Table of Contents

The Online Tune Footprint Viewer

The understanding of bunch by bunch differences is crucial in a machine with several bunches as the LHC. Indeed, the collision pattern of the different bunches can be very different, the beam-beam effects expected are therefore different for every bunch. While a precise evaluation of the effects requires a deeper analysis, the tune footprint allows to assess, at least qualitatively, many aspects in a rather simple way. It is then natural to bring this information to the control room, as this can allow to better understand measured bunch by bunch differences and guide the operator during, for example, an optimisation of the lifetime by variation of the lattice tunes.

Implementation

Each point of the footprint is the result of the tracking of one particle through the lattice, including the beam-beam interactions. This kind of computation can be done within the DYNAP module of MAD - X. In order to easily integrate the simulation to the control system of the LHC, which is implemented in Java, a new module of JMAD [3], a Java-Application Programing Interface (API) for MAD - X , was developed in order to give access to the track- ing functionalities. This approach allows to profit from the tools developed in the frame of the ONLINE MODEL [4] for the acquisition of machine and beam parameters, via the LOGGING [5] and LSA [2] databases. The application is mainly divided in two sub-packages, responsible for the tracking and the user interface. In particular, the Graphical User Interface (GUI) package collects the data from the user and the different databases to define tracking jobs. Each tracking job is then launched in a separate thread each running its own MAD - X instance. The result is sent back to the GUI for plotting, processing and saving. This procedure allows to parallelise the computation of several footprints simultaneously,such as to characterize simultaneously the large variety of different PACMAN bunches. In realistic configurations, the workload becomes too heavy for a console in the control room, there the application is run on a dedicated remote server.

Automatic acquisition of machine and beam parameters

While in principle usable for simulations of footprint with parameters set by hand, the application does not offer much more than is already offered to the regular user via MAD - X scripting. The application rather aims at providing automatically footprints based on current machine and beam parameters. Respectively, the energy, the (bunch by bunch) intensity (FBCT), the current beta*s, the configuration of bunches, the octupole strength and the experiment s spectrometers strengths are loaded from the LOGGING database, the crossing angles and the full optics are loaded from the LSA database. The transverse emittances have to be set by hand, as no continuous and reliable measurements are currently available. The transverse separation at the IP requires a bit of complication. Indeed, it is not possible to measure it at any time of the LHC cycle based on the value of the separation knob stored in the LSA database, at least not to the precision required. The present version uses the luminosity measurement of each experiment to evaluate the separation, making assumptions on the cycle. In IP1, 2 and 5, where no levelling methods are used, the separation is set as full in case no luminosity is measured and 0 otherwise. In IP8, which is levelled with a transverse offset, the separation is computed based on the evaluation of the luminosity reduction factor R from the ratio of the measured luminosity in IP1 and IP8, taking into account the different number of bunches colliding in the two IPs as well as the different beta*. If set in Online mode, the application will reload these informations before each computation of a footprint.

Bunch selection

Simulating the footprint for all bunches during a physics fill is not only out of reach from the computing power point of view, but it would also extremely difficult to interpret the result. It is therefore important to choose a subset of bunches that will be representative of the whole beam, a recommended bunch selection is then proposed automatically. The bunches are divided in head-on families, i.e. bunches colliding head-on in the same interaction points. The set of bunches in each head-on family is represented by the two extreme bunches experiencing respectively the least and the largest number of long-range interactions. Within LHC configuration, there can be up to 7 HO family, therefore we need at most 14 footprints in order to obtain a relevant representation of the beam.

Maintenance

The automatic acquisition of parameters heavily relies on the knowledge of the standard operational cycle. The according methods have to be updated to any change of the operational configuration (e.g. levelling strategies). Being based on MAD - X module THINTRACK , the tracking requires thin lens optic. The thin lens optics must be generated based on the thick lens version used to generated the machine settings, checked and uploaded to the ONLINE MODEL . To avoid this workload one may want to use PTC tracking [6], which allow the use of thick lens. This upgrade could be done, at the expense of longer computing time and significant work on the implementation of the beam-beam elements in the thick lens lattice.

Usage

Working point optimisation

The single particle dynamic is strongly affected by the presence of non-linearities, arising from beam-beam interactions or from the lattice. A full understanding of the behaviour of the particles requires heavy analytical or numerical tools, in particular long term tracking simulations [1]. Far from allowing a quantitative evaluation of lifetime degradations due the non-linear effects, the footprints rather offer a possibility to make qualitative comparisons between the different bunches. Ultimately, one may use it in an empirical approach to look for better working points, i.e lattice tunes which would optimize the beam lifetime based on measured bunch by bunch differences. Indeed, being spread around in the tune diagram, the different bunches experiences different beam-beam effects with different strength. The comparison of measured lifetimes against positions and extend in the tune diagram for the different bunch families may be used to guide the operator in a tune scan, aiming at maximizing the overall lifetime. In the present LHC configuration, one might assume that the footprint, relatively to the unperturbed tune, is independent of the value of the unperturbed tune. Therefore, one can see the effect of a change of tune by shifting all the footprints. The overall lifetime therefore have a better chances to improve by choosing a shift that will move the footprints towards those who have the best lifetime.

Stability considerations

The LHC beams rely on Landau damping for the stability of impedance driven mode. Landau damping is a direct consequence of the tune spread created by the different non linearities. Even though the tune spread does not directly provide the stability diagram, it gives already a first insight in the evolution of the tune spread during operational processes, which could help explaining difference in the stability of different bunches. Going back to the example illustrated in Fig. D.1, it is clear that bunches that do not experience head-on collision in IP1 and 5 (blue footprints) have a much smaller tune spread than other bunches. A different behaviour is expected in term of coherent stability for those bunches with respect to the others. It is, for the moment, not planed to implement an online computation of the stability diagram, using PySSD , due to the heavy computational requirement.

How to

Basics

The minimal user may want to follow the instructions illustrated below. Once launched in online mode, the application will loop on the following procedure until the user stops it or and error is encountered. Eventually, the dump of one beam will stop the execution.

- Load current beam parameters
- Start the computation of footprints for the (automatically) selected bunches

- Display footprints
- Wait until all jobs are finished

- Example :

Advanced

A more advanced user may want to control the machine and beam parameters through the different control side panels. Also, the track panel allows to choose different parameter of the simulation. It is important to note that the computation time goes linearly with the number of particle to track. The current settings allow to easily spot the main feature of the footprints while keeping the computing time reasonably small.

Manipulation

The footprint describes the frequency spread of non-linear system, strictly speaking, every change of machine or beam parameter requires the footprint to be re-computed. However, one may use scaling laws to manipulate the footprint without having to recompute it, within a limited range and making a few assumptions. Firstly, neglecting the dynamic as well as the effect of resonances, the footprint, relatively to the lattice tune, is independent of the value of the lattice tunes. Secondly, under the same assumptions, the footprint due to beam-beam effects may be considered as linearly dependent on the intensity. This assumption is not valid for lattice non-linearities. The manipulation panel takes advantage of two these laws to manipulate the footprints.

Display

The footprints are display in a tune diagram (Qx vs. Qy), colour-coded according to their HO family. For comparison, the working point is marked with a red dot. The Resonance lines button allow to choose which resonances are displayed on top. When displaying a footprint, the already existing one of the same bunch will be overwritten. This can be avoided by checking Keep footprints. The algorithm to find the tunes relies on the peak finding in the FFT of the tracking data. In the case of non-zero coupling, it is likely for the algorithm to mistake horizontal and vertical peaks for the particle oscillating at small amplitude in one plane and large in the other. If Repair is checked, such errors will be detected, and the corresponding point will not be displayed.

References

1. H. Grote, F. Schmidt, and L. Leunissen, LHC dynamic aperture at collision, LHC Project Note 197 (CERN, Geneva, Switzerland, 1999).
2. C. Roderick and R. Billen, The LSA database to drive the accelerator settings, CERN-ATS-2009-100 (CERN, Geneva, Switzerland, 2009).
3. K. Fuchsberger, X. Buffat, Y. Levinsen, and G. Müller, in Proceedings of 2nd International Particle Accelerator Conference, San Sebastián, Spain, 4-9 September 2011, edited by C. Petit-Jean-Genaz, A. Blanco, I. Etxebarria, F. Perez, A. Wolski, and V. Schaa (JACoW, Geneva, Switzerland, 2011) pp. 2292 2294.
4. C. A. Pons, X. Buffat, M. Giovannozzi, G. Müller, S. Redaelli, K. Fuchsberger, M. Lamont, and F. Schmidt, in Proceedings of 1st International Particle Accelerator Conference, Kyoto, Japan, 23-28 May 2010, edited by A. Noda, C. Petit-Jean-Genaz, V. Schaa, T. Shirai, and A. Shirakawa (JACoW, Geneva, Switzerland, 2010) pp. 480 482.
5. R. Billen and C. Roderick, The LHC Logging Service. Capturing, storing and using time-series data for the world s largest scientific instrument, AB-Note-2006-046 (CO) (CERN,Geneva, Switzerland, 2006)
6. E. Forest, F. Schmidt, and E. McIntosh, Introduction to the Polymorphic Tracking Code, CERN-SL-2002-044 (AP) (CERN, Geneva, Switzerland, 2002).
7. J. Laskar and D. Robin, Part. Acc. 54, 183 (1996).

Contact persons

- **Xavier Buffat**

This topic: ABPComputing > FootprintViewerWiki
Topic revision: r1 - 2016-04-20 - XavierBuffat

Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback