Streak Camera System and Tune Resonance Tool

L. Dovlatyan, M. Ries, P. Goslawski, and friends at HZB[1, a]

University of California Berkeley, 94720 Berkeley, CA, USA

(Dated: August 2014)

CONTENTS

I. INTRODUCTION

The Helmholtz-Zentrum Berlin is a facility that operates two synchrotron light sources: BESSY II and the MLS. One of the long term goals of this center is to continually make bunch lengths as short as possible in storage rings. A way to go about this is through lattice design by changing the momentum compaction factor. An important part of lattice design involves picking a good working point to avoid the tune resonances of the machine. A tune resonance program was therefore developed which can be used to view the current working point and resonance lines given only a few input parameters from the EPICS control systems.

Another way of looking into bunch lengths at HZB is through diagnostic tools. A new streak camera was recently purchased for the MLS, and is currently being setup and tested. The Metrology Light Source (MLS) is an electron storage ring designed as a dedicated UV and VUV source; it has an asymmetric double-bend achromate design as well as six beamlines, two of which are located on a second floor, above the synchrotron. Both the new and the old streak cameras are setup with a UV beamline on the second floor, directly above the machine. Being on the second floor and having the cameras setup far away from the beamline required advanced optical paths in order to get a focused beam into the slit of the cameras.

II. TUNE RESONANCE PROGRAM[1]

The program is based on simple principle...

$$m * Q_x + n * Q_y = p \tag{1}$$

The program features a very helpful GUI jfosf ksof o kos fkosf okf okf okfsoksf oksf oks
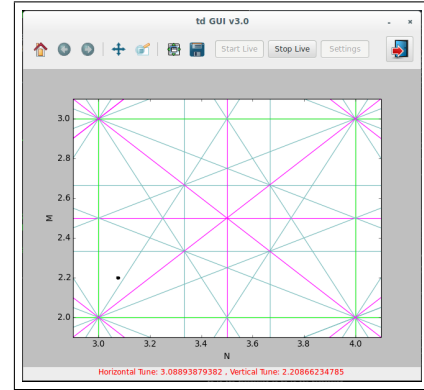
Continue with stuff and things blah blah blah blah blha blah blah

A. Developement

The program is written in Python and completely open source. This provided the opportunity to integrate with EPICS (Experimental Physics and Industrial Control System) through a special python package called PyEpics and build a GUI system using wxPython to make the program simple to use. Much of the program also relies heavily on numpy and matplotlib packages available for Python.

The biggest feature of the program is a 'live mode' option that is able to give the current tune position numerically and graphically. Using the proper epics channels, the program is able to grab the four values it needs to calculate the non integer tune: the RF cavity frequency, the harmonic number, and the horizontal and vertical betatron oscillation frequencies. The program can check and update the tune several times a second, but in order to have the visual display updated, a threading process was used to avoid having the GUI freeze up while in live mode. The threading process is able to do all the processing and updating of the matplotlib graph, continously updating the working point on the plot.

B.   Customizability

Various customizability options are available for the program. Changing color settings, removing certain resonance lines, and adjusting for mirrored tunes are just some features available for the user. Being able to display phase advance resonance lines is another unique feature included in the program.

Aside from accelerators having to position their working points away from resonane lines, they must also consider the phase advanced resonance lines as these are much more powerful and dangerous to approach with a working point. Given the number of unit cells for the machine, the program is able to display a tune diagram with the phase advance resonance lines along with the working points. This feature can complement the live mode option available with the program. If an operator wants to move the working point and cross resonance lines, the program can be used as a visual guide in order to avoid the phase advanced resonance lines (colored in red in figure 1) which will more than likely kill the beam.
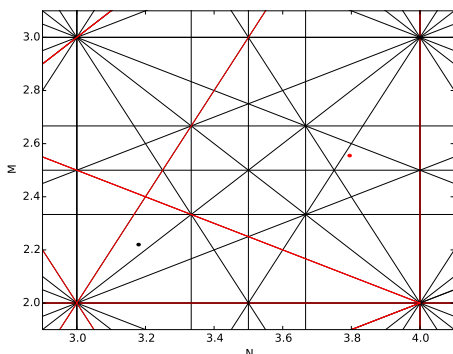

Figure 1: Tune resonance diagram displaying phase advance lines (4 unit cells) for the MLS

C.   Future Improvements

The program's availability on github allows for updates and improvements to be quickly added at the request of users.

III.   STREAK CAMERA

A.   Journal Substyle

To access particular features of the AIP substyle, you will specify an additional document class option: the journal substyle, e.g.,

\documentclass[aip,jcp]{revtex4-1}

in this case, *J. Chem. Phys.*, the default. A complete list of AIP journals with the corresponding journal substyle appears in Table **??**.

B.   Options for Citations and Bibliography

The citation style for AIP journals is:

- numerical (default style),

- author-year, and

- numerical author-year,

the latter two styles being only allowed for *Chaos* or *J. Math. Phys.*

The familiar numerical citations and numbered bibliography are the default for most journals: citations are superscript numbers, and the (numbered) bibliographic entries appear in the order cited.

Author-year citations are only allowed for *Chaos* or *J. Math. Phys.*, with citations given in author-and-year format. Bibliographic entries are sorted by alphabetical order of first author's surname, then by year.

Numerical author-year citations (only allowed for *Chaos* or *J. Math. Phys.*) are superscript numbers, just like numerical citations, but the bibliographic entries are sorted like the author-year entries and are numbered. This means that the first citation will not necessarily be 1.

To obtain the numerical style, simply accept the default, or supply a class option of numerical:

\documentclass[aip,numerical]{revtex4-1}

For author-year citations for *Chaos* or *J. Math. Phys.*, you may specify the author-year option:

\documentclass[aip,author-year]{revtex4-1}

Each of the above two options are part of standard REVTeX.

To obtain numerical author-year citations for *Chaos* or *J. Math. Phys.*, give the author-numerical option:

\documentclass[aip,author-numerical]{revtex4-1}

Note that the author-numerical option is not part of standard REVTeX so use of it outside of the AIP substyles may not have any effect.

C.   Formatting Options

There are two commonly used formats for an article you may write. One will comply with the manuscript submission formatting requirements of the editorial office of the journal you are submitting to. The other will emulate the format of your article in the published journal itself.

For journal submission, accept the default, or you may specify the preprint option:

\documentclass[aip,preprint]{revtex4-1}

To emulate the formatting of the journal, specify the reprint option:

TABLE I. Other class options

Function	class option
Citation and References	
superscript numbered	`numerical`[a,b]
author-year	`author-year`[c]
numbered author-year	`author-numerical`[c]
Format	
journal submission	`preprint`[a]
journal emulation	`reprint`

[a] Default option.
[b] Standard
[c] Only allowed for *Chaos* or *J. Math. Phys.*

`\documentclass[aip,reprint]{revtex4-1}`

Note that emulation is not by any means complete: the fonts used will differ, and therefore the length of the article will not represent an accurate estimate. Other details may also differ.

A summary of class options of interest to AIP authors appears in Table I.

[1]Source code can be found on the following github page: https://github.com/levondov/TuneResonancePython.
[2]Available with the REVTEX distribution, see http://authors.aps.org/revtex4/.
[3]L. Lamport, *LaTeX a Document Preparation System* (Addison-Wesley, Reading, MA, 1996).
[4]M. Goosens, F. Mittelbach, and A. Samarin, *The LaTeX Companion* (Addison-Wesley, Reading, MA, 1994).
[5]D. E. Knuth, *The TeXbook* (Addison-Wesley, Reading, MA, 1986).
[6]H. Kopka and P. Daly, *A Guide to LaTeX 2ε* (Addison-Wesley, Reading, MA, 1995).
[7]M. Goossens, S. Rahtz, and F. Mittelbach, *The LaTeX Graphics Companion* (Addison-Wesley, Reading, MA, 1997).
[8]S. Rahtz, M. Goossens *et al.*, *The LaTeX Web Companion* (Addison-Wesley, Reading, MA, 1999).