ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA120

COMPENSATION CONTROLS FOR AN ELLIPTICALLY POLARISING UNDULATOR

J. Willard*, T. Wilson, W. Wurtz
Canadian Light Source, Saskatoon, Saskatchewan, Canada

Abstract

At the Canadian Light Source (CLS) synchrotron, the addition of the Quantum Materials Spectroscopy Centre (QMSC) beamline requires the addition of an Elliptically Polarizing Undulator (EPU) insertion device to produce photons from the stored electron beam. Unlike the majority of such insertion devices, this EPU requires the production of photons of simultaneous arbitrary elliptical and linear phases, in addition to a range of energies. This EPU is also capable of creating perturbations of the stored electron beam sufficient to cause an interruption of an injection. In order to prevent this, compensation controls have been developed. These controls are accomplished with a combination of Experimental Physics and Industrial Control System (EPICS), mathematical models, and algorithms written in `C` and `MATLAB`.

INTRODUCTION

The Canadian Light Source (CLS) is a synchrotron radiation facility with an electron beam storage ring of 2.9 GeV. For the new Quantum Materials Spectroscopy Centre (QMSC) beamline, the Insertion Device (ID) is of the type Elliptically Polarizing Undulator (EPU) of an APPLE II manner, which uses four girders $Q_1 - Q_4$ as the frame for the magnetics as in Fig. 1. APPLE II EPUs are capable of creating photons both with elliptical polarization Φ_E and linear polarization Φ_L. However, in operation, generally only one of Φ_E and Φ_L are non-zero. This, combined with undulator gap, creates a system with two degrees of freedom.

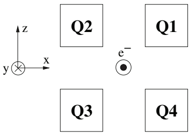

Figure 1: EPU Configuration.

The Dual-EPU for QMSC possesses two magnet arrays for producing photons of differing energies—effectively creating two EPUs which are shifted laterally, and must be used in a mutually exclusive manner—as depicted in Fig. 2. The high-energy array (EPU55) is not anticipated to pose significant issues. However, the low-energy array (EPU180) is capable of producing photons below 100 eV, for which beam optics can create distortions[1]. In order to compensate for this, EPU180 must be capable of generating photons of arbitrary superpositon of Φ_E and Φ_L, known as Universal Mode [1]. This results in a system with three degrees of freedom and additonal complexities.

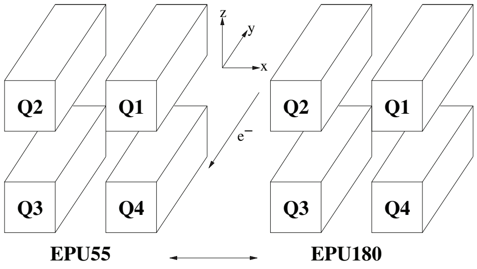

Figure 2: QMSC EPU Configuration.

In addition, the long spatial period of EPU180 also creates issues for the electron beam which may not be ignored. Tune shifts and a reduction in dynamic aperature would greatly reduce the ability of injected beams to join the stored beam. Current strips of the BESSY manner will be used in order to compensate for these effects [2].

Control Components

The QMSC EPU possesses twenty-four current strips for dynamic focusing, and four correction coils for compensation of the first and second field integrals. The compensation controls require the ability to set the current to each strip and coil individually. The differing compensations are:

- Universal Mode (Current Strips)
- Residual Focusing (Lattice Quadrupoles)[2]
- Skew Quadrupole (Current Strips)
- Normal Quadrupole (Current Strips)
- Multipole (Current Strips)
- Direct Entry (Current Strips)
- Field Integrals (Correction Coils)

The values that will be sent to each correction coil or current strip will then consist of a superposition of values from these differing compensations.

UNIVERSAL MODE

The Universal Mode is the most computationally taxing portion of the compensation controls, and it is anticipated that it will dominate the response time of the compensation controls. This software comprises two portions: software written in `MATLAB` to create a mathematical model of the magnetics of EPU180, and Universal Mode compensation

* juniper.willard@lightsource.ca

[1] The period of EPU55 is 55 mm, while the period of EPU180 is 180 mm.

[2] The Residual Focusing is not yet implemented, and will be necessary to correct the vertical tune shifting which the current strips are incapable of accomplishing.

software written in C—with the models generated by the former being used by the latter [3].

Overview

The purpose of the compensation software for Universal Mode is to calculate the necessary currents for the current strips throughout the use of EPU180. This is done via the calculation of a kickmap based upon the present position of the EPU and the mathematical model calculated earlier. The dominant equations for the kickmap with continuous variables are

$$\Phi(x, y, z) \equiv \left(\int_{-\infty}^{y} B_x(x, y', z)dy'\right)^2 + \left(\int_{-\infty}^{y} B_z(x, y', z)dy'\right)^2 \tag{1}$$

$$\begin{aligned} K_x(x, z) &= -\frac{\alpha^2}{2}\int_{-\infty}^{\infty}\frac{\partial\Phi}{\partial x}(x, y, z)dy \\ K_z(x, z) &= -\frac{\alpha^2}{2}\int_{-\infty}^{\infty}\frac{\partial\Phi}{\partial z}(x, y, z)dy \end{aligned} \tag{2}$$

where Φ is the focusing potential[3], and K is the kickmap as illustrated in [4]. The MATLAB Modeling Software furnishes the necessary integrated fields for each girder with discrete sampling along the axies of x_i, y_j, & z_k—at intervals of δ_x, δ_y, & δ_z respectively. A linear superposition of these four integrals will yield the field terms necessary for (1) with interpolation to the present EPU girder configuration. The resulting kickmap potential must then be integrated and derived to satisfy (2). These calculations must be recalculated continually.

Superposition & Interpolation

In the calculation of K_x of (2), the plane of $z = 0$ is used, while K_z is calculated upon the $z = \pm\delta_z$ planes. For the latter, the superposition of the integrated field on the $-\delta_k$ plane would be

$$\begin{aligned} I_x(x_i, y_j, \delta_z) = {} & I_{Q1,x}(x_i, y_j - Q_1, \delta_z - h_{12}) + \\ & I_{Q2,x}(x_i, y_j - Q_2, \delta_z - h_{12}) + \\ & I_{Q3,x}(x_i, y_j - Q_3, \delta_z + h_{34}) + \\ & I_{Q4,x}(x_i, y_j - Q_4, \delta_z + h_{34}) \end{aligned} \tag{3}$$

where each of the I terms represents an integral, each of the Q parameters represents the position of a girder along the y-axis, h_{12} represents the height of the Q_1-Q_2 girder pair, and h_{34} represents the height of the Q_3-Q_4 girder pair. Similar treatments are afforded to the $z = -\delta_z$ and the $z = 0$ planes. While the x co-ordinates remain on the x_i grid, the y and z co-ordinates are off the y_j and z_k grid—requiring two-dimensional interpolation of the form

$$\begin{aligned} f(a, b) = {} & (1 - a)(1 - b)f(0, 0) + a(1 - b)f(1, 0) + \\ & b(1 - a)f(0, 1) + abf(1, 1) \end{aligned} \tag{4}$$

which produces the integrals as seen in Table 1

Table 1: Integrated Kickmap Integral Terms

Horizontal Terms	**Vertical Terms**
$I_x(x_i, y_j, 0)$	$I_x(x_i, y_j, \pm\delta_z)$
$I_z(x_i, y_j, 0)$	$I_z(x_i, y_j, \pm\delta_z)$

and may be squared and summed as (1) to produce

$$\begin{aligned} \Phi(x_i, y_j, \delta_z) &= I_x(x_i, y_j, \delta_z)^2 + I_z(x_i, y_j, \delta_z)^2 \\ \Phi(x_i, y_j, 0) &= I_x(x_i, y_j, 0)^2 + I_z(x_i, y_j, 0)^2 \\ \Phi(x_i, y_j, -\delta_z) &= I_x(x_i, y_j, -\delta_z)^2 + I_z(x_i, y_j, -\delta_z)^2 \end{aligned} \tag{5}$$

Parallelisation Profiling of the initial revision of the software illustrated that ≈ 90% of the time elapsed in the calculation of the kickmap occured in the function for the above interpolation[4]. The independent nature of both the input and output of this function allow for this problem to be considered embarrassingly parallel. Due to the brevity of this function, the approaches of OpenMP and threadpooling were eschewed for independent iterators. Each iterator calls the interpolation function F four times in succession—where each call references a distinct and consecutive set of input data D from an array—and sums the results to an array element R as seen in Fig. 3.

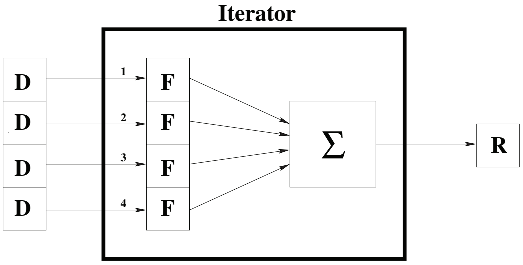

Figure 3: Kickmap Interpolation Independent Iterator.

Figure 4 illustrates each of the six iterators[5] I—executed on independent hardware threads—referencing the initial six sets of four data components D and the initial six results R. Due to the Hyper-Threading Architecture of Intel—which shares resources between Hyper-Threads on the same core—it is impossible for all independent iterators to conclude execution simultaneously [5].

Subsequently, if for example iterators 1, 4, & 6 complete prior to iterators 2, 3, & 5, they may then commence calculations on the subsequent datasets associated with them without delay as depicted in Fig. 5. In this manner, each of the six independent iterators may access the appropriate data to produce the appropriate results without conjoinment with any other iterators. Thus only one thread join is required once all interpolations have completed. This methodology

[3] The Φ of (1) is unrelated to either the Φ_E or Φ_L of the overal system.

[4] Using gprof with gcc of -O3 on an Intel Core i7-3770 at 3.40 GHz with four cores each containing two hardware threads.

[5] The parameters of the six iterators are found Table 1 and (5).

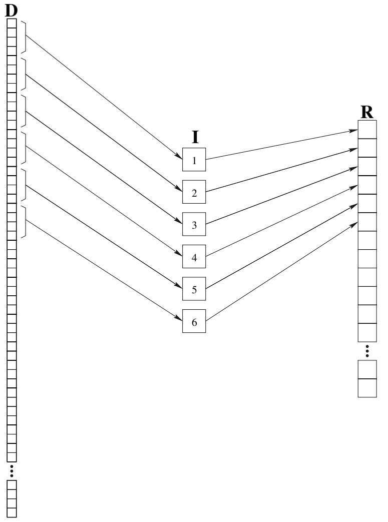

Figure 4: Initial Independent Iterators.

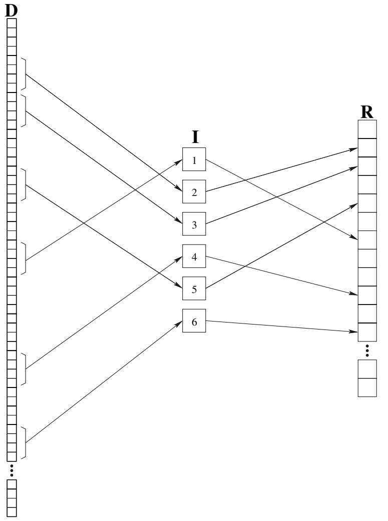

Figure 5: Subsequent Independent Iterators.

improved kickmap calculation by greater than 400% while testing the calculation time of 1000 kickmaps[6].

Integration & Differentiation

The kickmap potentials (5) may be integrated via Simpson's Method—with Trapezoidal Integration upon the ultimate point in the event of an even number of points—to produce the integrated kickmaps potentials

$$\begin{gathered}\int_{-\infty}^{\infty}\Phi(x_i, y, \delta_z)dy \\ \int_{-\infty}^{\infty}\Phi(x_i, y, 0)dy \\ \int_{-\infty}^{\infty}\Phi(x_i, y, -\delta_z)dy\end{gathered} \tag{6}$$

which are functions solely of x. The kickmaps may be calculated from (6) using the Centre-Point Method of differentiation, thus providing

$$\begin{aligned}K_x(x_i) &= -\frac{1}{4\delta_x}\left[\int_{-\infty}^{\infty}\Phi(x_{i+1}, y, 0)dy - \right. \\ &\qquad \left. \int_{-\infty}^{\infty}\Phi(x_{i-1}, y, 0)dy\right] \\ K_z(x_i) &= -\frac{1}{4\delta_x}\left[\int_{-\infty}^{\infty}\Phi(x_i, y, \delta_z)dy - \right. \\ &\qquad \left. \int_{-\infty}^{\infty}\Phi(x_i, y, -\delta_z)dy\right]\end{aligned} \tag{7}$$

For the special cases of $K(x_0)$ and $K(x_{n_x-1})$, the less desired Side-Biased Method of differentiation must be used.

Current Strip Currents

The general equations for the current strip responses are

$$\begin{aligned}J_{m,x}(x, z) &= \frac{1}{I_m}\int_{-\infty}^{\infty}B_{m,x}(x, y, z)dy \\ J_{m,z}(x, z) &= \frac{1}{I_m}\int_{-\infty}^{\infty}B_{m,z}(x, y, z)dy\end{aligned} \tag{8}$$

for an individual current strip at index m, and I_m—the current of said strip[7]. I_m may be calculated from (7, 8) with a linear series of equations

$$\begin{bmatrix}\alpha K_x(x_i) \\ \alpha K_z(x_i)\end{bmatrix} = \begin{bmatrix}J_{1,z}(x_i, 0) & \cdots & J_{24,z}(x_i, 0) \\ -J_{1,x}(x_i, 0) & \cdots & -J_{24,x}(x_i, 0)\end{bmatrix}\begin{bmatrix}I_1 \\ \vdots \\ I_{24}\end{bmatrix} \tag{9}$$

where $\alpha[1/Tm] = \frac{0.299792458}{E[GeV]}$ is the magnetic rigidity[8]. The pseudo-inverse of the J matrix from (9) has been previously calculated using `MATLAB`, and thus I_m may be obtained by simple matrix multiplication.

CALIBRATION TABLES

In addition to Universal Mode, there is the requirement for calibration tables for both the correction coils and the current strips. These tables are created with `MATLAB` scripts from measurements of the EPU behaviour. During operation, interpolation is done for the desired point from those points where measurements occured during calibration. The previous file storage of these calibration tables was that of space-delineated values. While this functioned well for two-dimensional tables, the three degrees of freedom for EPU180 necessitates three-dimensional tables—while the EPU55 will only require two-dimensional tables. An XML format was selected for these files, which allowed for expandable meta-data and inconsistent point spacing. The schema shown in Fig. 6 allows for differing values of rows

[6] In comparison to a single-threaded application

[7] The indices commence with the upper-inboard strip, and conclude with the lower-outbound strip.

[8] The storage ring energy at the CLS is 2.9GeV. It is not anticipated that a time-correlated value of this will be used.

and columns—and thus allowing for regions of interest to possess smaller quantisation. The two-dimensional tables for EPU55 are fitted into the three-dimensional paradigm with placeholder values for the unused positions.

```
<QMSC_Table version="0.2">
  <table_info>
    <filename>output.xml</filename>
    <date>28-Jun-2017 14:09:38</date>
  </table_info>
  <table>
    <phi_l_page table_date="05-1-2015 11:11" table_name="test1" value="1">
      <gap_row value="1">
        <phi_e_col value="1">1.11</phi_e_col>
        <phi_e_col value="2">1.12</phi_e_col>
        <phi_e_col value="3">1.13</phi_e_col>
      </gap_row>
      <gap_row value="2">
        <phi_e_col value="1">1.21</phi_e_col>
        <phi_e_col value="2">1.22</phi_e_col>
        <phi_e_col value="3">1.23</phi_e_col>
      </gap_row>
    </phi_l_page>
    <phi_l_page table_date="05-1-2015 11:11" table_name="test2" value="2">
                                         ⋮
    </phi_l_page>
</QMSC_Table>
```

Figure 6: XML Schema.

To reduce disruptions of the calibration routines, it was decided that the calibration `MATLAB` scripts would continue to create the space-delineated files, and another `MATLAB` script would splice these together to create the required XML files.

XML Parsing To parse these XML files, a library was written in `C` using `libxml2`. The XML data is parsed into a tree and traversed for the relevant information, while some sanity and relevancy checking is perfomed. Once parsed, the relevant information is stored as a three-dimensional array in memory—of Φ_L pages, *Gap* rows, and Φ_E columns—with associated meta-data stored in a structure.

In use, the vertices encompassing the desired point P are sought using binary searching for adjacent pages—forming planes $V_1 - V_4$ and $V_5 - V_8$—and subsequently for adjacent rows and columns within said pages. The result of this is eight points which surround the desired point as illustrated in Fig. 7.

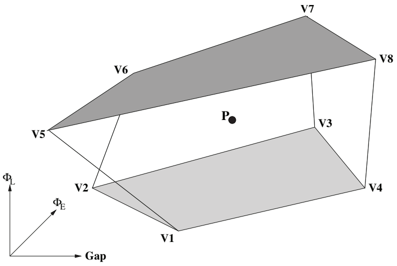

Figure 7: XML Search for Point.

Search Validation In order to validate search results, the Point in Polygon method is used. Ray casting is accomplished along the first dimension, and edge crossings are summed using successive application of boolean XOR. In the event that the sum is odd, the point is contained in the polygon. In the event that the sum is even, the point is exterior to the polygon—as depicted in Fig. 8. The Point in Polygon algorithm also identifies opposing edges required for interpolation. With Point in Polygon, a projection of the point upon each of the six polygon sides of the polyhedron contained in Fig. 7 will demonstrate if the point is contained within the polyhedron.

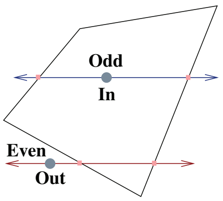

Figure 8: Point-in-Polygon.

Interpolation Once the points surrounding the desired point are located, interpolation for the value at the desired point P may commence. Due to the potential for enclosures to fail to be orthogonal, a weighted average method of interpolation was discarded in favour of trilinear interpolation. Successive interpolations are enacted to accomplish bilinear interpolation on two oposing faces, while linear interpolation is conducted on the results as demonstrated in Fig. 9. In each of these interpolations, it is not only the three dimensions which must be interpolated, but the associated values must also be. Initially, point C_{12} is interpolated from V_1 and V_2—likewise C_{34} is interpolated from V_3 and V_4. Subsequently, C_1 may be interpolated from C_{12} and C_{34}. In a similar manner, C_2 is arrived upon from V_5 - V_8 via C_{56} and C_{78}. Finally, the desired point P may be interpolated from C_1 and C_2.

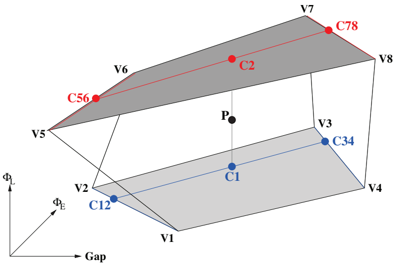

Figure 9: Trilinear Interpolation.

COMPONENT INTEGRATION

In order to integrate the various values from differing compensation paradigms of the correction coils and current strips, matrices are used, with one matrix for the correction coils, and another for the current strips of each of the EPU arrays. This is accomplished within Experimental Physics and Industrial Control System (EPICS) which provides Process Variables (PVs) with which other systems may interact [6].

Within the EPICS application, each element of the compensation controls is controlled by a State Notation Language (`SNL`) program which provides a state machine of predictable and repeatable behaviour. As each `SNL` program compiles to `C`, functions and libraries written in `C` are easily integrated. The aforemented Universal Mode and Calibration Table Software are integrated in such a manner. Once the values are produced by these software, a superposition of these values is required prior to directing currents to individual correction coils and current strips.

Correction Coils

The correction coil superposition takes the form of

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} K_H^{up} \\ K_H^{dn} \\ O_H \\ K_V^{up} \\ K_V^{dn} \\ O_V \end{bmatrix} = \begin{bmatrix} I_H^{up} \\ I_H^{dn} \\ I_V^{up} \\ I_V^{dn} \end{bmatrix} \tag{10}$$

where the elements for both the upstream and downstream correction coils are found in Table 2.

Table 2: Correction Coil Superposition Elements

	Horizontal		Vertical	
Element	**Up**	**Down**	**Up**	**Down**
Kick Value	K_H^{up}	K_H^{dn}	K_V^{up}	K_V^{dn}
Offset	O_H		O_V	
Current	I_H^{up}	I_H^{dn}	I_V^{up}	I_V^{dn}

Each of I_H^{up}, I_H^{dn}, I_V^{up}, and I_V^{dn} are sent to the corresponding power supplies for the correction coils [7].

Current Strips

For the current strips, several differing quantities are inputs:

- Quadrupole Normal
- Quadrupole Skew
- Multipole Arbitrary
- Universal
- Direct

The Quadrupole Normal is the field required for locating the centre of the current strips. Since the correction coils for the QMSC EPU are unable to apply a Skew Quadrupole field, the current strips are required for this purpose [8]. The Multipole Arbitrary Field will allow for an arbitrary field to be enacted for physics studies, an example of which would be an octopole magnetic field. The Direct Field will allow an operator to input arbitrary values manually. If each V were a coefficient that an operator may define, U represented the currents calculated for Universal Mode, M were the values for Multipole Mode, D were entered by the operator, and I were the currents for each of the current strips—and assuming an array of current strips of ideal charactaristics and allignment—the superposition for the current strip currents would be [9]

$$\begin{bmatrix} U_1 & -1.00 & 0.815 & M_1 & D_1 \\ U_2 & -0.31 & 0.824 & M_2 & D_2 \\ U_3 & -0.32 & 0.900 & M_3 & D_3 \\ U_4 & -0.19 & 0.951 & M_4 & D_4 \\ U_5 & -0.13 & 0.984 & M_5 & D_5 \\ U_6 & -0.04 & 1.000 & M_6 & D_6 \\ U_7 & 0.04 & 1.000 & M_7 & D_7 \\ U_8 & 0.13 & 0.984 & M_8 & D_8 \\ U_9 & 0.19 & 0.951 & M_9 & D_9 \\ U_{10} & 0.32 & 0.900 & M_{10} & D_{10} \\ U_{11} & 0.31 & 0.824 & M_{11} & D_{11} \\ U_{12} & 1.00 & 0.815 & M_{12} & D_{12} \\ U_{13} & 1.00 & 0.815 & M_{13} & D_{13} \\ U_{14} & 0.31 & 0.824 & M_{14} & D_{14} \\ U_{15} & 0.32 & 0.900 & M_{15} & D_{15} \\ U_{16} & 0.19 & 0.951 & M_{16} & D_{16} \\ U_{17} & 0.13 & 0.984 & M_{17} & D_{17} \\ U_{18} & 0.04 & 1.000 & M_{18} & D_{18} \\ U_{19} & -0.04 & 1.000 & M_{19} & D_{19} \\ U_{20} & -0.13 & 0.984 & M_{20} & D_{20} \\ U_{21} & -0.19 & 0.951 & M_{21} & D_{21} \\ U_{22} & -0.32 & 0.900 & M_{22} & D_{22} \\ U_{23} & -0.31 & 0.824 & M_{23} & D_{23} \\ U_{24} & -1.00 & 0.815 & M_{24} & D_{24} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ V_{skew} \\ V_{norm} \\ V_{multi} \\ 1 \end{bmatrix} = \begin{bmatrix} I_{strip1} \\ I_{strip2} \\ I_{strip3} \\ I_{strip4} \\ I_{strip5} \\ I_{strip6} \\ I_{strip7} \\ I_{strip8} \\ I_{strip9} \\ I_{strip10} \\ I_{strip11} \\ I_{strip12} \\ I_{strip13} \\ I_{strip14} \\ I_{strip15} \\ I_{strip16} \\ I_{strip17} \\ I_{strip18} \\ I_{strip19} \\ I_{strip20} \\ I_{strip21} \\ I_{strip22} \\ I_{strip23} \\ I_{strip24} \end{bmatrix} \tag{11}$$

ACKNOWLEDGEMENT

The CLS is supported by CFI, NSERC, NRC, CIHR, the Province of Saskatchewan, WD, and the University of Saskatchewan.

REFERENCES

[1] J. Bahrdt, G. Wüstefeld, “Symplectic tracking and compensation of dynmic field integrals in complex undulator structures”, Phys. Rev. ST AB **14**, 040703 (2011).

[2] W. Wurtz, D. Bertwistle, M. Sigrist, “Simulation of a long-period EPU operating in universal mode at the Canadian light source”, in *Proc. IPAC’14*, Dresden, Germany, June 2014, paper WEPRO024, pp 1995-7.

[3] W. Wurtz, “Manual for EPU control system physics software: universal”, CLS Internal Document (2014).

[4] P. Elleaume, “A new approach to the electron beam dynamics in undulators and wigglers”, in *Proc. EPAC’92*, Berlin, Germany, March 1992, pp 661-3.

[5] D. Marr, et. al, “Hyper-threading technology architecture and microarchitecture”, *Intel Tech. J.*, **6** (1).

[6] EPICS Framework, Version 3.14.12, Computer Software, `http://www.aps.anl.gov/epics`

[7] C. Baribeau, M. Sigrist, “QMSC insertion device commissioning plan”, CLS Document 35.12.90.1 (2017).

[8] W. Wurtz, “Beam dynamics simulations with QMSC EPU180 operating in universal mode”, CLS Document 5.18.25.5 (2013).

[9] W. Wurtz, “Compensating for the effects of the QMSC Dual-EPU on the electron beam”, CLS Document 5.18.25.8 (2015).