one computes the propagation of the radiation second moment matrix $\Sigma(s)$, defined as $\Sigma_{jk}(s) = \langle \vec{\tilde{z}}_j \vec{\tilde{z}}_k \rangle$ with $\vec{\tilde{z}}(s) = \vec{z}(s) - \vec{z}_0(s)$, i.e. subtracting off the reference orbit. The second moments propagate as

$$\Sigma(s) = M(s)\Sigma_0 M^T(s) \tag{1}$$

with Σ_0 as the initial values of the second moments.

As a second level of sophistication, we will use *linear canonical transforms* (LCTs) to transport wavefronts [3]. LCTs have an intimate connection to the so-called *ABCD*

* Work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, including grant No. DE-SC0020593

[1] `https://github.com/radiasoft/rslight`

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-WEPV024

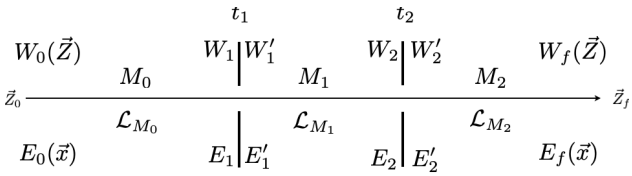

Figure 1: Structure for `MABTrack`, Matrix Aperture Beamline Tracking code for the case with 2 physical apertures $t_1(\vec{x})$ and $t_2(\vec{x})$. Transfer matrices (M_0, M_1, and M_2) are assumed to be known, either analytically, or via a ray tracing code such as Shadow. Either electric fields $E_0(\vec{x})$ or Wigner functions $W_0(\vec{Z})$ may be propagated through the beamline.

matrices of ray optics, and constitute a generalization of the various integral transforms—including Fresnel transforms, Fourier transforms, fractional Fourier transforms, scaling, and chirp multiplication (multiplication by a Gaussian)—used for computations in wave optics. In one degree of freedom, one may write the LCT of a function g in the form [3]

$$\tilde{g}^M(u) = \mathrm{e}^{-\mathrm{i}\pi/4}\sqrt{\beta} \times \int_{\infty}^{\infty} \exp[\mathrm{i}\pi(\alpha u^2 - 2\beta u u' + \gamma u'^2)] g(u')\,\mathrm{d}u'. \quad (2)$$

As an alternative to the parameters α, β, γ, one may instead parameterize 1D LCTs in terms of the entries of a 2×2 symplectic matrix M—an *ABCD* matrix, the superscript seen on the left-hand side of (2)—according to the rule

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} \gamma/\beta & 1/\beta \\ -\beta + \alpha\gamma/\beta & \alpha/\beta \end{pmatrix}. \quad (3)$$

When computed naïvely, the integral transform (2) represents a computationally expensive task. To address this difficullty, one may take advantage of the *group property* of the LCT, which states that

$$\tilde{g}^{M_2 M_1} = \tilde{g}^{M_2} \circ \tilde{g}^{M_1}. \quad (4)$$

This property tells us that if one can factor the matrix M into a product of (simpler) matrices M_1 and M_2, and if one can easily compute the LCTs corresponding to those simpler matrices, then one may easily compute the full LCT by composition. This remains true even if when we factor M into a product of more than two matrices.

In special cases, the LCT becomes quite tractable. In the case $\alpha = \gamma = 0$ and $\beta = 1$, for example, the LCT becomes, apart from a normalization, a Fourier transform. Other choices of the parameters lead to the other well-known transforms mentioned above, for which fast algorithms are known. Moreover, it is always possible to factor M into a product of matrices corresponding to such transformations. Indeed, there exists a substantial literature on this topic. See, for example, [4–10]. It is worth mentioning that, as described in some of the references, that the general procedure described above works also for 2D LCTs. In addition, we mention that, despite the large literature, it appears that no (publicly available) LCT library currently exists. Hence the need to write one of our own.

To implement the LCT, we have begun with the case of one degree of freedom, and have implemented (i) some needed utility functions, (ii) the component LCT functions, and (iii) functions to decompose the given LCT into simpler parts and to compose the resulting component LCTs. Because this code is being written in Python, we make—in writing all these functions—extensive use of `numpy`[2], a high-performance numerical library for Python.

The mentionad utilities enable us to convert between parameters a, b, c, d and α, β, γ (`convert_params_3to4` and `convert_params_4to3`); construct the abscissae for a given signal (`abscissae`); and resample a given signal (`resample_data`), which is needed because chirp multiplication increases the effective time-bandwidth product [5, 9]. All these utilities have been implemented and tested.

For the component LCTs, we need scaling (`scale_data`), the Fourier transform (`fourier`), and chirp multiplication (`chirp_multiply`). All three of these have been written, and we are in the process of testing.

The remaining functions we need are one for decomposing a given LCT into simpler parts (`lct_decompose`), and a second for composing the component LCTs (`apply_lct`). These functions have also been written, and they have passed some (rather trivial) tests. Final testing will happen after we complete testing of the component LCTs. When the LCT algorithms have been tested, they will be integrated into `MABTrack`. This effort will then be generalized to 2D LCTs.

The final level of sophistication in `MABTrack` involves direct propagation of Wigner functions which allows for inclusion of partial coherence. See [11] for details of this approach. Alternate approaches to partial coherence include propagation of the cross spectral density (Fourier transform of Wigner function), macroparticle (MP) sampling of the electron beam distribution and coherent propagation, and Coherent Mode Decomposition(CMD), leading to a smaller number of coherent modes than the MP method. We will explore all these methods, comparing efficiency to decide on our partially coherent model. As a fast CMD method has recently been implented in SRW, we will be applying this to bending magnet and undulator beamlines and then performing the coherent propagation via LCT using the computed ABCD matrices from Shadow.

MACHINE LEARNING MODELS

Preliminary efforts to apply ML algorithms to learn a surrogate beamline model were reported in [1] in which a sample KB beamline was used to train a neural network with SRW simulations to reconstruct mirror errors. We have since generalized the tools for the SRW simulations, integrating them into our Sirepo GUI interface to SRW[3]. The data generated by the resulting simulations can then be used to train ML algorithms. We have included more elements in the

[2] `https://numpy.org/`
[3] `https://www.sirepo.com/srw#`

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-WEPV024

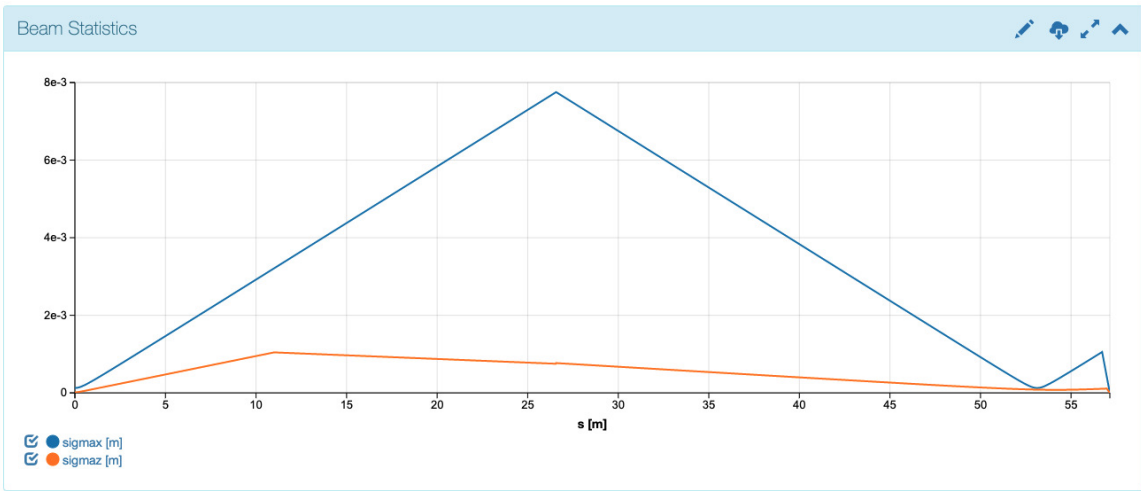

Figure 2: Horizontal (blue) and vertical (orange) beam-size propagation in the beam statistics report for the TES beamline.

interface for the data generating scripts initially developed as part of Phase I. In particular, we have implemented the following new elements and their corresponding parameters:

- aperture (position, and horizontal and vertical size);
- circular cylinder (position, rotation, and radius);
- toroidal mirror (position, rotation, and tangential and sagittal radii).

Figure 3 shows a screenshot of the interface for these new elements.

Continuing to develop our ML beamline capabilities, a subset of the NSLS-II development team worked on the Deep Beamline Simulation software.[4] That effort aims to train neural networks on solved problems, with the goal of lowering the computational cost of x-ray beamline simulations by using machine learning to simulate beamlines with a fidelity similar to SRW. We want to determine whether or not a neural network trained on data generated by previously created simulation tools (Sirepo-Bluesky, SRW, Shadow) can effectively identify relevant patterns and then solve newly-posed problems of similar type. In Fig. 4 we show some preliminary results from that effort. Tensorboard has been used to plot the loss function as the network learns the distribution resulting from varying a physical aperture in the beamline.

INTERFACE WITH BLUESKY BEAMLINE CONTROL SOFTWARE

The BlueSky software[5][12] allows beamline scientists to control their beamline while performing x-ray experiments. BlueSky plans are created which lead to motion of beamline components such as mirrors, monochromators, slits, etc. In order to connect the physical beamline to a simulated beamline, the Sirepo-Bluesky library.[6][13] has been created. Sirepo allows access to beamline simulation codes such as SRW and SHADOW, and will also include the `MABTrack` code when completed.

We have addressed a number of topics to improve the Sirepo-Bluesky library. When first developed, the library focused on supporting the SRW simulation code within Sirepo. After their recent refactoring of the library, the team added support for the Shadow simulation code within Sirepo. This effort has resulted in a more modular code: Support for SRW and Shadow are now implemented in separate Python modules, which provides for a cleaner separation between corresponding classes. This work included the creation of corresponding tests for the newly added features, as well as updates to the previous tests to ensure that new additions do not cause any regression or performance issues. As part of updating the testing framework, we changed the Continuous Integration (CI) service provider from TravisCI to GitHub Actions. This change allows us to run the tests free of charge, and also provides a more reliable and future-proof infrastructure.

Now that Shadow has been integrated into Sirepo-Bluesky, one may perform the same Bluesky scans as were formerly possible, but now with *either* SRW or Shadow as the underlying engine. Scans of interest include counts read from Sirepo's virtual "detector", step scans of a single parameter scanned over a range of values, and grid scans of multiple parameters over multiple ranges of values. In addition, the new integration of Shadow into Sirepo-Bluesky provides access to the Beam Statistics Report implemented in Sirepo-Shadow.

Because RadiaSoft personnel lack direct access to the BNL environment, but want to perform testing in as similar an environment as possible, we examined the possibility of using `sirepo.lib` (part of Sirepo) for working with Sirepo simulations in the absence of a server. Enhancements to the local development environment that better support Jupyterhub now allow developers to run Sirepo-Jupyter on their local machine for faster testing of new features in Sirepo-Bluesky.

[4] `https://github.com/NSLS-II/deep-beamline-simulation`
[5] `https://nsls-ii.github.io/bluesky/`
[6] `https://github.com/NSLS-II/sirepo-bluesky`

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-WEPV024

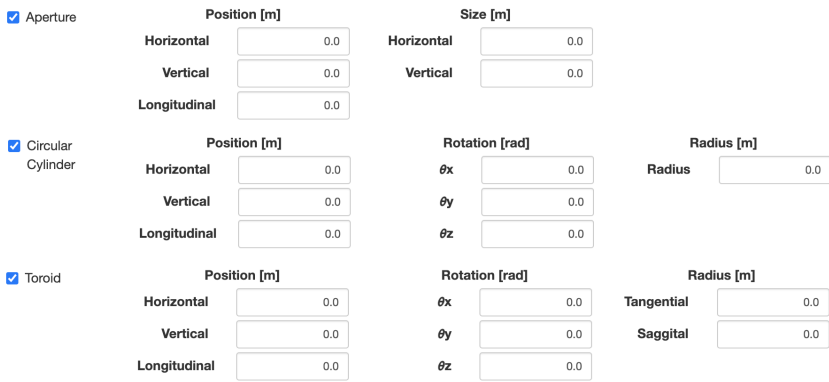

Figure 3: New beamline elements and their parameters for data generation. When the user provides a range of parameters, a script is produced that allows SRW to be run using `rsopt`, and will compute intensity data over the entire parameter range given.

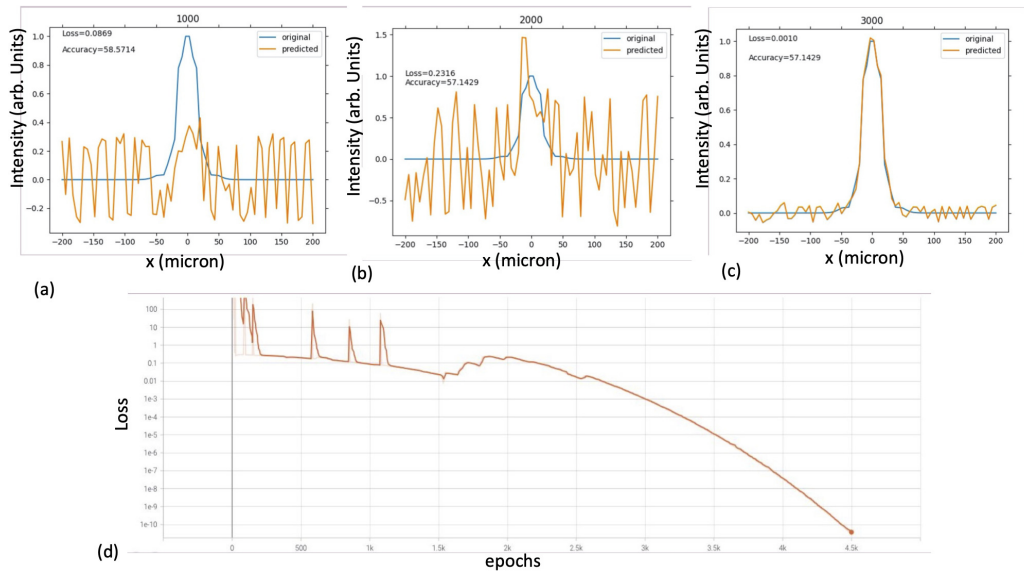

Figure 4: Tensorboard plots of loss function with varying epoch of (a)1000, (b)2000, and (c)3000. Gaussian radiation is produced in SRW, and a simple beamline with an aperture at 25 meters is varying, with the intensity observed at a watchpoint of 27 meters. (d) shows the accuracy as a function of all epochs during the learning process.

TES BEAMLINE MEASUREMENTS

Preliminary beam size measurements were carried out by Yonghua Du on the TES beamline, allowing us to begin comparison with the different types of propagation models. The TES beamline as set up in Sirepo SRW is show in figure citefig:TES.Sirepo.SRW. The layout for where the measurements were taken are shown in Figure 6. See [2] for more details about the layout and instrumentation of the TES beamline.

The beam size was measured at three positions: the fluorescent screen (FS) at ~ 50.242 m, the secondary slit aperture (SSA), and at the K-B mirror in the end station. The FS is mounted on the micrometer, which allows a measurement of the full horizontal beam size of ~ 1.5 mm. The beam size at the SSA was measured on the basis of SSA slit position and ion chamber (IC0) current. The full horizontal beam size is ~ 0.6 mm and vertical beam size is > 0.44 mm where ~ 50 % flux passed. The beam size at the K-B mirror was measured using the K-B step motor and the YAG crystal at the sample stage. The full horizontal beam size is ~ 1.1 mm and full vertical beam size is ~ 0.45 mm. During the measurement, we found that the vertical focusing point of the beam from toroidal mirror is between the KBV and sample position, not at the SSA. Comparison of these measurements to TES SRW simulations and moment propagation is ongoing, and improvements to the TES diagnostics are foreseen.

CONCLUSION

We have summarized our recent efforts to create reduced models suitable for an online model of x-ray beamlines and to allow access to these models integrated within the beamline control software, Bluesky. Working with the TES beam-

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-WEPV024

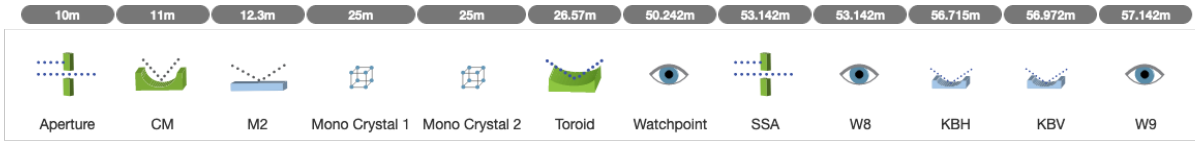

Figure 5: TES beamline in Sirepo SRW.

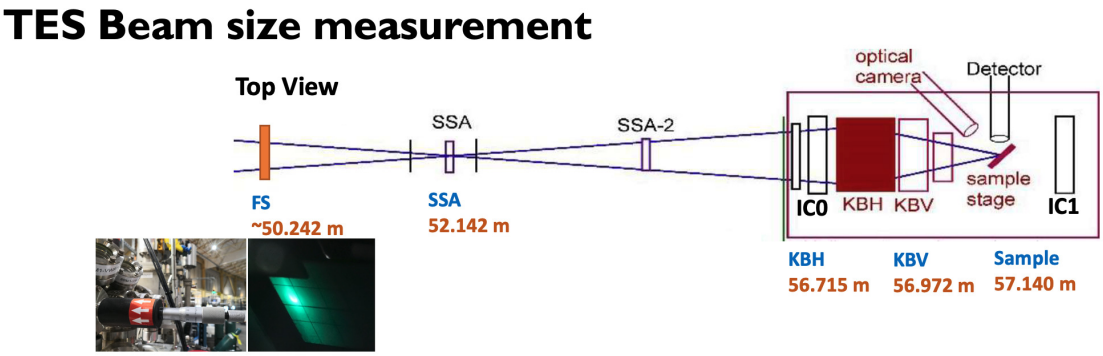

Figure 6: Layout for the TES beamsize measurements together with a photograph of the camera and resultant image on the fluorescent screen.

line, we are attempting a proof of principle demonstration of the creation of such an online model that will facilitate automated beamline control and reconfiguration.

ACKNOWLEDGMENTS

This work is supported, in part, by the US Department of Energy, Office of Science, Office of Nuclear Physics, including grant No. DE-SC0020593.

REFERENCES

[1] B. Nash, N. Goldring, J. Edelen, C. Federer, P. Moeller, and S. Webb, "Reduced model representation of x-ray transport suitable for beamline control," in *Advances in Computational Methods for X-Ray Optics V*, vol. 11493, International Society for Optics and Photonics, Aug. 2020, p. 114930C. DOI: 10.1117/12.2568187.

[2] P. Northrup, "The TES beamline (8-BM) at NSLS-II: Tender-energy spatially resolved X-ray absorption spectroscopy and X-ray fluorescence imaging," *J. Synchrotron Rad.*, vol. 26, no. 6, pp. 2064–2074, Nov. 2019. DOI: 10.1107/S1600577519012761.

[3] J. J. Healy, M. A. Kutay, H. M. Ozaktas, and J. T. Sheridan, Eds., *Linear Canonical Transforms: Theory and Applications*, ser. Springer Series in Optical Sciences. New York: Springer, 2016, vol. 198.

[4] B. M. Hennelly and J. T. Sheridan, "Fast numerical algorithm for the linear canonical transform," *J. Opt. Soc. Amer. A*, vol. 22, no. 5, pp. 928–937, 2005. DOI: 10.1364/JOSAA.22.000928.

[5] A. Koç, H. M. Ozaktas, C. Candan, and M. A. Kutay, "Digital computation of linear canonical transforms," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2383–2394, May 2008. DOI: 10.1109/TSP.2007.912890.

[6] A. Koç, H. M. Ozaktas, and L. Hesselink, "Fast and accurate algorithm for the computation of complex linear canonical transforms," *J. Opt. Soc. Amer. A*, vol. 27, no. 9, pp. 1896–1908, Sep. 2010. DOI: 10.1364/JOSAA.27.001896.

[7] J. J. Healy and J. T. Sheridan, "Fast linear canonical transforms," *J. Opt. Soc. Amer. A*, vol. 27, no. 1, pp. 21–30, Jan. 2010. DOI: 10.1364/JOSAA.27.000021.

[8] R. G. Campos and J. Figueroa, "A fast algorithm for the linear canonical transform," *Signal Process.*, vol. 91, no. 6, pp. 1444–1447, Jun. 2011. DOI: 10.1016/j.sigpro.2010.07.007.

[9] A. Koç, "Fast algorithms for digital computationof linear canonical transforms," PhD dissertation, Stanford University, Stanford, CA, Mar. 2011.

[10] S.-C. Pei and S.-G. Huang, "Fast discrete linear canonical transform based on CM-CC-CM decomposition and FFT," *IEEE Trans. Signal Process.*, vol. 64, no. 4, pp. 855–866, Feb. 2016. DOI: 10.1109/tsp.2015.2491891.

[11] B. Nash, N. Goldring, J. Edelen, S. Webb, and R. Celestre, "Propagation of partially coherent radiation using Wigner functions," *Phys. Rev. Accel. Beams*, vol. 24, no. 1, p. 010 702, Jan. 2021, ISSN: 2469-9888. DOI: 10.1103/PhysRevAccelBeams.24.010702.

[12] D. Allan, T. Caswell, S. Campbell, and M. Rakitin, "Bluesky's Ahead: A Multi-Facility Collaboration for an a la Carte Software Project for Data Acquisition and Management," *Synchrotron Radiation News*, vol. 32, no. 3, pp. 19–22, May 2019, ISSN: 0894-0886. DOI: 10.1080/08940886.2019.1608121.

[13] M. S. Rakitin *et al.*, "Introduction of the Sirepo-Bluesky interface and its application to the optimization problems," in *Advances in Computational Methods for X-Ray Optics V*, vol. 11493, International Society for Optics and Photonics, Aug. 2020, p. 1 149 311. DOI: 10.1117/12.2569000.