ISBN: 978-3-95450-265-3 doi:10.18429/JACoW-SAP2023-MOPB019

DEVELOPMENT STATUS OF BEAM DYNAMICS SOFTWARE APES FOR CEPC*

Weibin Liu[†1], Yuan Zhang[1], Tianmu Xin, Zhe Duan, Zhiyuan Li[1],
Yaliang Zhao, Huiping Geng, Mengyu Su[1], Yixian Dai[1], Xiaohan Lu, IHEP, Beijing, China
[1]also at UCAS, Beijing, China

Abstract

The physical design and beam dynamics study of the Circular Electron Positron Collider (CEPC) is an unprecedented challenge. In the simulation studies to evaluate its performance limitations and mitigation, the cross-talk between many physical phenomena must be properly modelled, including the crab-waist collision scheme with a large Piwinski angle, strong nonlinear effects, the energy sawtooth, beam-beam interactions, machine impedances, etc. To address this challenge, a software project APES was proposed in 2021 and received support from the IHEP Innovative Fund in 2022. The progress of the APES project are described in this paper.

INTRODUCTION

The CEPC was first proposed as a circular Higgs factory in 2012 and the conceptual design report was published in 2018 [1]. FCC-ee is a similar project proposed in CERN. During the design of these future colliders, the beam lifetime limitation due to the beamstrahlung effect, the synchrotron radiation induced by collision leading to a bunch lengthening and an increase in the beam energy spread, has been found and studied [2]. Different from conventional colliders, not only the transverse beam size, but also the longitudinal dynamics would be clearly influenced by the collision. That is why the 3D flip-flop instability may appear in CEPC or FCC-ee [3].

There would also be strong synchrotron radiation in the arc bending magnets of the machines, leading to a substantial "sawtooth"-shape variation of the central beam energy along the ring, which is the so-called sawtooth effect. The magnet tapering method has been proposed to mitigate this effect [4]. This requires new optics calculation method to consider the energy change along the ring.

After the crab-waist scheme was proposed around 2006 [5], the new collision scheme has become the baseline design for the following high performance circular e+e- colliders. However sub-millimeter scale β_y^* in future machines would induce strong lattice non-linearity and a very small dynamic aperture. During the lattice design and optimization, it has been found that the short-term dynamic aperture tracking could not predict the long-term beam lifetime [6].

In recent years, a new horizontal coherent beam-beam instability (X-Z instability) has been found [7]. The following simulation and analysis also show that the potential-well distortion effect would impact the behaviour of X-Z instability clearly [8, 9] . This tells us the cross-talk between beam-beam and longitudinal impedance could not be ignored.

Apart from these novel effects uncovered in the design study of CEPC and FCCee, the commissioning of SuperKEKB also reveal that there is still a gap in the modelings and simulations to explain and mitigate the difficulties in the practical machine tunings [10]. These have told us that the beam dynamics of future e+e- colliders would be very challenging.

A beam dynamics software project "APES" (Accelerator Physics Emulation System) has been proposed, with an objective to address the beam physics issues in CEPC in a unified, extendable manner:

- The modeling of the collider, especially the complicated interaction region and the cross-talk of common hardware shared by the two rings.
- Lattice design and performance evaluation, including symplectic tracking, optics calculation/matching, emittance calculation, modeling of the sawtooth and tapering effect, analysis of nonlinear optics performance, spin dynamics evaluation, machine error effects and correction algorithms, multi-objective optimization etc.
- Performance evaluation/prediction of the collider, the cross-talk between realistic lattice, beam-beam interaction, spin, collective effects and necessary hardware modeling.
- Interface with detector (MDI) and machine protection (particle-matter interaction), easy access and interaction with software dedicated for these purposes.

These functionalities and features will empower users to design, analyze, and optimize accelerator systems with greater accuracy and efficiency. Additionally, we can foresee several potential applications when these capabilities have been fully developed:

- Tracking simulations to evaluate the luminosity and beam lifetime with the realistic lattice and the strong-strong beam-beam interaction, as well as impedance effects.
- Simulations of the injection process, to evaluate the injection efficiency in the presence of machine imperfections, beam-beam interaction and impedance effects, and to prepare beam loss information for the design of collimators and evaluation of the influence to the detectors.

* Work supported by IHEP Innovative Fund
† liuwb@ihep.ac.cn

ISBN: 978-3-95450-265-3 doi:10.18429/JACoW-SAP2023-MOPB019

SOFTWARE PROGRESS

As mentioned earlier, APES is a comprehensive accelerator simulation software that will entail intricate accelerator modeling, intensive computation, and extensive data processing. Additionally, we will face challenges such as integrating various software, facilitating cross-platform usage, and enabling collaborative development among multiple individuals. Python, being a rapidly growing programming language in recent years, offers numerous advantages for coding and future development. It is known for its ease of learning, versatility, powerful standard library, rich ecosystem of third-party libraries and frameworks, and cross-platform compatibility. Moreover, there has been a significant trend of using Python in both domestic and international accelerator software development this year. Therefore, we have chosen Python as the foundation for our software development.

Framework of APES

In the APES package, a foundational set of classes has been constructed to form the core of the entire framework. The hierarchical structure of APES is depicted in Figure 1. This section primarily focuses on elucidating the fundamental logical pathway that underpins the design of this framework. For further details regarding the technological implementation and hierarchical relationships among classes please refer to related chapters in the code manual.

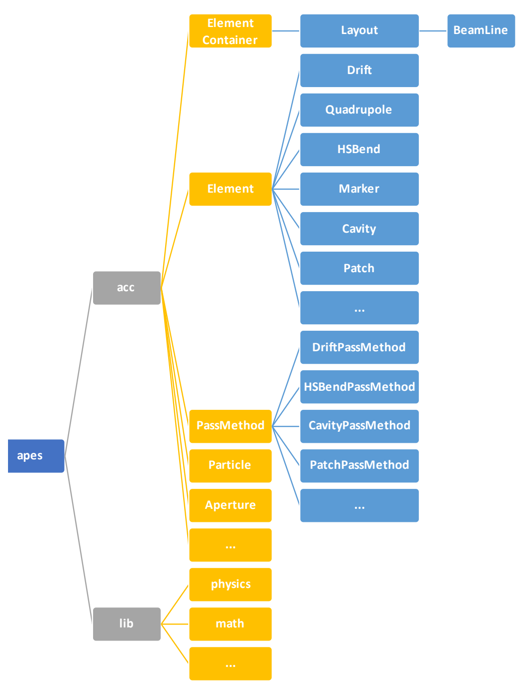

Figure 1: APES Hierarchical Structure Diagram.

Local Frame Within an accelerator, as a particle traverses through various components, it encounters fields in the local frame of each element. Consequently, the dynamics of the particle are articulated within these local frames. Presently, in APES, each element is associated with two reference frames - the entrance frame and the exit frame (Figure 2).

Global Frame As the name suggests, the Global Frame serves as a reference for delineating the relative positions of various entities, such as elements and particles, within the overarching framework of the accelerator (Figure 2).

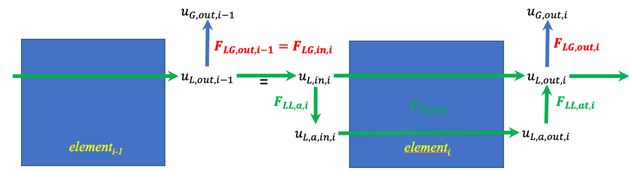

Figure 2: Element Coordinate Transformation Diagram.
u: coordinates
F: coordinate transformation
i: element index
L: Local frame
G: Global frame
in: at the entrance of the element
out: at the exit of the element
LG: from local frame to global frame
io: from the entrance to the exit of the element
ta: from ideal frame to actual frame
at: from actual frame to ideal frame

Patch To facilitate tracking of a particle as it navigates through individual elements of the accelerator, geometric transformations between different frames are imperative. In APES, these transformations are termed as "Patches". A Patch is essentially an affine transformation applied to the particle's coordinates, comprising translation and rotation components. Typically, each element is equipped with at least two Patches – one that addresses the transformation from the exit frame of the preceding element to its own entrance frame, and another to handle the transformation between its exit frame and the next element entrance frame. To enhance practicality, distinct Patches are employed to describe transformations between frames arising from different sources, such as frame shifts due to specialized installations or misalignments due to installation errors.

Element The concept of the Element has been mentioned but yet to be defined. In the code of APES, an element is a base class that contains some universally useful attributes and methods when deriving specific type of element from it. At this stage, the element class contain its own set of patch generator. Namely it can handle the transformation between the ideal and real entrance local frame, the one between the entrance and exit frame, and the one between the real and ideal exit frame. Together with the transfer map of each subclasses derived from the element class, the chain of patches and maps will form the main stream of operations the particles need to go trough during the tracking process.

ISBN: 978-3-95450-265-3 doi:10.18429/JACoW-SAP2023-MOPB019

Element Container For the construction of the accelerator within the Global Frame, an additional class termed "Element Container" is employed to augment the capabilities of the basic Element class. This grants users greater control and flexibility in manipulating elements (e.g., modifying surveys, scaling strengths, or altering species and directionality). The Element Container class exhibits three distinct functionalities. First, it maintains a table of "Instances" and acts as a "Container". Instances can be a composite of elements, element containers, or patches that represent groups of entities assembled for various purposes. Secondly, it employs Patches to establish bidirectional connections between the Global Frame and the Element's Local Frame. Lastly, the Element Container provides users with the ability to manipulate various parameters of the elements encompassed within its scope. Figure 3 portrays a conceptual illustration of the Element Container using a simplified cell to depict how a dipole element is encapsulated within its container. In basic applications, the containerization is largely automated, minimizing the necessity for user intervention

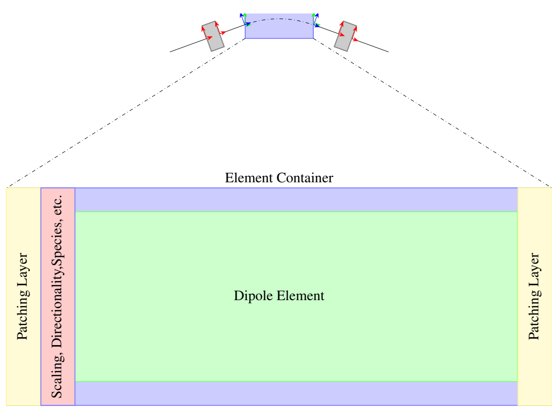

Figure 3: Cartoon illustration to the concept of element and element container.

Layout As a subclass derived from the ElementContainer class, an additional class called "Layout" has been established. The Layout class is designed with two primary objectives in mind. The first objective is to compute the survey information of the elements within the Global Frame. The second objective involves generating an "element table" which will be indispensable for future particle tracking.

To construct the accelerator, the current methodology involves adding entities to the "table" attribute of a Layout object. These entities can be elements, element containers, or patches, which can be added using the "append" or "insert" methods. Python's reference passing strategy proves to be particularly advantageous in this context, as it enables the seamless addition of the same instance to different sections of the layout, while managing their relative positions through patches.

BeamLine The "BeamLine" class plays a critical role in encapsulating the information extracted from Layout.table. Through the BeamLine class, a streamlined and concise table is produced, from which a sequence of maps is generated. These maps are integral to the process as they guide the path that particles will follow during the tracking phase.

In essence, the BeamLine class acts as a repository for the refined information drawn from the Layout class, ensuring that only the pertinent data required for particle tracking is retained in an organized manner. This, in turn, facilitates a more efficient and streamlined tracking process.

Some commonly used calculations are encapsulated in the BeamLine calss, such as Jacobian Matrix calculation, closed orbit calculation, Twiss calculation, etc. And more functionalities will be added soon.

PassMethod

The "PassMethod" class serves as the fundamental class for calculations, with specific subclasses created for each component type, including DriftPassMethod, QuadrupolePassMethod, HBendPassMethod, PatchPassMethod and others. These subclasses assign tracking maps for each element type. During optics calculations or particle tracking, a list of PassMethod instances is automatically generated. In principle, the pass method for any element can be freely configured.

Tracking Map of APES

The design of colliders requires accurate symplectic map of magnets to ensure right beam dynamics result. Due to very small β_y^*, the non-linearity of drift must be considered in modern/future high-performance e+ e- colliders. Nonlinear Maxwellian fringe field of magnets should also be considered. Currently, the program has implemented maps of drifts, dipoles, quadrupoles, sextupoles, octupoles, RF-cavities, solenoids, multipoles and more referring to the method of SAD [11]. It supports the modeling of hard and soft edge fringe fields of magnetic elements, tilted strong solenoids for particle detection in interaction regions (IRs), and the handling of combined solenoid/multipole elements. What's more, there is a optional map for the patch according to Etienne Forest. [12]. Initial benchmark tests comparing with SAD have proved the high accuracy of particle tracking using SuperKEKB lattice. The synchrotron radiation effect has not been included so far.

Currently the tracking map section is implemented in C language for optimized computational speed, building upon previous work. It is utilized through efficient function calls.

Beam Optics Calculation

The APES package has recently been enhanced with several important functionalities. These additions encompass various capabilities, including closed orbit calculation for a ring, Jacobian Matrix calculation, Twiss parameters calculation, and emittance calculation, among others. Additionally, there are ongoing developments for some important features.

ISBN: 978-3-95450-265-3   doi:10.18429/JACoW-SAP2023-MOPB019

For instance, there are two method for emittance calculation. One is the SLIM method published by Alexander W. Chao [13] and another one is the envelope matrix method published by K. Ohmi [14]. Specially for SLIM, we found that the dependence on transverse coordinates of radiation emitted in dipole was not mentioned in the paper, and this effect would be significant when some parameters are unusual. After modification, the emittance and the damping rate given by SLIM and by the envelope matrix method are reasonable compared with the exiting code, such as SAD.

APPLICATION EXAMPLES

We have completed the main components of the APES Framework and have added several beam physics calculation functionalities. Below are some examples of using APES.

Layout Modeling

Creating a layout is the initial step in simulating an accelerator. The following code demonstrates the process of defining elements and creating the layout for BEPCII BPR. The Affine transformation startAT establishes the initial coordinate system direction for the BPRLayout. This transformation is utilized to calculate the BPR survey. Subsequently, some elements are defined. BPRList stores all the elements in a specific order and is employed to define the BPRLayout. Figure 4 displays a portion of the element information and survey data for BPR, while Figure 5 shows the layout plotted using APES based on the survey data.

```
from apes.acc.Layout import *
from apes.acc.Element import *
from apes.acc.Particle import *
from apes.lib.math.AffineTransformation import *
...
...
initRotate = -0.011
startAT = AT.createAT(azimuth=initRotate)

IP = Marker('IP')
R4CBPM00 = Marker('R4CBPM00')

D4I01 = Drift('D4I01', length=0.539)
D4I01P = Drift('D4I01P', length=0.370)
...
PSCQ4I=Patch.createPatch('PSCQ4I', azimuth=
    initRotate, dx=0.0099988)
...
SSCQI = Quadrupole('SSCQI', length=0.59989336,
    k1=-1.073308053)
SSCQO = Quadrupole('SSCQO', length=0.59989335,
    k1=-1.073308053)
...
R4ISPB = HSBend('R4ISPB', length=0.600669, angle
    =
-0.03951362, anglein=-0.01975681, angleout
    =-0.01975681)
R4IMB = HSBend('R4IMB', length=1.034466, angle=
-0.113089342, anglein=-0.056544671, angleout
    =-0.056544671)
...
BPRList = [IP, D4I01, R4CBPM00, D4I01P, PSCQ4I,
    SCQI, ...]

BPRLayout = Layout('BPR', entryLGCT=startAT,
    elementList=BPRList)
surveytable = BPRLayout.getSurveyTable(level=1)
print(surveytable)
```

	Name	Type	Length	Xout	Yout	Zout	Pitchout	Yawout	Rollout	Element
0	__BPR_Start__	Marker	0.000	0.000	0.000	0.000	-0.000	-0.011	-0.000	<apes.acc.Element.Marker object at 0x7f87283ad...
1	IP	Marker	0.000	0.000	0.000	0.000	-0.000	-0.011	-0.000	<apes.acc.Element.Marker object at 0x7f8725a50...
2	D4I01	Drift	0.539	0.006	0.000	0.539	-0.000	-0.011	-0.000	<apes.acc.Element.Drift object at 0x7f872594c668>
3	R4CBPM00	Marker	0.000	0.006	0.000	0.539	-0.000	-0.011	-0.000	<apes.acc.Element.Marker object at 0x7f8725a60...
4	D4I01P	Drift	0.370	0.010	0.000	0.909	-0.000	-0.011	-0.000	<apes.acc.Element.Drift object at 0x7f8724341908>
5	P_SCQ_4I	Patch	0.000	-0.000	0.000	0.909	-0.000	0.000	-0.000	<apes.acc.Element.Patch object at 0x7f87256c8da0>
6	SSCQI	Quadrupole	0.600	-0.000	0.000	1.509	-0.000	0.000	-0.000	<apes.acc.Element.Quadrupole object at 0x7f872...
7	P_SCQ_4O	Patch	0.000	0.021	0.000	1.509	-0.000	-0.026	-0.000	<apes.acc.Element.Patch object at 0x7f872170af60>
8	D4I02	Drift	0.641	0.038	0.000	2.150	-0.000	-0.026	-0.000	<apes.acc.Element.Drift object at 0x7f872594c828>
9	R4IBPM00	Marker	0.000	0.038	0.000	2.150	-0.000	-0.026	-0.000	<apes.acc.Element.Marker object at 0x7f8725a60...
10	D4I02P	Drift	0.150	0.042	0.000	2.300	-0.000	-0.026	-0.000	<apes.acc.Element.Drift object at 0x7f872594c978>
11	R4ISPB	HSBend	0.601	0.070	0.000	2.900	-0.000	-0.066	-0.000	<apes.acc.Element.HSBend object at 0x7f8720e77...
12	D4I03	Drift	0.624	0.111	0.000	3.523	-0.000	-0.066	-0.000	<apes.acc.Element.Drift object at 0x7f872594cac8>
13	R4IQ1A	Quadrupole	0.254	0.127	0.000	3.776	-0.000	-0.066	-0.000	<apes.acc.Element.Quadrupole object at 0x7f872...
14	D4I04	Drift	0.174	0.139	0.000	3.950	-0.000	-0.066	-0.000	<apes.acc.Element.Drift object at 0x7f872594cc18>
15	R4IBPM01	Marker	0.000	0.139	0.000	3.950	-0.000	-0.066	-0.000	<apes.acc.Element.Marker object at 0x7f8725a60...
16	D4I05	Drift	0.068	0.143	0.000	4.018	-0.000	-0.066	-0.000	<apes.acc.Element.Drift object at 0x7f872594cd68>
17	R4IQ1B	Quadrupole	0.464	0.174	0.000	4.481	-0.000	-0.066	-0.000	<apes.acc.Element.Quadrupole object at 0x7f872...
18	D4I06	Drift	1.221	0.254	0.000	5.700	-0.000	-0.066	-0.000	<apes.acc.Element.Drift object at 0x7f872594ceb8>
19	R4IBV02	Drift	0.000	0.254	0.000	5.700	-0.000	-0.066	-0.000	<apes.acc.Element.Drift object at 0x7f8725992ac8>

Figure 4: Layout Table of BEPCII BPR.

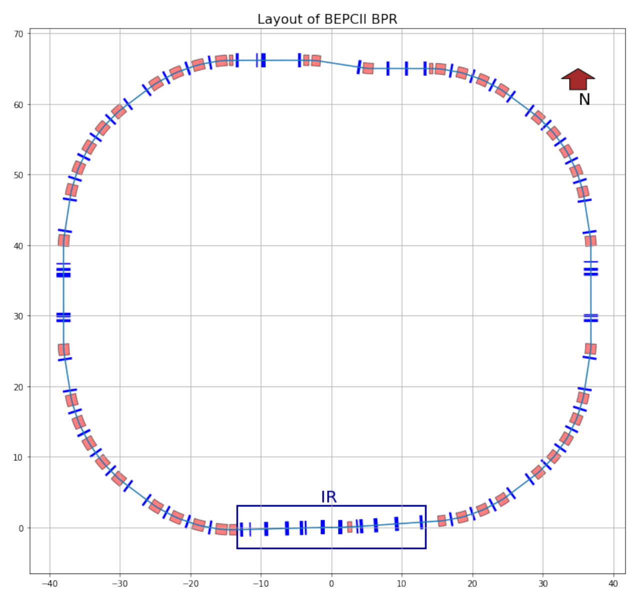

Figure 5: Layout of BEPCII BPR.

Modeling for Special Cases

With the assistance of Patch, modeling certain special cases becomes much simpler. Here is an example of modeling interaction region of Beijing Elctron Positron Collider (BEPCII). BEPCII is an asymmetric double-ring collider. The layout of positron ring (BPR) is shown in Figure 5. The electron and positron beames have a cross angle at the interaction region. There are superconducting quadrupoles (SCQs) on each side of the interaction point. Electron and positron beams pass SCQ off centre. The traditional method for dealing with this situation involves treating SCQ as a combined magnet of bend and quadrupole, which can be inconvenient for creating a lattice. However, with the introduction of APES, a new approach has been implemented

ISBN: 978-3-95450-265-3   doi:10.18429/JACoW-SAP2023-MOPB019

(Figure 6). We added one patch element before the SCQ and another after the SCQ to realize the the layout and lattice. When positron beam leaving the interaction point, the beam has a 11 mrad horizontal angle before entering the SCQ. The first patch element rotates the beam coordinate system by -11 mrad and then shifts it horizontally towards the center line of SCQ. This converts the coordinates of the positron particles to the SCQ coordinate system. The beam then passed through SCQ quadrupole. At the exit of SCQ, the second patch element shifts the coordinate system origin to the beam exit position on the exit plane of the SCQ. Then the coordinate system is rotated horizontally to align the longitudinal axis with the center line of the next element. By implementing these patch elements, the inconvenience of creating a lattice with the SCQs can be overcome.

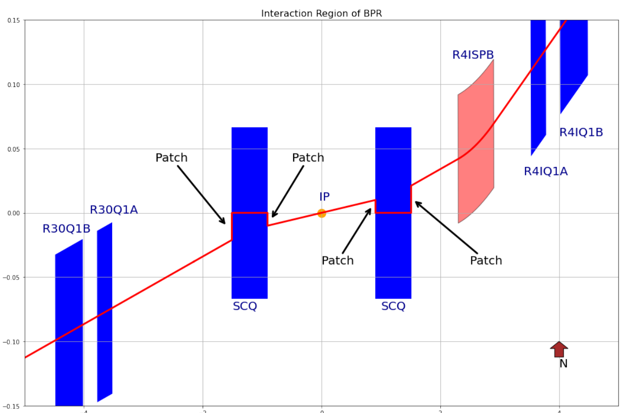

Figure 6: Interaction Region of BEPCII BPR.

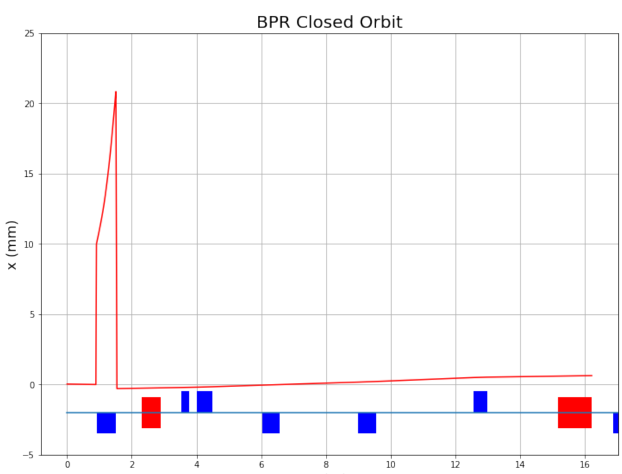

Figure 7: BEPCII BPR Horizontal Closed Orbit in IR.

Beam Line

When a layout of a lattice is created, the BeamLine can be get easily. The following code defines a reference particle of Positron and generates a BeamLine using the previously defined BPRLayout. Certain parameters are spicified: *isRing=True* indicates that the BeamLine is a ring, and a dictionary is assigned to *paraDict*, with some elements in the dictionary being divided into 10 slices for calculation. With this BeamLine, further beam optics calculations can be performed. Figure 7 depicts the closed orbit of the BEPCII BPR in the interaction region, computed using APES. The left side of the figure provides a clear representation of the orbit in the SCQ.

```
refParticle = Positron(P=1.89E9)
beamline = BPRLayout.getBeamLine('BPR',
    refParticle=refParticle, isRing=True,
    checkRing=True, paraDict={'D4I02': {'Slice':
     10}, 'D3O02': {'Slice': 10}})
```

Alignment Error

With the introduction of the Patch class and its pass method, handling element misalignment becomes significantly simpler. The alignment error can be assigned during the element definition or specified at a later stage. During survey or optics calculations, the alignment error is treated as two patches, as mentioned above. The following code snippet demonstrates how the BEPCII positron transfer line is managed in the presence of misalignment. First, an affine transformation that includes rotation and translation is defined to represent the misalignment of the TCQ1. Then, the alignment error of TCQ1 is updated. The same treatment is applied to the other elements as well. By activating the alignment error switch, a BeamLine that includes alignment errors is obtained. Finally, the optics calculation is performed. Figure 8 illustrates the horizontal orbit of the BEPCII e+ transport line affected by the misalignment without orbit correction.

```
...
tcq1AT = AT.createAT(elevation=ax, azimuth=ay,
    tilt=az, dx=dx, dy=dy, dz=dz)
TCQ1.update({'inAlignmentLLCT': tcq1AT})
...
beamline = TPLayout.getBeamLine('TP',
    refParticle=refParticle, isRing=False,
    alignmentError=True)
result = beamline.calBeamLineTwiss(X0=np.zeros
    (6), betax=12.5, alphax=0, betay=12.5,
    alphay=0)
...
```

SUMMARY

APES has made significant progress by completing the initial framework, which includes preliminary accelerator design and computational capabilities. Furthermore, there are plans to gradually incorporate more accelerator physics calculation features.

ACKNOWLEDGEMENTS

The authors would like to thank K. Ohmi, D. Zhou, K. Oide and E. Forest for their support and fruitful discussions. This project is supported by IHEP Innvotation Study.

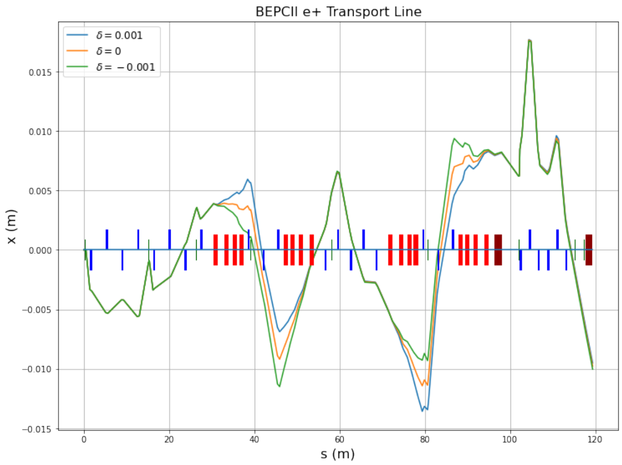

Figure 8: BEPCII e+ Transport Line Horizontal Orbit with Alignment Errors measured in the summer of 2022.

REFERENCES

[1] CEPC Study Group, "CEPC Conceptual Design Report: Volume 1 - Accelerator". doi:10.48500/arXiv.1809.00285

[2] V.I. Telnov, " Restriction on the energy and luminosity of e^+e^- storage rings due to beamstrahlung ", *Phys. Rev. Lett.*, vol. 110, p. 114801, 2013. doi:10.1103/PhysRevLett.110.114801

[3] D. Shatilov, " FCC-ee Parameter Optimization ", *ICFA Beam Dyn. Newslett.*, vol. 72, p. 30, 2017.

[4] K. Oide *et al.*, "Design of Beam Optics for the Future Circular Collider e+e- Collider Rings", *Phys. Rev. Accel. Beams*, vol. 19, p. 111005, 2016. doi:10.1103/PhysRevAccelBeams.19.111005

[5] P. Raimondi, "Introduction to Super B-Accelerators", presented at the 2nd Workshop on Super B-Factory, Frascati, Italy, Mar. 2006.

[6] J. Wu *et al.*, "Dynamic aperture optimization with diffusion map analysis at CEPC using differential evolution algorithm", *Nucl. Instr. and Meth. A*, vol. 959, p. 163517, 2020. doi:10.1016/j.nima.2020.163517

[7] N. Kuroo *et al.*, "Cross-wake force and correlated head-tail instability in beam-beam collisions with a large crossing angle"*Phys. Rev. Accel. Beams*, vol. 21, p. 031002, 2018. doi:10.1103/PhysRevAccelBeams.21.031002

[8] Y. Zhang *et al.*, "Self-consistent simulations of beam-beam interaction in future e^+e^- circular colliders including beamstrahlung and longitudinal coupling impedance", *Phys. Rev. Accel. Beams*, vol. 23, p. 104402, 2020. doi:10.1103/PhysRevAccelBeams.23.104402

[9] C. Lin, K. Ohmi, and Y. Zhang, "Coupling effects of beam-beam interaction and longitudinal impedance", *Phys. Rev. Accel. Beams*, vol. 25, p. 011001, 2022. doi:10.1103/PhysRevAccelBeams.25.011001

[10] D. Zhou *et al.*, "Simulations and experimental results of beam-beam effects in SuperKEKB", Accepted by *Phys. Rev. Accel. Beams*.

[11] https://acc-physics.kek.jp/SAD/

[12] E. Forest *et al.*, "The correct local description for tracking in rings", Accepted by *Gordon and Breach Science Publishers,S.A.*

[13] Chao A. W. , "Evaluation of beam distribution parameters in an electron storage ring", *Journal of Applied Physics*, vol. 50, no. 2, pp. 595–598, 1979.

[14] K.Ohmi *et al.*, "From the beam-envelope matrix to synchrotron-radiation integrals", *Physical Review E*, vol. 49, pp. 751–765, 1994.