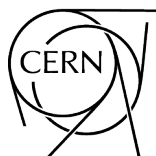


ICFA Mini-Workshop on Tracking for Collimation in Particle Accelerators

CERN, Geneva, Switzerland, 30 October 2015

Editor: S. Redaelli



CERN Yellow Reports: Conference Proceedings
Published by CERN, CH-1211 Geneva 23, Switzerland

ISBN 978-92-9083-521-9 (paperback)

ISBN 978-92-9083-522-6 (PDF)

ISSN 2519-8084 (Print)

ISSN 2519-8092 (Online)

DOI <https://doi.org/10.23732/CYRCP-2018-002>

Accepted for publication by the CERN Report Editorial Board (CREB) on 21 December 2018

Available online at <http://publishing.cern.ch/> and <http://cds.cern.ch/>

Copyright © CERN, 2018

© Creative Commons Attribution 4.0

Knowledge transfer is an integral part of CERN's mission.

CERN publishes this volume Open Access under the Creative Commons Attribution 4.0 license

(<http://creativecommons.org/licenses/by/4.0/>) in order to permit its wide dissemination and use.

The submission of a contribution to a CERN Yellow Report series shall be deemed to constitute the contributor's agreement to this copyright and license statement. Contributors are requested to obtain any clearances that may be necessary for this purpose.

This volume is indexed in: CERN Document Server (CDS), INSPIRE.

This volume should be cited as:

Proceedings of the ICFA Mini-Workshop on Tracking for Collimation, Geneva, Switzerland, 30 October 2015, edited by S. Redaelli, CERN Yellow Reports: Conference Proceedings, Vol. 2/2018, CERN-2018-011-CP (CERN, Geneva, 2018), <https://doi.org/10.23732/CYRCP-2018-002>

A contribution in this volume should be cited as:

[Author name(s)], in Proceedings of the ICFA Mini-Workshop on Tracking for Collimation, Geneva, Switzerland, 30 October 2015, edited by S. Redaelli, CERN Yellow Reports: Conference Proceedings, Vol. 2/2018, CERN-2018-011-CP (CERN, Geneva, 2018), pp. [first page] – [lastpage], <https://doi.org/10.23732/CYRCP-2018-002>. [first page]

Abstract

The ICFA mini-workshop on “Tracking for Collimation in Particle Accelerators” was held on 30 October 2015 as a satellite meeting to the 5th Joint High-Luminosity LHC/LARP Collaboration meeting.

This event gathered together various experts participating to the development and exploitation of codes for simulating the performance of collimation systems in super-colliders. This was the first topical workshop organised on this subject, and was motivated by the significant effort developed in recent years on these topics. In particular, it provided a forum for the discussion between various participants to the European Programmes within the FP7 Capacity programme, who collaborated on collimation activities for the upgrade of the LHC collimation system. The simulation tools discussed have an interest that goes well beyond. We invited also colleagues from laboratories in USA and extended the discussions to other operating or past particles accelerators. The scope was extended to include the preparation of written proceedings for each contributions, to document the status of various simulations codes. For this written document, we also accepted relevant contributions on topics that were not discussed at the workshop.

Preface

Beam collimation and machine protection aspects have become essential aspects for the operation of modern high stored-energy accelerators. The understanding of operating facilities and performance extrapolations for future machines require unprecedented accuracy and reliability in numerical simulations of beam-cleaning systems. In particular, this is important for colliders based on superconducting magnets that are at risk of quenches in presence of beam losses. Tiny fractions of the circulating-beam intensity are sufficient to perturb the behaviour of superconductors. For example, for the Large Hadron Collider (LHC) the design stored-beam energies are a few billion times larger than what can quench superconducting materials whenever a concentrated, local beam loss occurs in a superconducting magnet!

Accurate simulations of collimation mechanisms are critical for the LHC and for its High-Luminosity (HL) upgrade project, which aims at doubling the LHC's stored-beam intensity, as well as for the Future Circular Collider (FCC) presently under study. Moreover, simulations are also important for a number of other lower-energy accelerators that operate with high hadron-beam intensities.

There are several challenges when simulating high-efficiency cleaning systems. First of all, these simulations require precise tracking of halo particles with large betatron and off-momentum amplitudes. This must be combined to a very accurate modelling of collimator geometries and of the scattering process of particles in the collimator jaws. Finally, detailed aperture checks are required to identify precisely the longitudinal locations of losses, and the local transverse distributions. All this must be put together in integrated simulation tools that are fast enough to provide, within a reasonable time, high-statistics results even at very low-loss locations. In this respect, CPU-optimized simulations are a concern for large-size accelerators, like for example the FCC.

All this indicates that we are facing a multi-disciplinary domain involving several different research areas related to, but at the same time going well beyond, accelerator physics. It is therefore no surprise that in recent years several new institutes and universities have joined this effort, in addition to the laboratories that historically led key tool development because they housed super colliders, such as CERN, Tevatron, RHIC, which are all represented in these proceedings. The addition of new partners and novel approaches to this research domain is a very welcomed initiative to further improve various aspects of these simulations as well as to foster new collaborations.

Several tools have been developed by different teams, putting together the required competence to improve simulation accuracy and reliability on all fronts, although not every tool has reached the same level of maturity and efficiency. This workshop has been organised with the goal to review the current status in the field and to promote benchmarking and further developments.

This workshop gathered together the key developers and users of different codes, and international experts to address:

- Status and prospect of different codes being developed for collimation studies;
- Coupling of different tools to provide integrated frameworks;
- Recent advanced implementations (halo models, hollow e-lenses, crystals, dynamical simulations)
- Status of collimation simulations for heavy ion beams;
- Results of recent benchmark studies of numerical simulations against beam data from operating machines, spanning over a large range of beam energies.

In addition to the written contributions of oral presentations that took place at the workshop, we invited also other colleagues to contribute papers on the workshop topics, as we considered the material very pertinent to the topics discussed here.

I would like to sincerely acknowledge the authors of these proceedings for enthusiastically contributing to this document that provides, in my opinion, a very nice reference on the present state-of-the-art of collimation

simulation tools in hadron collider. I would also like to acknowledge personally a number of colleagues that agreed to act as reviewers of the paper: R. Bruce, O. Brüning, H. Burkhardt, F. Cerutti, M. Giovannozzi, W. Herr, B. Holzer, A. Mereghetti, G. Rumolo, R. Tomás. In particular, I thank M. Giovannozzi who also helped with precious support in the preparation of this book.

This workshop was supported by the HL-LHC Project (WP5, Collimation Upgrade), by the European Commission under the FP7 Capacities project HiLumi LHC, grant agreement 284404, under the FP7 Capacities project EuCARD-2, grant agreement 312453, and under the HORIZON2020 Integrating Activity project ARIES, grant agreement 730871. These panels and agencies are warmly acknowledged.

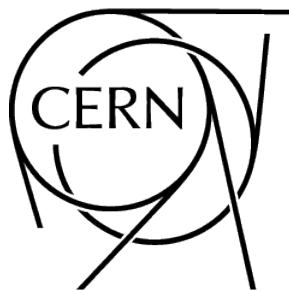
The websites of the “Tracking for Collimation in Particle Accelerators” workshop and of the 5th Joint HiLumi LHC/LARP meeting are:

<https://indico.cern.ch/event/455493>

<https://indico.cern.ch/event/400665>



Research supported by the High Luminosity LHC project



Contents

Preface

.....	iv
Status of SixTrack with collimation	
<i>R. Bruce et al.</i>	1
SixTrack Status	
<i>R. De Maria et al.</i>	11
FLUKA coupling to Sixtrack	
<i>E. Skordis et al.</i>	17
Status of MERLIN	
<i>H. Rafique et al.</i>	27
BDSIM: Automatic Geant4 Models of Accelerators	
<i>L. J. Nevay et al.</i>	45
MARS15-Based System for Beam Loss and Collimation Studies	
<i>N.V. Mokhov and I.S. Tropin</i>	57
Simulation Tools for Heavy-Ion Tracking and Collimation	
<i>P. D. Hermes et al.</i>	73
Crystal implementation in SixTrack for proton beams	
<i>D. Mirarchi et al.</i>	91
Updated implementation of collimator materials in SixTrack and MERLIN codes	
<i>E. Quaranta et al.</i>	109
Dynamic simulations in SixTrack	
<i>K. Sjobak et al.</i>	123
HL-LHC Hollow Electron Lens Integration using MERLIN	
<i>H. Rafique et al.</i>	135
Simulating off-momentum loss maps using SixTrack	
<i>H. Garcia-Morales et al.</i>	163
First simulations of collimation cleaning performance for the FCC-hh	
<i>M. Fiascaris et al.</i>	171
Simulations of collimation losses at RHIC	
<i>G. Robert-Demolaize and A. Drees</i>	181
Simulation and optimization of beam losses during continuous transfer extraction at the CERN Proton Synchrotron	
<i>J. Barranco Garcia and S. Gilardoni</i>	189
FLUKA-SIX TRACK Coupling for the SPS Scrapers	
<i>R.B. Appleby et al.</i>	205

Status of SixTrack with collimation

*R. Bruce^{*1}, M. Fiascaris¹, M. Fitterer², H. Garcia Morales⁴, P.D. Hermes¹, R. Kwee-Hinzmann⁴, A. Mereghetti¹, D. Mirarchi¹, J. Molson³, E. Quaranta¹, S. Redaelli¹, B. Salvachua¹, A. Santamaria Garcia¹, K. Sjobak¹, C. Tambasco¹, and J. Wagner¹*

¹CERN, Geneva, Switzerland

²FNAL, Batavia, IL, USA

³LAL, Univ. Paris-Sud, CNRS/IN2P3, Université Paris-Saclay, Orsay, France

⁴Royal Holloway, University of London, Surrey

Abstract

In this paper, we review the functionality and status of the collimation version of SixTrack. It is a simulation tool that contains both an accurate magnetic tracking of an ensemble of particles, as well as a model for particle-matter interaction inside collimators, in order to model the efficiency of a proton collimation system. We summarize recent upgrades and improvements and we review also a benchmark with measured data before finally discussing plans for future development.

Keywords

LHC; collimation; tracking simulations.

1 Introduction

The Large Hadron Collider (LHC) [1, 2] at CERN is designed to store and collide protons with an unprecedented beam energy of 7 TeV and a total stored energy of about 362 MJ per beam. For the upgrade of the LHC, HL-LHC [3], an increase of the stored energy to almost 700 MJ is foreseen. The LHC beams are highly destructive because of the high stored energy, and all beam losses must be tightly controlled. For this purpose, a multi-stage collimation system is used [1, 4–9], in order to intercept unavoidable beam losses in a safe way.

To assess the performance of the collimation system, a detailed understanding of the cleaning of beam protons by the collimators is needed. The collimators are not perfect absorbers and a certain fraction of the protons intercepted by the collimation system is outscattered and risks impacting on the cold magnets, where they could induce quenches. It is therefore crucial that the efficiency of the collimation system can be accurately modelled already on the design stage, in order to ensure that the accelerator is able to operate smoothly without lengthy interruptions caused by quenches or beam dumps.

Protons lost in the cold magnets have usually hit a collimator and afterwards have travelled some distance through the magnetic lattice of the ring—in many cases, protons are lost several turns after their first collimator impact. Therefore, we need accurate simulation tools that model both the tracking through the magnetic lattice as well as the particle-matter interaction inside collimators. The magnetic tracking is critical in particular because of the large amplitudes and energy deviations of the halo particles. SixTrack with collimation [10–14], which we in this paper simply call here “SixTrack”, was developed for this purpose. In this paper, we review the functionality, status and future plans of this code.

2 Functionality and of SixTrack

SixTrack is a multi-turn tracking code, which takes the six-dimensional phase space into account in a symplectic manner [10–12]. SixTrack performs a thin-lens element-by-element tracking through the

^{*}roderik.bruce@cern.ch

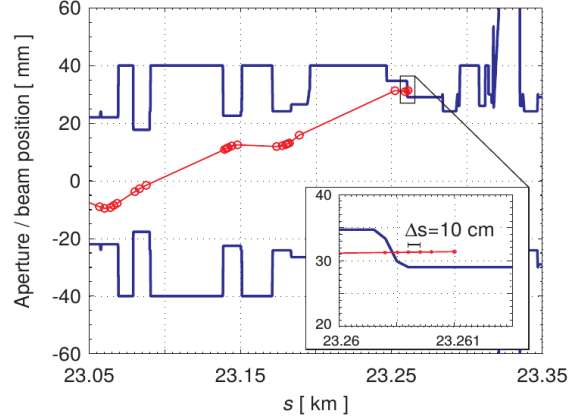


Fig. 1: Example of a halo particle (red line) that hits the LHC aperture (blue line) [17].

magnetic lattice, including multipoles up to order 20. Initially, SixTrack was developed for dynamic aperture studies where a very high numeric stability is needed when tracking particles over a large number of turns. During the design of the LHC, the K2 scattering routine was incorporated in SixTrack [13, 15]. When a particle is tracked around the magnetic lattice and encounters a collimator, this routine is called to simulate the particle-matter interaction inside the collimator material. SixTrack takes as input a database with information about all collimators (settings, material, length, angular orientation etc.), as well as a sequence of magnetic elements, which can be created by MAD-X [16]. This provides a tight integration with the LHC magnetic imperfection model.

The scattering routine accounts for multiple Coulomb scattering and ionization energy loss, as well as several point-like processes: nuclear elastic scattering, nuclear inelastic scattering, single diffractive scattering (treated separately from other inelastic processes), and Rutherford scattering. If a proton undergoes a nuclear inelastic interaction, which is not single diffractive, it is assumed that the proton disintegrates and it is considered lost on the collimator. The induced hadronic and electromagnetic showers are not modelled by SixTrack.

If any of the other physical processes takes place, where the interacting beam proton survives, the tracking continues through the collimator material. There is a possibility that the tracked proton can exit the collimator jaw, after having received kicks in energy and angle by the scattering processes. In this case, that proton is injected again into the magnetic tracking.

A particle is considered lost either when it undergoes an inelastic interaction, as detailed above, or if it hits the aperture. The particle trajectories are written out by SixTrack in all elements and checked in a post-processing step against a detailed aperture model with 10 cm longitudinal precision [17]. This requires a detailed aperture model of the machine. The aperture check is illustrated in Fig. 1, where both the interpolated trajectory and the aperture model are shown.

For simulations of beam cleaning, the starting conditions are particle coordinates in the halo, which have already an amplitude large enough to hit a collimator. The diffusion that initially sends particles onto the collimators is not modelled. This has the advantage that it is possible to track many millions of halo particles to achieve sufficient statistics so that losses significantly below the quench level can be resolved, and also at less exposed locations. Typically, at least 6.4×10^6 halo protons are tracked for 200 turns. If the simulation would include diffusion and track the full beam core, the needed computing time would rise by many orders of magnitude and the simulation would become impractically long. On the other hand, a full beam distribution is tracked for other applications, such as failure studies.

The simulation output contains coordinates of all loss locations, which can be used either to directly assess the loss pattern and the cleaning efficiency, or as inputs to further simulation studies of energy deposition. An example of the simulated loss pattern is shown in Fig. 2, where the losses are

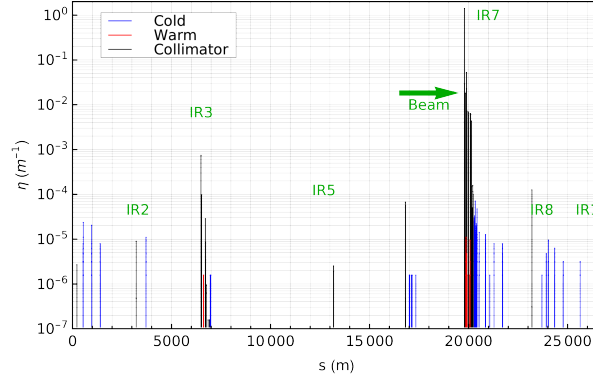


Fig. 2: Cleaning inefficiencies around the LHC obtained SixTrack, for a horizontal halo in B1, assuming the machine configuration used in the 2015 run of the LHC [18, 19].

expressed in terms of the local cleaning inefficiency η :

$$\eta = \frac{N_{\text{loc}}}{N_{\text{tot}} \Delta s}. \quad (1)$$

Here N_{loc} is the local losses over a distance Δs and N_{tot} is on total losses on collimators.

3 Recent improvements

Several improvements have recently been implemented in SixTrack with collimation, which we summarize below.

In the original version of SixTrack, the starting distribution was generated at IP1, at the start of the ring. The matched phase space in the collimation plane is populated uniformly in a thin segment around the normalized betatron amplitude corresponding to the TCP half opening. The shape in phase space is thus a thin hollow ellipse. This has the disadvantage that not all protons hit the primary collimator on the first turn. The halo can then deform due to non-linear fields such as sextupoles, which makes it difficult to control well the impact parameters.

Therefore, a new type of halo has been implemented, which we call direct halo [20]. The starting conditions are then created directly at the collimator. It is identical to the annular halo except that particles in the collimation plane are generated only in the areas of the phase space that are outside the collimator cuts. Thus, with the direct halo, all halo particles hit the TCPs on the first turn, so that the impact distribution on the TCPs is much easier to control and it is usually also more efficient in terms of computing time. Example distributions of the phase space at the TCP for the two cases are shown in Fig. 3.

Furthermore, some updates have been carried out on the K2 scattering routine [14]. The changes concern the proton-proton single diffractive cross section, considering a recent parametrization based on the renormalized pomeron flux exchange, the proton-nucleus inelastic and total cross sections, using recent data from the Particle Data Group, and the proton-proton elastic cross section, based on TOTEM data. Furthermore, the carbon material properties have been revised based on the composite material used in the collimators. The ionization energy loss and the multiple Coulomb scattering models have also been improved.

The scattering routine must also handle all possible materials of LHC collimators. The present system contains jaws made of carbon fibre composite (CFC), tungsten, or copper. For HL-LHC, it is planned to install collimators made of new advanced materials for improved impedance and robustness [22]. In order to estimate the cleaning performance with these new materials, they have been recently implemented in the SixTrack scattering routine [23, 24].

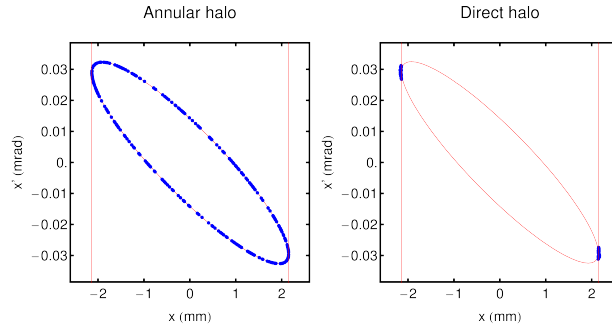


Fig. 3: Examples of the horizontal phase space of the starting distributions at the TCP for annular halo (left) and direct halo (right). Each blue point represents a single particle, and the red ellipse the matched 5.7σ envelope. The vertical red lines represent the cuts of the TCP jaws. The figure is taken from Ref. [20] under the creative commons license [21].

As an alternative to the built-in scattering routine, the option of linking SixTrack to the particle physics Monte-Carlo code FLUKA [25, 26] has been implemented, the so-called SixTrack-FLUKA coupling [27–29]. The details of this are, however, beyond the scope of this paper, since another contribution to these proceedings is dedicated to this [30]. SixTrack has also been linked to the MERLIN scattering library [31, 32], as well as to Geant4 [33, 34]. Several scattering models are therefore available to the user, and a comparison of them can be found in Ref. [35].

The choice of scattering model can be particularly important for new regimes in energy, such as for the Future Circular Collider [36], for which SixTrack is also used to assess the collimation performance [37]. These studies triggered further improvements and generalizations to SixTrack, e.g. to handle machines with a very large number of elements, as well as automatic calculations of global inefficiencies for the off-momentum halo. Further details on these studies are given in another article in these proceedings [38].

Another important update concerns the check of losses against the aperture model, which in the original SixTrack version was done using post-processing, as described above. A new implementation does this aperture check online, within the main SixTrack code. This allows an improvement in terms of needed computing time, and a less complex simulation setup. However, the possibility of using the same trajectories with many random imperfections on the aperture is lost. This can however be done with yet another improvement, which writes the trajectories into a more efficient binary format (HDF5) instead of as a text file [39].

There has also been a recent implementation in the main version of SixTrack of a dynamic kick module, called DYNK [40, 41], which allows settings of machine elements, such as magnets or RF cavities, to be changed dynamically during the simulation. A detailed description of the present state of this functionality is given in Ref. [41], however, we highlight in this paper that the DYNK module can be used to extend the range of applications of SixTrack with collimation.

For example, certain beam failures such as asynchronous beam dumps can now be simulated with SixTrack. During an asynchronous beam dump, the extraction kickers fire at the wrong moment, when beam is passing. This means that the protons moving through the kicker fields receive intermediate kicks, which risk sending them directly onto sensitive elements without hitting the primary collimators first. This accident scenario has been very important for the LHC, since the risk of damaging machine components has limited the reach in β^* [42, 43]. With the DYNK module, the rising field of the misfiring extraction kickers can be included in SixTrack. Since every passing bunch would receive a different kick, many bunches are simulated separately and the results are summed in the end. An example of the simulated loss distribution is shown in Fig. 4.

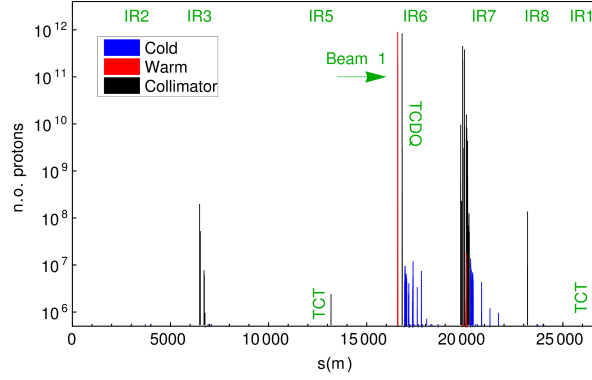


Fig. 4: Loss distribution around the LHC ring for B1 during an asynchronous beam dump, as simulated with SixTrack using the DYNK module for the 2015 standard physics configuration. Only 20 bunches, which are kicked onto amplitudes where they risk to damage equipment in the ring, were simulated and summed. If all bunches would be included, higher losses would be expected in IR6. The figure is taken from Ref. [43] under the creative commons license [44].

Another use of the DYNK module is to simulate the losses occurring during a shift of the RF frequency, similarly to what is performed operationally when the off-momentum cleaning is qualified with loss maps. These studies are described in detail in another paper of these proceedings [45]. Further applications include failures of crab cavities in HL-LHC [46–48].

Furthermore, alternative collimation schemes can be studied with SixTrack, which has required updates to the code. Firstly, the scheme of using a bent crystal instead of a standard primary collimator has been investigated [49]. If a proton hits the crystal at a specific angle, it can travel in the potential well between the parallel planes of the crystal, with a very small probability of interacting. Since the planes are bent, the proton can exit the crystal with a significant angular offset, which is enough to make the proton hit a downstream absorber with a large impact parameter. In order to simulate this collimation scheme, physics routines for the proton-crystal interaction have been implemented, with details given in Refs. [50, 51].

Another new collimation scheme under study is the use of a hollow electron lens to increase the diffusion speed of the halo, inside the cut of the primary collimator [52–56]. This allows depletion of the halo in a controlled way, thus avoiding spurious dumps if beam is scraped on the primary collimators during orbit jitters. SixTrack studies on this collimation scheme are being set up, and the hollow electron lens has been implemented as an element in SixTrack. A rich simulation program is planned to fully explore the performance.

Finally it should also be noted that an effort has been made to perform collimation studies with heavy ions using SixTrack, where the main difficulty arises from the different particle-matter interaction of heavy ions. A first approach used starting conditions from FLUKA of scattered and fragmented ions exiting the primary collimator and assumed all downstream collimators to be perfect absorbers [57]. A further development of the heavy-ion simulations included new tracking maps in SixTrack for heavy ions, based on a Hamiltonian formalism that includes both the mass and the charge of the particles [58], and also a full coupling with FLUKA to have particle-matter interactions sampled online in all collimators, as for protons. More details are given in Ref. [59].

Apart from the more extensive updates described above, an active development has recently led to improved output files that store more information of the history of the protons, as well as various optimizations, simplifications, and fixes of the collimation routines and sanity checks of the inputs.

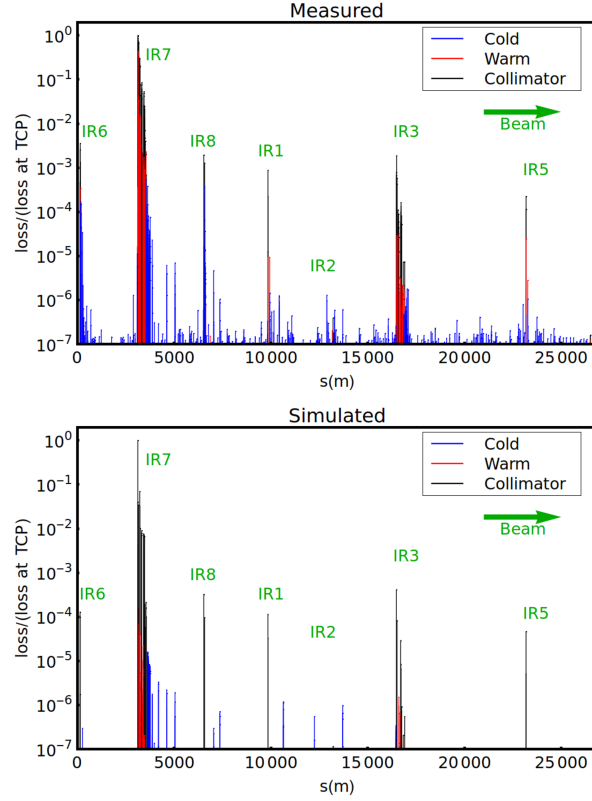


Fig. 5: Distributions of beam losses in the LHC ring as measured by BLMs during a 2011, (top) and from a SixTrack simulation (bottom), for the 2011 LHC machine configuration at 3.5 TeV and $\beta^*=1.5$ m. The simulation and measurement are both normalized to the highest loss, and the initial losses occur in the horizontal plane in B1. The figure is taken from Ref. [20] under the creative commons license [21].

4 Comparison with data

In order to be confident in the predictions by SixTrack for future accelerators, it is important to demonstrate its reliability by comparing to data from the present LHC. This is a continuous effort, which started when the first LHC data became available and is still ongoing. We summarize here one such comparison that was performed in Ref. [20].

In the lower part of Fig. 5, we show the simulated losses from SixTrack around the ring, for the case of a perfect machine without errors, using the 2011 machine configuration with a beam energy of 3.5 TeV and $\beta^*=1.5$ m. A 1 m binning of the the simulated losses has been used, except at the TCPs, which are considered as separate bins although they are only 60 cm long.

This simulation result is shown together with an example of measured losses during a qualification loss map, where controlled losses were excited with a low-intensity beam by approaching the third order resonance. It can be seen in Fig. 5 that there is a very good qualitative agreement between the measurement and the simulation. The highest losses occur at the collimators in the betatron cleaning insertion IR7 and the second most important loss location is the momentum collimators in IR3. The TCTs upstream of the experiments, as well as the beam dump protection collimators in IR6, also intercept significant losses. It can be seen that the simulation predicts all potentially limiting cold loss locations.

It should be noted, however, that there are some significant quantitative deviations between simulations and measurements, for example at the TCTs. In order to understand this, we note that the BLMs do not measure the direct proton losses that are shown for the simulation, but instead the shower particles produced by the primary losses. The BLM signal per locally lost proton could vary significantly between

loss locations, depending on the local geometry, materials, BLM location with respect to the loss position, and the spatial and angular distribution of the losses. Because of that, one cannot expect a high level of quantitative agreement when comparing the weighted convolution of all upstream showers in a BLM with the loss locations of primary beam protons. To do a quantitative comparison, it is therefore necessary to simulate also the showers in a second step, using e.g. FLUKA.

The details of such a comparison is shown in Ref. [20]. It is demonstrated that, if imperfections are included in SixTrack, a quantitative agreement within a factor 2–3 can be obtained between simulated and measured BLM signals for cleaning losses. More recent studies using the SixTrack-FLUKA coupling and a more updated FLUKA geometry for the shower simulation show a similar level agreement for the perfect machine [60]. An agreement of a factor 2–3 is also seen when comparing the SixTrack simulations of asynchronous dumps with BLM data [43, 61]. In these cases, no dedicated FLUKA simulation was performed, but the BLM response was instead estimated from dedicated beam measurements [62].

The level of obtained agreement is still considered a very good, given the high complexity of the simulation chain, the many unknown imperfections, and the fact that the loss levels around the ring span 7 orders of magnitude. Nevertheless, this uncertainty should be kept in mind in the design of future machines when relying on SixTrack results.

5 Summary and outlook

We have given a summary of the functionality and status of the collimation version of SixTrack, which has been the standard tool for quantifying the collimation cleaning performance of the LHC. SixTrack has been shown to produce accurate loss patterns within the expected uncertainties, but further improvements are still possible. The development of SixTrack continues, and future work includes deeper studies of the scattering model and detailed comparisons to other codes, as well as general improvements in the code structure, input and output files, and a merging of different branches of SixTrack with special functionalities into one main version. The work on halo modelling continues as well, and it is planned to implement a general halo sampling that could also produce starting conditions for simulations of off-momentum collimation. Other important topics for future work are to adopt it better for large-scale simulations, using e.g. the BOINC system [63] with access to thousands of CPUs in parallel, and to continue and refine the comparison with data for other machine configurations.

References

- [1] O. S. Brüning, *et al.* LHC design report v.1 : The LHC main ring. *CERN-2004-003-V1*, (2004).
- [2] L. Evans and P. Bryant (editors). LHC machine. *JINST*, 3:S08001, (2008).
- [3] G. Apollinari, *et al.* High-Luminosity Large Hadron Collider (HL-LHC): Preliminary Design Report. *CERN-2015-005*, (2015). R. Bruce.
- [4] J. B. Jeanneret. Optics of a two-stage collimation system. *Phys. Rev. ST Accel. Beams*, 1(8):081001, (1998).
- [5] R.W. Assmann. Collimators and Beam Absorbers for Cleaning and Machine Protection. *Proceedings of the LHC Project Workshop - Chamonix XIV, Chamonix, France*, page 261, (2005).
- [6] G. Robert-Demolaize. *Design and Performance Optimization of the LHC Collimation System*. PhD thesis, Universite Joseph Fourier, Grenoble, (2006).
- [7] R.W. Assmann *et al.* The Final Collimation System for the LHC. *Proc. of the European Particle Accelerator Conference 2006, Edinburgh, Scotland*, page 986, (2006).
- [8] C. Bracco. *Commissioning Scenarios and Tests for the LHC Collimation System*. PhD thesis, EPFL Lausanne, (2008).
- [9] D. Wollmann, *et al.* First cleaning with LHC collimators. *Proceedings of the International Particle Accelerator Conference 2010, Kyoto, Japan*, page 1237, (2010).

- [10] F. Schmidt. SixTrack. User's Reference Manual. *CERN/SL/94-56-AP*, (1994).
- [11] Sixtrack web site. <http://sixtrack.web.cern.ch/SixTrack/>.
- [12] R. de Maria *et al.* Status of sixtrack. *Proceedings of the Tracking for Collimation Workshop, CERN, Geneva Switzerland*, (2015).
- [13] G. Robert-Demolaize, *et al.* A new version of sixtrack with collimation and aperture interface. *Proc. of the Particle Accelerator Conf. 2005, Knoxville*, page 4084, (2005).
- [14] C. Tambasco. An improved scattering routine for collimation tracking studies at LHC. Master's thesis, Università di Roma, Italy, (2014).
- [15] N. Catalan Lasheras. *Transverse and Longitudinal Beam Collimation in a High-Energy Proton Collider (LHC)*. PhD thesis, University of Zaragoza, (1998).
- [16] MAD-X program. <http://cern.ch/mad/>.
- [17] S. Redaelli, R. Assmann, and G. Robert-Demolaize. LHC Aperture and Commissioning of the Collimation System. *Proceedings of the LHC Project Workshop - Chamonix XIV, Chamonix, France*, page 268, (2005).
- [18] R. Bruce, *et al.* Baseline LHC machine parameters and configuration of the 2015 proton run. *Proceedings of the LHC Performance Workshop (Chamonix 2014), Chamonix, France*, (2014).
- [19] G. Valentino, *et al.* Performance of the LHC collimation system during 2015. *CERN-ACC-2015-376, Proceedings of the 6th Evian Workshop, Evian, France*, (2015).
- [20] R. Bruce, *et al.* Simulations and measurements of beam loss patterns at the CERN Large Hadron Collider. *Phys. Rev. ST Accel. Beams*, 17:081004, (2014).
- [21] Creative commons attribution license (CC BY 3.0). <https://creativecommons.org/licenses/by/3.0/>.
- [22] A. Bertarelli, *et al.* Novel materials for collimators at LHC and its upgrades. *Proceedings, 54th ICFA Advanced Beam Dynamics Workshop on High-Intensity, High Brightness and High Power Hadron Beams, HB2014 : East Lansing, MI, USA*, page 438, (2014).
- [23] E. Quaranta, *et al.* Collimation cleaning at the LHC with advanced secondary collimator materials. *Proceedings of the International Particle Accelerator Conference 2015, Richmond, VA, USA*, page 2068, (2015).
- [24] E. Quaranta *et al.* New materials for sixtrack. *Proceedings of the Tracking for Collimation Workshop, CERN, Geneva Switzerland*, (2015).
- [25] A. Ferrari, *et al.* FLUKA: a multi-particle transport code. *CERN Report CERN-2005-10*, (2005).
- [26] T. Bohlen, *et al.* The FLUKA code: Developments and challenges for high energy and medical applications. *Nuclear Data Sheets*, (120):211–214, (2014).
- [27] A. Mereghetti *et al.* Sixtrack-FLUKA active coupling for the upgrade of the SPS scrapers. *Proceedings of the International Particle Accelerator Conference 2013, Shanghai, China*, page 2657, (2013).
- [28] A. Mereghetti. *Performance Evaluation of the SPS Scraping System in View of the High Luminosity LHC*. PhD thesis, University of Manchester, (2015).
- [29] P. Hermes. *Heavy-Ion Collimation at the Large Hadron Collider : Simulations and Measurements*. PhD thesis, University of Munster, (2016).
- [30] V. Vlachoudis *et al.* Status of fluka coupling to sixtrack. *Proceedings of the Tracking for Collimation Workshop, CERN, Geneva Switzerland*, (2015).
- [31] J. Molson, *et al.* Simulating the LHC collimation system with the accelerator physics library MERLIN. *Proceedings of ICAP12, Rostock-Warnemünde, Germany*, pages 12–14, (2012).
- [32] R. B. Appleby, *et al.* The practical pomeron for high energy proton collimation. *The European Physical Journal C*, 76(10):520, (2016).
- [33] S. Agostinelli *et al.* Geant4 - a simulation toolkit. *Nuclear Instruments and Methods in Physics*

- Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 506:250 – 303, (2003).
- [34] J. Allison *et al.* Recent developments in geant4. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 835:186 – 225, (2016).
 - [35] J. Molson, *et al.* A comparison of interaction physics for proton collimation systems in current simulation tools. *Proceedings of the International Particle Accelerator Conference 2017, Copenhagen, Denmark*, page 2478, (2017).
 - [36] A. Ball *et al.* Installation of new BLMs on the D1. *CERN EDMS 134202*, (2014).
 - [37] M. Fiascaris, *et al.* First Design of a Proton Collimation System for 50 TeV FCC-hh. *Proceedings of the International Particle Accelerator Conference 2016, Busan, Korea*, page 2425, (2016).
 - [38] M. Fiascaris *et al.* First simulations of collimation cleaning performance for the FCC-hh. *Proceedings of the Tracking for Collimation Workshop, CERN, Geneva Switzerland*, (2018). exclude.
 - [39] R. Kwee-Hinzmann *et al.* Use of binary files in SixTrack. *presentation in the LHC Collimation Working Group, 2015.03.23*, (2015).
 - [40] K. Sjobak, *et al.* General functionality for turn-dependent element properties in SixTrack. *Proceedings of the International Particle Accelerator Conference 2015, Richmond, VA, USA*, page 468, (2015).
 - [41] K. Sjobak *et al.* Dynamic simulations in SixTrack. *Proceedings of the Tracking for Collimation Workshop, CERN, Geneva Switzerland*, (2015).
 - [42] R. Bruce, R. W. Assmann, and S. Redaelli. Calculations of safe collimator settings and β^* at the CERN Large Hadron Collider. *Phys. Rev. ST Accel. Beams*, 18:061001, (2015).
 - [43] R. Bruce, *et al.* Reaching record-low β^* at the CERN Large Hadron Collider using a novel scheme of collimator settings and optics. *Nucl. Instrum. Methods Phys. Res. A*, 848:19 – 30, (2017).
 - [44] Creative commons attribution license (CC BY 4.0). <http://creativecommons.org/licenses/by/4.0/>.
 - [45] H. Garcia Morales *et al.* Modelling of beam halos. *Proceedings of the Tracking for Collimation Workshop, CERN, Geneva Switzerland*, (2015).
 - [46] A. Santamaria, *et al.* Limits on failure scenarios for crab cavities in the HL-LHC. *Proceedings of the International Particle Accelerator Conference 2015, Richmond, VA, USA*, page 3923, (2015).
 - [47] A. Santamaria, *et al.* Machine protection from fast crab cavity failures in the High Luminosity LHC. *Proceedings of the International Particle Accelerator Conference 2016, Busan, Korea*, page 1481, (2016).
 - [48] K. Sjobak, *et al.* Time scale of crab cavity failures relevant for high-luminosity LHC. *Proceedings of the International Particle Accelerator Conference 2016, Busan, Korea*, page 4196, (2016).
 - [49] D. Mirarchi. Crystal Collimation for LHC, (2015).
 - [50] D. Mirarchi *et al.* Improvements of the crystal routine for collimation studies. *Proceedings of the International Particle Accelerator Conference 2014, Dresden, Germany*, page 886, (2014).
 - [51] D. Mirarchi *et al.* Update on crystal simulation for collimation. *Proceedings of the Tracking for Collimation Workshop, CERN, Geneva Switzerland*, (2015).
 - [52] V. Shiltsev, *et al.* LHC particle collimation by hollow electron beams. *Proc. of the European Particle Accelerator Conf. 2008, Genoa, Italy*, page 292, (2008).
 - [53] G. Stancari, *et al.* Collimation with hollow electron beams. *Phys. Rev. Lett.*, 107:084802, (2011).
 - [54] G. Stancari, *et al.* Conceptual design of hollow electron lenses for beam halo control in the Large Hadron Collider. *CERN-ACC-2014-0248*, (2014).
 - [55] S. Redaelli, *et al.* Plans for deployment of hollow electron lenses at the LHC for enhanced beam collimation. *Proceedings of the International Particle Accelerator Conference 2015, Richmond,*

- VA, USA, page 2462, (2015).
- [56] M. Fitterer, *et al.* Implementation of hollow electron lenses in sixtrack and first simulation results for the HL-LHC. *Proceedings of the International Particle Accelerator Conference 2017, Copenhagen, Denmark*, page 3795, (2017).
 - [57] P. Hermes, *et al.* Measured and simulated heavy-ion beam loss patterns at the CERN Large Hadron Collider. *Nucl. Instrum. Methods Phys. Res. A*, 819:73 – 83, (2016).
 - [58] P. Hermes, R. Bruce, and R. D. Maria. Symplectic tracking of multi-isotopic heavy-ion beams in SixTrack. *Proceedings of the International Particle Accelerator Conference 2016, Busan, Korea*, page 1450, (2016).
 - [59] P. Hermes *et al.* Simulation tools for ion collimation. *Proceedings of the Tracking for Collimation Workshop, CERN, Geneva Switzerland*, (2015).
 - [60] E. Skordis *et al.* Simulation comparisons to blm data. *Proceedings of the Tracking for Collimation Workshop, CERN, Geneva Switzerland*, (2015).
 - [61] E. Quaranta, *et al.* Validation of Simulation Tools for Fast Beam Failure Studies in the LHC. *Proceedings of the International Particle Accelerator Conference 2016, Busan, Korea*, page 2506, (2016).
 - [62] R. Bruce, *et al.* Collimation with tighter TCTs at $\beta^*=40$ cm. *CERN-ACC-NOTE-2015-0036*, (2015).
 - [63] LHC at home - volunteer computing for the LHC. <http://lhcathehome.web.cern.ch/>.

SixTrack Status

R. De Maria, A. Mereghetti, J. Molson, K. Sjobak
CERN, Geneva, Switzerland

Abstract

SixTrack is a 6D symplectic tracking code used to simulate single-particle dynamics in high-energy circular accelerators and storage rings. It is routinely used for the design and optimization of the LHC since its conception, and more recently for SPS, RHIC, HL-LHC, and FCC studies. This paper presents the status of the code, including the recent developments and those foreseen for extending the physics models and adding new features.

Keywords

SixTrack, symplectic tracking, single-particle simulations.

1 Introduction

SixTrack [1] is a 6D single-particle symplectic tracking code able to compute the trajectories of individual relativistic charged particles in circular accelerators for studying dynamic aperture or evaluating the performance of beam-intercepting devices like collimators [2]. It has been developed based on the 4D tracking code RaceTrack [3] by adding the longitudinal degree of freedom, introducing beam-beam forces, and extending the pre- and post-processing capabilities. It can compute linear and non-linear optics functions, time-dependent effects, and extract indicator of chaos from tracking data. SixTrack implements scattering routines and aperture calculations to compute “loss maps” (i.e., leakage from collimators as a function of longitudinal position along the ring) and collimation efficiency [4]. SixTrack also has a well-controlled floating-point arithmetic and fully reproducible results among many compilers and architectures [5].

Massive long-term tracking simulations are performed routinely using SixDesk [6], which is a running environment that is able to prepare, submit, manage, and post-process simulations using the CERN batch system [7] or through the LHC@Home platform for volunteering computing [8]. Often SixTrack lattices are prepared using MAD-X [9] models that are exported using the built-in converter.

The SixTrack project web page [10] contains links to the source code (see Section 5), documentation (see Section 4), and main references.

2 Features on the main branch

SixTrack has tracking maps for the following elements: drift (expanded [11] and full Hamiltonian [12, 13]), thin multipoles, orbit correctors, RF cavities, thick bends and quadrupoles (rigorously symplectic in 6D [14]), RF multipoles [15] (up to 4th order [16, 17]), crab cavities [18], 4D and 6D beam-beam models [19], e-lenses [20–22], and beam-beam compensation wires [23–25].

SixTrack is also able to compute one-turn maps to arbitrary order thanks to a differential algebra package. Optics functions are calculated in 5D by using a matrix formalism, and in 6D by using a differential algebra package and the Mais-Ripken formalism [26]. Optics functions are used to generate initial conditions for the tracked particles and parameters for the beam-beam lenses. In addition, tracking data can be post-processed in linear action-angle space for getting distance in phase space for Lyapounov exponent analysis, average actions, smear and harmonic analysis [27] using a modified version of the Plato library [28] and more recently NAFF [29].

Generic dynamic effects are introduced by the DYNK module, which allows applying arbitrary functions to a large fraction of element attributes [30, 31]. This feature has been successfully used to

model failure scenarios and energy error in crab cavities in the context of HL-LHC studies by applying phase modulation [32]. Generic particle exchange protocols using TCP/IP port (Fluka Coupling) and Linux pipe (BDEX module) can be used to interface tracking with external codes (e.g., FLUKA [33,34]).

Scattering models and aperture checks are implemented in a separate branch [4], which is planned to be merged in one of the future SixTrack version, similarly to an extension of the 6D formalism to track different ion species with arbitrary charge state relative to the same synchronous reference particle [35]. These code branches are generally used for collimation studies in the LHC, HL-LHC, and SPS [2].

The initial sample of particles to be tracked can be either generated in SixTrack via scans in linear action-angle space and energy offset or read from a file. The output of the simulation consists of turn-by-turn data as well as the one-turn map (in the form of closed orbit, tunes, and Mais-Ripken normalization matrix), in addition to the output of the post-processing analysis on Lyapounov pairs. Moreover, it is possible to dump particle coordinates or distribution moments (DUMP module) in arbitrary machine elements and file names. After simulation, output files can be packed in zip files (ZIPF module) to simplify data retrieval from remote running environments [27].

3 Branches

Compilation flags allow to compile SixTrack with specific functions or to choose the size of the allocated memory. In addition, few branches of the code exist, implementing additional functionalities. Not all functionalities enabled by the compilation flags and/or in the branches are mutually compatible, which is the reason why they are kept as such. This is non-ideal for those functionalities that could be merged offering a richer simulation setup choice. A development programme to merge the following functionalities has started:

Collimation : This option integrates scattering routines, management of collimator geometries, aperture checks, tracking of multiple sets of 64 particles at the same time; however, it breaks the post-processing routines for dynamic aperture and numerical reproducibility. Unlike other branches, this one is implemented as compilation flag, and is maintained in the same repository and updated together with the main code.

FLUKA coupling : This branch offers the possibility of a mutual exchange of particles with FLUKA [33, 34] through a TCP/IP socket during tracking; hence, the dynamics in the accelerator lattice can be coupled with the accurate scattering engine of FLUKA, allowing functionalities ranging from those similar to that of the collimation flag up to full on-line energy calculations; this branch comes with the addition of an on-line aperture checking, an early version of the DYNK and DUMP modules (afterwards imported into the main branch of SixTrack), the latter being a module for simple statistics on the tracked beam, and a generalised external engine for beam sampling.

Multiple ion species : This is a sub branch of the FLUKA coupling branch, in which the particle coordinates and tracking routines are extended to correctly compute the trajectories of particle with different mass and charge with respect to those of the reference particle.

4 Documentation

SixTrack documentation includes a user manual, a physics manual published as a draft in the SixTrack website, a developer manual, and a developer Wiki. The user manual [36] contains instructions on how to setup the simulation and how the quantities in the input files relate to the underlying physical model. It always refers to the latest production version of the code. A development version of the manual [37], synchronised with the latest commit in the master branch, is published in parallel. The physics manual [38] presents a short, but complete and self-contained, exposition of the physical models and mathematical methods used by SixTrack. The developer manual [39] guides the reader in the development platform and contribution strategy, as well as giving practical guidance on how to build, test, and submit patches (“pull request” in GitHub jargon) to the main branch. The developer Wiki [40] contains a collection

of notes written by the developers explaining the meaning and the usage of the internal variables and subroutines.

5 Code Developments and testing

SixTrack is developed by a community of expert users who implement and document new physics, and maintainers who coordinate the parallel developments, advise and review the code of the developers, schedule releases, and maintain the infrastructures.

The source code and development platform is hosted on GitHub [41]. The old commit history from the previous SVN repository, which has been discontinued, has been preserved. Developers are encouraged to contribute by forking the project and submit pull requests. The tool has greatly improved support for working with branches compared to SVN.

SixTrack code can build on several OS (Linux, Windows, OSX and recently FreeBSD, NetBSD and Android) with different compilers (gfortran, Intel Fortran compiler, NAG, Lahey, PGI) using a CMAKE build system [27].

A test suite is available and covers about 45% of the code in the main branch. Results of each test are identical in all platforms. The test suite runs through CTest and results are reported on CDash [42]. Tests using the collimation flags and other branches are not yet integrated in the test suites.

6 Conclusions and Outlook

SixTrack is a code with a long history of development, intimately linked to the optimization of the LHC. The main assets are the computational speed, thanks to a careful optimization of the main tracking loop, and the integration in the complex simulation environment for LHC, relying on the high-performance computing resources available at CERN and on the LHC@Home BOINC project. In the past few years, a large set of new features to cover the needs of new projects (e.g. HL-LHC and FCC) have been implemented thanks to the work of many developers, and an increased level of support in the code management.

Future developments aim at reducing incompatible branches by absorbing functionality in the main branch. This should leave only a small set of selectable features which introduce strong simulation speed or memory penalties for standard simulations. The testing platform should include more test cases to increase the tested code coverage. The physics and build manuals are in the pipeline for completion and publication. The following additional physics models are currently under development: extend DYNK to cover beam-beam lenses, implement an on-line aperture checking, extend the initial conditions module to cover distributions (BDIST module), complete and test solenoid, implement generic high order RF-multipoles, and introduce radiation effects.

SixTrackLib [43] is a new development effort to provide a library which exposes SixTrack tracking maps to into external programs. It is designed to support CPU and GPU through CUDA and OpenCL and it is written in portable C99. The long-term goal is to replace the internal Fortran tracking routines with those of SixTrackLib in order to take advantage of GPU resources on LHC@Home volunteer machines and HPC clusters.

References

- [1] F. Schmidt *et al.*, “SixTrack Version 4.2.16 Single Particle Tracking Code Treating Transverse Motion with Synchrotron Oscillations in a Symplectic Manner,” CERN/SL/94-56, 2012.
- [2] R. Bruce *et al.*, “Status of Sixtrack with collimation”, these proceedings.
- [3] A.F. Wrulich, “RACETRACK: a computer code for the nonlinear particle motion in accelerators,” DESY 84-026, 1984.

- [4] G. Demolaize, *et al.*, “A New Version of SixTrack with Collimation and Aperture interfaces”, in *PAC’05 Proceedings*, USA, 2005.
- [5] McIntosh E. and Wagner A., “CERN Modular Physics Screensaver or Using Spare CPU Cycles of CERN’s Desktop PCs”, in *Proceedings of 14th International Conference on Computing in High-Energy and Nuclear Physics*, CERN Geneva 2005.
- [6] E. McIntosh, R. De Maria, “The SixDesk Run Environment for SixTrack,” CERN-ATS-Note-2012-089 TECH, 2012.
- [7] CERN Batch service, <http://information-technology.web.cern.ch/services/batch>
- [8] M. Giovannozzi *et al.*, “LHC@Home: A volunteer computing system for massive numerical simulations of beam dynamics and high energy physics events” in *IPAC’12 Proceedings*, New Orleans, LA, USA, May 2012, paper MOPPD061.
- [9] MAD-X Project website, <http://cern.ch/mad>
- [10] SixTrack Project website, <http://cern.ch/sixtrack>
- [11] G. Ripken, F. Schmidt, “A symplectic six-dimensional thin-lens formalism for tracking,” DESY 95-63 and CERN/SL/95-12(AP), 1995.
- [12] K. Heinemann, G. Ripken, F. Schmidt, “Construction of nonlinear symplectic six-dimensional thin-lens maps by exponentiation,” DESY 95-189, 1995.
- [13] M. Fjellstrom, “Particle Tracking in Circular Accelerators Using the Exact Hamiltonian in SixTrack”, CERN-THESIS-2013-248, 2013.
- [14] D.P. Barber, *et al.*, “A non-linear canonical formalism for the coupled synchro-betatron motion of protons with arbitrary energy,” DESY 87-36, 1987.
- [15] A. Latina, R. De Maria, “RF multipole implementation,” CERN-ATS-Note-2012-088 TECH, 2012.
- [16] D.R. Brett, *et al.*, “Accurate crab cavity modeling for the high luminosity Large Hadron Collider,” Phys. Rev. ST Accel. Beams 17, 104001.
- [17] J. Barranco Garcia, *et al.*, “Long term dynamics of the high luminosity Large Hadron Collider with crab cavities,” Phys. Rev. Accel. Beams 19, 101003.
- [18] R. Calaga *et al.*, “LHC crab-cavity aspects and strategy”, in *IPAC’10 Proceedings*, Japan, 2010.
- [19] L.H.A. Leunissen, *et al.*, “6D beam-beam kick including coupled motion,” Phys. Rev. ST Accel. Beams 3 (2000) 124002.
- [20] V. Previtali, *et al.*, “Numerical Simulations of a Hollow Electron Lens as a Scraping Device for the LHC,” in *IPAC’13 Proceedings*, Shanghai, China, May 2013, paper MOPWO044.
- [21] M. Fitterer, *et al.*, “Electron lens news”, 3rd General SixTrack Meeting, 7th December 2016, <https://indico.cern.ch/event/590257/>.
- [22] M. Fitterer, *et al.*, “Implementation of Hollow Electron Lenses in SixTrack and first simulation results for the LHC,” *IPAC’17 Proceedings*, Copenhagen, Denmark, May 2017, paper THPAB041.
- [23] A. Patapenka, “Wire implementation in SIXTRACK”, Beam-Beam and Luminosity Studies meeting, 3 August 2015, <https://indico.cern.ch/event/436670/>.
- [24] A. Patapenka, “Wire model in Sixtrack - simulation results,” in *Simulations and Measurements of Long Range Beam-Beam Effects in the LHC*, 2015, <https://indico.cern.ch/event/456856/>.
- [25] A. Patapenka, “First order transport map of wire element,” in *3rd General SixTrack Meeting*, 7 December 2016, <https://indico.cern.ch/event/590257/>.
- [26] H. Mais and G. Ripken, “Theory of coupled synchro-betatron oscillations”, DESY-M-82-05, DESY, 1982.
- [27] K. Sjobak, *et al.*, “New Features of the 2017 SixTrack release”, in *IPAC’17 Proceedings*, Copenhagen, Denmark, May 2017, paper THPAB047.
- [28] M. Giovannozzi, *et al.*, “PLATO: a program library for the analysis of nonlinear betatronic motion”,

- Nucl. Instrum. and Methods A, vol. 388, 1996.
- [29] S. Kostoglou, *et al.*, “Development of Computational Tools for Noise Studies in the LHC”, in *IPAC’17 Proceedings*, Copenhagen, Denmark, May 2017, paper THPAB044, <https://doi.org/10.18429/JACoW-IPAC2017-THPAB044>.
 - [30] K. Sjobak, *et al.*, “General functionality for turn-dependent element properties in SixTrack”, in *IPAC’15 Proceedings*, Richmond, VA, USA, May 2015, paper MOPJE069.
 - [31] K. Sjobak, “Dynamic simulations in Sixtrack”, these proceedings.
 - [32] S.J. Wretborn, *et al.*, “Study of off-momentum losses at the start of the ramp in the Large Hadron Collider”, September 6th, 2017, CERN-ACC-NOTE-2017-0065, <http://cds.cern.ch/record/2298696>.
 - [33] A. Mereghetti, *et al.*, “SixTrack-Fluka Active Coupling for the Upgrade of the SPS Scrapers,” in *IPAC’13 Proceedings*, Shanghai, China, May 2013, paper WEPEA064.
 - [34] R.B. Appleby, *et al.*, “Fluka Coupling for SPS Scrapers”, these proceedings.
 - [35] P. Hermes, *et al.*, “Symplectic Tracking of Multi-Isotopic Heavy-Ion Beams in SixTrack,” in *IPAC’16 Proceedings*, Busan, Korea, May 2016, paper TUPMW015.
 - [36] F. Schmidt, *et al.*, “SixTrack User Manual,” <http://sixtrack.web.cern.ch/SixTrack/doc/manual/six.html>, 2017.
 - [37] F. Schmidt, *et al.*, “Sixtrack Physics Manual (Dev),” http://sixtrack.web.cern.ch/SixTrack/doc/manual_dev/six.pdf, 2017.
 - [38] R. De Maria, *et al.*, “Sixtrack Physics Manual (DRAFT),” http://sixtrack.web.cern.ch/SixTrack/doc/physics_manual/sixphys.pdf.
 - [39] K. Sjobak, J. Molson, “Compiling, building, and testing SixTrack,” http://sixtrack.web.cern.ch/SixTrack/doc/building_sixtrack/building_sixtrack.pdf.
 - [40] “SixTrack Developer Wiki,” <http://twiki.cern.ch/twiki/bin/view/LHCAtHome/SixTrack>.
 - [41] SixTrack source code repository, <http://github.com/SixTrack/SixTrack>.
 - [42] CDash website, <https://abp-cdash.web.cern.ch/abp-cdash/index.php?project=SixTrack>.
 - [43] SixTrackLib source code repository, <http://github.com/SixTrack/SixTrackLib>.

FLUKA coupling to Sixtrack

E. Skordis, V. Vlachoudis, R. Bruce, F. Cerutti, A. Ferrari, A. Lechner, A. Mereghetti, P.G. Ortega, S. Redaelli, D. Sinuela Pastor
CERN, Geneva, Switzerland

Abstract

SixTrack [1] and FLUKA [2,3] are simulation tools regularly used at CERN to perform LHC collimation studies. Until recently the communication between the two codes was weakly based on the use of file exchange. The present paper describes the current status of the FLUKA – SixTrack coupling, a TCP/IP interface to provide seamless communication between them and to finally behave like a single running program.

Keywords

FLUKA, SixTrack, coupling, collimation losses

1 Introduction

SixTrack is a single particle tracking code for long-term, 6D tracking in high-energy rings, while FLUKA is a general-purpose particle physics Monte Carlo code used in particular for machine protection, design of accelerator components, radiation to electronics (R2E), activation, and collimation studies. These imply energy deposition calculation with description of electro-magnetic (EM) showers as well as account of multi-turn effects and call for a synergy between the two codes. The standard work path for predicting collimation loss maps up to now required a dedicated version of SixTrack complemented with simplified interaction routines [4]. These loss maps, consisting of primary inelastic interactions, were provided to FLUKA to describe the particle shower development by microscopic models. The FLUKA-SixTrack coupling aims at combining the two codes and using each one in the respective domain of application.

1.1 History

The idea of the coupling dates back to 2006, when a first version was created by V. Vlachoudis using a toy tracker (single-turn matrix) communicating with FLUKA through network ports. That system was based on the assumption that the tracking was faster than FLUKA, so emphasis was given on the creation of a dedicated server that was weight balancing the work load of multiple tracker and FLUKA clients. In 2007, S. Gilardoni and R. Bruce developed another solution, in which FLUKA was called as system routine inside IcoSim [5], a MatLab-based tracker featuring a complete 4D tracking implementation (up to sextupoles, with no higher-order multipoles, no skew elements/solenoids/crab cavities...). The coupling code was working through file exchange and was performing a full FLUKA run on each turn. The initialization of FLUKA on each turn was inducing an excessive overload in the tracking time.

The present coupling is based on the FlukaIO library created in 2010 by V. Vlachoudis and D. Sinuela Pastor, i.e. an independent C++ TCP/IP library allowing the communication of any tracking code with FLUKA (or other Monte Carlo code). The first application was performed by F. Cerutti and A. Mereghetti for the SPS scrapers with the use of IcoSim and FLUKA [6]. Thanks to the further work of P. Hermes, A. Mereghetti, P.G. Ortega and V. Vlachoudis in the context of the LHC Collimation Working Group [7], the code was extended to SixTrack with main focus on LHC applications.

2 Motivation and benefits

The coupling defines a clean separation line between the two codes:

- Particle tracking through the accelerator lattice is performed by SixTrack
- Beam-matter interaction is simulated by FLUKA

This allows a realistic multi-turn approach with the state-of-art account of physics processes, leading to more accurate predictions. It avoids rough simplifications in the modelling of complex interactions such as single-diffractive scattering or ion dissociation and fragmentation. It limits human intervention (removing the need to manually check every time file content, adopted units, etc.), as a consequence the overall process is less error-prone. It can handle moving beam-intercepting devices runtime, with the possibility to use any roto-translation of 3D geometries, and it can deal with energy ramping. Last but not least, it can benefit from the regular development of both codes and their tools.

Furthermore, the FLUKA-SixTrack coupling becomes very attractive when dealing with heavy ions and, in perspective, crystal channelling, as discussed in the following.

2.1 Heavy ion beams

Heavy ions when passing through matter, as in the collimator jaws, can undergo both electromagnetic dissociation and nuclear inelastic interactions, typically leading to lighter fragments that can in turn re-interact in the collimator materials. It is not enough to know the probability for generating a given fragment, since its momentum is not fixed. This makes fragments nominally far from the nominal beam rigidity to fall in reality inside the machine acceptance (e.g., tritium ions). Moreover, the cross sections are energy dependent and ionization plays for ions a non-negligible role in changing their energy along their path in matter. The energy loss evaluation needs a treatment significantly more sophisticated than a simplified Bethe formula, as adopted in some tracking codes. Landau fluctuations and Mott corrections have to be taken into account, as well as bremsstrahlung and pair production in addition [8]. Thus, the tracking of heavy ions requires a careful treatment and the coupling intrinsically overcomes all above issues, relying on their proper treatment.

2.2 Crystal channeling

The use of bent crystals for beam manipulation to different purposes, such as collimation, is an option actively pursued and systematically investigated on the LHC [9]. In this context, the development of a microscopic description of the coherent effects featuring their interaction with hadron beams, to be integrated in accelerator simulation tools of wide application, is a fruitful investment. This motivation triggered the recent implementation of crystal channelling in an event generator based on FLUKA that was benchmarked against the UA9 experiment data for proton beams [10]. The present version includes planar channelling of any positively charged particle, crystal torsion and miscut, nuclear interaction suppression, dechanneling, volume reflection and volume capture.

Figure 1 shows a comparison between a typical measurement and the current model, giving an overview of all the most relevant effects.

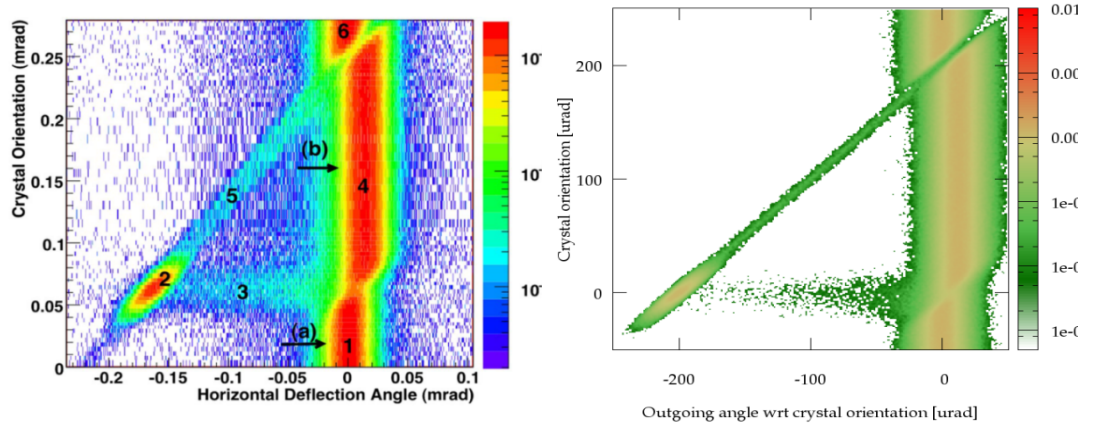


Fig. 1: Comparison of UA9 measurement [11] (left) and simulation (right).

3 Key ingredients

The FLUKA-SixTrack coupling requires a FLUKA input file describing the concerned insertion regions, even if they are restricted to the bare intercepting device of interest. Moreover, if a second simulation stage with FLUKA is required (e.g., to compute energy deposition in sensitive elements), then it is convenient that the same detailed geometries of beam-intercepting devices used in the second simulation stage are used also in the first one. Hence, the FLUKA input file for FLUKA-SixTrack coupling simulations is normally prepared with the FLUKA element database (FEDB) and the Line Builder [12]. These two tools provide a powerful way of building automatically beam line geometries perfectly complying with tracking requirements.

3.1 The FLUKA Element Database (FEDB)

We have been performing FLUKA calculations for the LHC machine since 2004. To optimize the workflow, we have created a dedicated database called “*FLUKA element database*”, which nowadays consists of FLUKA models of relevant objects from all CERN accelerators starting from the LINAC injector up to the LHC (see an example in Fig. 2). It offers flexibility and ease in modelling accelerator components as standalone objects, crucial when adding additional degrees of detail, following the evolution of the design of future devices, or comparing different technical solutions. Moreover, it minimizes errors and duplication of work since all models are stored as common and shared resources under a revision tracking system (SubVersion repository). Lastly, it offers portability, with prompt propagation of updates, as improvements are inherited by all the people concerned.

3.2 The Line Builder (LB)

The Line Builder [12] is a tool developed by A. Mereghetti, which allows to assemble FLUKA geometries of beam lines taking as input the accelerator lattice sequence, magnet strengths and optical parameters. It offers:

- flexibility, through the automatic synchronization with the machine optics, essential ingredient for a sound modelling of the line;
- consistency, protecting against wrong settings of the beam line, e.g. misplacements, wrong orientations, mismatches in magnetic fields and collimator aperture;
- portability, thanks to the automatic use of the last updates of the FEDB.

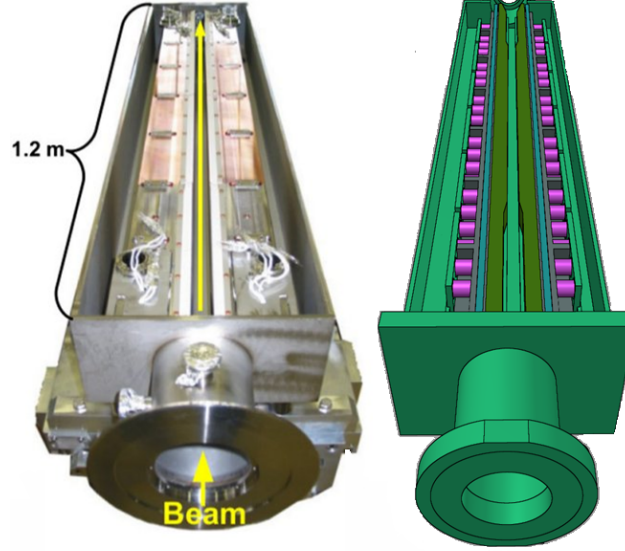


Fig. 2: An LHC collimator (left) and its FEDB model (right).

3.3 Users' settings

FLUKA simulations can be very CPU-intensive for detailed energy-deposition studies in large LHC geometries at TeV beam energies. This is because one has to apply low secondary particle production and transport cuts down to MeV energies or less, in particular for electrons, positrons and photons, and needs many fine-grained scoring meshes, to calculate point-like quantities that are necessary, for instance, to compare to quench levels or stress limits.

For tracking study purposes, CPU time can be quite significantly optimized by adopting high cuts and switching off unnecessary physics processes ($e^-/e^+/\gamma$ production and transport, inelastic collision product generation and transport). Transport thresholds must be carefully set in FLUKA, to properly describe particles which can still propagate in the machine, and hence contribute to the global loss map, while suppressing the local shower development, this way achieving sound results in reasonably short times.

4 Workflow

The coupling is a third-party library to the two codes (FLUKA and SixTrack) that allows the communication through the use of TCP/IP messages. FLUKA and SixTrack run at the same time, in parallel, either on the same computer or different ones, talking to each other over the local network. One or more portions of the accelerator lattice are labelled for transport in FLUKA, whereas the rest is handled by SixTrack (Fig. 3). The beam particles are transported turn by turn by SixTrack throughout the lattice. When they reach a labelled element, they are transferred to FLUKA for transport in its 3D geometry, simulating the interaction with accelerator components. At the end of the FLUKA insert, marked by a geometry interface, particles are sent back to SixTrack. The exchange of particles is performed at run-time, through a network port using the dedicated FlukaIO library. This loop continues until the particles are either lost in the FLUKA portion or in the rest of the machine.

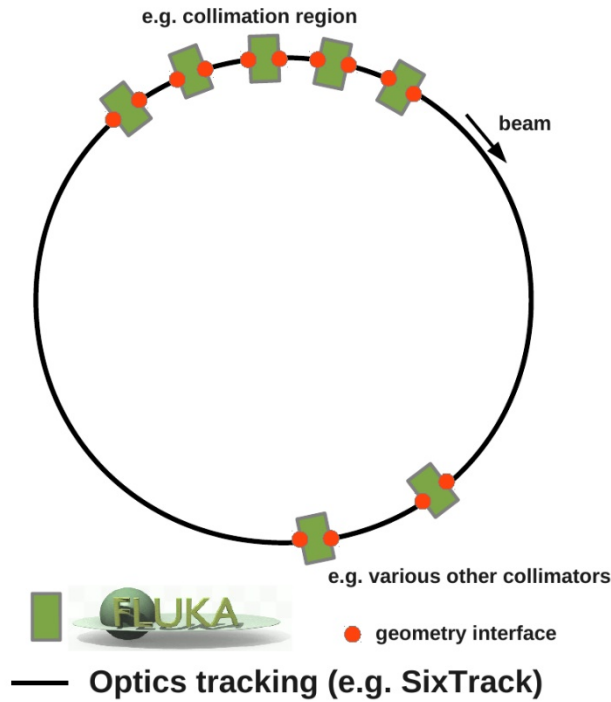


Fig. 3: Schematics of the SixTrack-FLUKA coupling.

5 Present Status

The present version of the code is able to perform the coupling with an accelerator lattice description in either thin or thick lenses. Both 4D and 6D tracking including RF acceleration are supported. More than one insertion and more than one element per insertion can be flagged for coupling. The starting beam distribution is provided by an external generator. Each particle is assigned a unique ID for off-line analyses of chained interactions in different collimators. The aperture checking must be performed on-line, otherwise unphysical particle distributions hitting beam-intercepting devices will be tallied. Hence, this is an essential ingredient for both loss maps around the ring and energy deposition onto beam-intercepting devices. The on-line aperture checking greatly expands the functionalities of the LIMB block native to SixTrack, including also the most common LHC aperture type RECTELLIPSE. Although pre-processing tools are used to prepare most of the necessary input files, it is the user's ultimate responsibility to guarantee the integrity of input information, e.g. the transformations from the local SixTrack reference system to the global one of FLUKA and vice-versa, as well as the length of the FLUKA geometry and the synchronous length declared in input to SixTrack (to correctly handle the longitudinal dynamics in the FLUKA insertions).

Additional features one can benefit from are: the application of dynamic kicks in SixTrack to magnetic elements, e.g. to simulate the raising of magnetic bumps or current ripples in power supplies; moving FLUKA bodies run-time, to simulate beam-intercepting devices with impact parameters varying with time; useful information dumps for monitoring the simulation progress and the evolution of key quantities of the beam being tracked; the use of complex 3D geometries of portions of beam lines, including also magnetic elements with their fields.

5.1 Repository

The necessary material (i.e. the coupling code, with the associated SixTrack and FLUKA user routines, the LB library and the FEDB) is available in a shared project space at

/afs/cern.ch/project/fluka through an access list. Otherwise, every repository (i.e. coupling, FEDB, LB) can be manually checked out by authorized users from the central CERN SVN service:

```
svn co svn+ssh://svn.cern.ch/repos/fedb
svn co svn+ssh://svn.cern.ch/repos/LB
svn co svn+ssh://svn.cern.ch/repos/fluka_coupling
```

6 LHC applications

It is often the case that a qualitative comparison of loss maps with Beam Loss Monitor (BLM) signal patterns is done to provide a first feedback on the loss locations and amplitude, especially when verifying the running configuration of the collimation system for data taking. However, for a quantitative benchmarking of the level of accuracy the simulation chain provides, a second especially meticulous step is required after the tracking, to proceed into full shower development simulations and evaluation of measurable quantities such as the BLM recorded dose.

Fig. 4 shows the FLUKA geometry of the LHC IR7 Long Straight Section (LSS) assembled with the use of the Line Builder and the FEDB, with which energy deposition simulations were carried out. For the collimator jaws and tanks, the same model is used for this second step as for the previous SixTrack-FLUKA coupling step, boosting the consistency between the two simulation steps. In order to properly evaluate the BLM signals and other quantities of interest, various machine elements have also been modeled in detail in FLUKA, such as the LSS warm dipoles and quadrupoles, passive absorbers, collimator supports, BLMs, tunnel walls, Super Conducting (SC) dipoles and quadrupoles, etc.

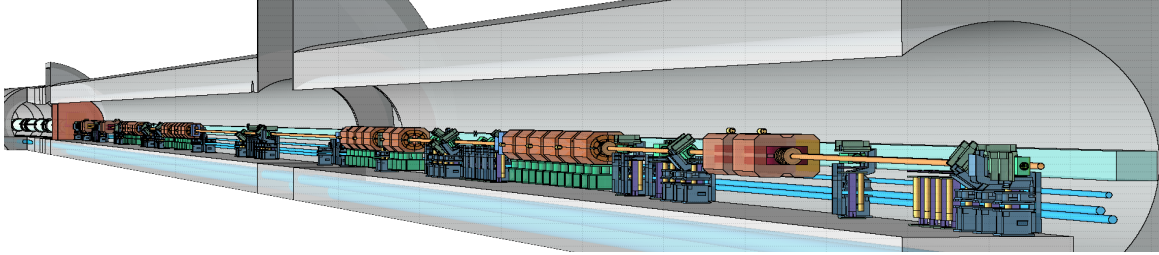


Fig. 4: LHC IR7 LSS FLUKA geometry visualised with Flair.

Fig. 5 shows the comparison between simulated BLM signals and measurements taken during the 2013 proton collimation quench test carried out at 4 TeV beam energy [13, 14]. The source term, i.e. the distribution of inelastic interactions at the collimators (including single diffractive events), was generated with the SixTrack-FLUKA coupling, as an alternative to the older approach using SixTrack with its embedded interaction models. In this case the two simulation approaches provide alike results, since the original proton loss distributions were pretty similar and BLMs are sensitive to the development of the secondary particle cascades, described by FLUKA in both simulation set-ups. An excellent agreement can be observed between simulated and experimental BLM signals for more than one hundred BLMs in the IR7 LSS with signals spanning over a few orders of magnitude, apart from an underestimation tendency in the Beam 2 active absorber region on the left end.

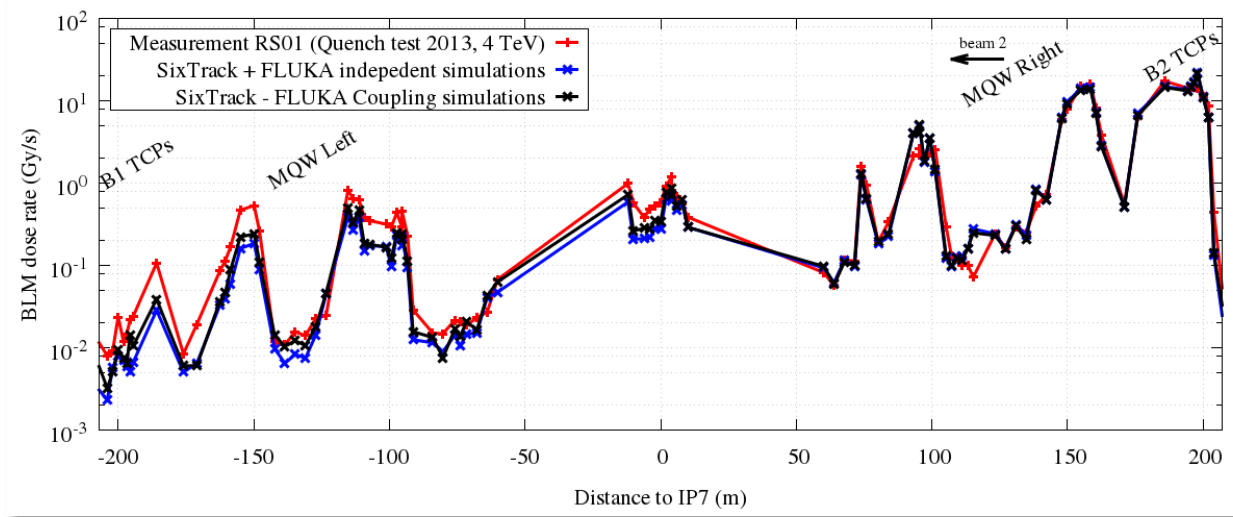


Fig. 5: Benchmarking of the simulated BLM signals against measurements taken during the 2013 proton collimation quench test at 4 TeV beam energy.

Another usage of this two-step method is the evaluation of the BLM response per proton lost in a specific collimator. By considering only the losses occurring on the latter, it is possible to calculate the signal produced in the BLM corresponding to the same collimator (primary signal) as well as the induced crosstalk signal recorded by the upstream and downstream BLMs (Fig. 6). Table 1 shows an example of a simulated BLM response matrix for losses on the vertical and horizontal primary collimators in IR7 that can be used in order to obtain the loss plane decomposition from measured signals. Table 2 provides the response matrix of losses on the horizontal and vertical tertiary collimators, where the apparent increase in response mainly originates from the different jaw material (Tungsten instead of Graphite) as well as the different BLM position with respect to the impacted jaws.

The tables highlight that for losses on different collimators, even of the same family, a different primary BLM response is to be expected, as well as that the crosstalk plays a very significant role. For instance, we see that the response of the BLM at the TCP.C (horizontal primary collimator) is increased by 80% if losses move from the horizontal to the vertical plane, first impacting the preceding collimator. Consequently, the actual loss plane combination cannot be directly identified by the measured BLM signals, but implies the response matrix application.

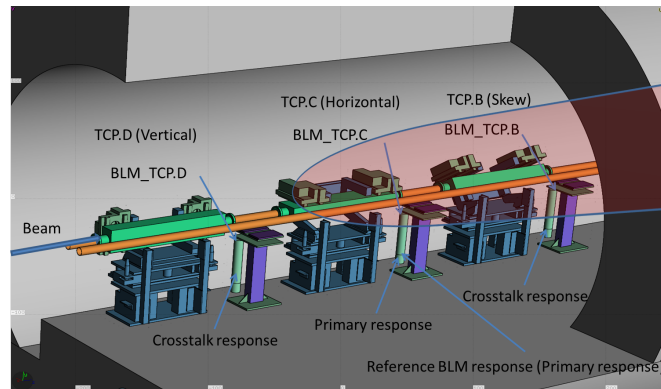


Fig. 6: FLUKA model of the IR7 vertical (TCP.D), horizontal (TCP.C) and skew (TCP.B) collimators, with a figurative proton interaction in the TCP.C inducing primary and crosstalk BLM responses.

Table 1: BLM response matrix for the IR7 primary collimators at 3.5 TeV proton beam energy. Primary and crosstalk responses appear in **Bold** and *Italic*, respectively. Values are normalised to $4.6 \cdot 10^{-12}$ Gy/p, i.e. the primary response of BLM_TCP.C.

BLM Collimator	BLM_TCP.D	BLM_TCP.C	BLM_TCP.B
TCP.C (Horizontal)	<i>0.01</i>	1	<i>2.53</i>
TCP.D (Vertical)	0.58	<i>1.80</i>	<i>2.13</i>

Table 2: BLM response matrix for the IR1 tertiary collimators (TCT) at 3.5 TeV proton beam energy. Primary and crosstalk responses appear in **Bold** and *Italic*, respectively. Values are normalised to $4.6 \cdot 10^{-12}$ Gy/p, i.e. the primary response of BLM_TCP.C in Table 1.

BLM Collimator	BLM_H1	BLM_V1
TCT_H1 (Horizontal)	7.2	<i>1.15</i>
TCT_V1 (Vertical)	<i>0.4</i>	3.25

7 Summary and perspectives

The integrated FLUKA/SixTrack coupling offers an accurate tool for multi-turn simulations of cleaning systems. It benefits from sophisticated physics models, particularly relevant for single-diffractive scattering and ion interactions. With the use of appropriate transport settings in FLUKA, the CPU time can be kept reasonably low, even when simulating complex accelerators like the LHC. The FlukaIO library, providing the two codes with the communication infrastructure, is flexible enough to be linked with other tracking and Monte Carlo codes. Currently it is operational with FLUKA for SixTrack, IcoSim and IcoSim++.

Further improvements are represented by the operational integration in FLUKA of the crystal channeling event generator, the use of beam-beam collision products as embedded source term (allowing to track diffractive and elastically scattered protons or ion fragments generated at the experimental interaction points of a collider), the replacement of the FLUKA user routines by dedicated input cards, and the possible re-introduction of the external server mechanism to handle anisotropic computation times if needed.

References

- [1] R. De Maria *et al.*, these proceedings
- [2] G. Battistoni *et al.*, “Overview of the FLUKA code”, *Annals of Nuclear Energy* 82, 10-18 (2015)
- [3] A. Fassò, A. Ferrari, J. Ranft, and P.R. Sala, “FLUKA: a multi-particle transport code”, CERN-2005-10 (2005), INFN/TC_05/11, SLAC-R-773
- [4] R. Bruce *et al.*, these proceedings
- [5] H. Braun *et al.*, “Collimation of heavy ion beams in LHC”, *Proceedings of the 2004 European Particle Accelerator Conference*, Lucerne, Switzerland, 551 (2004); N. Holden, “Development of the ICOSIM Program and Application to Magnetised Collimators in the LHC”, Technical Report CERN-AB-Note-2008-054, CERN, Geneva, Dec 2008. <http://cds.cern.ch/record/1151294>
- [6] A. Mereghetti, “Performance Evaluation of the SPS Scraping System in View of the High Luminosity LHC”, PhD Thesis, University of Manchester, UK (2015)
- [7] P. Garcia Ortega, “Status of SixTrack-FLUKA coupling”, Talk at the LHC Collimation Working Group #181, CERN, Geneva, Oct 2014. <https://indico.cern.ch/event/343152>
- [8] U.I. Uggerhøj, “Penetration Phenomena at Relativistic Energies”, *Matematisk-fysiske Meddelelser* 52, 699 (2006)
- [9] W. Scandale *et al.*, “Observation of Channeling for 6500 GeV/c protons in the crystal assisted collimation setup for LHC”, *Phys. Lett. B* 758, 129-133 (2016)
- [10] P. Schoofs, “Monte Carlo Modelling of Crystal Channelling at High Energies”, PhD Thesis, EPFL, Lausanne, Switzerland (2014)
- [11] W. Scandale *et al.*, “High-efficiency Volume Reflection of an Ultrarelativistic Proton Beam with a Bent Silicon Crystal”, *Physical Review Letters* 98, 154801 (2007)
- [12] A. Mereghetti *et al.*, “The Fluka LineBuilder and Element DataBase: Tools for Building Complex Models of Accelerator Beam Lines”, *proceedings of IPAC12*, New Orleans, US (2012)
- [13] B. Salvachua *et al.*, “Collimation quench test with 4 TeV proton beams”, CERN-ACC-NOTE-2014-0036 (2014)
- [14] E. Skordis *et al.*, “Impact Of Beam Losses In The LHC Collimation Regions”, *IPAC15*, Richmond, USA

Status of MERLIN

H. Rafique^{1 2 3}, *R. B. Appleby*^{2 3}, *R. J. Barlow*¹, *J. G. Molson*⁵, *S. Tygier*^{2 3}, *A. Valloni*^{2 4}

¹University of Huddersfield, Huddersfield, UK, ²University of Manchester, Manchester, UK,

³Cockcroft Institute, Daresbury, UK, ⁴CERN, Geneva, Switzerland,

⁵LAL, Univ. Paris-Sud, CNRS/IN2P3, Université Paris-Saclay, Orsay, France

Keywords

MERLIN; collimation; HL-LHC; tracking.

1 Introduction

MERLIN is an accelerator physics library written in C++, created by Nick Walker at DESY in 2000, to study the International Linear Collider [1] (ILC) beam delivery system ground-motion [2]. Later, the main linac and damping rings were added, [3] necessitating wakefield, collimation, and synchrotron radiation processes [4]. As the ILC is an electron linac, the TRANSPORT maps were deemed acceptable for particle tracking. Later, Andy Wolski added synchrotron functionality, including a module to calculate the Courant-Snyder parameters, closed orbit and dispersion, and symplectic integrators for many turn simulations.

The current loss map tool for the LHC is SixTrack [5], a particle tracking code that was updated to include the K2 [6] scattering routines for collimation [7] and has also been combined with the FLUKA [8].

It was decided to update MERLIN to include the requirements for a complementary simulation of the LHC and future collimation system. MERLIN is written in C++ making it modular, and easy to modify. It offers thick lens tracking, an on-line aperture check, and a number of physics processes. The scattering physics has recently been updated to include more advanced proton-nucleon elastic and single diffractive scattering [9]. These updates, ensure that MERLIN offers a fast, accurate, and future-proofed tool for ultra-relativistic proton tracking, collimation, and a robust hollow electron lens process [10].

MERLIN has been used for simulating loss maps for the existing and future upgrades to the LHC collimation systems [11, 12]. The source code is now available on GitHub [13], and is described in detail in [9] and [10]. In this article we describe the features of MERLIN that are used in collimation studies. Dedicated articles on MERLIN's composite materials and hollow electron lens are included in these proceedings [14, 15].

2 MERLIN

For a typical loss map simulation, MERLIN must be given a set of input files that describe the accelerator lattice, the machine aperture and the collimator settings. Then a beam of particles is created, based on the accelerator optics and distribution requested, and tracked around the lattice. When a particle hits a collimator, the scattering through the material is simulated. Finally, if a particle loses sufficient energy in a scatter, or hits a non collimator component, its loss is recorded and a loss map is generated.

The user may define their accelerator in the form of an `AcceleratorModel`, using the `MAD Interface` class to read a standard MADX [16] TFS table. Further input files may be read by the `ApertureConfiguration` class to define the apertures of the accelerator, and the `Collimator Database` to set up collimators. The user may then calculate the lattice functions of the accelerator, and in turn use these to define a beam, which leads to the construction of a `ParticleBunch` that is matched to the accelerator at the desired position. A particle tracker may be constructed, selecting from either TRANSPORT or SYMPLECTIC integrator sets, and physics processes may be attached to it. Finally, the user

may run the tracking simulation, and create outputs using one of the existing output functions, or define their own. This standard flow of data between MERLIN's main classes for a collimation simulation is shown in Fig. 1.

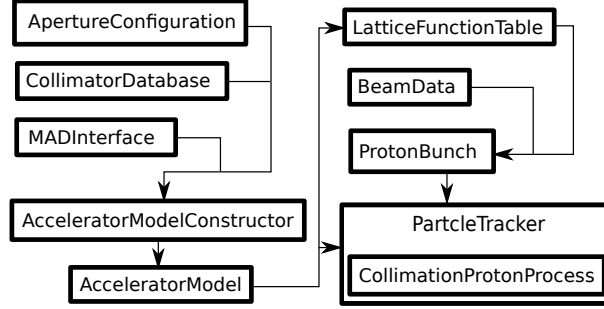


Fig. 1: The data flow of a typical collimation simulation using MERLIN

2.1 Lattice setup

MERLIN stores an accelerator as an `AcceleratorModel` object. This requires three input files for the purpose of LHC and HL-LHC collimation, the first for an accurate model of the individual components of the accelerator, the second to define the apertures of the machine, and the third to define collimator parameters. In order to translate these inputs into a MERLIN `AcceleratorModel`, a number of input interfaces and configuration utilities are required.

The `AcceleratorModel` is an ordered vector of `AcceleratorComponents`, which is used by the `ParticleTracker` to set the integrators that describe the paths taken by individual particles as they travel through the elements. Each `AcceleratorComponent` contains an EM field, a geometry, an aperture, and a wake potential object. Special cases exist, for example a `Collimator` also contains a jaw material which is required for scattering. Figure 2 shows the inheritance of the standard `AcceleratorComponents` (including some internal data types prefixed with `TAcc`), both `Collimator` and `Hollow ElectronLens` elements are tracked as, and therefore derived from, the `Drift` component (a vacuum pipe with no field).

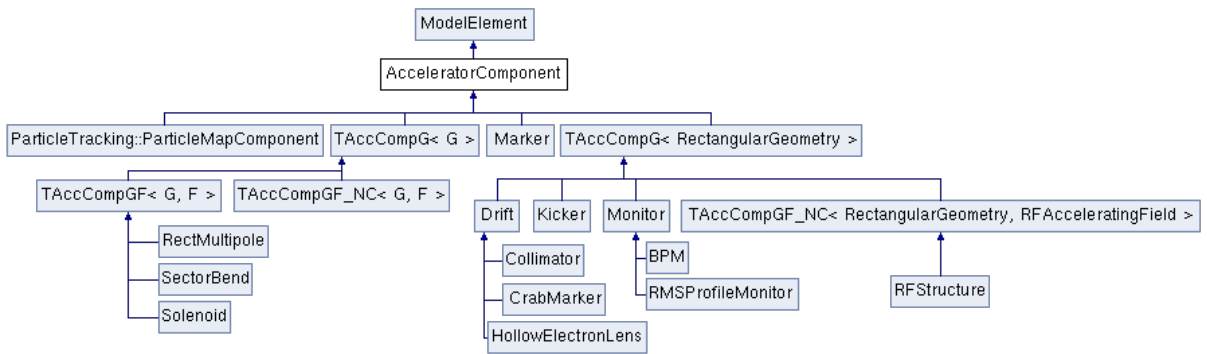


Fig. 2: Inheritance diagram of the accelerator components currently available in MERLIN

A standard MADX thick lens TFS table is passed through the `MADInterface` class to extract the `AcceleratorComponents`. `MADInterface` reads the column headers so that as long as the required parameters are present, they do not need to be in a fixed order. It iterates through each element and creates the appropriate `AcceleratorComponent`, appending it to the `AcceleratorModelConstructor`, a standalone class used in `MADInterface`.

MERLIN contains all the standard components required for modelling typical synchrotrons, detailed in Table 1. Any element may be treated as a drift using the `MADInterface::TreatTypeAsDrift()` function, though this is unwise for certain magnets. MERLIN currently handles a small number of elements as drifts as standard because there is no integrator to perform the expected function, or the expected function cannot be performed using an integrator (for example the hollow electron lens).

Table 1: Most common accelerator components and their MADX TFS and MERLIN names. For cases such as MULTIPOLE the appropriate element is selected based on the parameters in the lattice. For RBEND, MERLIN uses a SectorBend with appropriate pole face rotations.

Component	MADX name	MERLIN name
Vacuum pipe	DRIFT	Drift
RF cavity	LCAV	SWRFStructure
RF cavity	RFCAVITY	SWRFStructure
Collimator	RCOLLIMATOR	Collimator
Collimator	ECOLLIMATOR	Collimator
Collimator	COLLIMATOR	Collimator
Rectangular dipole	RBEND	SectorBend
Sector dipole	SBEND	SectorBend
Multipole	MULTIPOLE	Sextupole, Octupole
Vertical corrector	YCOR	YCor
Vertical kicker	VKICKER	YCor
Horizontal corrector	XCOR	XCor
Horizontal kicker	HKICKER	XCor
Quadrupole	QUADRUPOLE	Quadrupole
Skew quadrupole	SKEWQUAD	SkewQuadrupole
Solenoid	SOLENOID	Solenoid
Hollow electron lens	-	HollowElectronLens
Sextupole	SEXTUPOLE	Sextupole
Octupole	OCTUPOLE	Octupole
Skew sextupole	SKEWSEXT	SkewSextupole
Monitor	MONITOR	BPM, RMSProfileMonitor
Marker	MARKER	Marker

2.2 Apertures

MERLIN features on-line aperture checking. Particles' positions are checked against the local aperture at each element as they are tracked through the lattice. The aperture can either be used to stop particles or to trigger the scattering routine in order to simulate the passage through matter.

MERLIN contains a number of aperture shapes, as shown in Fig 3. The basic apertures are defined by up to four geometric parameters, e.g. radius for `CircularAperture` and half-width and half-height for `RectangularAperture`. All apertures contain a `PointInside()` function, which takes the spatial coordinates of a particle (x, y, z) , and checks if those coordinates are inside the aperture.

MERLIN provides the `InterpolatedAperture` set of classes, which allow the geometric parameters to vary along the length of an element. This is useful for describing the beam pipe around the interaction points in the LHC.

For collimator scattering, MERLIN has the `CollimatorAperture` family of classes. During collimation, if the particle hits the aperture in an element that is a `Collimator` and has a `CollimatorAperture`, then instead of being immediately lost, it is passed to a scattering routine. This collimator scattering is described in section 2.7.

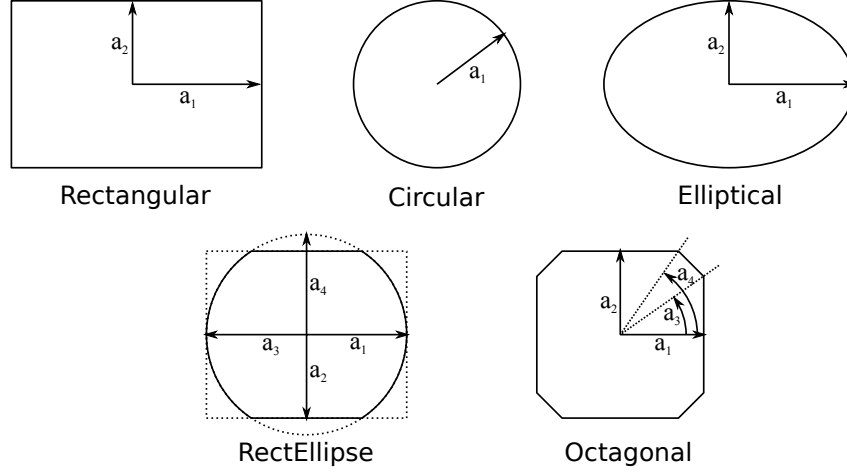


Fig. 3: Aperture shapes and geometric parameters

The apertures for a lattice can be read in from a TFS file using the `ApertureConfiguration` class. This reads an input table containing the shape and geometric parameters of the apertures, and attaches the appropriate aperture classes to the lattice elements.

2.3 Particle Bunch

MERLIN uses three coordinate pairs, (x, x') , (y, y') , (ct, δ) , to define a particle as a six dimensional vector \mathbf{p} , as shown in Equation 1. x and y are the transverse horizontal and vertical coordinates, x' and y' are the corresponding angles, δ is the longitudinal energy offset, and ct is the displacement from the reference position in s .

$$\mathbf{p} = \begin{pmatrix} x \\ x' \\ y \\ y' \\ ct \\ \delta \end{pmatrix}. \quad (1)$$

MERLIN stores particles as `PSVectors`, a class that contains the particle coordinate vector as components, as well as a number of other variables, all of which are detailed in Table 2.

The `ParticleBunchConstructor` class is used to create an initial bunch matched to the machine lattice functions at any chosen injection position. In order to do this a `BeamData` object must be created and fed to the bunch constructor.

`BeamData` provides a data structure for definition of the 6D beam phasespace. Using the `LatticeFunctionTable` the user may define parameters at the injection position, which are fed to the `ParticleBunchConstructor`. The components of `BeamData` are shown in Table 3.

MERLIN provides a number of bunch distributions, all of which are stored in the `ParticleBunchConstructor`. The majority are described in [9]. When using the constructor the bunch is matched to the lattice functions at the position of creation. The user may specify the construction of the bunch via an input file.

For the study of the HEL in the LHC and HL-LHC, a HEL halo distribution was created. This is a simple halo bunch that is populated between $\sigma_{x_{min}}$ and $\sigma_{x_{max}}$ in x , and $\sigma_{y_{min}}$ and $\sigma_{y_{max}}$ in y , thus a matched halo distribution in xy phasespace is generated, as shown in Fig. 4. The minima and maxima

Table 2: Components of the PSVector class

Accessor	Component	Index
x	x	0
xp	x'	1
y	y	2
yp	y'	3
ct	ct	4
dp	δ	5
type	Type of last particle scatter	6
s	Location in lattice	7
id	Individual particle ID	8
sd	Single diffractive flag	9

Table 3: Components of the BeamData class. ¹ Must be specified. ² units are dependent on the type of distribution selected in the ParticleBunchConstructor.

Component(s)	Parameter(s)
x0, xp0, y0, yp0, ct0, dp0	Beam centroid ¹
beta_x, beta_y, alpha_x, alpha_y	Lattice functions ¹
emit_x, emit_y	Emittances ¹
sig_dp	Relative energy spread
sig_z	Bunch length
p0	Reference momentum ¹
c_xy, c_xyp, c_xpy, c_xpyp	x-y coupling
Dx, Dxp, Dy, Dyp	Dispersion ¹
charge	Particle charge ¹
min_sig_x, max_sig_x	Minimum and maximum beam size in x (in σ) ²
min_sig_y, max_sig_y	Minimum and maximum beam size in y (in σ) ²
min_sig_z, max_sig_z	Minimum and maximum bunch length ²
min_sig_dp, max_sig_dp	Minimum and maximum energy deviation ²

are specified using the BeamData class as discussed previously. All other coordinates are matched to regular beam parameters and the optics of the machine at the point of injection in the simulation.

2.4 Integrators

Tracking of particles through each lattice element is performed by integrators. An integrator takes the vector of coordinates of each particle at the beginning of an element and transforms it to the coordinates at the end. A specific integrator is used for each element type and an integrator set will contain an integrator for each common element.

MERLIN contains multiple inbuilt integrator sets including TRANSPORT, which uses second order transport maps and SYMPLECTIC, which defines symplectic integrators for each element. These can be selected using the SetIntegratorSet() method of the tracker. It is also possible to add additional integrators or override individual integrators within a set.

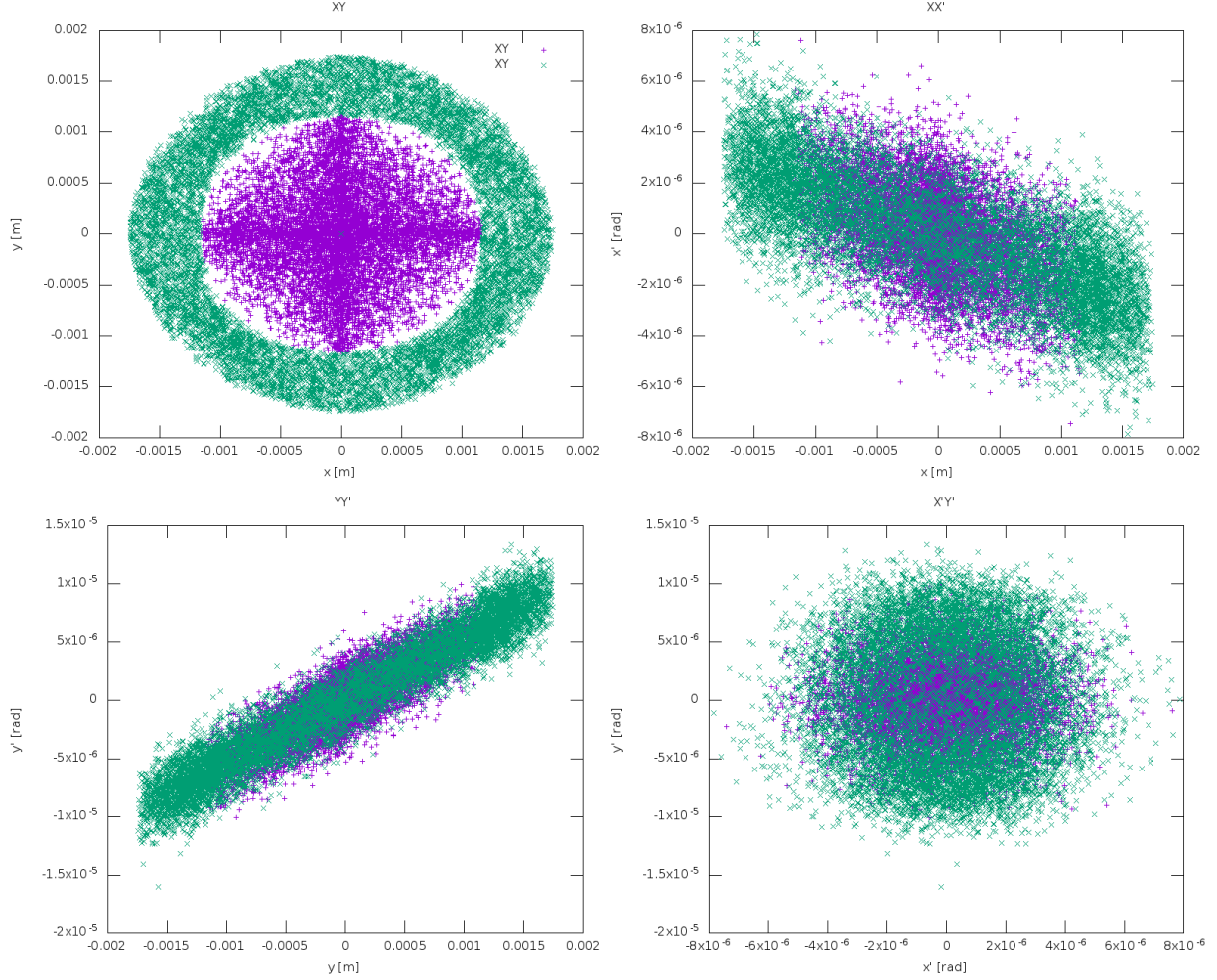


Fig. 4: HEL halo distribution in xy , xx' , yy' , and $x'y'$ phasespace. Purple points are a ‘core’ bunch populated between $0-4\sigma_x$ and σ_y , green points are a ‘halo’ bunch populated between $4-6\sigma_x$ and σ_y . This bunch is created at an ‘injection’ position of HEL in the nominal LHC.

2.5 Synchrotron Motion

Radiation damping and beam acceleration provide another mechanism for particle loss. The off-momentum collimation insertion in IR3 of the LHC is designed for this purpose. The RF bucket, and synchrotron motion in MERLIN is demonstrated in Fig. 5, which shows a Poincaré section, the phasespace over multiple turns, from an LHC simulation with collimation disabled.

Figures 6 and 7 show the same synchrotron motion, but now with collimation enabled, first with only the IR7 transverse primary collimators, and then with both IR7 and the IR3 longitudinal primary collimators. It can be seen that the IR3 collimation region can be used to tightly control energy spread.

2.6 Lattice Functions

The Courant-Snyder parameters, along with the closed orbit and dispersion, give a description of the beam envelope and linear optics around the accelerator lattice. They are useful for confirming that the simulation code has a correct model of the accelerator lattice and accurate modelling of particle dynamics. They are also needed for setting up collimator jaw positions and initial beam parameters.

For collimator jaw openings we use units of σ , which is proportional to the RMS beam emittance

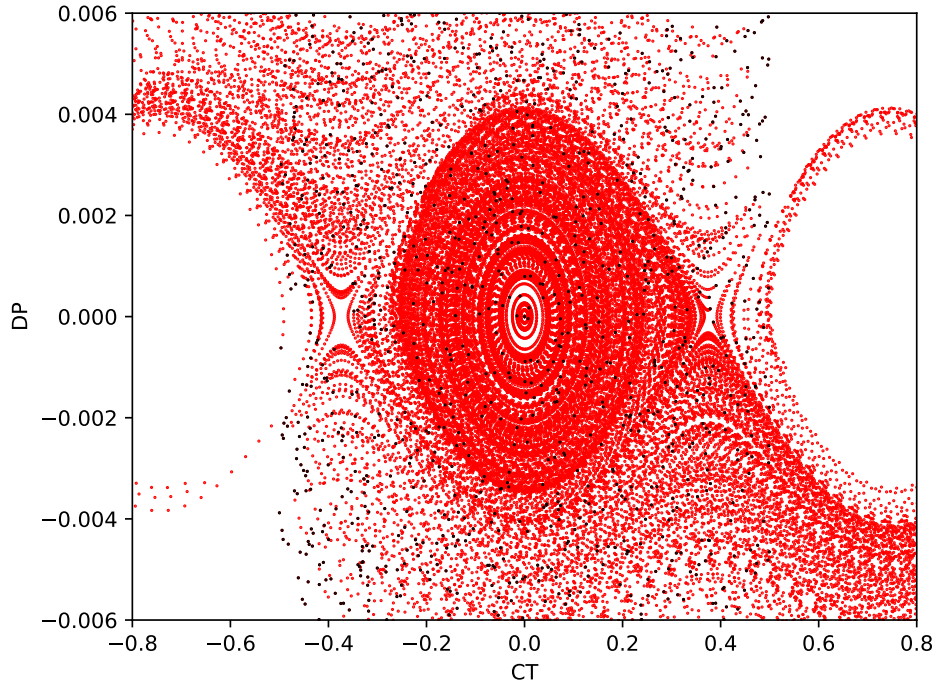


Fig. 5: Poincaré section in ct, δ phasespace of a large initial distribution (black) over 100 turns in the LHC (red), showing RF bucket and synchrotron motion.

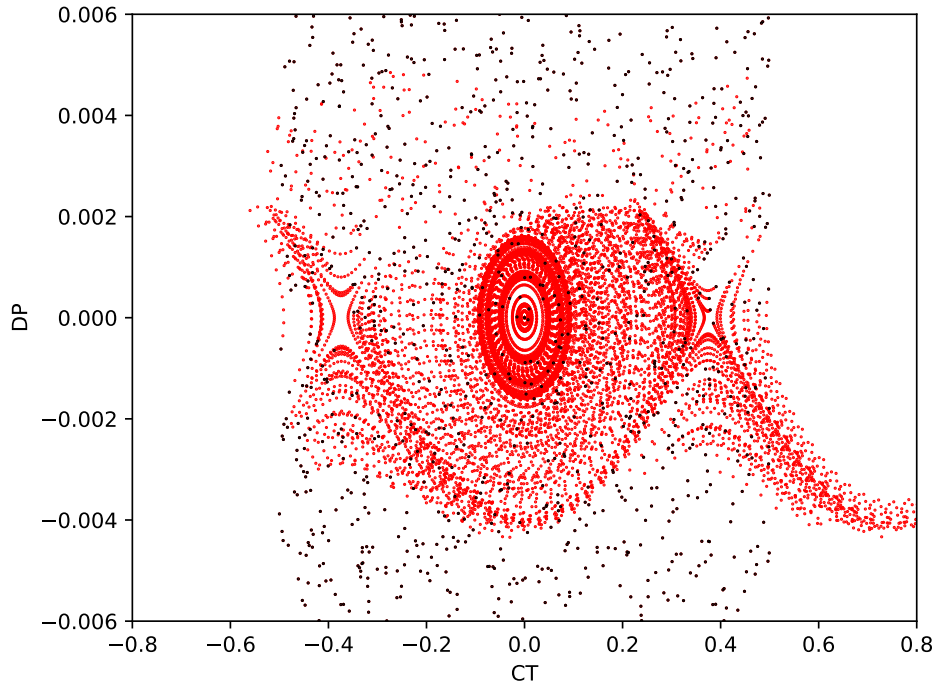


Fig. 6: Poincaré section in ct, δ phasespace of a large initial distribution (black) over 100 turns in the LHC (red) with collimation enabled and the IR7 transverse TCPs in place.

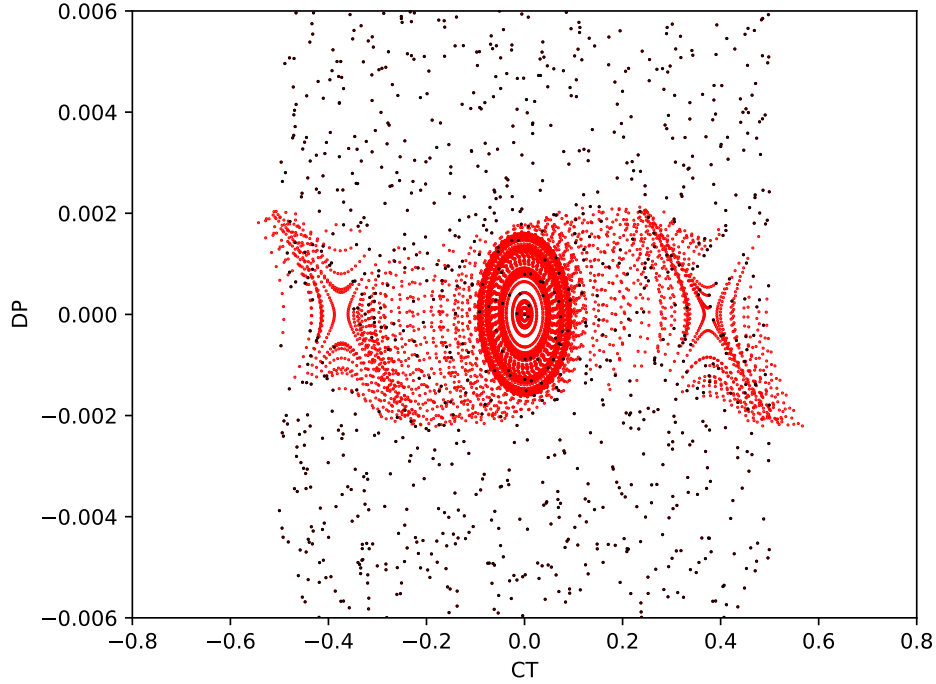


Fig. 7: Poincaré section in ct, δ phasespace of a large initial distribution (black) over 100 turns in the LHC (red), with collimation enabled and both the IR7 transverse and IR3 longitudinal TCPs in place.

and beta function on the given plane at the requested position in the lattice. For example in the x plane $\sigma_x = \sqrt{\beta_x \epsilon_x}$ (when dispersion is zero). This means that when setting collimator apertures we require the lattice functions. Changes in optics – for example the beta squeeze at the experiments – cause a change in the position of the collimator jaws.

MERLIN calculates the lattice functions by tracking particles, rather than by the transfer matrix methods found in optics codes. The `LatticeFunctionTable` class takes the `AcceleratorModel` and beam energy to first find the closed orbit and then calculate the lattice functions.

The closed orbit is found iteratively: a set of particles with small offsets in each phasespace coordinate is tracked through the lattice and a transfer matrix is calculated from the final coordinates. From this an approximate closed orbit is found, and a new iteration is performed around it. This is repeated until the closed orbit converges.

The lattice functions are then found by again tracking a set of particles with offsets through the lattice, recording their positions after every element. From these coordinates the lattice functions at each element can be calculated.

Figure 8 shows the β function and horizontal dispersion around the full ring for the round 15 cm squeezed HL-LHC optics. The increased β in the arc adjacent to the high luminosity experiments due to the ATS optics is clearly visible. Figure 9 shows zoomed views of the β function and horizontal dispersion at each of the 4 experiments.

The optics functions found by MERLIN using tracking can be compared to those found using MAD-X’s matrix methods. This is useful to validate the tracking model in MERLIN. Figure 10 shows the β and α functions from the two codes, and the difference between them. The greatest difference is in the inner triplets where the β function deviates by 25 mm in 20 km.

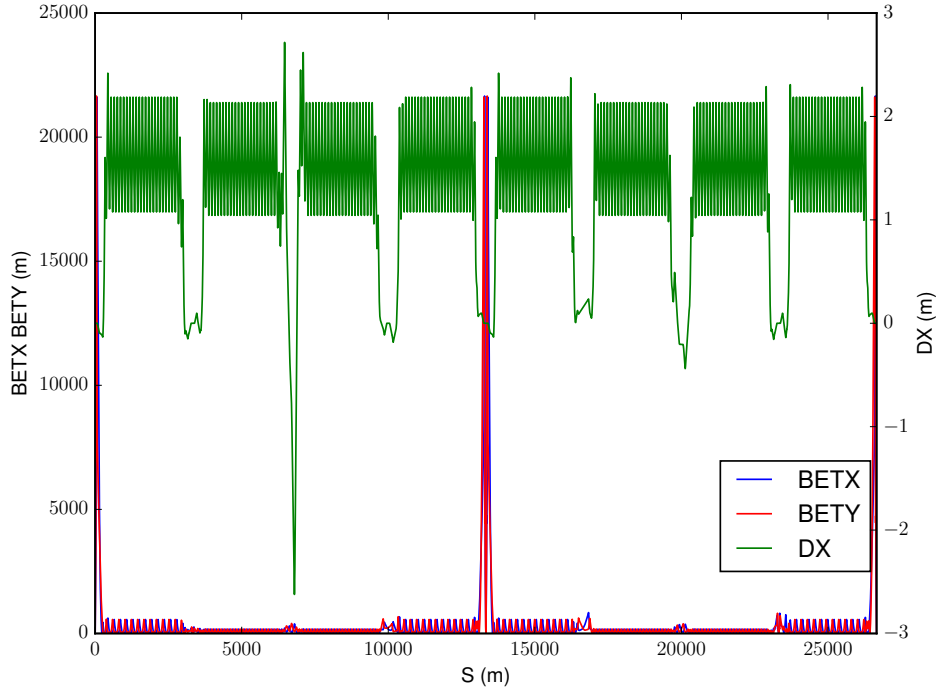


Fig. 8: β -functions and dispersion for HL-LHC ring, 15 cm round optics

2.7 Collimation

The `CollimatorDatabase` is used to construct collimator apertures using an input file. The jaw half gaps, rotation angle, tilt, and material are defined in the input file and set accordingly when read by the `CollimatorDatabase` class.

The modular approach to collimation allows the user to override the definitions of any aspect.

For optimisation the `CrossSections` class calculates and stores all cross sections for a given material. These cross sections are called by the `ScatteringProcess` classes when performing point like scattering, and in `ScatteringModel::PathLength()` to retrieve the total mean free path λ_{tot} . By using this class to compute and save the cross sections, MERLIN minimises computation time at the cost of an inexpensive amount of memory. `CrossSections` stores the advanced `ppElasticScatter` and `ppDiffractiveScatter` classes (for more details on these classes see [9]), allowing access to them during the collimation processes.

`ScatteringProcess` is an abstract base class for individual point-like scattering processes. It contains a pointer to the `Material` and `CrossSections` classes, the process cross section, the beam energy, and two functions: `Configure()` and `Scatter()`.

MERLIN contains a number of `ScatteringProcesses`, including the SixTrack-like variants (based on K2 scattering); those currently available are shown in Fig. 11.

The user can select from several predefined `ScatteringModels`. `ScatteringModelMerlin` provides the full scattering physics as described in [17]. `ScatteringModelSixTrack` provides a model based on the K2 scattering found in SixTrack. `ScatteringModelSixTrackIoniz`, `ScatteringModelSixTrackElastic`, `ScatteringModelSixTrackSD` provide hybrid models useful for testing the individual scattering processes independently. Table 4 shows combinations of processes in each model. It is also possible for the user to create a `ScatteringModel` and customise it by adding their required processes.

The `ScatteringModel` class contains the functions required for performing collimation. A pre-

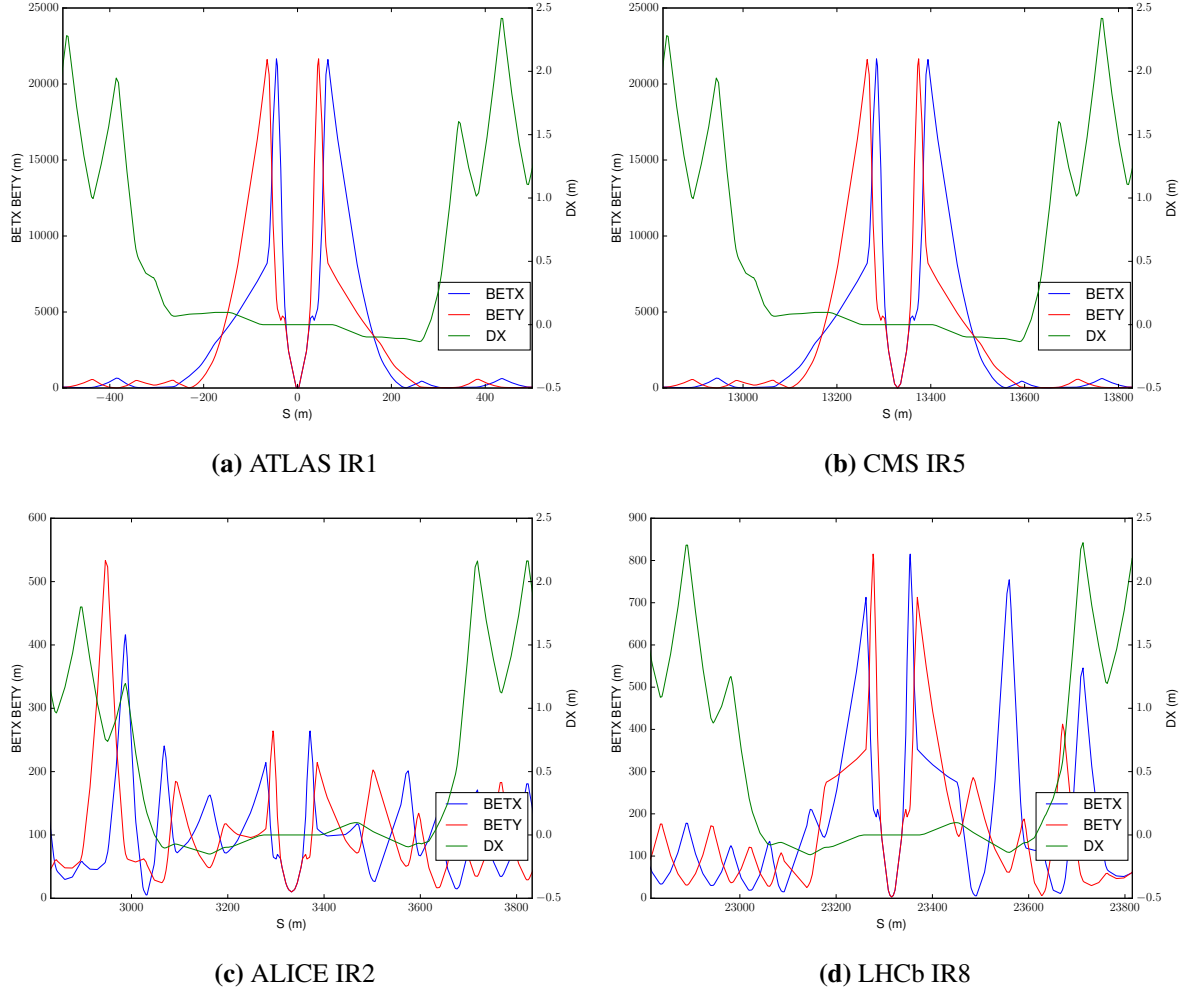


Fig. 9: β -functions and dispersion for HL-LHC experiments, 15 cm round optics

Table 4: Preset combinations of `ScatteringProcesses` and ionisation in MERLIN. ST refers to the SixTrack-like process, M to the MERLIN process, and all combinations include an inelastic process.

Process	SixTrack	SixTrackElastic	SixTrackSD	SixTrackIoniz	Merlin
Rutherford	ST	ST	ST	ST	M
pn Elastic	ST	M	ST	ST	M
pN Elastic	ST	M	ST	ST	M
Single Diffractive	ST	ST	M	ST	M
Ionisation	ST	ST	ST	M	M

defined or user-created combination of `ScatteringProcesses` may be used, and are handled by the `ScatteringModel` in order to compute cross sections, path lengths, and perform bulk (ionisation and MCS) and point-like scattering. The `ScatteringProcess` must be attached to the `CollimateProton` Process.

The path length is the average distance a proton travels through a material before colliding with a material nucleus, it is calculated in the `PathLength()` function for each proton at each iteration of scattering, using the mean free path.

As a proton travels through a material it collides with electrons, these collisions may result in

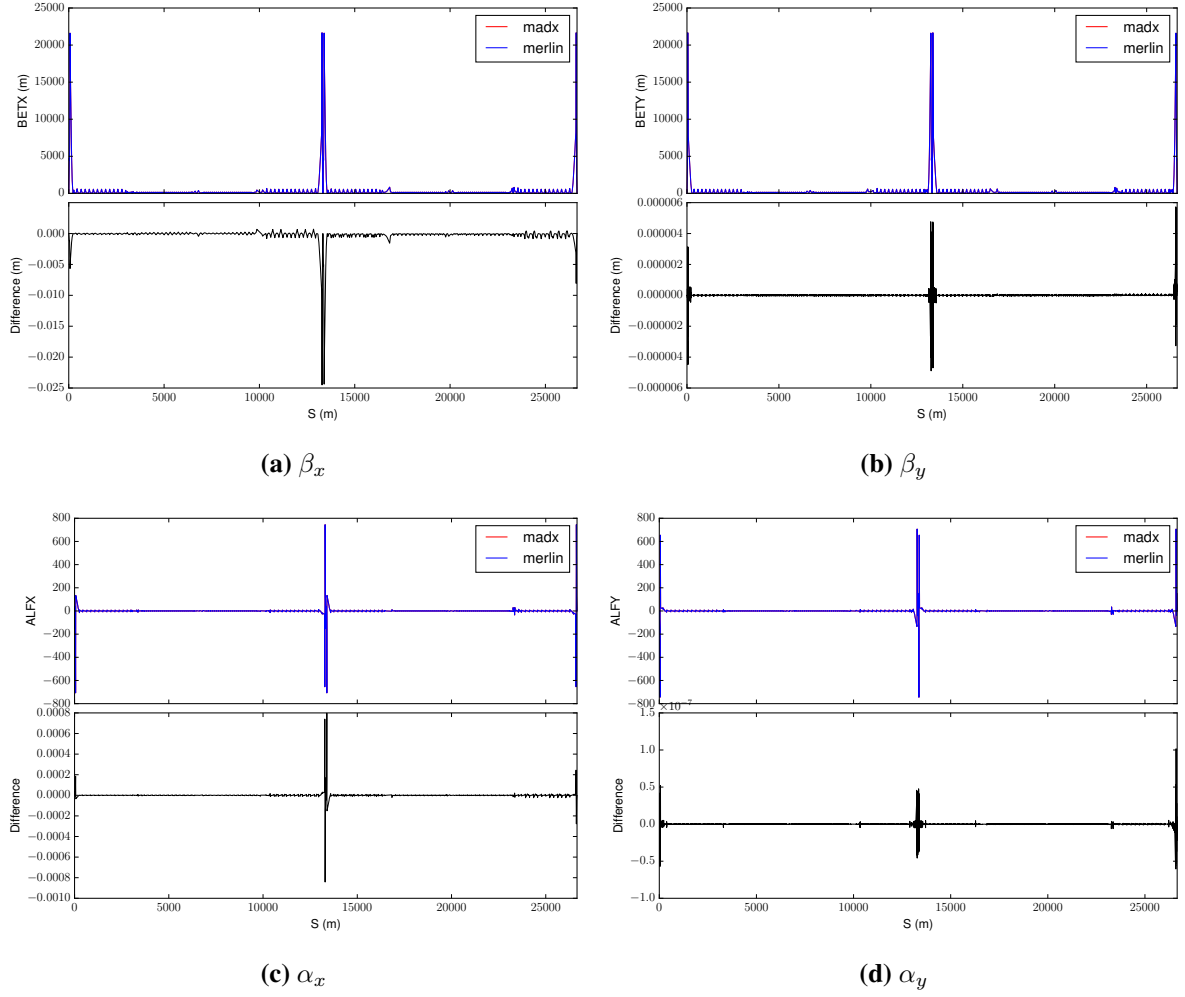


Fig. 10: Difference between MERLIN and MAD-X for β and α -functions for full ring 15 cm round optics.

the removal of electrons, and a loss in proton energy. The interaction is defined by the Bethe-Bloch equation [18]. MERLIN offers an overloaded `EnergyLoss()` function to calculate the energy loss that takes place due to ionisation in a material. The basic function performs energy loss according to the Bethe-Bloch equation [18]. The overloaded function is a more complete treatment of the energy loss due to ionisation using higher-order corrections (considered in [9]). In summary, MERLIN only adds the effects that are relevant to LHC energies: the effect due to the dielectric polarisability of solid materials, the Mott correction which is an enhancement from close collisions due to spin, and the finite size correction taking into account the size and structure of the proton. As well as these corrections, the energy spread of the outgoing proton is sampled using the Landau distribution, which is a more accurate representation of the physical effect [18].

A proton travelling through a material will perform many small-angle elastic scatters from the electrons and nuclei, known as multiple Coulomb scattering (MCS), this is performed in the `Straggle` function.

`ParticleScatter()` is called when a particle has travelled its path length and remains in the material, which means it will interact with a material nucleus or nucleon (*i.e.* a point-like scatter).

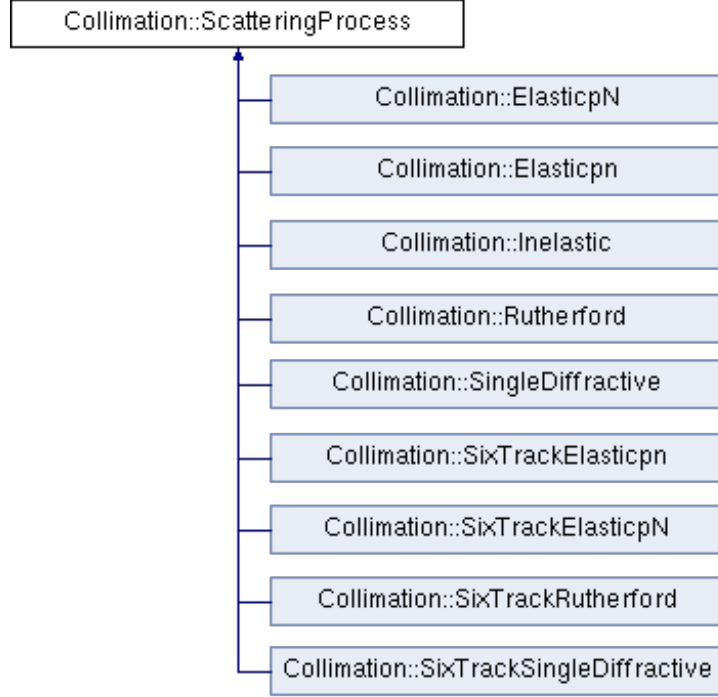


Fig. 11: Scattering processes currently available in MERLIN

2.8 Outputs

The main output from a collimation simulation is the loss map. This is a count of particle losses binned either by element or by position around the ring. MERLIN records these using the `CollimationOutput` class. Typically, losses from collimators and other elements are recorded together. No significant post processing is needed, as all the aperture checking happens on-line. Figure 12 shows a loss map for the LHC ring at the 6.5 TeV squeezed configuration. The losses in the betatron collimation region in IR7 are clearly visible, as are losses in the TCTs around the experiments.

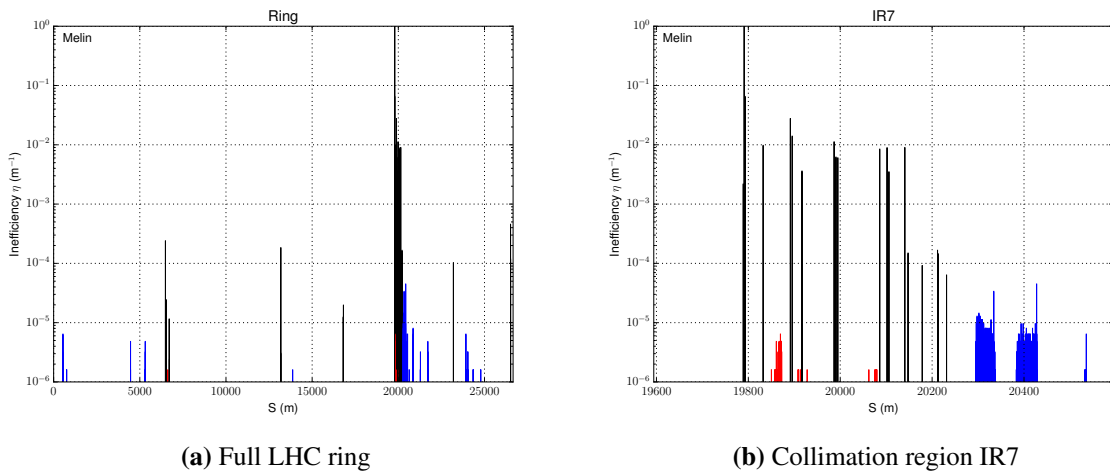


Fig. 12: Loss maps on a log scale for LHC at 6.5 TeV squeezed settings. Black, red and blue show collimators, warm and cold elements respectively.

MERLIN also provides outputs that are useful for diagnostics and better understanding of the collimation process. The `ScatteringModel::JawImpact()` function outputs the coordinates of particles that impact the front face of selected collimators. As well as coordinates, the turn at which the impact occurred is also output, allowing the user to observe any change in the impact parameter over time. This output is useful for observing the effect of the HEL. An example of this output for a single turn in a collimation simulation is shown in Fig. 13, showing the correspondence between the initial distribution (which starts immediately in front of the primary collimator), with the recorded impact coordinates.

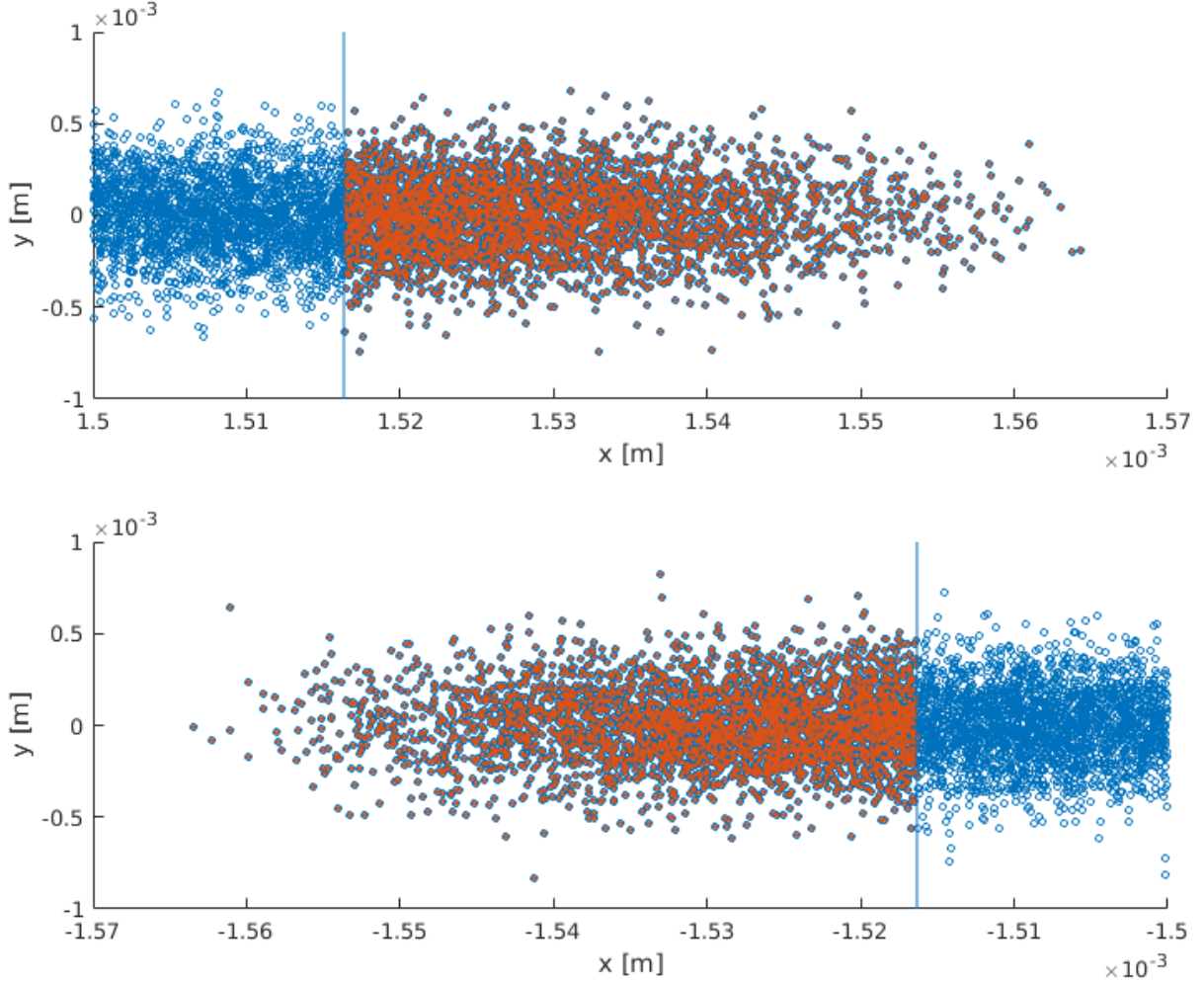


Fig. 13: Initial distribution (blue) and impacts recorded on the primary horizontal collimator (orange) using `JawImpact`, for the positive (above) and negative (below) collimator jaws. The blue line indicates the collimator aperture, where the jaw begins. This simulation is for the 6.5 TeV LHC at flat top, using beam 2.

The `ScatteringModel::ScatterPlot()` function stores the position of particles at each path length step in order to plot scattering tracks along the collimator. This function has been a useful tool for debugging the collimation process, ensuring that aperture checks are performed at appropriate intervals, and showing the effect of the collimation bin size. In Fig. 14 the effect of the collimation bin size is illustrated. Particles undergo scattering, MCS, and ionisation energy loss for as long as they are in the collimator jaw. Even if a particle has exited a collimator jaw, the aperture check cannot take place until the particle has travelled a path length l_{path} , or at the end of the collimation bin. This is evident as particle tracks abruptly stop at 10 cm intervals in the figure. By reducing the bin size, a small path length

is forced, and computation time will increase, however by using a larger bin size protons may undergo significantly more bulk scattering (MCS and ionisation energy loss) when they have in fact already left the collimator jaw. The 10 cm bin size that is used by default allows regular aperture checks without enforcing too small a path length, or compromising the condition of protons that return to the bunch after undergoing scattering in a collimator jaw.

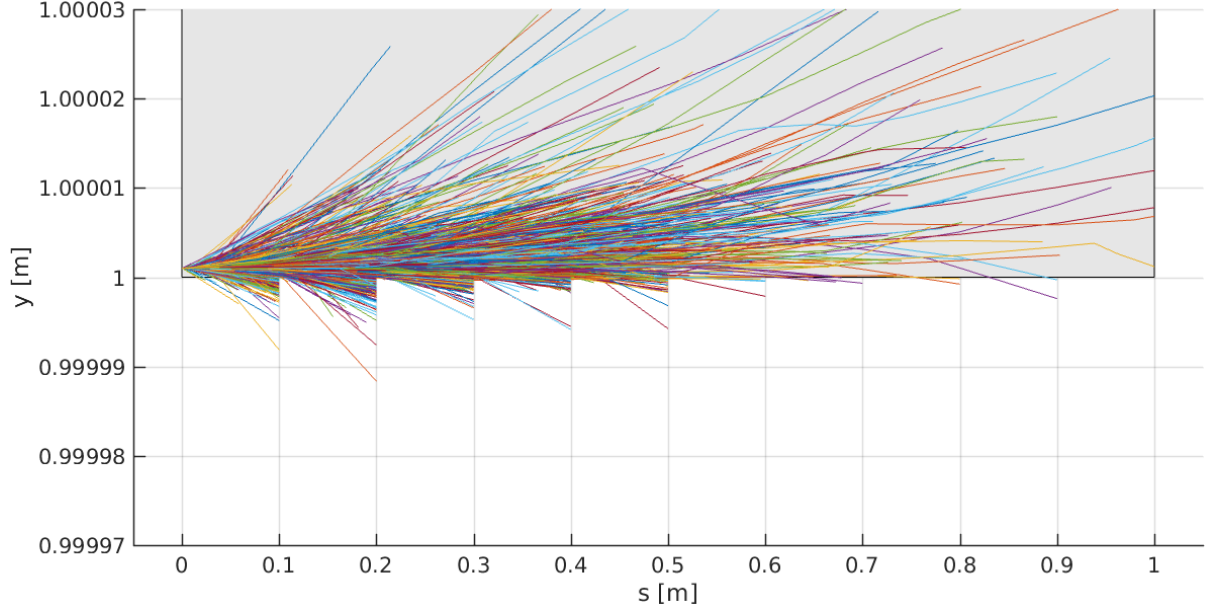


Fig. 14: ScatterPlot() output showing the proton tracks taken in a 1 m long copper collimator with an impact parameter of $1 \mu\text{m}$ in the y plane, using a 10 cm collimation bin size. The grey area indicates the collimator jaw, and the particles are not tracked by this output if they exit the collimator jaw.

ScatteringModel::JawInelastic() stores the coordinates of inelastic interactions. This provides a necessary comparison tool to observe the effect of different collimator materials; an example histogram of the distribution of losses in a given collimator is shown in Fig. 15.

ScatteringModel::SelectScatter() stores the coordinates of selected interactions, outputting the momentum transfer t and the polar angle θ as well as other quantities. As the raw data files produced can be very large, the OutputSelectScatterHistogram() function was created to histogram the data and produce a smaller output. An example of the histogrammed polar angle data is shown in Fig. 16.

3 Development

MERLIN consists of 39k lines of C++ code (LOC), along with 4.8k LOC of examples and 1.8 LOC of tests. CMake is used as the build system, allowing it to be compiled on a range of platforms. The source code is held in Git revision control repository [13].

MERLIN has a growing automated test suite. This is run daily on several physical and virtual machines with a range of operating systems, compilers and CPU architectures. The results uploaded to the CERN CDash server [19]. This allows rapid identification of issues in new development. The test suite performs dynamic analysis using the Vagrind toolset [20] to identify memory leaks and other related errors.

The Git distributed revision system offers a systematic method for making and recording changes to the source code. New features and fixes can be developed on a branch, and then merged into the master branch when tested and ready. This keeps the master branch in a usable condition and makes it easy to

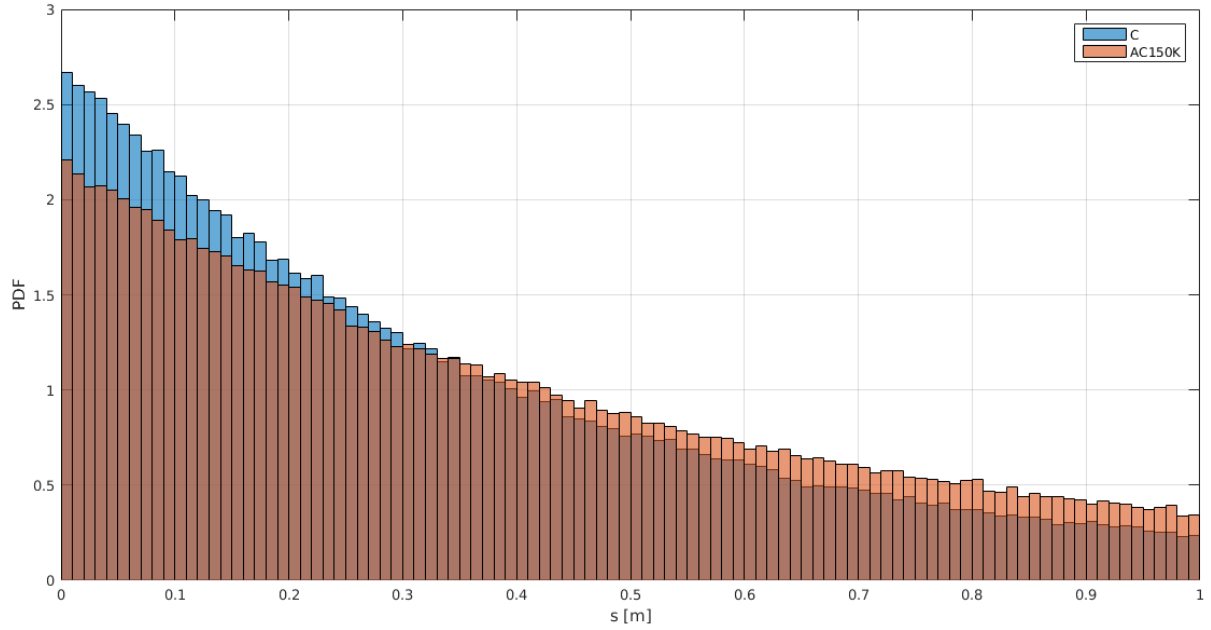


Fig. 15: Inelastic proton interactions (proton losses) in a secondary collimator in the nominal LHC using the JawInelastic output. Comparing losses in pure carbon (blue) with CFC AC150K (orange).

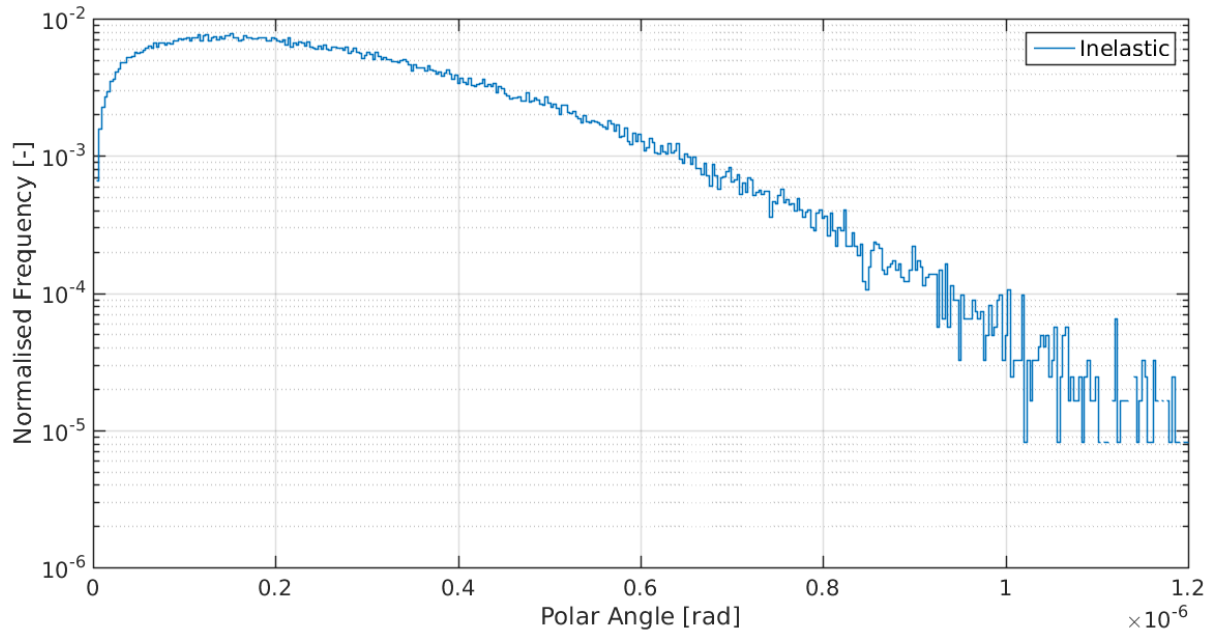


Fig. 16: Polar angle histogrammed in the OutputSelectScatterHistogram() output function, showing the angular distribution of particles that have undergone inelastic interactions in a collimator made of CFC AC150K. In reality this angular spread is given by MCS.

back out a change if it is found to have a negative effect. The master branch is hosted on the GitHub service, which also provides issue tracking and management of branches and merges.

4 Acknowledgements

This work is supported by STFC (UK) grant High Luminosity LHC : UK (HL-LHC-UK), grant number ST/N001621/1.

References

- [1] International Linear Collider, Linear Collider Collaboration, <https://www.linearcollider.org/ILC>, last accessed April 24th 2018.
- [2] F. Poirier et al., An ILC Main Linac Simulation Package Based on MERLIN, Proc. 10th European Particle Accelerator Conference, 2006.
- [3] D. Kruecker et al., MERLIN-Based Start-To-End Simulations of Luminosity Stability for the LHC, Proc. 22nd Particle Accelerator Conference, 2007.
- [4] D. Kruecker et al., Simulation Studies on Coupler Wakefield and RF Kicks for the International Linear Collider with MERLIN, Proc. 11th European Particle Accelerator Conference, 2008.
- [5] CERN BE-ABP, SixTrack: Single Particle Tracking Code, <http://frs.home.cern.ch/frs/>, last accessed April 24th 2018.
- [6] T. Trenkler and J. B. Jeanneret, K2, a software package evaluating collimation systems in circular colliders (manual), CERN SL/94105 (AP), 1994.
- [7] G. Robert-Demolaize et al., A New Version of SixTrack with Collimation and Aperture Interface, Proc. 21st Particle Accelerator Conference, 2005.
- [8] The FLUKA Code: Developments and Challenges for High Energy and Medical Applications, T.T. Bohlen, F. Cerutti, M.P.W. Chin, A. Fassio, A. Ferrari, P.G. Ortega, A. Mairani, P.R. Sala, G. Smirnov, and V. Vlachoudis, Nuclear Data Sheets 120, 211-214, 2014.
- [9] James Molson, Proton scattering and collimation for the LHC and LHC luminosity upgrade, PhD Thesis, University of Manchester, 2014.
- [10] Haroon Rafique, MERLIN for High Luminosity Large Hadron Collider Collimation, PhD Thesis, University of Huddersfield, 2016.
- [11] A. Valloni, H. Rafique, A. Mereghetti, J.G. Molson, R. Appleby, R. Bruce, E. Quaranta, S. Redaelli, MERLIN Cleaning Studies with Advanced Collimator Materials for HL-LHC, Proc. 7th International Particle Accelerator Conference, 2016.
- [12] A. Valloni, H. Rafique, R. B. Appleby, R. Bruce, J. G. Molson, A. Mereghetti, S. Redaelli, S. C. Tygier, MERLIN Simulations of the LHC Collimation System with 6.5 TeV Beams, Proc. 7th International Particle Accelerator Conference, 2016.
- [13] MERLIN on GitHub, MERLIN Collaboration, <https://github.com/MERLIN-Collaboration/MERLIN>, last accessed April 24th 2018.
- [14] A. Valloni, H. Rafique, R. B. Appleby, R. J. Barlow, J. G. Molson, S. Tygier, MERLIN Composite Materials, These proceedings, 2017.
- [15] H. Rafique, R. B. Appleby, R. J. Barlow, J. G. Molson, S. Tygier, A. Valloni, HL-LHC Hollow Electron Lens Integration using MERLIN, These proceedings, 2017.
- [16] MAD - Methodical Accelerator Design, CERN, <http://mad.web.cern.ch>, last accessed April 24th 2018.
- [17] R. B. Appleby, R. J. Barlow, J. G. Molson, M. Serluca, A. Toader, The practical Pomeron for high energy proton collimation, Eur. Phys. J. C (2016) 76:520, <http://doi.org/10.1140/epjc/s10052-016-4363-7>

- [18] Passage of particles through matter, Particle Data Group, <http://durpdg.dur.ac.uk/lbl/index.html>, 2005.
- [19] CERN ABP CDash server, <http://abp-cdash.web.cern.ch/abp-cdash/>, last accessed April 24th 2018.
- [20] Valgrind, <http://www.valgrind.org/>, last accessed April 24th 2018.

BDSIM: Automatic Geant4 Models of Accelerators

L. J. Nevay^{*1}, A. Abramov¹, S. T. Boogert¹, L.C. Deacon², H. Garcia-Morales¹, S. M. Gibson¹, R. Kwee-Hinzmann¹, W. Shields¹, J. Snuverink¹, and S. Walker¹

¹John Adams Institute at Royal Holloway, University of London, Egham, UK

²University College London, London, UK

Abstract

BDSIM is a program that uses a suite of high energy physics software including Geant4, CLHEP & ROOT, to seamlessly track particles through a 3-dimensional accelerator model utilising the full range of particles and physics processes from Geant4. BDSIM was originally developed to simulate linear colliders such as the International Linear Collider (ILC), but has more recently been extended for application to the Large Hadron Collider (LHC), and generally to storage rings. The significant modernisation and revision of the implementation undertaken from 2013 to facilitate these simulations is presented here along with an example of the capabilities.

Keywords

Radiation simulation; tracking; energy deposition; beam loss; collimation.

1 Introduction

To ensure the expected operation of an accelerator, any beam losses must be accurately characterised for a variety of reasons. Firstly, to avoid damage to the accelerator and surrounding equipment as well as minimise radioactivation. Where superconducting magnets are used, cryogenic heat loads due to beam loss must be accurately simulated to ensure the magnets do not quench. Secondly, it's prudent to simulate the beam losses that reach detectors to understand the background in the measurements they are making. Lastly, the cleaning performance of a collimation system can determine the success of the machine operation and detector performance and so must also be simulated.

The first aspect of beam loss simulation is a tracking simulation that records the loss of a particle when it meets the definition of an aperture. Codes that are used to design accelerators such as MAD-X [1] typically include tracking codes, such as the Polymorphic Tracking Code (PTC) [2], for this purpose. Whilst such a simulation may stop at this point, a high energy particle will not and it will undergo various discrete and continuous processes that lead to energy loss and secondary particle production as the initial *primary* particle travels until it exhausts its kinetic energy and comes to rest. Depending on the energy of the primary particle and material it interacts with, the length scale of this process and final rest position may be localised to a single accelerator component or may be far away in an apparently unrelated part of the accelerator after passing through many nonlinear fields.

For example, in a proposed high energy linear collider such as the International Linear Collider (ILC) [3], loss of an electron or positron from the beam can result in the production of a high energy (up to $E \sim 230$ GeV) muon [4]. Such a muon may travel far through the accelerator, surrounding tunnel, soil and shielding, and penetrate the detector. It may also decay during its passage through these various components. In the case of a proton collider such as the LHC, the primary concern is machine protection. The LHC is designed to have a stored beam energy of 350 MJ [5], and given the superconducting magnets

^{*}laurie.nevay@rhul.ac.uk

used to guide the circulating beams must remain at cryogenic temperatures, the loss of only a very small fraction of a single bunch would induce a superconducting quench and result in damage to the machine.

BDSIM [6–9] is an open-source C++ program that allows such scenarios to be simulated by building a 3D model of an accelerator. The model is constructed from its optical lattice description using a library of generic geometries for each type of magnet used in an accelerator. The model is built using Geant4 [10], a Monte-Carlo framework to simulate the passage of particles in matter. BDSIM provides the appropriate fields and numerical integrators for each type of commonly used accelerator magnet allowing particles of a chosen type and distribution to be tracked through the model in a manner similar to a conventional tracking code. Geant4 provides an extensive library of continuous and discrete processes that are used through the simulation in addition to tracking through electro-magnetic fields [10, 11].

Should a high energy particle hit the beam pipe for example, secondary particles may be produced and tracked along with the possibly surviving primary particle throughout the model. BDSIM therefore provides the ability to simulate the energy deposition and the final stopping location of particles as well as the beam loss locations. The expected flux of all particles can be simulated at given locations allowing detector background from the accelerator to be simulated. Moreover, the surviving fraction and distribution of particles from any intercepting device, such as a collimator, can be simulated, which would not normally be possible with a conventional tracking code. The ability to produce and track all secondary particles allows the accompanying energy deposition and dose to be predicted. Depending on the application and information required, suitable cuts in particle properties (kinetic energy, location, species, etc.) can be used to minimise the computational time required as the full simulation with all secondary particles is considerably longer than a tracking simulation of primary particles only.

For accurate accelerator tracking, transfer maps based on the solution to equations of motion are used to provide symplectic thick-lens tracking for linear components. For non-linear elements, numerical integrators are used to calculate the motion due to the Lorentz force. In both cases, the integrators are constructed with a strength parameter based on the nominal beam particle and rigidity that is scaled for each incoming particle. These therefore ignore the field as provided through Geant4. In all cases, Geant4 chooses a step length for the particle to be advanced by considering geometrical intersections, the occurrence of discrete physics processes or to match the required degree of accuracy in tracking.

Each integrator class in BDSIM is capable of handling all particle types (stable and unstable) and rigidities and so provides extra functionality beyond just the tracking algorithm. In the case where the transfer maps would no longer apply, such as non-paraxial particles, the integrator falls back to using a Geant4 4th order Runge-Kutta integrator that uses the field provided for that element by BDSIM. Special provision is made to efficiently deal with spiralling particles as may happen with low energy secondary particles in strong magnetic fields.

The expanded BDSIM geometry library for both aperture and magnet yoke geometry are presented alongside new tunnel geometry provision. Revision of the use of the Geant4 physics processes is discussed, followed by a description of the output format and coordinate systems used. Finally, a hypothetical beam line example is described with sample output.

2 Geometry Library

BDSIM originally only provided circular beam pipes with cylinders of iron surrounding them to represent the yoke of the magnet [7]. As the aperture is the first point where a particle leaves the vacuum, it is critical that this be an accurate part of the simulation. Furthermore, whilst a cylinder of a single material is suitable for a theoretical study of an accelerator at the design stage, more accurate geometry is required for magnets for existing accelerators. An increased variety of geometry however, leads to many potential combinations of apertures and magnet geometry styles. The code to produce such geometry should avoid duplication and should be easily extensible, so the new geometry code implemented in BDSIM is based on the *factory* design pattern [12]. Here, there is a factory class for each style of aperture that produces

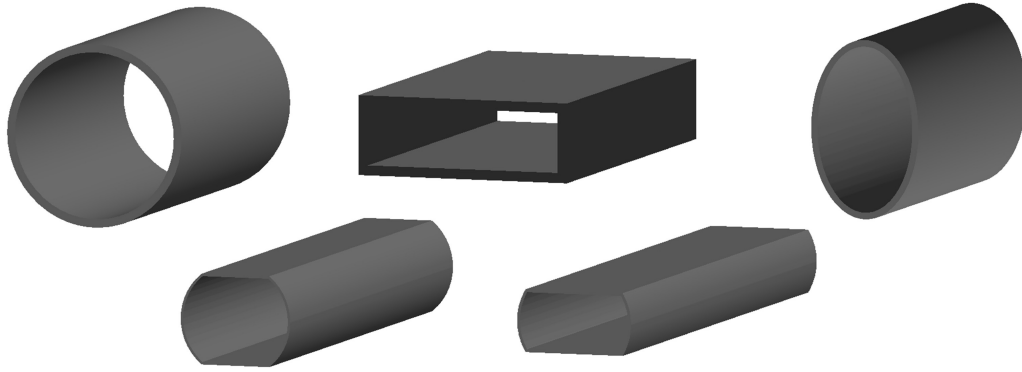


Fig. 1: Selection of aperture models in BDSIM as seen using the visualisation system. From top-left, clockwise: *circular*, *rectangular*, *elliptical*, *rectellipse* and *lhcscreen*.

either a straight or angled section of that aperture. Other classes, such as a magnet construction class, can use the aperture factory to build the required section of beam pipe. Like a real factory, the aperture factory does not retain ownership of its product.

Geant4 provides a variety of classes for primitive shapes that can be used in a constructive solid geometry (CSG) scheme to produce almost any desired shape [10]. The geometry factories written for BDSIM provide parameterised geometry using the Geant4 primitives to make a Geant4 model saving the user from writing a large amount of C++.

With many possible combinations of aperture and magnet styles, we must ensure that all work together without producing geometrical overlaps. Geometrical overlaps cause incorrect navigation of the geometry hierarchy in Geant4 and therefore incorrect tracking and use of physics processes due to the ambiguity of which volume a particle is in. Despite the unknown nature of the geometry at compilation, the program must avoid such scenarios. Geant4 geometry classes have no concept of extent once created, and even if the original parameters used to construct a shape are recovered through inspection, this does not indicate whether two shapes will fit beside each other or overlap. Geant4 provides overlap checking, but only by sampling particle trajectories through the geometry. To overcome this, in BDSIM, a series of extents that represent a simple cube or cylinder are recorded with each Geant4 solid in BDSIM to allow truly dynamically generated parameterised geometry from user input to be safely constructed into more complicated objects and beam lines whilst ensuring there are no geometrical overlaps.

The use of the factory pattern allows extra styles of geometry to be added as required with only minimal modification to the existing code, as well as the guarantee that this new geometry will work correctly with all other existing geometry. To provide this functionality, the previous geometry construction code was rewritten.

Additionally, the beam line construction code was rewritten to be more general and make better use of 3-dimensional transforms available through CLHEP. This allows beam lines to be built not just in a plane, but completely in 3 dimensions with tilts and rotations making BDSIM useful for transfer lines and gantry systems as well as large flat machines.

2.1 Aperture

MAD-X provides 8 different aperture models as well as a user defined boundary [1]. As these cover the most commonly used apertures in accelerators as well as MAD-X being a common source of input model, it was logical to add these to BDSIM. The 8 models (*circle*, *rectangle*, *ellipse*, *rectcircle*, *lhcscreen*, *rectellipse*, *racetrack*, *octagon*) were added to BDSIM as different factories. The aperture may be specified as a default for the whole model as well as on an element-by-element basis. A selection of apertures are shown in Fig. 1 as rendered by the Geant4 visualiser.

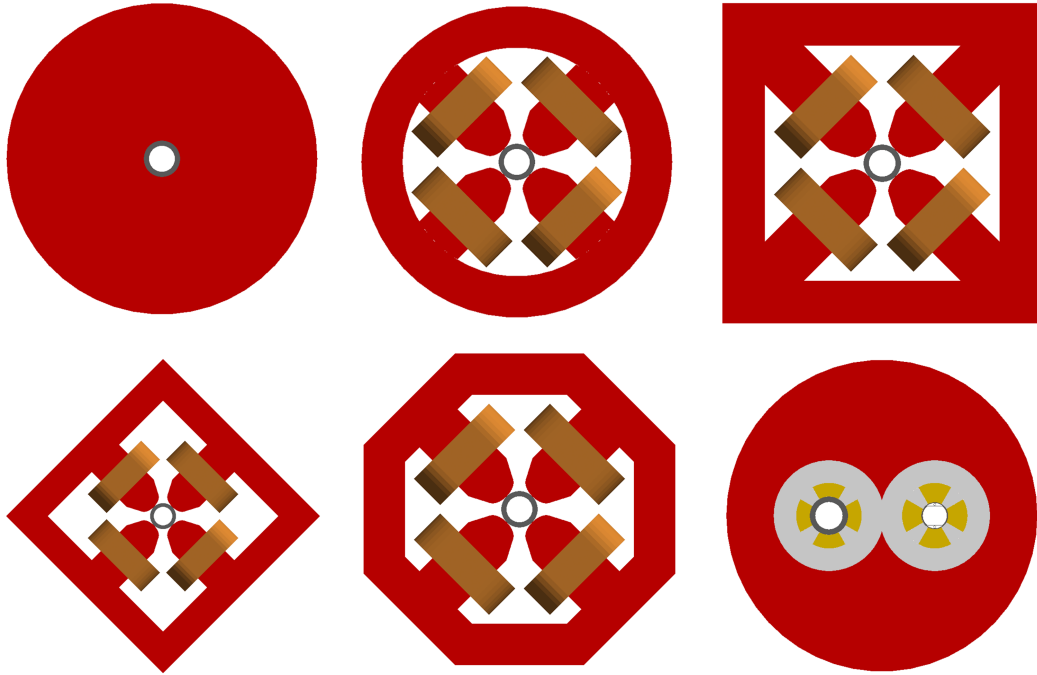


Fig. 2: Different magnet geometry styles for a quadrupole. Clockwise from the top left, these are *cylindrical*, *polescircular*, *polessquare*, *lhcrigh*, *polesfacetcrop* and *polesfacet*.

Aside from the aperture dimensions, the user may specify the thickness of the beam pipe that defines the aperture as well as the material (atomic composition, density and state) of the beam pipe and the vacuum volume inside the beam pipe. Additionally, a more complex model of the *lhcscreen* aperture was added including a copper screen, cooling pipes and outer layer [5, 8].

2.2 Magnet Geometry

In addition to the aperture models, more varied magnet geometry was required. The original style provided with BDSIM is an annular cylinder of solid user-assignable material surrounding the beam pipe, which is iron by default. Whilst this may be an acceptable simplification in some cases, the majority of accelerators use normal-conducting magnets with poles and coils, and so BDSIM was extended to provide different styles of magnet geometry.

To extend the magnet geometry capabilities required that the magnet construction code was rewritten in a factory pattern with one factory class providing one style of geometry for all types of magnets. Eight styles (*cylindrical*, *polescircular*, *polessquare*, *polesfacet*, *polesfacetcrop*, *lhclef*, *lhcrigh* and *none*) are now provided with BDSIM. A selection of styles for a quadrupole magnet are shown in Fig. 2. The majority of the new styles provide magnet poles as well as coils that scale in proportion to the available space between the beam pipe and the user-defined parameter *horizontalWidth* for the full width of the magnet body. The poled geometry factories were written with common code to build the poles and a variety of yoke shapes connecting them. Both a dipole and quadrupole with the design of the LHC arc magnets are provided. Although the design of these is fixed, a small degree of tolerance in increased aperture is provided by removing parts of the coils if required.

Separate pieces of geometry for the end of the coils where they connect to the other side of the pole are provided by the poled geometry factories and these are placed only where the edge of a magnet is not adjacent to another magnet. Unused ones are cleared from memory before commencing the simulation to reduce memory usage.

A null option with no geometry was added for testing purposes, to allow a magnet where only the

beam pipe is constructed. A similar feature for the aperture definition is being added that will provide only a vacuum volume. Used together, these options will provide vacuum only tracking in a similar fashion to a more typical tracking code.

2.3 Tunnel Geometry

BDSIM originally provided no formal support for tunnel geometry. If the intention is to simulate highly penetrating radiation, it is important to include the tunnel as this will effectively filter and guide the radiation seen further along the beam line.

Similarly to the aperture and magnet geometry, a series of factories were written to provide five different styles (*square*, *rectangular*, *circular*, *elliptical* and *rectaboveground*) of tunnel geometry. *rectaboveground* represents a rectangular tunnel based on a thick flat slab of concrete, suitable for a ground level accelerator with surrounding shielding.

Aside from the factories to create segments of tunnel, a builder was written to provide a tunnel that will follow a given lattice. This surveys the already constructed beam line geometry and creates tunnel sections when the cumulative angle, offset or length reach a given tolerance. This allows an approximate tunnel to be simulated for a machine that is still under design with no definite tunnel schematic.

2.4 Curved Geometry Modelling

Nearly all accelerator components can be categorised as either straight or with a finite bending angle in one plane. The Geant4 CSG solids that can be used to provide curved geometry are a torus and a rotation solid. Using either of these for arbitrary aperture and magnet cross sections would lead to large placement offsets due to the typically large bending radii of high energy accelerator magnets. This may lead to lack of precision in the placement or length of the object as well as complicate the placement of each object in the beam line. To avoid this, the geometry in BDSIM is only made in straight sections. To achieve a bent section, the section is split into many small straight sections with angled faces. The number of sections a bend is split into is calculated from a maximum allowable tolerance between the arc and the chord of any segment. This is controllable by the user, but the default is 1 mm.

Constructing models in this way makes the relatively complex geometry construction code manageable. Straight sections with angled faces are constructed by creating a longer than desired straight section and then intersecting it with a cylinder with angled faces (the `G4CutTubs` class). A single section in a bend is constructed only once but placed many times, which significantly reduces memory usage whilst allowing the aperture to be more accurately represented.

2.5 Misalignments and Imperfections

For the most accurate simulation of an accelerator, measured magnet misalignments and field imperfections should be incorporated into a model. BDSIM is able to simulate each of these.

In the case of component misalignments, each component in BDSIM can be specified with an additional transverse offset or tilt (rotation about the curvilinear S axis). However, as a 3-dimensional model is created and surrounded by air, the components must meet exactly. In the case of dipoles (rectangular and sector bends), the transverse offset is ignored as this would change the length of the lattice. However, the tilt is taken into account and should the dipole have pole face rotations, the beam pipes before and afterwards are constructed with the correctly angled faces to allow seamless tracking. The fields are linked to the geometry so any misalignment is automatically handled in tracking.

Field imperfections are not treated directly as each beam line element has a pure field and the associated numerical integrator for tracking. However, BDSIM provides the ability to construct thin multipole elements. As each element is required to have 3 dimensions, these are constructed as a 1 μm long box. A symplectic thin-lens integrator is used for tracking. A common strategy is to place a thin

multipole at both ends of each magnet as well as in the centre of it (assuming it is split) with measured imperfections or higher-order multipole coefficients. BDSIM also provides the ability to import field maps and perform tracking using Geant4's 4th order Runge-Kutta integrators, although whilst suitable on a small scale, these are non-symplectic and therefore not suitable for tracking in large circular machines.

2.6 Externally Provided Geometry

Whilst an extensive library of generic geometry is provided with BDSIM that allows quick progression from optical model to 3D geometry, more accurate geometry for a given existing accelerator may be desired. In this case, BDSIM provides the ability to load externally provided geometry in GDML format [13], and more formats may be added in future. Pieces of external geometry may be placed *as is* in the beam line, or in the case of magnets, wrapped around a beam pipe provided by BDSIM.

3 Physics Processes

Before Geant version 4.10, the developer of a program using Geant4 was required to instantiate classes for particles and physics processes as well as attach them to each other in the required way to provide a *physics list*. Geant4 has since encapsulated this construction in a set of physics lists that can be added together in a modular fashion. Custom physics processes or combinations thereof can be written to fit into this system. This provides a uniform and validated approach to a large library of physics processes as well as known outcomes. The implementation of the physics process construction in BDSIM was rewritten to use the new Geant4 modular physics lists classes. This now allows the physics lists to be used in a modular fashion that was not previously possible without modifying the source code and recompiling.

From Geant version 4.10.1 onwards, a generic interface to process cross-section biasing was introduced. This feature has now been exposed in BDSIM allowing the user to define a particle type, process and biasing factor as well as which volumes to attach it to. This allows efficient simulation of rare processes or situations such as interaction with residual gas in the accelerator vacuum.

4 Coordinate System & Data Handling

4.1 Curvilinear Coordinates

BDSIM, via Geant4, builds a 3-dimensional model in Cartesian coordinates. The model is built following a description of a lattice in curvilinear coordinates that follows the axis of the accelerator. Especially for a circular accelerator, it is more useful to describe beam losses in curvilinear coordinates than Cartesian ones.

Geant4 allows the developer to get the transform from any volume to the top of the geometry tree (the *World* volume) and vica-versa, giving *global* and *local* coordinates systems respectively. Geant4 also provides a mechanism to get the relative transform to a different level in the geometry hierarchy, however, given the potentially variable hierarchy depth of different components and styles available in BDSIM as well as the unknown depth of externally provided geometry, it is not possible to know beforehand the number of levels to navigate. Even then, curvilinear coordinates are desired and not global or local Cartesian coordinates.

To overcome this, BDSIM uses Geant4's capability to build a parallel world - a separate geometry model with a different representation of, in this case, the accelerator. A beam line of simple cylinders is built that mimics the reference trajectory of the beam line (without any orbit bumps) as shown in Fig. 3.

Any global coordinate in the original world (termed '*mass world*' by Geant4) may be read out in the parallel *curvilinear world*. The local to global transform from the simple cylinders may be used, providing the local Cartesian coordinates inside a given cylinder, which is degenerate with the transverse curvilinear coordinates of the accelerator. A registry of each cylinder is kept along with the curvilinear

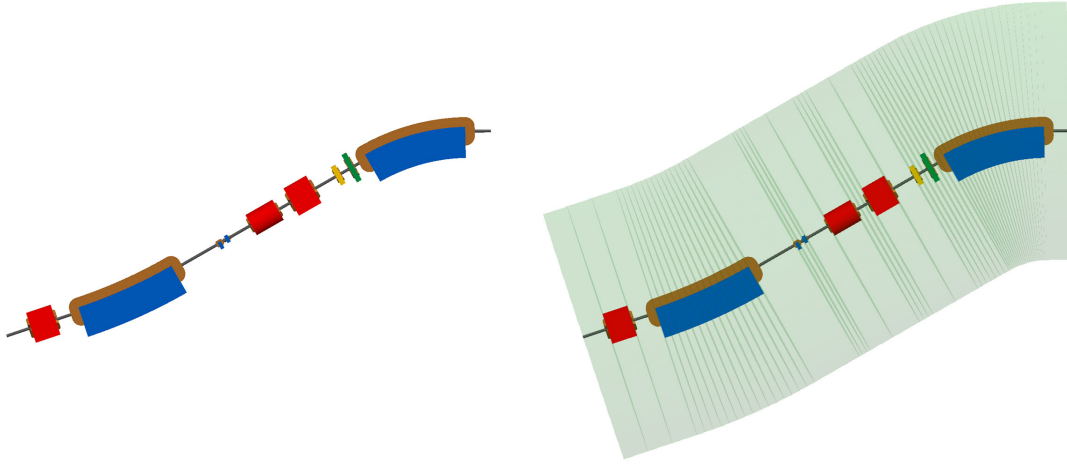


Fig. 3: Plan view of an example beam line with and without the parallel world curvilinear cylinder geometry overlaid in semi-transparent green.

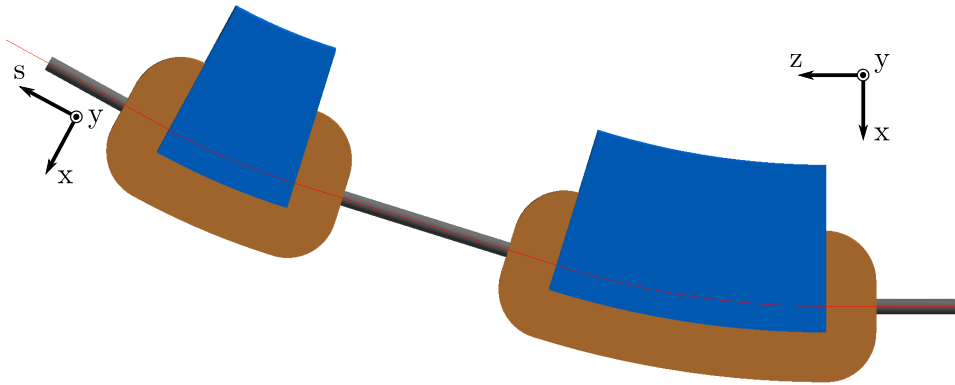


Fig. 4: Plan view of a simple beam line with two dipoles showing the difference between the curvilinear coordinate system x, y, s on the left and the global Cartesian system x, y, z on the right. The reference trajectory is shown as an overlaid red line.

s coordinate for the centre of that cylinder allowing s, x, y to be determined. The difference in these coordinate frames is shown in Fig. 4.

4.2 Output Format

Geant4 provides access to the full history of an event including all secondary particles and trajectories produced, but this is too much information to deal with in a manageable or meaningful way. Instead of this, a summary of each event as well as information requested by the user is recorded in a set of output C++ classes. At the end of each event these classes can be written to different output formats. The primary format for output is a ROOT [14] file. This allows direct serialisation of the output classes in a well-documented, freely-available compressed binary format. The ROOT framework provides a data visualiser and analysis framework that is highly suited to the analysis of this form of data and is widely used in the high energy physics community. The same classes used to write the data are used by an analysis tool, REBDSIM, provided with BDSIM to simplify loading the data. The analysis library may be used in compiled code, the ROOT interpreter and the Python interpreter with equal functionality. The BDSIM output files contain four main structures stored in ROOT *Trees*:

1. The complete set of options used for the simulation.

2. A record of the coordinates and transforms for the model.
3. A summary of all events (*Run* information)
4. Event by event summaries.

The event tree stores information on a per event level, where an event typically starts with tracking an individual particle from the primary beam distribution. For each event, the initial coordinates, energy deposition histograms, duration and primary trajectory are recorded. The user may define elements in the beam line to attach *samplers* to, which are 1 pm thin planes placed at the end of each element that record the coordinates and species of all particles passing through them. The samplers are also recorded per event in the event tree.

5 Circular Machines

BDSIM was originally developed for linear, single-pass machines. It has recently been extended to facilitate tracking in circular machines. As BDSIM builds a 3-dimensional model, no modification was required to track particles in a circular machine. However, further modification was required for control over the simulation in a multi-turn setting.

In a Geant4 application, each event finishes when all particles are tracked down to zero energy or all leave the World volume. In the case of a storage ring and in the absence of synchrotron radiation, a particle inside the dynamic aperture of a circular machine may be stored indefinitely. As each beam particle is simulated individually there are no collective or stochastic effects that would naturally cause the particle to be lost over time. Therefore, the simulation would continue indefinitely. To allow the simulation to terminate and control the maximum number of turns taken in any one event, a new element called the *terminator* was added. This counts the passage of the primary particle on each revolution and dynamically changes to an infinite absorber after a certain number of turns have passed. In the case of a conventional tracking code, the beam line sequence is a series of maps that is explicitly applied to a set of particle coordinates so the user must specify a maximum number of turns.

Due to the 3-dimensional nature of a BDSIM model, a further difference between BDSIM and conventional tracking codes was discovered. Not all optical models of circular machines close perfectly in Cartesian coordinates. In the case of the LHC, the common dipoles used to bring the beams in and out of collision are treated as sector bends in the optical model leading to a very small inaccuracy. In a Cartesian model, the $\sim 70\mu\text{m}$ transverse offset causes the beam to be lost within a few turns. This error in the optical model is extremely small compared to the alignment tolerances and scale of the machine and so is not considered problematic normally. To compensate for this effect, a new beam line element called the *teleporter* was introduced that transports the particle from the exit of the machine to the entrance such that a reference particle with no transverse displacement would return to the same position.

6 Hypothetical Example

To demonstrate the capabilities of BDSIM a hypothetical example beam line was written. The lattice consists of two dipoles, 3 quadrupoles, a sextupole and octupole as well as a pair of horizontal and vertical steering dipoles. A visualisation of the machine in BDSIM is shown in Fig. 5. This provides a selection of magnets and geometry styles, but is not designed for a specific optical purpose. This in turn leads to a desirable beam loss to demonstrate the features and analysis.

1000 particles were randomly sampled from a Gaussian beam as described in Table 1. The default options for BDSIM and the list of physics processes used was *em* and *synchrad* which map to the Geant4 physics process constructors `G4EMStandardPhysics` and the registration of the process `G4SynchrotronRadiation` to all charged particles. The resulting output file from BDSIM (~ 28 MB in size) was analysed using REBDSIM that provides event-by-event analysis of BDSIM output file. It produces a series of histograms as specified in a simple ASCII text file (*analysisConfig.txt*) written by

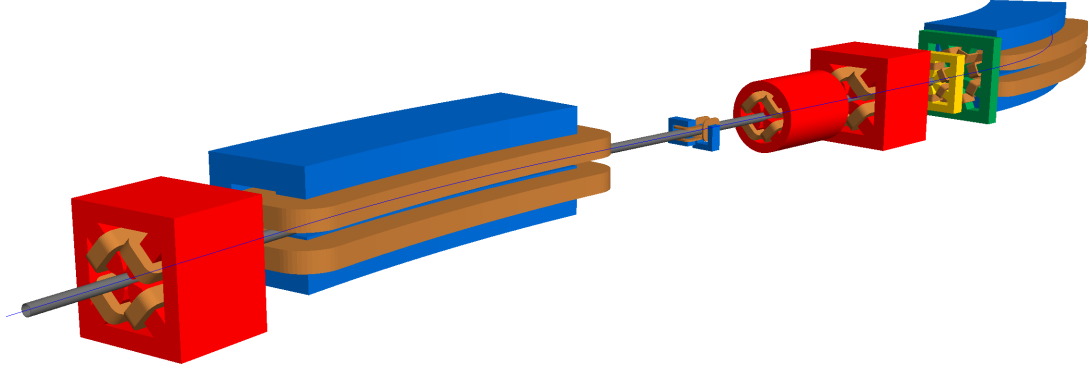


Fig. 5: Hypothetical beam line using a variety of magnets and geometry styles with the reference trajectory shown by an overlaid blue line.

Table 1: Beam parameters used to simulate hypothetical example beam line energy deposition

Parameter	Value	Unit
Particle	e^-	
E	3	GeV
σ_x	1	mm
$\sigma_{x'}$	1	μ rad
σ_y	1	mm
$\sigma_{y'}$	1	μ rad
σ_E/E	0.01	

the user. The analysis configuration file used for this analysis is shown in Fig. 6. REBDSIM produces an output file also in ROOT format with the histograms categorised in directories by whether they were derived from a run or event level analysis.

Figure 7 shows the primary *hits*, *losses* and energy deposition in the accelerator material as a function of curvilinear s position. The primary hit is the first point on the primary trajectory where a physics process determined the step length taken and was therefore not transportation or boundary crossing step. The loss point is the end of the primary trajectory and this represents the last point at which the particle is classified as a primary particle in the simulation. All geometry is sensitive to energy deposition and this is recorded in a histogram per event as a function of s .

Figure 7 has several notable features. For both vertical axes the data is the mean across all 1000 events with the vertical error bars representing the r.m.s. The vertical scales are therefore the expected rate as opposed to the total number observed in this simulation.

Also of note is that the primary hits are all within the first metre of the beam line. Here, these ‘hits’ are due to synchrotron radiation in the first dipole, as the synchrotron radiation is the first physics process invoked and the energy of the primary particle is affected. If a different condition is desired, the user may easily write their own analysis of the primary trajectory. The near constant low-level energy deposition from $s = 1 - 3\text{m}$ is predominantly due to the fan of synchrotron radiation being absorbed by the beam pipe inside the first dipole.

7 Summary & Outlook

The latest developments to BDSIM have been described that significantly revise many features throughout the code. The new geometry factories for aperture, magnets and tunnel sections described provide a much richer set of simulation abilities than previously available. The revised output format and REBD-


```

analysisConfig.txt
Debug 0
InputFilePath ./output_event.root
OutputFileName ./ana_1.root
CalculateOpticalFunctions 1
CalculateOpticalFunctionsFileName ./ana_1.dat
# Object treeName Histogram Name # Bins Binning Variable Selection
Histogram Event. ElossAgain {50} {0:10:2} Eloss.S Eloss.energy
Histogram Event. Primaryx {100} {-0.1:0.1} Primary.x 1
Histogram Event. Primaryy {100} {-0.1:0.1} Primary.y 1
Histogram Options. seedState {200} {0:200} Options.GMAD::OptionsBase.seed 1
Histogram Model. componentLength {100} {0:0:100} Model.length 1
Histogram Run. runDuration {1000} {0:1000} Info.duration 1
Histogram Event. 8XY_Distribution {50,50} {-2:2,-2:2} Sampler_d2_8.x:Sampler_d2_8.y Sampler_d2_8.partID==11
Histogram Event. 8XY_Distribution_electron {50,50} {-2:2,-2:2} Sampler_d2_8.x:Sampler_d2_8.y Sampler_d2_8.partID==11
Histogram Event. 8XY_Distribution_photon {50,50} {-2:2,-2:2} Sampler_d2_8.x:Sampler_d2_8.y Sampler_d2_8.partID==22
Histogram Event. 8Energy {100} {0:3.01} Sampler_d2_8.energy 1
Histogram Event. 4XY_Distribution {50,50} {-0.1:0.1,-0.1:0.1} Sampler_d2_4.y:Sampler_d2_4.x 1
Histogram Event. 4XY_Distribution_positron {50,50} {-0.1:0.1,-0.1:0.1} Sampler_d2_4.y:Sampler_d2_4.x Sampler_d2_4.partID==11
Histogram Event. 4XY_Distribution_electron {50,50} {-0.1:0.1,-0.1:0.1} Sampler_d2_4.y:Sampler_d2_4.x Sampler_d2_4.partID==11
Histogram Event. 4XY_Distribution_photon {50,50} {-0.1:0.1,-0.1:0.1} Sampler_d2_4.y:Sampler_d2_4.x Sampler_d2_4.partID==22
Histogram Event. 4Energy {100} {0:3.01} Sampler_d2_2.energy 1

```

Fig. 6: Example analysis configuration file for REBDSIM used to perform analysis. A series of both 1D and 2D histograms are defined along with their binning and which variable in the data to use. The selection column is used for weighting where a Boolean expression can be used for weight 1 or 0.

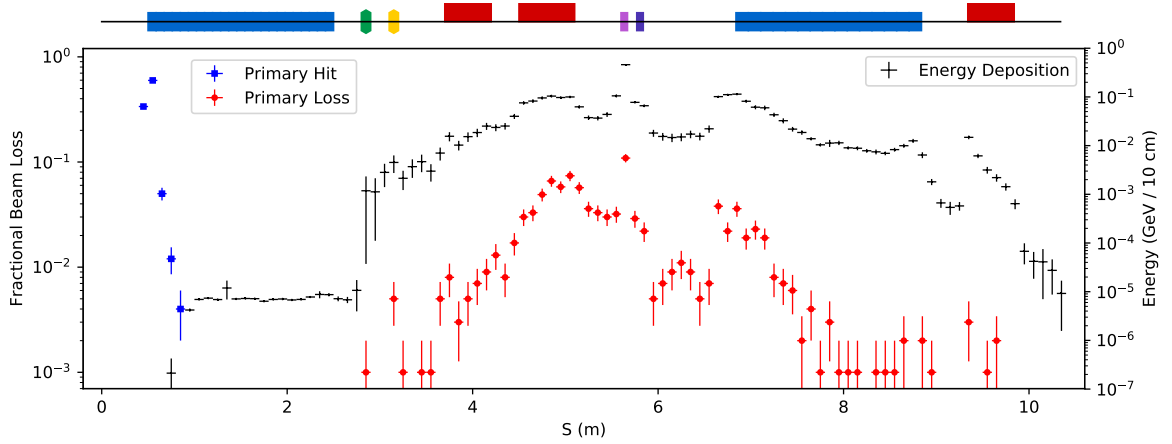


Fig. 7: Beam interaction ('hit') points, loss points and energy deposition throughout the lattice as a function of curvilinear s position. A schematic of the lattice is shown above the graph with the dipoles in blue, quadrupoles in red, sextupoles in yellow, octupole in green and corrector dipoles in light and dark purple.

SIM analysis tool provide the new ability to do an event-by-event analysis with the correct statistical analysis.

These developments have been implemented in a careful and generalised manner so that they are not only applicable to the developers' studies but so that BDSIM is capable of simulating beam losses and backgrounds for nearly any accelerator in the high energy physics community. Whilst it is not possible to predict every desired feature, a strongly object-orientated approach with well defined and documented patterns allows others to extend BDSIM with their own geometry and features suitable for other experiments.

Acknowledgements

The development of BDSIM has received funding from the John Adams Institute at Royal Holloway. This work was supported in part by EU FP7 EuCARD-2, Grant Agreement 312453 (WP13, ANAC2) and STFC, United Kingdom.

References

- [1] The MAD-X Program, CERN, <http://madx.web.cern.ch/madx>
- [2] F. Schmidt, E. Forest and E. McIntosh, Introduction to the Polymorphic Tracking Code: Fibre Bundles, Polymorphic Taylor Types and Exact Tracking, CERN-SL-2002-044-AP (2002).
- [3] C. Adolphsen *et al.* The International Linear Collider Technical Design Report, Vol 3.II: Accelerator Baseline Design, CERN-ATS-2013-037 (2013).
- [4] O. Markin, *arXiv* 1402.2515 (2014).
- [5] O. Brüning *et al.*, LHC Design Report, CERN 2004-003-V-1 (2004), <http://dx.doi.org/10.5170/CERN-2004-003-V-1>
- [6] <http://www.pp.rhul.ac.uk/bdsim/>
- [7] I. Agapov *et al.* Nucl. Instrum. Methods A **606** (2009) 708.
- [8] J. Snuerink *et al.* Beam Delivery Simulation - Recent Developments and Optimization, Proc. International Particle Accelerator Conf., Richmond, Virginia, USA, 2015.
- [9] L. Nevay *et al.* Beam Delivery Simulation: BDSIM Automatic Geant4 Models of Accelerators, Proc. International Particle Accelerator Conf., Busan, Korea, 2016.
- [10] S. Agnostinelli *et al.* (The Geant4 Collaboration), Nucl. Instrum. & Methods A **506** 3 (2003) 250-303.
- [11] J. Allison *et al.* (The Geant4 Collaboration), Nucl. Instrum. & Methods A **835** (2016) 186-225.
- [12] E. Gamma, R. Helm, R. Johnson and J. Vlissides, Design Patterns, (Addison-Wesley Professional, 1994), Chap. 3.
- [13] R. Chytrcek, J. McCormick and W. Pokorski, IEEE Trans. Nucl. Sci. **53** (2006) 2892.
- [14] R. Brun and F. Rademakers, Nucl. Inst. & Meth. in Phys. Res. A **389** (1997) 81-86

MARS15-Based System for Beam Loss and Collimation Studies

*N.V. Mokhov and I.S. Tropin **

Fermilab, Batavia, IL, USA

Abstract

The system includes the newest version of the MARS15 code for particle production and transport in accelerator components integrated with a novel high-precision tracker/stepper for multi-turn tracking merged with MAD-X. The overall approach is described with all the essential features and programming issues highlighted. The application is illustrated for the Fermilab Recycler Ring and Integrable Optics Test Accelerator (IOTA) facilities as well as for the Higgs Factory muon collider and International e^+e^- Linear Collider (ILC) projects.

Keywords

MARS; MAD-X; ROOT; beam line builder; beam loss; collimation

1 Introduction

At hadron colliders, as at any other accelerator, the creation of beam halo is unavoidable. This happens because of beam-gas interactions, intra-beam scattering, beam particle collisions at the interaction points (IP), and particle diffusion due to RF noise, ground motion and resonances excited by the accelerator magnet nonlinearities and power supplies ripple Ref. [1]. As a result of halo interactions with limiting apertures, hadronic and electromagnetic showers are induced in accelerator and detector components causing numerous deleterious effects ranging from minor to severe. An accidental beam loss caused by an unsynchronized abort launched at abort system malfunction can cause catastrophic damage to the collider equipment. Only with a very efficient beam collimation system one can reduce uncontrolled beam losses in the machine to an allowable level Ref. [1, 2]. Modeling of beam losses in the machine, suppression of their rates at the critical locations via appropriately designed multi-stage collimation systems, and evaluation of the beam loss impact on the collimation system components, superconducting and conventional magnets are done by using – in concert – beam tracking, particle-matter interaction and material response simulation tools.

The contemporary approach consists of partitioning the spatial regions of interest into parts where the corresponding methods are applicable best. Inside the beam aperture, trajectories of the beam and beam halo particles are simulated by means of the deterministic methods of the accelerator physics implemented in the codes such as STRUCT Ref. [3] and SIXTRACK Ref. [4, 5]. In the regions filled with a substance, stochastic Monte-Carlo methods of particle interaction and transport are used, with the best fit to accelerator applications in the FLUKA Ref. [6] and MARS15 Ref. [7, 8] codes. Another code – BDSIM Ref. [9] – combines C++ in-vacuum accelerator style particle tracking and GEANT4 physics Ref. [10]. Successful applications of this approach at the SSC, LHC and Tevatron colliders as well as at their injector chains and fixed target experiments have proven its reliability and effectiveness.

When the approach is used, one needs to implement a way for the bi-directional exchange of the phase coordinates of particles crossing the boundaries between the regions. It seems obvious also that the geometry definition has to be shared between the two codes. A previous experience with the STRUCT-MARS bundle showed that the fulfilment of the requirements mentioned above is necessary to keep a consistency of the joint model and avoid doubling the functionality of one code inside another. For

*Corresponding author: tropin@fnal.gov

example, in the early version of the STRUCT-MARS bundle, the simplex (one-directional) data transfer was used: the phase coordinates of beam particles which were determined as lost on the aperture by STRUCT, were stored to a file to be read by MARS as a source term. Originally, there was no way to send back a particle from MARS to STRUCT in case of its backscattering to the aperture. Due to this limitation, a propagation of the beam particles through a thin scatterer (primary collimator) had later been re-implemented from MARS to STRUCT in a simplified form. Inasmuch, a description of the apertures in the two codes was implemented independently by means of different tools and even with respect to different coordinate systems. A special care was taken to keep the aperture definitions in the two codes consistent with each other. A quite important feature of the STRUCT-MARS coupled system in beam loss simulations was a modelling with a rather low threshold momentum $0.7p_0 \leq p \leq p_0$ for particles escaping primary collimators and other components and tracked through the system. Here p_0 is the beam momentum. Corresponding physics processes in this momentum region were described with a fast simplified MARS module included in the STRUCT code. This feature allowed getting representative beam loss distributions with a contribution from many other lattice components in addition to those from the primary collimators. The full shower simulations with MARS down to thermal neutrons were done then using such a source term to get detailed energy deposition and radiation maps.

The current MARS15-based system for beam loss and collimation modeling is described in this paper. It inherits the earlier approach of the MAD-MARS beam line builder Ref. [11–13] for creation of the beam line (accelerator) geometry now described by means of the MAD-X code Ref. [14] and usable directly in MARS15. Coupling to the MAD-X code gives us a possibility for precise and fast tracking inside the beamline vacuum with the MAD-X PTC module. The data between the MARS15 code and the PTC module are passed as parameters of corresponding functions. Such a way of the data transfer between the program modules is much more effective compared to other possible implementations. It seems obvious that copying data between actual and formal parameters of a function inside address space of a process is definitely much faster than passing data over network port as used, for example, in Ref. [15].

The PTC tracker and MARS15 use the same geometry definition to check the aperture boundaries. The ROOT TGEO package Ref. [16] used in MARS15 allows association of extensions defined by a user with the objects defined in the geometry model. This feature can be used by a user to attach to the node a stepper – the object which is used to make a step in the volume. The functionalities of this feature are described in Sec. 2. The tools require the ROOT version 6.12.04 and MAD-X v5.03.07 or newer ones. The tools used to prepare the beam line geometry shared between the MAD-X PTC tracking module and MARS15 are described in Sec. 3. Several use cases are given in Sec. 4.

2 ROOT-Based Geometry Library for MARS - libstgeo

The implementation of the MARS15 geometry is done using the features provided by the LIBGEOM library which is a constituent of the ROOT system Ref. [16] developed at CERN. It is often referred to as ROOT TGEO because most of the class names provided by the library have prefix TGEO. The ROOT geometry can be imported to the MARS15 application or created in a function called in the application. One can also combine an imported geometry module, e.g., the one of an entire collider detector in the GDML format, with the full 3D description of the collider geometry (Fig. 1) created by means of the tools described in Sec. 3. The TGEO package provides many remarkable features. One of them is a hook allowing to associate an electromagnetic field as well as any object of the user-defined class with the geometry objects. These features are actively used by the MARS15 developers. Recently it was decided to group the frequently used extensions into the library made available for all the MARS15 users.

In general, a geometry module of the considered Monte-Carlo codes performs the following two tasks: (1) finds an identifier of a region where a particle step starts, called a "where-am-i" task; (2) calculates the particle coordinates at the end of the step done in the given direction or at the point where the step crosses the nearest geometry surface – "make-a-step" task. For shortness, the class of objects

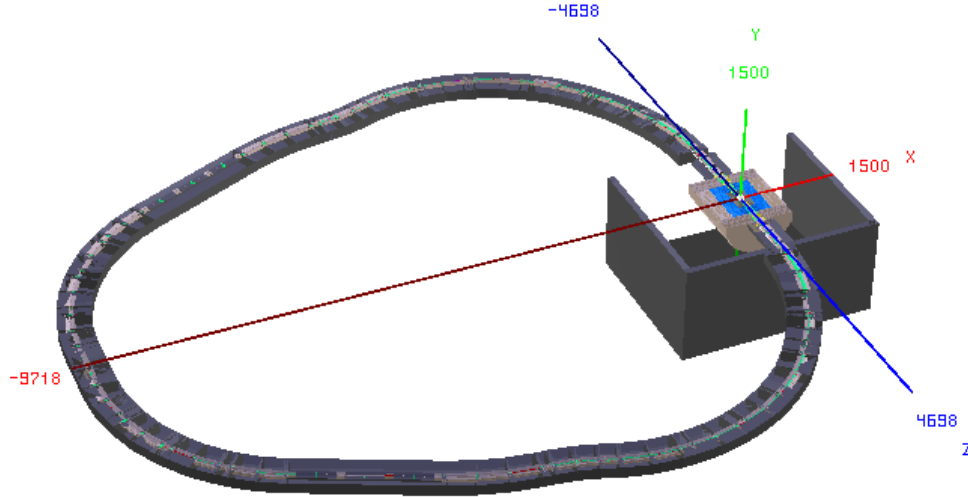


Fig. 1: MARS15 3D model of the Higgs Factory muon collider with the SiD-like detector where the silicon vertex detector and tracker are based on the upgraded version of the CMS detector

implementing the solution of the latter task referred to as "stepper".

In the standard implementation, a solution of the aforementioned tasks were built into the MARS native geometry module. For example, the stepper for charged particles in the magnetic field was implemented by means of a helix algorithm. A user had a control over the algorithm by means of parameters defined in an input file. It was challenging to replace the helix algorithm by something more sophisticated in multi-turn accelerator applications. After integration of the ROOT TGEO package to MARS15, it is now easy to do. By design, a stepper object can be associated either with the volume or node of the TGEO geometry. At the runtime, when the MARS15 tracking module requests a solution of the "make-a-step" task, the library function checks whether the user-defined stepper object is associated with the current node or the volume. If "Yes", then it is tested if the stepper is able to make the provided step. If successful, the stepper makes the step. If there were no user-defined steppers associated with the volume, one of the two steppers provided by the library is used – the straight stepper or helix stepper. The latter is used when magnetic field is associated with the volume. Straight steppers are used in the regions without magnetic field. To keep a backward compatibility, both of the default steppers are controlled via parameters in MARS.INP file. This way the developer can create his/her own stepper used to transport particles in the dedicated regions. No changes in the existing MARS15 code are required.

For implementation of a stepper, the library provides the following abstract base class:

```
#ifndef TGeo_GENERIC_STEPPER_H
#define TGeo_GENERIC_STEPPER_H
#include <TGeoManager.h>
#include <TGeoExtension.h>
class GenericStepper: public TGeoExtension {
private:
    mutable unsigned int RefCounter;

protected:
    GenericStepper():RefCounter(0){};
public:
    virtual TGeoExtension* Grab()
    { RefCounter++; return this;}
    virtual void Release() const {
```

```

    assert(RefCount > 0);
    RefCounter--;
    if (RefCount==0) delete this;
};
// vin[0:2] - r; vin[3:5] - dir; vin[6] - momentum
virtual Int_t propagate(const Double_t charge,
                       const Double_t mass,
                       Double_t* s,
                       const double vin[7],
                       double vout[7], Double_t* toff) = 0;
virtual bool IsApplicable(const Int_t PartID,
                          const Double_t vin[7],
                          const Double_t toff)=0;
};
#endif

```

As one can see, the class `GenericStepper` is derived from `TGeoExtension` – the class defined in the ROOT TGEO package. A file named `GenericStepper.h` containing the declaration shown above can be found in the MARS15 distribution in the directory `$MARS15/linux/include/streg`. One can note also that `GenericStepper` is the abstract class, i.e., to implement a user-defined stepper, one needs to create a derived class containing two functions `IsApplicable` and `propagate`. The `IsApplicable` function returns true if the algorithm implemented in the `propagate` function, is applicable to a particle identified by `PartID` and having position, direction and momentum specified in a vector `vin`, and the time of flight `toff`. Components of the vector `vin` are `vin(0 : 2) = x, y, z` – particle coordinates (cm) in the global frame of the application, and `vin(3 : 5) = { $\Omega_x, \Omega_y, \Omega_z$ }` – direction cosines of the unit vector, defining direction of the particle motion, `vin(6) = p` – the particle momentum. The `const` specifier used in the declaration, indicates that all parameters are used only for input and are not changed by the function.

The function `propagate` makes a step for the particle with the given charge and mass. Parameters `s` and `toff` are used for input and output. At the input, `s` refers to the step size requested by the tracking module, a pilot step. If this step is shortened by a boundary crossing, it is expected that the function will replace the pilot step by the one till the boundary crossing. Similarly, the input `toff` is the reference to the time of flight at the step origin, with its value to be replaced by the time at the step end. The function returns zero upon a successful completion, otherwise it returns -1. There is a special return code -9050898 issued when there are no errors inside the function, but the current trajectory needs to be stopped. It is used when the stepper decides to pass the particle to another tracking module.

The following steppers are currently available in the library:

1. `MagFieldRK4Stepper` – a stepper in magnetic field, 4-order Runge Kutta solver;
2. `BFieldRKStepper` – a stepper in magnetic field, 8-order Runge Kutta solver.
3. `CavityStepper` – a stepper in time-dependent electromagnetic field, 8-order Runge-Kutta solver.

All these steppers have been carefully tested. One of the tests was a comparison of the individual orbits generated with the MAD-X PTC and MARS15 `BFieldRKStepper` stepper in the Fermilab Recycler Ring. Figure 2 shows a perfect agreement between the two results.

Let's note that the applicability rules for the steppers described require the presence of the field in the region which the stepper is attached to. In terms of the ROOT TGEO package, it means that the field object has to be associated with the volume by means of `SetField` method defined in the `TGeoVolume` class. The latter should be derived from the abstract class `TVirtualMagField` defined in ROOT. The library provides an extendable set of such classes for electromagnetic fields defined in local coordinate systems and allowing tabular or analytical representation. In most cases, the implemented steppers are used to solve the equation of motion in bounded regions. In general, this technique can be

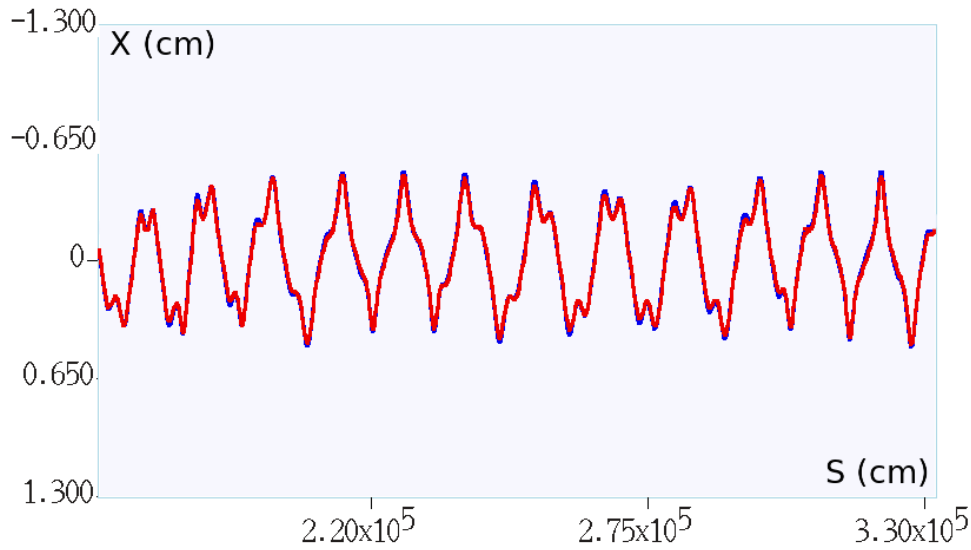


Fig. 2: Beam orbits in the Fermilab Recycler calculated with the MAD-X PTC module (blue) and the MARS15 BFieldRKStepper (red)

used to simulate arbitrary quasi-continuous physical interactions which can be experienced by particle on the path between the **discrete** strong, weak and electromagnetic interactions modelled by MARS15. Examples include modelling coherent interactions in a bent crystal and impact of a gravitational force.

3 MAD-X/ROOT MARS beam line builder (MRMBLB)

The tools described here are a successor of the code Ref. [13] with the following main distinctions:

- A geometry model is created by means of the ROOT TGEO library, while previously a MARS "non-standard" geometry option was used.
- A MAD-X input file is now used as input, the previous format [13] is also supported.
- The data provided in the MAD-X BEAM statement and tables can be accessed from the MARS15 code via the MAD-X API calls.
- The MAD-X survey table is used for the element positioning, while in the previous version a special function was used that duplicates the functionality of the MAD-X survey module.

As before, the user is expected to provide a so-called beam line element generator (henceforth called "factory") to allow descriptions of a single type/class, used in the beam line, to be searched then by a MAD-X class name. It is expected that the elements of the same type (having the same design) are declared as belonging to the same class in the MAD-X input file. This assumes that such a file is a database providing complete information on all the beam line elements.

Prior to creation of the complex builders for the magnets, collimators, etc., one would propose to figure out what beam pipe is most frequently used for the drift spaces in the beam line and implement a factory for that element first, then register the created factory as a builder of elements of the class having name "default". After that, if the beam line builder does not find the specific factory for the class, then an order will be placed to the factory to assemble default elements. After implementing such a default factory, one can visualize the ring or the beam line in the MARS15 GUI or the ROOT 3D viewer. The visualization is a powerful tool at the time of creation of the more complex elements. Here is an example of creating a factory for the beam pipes which can have magnetic field inside (if the field is defined for the elements in the MAD-X file). A declaration file for the factory reads:


```

#ifndef MagRoundVacuumPipe_H
#define MagRoundVacuumPipe_H
#include "NodeGenerator.h" // Declarations provided by the beamline builder
class TVirtualMagField;
class MagRoundVacuumPipe: public NodeGenerator
{
    Double_t wall_thickness;
    Double_t R;
    TGeoMedium* STST;
    TGeoMedium* VAC;
    Int_t CopyNo;
public:
    // Constructor
    MagRoundVacuumPipe(TString name,
                       Double_t ExternalRadius,
                       Double_t WallThickness,
                       TGeoMedium* WallMaterial);

    virtual TGeoVolume* Assemble(const std::vector<Section*>::iterator& ElInfo,
                                TGeoVolume* SecVol, TGeoMatrix* FildMtx);
};
#endif

```

The C++ header shown above declares a class `MagRoundVacuumPipe` as derived from a `NodeGenerator`. The latter is the abstract class for the factories provided by the MRMBLB library. When an object of this factory is created, the four parameters need to be provided to the constructor as indicated in the declaration. The first parameter is a name of the class for elements which factory will generate. If the name coincides with the class of element in the MAD-X file, then the factory will be used to generate a geometry of that element. The next three formal parameters following the name are used to specify parameters of the pipes to be generated in the factory: `ExternalRadius`, `WallThickness` and `WallMaterial` – the names are selfexplanatory. Having this declaration saved as `MagRoundVacuumPipe.h`, one can create the function which generates the beam line geometry in the application:

```

#include "blreg1.h"
#include "marbl.h"
#include "MagRoundVacuumPipe.h"
extern "C" {
    void tgeo_init()
    {
        TGeoMedium* AIR = gGeoManager->GetMedium("AIR");
        TGeoMedium* STST = gGeoManager->GetMedium("STST");
        Double_t Rmax=GetMarsREXT();
        Double_t Zmax=GetMarsZMAX();
        TGeoVolume* World = gGeoManager->MakeTube("World",AIR,0.,Rmax,Zmax);
        // Container for the beamline
        NodeGenerator::DefineWorld(World);
        // Radius for default cylindrical container for the beamline element
        NodeGenerator::SetRSec(Rmax);
        Double_t ExtR,WallThickness;
        // Create factory to be used inside madx function
        new MagRoundVacuumPipe("default",
                               ExtR=7.0,WallThickness=0.1,STST);
        // fname is name of the MAD-X input file
        std::string fname = "input.madx";
        madx(fname.c_str());
    }
}

```

```

    gGeoManager->SetTopVolume(World);
    gGeoManager->CloseGeometry();
}
}

```

In the given example, the ROOT geometry for the sequence of the elements defined in the MAD-X file `input.madx` – which is given as an argument to the MAD-X library function – is created. The factories of elements, declared before a call to MAD-X, are involved in the building process. In this example, the only generator for `MagRoundVacuumPipe` is used. At the time of the `tgeo_init` call, the MARS15 service functions have already converted a description of materials defined in the `MATER.INP` input file to the ROOT representation. Therefore, one can use the ROOT tools to retrieve media from the in-memory database either by name, as shown in the example above, or by index used in the `MATER.INP` file. The next action is creation of the ROOT geometry model. The top volume, `World`, in the presented example has a cylindrical shape and is filled with AIR. Dimensions of the volume – the radius, R_{max} , and the half length, Z_{max} – are retrieved from the MARS15 input data by means of two service functions which are components of the MARS15-ROOT interface.

The MARS15 application with the code provided above can be compiled, however, the linker will complain about the absence of the implementation of functions for the `MagRoundVacuumPipe` class. An example of its implementation is provided in Appendix A. When implementing the class, the user can assign particular steppers to the ROOT geometry objects. The stepper provided in the MARS15 library can be used here, or a stepper specific for the application can be created by the developer of the application.

For the tasks involving the long-term beam transport, a formal procedure and tools are proposed which define design of the applications with joint transport of the particles in MARS15 and in an arbitrary beam tracker. For prototyping the procedure, the MARS15 applications developed for simulations of energy deposition effects in the newly developed collimation system for the Fermilab Booster and the Recycler were used. In these applications a coupled transport with the MAD-X PTC tracker was implemented (Sec. 3.2).

3.1 MAD-X and MARS runtime interactions

The approach provided by the functions of the library consists of a concurrent execution of MAD-X and MARS15 codes where each of the participants keeps their own execution context. The developer can run the MAD-X session inside the MARS15 application, execute MAD-X commands from the MARS15 code, request a data kept in the MAD-X data structures by means of the MAD-X Application Program Interface (API).

To start a MAD-X session, the C function named `madx` should be called at the initialization stage. The function takes the string with name of the MAD-X input file as an argument and begins the session from execution of this file by means of the native MAD-X interpreter. It is supposed that the input file contains the following set of the MAD-X commands:

1. Beam description.
2. Definition of the sequence of elements which the beam is supposed to be propagated through.
3. Creation of the survey table for the sequence.
4. Creation of the twiss table.

The experience shows that the requirements 1 and 2 are always fulfilled in the MAD-X input file, however, often care should be taken to create tables mentioned in items 3 and 4.

By design, duration of the MAD-X session coincides with the runtime of the MARS15 executable. In particular, it means that the `STOP` statement, which resides at the very end of the MAD-X input file, needs to be removed or commented. Otherwise, the MAD-X session will be finished immediately after

execution of that statement and the tables built by the MAD-X script will be destroyed. An absence of the survey table makes it impossible the normal completion of function – the survey table taken from the in-memory data is used then for placement elements constructed by the MRMBLB. Upon successful completion of the MADX function, any MAD-X statement can be executed in an application code by means of the `madx_stmt` function. As an example, initialization of the PTC module reads as the following

```
madx_stmt("ptc_create_universe;");
madx_stmt("ptc_create_layout,model=3,method=6,nst=5,exact=true;");
PTCTrackingActive();
```

One can see that the `madx_stmt` function takes a single argument – a string filled by the MAD-X statement – which is passed for execution to MAD-X interpreter. The sequence commands like those shown above need to be executed in the source module at the initialization stage to make it possible to launch track in the PTC tracker by means of the MAD-X `ptc_start` statement. The last function – `PTCTrackingActive` – sets a global flag signaling to the MAD-X session manager that the external tracker is in use. This flag is also used for the flow control of the special stepper types which needs to be attached to the aperture volumes where one is supposed to use MAD-X. Setting the flag means also that a user shouldn't care about the execution of the closing sequence:

```
madx_stmt( "ptc_track_end; ");
madx_stmt( "ptc_end; ");
```

These functions and, therefore, the corresponding statements, will be automatically executed at the end of the MARS15 application followed by the MAD-X STOP command.

3.2 Interaction with external tracker

In the MRMBLB, the terms of the Open System Interconnection (OSI) reference model are used to define the tasks to be solved for establishing the interconnection between the processes, whether they are running on the same host or in a network environment. The functions are to be implemented for the following layers of the model:

Presentation layer – defines the rules (protocol) of the data conversion requested by the participants of the interaction. The functions of this layer implement the mapping of the phase space used in the MARS15 model to the one used in the PTC code and vice versa. A strict requirement is that the protocols and functions of the layers below are applied only if the mapping exists.

Session layer – defines the rules for service requests and service responses. In the current implementation, the session is always initiated by the MARS15 code. It requests the beam tracker to accept the particle with the given phase coordinates. The session duration coincides with the time required by the beam tracker to simulate the track. During the session, the beam tracker sends requests to MARS15 to check the aperture boundary crossing on the way between two phase points. The MARS15 responds with the code indicating whether the particle is lost or not on the way. In case of the boundary crossing, the phase coordinates of the lost particle are saved in the MARS15 stack for the further simulation. The session is finished when the particle leaves the geometry setup, crosses the aperture boundary or passes the predefined number of turns in a ring inside the beam aperture. In the current implementation, the MARS15 tracking code is suspended for all the session time. At the end of the session, the code kills the track and tries to find a particle in the stack to continue the simulation history.

Transport layer – defines the rules for data transfer. The data is transferred between the MAD-X PTC tracker and MARS15 as parameters of the functions. It means that the implementation of the transport level protocol is delegated to developers of the compiler used for compilation of the

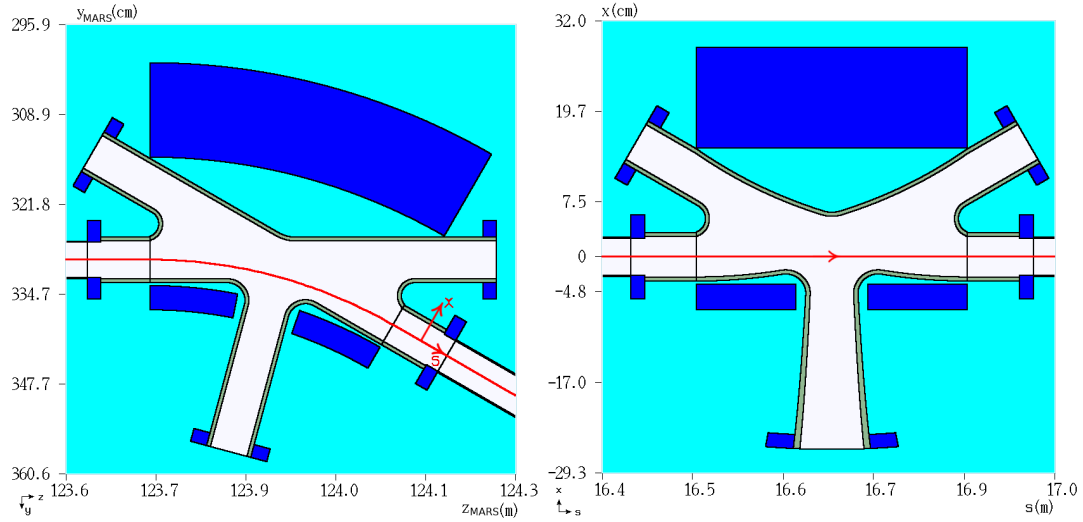


Fig. 3: Orbit-plane view of the B30 dipole region in the Fermilab Integrable Optics Test Accelerator (IOTA) ring as implemented in the MRMBLB model in the global frame (left) and in the MAD-X $\{x, y, s\}$ reference system (right). Axes of the MAD-X reference system are shown in red, arrow points a direction of the s -axis

codes. When the data needs to be passed over the network, a developer is forced to use a third party tool, like a socket library. Usually, this is to be avoided.

All these functionalities should be implemented for a set of dedicated volumes, i.e., those related to the beam line element apertures. The tools described in Sections 2 and Appendix A provide a straightforward way to realize this. In case of an arbitrary beam tracker, communications with the beam tracker developers might be needed. For example, the presentation layer functions can be specific for the chosen beam tracking program. It should also be taken into account that the applicability of the beam tracking code can be limited by the approximations used to create the model. As a result, not all the phase coordinates provided by the MARS15 session can be mapped to those of the beam tracker. The member-function `IsApplicable` of a stepper is the good place to address all the limitations. The mapping can be implemented in two stages. At the first stage, MARS15 global coordinates and direction of motion are transformed to the MAD-X $\{x, y, s\}$ reference system, as shown in Fig. 3. Such a kind of the direct and reverse transformations is provided by the MRMBLB library. Then, the particle coordinates and direction vector, expressed in this intermediate frame together with the particle momentum and time of flight, are converted to the phase coordinates of the particular beam tracking code.

The beam tracking session is to be initiated in the `propagate` function of the stepper class. It should be a synchronous call of a function implemented in C, C++ or FORTRAN, which propagates the track in the beam tracker starting from the phase coordinates given as input parameters. In the prototype version with the MAD-X PTC tracker, the session is initiated by a consequent execution of the two MAD-X commands, `ptc_start` and `ptc_track` using the `madx_stmt` function described in Sec. 3.1. The PTC module is used in the "element-by-element" mode.

According to the session scenario, the next step would be to get a feedback from the beam tracker. To do that, a call to the function is to be added which takes the particle coordinates at two consequent points along the s -axis as input parameters. For the "thick" PTC tracker, it could be the points at the entrance and exit of the element. For the "thin" tracker, like the SIXTRACK or STRUCT, it could be the phase coordinates after the current and previous thin lenses. The purpose of the function is to detect the boundary crossing along the path between these points. In the implementation with the MAD-X PTC tracker, corresponding changes were made to the PTC module, `madx_ptc_track_run.f90` file of the

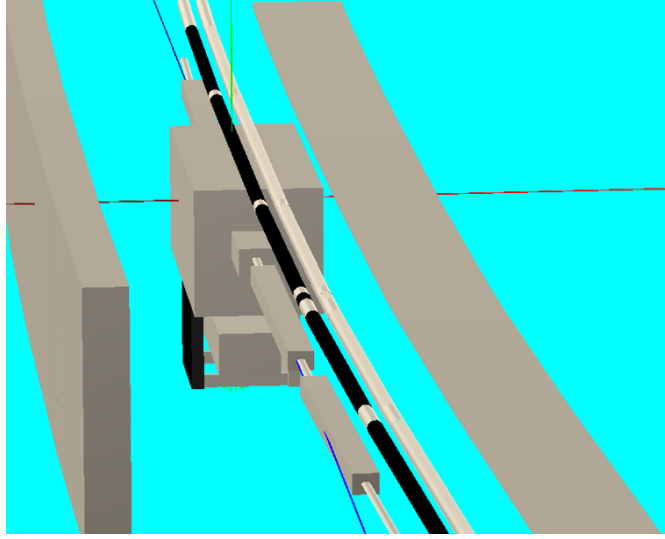


Fig. 4: 3D representation of the SCOLL613 secondary collimator

MAD-X distribution. To detect a boundary crossing, one of the trackers available in the `stgeo` library (See Sec. 2) is used. If the aperture boundary check is implemented in the beam tracker, then it can be used instead. Note that the aperture definition in the tracker should be used with care. As one can see in Fig. 3, the magnet aperture, which looks quite simple in the Lab system, can become non-uniform along the element, being expressed in the curvilinear design orbit reference frame. It is up to the developer of this function to decide whether to perform the boundary crossing for all elements, selected elements or elements positioned in the selected ranges. As prescribed by the session protocol, when a boundary crossing was detected, then the particle should be sent to the MARS15 stack. To do that, the functions named `stk_push` of the `libm15stack` library should be used. The return value should be agreed with developers of the beam tracking code. It needs to be defined to indicate whether the beam tracking code should kill the track or continue it. The implementation was done for the MAD-X PTC module. The user can implement an interface with the tracker specific for his/her application using the procedure described.

The scheme doesn't assume that the beam tracker provides any feature for parallel execution of the code. The MPI parallelism is implemented at the level of simulation of the MARS events. This was successfully tested on the ALCF THETA cluster for jobs of tens of thousand ranks.

4 Use cases

4.1 The Fermilab Recycler

The Fermilab Recycler is an 8-GeV storage ring used in the accelerator complex to provide an intense 120 GeV proton beam to the neutrino experiments. The Recycler resides in the same tunnel as the Fermilab Main Injector (MI) – accelerator which ramps proton beam energy from 8 GeV to 120 GeV. To obey the needs of the experiments, there is a plan to increase – in the nearby future – the 120-GeV beam power from 400-700 kW up to about 1 MW.

To keep beam losses and activation in the critical locations at the current or lower levels, the new multi-stage collimation system was designed and installed. The problem was that the collimator positions were limited longitudinally by the spaces available in the lattice and laterally by the ceiling and the tunnel wall. The place “613-616” was found in the tunnel at the junction with the NuMI tunnel. A detailed MRMBLB 3D model was built for the collimation region, some fragments are shown in Fig. 4 and Fig. 5.

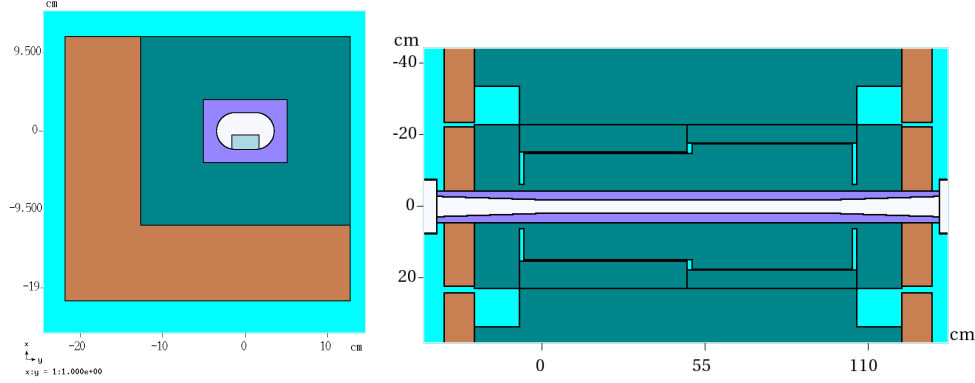


Fig. 5: Transverse cross sections of the primary collimator assembly (left) and the secondary 20-ton collimator (right)

Results of the multi-turn tracking and full MARS15 simulations were used for optimization of the collimation efficiency, the primary, secondary and tertiary (mask) collimator designs, with the heat load, cooling, activation and radiation shielding issues addressed on top of these. The fraction of the halo particles absorbed by secondary collimators after the passage through primary collimator are 24% (SCOLL613) and 63% (SCOLL616). These numbers are mostly due to a single pass of the beam halo protons through a primary collimator.

When passing through the primary collimator, beam protons undergo multiple scattering resulting in random energy loss and change their the direction of motion. Due to random energy-angular kicks caused by the primary collimator, proton path-lengths inside aperture are also random. Fig. 6 shows that most often protons are absorbed after four turns and it takes less than 20 turns for most protons to be lost in the lattice.

Calculated prompt dose isocontours in the collimation region are shown in Fig. 7. It was assumed here that 2 kW is delivered by particles of the beam halo onto the upstream boundary of the primary collimator made of TZM and having thickness $125\mu m$. The collimator is positioned at the boundary of 3σ beam envelope. Jaws of two secondary collimators are aligned along the 3.5σ boundary.

4.2 ILC

It has been recognized that the loss of dark current (DC) generated in the superconducting RF (SRF) cavities is the dominant source of radiation loads on the International e^+e^- Linear Collider (ILC) Main Linac components and related radiation environment in the tunnel. The DC-induced loads are several orders of magnitude higher than those induced by beam-gas Coulomb scattering and beam-gas bremsstrahlung. The DC production and handling algorithm in a detailed 3D MARS15 model, results of the beam loss, radiation load and dose calculations along with possible mitigation are described in Ref. [17]. Figure 8 (left) shows a fragment of the MARS15 3D model of a 9-cell SRF cavity.

A dark current (DC) formula reads

$$J = a \phi^{-1} F^2 \left[-\frac{\nu(f) b \phi^{3/2}}{F} \right] \quad (1)$$

where J (A/m^2) is DC density, $a \approx 1.541434 \times 10^{-6} (A eV V^{-2})$, $F = E \beta$, $E (\frac{V}{nm})$ – time dependent electric field in the immediate vicinity of the cavity surface, $\beta = 100$ is the field enhancement factor, $\phi = 4.2(eV)$ is the local work function, $f = 1.439964 (eV^2 V^{-1} nm) F/\phi^2$, $\nu(f) \approx 1 - f + (1/6) f \ln f$, $b \approx 6.830890 eV^{-3/2} V nm^{-1}$.

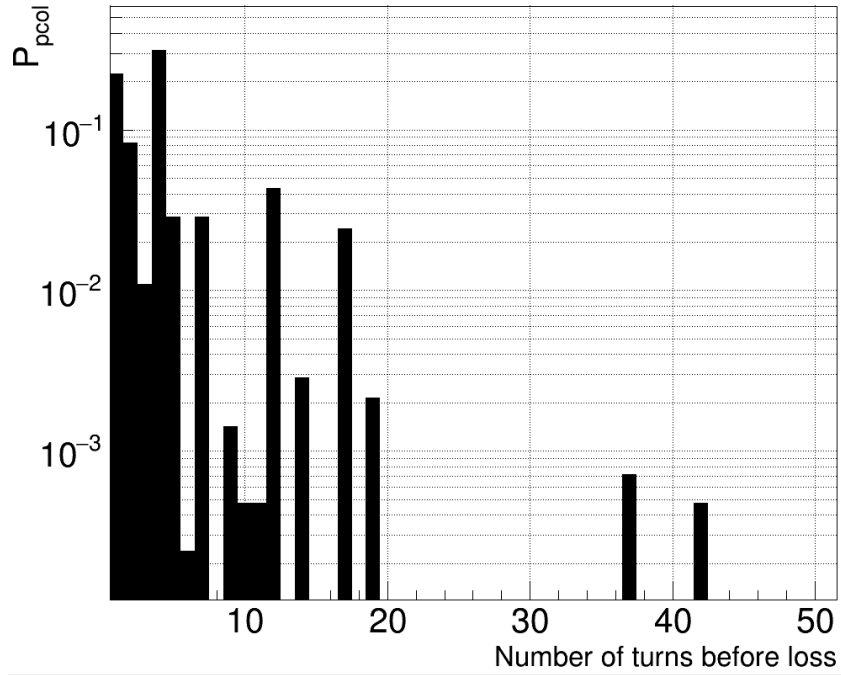


Fig. 6: Probability to be lost for the beam halo protons passed through the primary collimator as a function of the number of turns in the Recycler

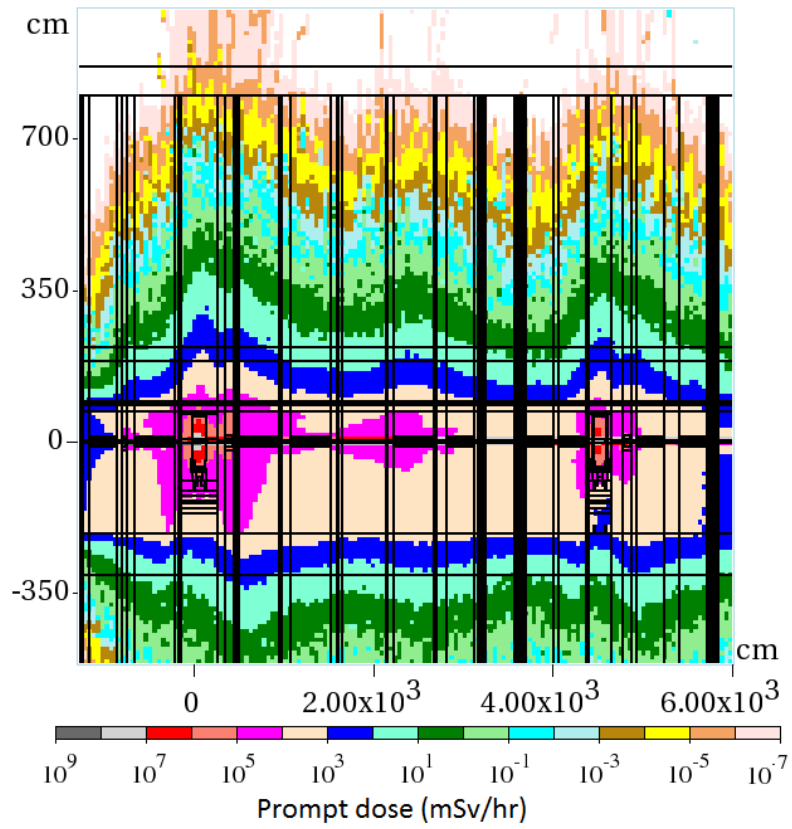


Fig. 7: Prompt dose distribution in the Recycler collimation region

Several DC electron tracks generated in the model are shown in Fig. 8 (right).

Analysis has shown that, in the case when field emission is described by Eq. (1) and emission points are distributed uniformly over the cavity surface, the forward DC from the cavity № 30 toward quadrupole 2 does not exceed 2.1 nA (see Fig. 9), with the average current over the 30 cavities being approximately 1 nA .

However, if a cavity with significant surface imperfections is placed at the very beginning of such a beam line segment, DC in the last cavity of the segment can exceed the assumed level of 50 nA due to the significant amplification shown in Fig. 9.

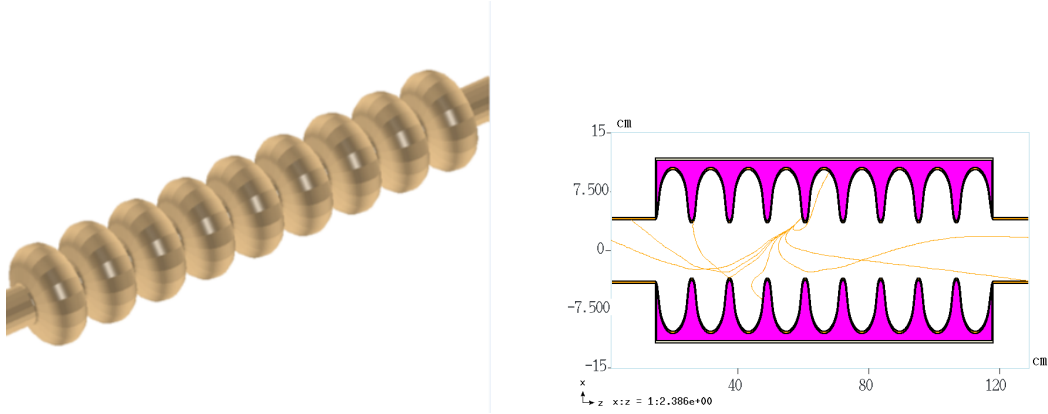


Fig. 8: MARS15 3D model of 9-cell SRF cavity (left) and sample DC electron trajectories in an SRF cavity (right)

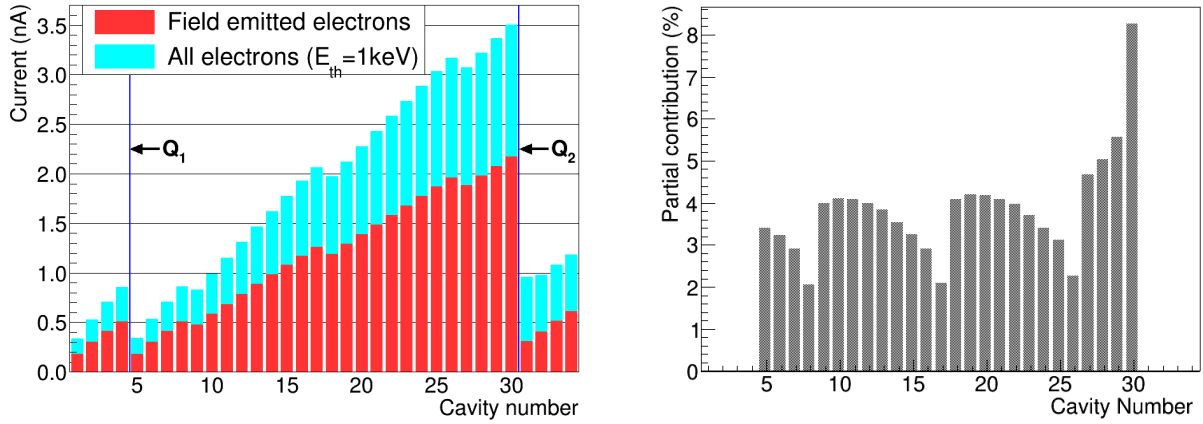


Fig. 9: Left: Current (nA) of accelerated DC electrons in the aperture at the downstream end of each cavity. The results include both field emitted electrons which did not experience any interaction with cavity walls and electrons that already experienced either elastic or inelastic scattering in cavity walls. The labels Q1 and Q2 indicate locations of quadrupole magnet 1 and 2, respectively. Right: Partial contribution of electrons emitted from the upstream cavities to the forward current at downstream end of cavity № 30

Appendices

A Node generator implementation

The implementation for the `MagRoundVacuumPipe` class according to declaration presented in Sec. 3 is given below:

```
#include <sstream>
#include <TGeoManager.h>
#include <TGeoTube.h>
#include "MagRoundVacuumPipe.h"
#include "TMADX_BField.h"
#include "MADXStepper.h"

MagRoundVacuumPipe::MagRoundVacuumPipe(TString name,
                                         Double_t ExternalRadius,
                                         Double_t thick,
                                         TGeoMedium* WallMedium):
    NodeGenerator(name),
    wall_thickness(thick),
    R(ExternalRadius),
    STST(WallMedium),
    CopyNo(0)
{
    VAC = gGeoManager->GetMedium("BVAC");
}

// Creation of element
TGeoVolume* MagRoundVacuumPipe::Assemble(
    const std::vector<Section*>::iterator& ElInfo, // Information about current ele
    TGeoVolume* Sec, // input, container volume for the element
    TGeoMatrix* tr) // input, matrix of element, useful in field definitions
{
    CopyNo++; // Copy number
    TGeoShape* SecShape=Sec->GetShape();
    Double_t nl[3]={0.,0.,-1.};
    Double_t nh[3]={0.,0.,1.};
    Double_t SecHalfL; // Half length of the element
    bool IsCtubSection=false;
    // Determine shape of a container section
    if(SecShape->IsA() == TGeoTube::Class()) {
        // Cylinder with parallel faces
        SecHalfL=((TGeoTube*) SecShape)->GetDz();
    } else if(SecShape->IsA() == TGeoCtub::Class()) {
        // Faces are not parallel to each other
        IsCtubSection=true;
        SecHalfL=((TGeoCtub*) SecShape)->GetDz();
        // Normal vecor for upstream and downstream faces
        const Double_t* n1=((TGeoCtub*) SecShape)->GetNhigh();
        const Double_t* n2=((TGeoCtub*) SecShape)->GetNlow();
        memcpy(nl,n2,3*sizeof(Double_t));
        memcpy(nh,n1,3*sizeof(Double_t));
    }

    TMADX_BField* FieldInAperture=0;
    // Check precence of field
    if(((ElInfo)->K).size()) {
        // Element has a magnetic field,
        // create mag.field object to be attached to the aperture:
```

```

    FieldInAperture=new TMADX_BField(ElInfo);
}
// Outer steel
std::ostringstream ostr;
ostr << (*ElInfo)->nm << CopyNo;
TGeoVolume* STube;
if(IsCtubSection) {
    STube = gGeoManager->MakeCtub(ostr.str().c_str(),STST,
                                0.L, R,SecHalfL,
                                0.L,360.L,
                                nl[0],nl[1],nl[2],
                                nh[0],nh[1],nh[2]);
} else {
    STube = gGeoManager->MakeTube(ostr.str().c_str(), STST,
                                0.0L, R, SecHalfL);
}
// Inner vacuum
ostr.str("");
ostr << (*ElInfo)->nm << "RAP" << CopyNo;
TGeoVolume* BeamAperture;
if(IsCtubSection) {
    BeamAperture = gGeoManager->MakeCtub(ostr.str().c_str(),VAC,
                                0.L, R-wall_thickness,SecHalfL,
                                0.L,360.L,
                                nl[0],nl[1],nl[2],
                                nh[0],nh[1],nh[2]);
} else {
    BeamAperture = gGeoManager->MakeTube(ostr.str().c_str(), VAC,
                                0., R-wall_thickness,
                                SecHalfL);
}
// Associate field with volume
BeamAperture->SetField(FieldInAperture);
// Create stepper:
MADXStepper* Stepper = new MADXStepper(ElInfo);
// Attach stepper to the volume:
BeamAperture->SetUserExtension(Stepper);
STube->AddNode(BeamAperture,1);
return STube;
}

```

Acknowledgements

The authors are grateful to I. Rakhno for collaboration and useful comments.

This document was prepared using the resources of the Fermi National Accelerator Laboratory (Fermilab), a U.S. Department of Energy, Office of Science, HEP User Facility. Fermilab is managed by Fermi Research Alliance, LLC (FRA) acting under contract No. DEAC02-07CH11359.

This research used a computing award allocation from the ASCR Leadership Computing Challenge at the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

References

- [1] N.V. Mokhov, “Beam Collimation at Hadron Colliders”, ICFA Workshop on Beam Halo Dynamics, Diagnostics and Collimation (HALO-03), Montauk, NY, May 2003, Ed. J. Wei, W. Fischer and

- P. Manning; AIP Conf. Proc., Vol. 693, pp. 14-19.
- [2] N.V. Mokhov *et al.*, “Tevatron Beam Halo Collimation System: Design, Operational Experience and New Methods”, JINST, **6** (2011) T08005.
 - [3] I.S. Baishev, A.I. Drozhdin, and N.V. Mokhov, “STRUCT Program User’s Reference Manual”, Technical Report SSCL-MAN-0034, SSC Laboratory, 1994.
 - [4] <http://sixtrack.web.cern.ch/SixTrack/>
 - [5] G. Robert-Demolaize *et al.*, “A New Version of SixTrack with Collimation and Aperture Interface”, Proc. of PAC2005, Knoxville, TN, USA (2005) pp. 4084-4086.
 - [6] A. Ferrari, P.R. Sala, A. Fassio and J. Ranft, “FLUKA: A Multi-Particle Transport Code”, CERN-2005-010 (2005). <http://www.fluka.org/fluka.php/>
 - [7] N.V. Mokhov, “The MARS Code System User’s Guide”, Fermilab-FN-628 (1995), <https://mars.fnal.gov>
 - [8] N.V. Mokhov, P. Aarnio, Yu. Eidelman, K. Gudima, A. Konobeev, V. Pronskikh, I. Rakhno, S. Striganov, and I. Tropin, “MARS15 Code Developments Driven by the Intensity Frontier Needs”, Prog. Nucl. Sci. Technol., **4**, pp. 496-501 (2014).
 - [9] L. Nevay *et al.*, Proc. IPAC 2014, Dresden, p. 182 (2014)
 - [10] J. Allison *et al.*, Nucl. Instr. Methods, **A835** (2016) pp. 186-225, <https://geant4.cern.ch>
 - [11] D.N. Mokhov, O.E. Krivosheev, E. McCrory, L. Michelotti and J.F. Ostiguy “MAD Parsing and Conversion Code”, Fermilab-TM-2115 (2000)
 - [12] O.E. Krivosheev, E. McCrory, L. Michelotti, D.N. Mokhov, N.V. Mokhov and J.F. Ostiguy, “A Lex-based MAD Parser and its Applications”, in *Proc. 2001 Particle Accelerator Conference*, pp. 3036-3038, Chicago, IL, June 18-22, 2001; Fermilab-Conf-01/142-T (2001).
 - [13] M.A. Kostin, O.E. Krivosheev, N.V. Mokhov and I.S. Tropin, "An Improved MAD-MARS Beam Line Builder: User’s Guide," FERMILAB-FN-0738.
 - [14] MAD - Methodical Accelerator Design, CERN - BE/ABP Accelerator Beam Physics Group. <http://madx.web.cern.ch/madx/>
 - [15] A. Mereghetti, F. Cerutti, R. De Maria *et al.* “SixTrack-Fluka Active Coupling for the Upgrade of the SPS Scrapers”, Proc. 4th International Particle Accelerator Conference (IPAC 2013): Shanghai, China, May 12-17, 2013, <http://JACoW.org/IPAC2013/papers/wepea064.pdf>
 - [16] R. Brun and F. Rademakers, "ROOT - An Object Oriented Data Analysis Framework", Proceedings AIHENP’96 Workshop, Lausanne, Sep. 1996, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) 81-86. <http://root.cern.ch/>
 - [17] N.V. Mokhov, I.L. Rakhno, I.S. Tropin, “Modeling Radiation Loads in the ILC Main Linac and a Novel Approach to Treat Dark Current”, Fermilab-FN-1035-APC (2017).

Simulation Tools for Heavy-Ion Tracking and Collimation

P. D. Hermes, R. Bruce, F. Cerutti, A. Ferrari, J.M. Jowett, A. Lechner, A. Mereghetti, D. Mirarchi, P.G. Ortega, S. Redaelli, B. Salvachua, E. Skordis, G. Valentino, V. Vlachoudis
CERN, Geneva, Switzerland

Abstract

The LHC collimation system, which protects the LHC hardware from undesired beam loss, is less efficient with heavy-ion beams than with proton beams due to fragmentation into other nuclides inside the LHC collimators. Reliable simulation tools are required to estimate critical losses of particles scattered out of the collimation system which may quench the superconducting LHC magnets. Tracking simulations need to take into account the mass and charge of the tracked ions. Heavy-ions can be tracked as protons with ion-equivalent rigidity using proton tracking tools like SixTrack, as used in the simulation tool STIER. Alternatively, new tracking maps can be derived from a generalized accelerator Hamiltonian and implemented in SixTrack. This approach is used in the new tool heavy-ion SixTrack. When coupled to particle-matter interaction tools, they can be used to simulate the collimation efficiency with heavy-ion beams. Both simulation tools are presented and compared to measurements.

Keywords

Heavy-ions; ions; collimation; ion tracking; hiSixTrack; STIER

1 Introduction

The CERN Large Hadron Collider (LHC) [1] is a proton and heavy-ion collider reaching unprecedented beam intensities and particle momenta. In operation with $^{208}\text{Pb}^{82+}$ ions, the LHC has reached an integrated stored beam energy of 9.7 MJ, compared to only 3.6 MJ anticipated in the LHC design phase [1,2]. The destructive power of the stored $^{208}\text{Pb}^{82+}$ beams in 2016 is comparable to an explosion of more than 2 kg of TNT. Compared to that, the envisaged LHC upgrade to High Luminosity LHC (HL-LHC) [3] foresees another increase of the stored beam energy by a factor of two.

If large fractions of the stored beam energy are deposited in the LHC hardware, severe damage can be caused. But even tiny fractions of the stored beam energy are sufficient to induce magnet quenches (the transition from the superconducting to normal conducting state), which would reduce the time available for operation in stable beams. Compared to proton beams undesired losses and hence quench risk from heavy-ion collimation debris is significantly larger [4]. The origin for this observation is the fragmentation of heavy-ions into lighter ion fragments at their passage through the collimator.

The LHC collimation system [1, 5] has the task of intercepting undesired beam loss to protect the LHC hardware and avoid magnet quenches. Given the importance of collimation, it is indispensable to have reliable and accurate simulation software to anticipate the performance of the LHC collimation system with both proton and heavy-ion beams. Compared to tools for proton collimation simulations, heavy-ion simulation tools available before 2014 had a limited predictive power, due to different simplifications used in their physics models.

In this chapter, we introduce the peculiarities related to the simulation of heavy-ion collimation. Based on these considerations two new simulation tools STIER and hiSixTrack available since 2014 and 2016 are presented. Typical applications are discussed and an example analysis from the operation of the LHC with heavy-ion beams in 2015 is briefly summarized.

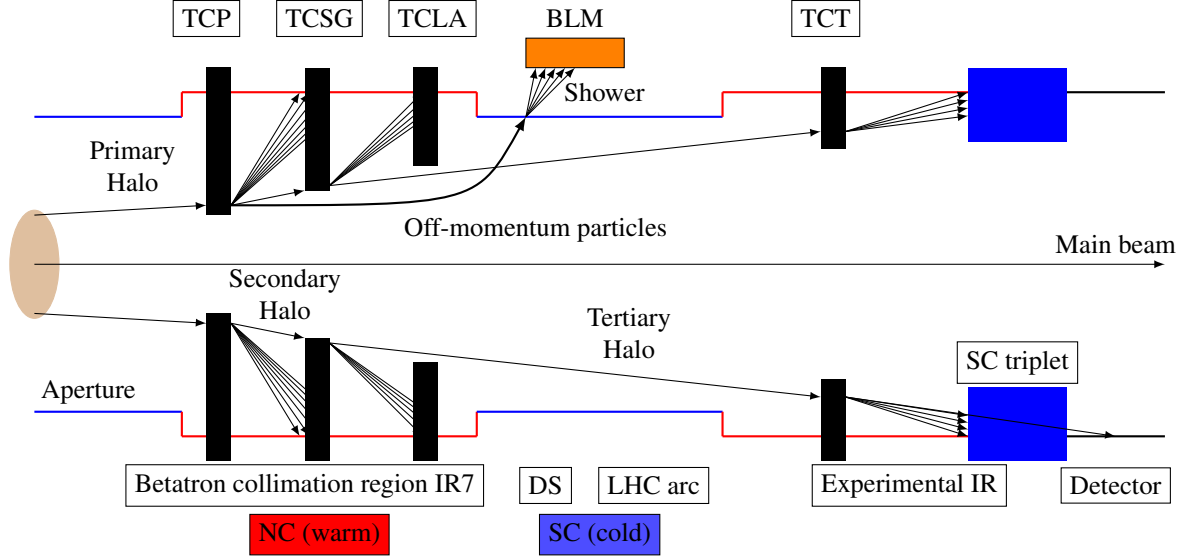


Fig. 1: Schematic overview of the LHC multi stage collimation system. Normal conducting (NC) regions are drawn in red, superconducting (SC) regions in blue. Based on [6].

2 Collimation of Heavy-Ion Beams

2.1 The LHC Collimation System

2.1.1 Overview

To protect the LHC hardware from potential damage by undesired beam loss, the LHC is equipped with a multi-stage collimation system. One of the aims of the collimation system is to protect the superconducting LHC magnets from beam-induced magnet quenches, e.g. the transition from their superconducting to the normal conducting state. The latter would trigger an immediate beam dump and require several hours of cryogenic recovery, undesirably reducing the LHC duty time for physics. The collimation system, illustrated for the betatron collimation region IR7 in Fig. 1 is designed for proton beams. It relies upon a defined sequence of collimation instances. Primary collimators (TCP) intercept beam particles at large transverse amplitudes. In the TCP, the beam particles shall be scattered into the secondary collimators (TCSG). Shower absorbers (TCLA) downstream of them provide protection from secondary hadronic and electromagnetic showers. Tertiary collimators (TCT) locally protect the superconducting triplets in the experimental insertions from particles scattered out of the secondary collimators.

In spite of the sophisticated collimation system with more than 100 collimators and the availability of many degrees of freedom to optimize the collimation performance, particles can still leave the collimation system and hit the superconducting LHC magnets. The superconducting magnets exposed to the highest amount of collimation losses, and hence under the highest quench risk, are those in the dispersion suppressor region (DS) downstream of IR7. The superconducting dipole magnets in this region act as a spectrometer on particles scattered out of the collimation system, which are not captured by the TCSG collimators. If particles have lost momentum compared to the main beam, or their mass and charge changed, their bending radius in the magnet will be different with respect to that of the reference particle. This effect leads to a transverse offset, ultimately resulting in a particle loss if the transverse offset exceeds the available beam pipe aperture in the magnet. To keep track of particle losses throughout the ring, the LHC is equipped with beam loss monitors (BLM) [7, 8] measuring secondary showers from particles interacting with the LHC hardware. The LHC beams can be transversally excited to induce losses at the collimation system and measure the cleaning inefficiency in a given machine configuration (qualification loss map) [9].

2.1.2 Measurements of the Heavy-Ion Collimation Efficiency

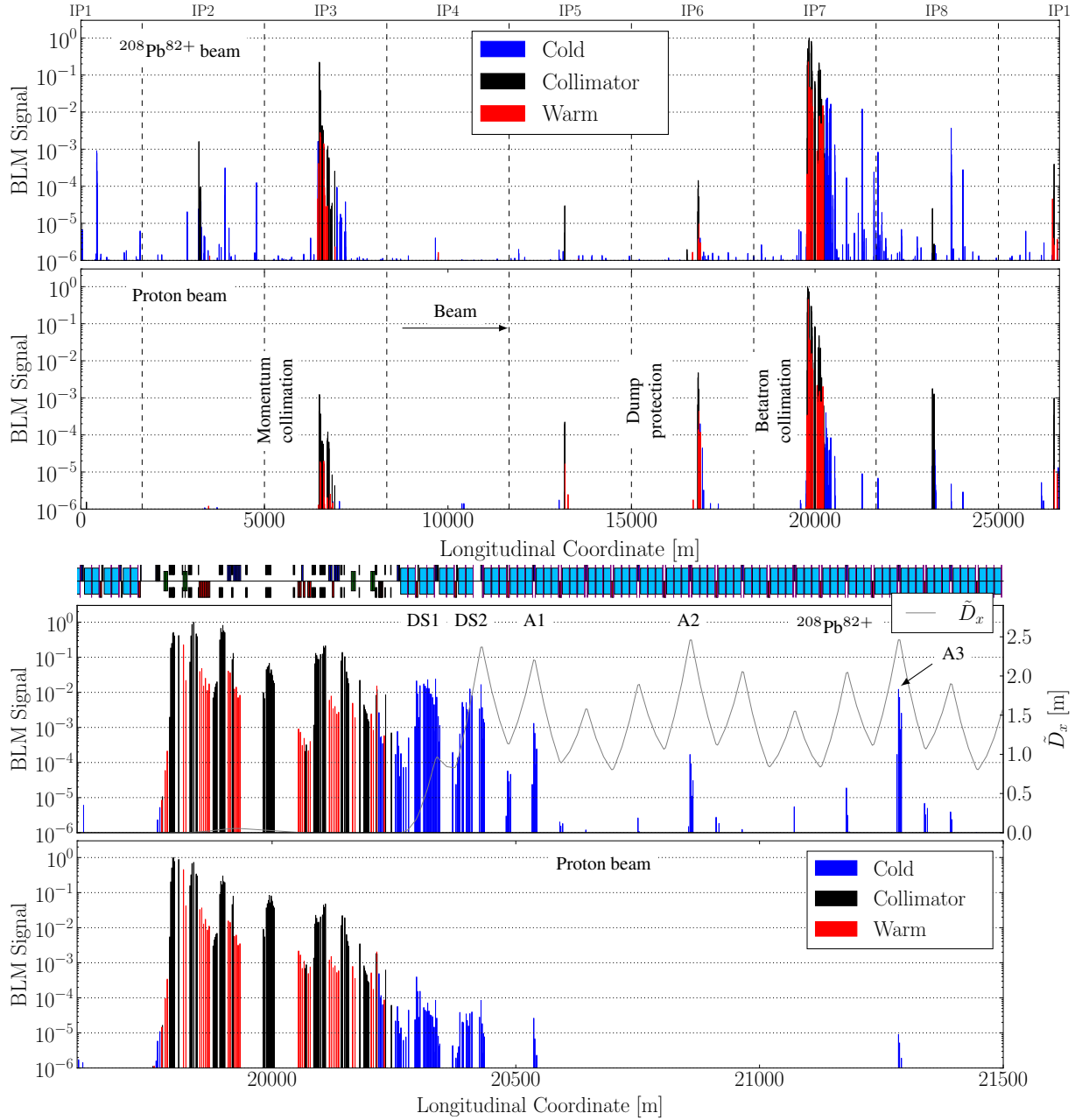


Fig. 2: B1H qualification loss maps measured in 2011 with proton [10] and $^{208}\text{Pb}^{82+}$ beams [11] at 3.5 Z TeV with identical collimator settings and optics, except in IR2. The longitudinal coordinate describes the distance from IP1. The vertical dashed lines mark the LHC octants. The BLM signals are normalized with respect to the highest measured BLM signal. Top: full LHC ring, bottom: zoom to IR7.

In Fig. 2, qualification loss maps for the horizontal plane of LHC beam 1 are compared for proton and heavy-ion beams at an energy of 3.5 Z TeV in 2011 (Z indicating the particle charge multiplicity) [12]. The BLM signal is always normalized with respect to the highest loss signal measured throughout the ring. The loss signals are color-coded to distinguish between losses in superconducting elements (blue), normal conducting elements (red) and losses at collimator (black).

In the upper plot in Fig. 2 it is clearly visible how the highest losses throughout the ring occur at the betatron collimation system in IR7 for both cases. For heavy-ion beams, also high losses in the

momentum collimation region IR3 are measured, which are smaller by two orders of magnitude for proton beams. Furthermore, the global picture illustrates that high loss peaks in superconducting LHC regions are visible for operation with $^{208}\text{Pb}^{82+}$ ions all over the LHC ring. For proton operation, the losses in cold regions are localized closely to the collimators. The highest losses in cold regions are visible in the dispersion suppressor region immediately downstream of IR7 for both cases. However, the loss signal in heavy-ion operation is larger by approximately two orders of magnitude. Hence, the subject of heavy-ion collimation should be studied rigorously to ensure operation without beam induced magnet quenches.

The importance of heavy-ion collimation is furthermore underlined by the 2015 heavy-ion collimation quench test, in which the first beam induced magnet quench was achieved at the LHC [13]. In this experiment, high losses at the collimation system were provoked and the particles scattered out of the collimators quenched the MBB.9R7 when a power load of $P_q = (15 \pm 1)$ kW was deposited on the LHC collimators. The experimental data gathered in this experiment indicates that the beam intensity envisaged for HL-LHC may not be achievable with the present collimation system [4].

The theoretical understanding of losses from collimation inefficiency is crucial for the optimization and upgrade of the collimation system. Laborious simulations require to compute the particle motion through the accelerator lattice, simulate the interaction of the beam particles with collimators, follow the tracks of the out-scattered particles in the lattice and determine their loss location by comparing the tracks with the machine aperture. It is hence required to combine accurate particle tracking simulations with particle-matter interaction simulations and exchange information between them. From the comparison of measured loss maps it becomes clear that the collimation of proton and heavy-ion beams are fundamentally different, which must necessarily reflect in the simulation software. This applies especially to the losses in the IR7 DS region, which are induced by particles of an unmatched rigidity scattered out of the collimators. The keystone in the understanding of this difference lies in the interactions protons and heavy-ions can undergo in the collimators.

2.2 Particle Matter-Interaction in Collimators

Heavy-ions are composed of multiple protons and neutrons. The nuclide used in the LHC heavy-ion runs so far is $^{208}\text{Pb}^{82+}$, with 208 nucleons in total. At the passage through the collimator, the heavy ions can interact with the atoms of the collimator material which can lead to scattering, energy loss and fragmentation. The type of interaction that is undergone is statistically dependent on the projected distance b of the heavy-ion (projectile) from the nucleus of the atom of the collimator material (target), the so called impact parameter (see Fig. 3). The most important effects in the theoretical description of heavy-ion-matter interaction are listed in the following.

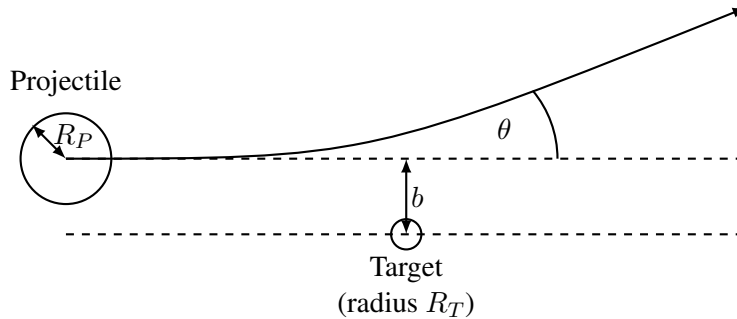


Fig. 3: Schematic illustration of the impact parameter.

2.2.1 Multiple Coulomb Scattering

Multiple Coulomb Scattering (MCS) is the repeated elastic scattering of the projectile due to the Coulomb interaction with the target nuclei. MCS leads to a broad spectrum of scattering angles when a projectile traverses the collimator. The mean scattering angle $\langle \Delta x'_{\text{MCS}} \rangle$ can be described by the Molière formula, as presented in [14]. Given its independence of the particle mass and charge, the mean MCS angle is identical for protons and heavy-ions (see Tab. 1).

2.2.2 Energy Loss From Ionization

The traversing particles can ionize atoms of the target material, resulting in energy loss of the projectile. The mean energy loss from ionization per traversed distance in the material can be well quantified by the Bethe-Bloch formula [14]. The latter shows that the mean energy loss per length is quadratically proportional to the particle charge q .

2.2.3 Photonuclear Interactions

The charged projectile is surrounded by a Lorentz-contracted electromagnetic field which yields in a high photon flux between projectile and target nucleus. These photons can excite one or both of the nuclei into a higher energetic state. The nucleus deliberates from this excited state preferably by the emission of one neutron. Generally, this type of photonuclear reaction, referred to as electromagnetic dissociation (EMD) of the order m , can be summarized as [15]

$$\text{EMD}_m : {}^{208}\text{Pb}^{82+} + {}^{12}\text{C} \rightarrow ({}^{208-m})\text{Pb}^{82+} + {}^{12}\text{C} + m\text{n}. \quad (1)$$

Photonuclear reactions typically occur at ultraperipheral nuclear encounters, e.g. the sum of the nuclear radii is smaller than the impact parameter $b \geq R_P + R_T$. The most important process of EMD is the neutron emission of first order. For the reference species ${}^{208}\text{Pb}^{82+}$ this corresponds to the dissociation into ${}^{207}\text{Pb}^{82+}$. This isotope of Pb was shown to induce distinct losses in the LHC that can potentially impose a limitation on the achievable heavy-ion luminosity (see Sec. 5.2).

2.2.4 Nuclear Interactions

If the impact parameter is smaller than the sum of the radii of projectile and target nucleus, the particle can undergo nuclear interactions [16]. In these interactions the overlapping nuclei ablate parts of each other, resulting in nuclear fragmentation into a wide spread of different nuclei. Nuclear reactions of ${}^{208}\text{Pb}^{82+}$ with the carbon atoms of the collimator material can be summarized by the reaction formula

$${}^{208}\text{Pb}^{82+} + {}^{12}\text{C} \rightarrow {}^{A_X}\text{X}^{Z_X+} + {}^{A_Y}\text{Y}^{Z_Y+} + {}^{12}\text{C} + N_n\text{n} + N_p\text{p} + \dots \quad (2)$$

Here, X and Y are nuclides of a different species than ${}^{208}\text{Pb}^{82+}$ with their respective mass and charge numbers A_X, Z_X and A_Y, Z_Y . Furthermore N_n neutrons and N_p protons may be produced in the interaction. It is clear that also other particles, e.g. pions, electron, photons, etc. are produced in these interactions. They are neglected in the framework of heavy-ion collimation software because the energy they carry is small compared to the heavy-ion fragments.

The most important nuclear interaction protons undergo with the nuclei of the collimator material is single diffractive scattering [17]. In this type of interaction, the proton is excited into a high mass state which desintegrates into a new proton and another particle. Evidently, the momentum of the created proton is reduced with respect to the incidenting one. This momentum offset is the cause for the subsequent particle loss in the DS if the scattering angle in the TCP is not sufficiently large to intercept the proton with the secondary collimators.

Table 1: Characteristic quantities for the most important physical processes of protons and lead ions traversing carbon-fiber composite (CFC) [1] at 7 Z TeV. Data taken from [20] and scaled to the density of CFC.

Physics Process	Unit	Proton	$^{208}\text{Pb}^{82+}$
$-\frac{dE}{E dx}$	$[10^{-5} \text{ m}^{-1}]$	11	917
$\langle \Delta x'_{\text{MCS}} \rangle$	$[\mu\text{rad}/\sqrt{m}]$	4.0	4.0
λ_{nuclear}	[cm]	47.9	3.1
λ_{EMD}	[cm]	-	23.9

2.2.5 Nuclear Evaporation and Statistical Fragmentation

Heavy ions which have undergone a nuclear or photonuclear interaction are often in excited states. The ground state is reached by a variety of physical processes, including nuclear fission, nucleon emission or photon emission. The corresponding nuclear disintegration is referred to as nuclear evaporation [18]. Ion fragments of low masses may undergo statistical fragmentation, described by the Fermi breakup model [19].

2.2.6 Summary

The key parameters quantifying the aforementioned interactions are listed in Tab. 1. The mean scattering angle from MCS is identical for protons and heavy ions. The mean energy loss per length unit is larger for heavy-ions by almost two orders of magnitude. The nuclear interaction length λ_{nuclear} is ten times smaller for $^{208}\text{Pb}^{82+}$ than for protons. Significantly more particles with rigidity offsets are hence generated at the passage of $^{208}\text{Pb}^{82+}$ through the collimator material.

3 The Physics behind Heavy-Ion Tracking

3.1 Sources of Dispersion

Let A, Z, m be the mass number, charge number and rest mass of the tracked nucleus, and A_0, Z_0, m_0 the corresponding quantities for the reference particle. The magnetic rigidity for the reference particle with momentum P_0 , charge $q_0 = Z_0 e$ in a magnetic field B is given as

$$B \rho_0 = \frac{P_0}{q_0}, \quad (3)$$

where ρ_0 is the bending radius of the particle trajectory in the magnetic field. A particle with different momentum P , mass m or charge q will be subject to dispersion which results in a different bending radius ρ . The dispersion can be quantified by considering the ratio of magnetic rigidities of the particle with respect to the reference particle:

$$\frac{\rho}{\rho_0} = \frac{P}{P_0} \frac{q_0}{q}. \quad (4)$$

If the particle is of the same species as the reference particle ($q = q_0$ and $m = m_0$), the bending radius is given as

$$\frac{\rho}{\rho_0} = \frac{P}{P_0} = \frac{\beta\gamma}{\beta_0\gamma_0} = (1 + \delta_M), \quad (5)$$

where we used the equality $P = m\beta\gamma c$, relating the momentum with the rest mass, relativistic velocity $\beta = \frac{v}{c}$ and Lorentz factor $\gamma = \frac{1}{\sqrt{1-\beta^2}}$. The quantity δ_M describes the momentum offset of a particle with respect to the reference particle if it is of the same species as the latter (mono-nuclide scenario):

$$\delta_M = \frac{P - P_0}{P_0} = \frac{\beta\gamma - \beta_0\gamma_0}{\beta_0\gamma_0}. \quad (6)$$

If the particle is of a different species than the reference particle (multi-nuclide scenario), the ratio of bending radii becomes

$$\frac{\rho}{\rho_0} = \frac{m}{m_0} \frac{q_0}{q} \frac{\beta\gamma}{\beta_0\gamma_0} = \frac{1 + \delta_P}{\chi}, \quad (7)$$

where δ_P and χ are defined as

$$\delta_P = \frac{\beta\gamma - \beta_0\gamma_0}{\beta_0\gamma_0} = \frac{P \frac{m_0}{m} - P_0}{P_0} \quad \text{and} \quad \chi = \frac{q}{q_0} \frac{m_0}{m}. \quad (8)$$

The quantity δ_P describes the ion's offset of the momentum per mass unit relative to the reference particle. The dispersion due to the different mass and charge is described by χ , which is the mass to charge ratio, relative to the reference particle. These independent quantities describe the dispersive offset in a given optical accelerator lattice.

3.2 Tracking Ions as Protons of Equivalent Rigidity

The comparison of Eq. (5) and (7) shows that the bending behaviour of an ion different from the reference species (described by χ and δ_P) is identical to that of an ion of the reference species, if the momentum offset of the latter is given by

$$\delta_M = \frac{1 + \delta_P}{\chi} - 1. \quad (9)$$

Hence, tracking software designed for tracking mono-nuclide beams can be used also for multi-isotopic beams, if an *effective momentum* is applied to the particles of the reference species. The most important tracking software for proton beams used at CERN is SixTrack [], providing an environment for symplectic proton tracking over a large number of turns. If an effective momentum P_{eff} to the protons is applied, the proton trajectory will be identical to the heavy-ion trajectory. Using Eq. (9), the effective proton momentum to simulate the motion of an arbitrary ion with (q, m) in a magnetic lattice matched to the reference ion with (q_0, m_0) can be deduced as:

$$P_{\text{eff}} = P_0 \frac{1 + \delta_P}{\chi} = \frac{P}{Z}. \quad (10)$$

This approach is used in the simulation tool STIER [12] (see Sec. 4.3).

3.3 Symplectic Tracking Maps from a Multi-Nuclide Accelerator Hamiltonian

Instead of using unphysical parameters in the tracking (effective proton momenta), a more generic approach of computing heavy-ion trajectories can be provided from a generalized mathematical formalism. Starting point for the description of symplectic multi-nuclide tracking is a generalized accelerator Hamiltonian. The latter should take into account that the tracked particle may have mass and charge different from the reference particle. The derivation of this Hamiltonian is explained in detail in [4, 22]. We refer to the reference frame shown in Fig. 4. The final shape of the generic accelerator Hamiltonian for

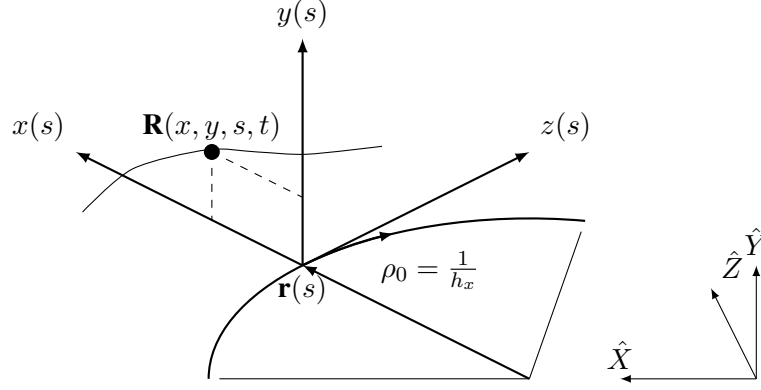


Fig. 4: Accelerator coordinate system (x, y, z) moving with $\mathbf{r}(s)$. The radius of the bent trajectory is $\rho_0 = \frac{1}{h_x}$. Figure taken from [21].

arbitrary particles with relative mass to charge ratio χ , momentum per mass offset $\delta = \delta_P$ moving in a magnetic field $\mathbf{B} = \nabla \times \mathbf{A}$ is given by

$$H = p_\sigma - (1 + h_x(s)x) \left(\sqrt{(1 + \delta)^2 - (p_x - \chi a_x(s))^2 - (p_y - \chi a_y(s))^2} + \chi a_s(s) \right). \quad (11)$$

The quantity $h_x = \frac{1}{\rho_0}$ is the curvature of the reference trajectory. The canonical coordinates associated to this Hamiltonian are

$$(x, p_x) (y, p_y) (\sigma, p_\sigma), \quad (12)$$

with the longitudinal position coordinate σ and its canonical conjugate p_σ given as

$$\sigma = s - \beta_0 c t, \quad p_\sigma = \frac{\frac{m_0}{m} E - E_0}{\beta_0 P_0 c}. \quad (13)$$

Here, s is the longitudinal particle position, c is the speed of light, t the time and E, E_0 the energy of the particle and reference particle. Starting from this generic accelerator Hamiltonian, the tracking maps for the accelerator lattice elements can be derived, as discussed in [22].

4 Heavy-Ion Collimation Simulations

4.1 Introduction

Before discussing heavy-ion collimation tools we give a brief overview on the simulation of proton collimation. Proton collimation at the LHC the best studied scenario with the largest amount of measured data available to compare with the simulations. Furthermore, proton collimation is a special scenario of heavy-ion collimation (with an ion composed only of a proton), hence existing proton simulation tools can be upgraded for additional features making it possible to simulate heavy-ion collimation.

The present state of the art simulation tool for proton collimation simulations is SixTrack [17, 23–25], providing symplectic tracking of protons with an integrated Monte-Carlo routine to simulate proton-matter interaction. SixTrack is based on symplectic tracking maps derived from the mono-nuclide accelerator Hamiltonian with $\chi = 1$ and $m = m_0$ for all particles [26–28]. The interaction of protons with the collimators is simulated by means of an integrated Monte-Carlo event generator which simulates scattering from MCS, energy loss from ionization and nuclear effects [17]. The tracks of particles scattered out of the collimators are compared to a detailed model of the LHC aperture to identify the loss location. Particles undergoing inelastic interactions in the collimator which are not single diffractive are

considered to be lost. The number N_p of lost protons per bin of $\Delta s = 10$ cm is normalized by number of particles lost in the LHC collimators N_{tot} to derive the local cleaning inefficiency $\eta(s)$

$$\eta(s) = \frac{N_p(s)}{\Delta s N_{\text{tot}}} . \quad (14)$$

Simulations of the heavy-ion collimation efficiency must take into account the different interactions of heavy-ions with the collimator materials, in particular by EMD and nuclear fragmentation processes. A suitable heavy-ion tracking software must take into account mass and charge of the out-scattered ion-fragments to accurately model dispersion. Hence, specialized software must be used for heavy-ion collimation simulations, requiring additional features compared to SixTrack. In this section an overview of the history of heavy-ion collimation simulation tools for LHC applications is given.

4.2 ICOSIM (2004)

In 2004, the simulation software Ion Collimation Simulation (ICOSIM) was developed to anticipate the heavy-ion collimation efficiency for the LHC [20, 29]. The simulation approach is based on several consecutive steps. In a first linear mapping, an initial sample of particles is tracked turn-by-turn at the primary collimator. Every 100 turns, random transverse kicks are applied to simulate diffusion. This initial tracking is performed until all ions of the initial sample have hit the primary collimator, to determine their impact coordinates. In the next step, the interaction of the ions with the collimator material is simulated using an integrated Monte-Carlo routine, based on tabulated fragmentation cross sections that are generated beforehand with the Monte-Carlo event generator FLUKA [18, 30, 31]. The particle-matter interaction neglects angular scattering from the fragmentation process and only keeps track of the heaviest fragment created in an interaction. The out-scattered ion fragments are then tracked through the LHC ring element by element using matrix multiplication including dispersion in linear approximation. The loss location of the ions is determined by comparing the particle positions with a simplified model of the magnet aperture.

The comparison of lossmaps simulated with ICOSIM to measured BLM data shows that the physics models used in ICOSIM are not detailed enough to accurately model heavy-ion losses from collimation (see [12]). This observation motivated the development of the new simulation tool STIER to overcome the limiting simplifications.

4.3 STIER (2014)

SixTrack with Ion-Equivalent Rigidities (STIER) was developed in 2014 to increase the agreement between measurement and simulations. It is based on proton tracking in SixTrack, using the approach presented in Section 3.2 tracking heavy-ions as protons of equivalent rigidity.

It is hence necessary to perform the particle-matter interaction simulation externally. Given the hierarchy of the LHC collimation system the dominating part of collimation losses can be simulated by considering only the interaction of the primary $^{208}\text{Pb}^{82+}$ beam with the primary collimator, which can be done e.g. with FLUKA. STIER simulations carried out so far used a three-stage simulation chain consisting of

1. An initial optics calculation with MAD-X to produce the SixTrack input and obtain the phase space parameters of the particles impacting the TCP. The latter are used to calculate the impact angle $x'_{l/r}$ on the left and right TCP jaw [32]

$$x'_{l/r} = \mp N_P \alpha_x \sqrt{\frac{\epsilon_x}{\beta_x}} , \quad (15)$$

where α_x, β_x are the Twiss parameters and ϵ_x is the geometric emittance.

2. The fragmentation of the heavy ions at the primary collimator is simulated by means of FLUKA. The collimator is modelled as a block of carbon and the coordinates and species of the out-scattered ion fragments are saved as output. In a post/pre-processing computation the ion-equivalent proton momenta are computed and saved together with the 4D position coordinates in an input file for SixTrack.
3. The data sampled with FLUKA represents the expected particle distribution of heavy-ion fragments after the passage of the TCP. Hence, the tracking simulation must be started at the TCP. The particle-matter interaction routine in SixTrack is specific to protons, such that it must be evaded in heavy-ion collimation simulations. However, the collimators must be included in the simulation, to identify heavy-ion losses at collimators. The SixTrack input is therefore modified to treat the collimators as perfect absorbers. The acceleration by the RF cavities is in reality dependent on the particle charge [22], so STIER can only provide 4D tracking of the heavy ions.

Evidently, STIER overestimates losses at collimators and subsequent interactions of fragments scattered out of the TCP are not modelled. However, heavy-ion loss maps simulated with STIER show a good agreement with measured data (see Section 5.1). Compared to its predecessors the predictive power of STIER is significantly enhanced, mainly due to the inclusion of angular scattering and energy transfer from the fragmentation process [12]. The development of STIER was a major step in the theoretical understanding of heavy-ion losses at the LHC.

4.4 hiSixTrack (2016)

Simulations with STIER can deliver good predictions of the LHC cleaning performance but the applied simplifications restrict the spectrum of applications. Besides simulations of the cleaning performance, estimates on the deposited energy in the magnet coils are important for the operation of the LHC [33]. They can be obtained by detailed shower propagation simulations, typically carried out with FLUKA. The input for these simulations is the distribution of impacting ions on the collimators, which must be generated with a collimation simulation tool. STIER can not be used for this purpose because the ions are tracked as protons and the particle-matter interaction simulation requires the information on the particle species. Furthermore, particles scattered out of collimators subsequent to the TCP may contribute to the loss pattern, which can only be simulated by performing particle-matter interactions at all collimators. Therefore it is necessary to store information on the particle species in the tracking routine. To enable these functionalities, the software heavy-ion SixTrack (hiSixTrack) [22, 34] was implemented as a successor of STIER.

4.4.1 Tracking

hiSixTrack is based on SixTrack in which additional tracking arrays store mass and charge of the tracked ions have been implemented. The proton tracking maps were replaced by the multi-nuclide tracking maps presented in [4, 22]. The calculation of χ requires to define nucleon number, charge number and rest mass of the reference species. Thanks to the generic implementation, hiSixTrack is the first heavy-ion tracking tool allowing to simulate longitudinal particle dynamics and hence synchronous six-dimensional symplectic tracking of arbitrary particles.

The tracking routine of hiSixTrack was extensively tested to ensure their accuracy. In Fig. 5, heavy-ion tracks simulated with hiSixTrack and STIER are compared against each other. The tracks are calculated for different nuclides with $\delta = 0$ but $\chi \neq 1$ starting from IP1 in a LHC lattice matched to $^{208}\text{Pb}^{82+}$. Similar comparisons were carried out also for other scenarios with different values of δ and χ (see [4]). In all studied cases a full numerical agreement of the heavy-ion tracks was observed between hiSixTrack and STIER, even for a large number of tracked turns through the LHC ring. Both models hence simulate the 4D particle tracks with the same precision.

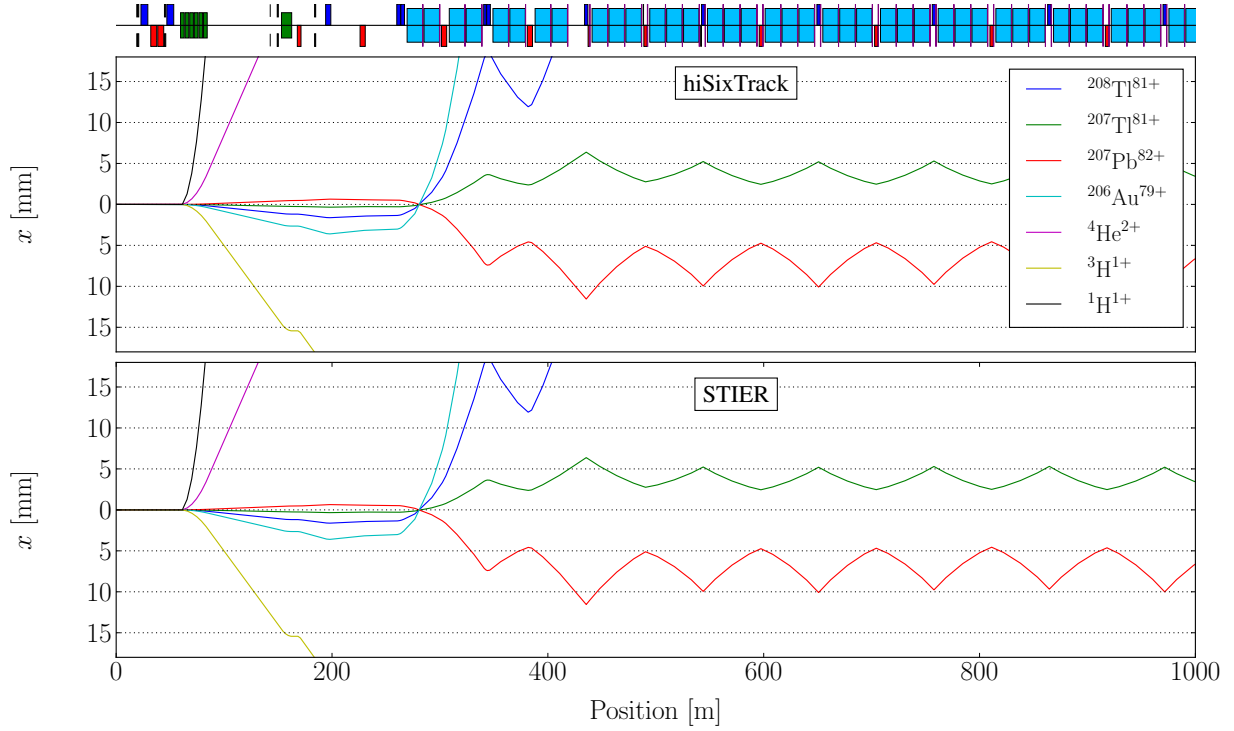


Fig. 5: Heavy-ion tracks for different on-momentum ($\delta = 0$) isotopes unmatched to the magnetic lattice ($\chi \neq 1$) starting with the same initial conditions in IP1, simulated with hiSixTrack (top) and STIER (bottom).

In addition to the features provided by STIER, the software hiSixTrack enables the possibility of six-dimensional tracking and on-line coupling to FLUKA (see next chapter). The generic implementation based on the multi-nuclide accelerator Hamiltonian is the most generic approach possible, enabling a high number of unprecedented applications. The generic tracking routine can be further extended to simulate other effects or tracking maps for other beam line elements, e.g. beam-beam interaction, crab cavities or electron lenses [21].

4.4.2 Coupling to FLUKA

The software hiSixTrack can be coupled to FLUKA, allowing the performance of FLUKA particle-matter interaction simulations at each of the collimators. The underlying framework for the so called hiSixTrack-FLUKA coupling [34] is the SixTrack-FLUKA coupling for the simulation of proton collimation [35]. Compared to the latter, the coupling of hiSixTrack to FLUKA requires the exchange of additional information, in particular the information on the species of the exchanged ion. The framework allows to simulate multiple interaction processes in the collimation system, which are impossible to take into account with the STIER approach, where the collimators must be treated as perfect absorbers.

4.4.3 Summary

The coupling between the tracking tool hiSixTrack and FLUKA combines two of the most sophisticated simulation tools for their respective purpose to a powerful environment for the simulation of heavy-ion collimation and other applications. A comparison between simulated and measured loss maps is presented in the next section.

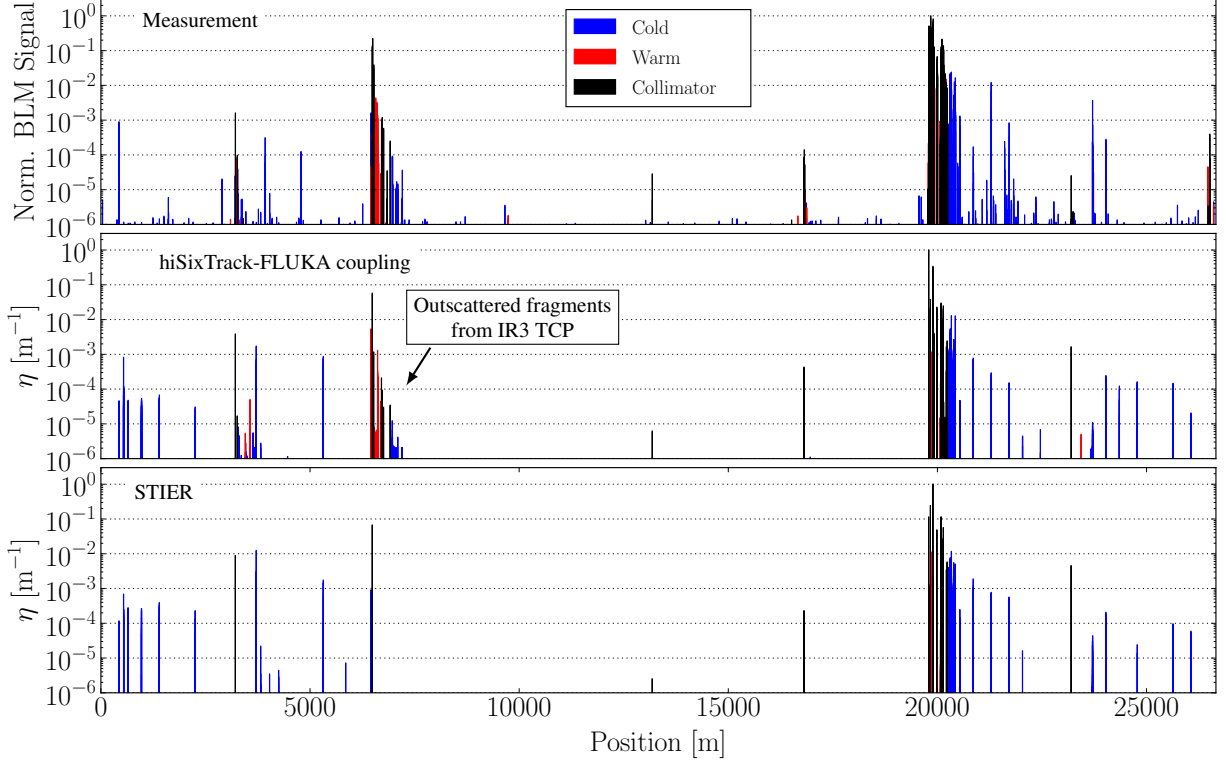


Fig. 6: B1H betatron loss maps measured in the 2011 heavy-ion run compared to simulations with the hiSixTrack-FLUKA coupling and STIER.

5 Applications

5.1 LHC Qualification Loss Maps

Simulated loss maps for heavy-ion collimation are based on a more generic definition of the local cleaning inefficiency

$$\eta(s) = \frac{\sum_i E_i(s)}{\Delta s E_{\text{tot}}}, \quad (16)$$

where E_i is the energy if the ion i lost at the position s and E_{tot} is the total energy of all ions lost in the LHC collimators. This re-definition of the local cleaning inefficiency is necessary because the energy of ion fragments can vary significantly. The energy of a $^{208}\text{Pb}^{82+}$ ion scattered out of a LHC collimator is typically larger by two orders of magnitude compared to an outscattered proton. In addition, the energy per nucleon of the individual ions can vary significantly (see [4]), hence the local cleaning inefficiency is not correctly modelled by counting the number of impacting nucleons.

A comparison of measured and simulated loss maps is shown in Fig. 6 for the LHC in the configuration of the 2011 heavy-ion run at an energy of 3.5 Z TeV. The collimator and optics settings are summarized in [12]. It shall be pointed out that the measured BLM signal and the simulated local cleaning inefficiency should not be compared quantitatively. The measured BLM signal depends on an unknown response function for upstream particle losses, while the simulated cleaning inefficiency is determined with a full and immediate detection of particle losses at the beam pipe.

Both simulation tools show a good agreement with the measured data. The losses in the superconducting regions downstream of IR7 are qualitatively well reproduced. A significant difference between the simulation codes is the modelling of the losses downstream of the momentum collimation region IR3. While these losses are not simulated in STIER, because of the absence of a multi-passage interaction model, they are visible in the simulation with the hiSixTrack-FLUKA coupling. This shows that the

inclusion of the interaction at all collimators does indeed change the loss pattern.

5.2 Mitigation of Losses in Operation

Collimation simulation tools play a crucial role in the understanding and potential mitigation of undesired losses in operation. This shall be illustrated on the example of the 2015 heavy-ion run with $^{208}\text{Pb}^{82+}$ ions.

In late 2015, the LHC was operated with $^{208}\text{Pb}^{82+}$ ion beams at an energy of 6.37 Z TeV [36]. During operation in collision, high losses at the horizontal tertiary collimator left of IR2 (TCT2) were measured (see top panel of Fig. 7). They reached amplitudes of 10^{-2} and induced undesired background to the ALICE experiment. From the affected TCT it was possible to conclude that the losses were caused by particles of beam 1 in the horizontal plane. To understand and potentially mitigate these losses, simulations with STIER were thoroughly analyzed [4].

The STIER simulations for B1H showed that the simulated loss signal at the TCT2 is significantly higher for particles scattered out of the left TCP jaw, than for those scattered out of the right jaw (see middle and bottom panel in Fig. 7). Instead of being lost at the TCT2, the particles starting from the right TCP jaw are lost in the momentum cleaning insertion IR3. The first conclusion to draw from the STIER simulation is that the losses observed in the IR2 TCT should be reduced if the left TCP jaw was retracted and all primary losses took place at the right TCP jaw.

The analysis of the loss composition at the TCT2 shows that 92.5% of the losses are caused by the isotope of the reference species $^{207}\text{Pb}^{82+}$ starting from the left jaw of the primary collimator (see Tab. 2). This isotope is created with a high abundance via EMD. Assuming that a realistic impact parameter was chosen for the FLUKA simulation ($3\mu\text{m}$ in this study case), one can quantify the potential for loss reduction to be approximately one order of magnitude if the left TCP jaw was retracted.

The track of this nuclide simulated with STIER is shown in Fig. 8. It is clearly visible how the particle track is not intercepted by the secondary collimators. The horizontal TCT in IR8 (TCTH.4L8.B1, called TCT8 in the following) is passed by at a very short distance of less than 2σ . The isotope ultimately impacts the TCTH.4L2.B1 after having moved through the LHC lattice for 10km. This result shows that it is essential to have a reliable and accurate tracking routine also for heavy-ions, because the ion fragments from IR7 can travel for long distances through the LHC.

The analysis of the particle track thus shows that, if the simulation is accurate, the secondary beam of $^{207}\text{Pb}^{82+}$ can be intercepted by the TCT in IR8. By estimating the beam size and the simulated distance of the $^{207}\text{Pb}^{82+}$ track from the IR8 TCT, it is also possible to quantify the possible loss reduction as a function of the TCTH.4L8.B1 half gap. A detailed description is given in [4].

Both strategies of loss mitigation were tested with beam in the LHC, and the measured cleaning inefficiency at the TCT2 is illustrated in Fig. 9. In both tested scenarios the prediction made with STIER could be confirmed even quantitatively in LHC operation¹. As expected, the measured BLM signal at

¹Note that a quantitative comparison is only possible because the data is measured at the same BLM.

Table 2: Loss composition at the TCT2 simulated with STIER for B1H in the 2015 configuration using a TCP impact parameter of $2\mu\text{m}$. The energy fraction is calculated from the number of particles of each isotope, weighted with the particle momenta.

Isotope (A,Z)	TCP jaw	Fraction (%)
$^{207}\text{Pb}^{82+}$	left	92.5
$^{204}\text{Tl}^{81+}$	right	3.6
$^{202}\text{Hg}^{80+}$	left	2.2
$^{199}\text{Au}^{79+}$	right	0.3

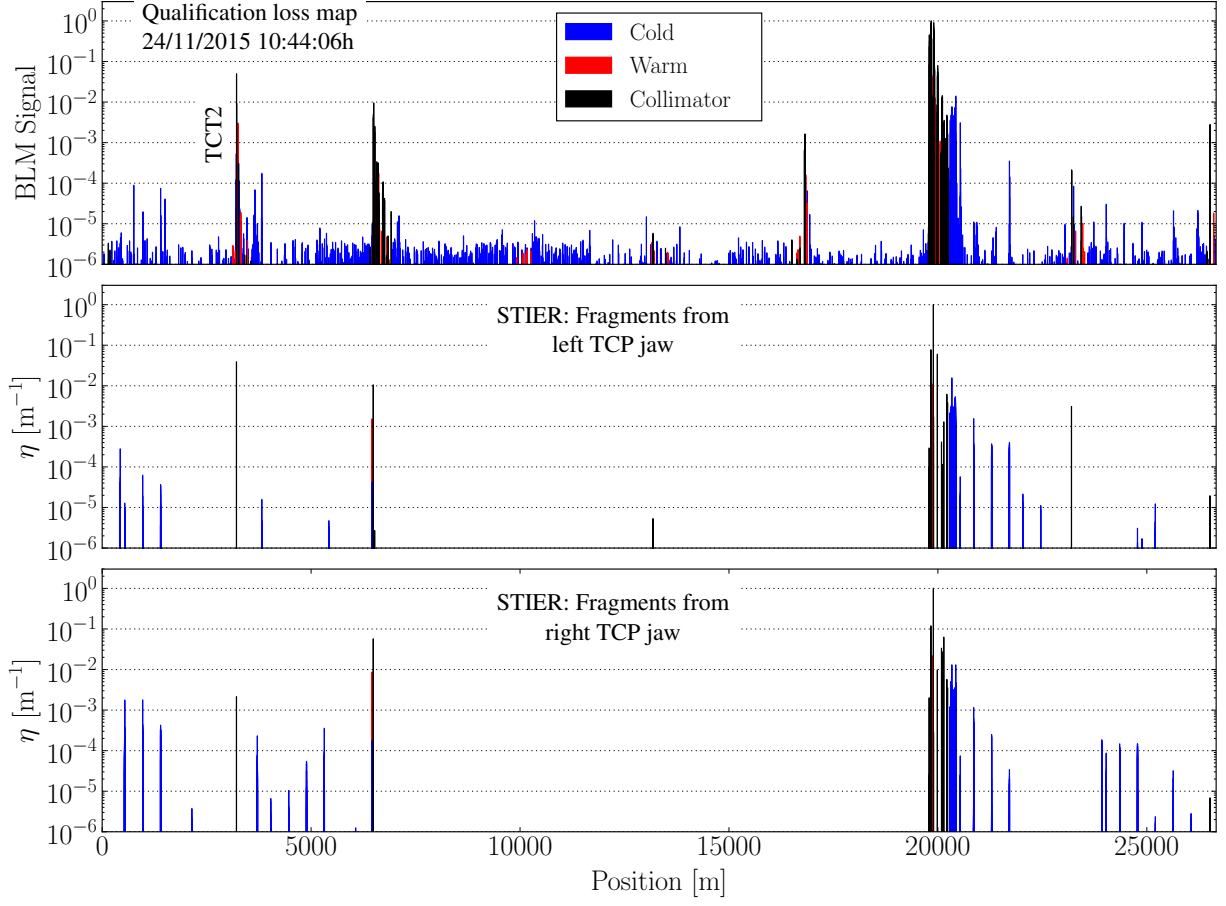


Fig. 7: Top panel: BIH qualification loss map measured in the 2015 heavy-ion run. Middle and bottom panel: STIER simulation result of the $^{208}\text{Pb}^{82+}$ cleaning inefficiency for BIH in the configuration of the 2015 heavy-ion run at 6.37 Z TeV. The simulation result is shown for particles starting at the left and at the right horizontal TCP jaw separately. Both simulations are carried out with 6×10^6 initial $^{208}\text{Pb}^{82+}$ particles.

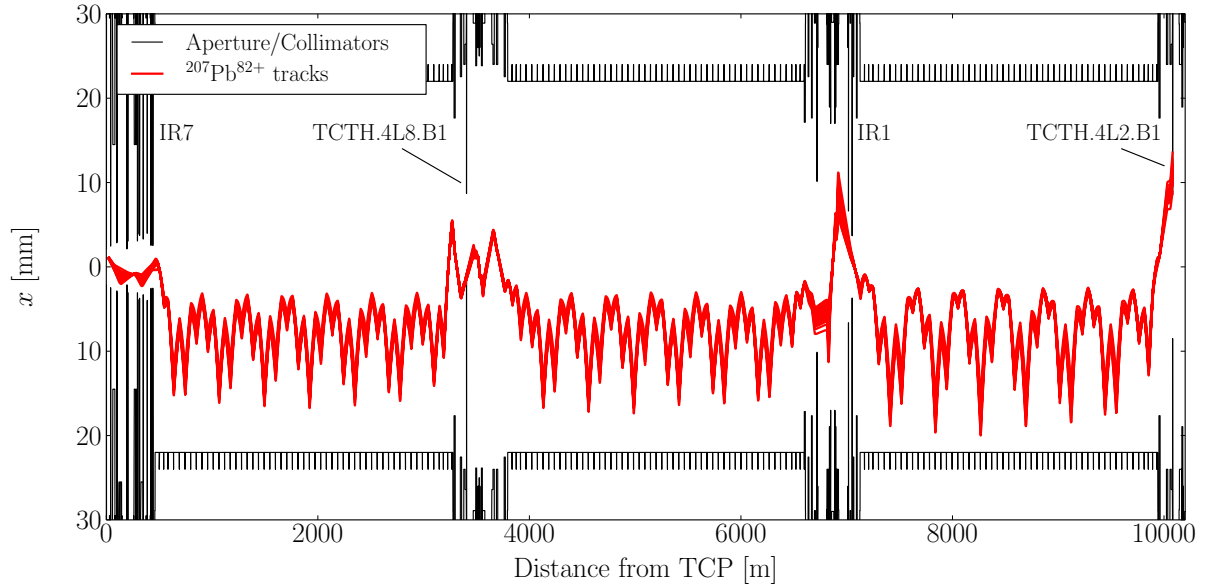


Fig. 8: Simulated track of the secondary $^{207}\text{Pb}^{82+}$ beam starting at the left jaw of the horizontal TCP in IR7, which is lost at the TCT2.

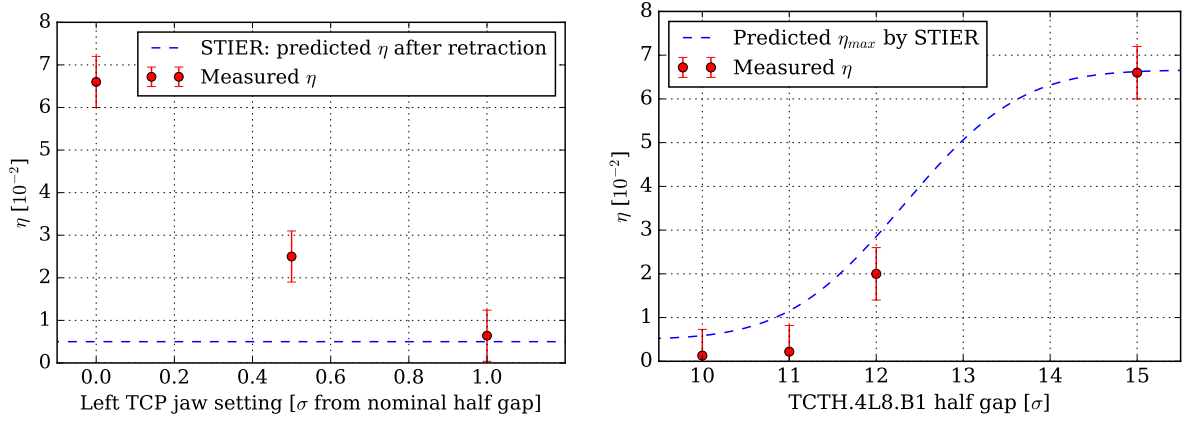


Fig. 9: Left: Measured loss signal at the TCT2 with different settings of the left TCP jaw. The blue dashed line shows the STIER prediction for a fully retracted jaw. Right: Measured loss signal at the TCT2 with different settings of the TCT8

the TCT2 is reduced by approximately one order of magnitude when the left TCP jaw is retracted. The measured TCT2 BLM signal with the different settings of the TCT8 is also in good agreement with the STIER prediction (note that in this case, the prediction refers to the maximum expectable signal for several reasons, see [4]).

In conclusion, the example illustrates how heavy-ion collimation simulations can be used to identify the origin of disturbing loss signals and to develop strategies to mitigate them. Although the 2015 collimator settings were not changed (e.g. non-symmetric TCP settings could have been applied), the knowledge of the loss origin could deliver essential input for the optimization of the collimation system. The experience gained was taken into account in the preparation of the 2016 heavy-ion run.

6 Summary and Outlook

Collimation simulation tools are crucial to provide safe operation uninterrupted by magnet quenches with both proton and heavy-ion beams. In this chapter we have outlined the requirements for heavy-ion collimation simulations, essentially consisting of a combination of reliable fragmentation simulations and accurate tracking tools modelling dispersion from mass and charge differences. It was shown how the heavy-ion tracking tools used for the LHC have evolved over more than a decade with improving the accuracy of the simulated loss maps.

The simulation tool STIER was introduced as a new development to improve the predictive power compared to its predecessor ICOSIM. The present state of the art simulation tool hiSixTrack is providing sophisticated physics models and can be used for a broad variety of applications. With its coupling to FLUKA it can be used to simulate the heavy-ion collimation efficiency with unprecedented accuracy.

The impact heavy-ion collimation simulations can have on the machine in operation was demonstrated for the example of the 2015 heavy-ion run. In this run, the STIER simulations could be used to understand heavy-ion loss patterns. The information was ultimately used to successfully manipulate the losses in the LHC. Such techniques may be crucial also in future heavy-ion operation.

Further analyses of energy deposition in LHC magnets have been carried out with FLUKA, using hiSixTrack simulation data as input [37]. This information will be used in the future to define BLM thresholds and estimate quench limits by combining simulations with experimental data.

References

- [1] O. Brüning *et al.*, *LHC Design Report (Volume I, The LHC Main Ring)*, vol. CERN-2004-003-V-1. CERN, 2004.
- [2] J. M. Jowett *et al.*, “The 2016 Proton-Nucleus Run of the LHC,” *Proceedings of IPAC 2017, Copenhagen, Denmark*, no. TUPVA014, 2017.
- [3] G. Apollinari and others (eds), “High-Luminosity Large Hadron Collider (HL-LHC): Preliminary Design Report,” *CERN Publication*, no. CERN-2015-005, 2015.
- [4] P. D. Hermes, *Heavy-Ion Collimation at the Large Hadron Collider - Simulations and Measurements*. Doctoral thesis, Westfälische Wilhelms-Universität Münster, Germany, 2016.
- [5] R. W. Assmann *et al.*, “The final collimation system for the LHC,” *Proceedings of EPAC 2006, Edinburgh, Scotland*, no. TUODFI01, pp. 986–988, 2006.
- [6] S. Redaelli, “The collimation system: defence against beam loss,” *CERN Courier*, 2013.
- [7] E. Holzer *et al.*, “Beam loss monitoring system for the LHC,” *CERN-AB-2006-009 BI*, 2005.
- [8] E. Holzer *et al.*, “Development, Production and Testing of 4500 Beam Loss Monitors,” *Proceedings of EPAC 2008, Genoa, Italy*, no. TUPC037, p. 1134, 2008.
- [9] R. Bruce *et al.*, “Measurements of heavy ion beam losses from collimation,” *Phys. Rev. ST Accel. Beams*, vol. 12, no. 011001, 2009.
- [10] R. Bruce *et al.*, “Simulations and measurements of beam loss patterns at the CERN Large Hadron Collider,” *Phys. Rev. ST Accel. Beams*, vol. 17, no. 081004, 2014.
- [11] P. D. Hermes *et al.*, “Studies on Heavy Ion Losses from Collimation Cleaning at the LHC,” *Proceedings of HB 2014, East-Lansing, MI, USA*, no. MOPAB43, pp. 138–142, 2014.
- [12] P. D. Hermes *et al.*, “Measured and Simulated Heavy-Ion Beam Loss Patterns at the CERN Large Hadron Collider,” *Nucl. Instr. Meth. Phys. Res. A*, vol. 819, pp. 73–83, 2016.
- [13] P. D. Hermes *et al.*, “LHC Heavy-Ion Collimation Quench Test at 6.37 Z GeV,” vol. CERN-ACC-NOTE-2016-0031 MD, 2016.
- [14] J. Beringer *et al.*, “Review of Particle Physics,” *Phys.Rev.*, vol. D86, p. 010001, 2012.
- [15] H. H. Braun *et al.*, “Hadronic and electromagnetic fragmentation of ultrarelativistic heavy ions at LHC,” *Phys. Rev. ST Accel. Beams*, vol. 17, p. 021006, 2014.
- [16] J. Bartke, *Relativistic Heavy Ion Physics*. World Scientific Publishing, 2009.
- [17] C. Tambasco, “An improved scattering routine for collimation tracking studies at LHC,” Master’s thesis, 2014. CERN-THESIS-2014-014.
- [18] A. Fassò *et al.*, “The physics models of FLUKA: status and recent developments,” *Proceedings of CHEP 2003, La Jolla, California, USA*, no. MOMT005, 2003.
- [19] B. V. Carlson *et al.*, “Fermi breakup and the statistical multifragmentation model,” *Nuclear Physics A*, vol. 876, pp. 77 – 92, 2012.
- [20] H. H. Braun *et al.*, “Collimation of heavy ion beams in LHC,” *Proceedings of EPAC 2004, Lucerne, Switzerland*, no. MOPLT010, p. 551, 2004.
- [21] R. De Maria, M. Fitterer, and M. Fjellstrom, “SixTrackLib Manual,” 2015.
- [22] P. D. Hermes, R. Bruce, and R. De Maria, “Symplectic Tracking of Multi-Isotopic Heavy-Ion Beams in SixTrack,” *Proceedings of IPAC 2016, Busan, Korea*, no. TUPMW015, pp. 1450–1453, 2016.
- [23] F. Schmidt, “SixTrack Version 4.2.16 Single Particle Tracking Code Treating Transverse Motion with Synchrotron Oscillations in a Symplectic Manner - User’s Reference Manual,” 1994. CERN/SL/94-56 (AP).
- [24] T. Trenkler and J. Jeanneret, “K2: A Software Package evaluating Collimation Systems in Circular Colliders (Manual),” *CERN SL Note, SL/Note 94-105 (AP)*, 1994.

- [25] G. Robert-Demolaize *et al.*, “A new Version of SixTrack with Collimation and Aperture Interface,” *Proceedings of PAC 2005, Knoxville, Tennessee, USA*, no. FPAT081, p. 4084, 2005.
- [26] D. P. Barber, G. Ripken, and F. Schmidt, “A non-linear canonical formalism for the coupled synchro-betatron motion of protons with arbitrary energy,” vol. DESY-87-036, 1987.
- [27] K. Heinemann, G. Ripken, and F. Schmidt, “Construction of Nonlinear Symplectic Six-Dimensional Thin-Lens Maps by Exponentiation,” vol. DESY-95-189, 1995.
- [28] G. Ripken and F. Schmidt, “A symplectic Six-Dimensional Thin-Lens Formalism for Tracking,” vol. CERN-SL-95-12 (AP), 1995.
- [29] N. Holden, “Development of the ICOSIM Program and Application to Magnetised Collimators in the LHC,” Tech. Rep. CERN-AB-Note-2008-054, 2008.
- [30] A. Ferrari, P. R. Sala, A. Fasso, and J. Ranft, “FLUKA: A multi-particle transport code (Program version 2005),” *CERN-2005-10 (2005)*, *INFN/TC_05/11*, *SLAC-R-773*, 2005.
- [31] T. T. Böhlen *et al.*, “The FLUKA code: developments and challenges for high energy and medical applications,” *Nuclear Data Sheets*, vol. 120, pp. 211–214, 2014.
- [32] H. Wiedemann, *Particle accelerator physics I: basic principles and linear beam dynamics*, vol. 1. Springer Verlag, 1999.
- [33] E. Skordis *et al.*, “Impact of Beam Losses in the LHC Collimation Regions,” *Proceedings of IPAC 2015, Richmond, VA, USA*, no. TUPTY046, pp. 2116–2119, 2015.
- [34] P. D. Hermes *et al.*, “Simulation of Heavy-Ion Beam Losses with the SixTrack-FLUKA Active Coupling,” *Proceedings of IPAC 2016, Busan, Korea*, no. WEPMW029, pp. 2490–2493, 2016.
- [35] A. Mereghetti *et al.*, “SixTrack-FLUKA active coupling for the upgrade of the SPS Scrapers,” *Proceedings of IPAC 2013, Shanghai, China*, no. WEPEA064, pp. 2657–2659, 2013.
- [36] J. M. Jowett *et al.*, “The 2015 Heavy-Ion Run of the LHC,” *Proceedings of IPAC 2016, Busan, Korea*, no. TUPMW027, pp. 1493–1496, 2016.
- [37] E. Skordis *et al.*, “Study of the 2015 Top Energy LHC Collimation Quench Tests through and advanced Simulation Chain,” *Proceedings of IPAC 2017, Copenhagen, Denmark*, no. MOPAB012, 2017.

Crystal implementation in SixTrack for proton beams

D. Mirarchi, S. Redaelli, W. Scandale*

CERN, European Organization for Nuclear Research, CH-1211 Geneva 23, Switzerland

Abstract

Crystal channeling is a property of crystals resulting from the extreme order in which the atoms are arranged: the crystalline lattice. Positive charged particles can get trapped between crystalline planes, and bending highly pure crystals allows deflecting high energy beams. Thus, they are very interesting for applications as primary collimator. The crystals installed in the Large Hadron Collider (LHC) provide a deflection that is equivalent to a magnetic field of about 300 T ideally acting only on halo particles. The implementation of a crystal routine in the framework of the collimation tools used at CERN for simulations of expected beam loss pattern around the entire machine (*SixTrack*) is therefore mandatory. The crystal routine presented here is suited for high statistics tracking simulations in large hadron accelerators. An introduction to the crystal physics relevant for our purposes is reported, on which the models implemented in the routine are based. Then, the implementation of these models in *SixTrack* is described.

Keywords

SixTrack; crystal; collimation.

1 Introduction

Coherent interactions in crystals are a very wide subject with many applications [1]. The main process of our interest is the planar channeling of positive charged particles. This allows to use efficiently bent crystals for the manipulation of hadron beams. In particular, the large deflection that can be given to channeled particles, makes crystals suitable for application as primary collimators. Moreover, the reduced nuclear interaction rate of particles trapped between crystalline planes with respect to particles traveling in amorphous materials, lead to a significant reduction of off-momentum losses around the accelerator [2].

The crystal routine presented here is based on a Monte-Carlo approach, so that interactions experienced when a particle traverses a crystal are randomly extracted from probability distributions as a function of the impact parameter. Particles are not tracked step by step inside the crystal volume but the various interactions possible are defined with a Monte-Carlo approach. This implementation was motivated by the need of a very fast routine that is suitable for high statistics simulations, i.e. about 10^7 protons tracked for thousands of turns in the accelerator. However, it can be considered as an emulation of the interactions that particles experience in a bent crystal, rather than a simulation that would involve solution of the equation of motion inside a crystalline potential. Although an approach based on first principles would allow simulations of a any crystalline structure, the time needed to solve the equation of motion would make complete loss map simulations unfeasible. Given the very high statistics and number of turns in the machine needed, an approach based on random extraction of interaction experienced from probability distributions was preferred. This is possible because all the known interactions that can occur in bent crystals are well described in the literature, and the few free parameters can be tuned using

*daniele.mirarchi@cern.ch

experimental data. It is also important to note that this routine, and what is reported later, is only valid for planar channeling of protons in silicon strip and quasi-mosaic bent crystals.

The tracking in the magnetic lattice of the machine is performed with SixTrack [3–6], which allows a symplectic, fully chromatic and 6D tracking, taking into account interactions with the ring collimators and the detailed aperture model of the entire machine. This code represents the standard tool used for accelerator physics studies at CERN, such as dynamic aperture and beam collimation. The complete benchmarking of the crystal routine and of its implementation in SixTrack are reported in [7, 8].

2 Crystal physics

At beginning of the twentieth century, physicists observed that a beam of charged particles can emerge from crystals, rather than be absorbed completely as happens in any other amorphous material of sufficient thickness. Driven by these observations Stark [9] made the hypothesis that in crystals an ordered internal structure is present. Thus, subsequent coherent interactions with such an ordered lattice can allow the particles to emerge from crystals. In amorphous materials the energy released by ionization as a result of the large number of random scatters leads to particle absorption. Only Si crystals are treated here, since they are the most suitable candidates to be used as particle deflectors, given their well developed manufacturing processes that lead to crystalline structures almost without imperfections [10, 11]. If this perfect crystalline structure is well oriented with respect to the incident particles, they will see it either as ordered planes or rows of atoms. Thus, particles can undergo coherent scattering and become trapped between planes (*planar channeling*) or on an axis (*axial channeling*). Given the higher efficiency of the planar channeling compared to the axial one, it is preferred for our applications. Moreover, planar channeling is preferred also because of easier operational manipulations needed to establish stable channeling. An additional degree of freedom would be needed to achieve the right orientation with respect to the beam envelope to channel particles between crystalline axis.

The theoretical approach reported in this paper is based on a classical treatment of interactions between particles and crystals, rather than using quantum mechanics. This is justified by two main reasons:

1. As shown in Section 2.1 particles trapped between crystalline planes oscillate in a harmonic potential. Therefore, their transverse energy is quantized and the number of energy levels is given by [12]:

$$n = \frac{d_p}{\hbar\sqrt{8}} \sqrt{U_{max}m\gamma}, \quad (1)$$

where d_p is the distance between crystalline planes, U_{max} is the maximum of the potential well, and $m\gamma$ is the relativistic mass of the particle. Thus, if $n \gg 1$ it can be approximated as a continuous spectrum.

2. If the transverse de Broglie wavelength ($\lambda = h/p$, where p is the particle momentum) is much smaller than the channel width, the tunneling effect can be neglected.

At 120 GeV, the lowest energy of interest for collimation test purposes at CERN, one obtains: $n \sim 10^{13}$ and $\lambda \sim 10^{-17}$ m where $U_{max} \sim 20$ eV and $d_p = 1.92$ Å are used. Thus, the two conditions above are always fulfilled by ultra-relativistic particles.

2.1 Straight crystals

In order to derive a theoretical formulation of channeling phenomena, the potential between a particle and an atom is required. According to the Thomas-Fermi model the potential ($V(r)$) can be described as:

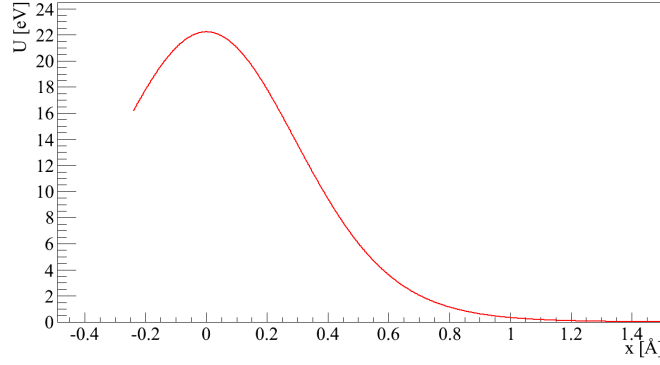


Fig. 1: Potential given by a single (110) silicon plane in the Molière approximation at room temperature.

$$V(r) = \frac{Z_i Z e^2}{r} \Phi\left(\frac{r}{a_{TF}}\right), \quad (2)$$

where $Z_i e$ is the charge of the impinging particle, Z is the atomic number of the target atom, r is the relative distance, and $\Phi(\frac{r}{a_{TF}})$ is a Molière screening function that takes into account the electronic cloud around the nucleus [13]. Lindhard [14] asserts that: “under the hypothesis of small impact angle of the impinging particle with respect to the crystalline plane, we can consider the average potential generated from the entire crystalline plane as a continuous potential” given by:

$$U_p(x) = Nd \iint_{-\infty}^{+\infty} V(x, y, z) dy dz, \quad (3)$$

where x is the coordinate perpendicular to the crystal planes, N is the atomic density, d the distance from the plane and $V(x, y, z)$ is the potential in equation (2). Thermal agitation must also be taken into account. Considering this motion as independent of the location in the crystal and using a Gaussian spatial distribution for the atoms in the plane, the potential is given by the average over this distribution. The potential seen by a positive charged particle from an entire crystalline plane is illustrated in Fig.1.

Superimposing two planes, it is easily understandable that the potential (close to the minimum) seen by a particle between them will be a harmonic potential. This follows from the assumption that particles are influenced only by the potential of the closest planes; thus the entire potential well affecting the particle motion between crystalline planes can be approximated as:

$$U(x) \approx U_p\left(\frac{d_p}{2} - x\right) + U_p\left(\frac{d_p}{2} + x\right) \approx U_{max} \left(\frac{2x}{d_p}\right)^2. \quad (4)$$

The outcome of exact calculation and the harmonic approximation of the potential above is shown in Fig. 2. A particle can be trapped between crystalline planes if its transverse momentum is lower than the maximum of the potential well. This implies the presence of a maximum impacting angle with respect to crystalline planes, the so called *critical channeling angle* (θ_c), below which channeling can occur. This angle depends on the particle momentum (pv) as [1]:

$$\theta_c = \sqrt{\frac{2U_{max}}{pv}}. \quad (5)$$

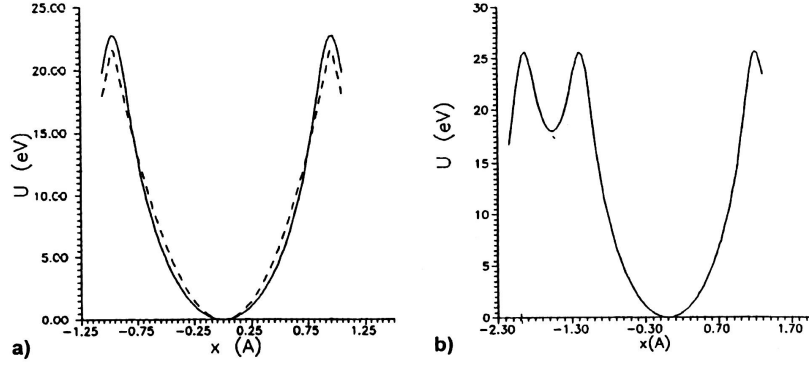


Fig. 2: Potential seen by a proton entering between crystalline planes at a small angle [1]. a) in silicon strip crystals where (110) planes are used, and with dashed line its harmonic approximation. b) in silicon quasi-mosaic crystals where (111) planes are used, with their characteristic ratio 1:3 of subsequent planes.

Using the harmonic approximation of the potential in which channeled particles are confined, they will follow a sinusoidal trajectory described by [1]:

$$x(z) = \frac{d_p}{2} \sqrt{\frac{E_t}{U_{max}}} \sin\left(\frac{2\pi z}{\lambda} + \phi\right). \quad (6)$$

The oscillation phase ϕ is determined by the conditions at the entry of the crystal, and the oscillation period in the channel (λ) is given by:

$$\lambda = \pi d_p \sqrt{\frac{pv}{2U_{max}}}. \quad (7)$$

In conclusion, particles undergoing planar channeling will oscillate between crystalline planes, in a relatively empty space compared to what is present in an amorphous material. This is one of the most important features of crystals used in a particle accelerator to coherently steer particles for halo collimation purposes. It is worth introducing here the two main families of bent crystals: Strip (ST) and Quasi Mosaic (QM) crystals. In ST crystals the (110) planes are used to channel particles, while in QM crystals the (111) planes. The main difference is that the (110) planes are equidistant, whereas a ratio 1:3 is present in subsequent (111) planes, leading to the potential in Fig. 2 a) and b), respectively. This difference can be neglected in terms of channeling efficiency if protons are channeled, while it may become significant in case of channeling of heavy ions. The difference could be explained by the higher probability to experience nuclear interactions when heavy ions are trapped in the smaller channel of QM crystals.

2.2 Bent crystals

When a crystal is bent, the behavior of channeled particles does not differ significantly compared to what occurs in straight crystals. One can demonstrate that bent crystals can be simulated by adding a centrifugal force contribution to the potential of equation (4) [12]. Therefore, it is possible to define an effective potential as:

$$U_{eff}(x) = U(x) + \frac{pv}{R}x. \quad (8)$$

where pv/R is the term of centrifugal force in a crystal with bending radius R . The trajectory performed between crystalline planes is still sinusoidal, but ranges around a new equilibrium point due to the cen-

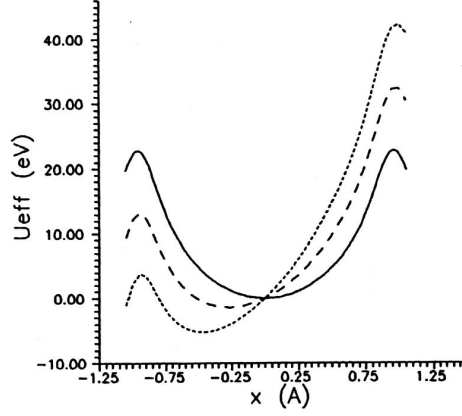


Fig. 3: Effective potential in bent strip silicon crystals [1]. Solid line refers to straight crystals, whereas dashed and dotted lines correspond to centrifugal forces of $\frac{pv}{R} = 1$ and 2 [GeV/cm], respectively.

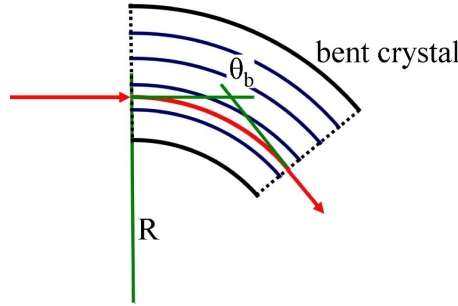


Fig. 4: Deflection given to particles channeled for the whole crystal length.

trifugal force. The dependence of this potential on the particle energy and the crystal bending radius is shown in Fig. 3. From this picture it is easy to infer the presence of a critical bending radius depending on the particle energy, after which the process of planar channeling is no longer possible, because of the insufficient depth of the potential well. The critical radius (R_c) can be understood as the radius for which the centrifugal force is equal to the maximum interplanar field. Thus the critical radius R_c for a given particle energy can be calculated from:

$$R_c = \frac{pv}{U'(x_{max})} \simeq \frac{pv x_{max}}{2U_{max}}, \quad (9)$$

where $U'(x_{max}) \approx 5$ GeV/m in Si crystals [15], and is calculated in $x_{max} = d_p/2 - a_{TF}$ and not in $x = d_p/2$ because of the finite atomic charge distribution.

The new equilibrium point can be derived using the assumptions of equation (4) (but using now x_{max} instead of $d_p/2$) in the equation (8), which gives:

$$U_{eff}(x) = U_{max} \left(\frac{x}{x_{max}} \right)^2 + \frac{pv}{R} x. \quad (10)$$

Thus the minimum of the potential above will be at:

$$x_{min} = -\frac{pv x_{max}^2}{2R U_{max}} = -x_{max} \frac{R_c}{R}, \quad (11)$$

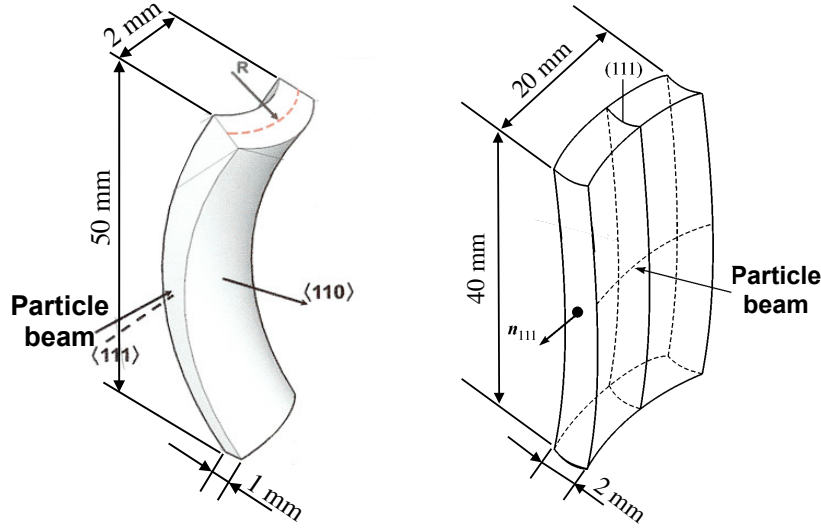


Fig. 5: Geometrical bending of a strip [10] (left) and a quasi-mosaic [16] (right) crystal, together with typical dimensions of crystals used for tests in the SPS.

In this condition, the reduced depth of the potential well¹ can be calculated as:

$$U_{max}^b = U_{eff}(x_{max}) - U_{eff}(x_{min}) = U_{max} \left(1 - \frac{R_c}{R}\right)^2, \quad (12)$$

and the critical angle in a bent crystal is then:

$$\theta_c^b = \theta_c \left(1 - \frac{R_c}{R}\right). \quad (13)$$

The equation of motion for particles channeled between bent crystalline planes can then be expressed as:

$$x = -x_{min} + x_{max} \sqrt{\frac{E_t}{U_{max}^b}} \sin\left(\frac{2\pi z}{\lambda} + \phi\right), \quad (14)$$

which describes a sinusoidal trajectory, as in straight crystals, but oscillating around a new minimum x_{min} , with a different amplitude $\left(x_{max} \sqrt{E_t/U_{max}^b}\right)$, and same oscillation period λ . If the channeling regime is maintained for the whole length of the crystal, the channeled particle is deflected by an angle equal to the geometrical crystal bending:

$$\theta_b = \frac{l}{R}, \quad (15)$$

where l is the crystal length, as shown in Fig. 4. Obviously, this can only be achieved with pure crystals where the crystalline planes maintain a uniform bending along the whole crystal length, which is nowadays ensured by new bending techniques [10, 16]. An example of crystal bending in the case of ST and QM crystals is shown in Fig. 5, left and right respectively.

¹i.e. the maximum transverse energy for which channeling is still possible.

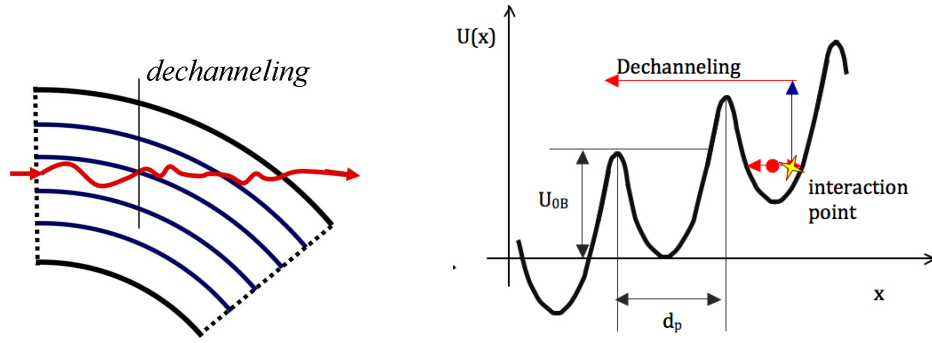


Fig. 6: Illustration of the dechanneling process.

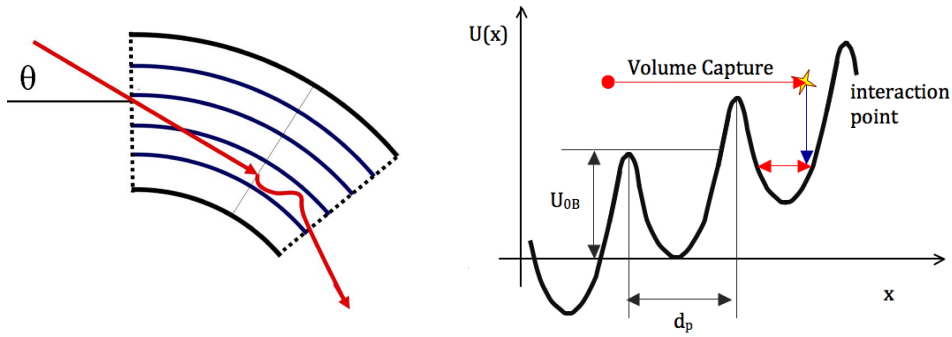


Fig. 7: Illustration of the Volume capture process.

2.3 Dechanneling and Volume Capture

When a particle is channeled between crystalline planes its transverse energy is not conserved, due to the scattering by electrons and nuclei. Channeled particles might therefore vary their transverse energy at each interaction, and can lose the condition for channeling if the interaction results in a total transverse energy above the maximum of the potential well. This condition is referred to as *dechanneling* (DC) and contributes to decrease the initial population of channeled particles, as shown in Fig. 6. On the other hand, a particle can enter the crystal structure with an energy slightly above the potential barrier. If the interaction results in a total transverse energy below the maximum of the potential well, a new energy state compatible with a bounded motion between crystalline planes can be achieved. This process is called *volume capture* (VC) and is shown in Fig. 7. Thus, the DC and VC processes can be considered as mutual to each other.

Let us now focus on the dechanneling process. As described above, channeled particles can increase their transverse momentum because of scattering from electrons in the channel, nuclei in the lattice or possible imperfections. Given the well established manufacturing process, the presence of imperfections can be neglected [11]. As discussed in [1, 12], the DC process can be described as an exponential decay of the initial population of channeled particles, which can be treated therefore as:

$$N(x) = N_0 \exp\left(-\frac{x}{L_D}\right), \quad (16)$$

where N_0 is the initial number of particles that respect the channeling conditions at the crystal entrance, and $N(x)$ is the number of particles still in channeling after a path x in the channel, for a given *char-*

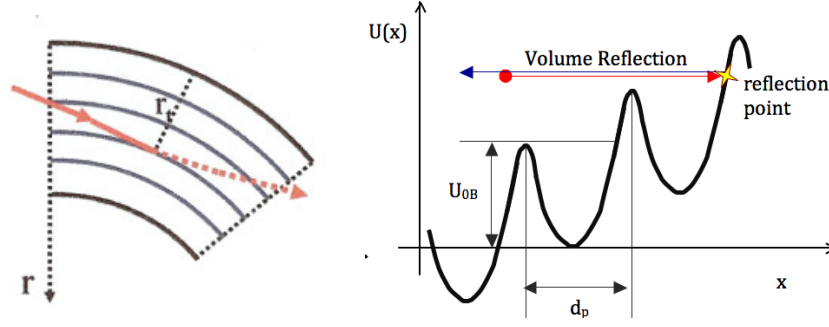


Fig. 8: Illustration of the volume reflection process.

characteristic dechanneling length (L_D). Using diffusion theory it is possible to derive the contribution given by interactions with electrons in the crystalline channel [1], leading to the *characteristic electronic dechanneling length*, which can be written as:

$$L_D^e = \frac{256}{9\pi^2} \frac{pv}{\ln(2m_e c^2 \gamma / I) - 1} \frac{a_{TF} d_p}{Z_i r_e m_e c^2}, \quad (17)$$

where I is the ionization potential ($I \simeq 172$ eV in Si), Z_i the electric charge of the channeled particle with its relativistic γ , while r_e and m_e are, respectively, the classical radius and rest mass of the electron. However, electronic dechanneling only describes a “slow” dechanneling regime, due to the very small variation in momentum from scattering with electrons in the channel, leading to an incomplete treatment of the whole process. Hard scattering on nuclei can lead to “fast” dechanneling as a result of single interactions. Therefore, a *characteristic nuclear dechanneling length* must also be taken into account for a reliable parameterization of the entire dechanneling process. This characteristic length for nuclear dechanneling can be derived by appropriate scaling of the electronic value, based on fine tuning using experimental data, as clarified in Section 3.2.1.

A solid theoretical analysis is needed to describe the probability that an incident particle will be captured in the crystal volume. This can be found in [17], where the dependence of such a probability is derived as a function of the particle energy E and crystal bending radius R , and can be written as:

$$P_{VC} = k \left(\frac{R}{R_c} - 0.7 \right) E^{0.2}, \quad (18)$$

where k is a constant tuned using comparisons between simulations and experimental data. When the particle is captured, it is treated as channeled and the possibility of a subsequent dechanneling interaction should be taken into account, as explained in Section 3.2.1.

2.4 Volume reflection

Particles that impinge on bent crystals with an incident angle in the range $\theta_c < \theta < \theta_b$, where θ_c and θ_b are the critical angle and bending angle of the crystal, can experience what is called *volume reflection* (VR). Protons that undergo this process are literally reflected by the interaction with the averaged potential of crystalline planes. In particular the reflection takes place when particles impinge on a crystalline plane with their momentum tangential to it. The extension of the angular range over which this tangency condition can be reached is determined by geometrical considerations only. It is clear that this condition can be achieved quite easily by particles entering bent crystals² with an angle slightly above the critical

²In direction of the bending.

one (where bounded states are no longer possible). If particles enter a bent crystal with an angle larger than the crystal bending, it will be impossible to reach a condition in which the momentum is parallel to the crystalline planes. An example is shown in Fig. 8.

Another key feature of the reflection process is how much the incident angle is modified, which is linked to the critical channeling angle in straight crystals. From simulation studies [18] it was possible to derive that, for crystals with $R \gg R_c$, the average deflection given to reflected protons is $\sim 1.6\theta_c$. This has been experimentally proven in [19]. The dependence of such deflections as a function of the important crystal parameters was then derived. A good description [17] that fits experimental data for both the average deflection (θ_{VR}) and its spread ($\Delta\theta_{VR}$), is:

$$\theta_{VR} = c_1 \theta_c \left(1 - c_2 \frac{R_c}{R} \right) \quad (19)$$

$$\Delta\theta_{VR} = c_3 \theta_c \frac{R_c}{R} \quad (20)$$

where c_1 , c_2 and c_3 are empirical coefficients tuned to reproduce experimental data, as explained in section 3.2.1.

3 Simulation tools

In large accelerating machines such as the Large Hadron Collider (LHC), the prediction of how particle losses are distributed along the ring is crucial. This is mainly because superconducting magnets are used, and energy deposited in them could cause quenches that must be avoided at any time. A very complex multi-stage collimation system composed of about fifty collimators per beam is used at the LHC. To adequately evaluate losses of the order of 10^{-5} , statistics of 10^6 – 10^7 particles intercepted by the collimation system are needed. For the present system, these particles are tracked typically for 200 turns. It is easy to understand that the results of interactions with obstacles along the particle trajectories are a mandatory feature of any simulation tool that aims to carry out collimation studies. Therefore, it is clear that very fast simulation routines are crucial to describe interactions with collimator jaws, and to obtain results in a reasonable computing time.

In the case of studies on crystal-assisted collimation for the LHC, the requirement to have a very fast routine that describes the interaction of protons with a bent crystal is even more critical. This is because the system is designed to provide beam losses of the order of 10^{-6} , meaning that statistics of more than 10^7 protons intercepted by the crystal are needed. Also note that when a crystal of a few mm of silicon is acting as a scraper (i.e. the crystal is not oriented for optimal steering of halo particles by crystal channeling), particles have to be tracked for about 3000 turns before being absorbed by the collimation chain, mainly due to the small angular spread given by multiple Coulomb scattering. Protons must traverse such a misoriented crystal many times in order to accumulate a kick that is enough to reach the next collimation stage. Therefore, a crystal routine suitable for such complex and demanding simulations has to determine the interaction experienced by any proton based on just a few extractions of random numbers.

The standard tool used at CERN for collimation studies is based on the particle tracking code named SixTrack [3]. It is written in Fortran 77 and was originally developed to study non-linearities and the dynamic aperture in circular accelerators. Over the years it was modified in order to track large numbers of particles, also taking into account interactions with collimator jaws. This led to a collimation version of SixTrack [5, 6, 20], in which a routine that treats the interactions of protons with bent crystals was implemented. This code allow to estimate the density of proton lost per meter on the geometrical machine aperture along the entire ring, with a resolution of 10 cm.

3.1 The scattering routine

When a proton intercepts a collimator jaw along its path, a dedicated routine is called in `SixTrack` to simulate the interactions with matter. A description of the initial routine based on the K2 code is given in [21]. This routine is regularly maintained. A detailed discussion of the physics models implemented, and the latest improvements, can be found in [22] and [23], respectively. This scattering routine is one of the key components for simulations of expected beam loss pattern around the entire accelerator, as a function of collimation settings and machine parameters. The benchmarking of simulated beam loss pattern in the LHC with respect to the measured one is reported in [24]. In this section only a list of the interactions treated is given. The same concepts and models have been implemented in the scattering routine used by the crystal routine described in section 3.2.

The interactions present in the scattering routine can be divided into two families: *nuclear point-like* and *continuous* interactions.

– **Continuous interactions:**

- Multiple Coulomb scattering, where particles are stochastically scattered from components of matter.
- Rutherford scattering, which takes place when particles have a small impact parameter on the material constituent leading to a larger scattering angle compared to Coulomb scattering.
- Energy loss by ionization, i.e. the energy released to atomic electrons along the path inside the material.

– **Nuclear point-like interactions:**

- Deep inelastic scattering, where the incident protons “disappear”. The hadronic shower produced is not simulated and the proton is considered as absorbed in the collimator jaw at that spatial point.
- Nuclear elastic scattering: protons emerge from the interaction with their momentum altered only in direction, but with a value that can be significantly larger than from Coulomb scattering.
- Proton-proton and proton-neutron elastic scattering. The results of these interactions are the same as the previous item, but are in this case a consequence of the interaction with an atomic constituent.
- Single-diffractive events, where protons emerge with their momentum only slightly altered in direction but significantly changed in magnitude.

3.2 The crystal routine

The crystal routine was originally developed as a stand-alone routine, written by Igor Yazynin in Fortran 77. It was then inserted in `SixTrack` by Valentina Previtali as part of her PhD work [25], where details of its implementation and issues of orientation with respect to the beam envelope can be found. The inclusion of possible crystal imperfections was also modeled, such as the presence of an amorphous layer and a miscut angle, further discussed in [25]. The amorphous material is described as a layer that surrounds the crystal bulk, with a thickness given as an input. When protons are incident on this layer, they are treated as traveling in an amorphous material. The miscut angle is modeled as an additional angle applied to protons incident on the crystal front-face, while reduced impact parameters are defined if protons enter the crystal body from the side facing the circulating beam. In this section a detailed description of the physics models present in the routine and their range of validity are discussed. The main body of the crystal routine is composed of three fundamental blocks:

- A routine for the treatment of coherent interactions in bent crystals.

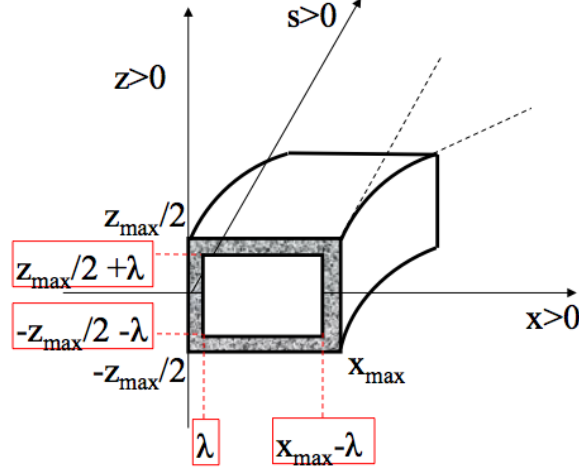


Fig. 9: Crystal routine reference system. The presence of a possible amorphous layer that surrounds the crystal bulk is also shown [25]

- A scattering routine, that treats the contributions of continuous and nuclear point-like interactions (see section 3.1).
- A routine that calculates the energy loss from ionization, described in [8].

3.2.1 Coherent interactions

The processes described in the routine are channeling for the whole crystal length, dechanneling, volume reflection, volume capture and dechanneling after volume capture. Inputs required by the routine, in the crystal reference system shown in Fig. 9, are:

- Crystal length (l) and bending radius (R).
- Momentum (p), and transverse coordinates (x) and (x') of the incident proton.
- The coordinates (z) and (z') are used only to evaluate if a proton is incident on a possible amorphous layer. Crystals are considered flat along this direction, which is an adequate approximation because of the very small dimensions of the beam compared to the crystal size (the usual crystal height is about 5 cm, whereas the beam halo has dimensions $\ll 1$ mm).

Using the quantities above it is possible to evaluate the critical channeling angle in straight (θ_c) and bent (θ_c^b) crystals together with the critical bending radius (R_c), using equations (5), (13) and (9) respectively. Then, the average deflection (θ_{VR}) and spread ($\Delta\theta_{VR}$) given by volume reflection are calculated together with the probability of volume capture (P_{VC}), using equations (19), (20) and (18) respectively. The free parameters used in these last three equations were tuned on experimental data taken in the framework of the H8-RD22 collaboration [19]. For strip crystals they are equal to:

$$c_1 = -1.5, \quad c_2 = 1.6667, \quad c_3 = 1.7, \quad (21)$$

$$k = 7 \cdot 10^{-4}.$$

For quasi-mosaic crystals c_1 is decreased by 7%, while c_3 is increased by 5%.

After calculation of all the parameters introduced above, a first selection based on incident angle is performed. It consists of checking that $\theta_{in} < \theta_c^b$, where θ_{in} is the linear sum of x' including a possible

miscut angle and crystal tilt with respect to the beam envelope. If this condition is satisfied the probability of capture between crystalline planes is calculated as follows:

$$P_{CH} = \frac{\sqrt{\theta_c^b{}^2 - \theta_{in}^2}}{\theta_c} . \quad (22)$$

This formula was derived to reproduce the channeling efficiency obtained with analytical simulation tools for a wide range of crystal parameters, i.e. length and bending, at a fixed energy. Using equation (13), if one assumes that a proton is incident with angle θ_{in} that is a fraction α of the critical value, thus:

$$\theta_{in} = \alpha \theta_c^b = \alpha \theta_c \left(1 - \frac{R_c}{R}\right) , \quad (23)$$

equation (22) can be expanded as follows:

$$P_{CH} = \left(1 - \frac{R_c}{R}\right) \sqrt{1 - \alpha^2} . \quad (24)$$

It is clear that equation (24) gives a $P_{CH} = 1$ in case of an incident beam with uniform divergence and optimal angle (i.e. $\alpha = 0$), on straight crystals (i.e. $R \rightarrow \infty$). This feature is not realistic, and such an approximation would be valid only if the protons were incident in the middle of the interplanar channel with transverse dimensions much less than the channel width. To allow the possibility for some incident protons to strike a crystalline plane, which precludes capture between two planes, equation (24) is replaced with:

$$P_{CH} = \left(1 - \frac{R_c}{R}\right) \sqrt{0.9 - 0.9\alpha^2} . \quad (25)$$

The maximum probability of capture between crystalline planes “saturates” at 95%. This value was adopted to fit experimental data for crystals with very small bending, as discussed in [8]. In particular, the difference between measured and simulated channeling efficiency was improved from about 10% to a few % for such crystals, while simulations of larger bending are not affected.

A random number with a uniform distribution in the range $[0, 1]$ is then generated, and if it is below the probability calculated using equation (25) protons are considered as initially trapped, otherwise they are flagged as being in the transition region between amorphous interactions and the volume reflection process. Then, for protons in this transition region, the deflection given is a linear interpolation between multiple Coulomb scattering (i.e. average deflection zero) and the expected value from the volume reflection process (i.e. θ_{VR} of equation 19). The incoming angle is modified as:

$$x' = x' + 0.45 \left(\frac{x'}{\theta_c^b} + 1 \right) \theta_{VR} . \quad (26)$$

The spatial coordinates are moved to the middle of the path in the crystal volume, where the scattering routine is called to evaluate the contribution of amorphous interactions along the whole path. The five particle coordinates (p, x, x', y, y') are then propagated up to the exit from the crystal volume. The energy loss by ionization is also calculated for the whole path using a dedicated routine. This approach is denoted as the *thin lens approximation*: the full path in the amorphous material is divided into two steps, with the results of the interactions that take place applied at the mid-point and then used to propagate the proton coordinates as it leaves the volume. This approach is valid because the crystal dimensions (a few mm) are small compared to the radiation and interaction lengths (about 9 cm and 46 cm in silicon, respectively).

For protons that are considered trapped between crystalline planes the characteristic electronic dechanneling length is calculated. In bent crystals the equation (17) is modified as [1]:

$$L_D^e = \frac{256}{9\pi^2} \frac{pv}{\ln(2m_e c^2 \gamma/I) - 1} \frac{a_{TF} d_p}{Z_i r_e m_e c^2} \left(1 - \frac{R_c}{R}\right)^2. \quad (27)$$

As explained in section 2.3 a characteristic nuclear dechanneling length is needed for full description of the dechanneling process. Unfortunately an analytical formula is not yet available in the literature. However, a characteristic length can be derived from an opportune scaling of the electronic value, and fine tuning was carried out to evaluate the best scaling constant, as explained in [26]. The characteristic nuclear dechanneling length is calculated in the routine as:

$$L_D^n = \frac{L_D^e}{200}. \quad (28)$$

This length is not calculated for all channeled protons but only a subset of them. The number of such protons is given by the ratio between the interplanar distance and the width of a crystalline plane. The width of a plane can be estimated using the Thomas-Fermi constant a_{TF} , while the interplanar distance for (110) silicon crystal planes is $d_p = 1.92 \text{ \AA}$ [1]. Thus:

$$\frac{a_{TF}}{d_p} = \frac{0.8853 a_B Z^{-1/3}}{1.92} \approx \frac{0.194}{1.92} \approx 0.1, \quad (29)$$

where $a_B = 0.529$ [1]. Thus, the nuclear dechanneling length is applied to only 10% of the protons initially channeled. Since the dechanneling process can be considered as an exponential decay of the initial population of trapped particles as a function of the distance travelled between crystalline planes, the possible point where dechanneling could take place is estimated as:

$$L = -L_D \ln(r), \quad (30)$$

where L_D is the value calculated using either equation (27) or (28), and r is a uniform random number in the range $[0, 1]$. Thus, if L is smaller than the crystal length, the dechanneling process takes place at that depth in the crystal. Hence, protons are transported up to this point as channeled (described below), and then are propagated up to the exit from the crystal as traveling in an amorphous material, using the thin lens approximation. If L is larger than the crystal length, the proton is considered trapped between crystalline planes for the whole path in the crystal. The coordinates of the protons are modified as follows:

$$\begin{aligned} x' &= \frac{l}{R} + \frac{\theta_c^b}{2} r_g, \\ x &= x + l \sin\left(\frac{x'}{2} + m\right), \\ y &= y + l y', \end{aligned} \quad (31)$$

where m is the miscut angle and r_g is a random number with a normalized gaussian distribution. The energy loss from ionization is also calculated, together with the probability to experience nuclear interactions between crystalline planes as described later.

Let us move to protons incident with $\theta_{in} > \theta_c^b$. Initially the possible point where volume reflection could take place (l_{VR}), and its projection along s (s_{VR}), are evaluated as:

$$l_{VR} = R \theta_{in} , \quad (32)$$

$$s_{VR} = \sin \left(\frac{\theta_{in}}{2} + m \right) l_{VR} .$$

The possible reflection point l_{VR} is compared with the crystal length l . If $l_{VR} > l$ the volume reflection process cannot take place, and the proton is transported along its whole path in the crystal as in an amorphous material using the thin lens approximation. In the other case (i.e. $l_{VR} < l$) the probability of volume capture (P_{VC}) is calculated using equation (18). A uniform random number r in the $[0, 1]$ range is generated, and if $r > P_{VC}$ the proton is considered as reflected in the crystal volume. In this case, its coordinates are transported to the reflection point and the deflection given by the volume reflection process is applied. This is computed as:

$$x = x + x' s_{VR} , \quad (33)$$

$$y = y + y' s_{VR} ,$$

$$x' = x' + \theta_{VR} + r_g \Delta \theta_{VR} .$$

The proton is transported along the path in the crystal as in an amorphous material using the thin lens approximation. If $r < P_{VC}$ the volume capture process takes place, and the coordinates (x, y) are transported to s_{VR} and a new dechanneling length is calculated. This length is based on an empirical model that fits experimental data taken in the framework of the H8-RD22 collaboration [27], and is equal to:

$$L_{VC} = L_D^e \left(\sqrt{0.01 - \ln(r)} - 0.1 \right)^2 . \quad (34)$$

Then this length is compared to the remaining path length in the crystal. If it is larger, the proton is considered captured between crystalline planes up to the end of its path, with consistent coordinate modifications. Otherwise, the proton undergoes dechanneling after the capture between planes. Then the proton is transported to the point of capture, then translated to the dechanneling point as for the channeling condition, and finally propagated up to the exit from the crystal using the thin lens approximation. Energy loss from ionization is calculated consistently for each different regime.

3.2.2 Scattering routine

The scattering routine is called when protons treated by the main crystal routine travel in amorphous silicon, as listed in the previous section. This routine includes the same physics models used in the standard routine of SixTrack to treat the interaction with collimator jaws, introduced in section 3.1. Thus, consistent modeling of nuclear point-like and continuous interactions with respect to standard CERN collimation tools is included in the crystal routine, maintaining the stand-alone nature of the routine.

The possibility to experience nuclear point-like interactions for particles trapped between crystalline planes is done by applying the scattering routine also during the path between crystalline planes, but considering only nuclear point-like interactions and using cross-sections scaled to the average nuclear density at the appropriate location. This is done in the assumption that the nuclear density between crystalline planes can be described as [1]:

$$\rho(x) = N_{am} \frac{d_p}{\sqrt{2\pi u_1^2}} \left[\exp \left(-\frac{x^2}{2u_1^2} \right) + \exp \left(-\frac{(x - d_p)^2}{2u_1^2} \right) \right] , \quad (35)$$

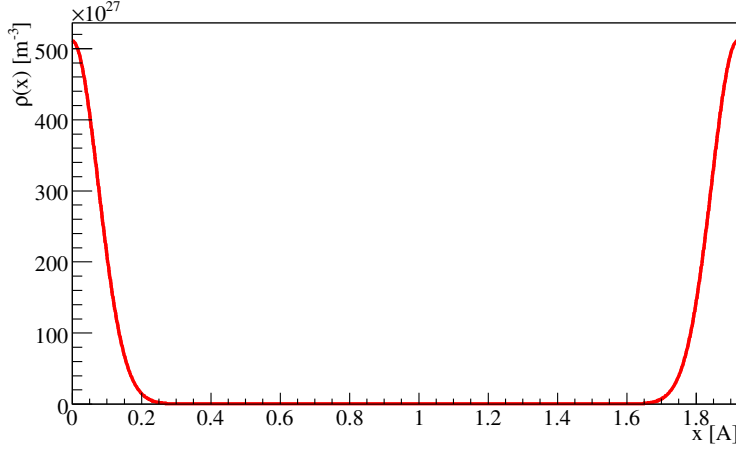


Fig. 10: Nuclear density between (110) planes of silicon, calculated with equation (35).

where x is the transverse distance from the crystalline planes placed at $x = 0$ and $x = d_p$, N_{am} is the density of misoriented silicon, and $u_1 = 0.075$ is the thermal vibration amplitude [1]. The nuclear density obtained using equation (35) is shown in Fig. 10. Using the equation of motion of protons between bent crystalline planes (equation (14)) it is possible to evaluate the maximum (x_M) and minimum (x_m) excursion from the equilibrium point, as:

$$\begin{aligned} x_m &= -\frac{d_p}{2} \frac{R_c}{R} - \frac{d_p}{2} \sqrt{\frac{E_t}{U_{max}^b}}, \\ x_M &= -\frac{d_p}{2} \frac{R_c}{R} + \frac{d_p}{2} \sqrt{\frac{E_t}{U_{max}^b}}. \end{aligned} \quad (36)$$

These two extremes oscillation amplitudes are calculated from the centre of the crystalline channel; a consistent shift of $-d_p/2$ must be applied to both of them in order to use the same reference frame as equation (35). Calculating the integral of equation (35) gives:

$$\int \rho(x) dx = N_{am} \frac{d_p}{2} \left[\operatorname{erf} \left(\frac{x}{\sqrt{2u_1^2}} \right) - \operatorname{erf} \left(\frac{d_p - x}{\sqrt{2u_1^2}} \right) \right]. \quad (37)$$

The average density seen along the trajectory can be calculated analytically, as:

$$\bar{\rho} = \frac{\int_{x_m}^{x_M} \rho(x) dx}{x_M - x_m}. \quad (38)$$

The equation above allows to avoid any numerical solution of complex integrals, maintaining the speed of the routine. What is obtained with this approximation is shown in Fig. 11. The average nuclear density seen by 400 GeV/c protons incident on a silicon strip crystal, 1.94 mm long and with a 10.26 m bending radius, is plotted as a function of transverse energy. The detailed benchmarking with respect to experimental data of nuclear interaction rate in bent crystals is reported in [7,8].

3.2.3 Limitations

It is crucial to have a clear understanding of possible limitations of any simulation routine in order to avoid those “working points”. The weakness of the routine implemented in SixTrack is represented by

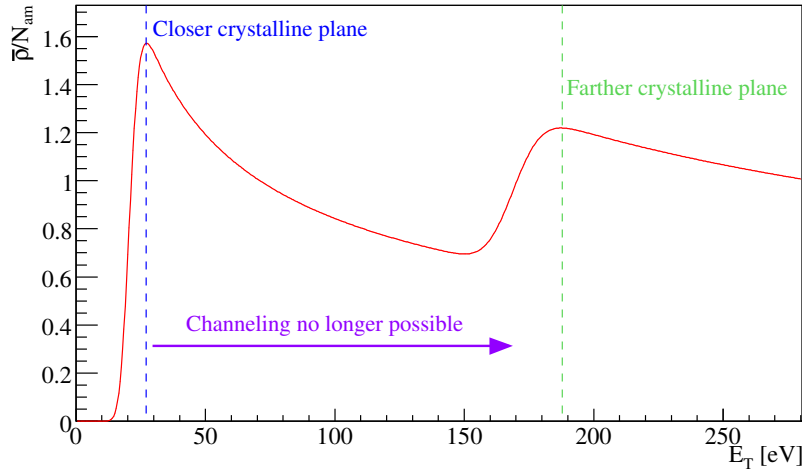


Fig. 11: Average nuclear density as a function of the transverse energy of 400 GeV/c protons, captured between (110) planes of silicon crystals 1.94 mm long and with a 10.26 m bending radius, as calculated using equation (38).

the description of nuclear dechanneling for extremely large bending angles. In these cases the potential well between crystalline planes is almost absent, and the few trapped particles are oscillating very close to the atoms. This condition is reached for a bending radius below three times the critical bending radius. An analytical treatment of such regimes is not yet available in the literature, and it could be reproduced only by simulation codes based on the integration of the equation of motion along the crystalline potential. Thus SixTrack simulations for crystals with $R < 3R_c$ cannot be fully reliable due to underestimated nuclear processes for such bending conditions.

4 Conclusions

The crystal routine and its implementation in SixTrack was described, together with the physics models used and their range of validity. This routine is suited for high statistics tracking simulations for beam loss pattern predictions in large hadron accelerators. The complete benchmarking of the crystal routine and of its implementation in SixTrack are reported in [7, 8]. The tools discussed allowed the design of a crystal collimation test stand in the LHC [8, 28] that led to the first observation of crystal channeling at the record energy of 6.5 TeV [29].

5 Acknowledgments

The authors would like to thank the LHC Collimation team in BE-ABP group, the UA9 Collaboration and the EN-STI group for the useful discussions and support during measurements.

References

- [1] V. Biryukov, Y. Chesnokov, and K. V.I, *Crystal channeling and its application at high energy accelerators*. Springer, 1996.
- [2] W. Scandale, G. Arduini, R. Assmann, F. Cerutti, S. Gilardoni, E. Laface, R. Losito, A. Masi, E. Metral, D. Mirarchi, *et al.*, “Strong reduction of the off-momentum halo in crystal assisted collimation of the SPS beam,” *Physics Letters B*, vol. 714, no. 2, pp. 231–236, 2012.
- [3] F. Schmidt, “SixTrack, User’s Reference Manual,” tech. rep., CERN SL/94-56 (AP), 1994.
- [4] <http://sixtrack.web.cern.ch/SixTrack/>.

- [5] G. Robert-Demolaize, R. Assmann, S. Redaelli, and F. Schmidt, “A new version of SixTrack with collimation and aperture interface,” in *Particle Accelerator Conference, 2005. PAC 2005. Proceedings of the*, pp. 4084–4086, IEEE, 2005.
- [6] <http://lhc-collimation-project.web.cern.ch/lhc-collimation-project/code-tracking-2012.php>.
- [7] D. Mirarchi, G. Hall, S. Redaelli, and W. Scandale, “A crystal routine for collimation studies in circular proton accelerators,” *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 355, pp. 378–382, 2015.
- [8] D. Mirarchi, *Crystal collimation for LHC*. PhD thesis, Imperial Coll., London, CERN-THESIS-2015-099, 2015.
- [9] J. Stark *Phys. Zs.*, vol. 13, p. 973, 1912.
- [10] V. Guidi, L. Lanzoni, and A. Mazzolari, “Study of anticlastic deformation in a silicon crystal for channeling experiments,” *Journal of Applied Physics*, vol. 107, no. 11, p. 113534, 2010.
- [11] S. Baricordi, V. Guidi, A. Mazzolari, D. Vincenzi, and M. Ferroni, “Shaping of silicon crystals for channelling experiments through anisotropic chemical etching,” *Journal of Physics D: Applied Physics*, vol. 41, no. 24, p. 245501, 2008.
- [12] A. Taratin, “Particle channeling in a bent crystal,” *Physics of Particles and Nuclei*, vol. 29, no. 5, pp. 437–462, 1998.
- [13] G. Molière *Z. Naturforsch A*, vol. 2, p. 133, 1947.
- [14] J. Lindhard and K. Dan *Vidensk. Selsk. Mat. Fys. Medd.*, vol. 34, p. 1, 1965.
- [15] E. Tsyganov, “Preprints tm-682,” *TM-684, Fermilab, Batavia*, 1976.
- [16] Y. M. Ivanov, A. Petrunin, and V. V. Skorobogatov, “Observation of the elastic quasi-mosaicity effect in bent silicon single crystals,” *Journal of Experimental and Theoretical Physics Letters*, vol. 81, no. 3, pp. 99–101, 2005.
- [17] Y. A. Chesnokov, V. Maishev, and I. Yazynin, “Volume capture and volume reflection of ultrarelativistic particles in bent single crystals,” *arXiv preprint arXiv:0808.1486*, 2008.
- [18] A. Taratin and W. Scandale, “Volume reflection of high-energy protons in short bent crystals,” *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 262, no. 2, pp. 340–347, 2007.
- [19] W. Scandale, A. Vomiero, S. Baricordi, P. Dalpiaz, M. Fiorini, V. Guidi, A. Mazzolari, R. Milan, G. Della Mea, G. Ambrosi, *et al.*, “Volume reflection dependence of 400 GeV/c protons on the bent crystal curvature,” *Physical review letters*, vol. 101, no. 23, p. 234801, 2008.
- [20] R. Assmann, M. Brugger, M. Hayes, J. Jeanneret, F. Schmidt, I. Baichev, and D. KaItchev, “Tools for predicting cleaning efficiency in the LHC,” in *Particle Accelerator Conference, 2003. PAC 2003. Proceedings of the*, vol. 5, pp. 3494–3496, IEEE, 2003.
- [21] T. Trenkler and J. Jeanneret, “K2, a software package evaluating collimation systems in circular colliders (manual),” tech. rep., CERN-SL-Note-94-105-AP, 1994.
- [22] N. Catalan-Lasheras, *Transverse and Longitudinal Beam Collimation in a High-Energy Proton Collider (LHC)*. PhD thesis, Zaragoza U., 1998.
- [23] C. Tambasco, “An improved scattering routine for collimation tracking studies at LHC,” Master’s thesis, La Sapienza University of Rome, 2013.
- [24] R. Bruce, R. Assmann, V. Boccone, C. Bracco, M. Brugger, M. Cauchi, F. Cerutti, D. Deboy, A. Ferrari, L. Lari, *et al.*, “Simulations and measurements of beam loss patterns at the CERN Large Hadron Collider,” *Physical Review Special Topics-Accelerators and Beams*, vol. 17, no. 8, p. 081004, 2014.
- [25] V. Previtali, *Performance evaluation of a crystal-enhanced collimation system for the LHC*. PhD thesis, EPFL These N.4794, CERN-THESIS-2010-133, 2010.

- [26] D. Mirarchi, S. Redaelli, W. Scandale, A. Taratin, and I. Yazynin, “Improvements of the crystal routine for collimation studies,” in *Proceedings of the International Particle Accelerator Conference*, p. 886, 2014.
- [27] I. Yazynin *private communication*.
- [28] D. Mirarchi, G. Hall, S. Redaelli, and W. Scandale, “Design and implementation of a crystal collimation test stand at the large hadron collider,” *Eur. Phys. J. C*, vol. 77, p. 424, 2017.
- [29] W. Scandale, G. Arduini, M. Butcher, F. Cerutti, M. Garattini, S. Gilardoni, A. Lechner, R. Losito, A. Masi, D. Mirarchi, *et al.*, “Observation of channeling for 6500 GeV/c protons in the crystal assisted collimation setup for LHC,” *Physics Letters B*, vol. 758, pp. 129–133, 2016.

Updated implementation of collimator materials in SixTrack and MERLIN codes*

E. Quaranta[†], A. Valloni, R. Bruce, A. Mereghetti, S. Redaelli
CERN, 1211 Geneva, Switzerland.

Abstract

The High-Luminosity upgrade of the Large Hadron Collider will push various operational parameters of the machine beyond the design values. The LHC beam collimation system turns out to be one of the bottlenecks for the achievement of the future challenging beam parameters. In particular, various limitations have been identified that call for an improvement of the materials used in various collimator types. An R&D program at CERN explores novel composite materials for collimators with improved mechanical robustness and reduced electromagnetic impedance that can address the main limitations of the present system. A new collimator material implementation introduced in SixTrack and MERLIN, simulation codes for beam collimation at the LHC, allows to simulate advanced collimators based on novel composite materials. An immediate application of this novelty in the two codes is to study the effect of the upgrades collimators on the cleaning performance of the collimation system. After presenting the methods used to model composite materials in each code, we apply it to the case of cleaning simulations with novel collimator materials in the LHC, where the results obtained with SixTrack and MERLIN are compared.

Keywords

SixTrack, MERLIN, composite materials, collimation, cleaning.

1 Introduction

The collimation system at the Large Hadron Collider (LHC) is designed to efficiently absorb high energy beam losses and ensure machine protection [1, 2]. Since the first design of the present system [2], non-metallic primary and secondary collimators (TCPs and TCSGs, respectively) turn to provide the largest contribution to the electromagnetic impedance of the whole machine [3]. The impedance is an electromagnetic concern of paramount importance that may obstacle the achievement of the High Luminosity (HL) upgrade of the LHC [4], which aims at smaller beam emittance and doubled beam current. The small collimator gaps required to achieve the design beam performance, as well as the low electrical conductivity of the carbon-fiber-carbon (CFC) composite used for the collimator jaws contribute to the large impedance of secondary collimators.

In order to reduce beam instability in operation caused by the impedance, an R&D program has started at CERN to explore novel composite materials that can replace the present CFC and match the HL-LHC requirements. The developed materials combine the excellent thermal properties of graphite or diamond with those of metals and metal-based ceramics of high mechanical strength and, above all, good electrical conductivity. A number of possible composites have been narrowed down to molybdenum-carbide-graphite (MoGr) and copper-diamond (CuCD) [5], which so far are the most promising candidates for the new generation of advanced collimators.

*Research supported by the High Luminosity LHC project and EuCARD-2 (Grant agreement 312453).

[†]Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milan, Italy.

Material properties are measured through rigorous experimental testing [6]. Then, numerical simulations use the observed properties to investigate the effect of different collimator materials on the cleaning performance of the LHC collimation system. The implementation of new composite materials has been introduced in the existing SixTrack and MERLIN codes, which are the tools routinely used for beam cleaning studies in large circular machines, such as the LHC.

SixTrack [7–9] is a tracking code, written in Fortran 77, and used for simulating longitudinal and transverse dynamics of single beam particles in ultra-relativistic accelerators. The code was extended [10] to evaluate the performance of a collimation system in terms of induced loss pattern and cleaning performance. This setup became the standard tool for collimation studies at CERN. The dynamics of a large number of protons that populate the beam halo can be studied throughout the simulated ring, which is accurately described by a detailed aperture model of the LHC. The interactions of the particles with the collimator materials, which generates different scattering mechanisms with either the atomic shells or the nuclei of the material, are also simulated. In SixTrack, scattering mechanisms are modelled by a Monte-Carlo code that has been recently reviewed and improved [11].

MERLIN [12] is an accelerator physics library originally written in the years 2000 for specific studies on the International Linear Collider [13]. Later, it was decided to update the MERLIN library to include the requirements for complementary simulations of the LHC collimation system. MERLIN is written in C++, which makes it a modular tool and therefore easy to modify. It offers thick lens tracking, an on-line aperture check, and a number of physics processes. The scattering physics in MERLIN has recently been updated [14].

Both SixTrack and MERLIN codes build composite materials for collimator in similar way, but differ in the approach to perform point-like scattering in the composites. In this work, the implementation methods used in the two codes are described and results of the scattering physics for composite materials are compared. Tracking simulations performed with SixTrack and MERLIN are then discussed, for the case of nominal LHC and 7 TeV proton beams, to assess the collimation efficiency when advanced collimators are deployed in the system.

1.1 Mechanisms of particle-matter interaction

Before going into the details of the material implementations adopted in the codes, it can be useful to recall the main scattering processes that a proton can undergo when interacts with matter. SixTrack and MERLIN allow to simulate a set of proton-matter interactions in the collimator jaw, which are classified in:

- *Continuous interactions*, i.e. ionization, multiple Coulomb scattering and Rutherford scattering.
- *Point-like nuclear interactions*, i.e. nuclear elastic, inelastic and inelastic-diffractive scattering.

1.1.1 Ionization

When a charged particle collides against the atomic electrons along the path of the traversed material, the energy transferred from the incoming particle to the electrons can be high enough to put them in motion, and then ionising the material. The energy loss along the path, commonly called stopping power, is expressed by the Bethe-Bloch equation [15]:

$$-\frac{dE}{dx} = K z^2 \frac{Z}{A} \frac{1}{\beta_{rel}^2} \left[\frac{1}{2} \ln \frac{2m_e c^2 \beta_{rel}^2 \gamma^2 T_{max}}{I^2} - \beta_{rel}^2 - \frac{\delta}{2} \right] \quad (1)$$

where Z and A are respectively the atomic number and the atomic mass of the material, m_e is the electron mass while z , β_{rel} and γ are respectively the charge, the velocity and the relativistic factor of the incident particle. I is the mean excitation energy, T_{max} is the maximum kinetic energy that an electron can gain in one single collision and finally δ is a correction term depending on the density of the material. K/A is

a constant, equal to $(4\pi N_A r_e^2 m_e c^2)/A = 0.307075 \text{ MeV g}^{-1} \text{ cm}^2$ for $A = 1 \text{ g mol}^{-1}$. The energy loss per unit length (in $\text{MeV g}^{-1} \text{ cm}^2$) given as a function of the particle energy in any material varies very slowly for $E > 100 \text{ TeV}$ [16]. In SixTrack, dE/dx at the reference energy of 450 GeV/c was initially used to estimate the energy lost by ionization in both the injection and the top energy case. However, to more accurately estimate dE/dx , the revised version of the scattering routine [11] now computes Eq. 1 for protons of any given energy. This calculation requires information on atomic parameters, such as Z , A , I , as well as on the material density, needed for the conversion to metric units (GeV/m).

1.1.2 Multiple Coulomb Scattering (MCS)

When a charged particle traverses a medium, it faces many small angle deflections, most of them caused by Coulomb scattering from the nuclei, so the origin of its name. According to Moliere's theory [17], the rms of the deflection angle θ after crossing a thickness s of material is given by:

$$\theta(s) = \frac{13.6 \text{ MeV}}{\beta_{rel} c p} z \sqrt{\frac{s}{X_0}} \left[1 + 0.038 \ln\left(\frac{s}{X_0}\right) \right] \quad (2)$$

where p is the momentum of the incident particle. X_0 is the radiation length of the material and indicates the mean length of the medium that the particle has to traversed to reduce its energy by $1/e$.

1.1.3 Rutherford scattering

This mechanism takes place when the incoming proton passes very close to a nucleus and is deflected away with a large scattering angle by the repulsive force due to the positive charge of the nucleus. It is still considered as Coulomb scattering, but with a finite cross section defined as:

$$\frac{d\sigma_{ruth}}{dt} = 4\pi\alpha^2 (\hbar c)^2 \left(\frac{Z}{t}\right)^2 e^{-0.856 \times 10^3 t R^2} \quad (3)$$

where $a = 1/137$, $(\hbar c)^2 = 0.389 \text{ GeV}^2 \text{ mbarn}$ and Z is the atomic number. The factor proportional to $1/t^2$ is the classical Rutherford formula for Coulomb scattering near a point charge. The exponential term is the nuclear form factor that accounts for the finite size of the nucleus. R is the radius of the nucleus that, if not measured, is calculated by $R = 2\hbar c \sqrt{b}$. b is the slope factor of the angular distribution for proton-nucleus elastic interaction, which is correlated to the mass number of the material by the formula $b = 14.1 A^{2/3}$.

1.1.4 Point-like interactions

Unlike the previous mechanisms that occur over a certain length of the material, proton-matter interaction can also involve an incoming proton and one of the components of the atomic structure of the traversed material (like nucleons, i.e. protons or neutrons, or the entire nucleus). The proton-nucleon interactions can be elastic, inelastic or inelastic-diffractive.

In *nuclear elastic* scattering, the beam particle survives to the event it underwent, changing direction and energy according to the two-body kinematics. With a look at the LHC, nuclear elastic scattering, together with ionisation and MCS, plays a major role in diluting the beam particles and reducing their energy when they survive to the impact onto a collimator.

In *nuclear inelastic* scattering the incoming particle do not come out from the material (and therefore considered lost there), and generate new (secondary) particles. These events are extremely important in the design of beam intercepting devices, as they provide an estimation of the beam attenuation.

Moreover, *single-diffractive* (SD) scattering also exists. The momentum transferred during the collision can excite a nucleon of the medium, that have enough energy to travel some distance through the medium. However, the probability that the incoming proton survives without being lost is null.

From here the reason why SD scattering is considered a quasi-elastic mechanism. This process is very important for LHC collimation because part of the betatron halo is converted into an off-momentum halo, due to the larger variation in energy of the particles after the event.

1.1.5 Particle mean free path

The *mean free path* is the length before a point-like interaction or a Rutherford scattering event occurs between the incoming particle and the atoms of the medium travelled. The mean free path λ_i of each process i can be expressed in terms of the cross section of the interaction with the atoms of the medium through the formula [18]:

$$\lambda_i = \frac{1}{\sigma_i} \frac{A}{\rho N_A} \quad (4)$$

where σ_i is the cross section of the process, ρ and A are respectively the material density and the atomic weight, and N_A is the Avogadro number. A *total collision length* is also defined as:

$$\frac{1}{\lambda_{tot}} = \sigma_{tot} = \sigma_{el} + \sigma_{SD} + \sigma_{inel} + \sigma_{ruth} \quad (5)$$

that takes into account all the possible elastic, quasi-elastic and inelastic collisions, as well as the Rutherford scattering events, that can occur along the path. In the LHC, the *inelastic collision length* λ_{inel} is particularly important for the collimator design, as it expresses the average length needed to achieve an attenuation of $1/e$ of the primary particle, by nuclear inelastic events.

2 Implementation of new materials in SixTrack

A dedicated database embedded in SixTrack, called SCDATA, stores the material properties required by the scattering routine to simulate the different scattering processes. Some of the input parameters are:

- Atomic and mass number,
- Material density,
- Radiation length,
- Mean excitation energy,
- Cross sections of the point-like interactions,
- Rutherford scattering cross section.

So far, only pure materials were implemented in the code: aluminium, beryllium, carbon, copper, lead, and tungsten. The values of the material properties were directly taken from Ref. [16] and used in the formulas discussed below to calculate the parameters required by the code, such as cross sections and nuclear interaction lengths.

Composite materials, some already used in the present LHC collimators and some being considered for the HL-LHC collimation upgrades, have been added to the existing database. The newly implemented materials are:

- Molybdenum-Graphite composite (MoGr, grade MG-6530Aa),
- Copper-Diamond composite (CuCD),
- Tungsten Heavy Alloy Inermet-180 (IT-180),
- Glidcop.

The material routine in SixTrack treats mono-element materials. In the presented approximation, composite materials are dealt with by calculating off-line the input parameters required by the code, starting from the material composition. The result is a “composit” nucleus weighted on the constituent

nuclei. The approximation considers an “averaged” composite material, therefore not described at the atomic level.

Here we discuss how the values of each parameter required by the SixTrack material database are obtained for composite materials. The atomic number Z and atomic weight A of each compound material was calculated as average weighted on the atomic fraction of their components i :

$$p = \sum_i at_i \cdot p_i, \quad (i = 1, 2, \dots) \quad (6)$$

where p is the property of the compound to be computed, p_i are the values of the property, extracted from the Particle Data Group database [16], for the i -th element present in the material and at_i the atomic content of each element in the composite.

The mean ionization energy I and the radiation length χ_0 must be provided to SixTrack to calculate the energy lost by ionization and correctly simulate the MCS events. According to [16], for a composite material, these parameters can be approximated by:

$$\frac{1}{p} = \sum_i \frac{wt_i}{p_i}, \quad (7)$$

where wt_i is the mass fraction of the i -th element in the compound. Once more, p_i refer to values in [16].

The incoming protons can interact with the nuclei of the material in different ways. Each process is characterized by its cross section (σ), that gives the probability for the scattering process to occur in a specific material. The total nuclear cross section (σ_{tot}) and the nuclear inelastic cross sections (σ_{inel}) at 450 GeV were implemented in SixTrack, calculated as follows:

$$\sigma = \frac{A}{N_A \rho \lambda}, \quad (8)$$

with N_A Avogadro’s number, A and ρ the average atomic weight and the density of the material, respectively. For composite materials, the collision length (λ_{tot}) and the inelastic length (λ_{inel}) are calculated as in Eq. (7). The Rutherford scattering cross section σ_{ruth} was derived through Eq. 3. In particular, the radius R and the slope factor b were computed by using the average mass weight for the “composite” nucleus, calculated as in Eq. 6.

Table 1 lists the most important parameters, along with the atomic composition, of the materials considered in the implementation. The properties of CFC are also reported in the table for comparison, although the material was already coded in SixTrack as pure carbon.

Table 1: Summary of the properties of the new composite materials added in SixTrack: average atomic number Z , average atomic weight A , density ρ , electrical conductivity σ_{el} , composition by atomic fractions, radiation length χ_0 , collision length λ_{tot} and inelastic scattering length λ_{inel} for 200 GeV/c protons.

	Z	A (g/mol)	ρ (g/cm ³)	σ_{el} (MS/m)	at. content (%)	χ_0 (cm)	λ_{tot} (cm)	λ_{inel} (cm)
CFC	6	12.01	1.67	0.14	100 C	25.57	35.45	51.38
MoGR	6.65	13.53	2.50	1	2.7 Mo ₂ C, 97.3 C	11.93	24.84	36.42
CuCD	11.90	25.24	5.40	12.6	25.7 Cu, 73.3 CD, 1 B	3.16	13.56	20.97
Glidcop	28.82	63.15	8.93	53.8	99.1 Cu, 0.9 Al ₂ O ₃	1.44	9.42	15.36
IT-180	67.66	166.68	18	8.6	86.1 W, 9.9 Ni, 4 Cu	0.39	6.03	10.44

This implementation of composite materials in SixTrack is recently become part of the official release of the code [19].

3 Implementation of new materials in MERLIN

In order to model composite materials for advanced collimators in MERLIN, a new `COMPOSITEMATERIAL` class was created. The `COMPOSITEMATERIAL` class allows the user to create a composite material from other existing or user defined materials. These constituent materials may be pure elements or other composites and may be added by fractions of mass (m_i) or number (n_i) using the functions *AddMaterialByMassFraction* or *AddMaterialByNumberFraction*. After all constituents have been added, the *Assemble* function calculates all material properties. An example of constructing CuCD using the `COMPOSITEMATERIAL` constructor, setting functions, and calculation functions is shown in Fig. 1. In order

```

1 CompositeMaterial* CuCD = new CompositeMaterial();
2 CuCD->SetName("CopperCarbonDiamond");
3 CuCD->SetSymbol("CuCD");
4
5 CuCD->AddMaterialByMassFraction(CD,0.3489);
6 CuCD->AddMaterialByMassFraction(Cu,0.6467);
7 CuCD->AddMaterialByMassFraction(B,0.0044);
8
9 CuCD->SetDensity(5400);
10 CuCD->SetConductivity(12.6E6);
11 CuCD->Assemble();
12 CuCD->VerifyMaterial();

```

Fig. 1: Example of constructing CuCD in MERLIN.

to construct a composite materials from the constituent elements, the mass fraction m_i of constituent i in the composite must be known, which is given by:

$$m_i = \frac{n_i A_i}{\sum_i n_i A_i}, \quad (9)$$

where A_i is the constituent atomic mass and n_i is the number fraction of the constituent in the composite. These fractions may be used to define some composite material properties as a weighted average of the constituent properties. For example the mean atomic mass of the homogeneous composite \bar{A} is:

$$\bar{A} = \sum_i n_i \cdot A_i. \quad (10)$$

and the radiation length χ_0 of a compound material is:

$$\frac{1}{\chi_0} = \sum_i \frac{m_i}{\chi_i} \quad (11)$$

where χ_i is the radiation length of the i th element. In order to define the path length for a proton scattering in the composite in MERLIN, *i.e.* the distance traversed before interacting with a nucleus, the sum of the calculated cross sections $\sigma_{pN \text{ tot}}$ is used to find the mean free path:

$$\lambda_{tot} = \frac{\bar{A}}{\sigma_{pN\ tot} \cdot \bar{\rho} N_a}, \quad (12)$$

where N_a is Avogadro's constant and ρ the density.

Cross sections for composite nuclear interactions are generated in MERLIN for the composite (as a homogeneous mixture) in order to calculate the mean free path, but are not used for point-like scattering. Working from the constituent cross sections, Eq. 13 is used to find the reference nuclear cross sections (total, inelastic, elastic, or Rutherford).

$$\sigma_{pN} = \sum_i n_i \sigma_{pN\ i}. \quad (13)$$

These properties are used in the application of bulk scattering in MERLIN, *i.e.* MCS and energy loss via ionisation for a composite material.

4 Comparison of composite material implementation in SixTrack and MERLIN

SixTrack and MERLIN use different approaches to treat point-like interactions between the particles of the incoming beam and the collimator material. In SixTrack (we call it “6T-method”), point-like scattering are performed from an imaginary composite nucleus, which is the weighted average of the constituent nuclei. In MERLIN, when a point-like events occurs, an elementary constituent nucleus of the material is randomly selected and the proton scatters from the constituent nucleus with corresponding cross sections and all other properties. In this paper, we refer to this approach as “MERLIN-method”. However, MERLIN can also reproduce a similar approach to that of SixTrack by enabling point-like scattering from the composite nucleus instead of the constituent nuclei. We call the latter “MERLIN-6T method”. The different approaches have been compared in order to quantify the errors [20].

To better study the effect on the scattering physics of the different approaches used in the two codes on the scattering physics, a test case was defined. It consists a pencil beam of 6.4×10^6 protons impacting upon 1 cm long block of material. Simulations have been performed in MERLIN and SixTrack, and the results are shown in Fig. 2, 3, 4. In each plot, the red curve represents simulations performed in MERLIN using the “MERLIN-method”, the blue curve the “MERLIN-6T-method” and the green curve the results obtained from SixTrack. Fig. 2 shows the change in polar angle θ and particle momentum dp for particles that undergo single diffractive (SD) events in the material block (either MoGr or CuCD). The “MERLIN-method” results in a smaller cross section than SixTrack. It is also the case that the momentum transfer is smaller in SixTrack. The distribution of θ and the spread in energy loss are larger in MERLIN, while SD events resulting in small energy loss are more frequent in SixTrack. Differences between the “MERLIN-method” and “MERLIN-6T-method” are not evident in these plots. The angular and momentum spread for particles that do not undergo point like scattering events are compared in Fig. 3: the trends appear identical. MERLIN and SixTrack provide very similar results, which is expected as for MCS and ionisation the method used for composites is the same in both codes. A comparison of the distributions for total elastic scattering, as simulated by the two codes, is shown in Fig. 4, for MoGr and CuCD: no relevant discrepancies between the results are found.

The comparison of the simulated distribution of point-like interactions have shown only small differences between SixTrack and MERLIN, in spite of the different approaches of the two codes in treating composite materials. For the purpose of beam dynamics studies for LHC collimators, we consider these differences small enough. We remind, indeed, that collimators are bulk devices, whose final properties are averaged over the full size of the collimator (several centimetres). Therefore, we are confident that the approximation done in the “SixTrack-method” are consistent and can reproduce the properties of composite materials for collimators without large errors.

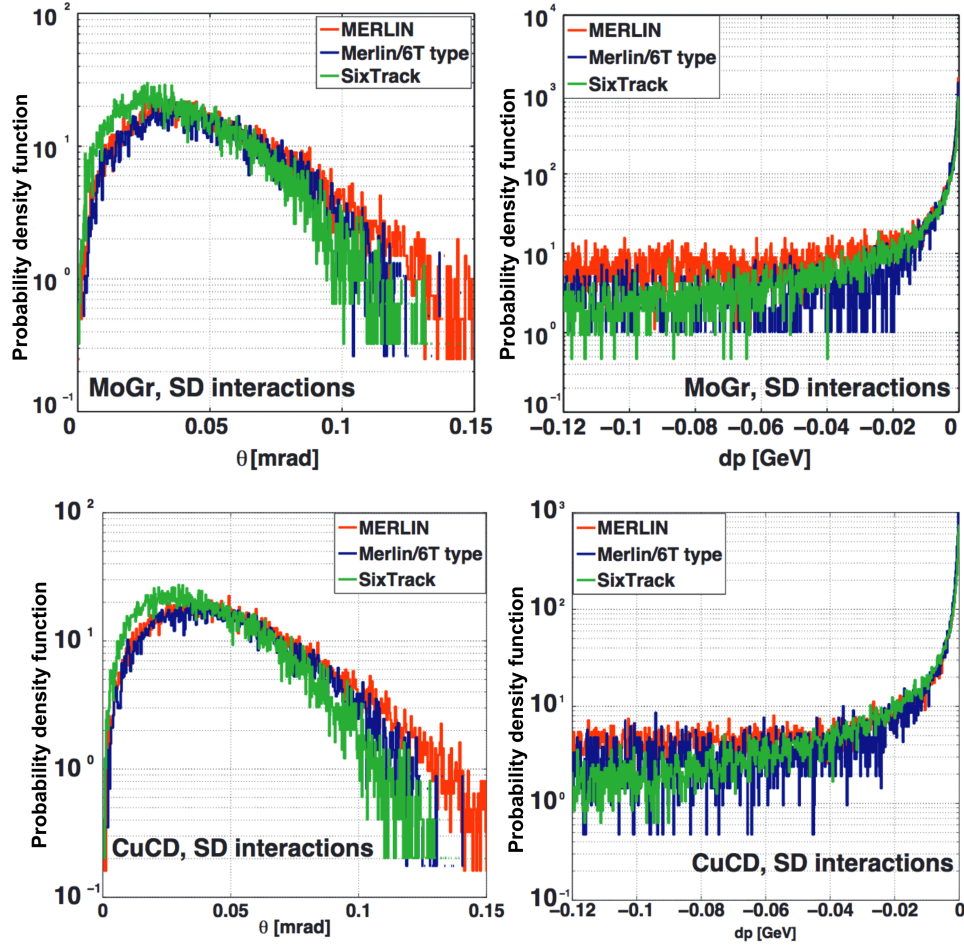


Fig. 2: Distribution of polar angle θ (left) and energy loss dp (right) for single diffractive events in MoGr (top) and CuCD (bottom) block. Comparison of simulation results using “MERLIN-method”, “MERLIN-6T-method” and SixTrack.

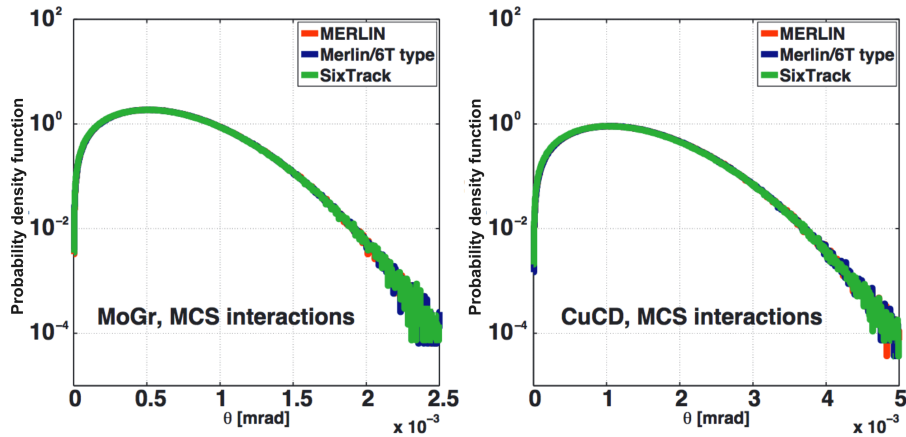


Fig. 3: Distribution of polar angle θ for multiple Coulomb scattering and ionisation events in MoGr (left) and CuCD (right) block. Comparison of simulation results using “MERLIN-method”, “MERLIN-6T-method”.

5 Evaluation of cleaning performance with advanced collimator materials

The LHC collimation system has to fulfil a number of different functions, such as the beam cleaning functionality [1,21]: the system must efficiently intercept the unavoidable losses due to the continuous

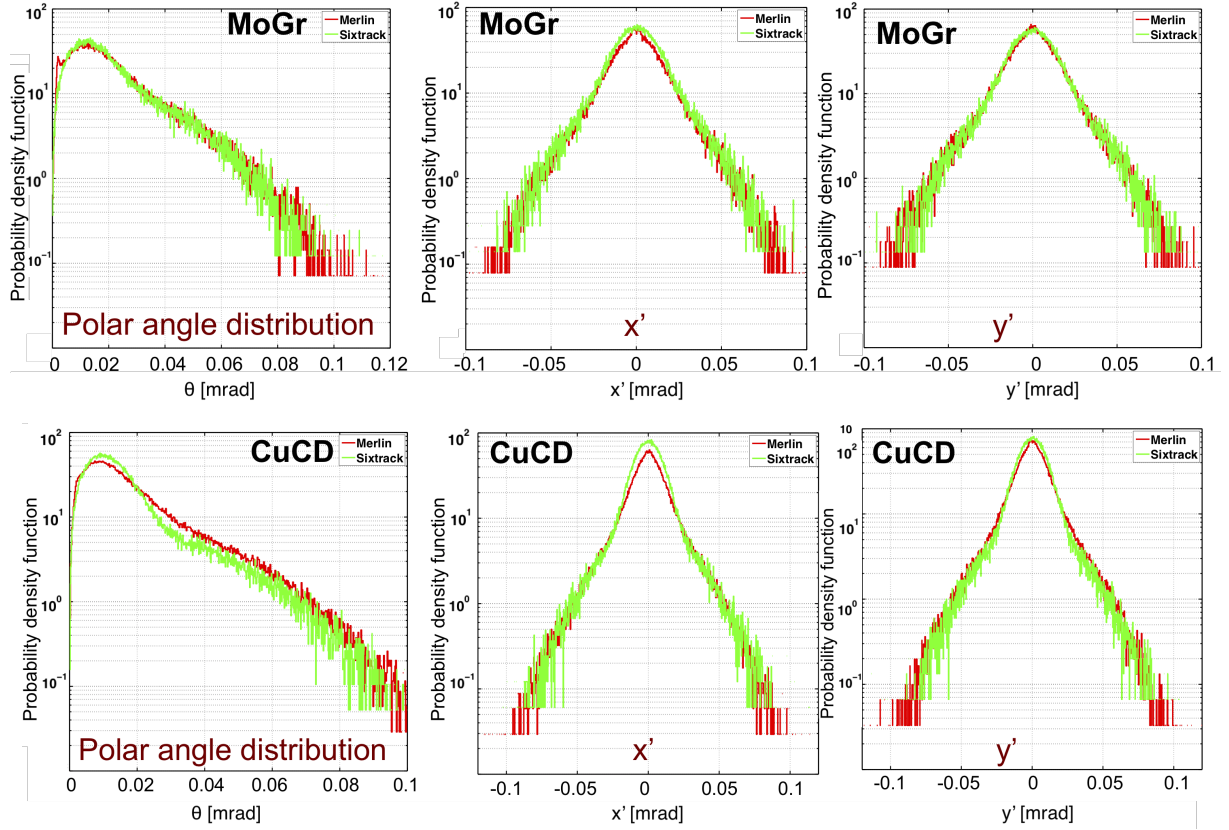


Fig. 4: Polar angle distribution (left) and distributions in x' (center) and y' (right) after 1 cm of CuCD (bottom) and in 1 cm of MoGr (up) of total elastic scattering, comparing MERLIN and SixTrack.

repopulation of the beam halo. The *Local Cleaning Inefficiency* η is used to know the distribution of the losses along the ring and it is expressed by:

$$\eta = \frac{N_{loss}}{\Delta s \cdot N_{abs}} \quad (14)$$

where N_{loss} is the number of particles lost in a Δs length and N_{abs} is the total number of particles absorbed in the collimation system.

In this paper, the effect of new collimator materials on the cleaning efficiency of the LHC collimation system was studied in SixTrack and MERLIN. Tracking simulations were performed for the two beams and in the two planes of halo distribution. The case of the nominal LHC machine at 7 TeV with optics squeezed to $\beta^* = 55$ cm was simulated¹. In simulations, the full collimation system was in place and the collimator openings were set according to Table 2: a 1σ retraction between primary and secondary collimators in the betatron collimation region IR7 was used, where σ is the local betatron beam size at the collimator assuming a nominal normalized emittance of $3.5 \mu\text{m}$ and the design β -function. Same optics parameters, collimator setting and particle statistics were used in both codes to allow a direct comparison of the results.

As of major concern for impedance, IR7 secondary collimators (TCSG) made of CFC were replaced in simulations with new collimator jaws in either MoGr or CuCD. The material implementation described in Section 2 was used in SixTrack, while the equivalent “MERLIN-6T-method” was adopted in MERLIN. In simulations, three cases were considered:

¹The machine optics is typically adjusted to have a local minimum at the interaction points (experiments), in order to minimize the beam size and thus maximise the interaction rate. The value of the beta function at the interaction points is therefore called β^* .

Table 2: Collimator settings used for in simulation. The values are expressed in units of standard deviation of the beam, calculated for a normalized emittance of $3.5 \mu\text{m rad}$. Note that TCL (Target Collimator Long) are located downstream of the secondary collimators (TCSG) in the collimation insertions IR3 and IR7, TCDQ (Target Collimator Dump Quadruple) protect the beam dump line in IR6, and TCT (Target Collimator Tertiary) provide local protection around the interaction points (IP1/2/5/8).

Region	Collimator Families	Settings [σ]
IR7	TCP / TCSG / TCLA	6 / 7 / 10
IR3	TCP / TCSG / TCLA	15 / 18 / 20
IR6	TCSG / TCDQ	7.5 / 8
IP1/5	TCTs	8.3
IP2/8	TCTs	25

Case 1 (reference) : all IR7 secondary collimators in CFC,

Case 2 : all IR7 secondary collimators in MoGr,

Case 3 : all IR7 secondary collimators in CuCD.

Results obtained with both simulation tools indicate that the losses in the cold dispersion suppressors magnets downstream of IR7, which represent the location of highest losses along the ring for the present system and therefore the main limitation to the cleaning efficiency, are essentially not affected by changing the material of the TCSGs [20, 22]. This result was expected since these losses are dominated by single diffractive events taking place in the upstream primary collimators [23] that are unchanged in these studies. As an example, Fig. 5 shows loss patterns in terms of cleaning inefficiency η simulated by SixTrack in the full LHC and around the IR7 region, compared in the case of CFC (Case 1) and CuCD (Case 3) secondary collimators.

We focus the following analysis on the loss distribution on the IR7 TCSGs, comparing the results obtained by the two codes. The losses on all secondary collimators, expressed in terms of number of the inelastic interactions occurring in the jaw volume, collimator by collimator, are depicted in Fig. 6 for the Case 1-2-3: on the left hand side, the results from SixTrack simulations, and on the right hand side those from MERLIN. Both codes show that the first two TCSG downstream of the primary collimators are progressively more loaded as the effective Z of the composites increases. These collimators intercept the products of the scattering with the primary collimators, and a larger Z values have a direct impact on the effectiveness of the particle absorption. The loss ratio calculated collimator by collimator for the novel composite materials over the present CFC (Fig. 7) shows a loss increase of about 10% for MoGr and 13-18% for CuCD. Differences in losses in the TCSGs further downstream are less apparent and, if any, indicate lower load for the new materials than for the present CFC. This is mainly caused by a larger fraction of secondary halo that is already absorbed by the first two TCSG.

The distribution of the particles lost along the length of the two jaws of the most loaded secondary collimator (TCSG.B5L7) is plotted in Fig. 8: as expected, it shows an exponential decrease due to inelastic scattering events along the jaw length and the slope looks steeper for CuCD and MoGr, materials with higher density and therefore shorter inelastic length, than the case of CFC.

Finally, it is important to stress that the number of destructive interactions of primary protons is not the dominant factor to understand beam losses on a collimator. The energy lost by ionization during the trajectory in the material and the generation of secondary particle shower from nuclear inelastic interactions contribute for the largest part to the total energy deposited in the collimator. Therefore, results of tracking simulations must be complemented by energy deposition and thermo-mechanical studies to have a more complete picture of the effect of beam losses on a beam-intercepting device. Simulations of energy density and mechanical response on new secondary collimators are being performed to confirm that the structural integrity of the composite material jaws will not be compromised by the increased

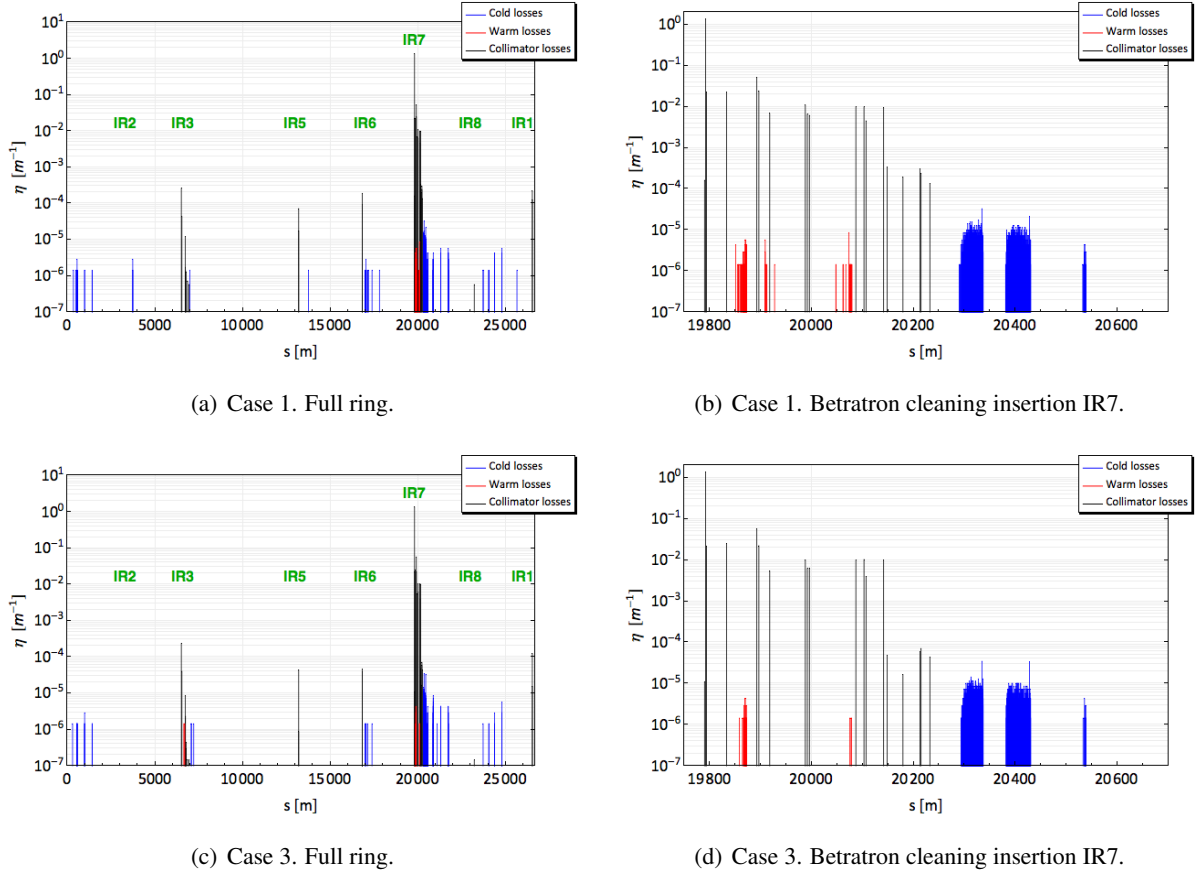


Fig. 5: Beam loss patterns at 7 TeV in the nominal LHC machine for Beam 1 horizontal halo, simulated by SixTrack. In blue, the losses in the superconductive magnets (cold losses), in red those in the normal-conductive magnets (warm losses) and in black the losses in the collimators.

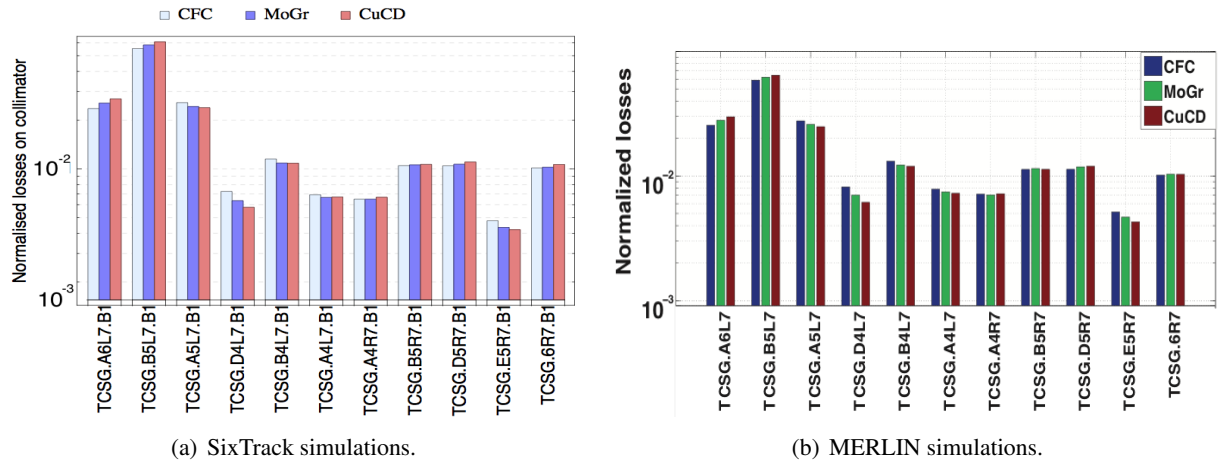


Fig. 6: Normalized losses on secondary collimators in IR 7 for different jaw materials, i.e. CFC, MoGr, CuCD, as estimated by SixTrack (left) and MERLIN (right).

energy deposited when denser materials than CFC are used.

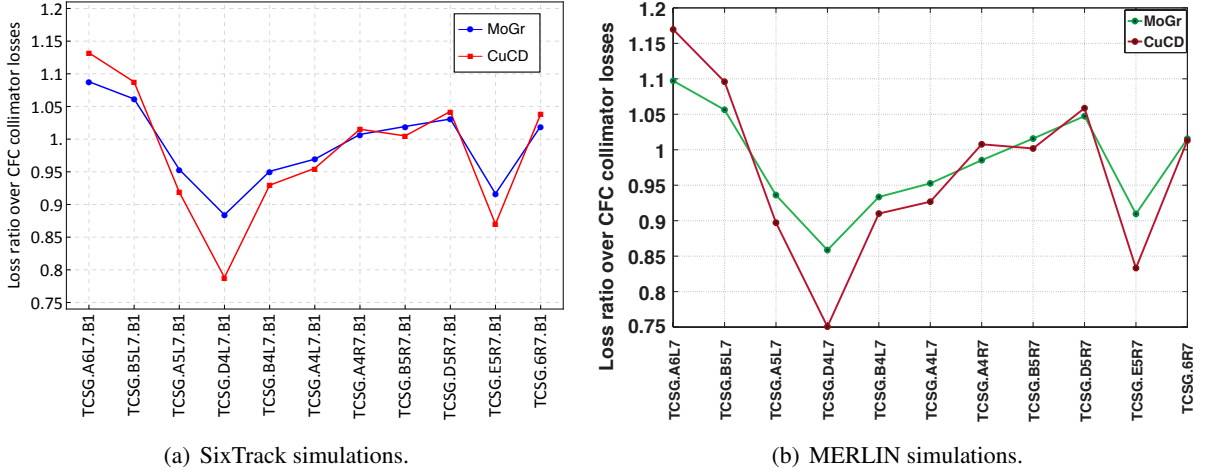


Fig. 7: Ratio between simulated losses on secondary collimators in IR7 for different jaw materials over the CFC ones, as estimated by SixTrack (left) and MERLIN (right).

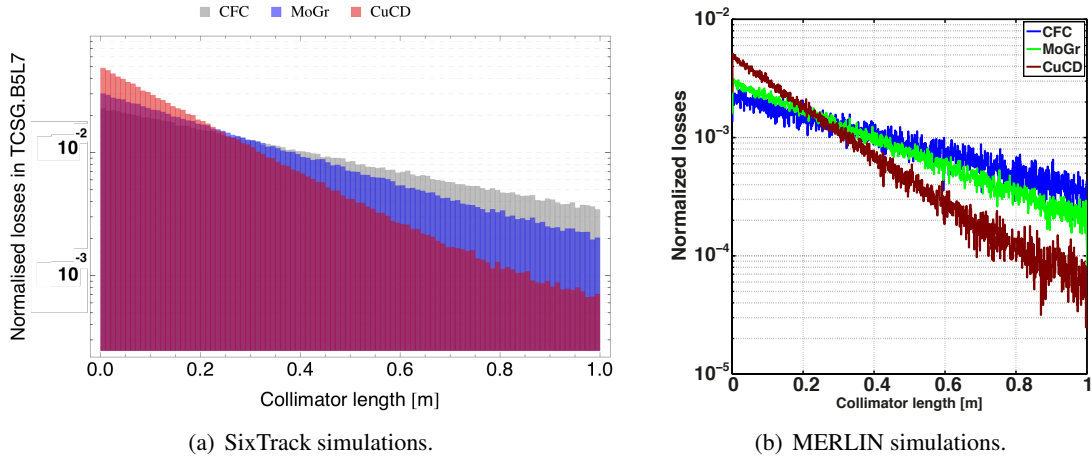


Fig. 8: Distribution of particles lost along the length of the most loaded TCSG for different jaw materials, as estimated by SixTrack (left) and MERLIN (right).

6 Conclusions and outlook

The need to reduce the impedance contribution of the LHC collimation system for future upgrades of the LHC motivated the development of novel composites that combine high-robustness and reduced electrical resistivity. The possibility to simulate proton interactions with composite materials has been successfully implemented in SixTrack and MERLIN, and is now available for simulations of beam collimation cleaning in the LHC. The SixTrack implementation relies on the calculation of effective material properties, such as cross sections of nuclear events and radiation length, as for a “composite” nucleus. The same approach was tested in MERLIN, which treats composite materials in a different way. From a first comparison of the results obtained with the codes, no relevant differences between the two approaches have been observed. This is an important validation of the newly proposed approach. However, further comparison with other tools, e.g. FLUKA [24,25] and GEANT [26], are planned to be performed in the future.

In this work, we also presented results of halo cleaning studies in the LHC with new collimator materials. Simulations were performed in the same conditions with SixTrack and MERLIN to allow a consistent comparison of the results. Advanced collimators, based on novel low-resistivity composite

materials, were used in simulation to replace present non-metallic secondary collimators. Simulation with both codes confirmed that the cleaning efficiency is only slightly improved when new secondary collimators are used. The outcome was not unexpected, as the main goal of this part of the upgrade was indeed the reduction of the electrical impedance. However, a major improvement of the cleaning performance can be achieved if primary collimators are replaced by a denser jaw material, as discussed in Ref. [27,28], or losses in the dispersion suppressor are locally cleaned [28–30]. The results of cleaning simulations with the full collimation layout are overall consistent with the expectations and this enforces further the validity of the new implementation for the treatment of composite materials.

References

- [1] O. Bruning *et al.* (eds), LHC Design Report, Technical report, CERN, Geneva, Switzerland, Rep. CERN-2004-003-V-1, 2004.
- [2] R.W. Assmann *et al.*, The final collimation system for the LHC, in: Proceedings of EPAC06, Edinburgh, Ireland, 2006.
- [3] N. Mounet *et al.*, Collimator impedance measurements in the LHC, in: Proceedings of IPAC13, Shanghai, China, 2013.
- [4] I. Bejar Alonso *et al.*, HiLumi LHC Technical Design Report, Technical report, CERN-ACC-2015-0140, 2015.
- [5] A. Bertarelli *et al.*, Novel materials for collimators at LHC and its upgrades, in: Proceedings of HB2014, East Lansing, Michigan, 2014.
- [6] N. Mariani, Development of novel advanced Molybdenum-based composites for high energy physics applications, Ph.D. thesis, Politecnico di Milano, 2014.
- [7] F. Schmidt, SIXTRACK version 1.2: user’s reference manual, *CERN/SL/94-56-AP*, 1994.
- [8] Sixtrack, <http://sixtrack.web.cern.ch/SixTrack/>.
- [9] R. De Maria *et al.*, Recent Developments and Future Plans for SixTrack, in: Proceedings of IPAC13, Shanghai, China, 2013.
- [10] G. Robert-Demolaize *et al.*, A new version of sixtrack with collimation and aperture interface, *PAC05, Knoxville, USA*, 2005.
- [11] C. Tambasco, An improved scattering routine for collimation tracking studies at LHC, Master’s thesis, Università La Sapienza, Roma, Italy, 2014.
- [12] Merlin, <https://github.com/MERLIN-Collaboration/MERLIN>.
- [13] Linear Collider Collaboration, International Linear Collider, <https://www.linearcollider.org/ILC>.
- [14] J. Molson, Proton scattering and collimation for the LHC and LHC luminosity upgrade., Ph.D. thesis, University of Manchester, 2014.
- [15] K. Nakamura *et al.*, Passage of particles through matter, *JPG 37, 075021*, 2010.
- [16] K.A. Olive *et al.*, Particle Data Group, 2014.
- [17] H. A. Bethe, Moliere’s theory of multiple scattering, *Physical Review*, 89, pages 1256–1266, 1953.
- [18] A.N. Kalinovskii *et al.*, Passage of High Energy Particles through Matter, AIP-Press, 1898.
- [19] A. Mereghetti *et al.*, SixTrack for Clening Studies: 2017 Update, in: IPAC17, Copenhagen, Denmark, 2017.
- [20] A. Valloni *et al.*, MERLIN cleaning studies with advanced collimator materials for HL-LHC, in: Proceedings of IPAC16, Busan, South Korea, 2016.
- [21] S. Redaelli, R.W.Assmann, G. Robert-Demolaize, LHC aperture and commissioning of the collimation system, pages 268–277, 2005.
- [22] E. Quaranta *et al.*, Collimation cleaning at the LHC with advanced secondary collimator materials,

- in: Proceedings of IPAC15, Richmond, Virginia, 2015.
- [23] C. Bracco, Commissioning Scenarios and Tests for the LHC Collimation System, Ph.D. thesis, EPFL Lausanne, 2008.
 - [24] A. Ferrari *et al.*, FLUKA : A multi-particle transport code (program version 2005), *CERN-2005-010*, 2005.
 - [25] T.T. Bohlen *et al.*, The FLUKA code: Developments and challenges for high energy and medical applications, *Nuclear Data Sheets*, (120), pages 211–214, 2014.
 - [26] Geant4, <https://geant4.web.cern.ch/geant4>.
 - [27] A. Valloni *et al.*, Comparison between different composite material implementations in Merlin, Presentation at the 71st Collimation Upgrade Specification Meeting, CERN, May 23rd, 2016.
 - [28] E. Quaranta, Investigation of collimator materials for the High Luminosity Large Hadron Collider, Ph.D. thesis, Politecnico di Milano, Italy, 2017 (to be published).
 - [29] E. Quaranta *et al.*, Towards optimum material choices for HL-LHC collimation upgrade, in: Proceedings of IPAC16, Busan, South Korea, 2016.
 - [30] D. Mirarchi *et al.*, Cleaning performance of the collimation system of the High Luminosity Large Hadron Collider, in: Proceedings of IPAC16, Busan, South Korea, 2016.

Dynamic simulations in SixTrack

K. Sjobak, V.K. Berglyd Olsen, R. De Maria, M. Fitterer, A. Santamaría García, H. Garcia-Morales, A. Mereghetti, J.F. Wagner, S.J. Wretborn
CERN, Geneva, Switzerland

Abstract

The DYNK module allows element settings in SixTrack to be changed on a turn-by-turn basis. This document contains a technical description of the DYNK module in SixTrack. It is mainly intended for a developer or advanced user who wants to modify the DYNK module, for example by adding more functions that can be used to calculate new element settings, or to add support for new elements that can be used with DYNK.

Keywords

DYNK; SixTrack; Particle Tracking; Dynamic Kicks; fast failures; ripple.

1 Introduction

The goal of the DYNK module in SixTrack [1–4] is to make it possible to change element settings on a turn-by-turn basis. This feature was first implemented for simulating the action of scrapers in the SPS [5], and then re-implemented in a more general form as described in [6]. After this re-implementation, the module has been and is being used for simulations of crab cavity failures [7–10], asynchronous beam dumps [11], off-momentum collimation studies through RF detuning [12,13], studies of observed particle losses at the start of the energy ramp in the LHC [14,15], and electron lens intensity modulation [16].

The use of the module is described in the SixTrack user manual [17]. For the most up-to-date version of the user manual, please see the source distribution of SixTrack [18]. This document describes the technical “inner workings” of the DYNK module, and is intended for those who want to extend the module with new mathematical functions or to handle new types of elements.

DYNK is designed to be extensible, and has already been extended several times by several people. It is the hope of the authors that this document will be helpful for those who wish to do this in the future. Note that it is strongly recommended to read about the usage of DYNK in the most recent version of the SixTrack user manual before reading this document.

2 Overview

The main code of DYNK is located in the Fortran module `dynk` found in the `dynk.s90` file. The `.s90` file format is free form Fortran (`.f90`) which will be pre-processed by ASTUCE¹. All the data associated with DYNK is contained within the `dynk` module as saved arrays and variables.

Additionally, the subroutine `initialize_element` was added to the `daten` block. It is responsible for general initialization of SINGLE ELEMENTs at the end of the `daten` subroutine, whether DYNK is in use or not. It is also used to re-initialize such elements if they are later changed by DYNK.

The general program flow is illustrated in Figures 1 and 2. These show how DYNK is initialized, and how it is called. The structure is only shown for 6D thin tracking, however the pattern is the same for the other options.

¹ASTUCE is a source code preprocessor that is used for SixTrack. It assembles the Fortran source files from one or more decks, typically containing a group of related subroutines; and blocks, small pieces of codes that are in-lined one or more places in one or more decks, typically used for defining Fortran COMMON blocks and kicks. Furthermore, the ASTUCE “pragmas” can include conditional statements, used to include or exclude parts of the source code based on compilation flags.

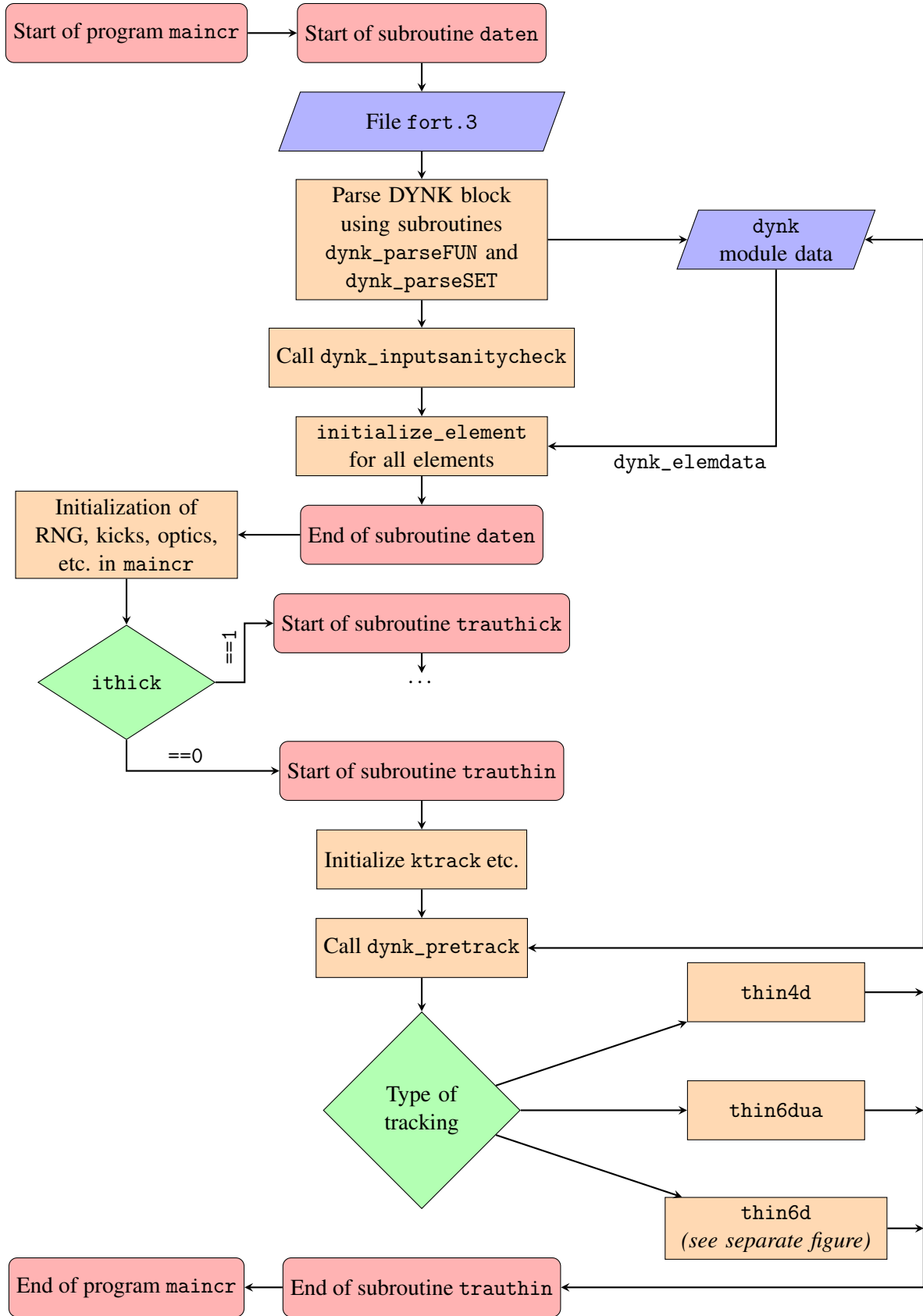


Fig. 1: Program flow in SixTrack with DYNK. The details of thin6d is shown in Figure 2. The beginning/end of processes are in red blocks, orange blocks are actions of interest, blue blocks are I/O, and green blocks are decisions.

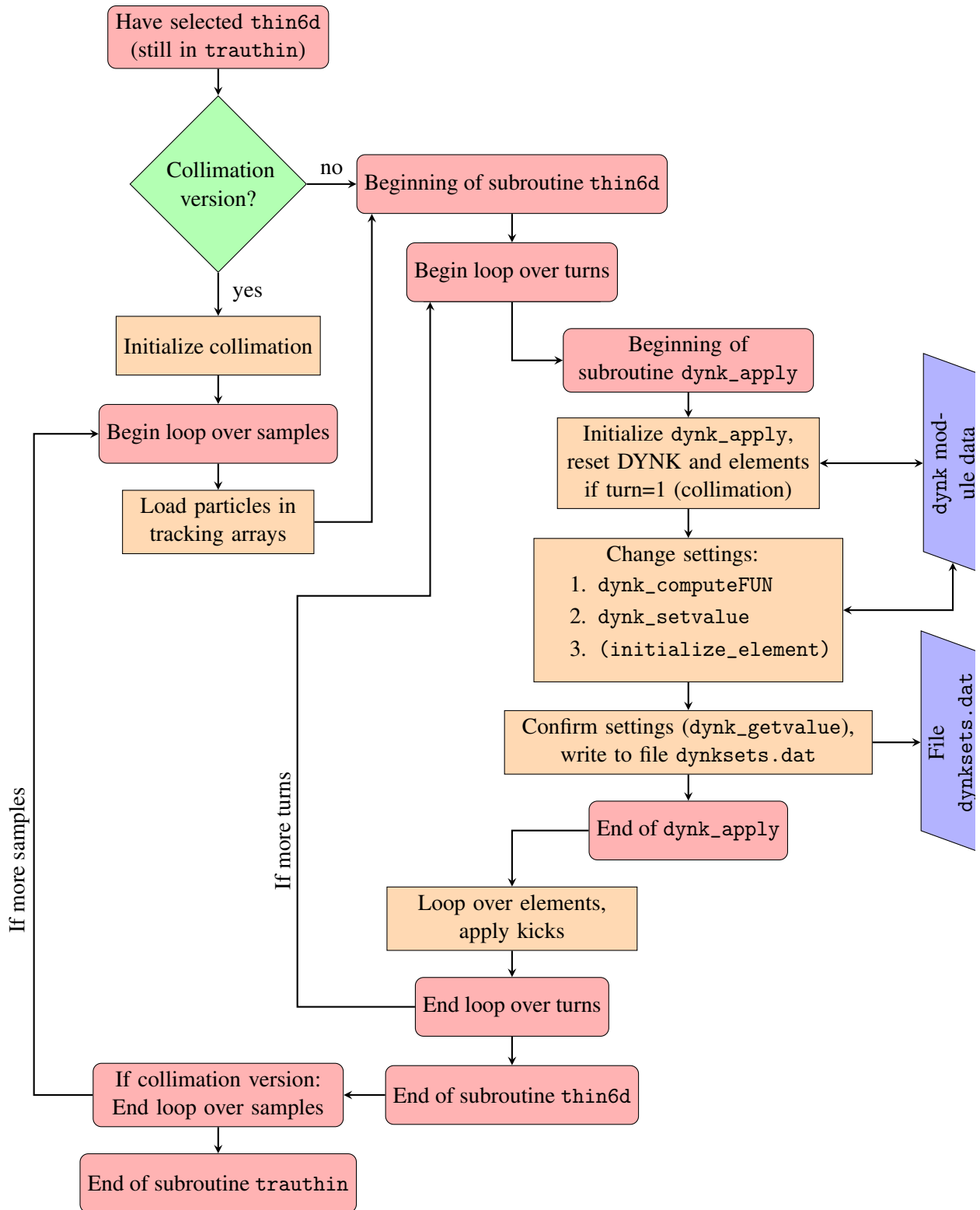


Fig. 2: Details of program flow when running subroutine thin6d; see Figure 1 for the rest of the program. The beginning/end of processes are in red blocks, orange blocks are actions of interest, blue blocks are I/O, and green blocks are decisions.

Note that all DYNK-related global variables have names postfixed with `_dynk`, while all functions and subroutines related to DYNK bear the prefix `dynk_`. Since the subroutine `initialize_element` is also used outside of DYNK, it does not contain this pre- or post-fix in its name. Furthermore, while this document is intended to give an overview of how DYNK functions, detailed descriptions of each subroutine, code stub, and variable are given in the code as comments.

3 Data structure

The configuration of DYNK is stored in the `dynk` module. These variables and arrays are separated in 3 main categories, as described in the following subsections.

3.1 Overall configuration

Whether DYNK is active at all, i.e. if a DYNK block is present in the `fort.3` input file, is controlled by the `ldynk` boolean variable. If this is `.FALSE.`, no DYNK functions are called in the tracking loop. Furthermore, it is possible to enable extra debugging output and run-time checks (`ldynkdebug`), and to disable writing of the `dynksets.dat` output file (`ldynkfiledisable`).

3.2 Functions

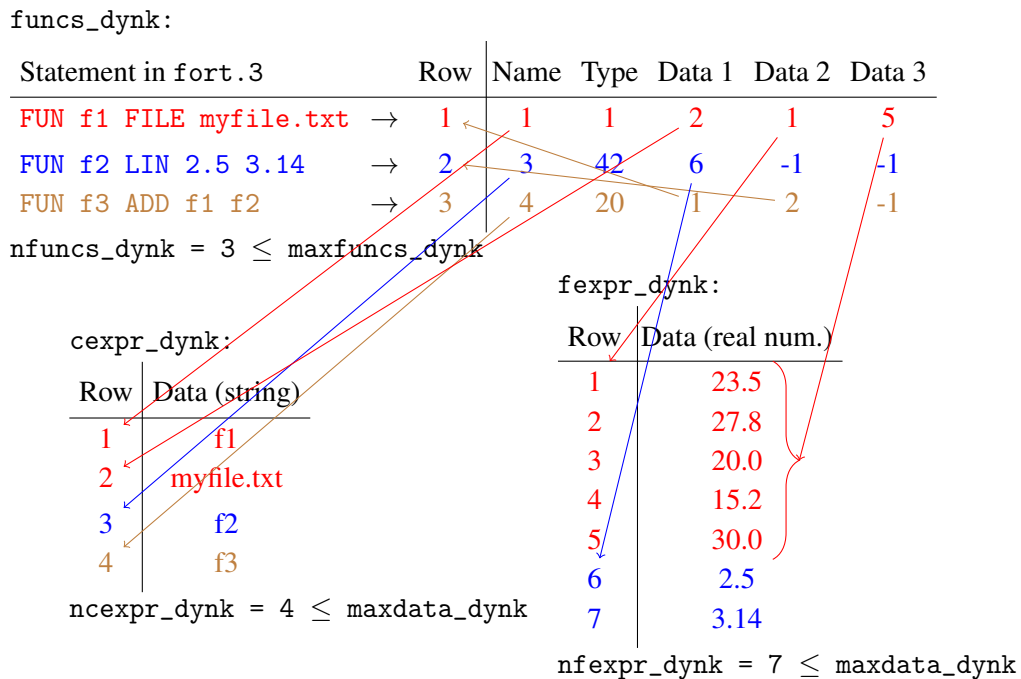


Fig. 3: Illustration of how three DYNK functions are stored in memory. The colours are used to separate the three functions used in the example (**f1**, **f2**, **f3**). The function **f1** is of type FILE, where the column Data1 points to where in `fexpr_dynk` the data begins, and Data2 indicates how much data there is. The function **f2** is of type LIN, where Data1 points to where in `fexpr_dynk` the two parameters are stored. The function **f3** is of type ADD, where Data1 and Data2 indicates which functions should be added by listing their row numbers in `funcs_dynk`.

At the core of DYNK is the possibility to compute function values. This can for instance be used to change the strength of elements as a function of the turn number. The details of how this works is described in Section 5. As illustrated in Figure 3, the configuration for these functions is described in the table `funcs_dynk`, with one row per function (FUN statements in `fort.3`). Each of these rows consist of five integers, for which the first two have the same meaning for all function types. The meaning of the

last three columns vary. Here, the first column always points to the location in the `cexpr_dynk` array where the (user-defined) name of the function is stored, while the second is an index which indicates the type of the function, and thus the interpretation of the following three columns. The number of function “slots” currently in use is defined by the variable `nfuncs_dynk` such that the last row that is currently in use is row number `nfuncs_dynk`². The size of the table, and thus the maximum amount of slots, is given by the constant parameter `maxfuncs_dynk`.

Since many functions require the storage of more than three integers, DYNK has an internal memory allocation mechanism supporting the three main data types: integers, double precision real numbers, and strings. This is implemented as three large arrays `iexpr_dynk`, `fexpr_dynk`, and `cexpr_dynk`. Each string has a maximum length `maxstrlen_dynk`, and the stored strings are generally expected to be terminated and padded with binary zeros. Similar to `nfuncs_dynk`, each of these arrays have an associated integer that keeps track of how many elements are currently in use. These usage counters are called `niexpr_dynk`, `nfexpr_dynk`, and `ncexpr_dynk`, and should point to the last valid index in their corresponding data array.

It is the developer’s responsibility to make sure that a new function (1) does not interfere with the data storage for other functions, and (2) does not exceed the available storage for this data type. For the first point, this means that the function must not modify the data stored by other functions, and it must correctly update the usage counters so that the next function to be defined can safely allocate more memory. In order to help with the second point, the subroutine `dynk_checkspace` is provided, which dynamically expands the arrays when needed. The expansion is normally done in chunks of 500 numbers or 200 strings. If more than this is requested, the allocation is exactly large enough to fit the requested amount.

3.3 Element settings

Statement in <code>fort.3</code>		Row	sets_dynk:				csets_dynk:	
			FUN idx.	First	Last	Shift	Element	Attribute
SET magnet1 average_ms f3 1 5 0	→	1	3	1	5	0	magnet1	average_ms
SET magnet1 average_ms f3 6 -1 -5	→	2	3	6	-1	-5	magnet1	average_ms
SET crabcc1 voltage f2 3 10 -3	→	3	2	3	10	-3	crabcc1	voltage

`nsets_dynk = 3 ≤ maxsets_dynk`

sets_unique_dynk:		
Row	Element	Attribute
1	magnet1	average_ms
2	crabcc1	voltage

`nsets_unique_dynk = 2 ≤ maxsets_dynk`

Fig. 4: Illustration of how three SET statements are stored in memory.

Similar to the functions, the element settings (SET statements in `fort.3`) are also stored in a few tables. However, since all settings are defined by the same fields, the memory management is considerably simpler. As illustrated in Figure 4 the main table is `sets_dynk`, which can store up to `maxsets_dynk` rows of integers. Each row of this table contains four columns. The first column is pointing to a row in the `funcs_dynk` array, indicating which function should be used for this element

²Note the Fortran convention of counting from 1.

setting. The two next columns indicate the range of turns for which the setting is active. The last column is an offset which is applied to the current turn number before computing the value of the function.

Additionally, there is a table `csets_dynk`, which has one row per SET with the same indexing as `sets_dynk`). Each row has two columns, both strings. The first column is the name of the SINGLE ELEMENT to be changed, and the second column is the name of the attribute that should be changed. This table is filled in the subroutine `dynk_pretrack`.

This architecture allows multiple functions to be defined for the same element, as long as the range of turns they are used for is not overlapping. This is verified by the `dynk_pretrack` subroutine, which also verifies that all SETs refer to valid elements and attributes. While this data structure makes it possible to apply different functions to the same element and attribute for different time periods, in many cases it is necessary to directly iterate over the element and attribute combinations that are used instead of the individual SET statements. To avoid excessive searching when iterating over unique elements and attributes, the table `csets_unique_dynk` is used. The table has the same format as `csets_dynk`, but has no duplicate entries. The number of rows used in this table is kept in the integer variable `nsets_unique_dynk`. Furthermore, the array `fsets_origvalue_dynk` contains the pre-changed value of the elements using the same indexing as `csets_unique_dynk`. This is necessary for collimation, where multiple samples of particles are tracked sequentially, making it necessary to be able to reset the state of the elements to how it was originally defined in `fort.2` and `fort.3`.

Finally, there are two arrays that are used to keep additional information on the level of structure elements, of which there may be several per single element. These arrays are named `dynk_izuIndex` and `dynk_elemdata`. The array `dynk_izuIndex` is used to store the index of the random number used to set the magnet error of the element in question. The array `dynk_elemdata` is used to store the various values needed to initialize certain types of elements, such as accelerating RF cavities, and will most likely be necessary for handling magnetic multipoles should this be implemented in the future.

4 Changing element settings: `dynk_apply`

If DYNK is active in the simulation, the subroutine `dynk_apply` is called in the tracking loop at the beginning of every turn, as shown in Figure 2. This subroutine then loops over the defined SETs, checking if it is active in the current turn. If this is the case, the current effective turn is calculated by adding the turn-shift (which is often 0) to the actual turn. The value of the active function is then calculated using `dynk_computeFUN` as described in Section 5.

Note that `dynk_apply` has an important role in initializing DYNK FUNCTIONS where the output depends on the internal state created by previous calls. This happens with the FIR and IIR filters, and also with the RNG-based functions `RANDG`, `RANDU`, and `RANDON`. This is done on the first turn, before any kicks are applied. The action taken is for example to initialize the “seed” that is updated at every call to the RNG function, with the initial seed read from `fort.3`.

4.1 The subroutine `dynk_setvalue`

When the new value has been calculated, it is applied to the specified element and attribute using `dynk_setvalue`. This function modifies the main element setting arrays in the same way as `daten` does when reading the input files, doing no calculations. In practice, this means that when changing single elements, it will write to the `ed/ek/el` and `elens_theta_max` arrays. The reason for this is to make the code, and its testing, as straight forward as possible.

One exception to the “no calculations” rule is that when changing the reference energy `E0` using `GLOBAL-VARS`, which triggers a recalculation of the reference momentum `e0f`, relativistic gamma `gammarr`, and the energy-dependent particle arrays `dpsv/dpsv1/dpd/dpsq/oidpsv/rvv`.

4.2 The subroutine `initialize_element`

If any further calculation is needed, this is done afterwards in the subroutine `initialize_element`. This is for example used for nonlinear elements, where the actual kick strength is given by an average strength (`ed`) and a random component (`ek`). Furthermore, a unit conversion is also applied.

Another example is crab cavities, which store the phase offset in `e1` when reading the input file, which is normally used to store the element length. This phase is therefore moved into a separate array `crabph`, which has the same shape and indexing as `e1`.

Note that `initialize_element` is not only used when changing element settings, but also used just after reading the input files in `daten`. In this case, the second argument “`lfirst`” is set to `.TRUE.`

4.3 The subroutine `dynk_getvalue`

The function `dynk_getvalue` returns the setting of an element’s attribute as it was set by `dynk_setvalue`. This is used for output to `dynksets.dat`. When `ldynkdebug=.TRUE.` it is also used to confirm that the setting was correctly applied.

4.4 The virtual element GLOBAL-VARS

In order to change certain global settings, such as the reference energy `E0`, the element name `GLOBAL-VARS` is used. Because of this, it is not possible to have an actual element in the single elements list with this name when `DYNK` is active.

The `GLOBAL-VARS` element is also treated specially in the `dynk_setvalue` and `dynk_getvalue` subroutines. Here, this “element name” is tested for and handled near the top of the routine, before trying to locate the normal elements.

5 How the functions are calculated

As shown in Table 1, many types of functions are available in `DYNK`. The data used for their evaluation is stored in the `dynk` module, as described in Section 3.2. Internally, each function type is given an integer index, which is used in a `select case` when evaluating the function. Note that these indices are grouped by function type, and this grouping should be respected when adding new functions. A detailed description of the use of each function is given in the user manual, and how they store their data is documented in comments in the `dynk_parseFUN` code.

In most cases the `DYNK FUNs` are functions of the current turn number, including a possible turn-shift, which may be specified in the `SET` command. Note that `DYNK` functions can be chained such that one function can call another one, which may call yet another, etc. This is done by for example the `ADD` function, which calls two other functions, adds their results together, and returns the value. When this happens, the current turn number – including any turn-shift – is passed on to the next function.

5.1 Calculating the function values `f(turn)`: `dynk_computeFUN`

The values of the functions are calculated in this Fortran function. It is called with an index into the relevant row in the `funcs_dynk` table, and the current (possibly shifted) turn number. Using the row index, it then reads the type index (second column), and executes the case containing the code for that function. The code may then read or modify any data in the `funcs_dynk` array and the `{c|f|i}expr_dynk` arrays. Note that it is expected that functions should only touch data in the arrays they themselves allocated in `dynk_parseFUN`, i.e. the functions should not interact outside of calling each other.

Index	Name	Short description
"System" functions:		
0	GET	Get original value of element/attribute.
1	FILE	Load from file (turn-by-turn).
2	FILELIN	Load from file (interpolate non-specified turns).
3	PIPE	Get setting from external program.
6	RANDG	Gaussian RNG.
7	RANDU	Uniform RNG.
8	RANDON	Random 1 or 0 with given probability.
Filters:		
10	FIR	Finite Impulse Response filter.
11	IIR	Infinite Impulse Response filter.
Operators (2-operand):		
20	ADD	Add the results of two other functions.
21	SUB	Subtract the results of two other functions.
22	MUL	Multiply the results of two other functions.
23	DIV	Divide the results of one function by the result of another.
24	POW	Exponentiate the result of one function with the result of another.
Operators (1-operand):		
30	MINUS	Result of another function, with opposite sign.
31	SQRT	Square root of the result from another function.
32	SIN	Sine of the result from another function.
33	COS	Cosine of the result from another function.
34	LOG	Natural logarithm of the result from another function.
35	LOG10	Base-10 logarithm of the result from another function.
36	EXP	Natural exponential function of the result from another function.
Polynomial functions:		
40	CONST	Constant value.
41	TURN	Current turn (after turn-shift).
42	LIN	Linear function of the turn number.
43	LINSEG	Linear function of the turn number (alternative input format).
44	QUAD	Quadratic function of the turn number.
45	QUADSEG	Quadratic function of the turn number (alternative input format).
Transcendental functions:		
60	SINF	Sine of the turn number, with specified ω , ϕ , and A .
61	COSF	Cosine of the turn number, with specified ω , ϕ , and A .
62	COSF_RIPP	Cosine of the turn number (alternate "RIPP" format).
Specialized functions:		
80	PELP	Parabolic-Exponential-Linear-Parabolic, used for energy ramping [20].
81	ONOFF	On for p_1 turns, then off for $p_2 - p_1$ turns, then repeat.

Table 1: Functions defined in DYNK and their indexes. For a full description, please see the user manual.

5.2 Initializing the functions: `dynk_parseFUN`

This function is responsible for parsing the user input (FUN statements), filling the `funcs_dynk` table, and allocating memory in the `{c|f|i}expr_dynk` arrays. It is one of the most complex parts of the DYNK module, and must be modified whenever one wants to add new DYNK functions. In `dynk_parseFUN`, each function type is initialized by a block of code. The selection of the code to execute depends on the function type listed in the second word of the relevant FUN statement. Note that all DYNK functions first call two support subroutines `dynk_checkargs` and `dynk_checkspace`, in order to check that the number of arguments is as expected for this function, and that there is enough space in the `{c|f|i}expr_dynk` arrays. If there is not enough space, it is automatically allocated, as described in Section 3.2.

Please also note that the initialization of the GET function, which returns the initial setting of an element, is only fully initialized after the `dynk_pretrack` subroutine has run.

6 Support for collimation version

The collimation version of SixTrack [19] is able to work around the 64 particle limit by tracking multiple samples of particles, i.e. loading the first 64 particles and then calling `thin6d` to track them for all turns, then loading the next 64 particles etc., as illustrated in Figure 2. For this to work correctly, DYNK must reset all element attributes at the beginning of each tracking simulation. This is accomplished in `dynk_apply`, which at the first turn of the first sample saves the original values (retrieved using `dynk_getvalue`), and then on the first turn of consecutive samples resets all elements and attributes touched by DYNK to the stored value. Furthermore, the outputfile `dynksets.dat` is only written while processing the first sample.

7 Checkpointing and restarting

The DYNK module supports checkpointing and restarting. For this to work, it must be able to truncate the `dynksets.dat` file to its position at the loaded restart point, and to store the current state of the DYNK functions. As all the current states of the functions are stored in the `{c|f|i}expr_dynk` arrays and in `fsets_dynk`, these arrays are written by the checkpoint/restart routines. Additionally, the array usage counters `n{c|f|i}expr_dynk` are also saved and restored. The rest of the state of DYNK is unchanged after initialization, and is thus recreated when reading the input files and initializing the simulations.

8 Summary and outlook

The DYNK module in SixTrack has been a success and has been in use for more than three years, enabling several new studies. Its overall architecture is designed to be extensible, to make it easy to add new functions, and to support acting on new types of elements and element attributes. The implementation of DYNK has also been taken as an opportunity to clean up several parts of the SixTrack code, as was done by implementing `initialize_element`. Furthermore, it not only supports the standard SixTrack, but also the collimation version and checkpoint/restart.

For the future, it is likely that several more functions and elements will be added, such as a wider variety of random distributions, and support for the beam-beam and multipole element. It may be interesting to support the setting of a global non-integer turn shift, in order to easily study the effects on different bunches along the ring. Furthermore, the performance of `dynk_setvalue` and `dynk_getvalue` might be improved by caching the element index, avoiding the search over all elements every time these functions are called. Finally, since DYNK can now change the reference energy using the GLOBAL-VARS mechanism, the large subroutines `thin6dua` and `thck6dua` can now in principle be removed, reducing the amount of code duplication.

References

- [1] F. Schmidt, “SixTrack Version 4.2.16 Single Particle Tracking Code Treating Transverse Motion with Synchrotron Oscillations in a Symplectic Manner”, CERN/SL/9456, 2012.
- [2] G. Ripken and F. Schmidt, “A symplectic six-dimensional thin-lens formalism for tracking”, DESY 95–63 and CERN/SL/95–12(AP), 1995.
- [3] R. De Maria, A. Mereghetti, and K. Sjobak, “SixTrack Status”, these proceedings.
- [4] K. Sjobak, R. De Maria, E. McIntosh, A. Mereghetti, J. Barranco, M. Fitterer, V. Gupta, and J. Molson, “New features of the 2017 SixTrack release”, IPAC’17, THPAB047.
- [5] A. Mereghetti et al., “SixTrack-FLUKA active coupling for the upgrade of the SPS scrapers”, IPAC’13, WEPEA064.
- [6] K. Sjobak, H. Burkhardt, R. De Maria, A. Mereghetti, and A. Santamaría García, “General functionality for turn-dependent element properties in SixTrack”, IPAC’15, MOPJE069.
- [7] A. Santamaría García, H. Burkhardt, K. Hernández Chahín, A. Macpherson, K. Sjobak, D. Wollmann, and B. Yee-Rendón, “Limits on failure scenarios for crab cavities in the HL-LHC”, IPAC’15, THPF095.
- [8] K. Sjobak, R. Bruce, H. Burkhardt, A. MacPherson, A. Santamaría García, and Regina Kwee-Hinzmann, “Time Scale of Crab Cavity Failures Relevant for High Luminosity LHC” IPAC’16, THPOY043.
- [9] A. Santamaría García, K. Sjobak, R. Bruce, H. Burkhardt, F. Cerutti, R. Kwee-Hinzmann, A. Lechner, and A. Tsinganis, “Machine protection from fast crab cavity failures in the High Luminosity LHC” IPAC’16, TUPMW025.
- [10] R. Apsimon, G. Burt, A. Dexter, P. Baudrenghien, K. Sjobak, and R. Appleby, “Modelling the low level RF response on the beam during crab cavity quench”, IPAC’17, MOPVA102.
- [11] R. Bruce, C. Bracco, R. De Maria, M. Giovannozzi, A. Mereghetti, D. Mirarchi, S. Redaelli, E. Quaranta, B. Salvachua, “Reaching record-low β^* at the CERN Large Hadron Collider using a novel scheme of collimator settings and optics”, Nuclear Instruments and Methods in Physics Research Section A, Volume 848, 11 March 2017, Pages 19–30, <https://doi.org/10.1016/j.nima.2016.12.039>.
- [12] H. Garcia-Morales et al., “Simulating off-momentum loss maps using SixTrack”, these proceedings.
- [13] H. Garcia-Morales, R. Bruce, and B. Salvachua Ferrando, “Off-momentum loss maps with one beam”, January 12th, 2016, CERN-ACC-NOTE-2016-0011, <https://cds.cern.ch/record/2121307>.
- [14] S.J. Wretborn, R. Bruce, H. Garcia Morales, and K. Sjobak, “Study of off-momentum losses at the start of the ramp in the Large Hadron Collider”, September 6th, 2017, CERN-ACC-NOTE-2017-0065, <http://cds.cern.ch/record/2298696>.
- [15] J. Wretborn, R. Bruce, H.G. Morales, and K. Sjobak, “Off-momentum collimation at the start of the ramp” Presented at the 215th LHC Collimation Working Group Meeting, April 3rd 2017, <https://indico.cern.ch/event/626908/>.
- [16] M. Fitterer, G. Stancari, A. Valishev, R. De Maria, S. Redaelli, K. Sjobak, and J.F. Wagner, “Implementation of hollow electron lenses in SixTrack and first simulation results for the HL-LHC” IPAC’17, THPAB041.
- [17] F. Schmidt, A. Alekou, M. Fitterer, J.F. Wagner, S.J. Wretborn, R. De Maria, S. Kostoglou, K. Sjobak, and T. Persson, “SixTrack version 4.7.16: Single Particle Tracking Code Treating Transverse Motion with Synchrotron Oscillations in a Symplectic Manner; User’s Reference Manual”, please see http://sixtrack.web.cern.ch/SixTrack/doc/manual_dev/six.pdf or

https://github.com/SixTrack/SixTrack/tree/master/Doc/user_manual
for the newest version.

- [18] SixTrack sources, <https://github.com/SixTrack/SixTrack/>
- [19] G. Robert-Demolaize, R. Assmann, S. Redaelli, and F. Schmidt, “A new version of SixTrack with collimation and aperture interface”, PAC’05, FPAT081.
- [20] Stephan Russenschuck, “Field Computation for Accelerator Magnets: Analytical and Numerical Methods for Electromagnetic Design and Optimization”, Wiley-WCH 2010, ISBN 978-3-527-40769-9.

HL-LHC Hollow Electron Lens Integration using MERLIN

H. Rafique^{1 4}, *R. B. Appleby*^{2 3}, *R. J. Barlow*¹, *J. G. Molson*⁵, *S. C. Tygier*^{2 3}, *A. Valloni*^{2 4}

¹University of Huddersfield, Huddersfield, UK, ²University of Manchester, Manchester, UK,

³Cockcroft Institute, Daresbury, UK, ⁴CERN, Geneva, Switzerland,

⁵LAL, Univ. Paris-Sud, CNRS/IN2P3, Université Paris-Saclay, Orsay, France

Abstract

The C++ accelerator physics library MERLIN has been developed for LHC collimation. Recently MERLIN has been upgraded to provide a robust tool for HL-LHC collimation, including the treatment of composite materials, and a hollow electron lens process. The HEL is being considered as a collimation enhancer for the HL-LHC. We use MERLIN to perform case studies of integrating a HEL at different points in the lattice, and in doing so, demonstrate recent updates to the code. Of particular interest when integrating is the roundness of the beam (in real space) at the position of the HEL, and the effect that this has on collimation enhancement.

Keywords

MERLIN; collimation; HL-LHC; hollow electron lens; tracking.

1 Introduction

The hollow electron lens (HEL) is an annular beam of electrons which may be used to interact with the LHC proton halo in order to control halo diffusion, thus providing a method of active halo control.

The HEL was demonstrated for the first time at the Tevatron [1] p^+p^- collider. The Tevatron Electron Lens (TEL) was used for abort gap cleaning [2] 24/7 for a duration of five years with only a few eight hour accesses to replace failed components [3], thus demonstrating compatibility with regular collider operation. TEL used a solid electron beam. The second lens, TEL-2, was used successfully for long range beam-beam compensation over a period of months [4]. TEL-2 was the first hollow electron lens to be used in a hadron collider, it demonstrated controllable halo particle removal without affecting the beam core [1]. The applications of electron lenses are summarised in [5].

The HL-LHC beam intensity is larger than in the LHC, thus at the same jaw openings the collimators can provide larger impedance contributions, and corresponding beam instability. The HEL can operate closer to the beam core than a collimator jaw without contributing significantly to the impedance budget, this may also allow some collimators to have larger jaw openings, as the HEL will kick halo protons onto the primary collimators to deplete the halo. As the HEL is not solid there is no material to damage, and due to the HELs SC solenoid the HEL beam is very well controlled. In controlling halo depletion the HEL offers a method of reducing the damage caused by catastrophic failures. If the halo is depleted before such a failure, the intensity of protons that would impact upon the collimators or magnets would be decreased. This is especially useful for the HL-LHC due to the use of crab cavities, which are a major concern in terms of possible failures which could displace the entire beam from the desired orbit by a number of σ [6].

The use of a HEL for LHC collimation was first suggested in 2006 [3], in which the highly reliable Tevatron hardware was summarised and scaling of this for use in the LHC was suggested. It was suggested in [3] that for the LHC, two to three HELs per beam, each operating with current of up to 10 A, length of 2 m, and voltage of 20 kV would be necessary.

Four existing operation modes exist for collimation enhancement using the HEL, these are detailed in [5], and summarised below:

- Direct Current (DC): The HEL current is continuously at maximum. Halo particles receive a kick that is proportional to the particle transverse displacement (as in equation 4), and collimation enhancement relies on the coupling of this kick with machine resonances.
- Alternating Current (AC): The HEL current is modulated over time in order to drive the betatron oscillations of halo protons. Collimation enhancement is achieved by increasing transverse displacement of the forced betatron oscillation.
- Diffusive: The HEL is either switched on/off, or the current is modulated, on a turn by turn basis. It is not possible to do this on a bunch by bunch basis in the LHC. By applying this randomly modulated kick to the halo protons, the natural diffusion of the halo, and thus the collimation enhancement is performed in a controlled manner.
- Turnskip: The HEL is switched to DC mode every n turns, in an attempt to drive the betatron oscillations as in the AC mode.

An investigation was performed in 2013 [7], in which SixTrack [9], the standard CERN collimation simulation tool was modified and used to perform simulations of a HEL in the nominal LHC lattice using hardware similar to the TEL. It was found that; the AC mode was most effective at exciting halo particles onto the primary collimator (around 75% removal in ≈ 20 s) after the AC parameters were optimised, the DC mode showed no noticeable effect, and the diffusive mode was less effective than the AC mode. It was also observed that doubling the current in diffusive mode was comparable to the AC mode removal rate.

Later in 2013 the HEL was mentioned in the HL-LHC preliminary design report [10], though not in the HL-LHC baseline at that time it was highlighted as a possible means for collimation enhancement in the HL-LHC.

After further study a conceptual design report (CDR) was produced in 2014 [11], in which experimental experience from TEL operation and numerical simulations were used to produce a conceptual design for a HEL that met the requirements for HL-LHC collimation enhancement. The hardware parameters of the TEL and LHC HEL are summarised in Table. 1.

Parameter	Tevatron	LHC
Interaction Length [m]	3	3
e^- Energy [KeV]	5	10
e^- Current [A]	1.2	5

Table 1: HEL hardware properties

At the proposed location for the HEL in the HL-LHC, the beam is not round. The HEL beam is round due to the cathode shape, and is maintained by the magnets and space charge forces [5]. Collimation enhancement is thus optimal in regions where the proton beam is round. We therefore compare the proposed position at RB46 (for beam 1) where the beam is non-round, to another position nearby where the beam is round, in order to compare the effect on collimation enhancement.

For this study the parameters detailed in Table. 2 were used. For SixTrack comparisons, the Tevatron HEL parameters shown in Table. 1 were used, with LHC optics version 6.503, shown in the LHC_{TeV} column of Table. 2. The remaining columns of Table. 2 compare the round and non-round positions for HL-LHC optics version 1.2. For the following HL-LHC simulations, collision optics (crossing off) using $\beta^* = 15$ cm were used, these were taken from the HL-LHC repository in December 2015.

2 Kick Calculation

The machine beam interacts with the electromagnetic (EM) field generated by the HEL beam, the HEL interaction is not based on scattering.

Parameter	LHC _{TeV}	HL-LHC Round	HL-LHC Non-round
s [m]	10032	9967	9908.4
β_x [m]	181.7	213.4	331.7
β_y [m]	180.4	215.9	211.9
α_x	-0.32	-0.82	-1.26
α_y	0.96	0.18	-0.09
D_x	-0.14	-0.35	-0.5
D_y	0.08	0.15	0.08
σ_x [μm]	292	316	395
σ_y [μm]	291	318	315
$\sigma_{x'}$ [μrad]	1.67	1.91	1.91
$\sigma_{y'}$ [μrad]	2.24	1.5	1.5
μ_x	25.16	24.35	24.25
μ_y	22.85	22.7	22.67

Table 2: HEL simulation lattice functions for nominal LHC (v6.503) and HL-LHC (v1.2) optics at both round and non-round positions in the HL-LHC.

In order to derive the force on a proton interacting with the HEL EM field, we assume a perfect HEL:

- The HEL is a perfect hollow cylinder, with no variation of thickness, or inner or outer radii (R_{min} and R_{max} respectively) through its interacting length L .
- The electron distribution is perfectly uniform both longitudinally and radially.
- The electron charge density is azimuthally symmetric.

These assumptions mean that any proton inside the HEL minimum radius R_{min} , i.e. the beam core, feel no net force and is therefore not affected by the HEL. This ideal HEL is shown in Figure 1.

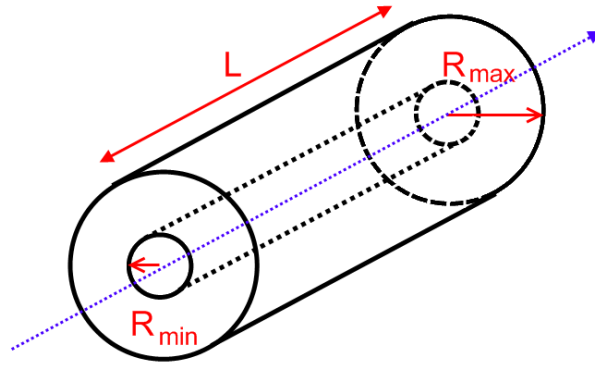


Fig. 1: Diagram of a perfect HEL beam, the blue line indicates the proton beam axis L is the active length, R_{min} the minimum radius, and R_{max} the maximum radius.

For modelling we also ignore the effect on protons from the magnetic fields, *i.e.* the SC solenoid, and the injection and extraction toroids. The HEL beam can overlap the proton beam at its injection and extraction points, we ignore this effect as it was demonstrated not to affect beam intensity or emittance at the Tevatron [5], and we assume the same to be true in the LHC case.

The full kick derivation may be found in [12]. We model the HEL beam as an infinite line charge, as shown in Figure 2.

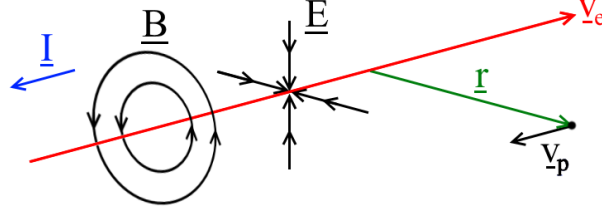


Fig. 2: HEL modelled as an infinite line charge, indicating direction of the EM fields that influence proton motion. The magnetic field lines B , the electric field lines E , the HEL current I , the electron velocity v_e , the proton velocity v_p , and the radial displacement of a proton r are shown.

The electric field E and magnetic flux density B generated by an infinite line charge are well defined (see [12]), and can be substituted into the Lorentz force equation to give the force on a proton with velocity v_p travelling parallel to the HEL at some transverse displacement r . Noting that the vectors B and v_p are perpendicular, we find the force on the proton in the lab frame to be:

$$F(r) = \frac{Iq(1 \pm \beta_e\beta_p)}{2\pi\epsilon_0 v_e r}. \quad (1)$$

Where I is the electron beam current, q is the proton charge, $\beta_e = \frac{v_e}{c}$ is the normalised electron velocity, and $\beta_p = \frac{v_p}{c}$ is the normalised proton velocity. The \pm originates from the addition or subtraction of the magnetic flux density term from the electric field term in the Lorentz force, which is dependent on the relative directions of proton and electron beams.

As the electric field E is generated by electrons, the direction of this force on a proton is attractive, i.e. towards the propagation axis of the charge, which in this case is the centre of the HEL beam.

We use the angular velocity to translate this force into the maximum angular kick given to a proton interacting with the HEL:

$$\theta_{max}(r) = \frac{1}{4\pi\epsilon_0 c^2} \frac{2LI_r(1 + \beta_e\beta_p)}{(B\rho)_p\beta_e\beta_p} \frac{1}{r}, \quad (2)$$

where I_r is the charge enclosed by the radius r , and L is the length of the HEL. We may define a function $f(r)$, which, for an ideal HEL with uniform charge density and radial symmetry, modulates the charge enclosed I_r in this expression to take into account the hollow cylindrical distribution of electrons:

$$f(r) = \begin{cases} 0, & r < R_{min} \\ \frac{r^2 - R_{min}^2}{R_{max}^2 - R_{min}^2}, & R_{min} < r < R_{max} \\ 1, & r > R_{max} \end{cases} \quad (3)$$

where:

r is the machine particle radius in transverse real space,

R_{min} is the minimum HEL radius in transverse real space, and

R_{max} is the maximum HEL radius in transverse real space.

Thus the magnitude of the HEL kick on a proton is defined as a function of the protons transverse radius r , the beam rigidity of the machine beam $(B\rho)_p$, the HEL current I , length L , and the HEL and machine beam normalised velocities β_e and β_p respectively, in equation 4:

$$\theta_{kick}(r) = f(r) \cdot \frac{1}{4\pi\epsilon_0 c^2} \frac{2L_{HEL}I(1 + \beta_e\beta_p)}{(B\rho)_p\beta_e\beta_p} \frac{1}{r}. \quad (4)$$

3 SixTrack Comparison

In order to benchmark the MERLIN HEL process, the test case of the Tevatron HEL hardware in the nominal LHC (lattice v.6.503) was repeated as in [7]. In this case the parameters used were those in the first column of tables 1 (with the caveat that the active length $L = 2$ m) and 2, all simulations in this section use these parameters unless otherwise stated. Here the original version of the HEL in SixTrack was used, as in [7], since this version, a new and updated implementation of the HEL in SixTrack has been created [8].

3.1 HEL Profile

In both SixTrack and MERLIN, there exist two HEL profiles. These are the selection functions $f(r)$ used to calculate the kick θ_{kick} shown in eqn. 4 as opposed to the maximum kick θ_{max} shown in eqn. 2. The ‘radial’ profile is a parameterisation of the measured profile of the prototype LHC HEL cathode. The selection function, shown in eqn. 3, is referred to as the ‘perfect’ or ‘simple’ profile; the profile given for a perfectly symmetrical e^- distribution with uniform charge density.

Figure 3 compares the two profiles for the Tevatron HEL hardware in the nominal LHC, as expected both codes are equivalent.

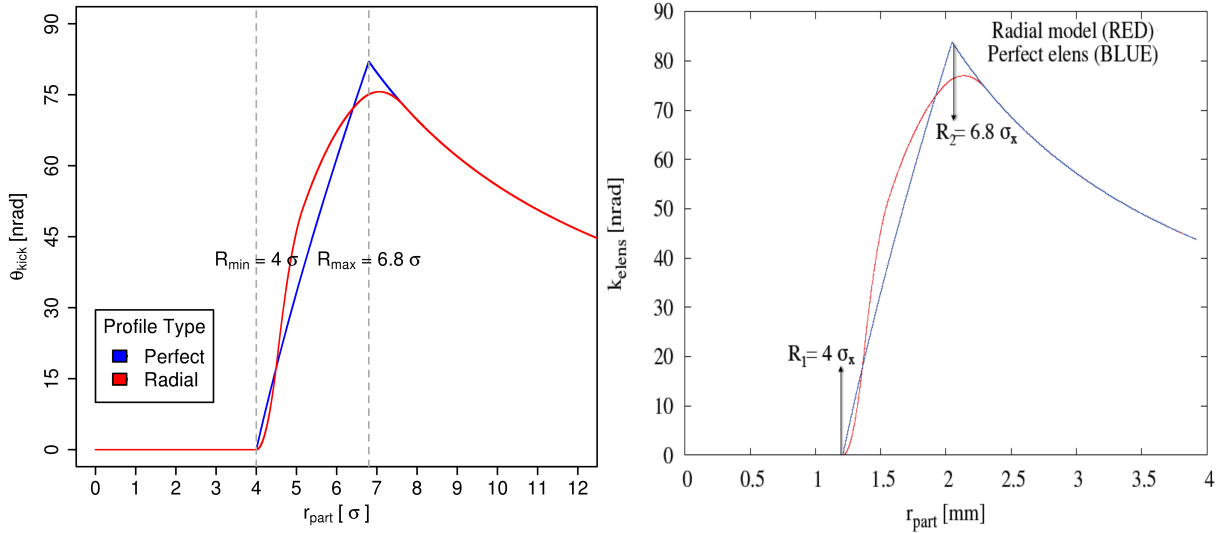


Fig. 3: HEL radial kick profiles, comparing MERLIN (right) with SixTrack (left), for identical Tevatron HEL parameters.

It is important to note that, as mentioned previously, the HEL force is attractive to protons, thus the profiles in Fig. 3 are in fact only magnitudes as the actual kick is negative, i.e. towards the centre of the beam axis. Also all models are constructed such that if a particle is within the HEL minimum radius R_{min} , there is no kick, a result of the assumption of uniform electron density and radial symmetry of the HEL beam. Also the HEL will effect particles outside the maximum HEL radius R_{max} , the kick being similar to that from a line charge at the centre axis.

We note that the radial profile maximum is not at R_{max} as we would expect. This could be due an artifact from measurement, or some other error. We will assume that this is correct however in order to compare directly with SixTrack.

3.2 Tracking Differences

It is important to note that there are differences in the tracking between MERLIN and SixTrack. We use the thick lens symplectic tracker in MERLIN, whereas a thin lens symplectic tracker is used in SixTrack. The difference is evident from Fig. 4, where the Poincaré sections for 64 equally spaced protons between $0 - 10 \sigma_x$ are plotted for 10^4 turns in the nominal LHC, at the position of the HEL. This initial distribution sets all other initial particle coordinates to 0, which means that it represents an ideal bunch with no transverse momentum components, and no longitudinal displacement. This is not representative of an accelerator beam, and is only used to demonstrate the effect of the HEL on what should otherwise be perfectly stable motion in the accelerator, which is represented as smooth ellipses in phase space as shown in Fig. 4. This small perturbation may affect all Poincaré sections shown in subsequent sections, but should not compromise the comparison in terms of halo depletion rates.

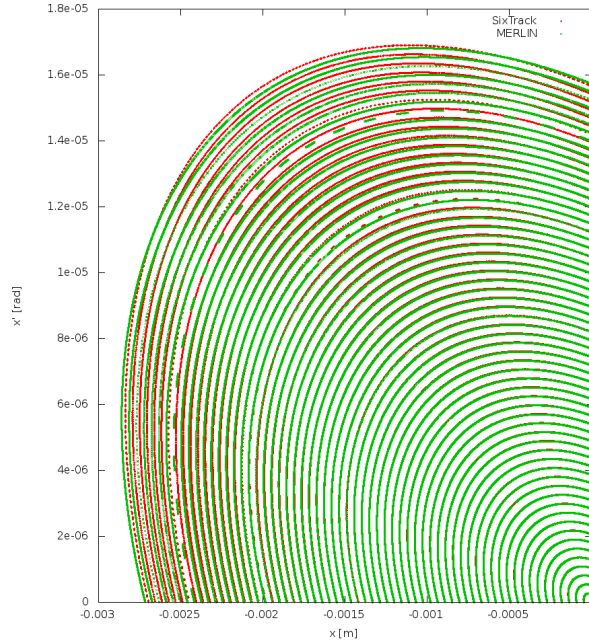


Fig. 4: Poincaré section for 64 protons in the nominal LHC, comparing SixTrack (red), and MERLIN (green) tracking. Particles are initially populated between $0 - 10 \sigma_x$.

3.3 Poincaré Analysis

Due to the differences in tracking between MERLIN and SixTrack we do not expect identical results when plotting the Poincaré sections for HEL operation modes. The first and simplest operation mode to be implemented is the DC mode, in which the HEL constantly runs at maximum current, and collimation enhancement relies on the coupling with machine resonances. This is manifest in the formation of islands in the Poincaré section as shown in Fig. 5.

We observe in Fig. 5, that in both MERLIN and SixTrack there is no perturbation to normal particle motion inside the HEL minimum radius R_{min} , this is as expected as both models of the HEL assume no effect inside R_{min} . In SixTrack islands are created near $4 \sigma_x$, whereas in MERLIN perturbations take the form of small ripples. It was later found that the machine optics used in MERLIN was different to that in SixTrack. When combined with tracking differences, this may account for variation between the two codes as the Poincaré sections are affected by octupole and chromaticity settings. Octupoles drive resonances, and chromaticity effects how the particles are swept over these resonances. The perturbations are of similar magnitudes in both codes.

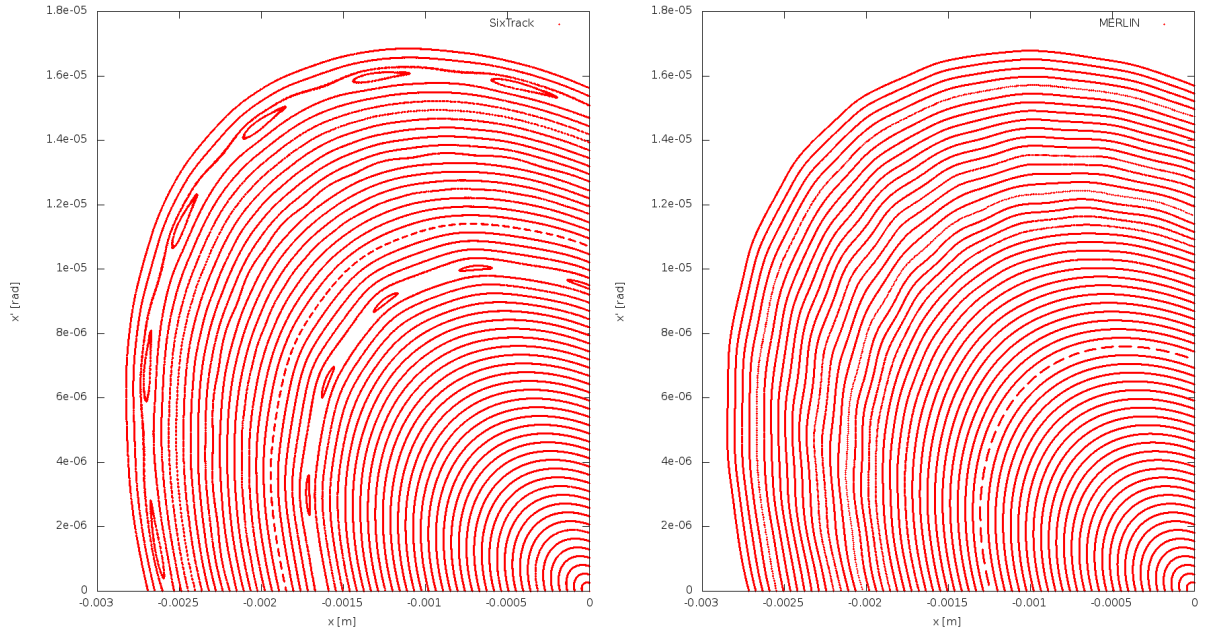


Fig. 5: Poincaré section for a DC HEL in the nominal LHC, comparing SixTrack (left), and MERLIN (right). Particles are initially populated between $0 - 10 \sigma_x$.

The AC mode relies on driving the betatron oscillations of the protons by modulating the HEL current at a frequency in resonance with the machine tune. This mode takes a large number of variables in order to define the modulation. A full investigation of this mode, including the optimisation of the parameters used, was performed in [7], and has not been repeated using MERLIN. Instead the optimal parameter settings (referred to as H20 in the previous investigation) from [7] is used for all AC operation in this article. The effect of this mode is shown in Fig. 6, where as well as the ripples caused by the DC operation, the Poincaré sections are widened in both MERLIN and SixTrack. Tracks from SixTrack are more diffuse than those from MERLIN, there are many possible reasons for this, including the lack of optimisation of the AC parameters in MERLIN, and possible differences in octupole settings.

When considering the HEL as a collimation enhancer, our aim is to force halo particles onto collimators. The primary collimator is typically located at a position of $\approx 6\sigma$ in the respective plane. Thus our goal is to excite a proton that exists between $4 - 6 \sigma$, to a larger transverse displacement such that it will be absorbed by the collimation system. In the case of a DC HEL, this will only occur if the proton crosses a resonance - accelerators are designed to minimise the probability of this. The proton must also have a transverse displacement near that of the collimator jaw, as the HEL gives only a small displacement in transverse phase space.

With the AC HEL, the widening of particle tracks in transverse phase space is observed to be larger than the displacement given by the DC HEL. This means that collimation enhancement should be greater for the AC mode when compared to the DC mode, as more halo particles will be excited to a displacement large enough to be intercepted by the collimation system.

Diffusive HEL operation gives a random kick to the halo on a turn by turn basis in order to enhance the diffusion of halo particles onto a collimator. Figure 7 compares the diffusive HEL operation in MERLIN and SixTrack. As expected the transverse displacement in both codes is much larger than all other operation modes. Not only does this mode offer the greatest collimation enhancement, it is not dependent on rigorous knowledge of the machine tune as in the AC case.

Comparing the maximum displacement after 10^4 turns, shown in Fig. 7, we may consider MER-

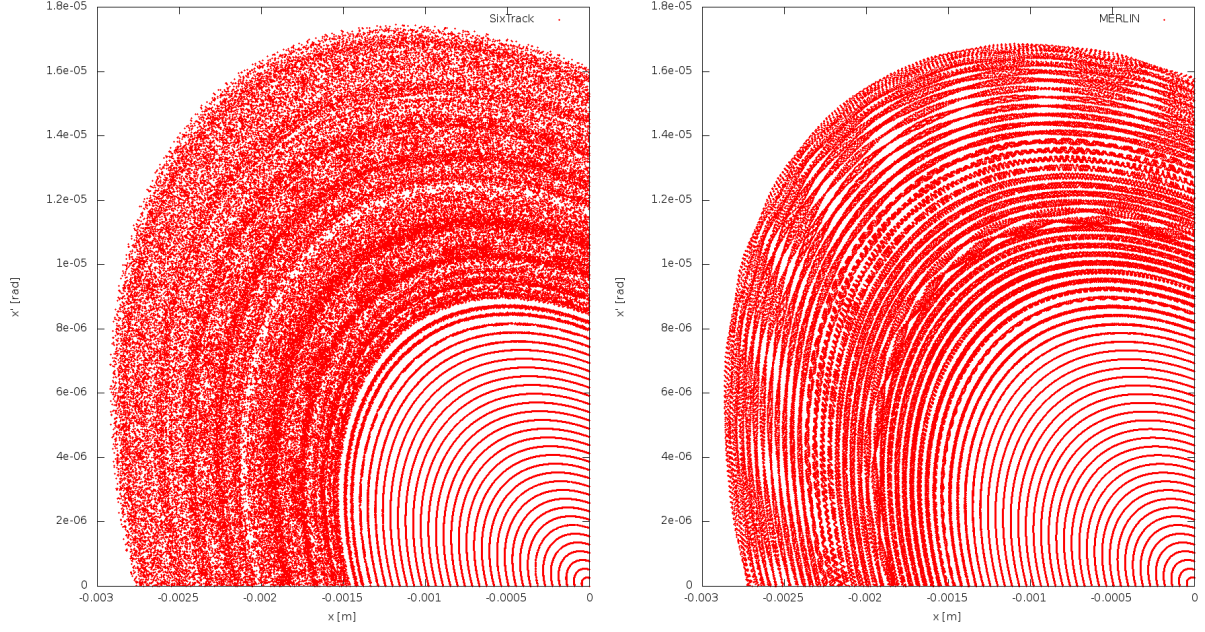


Fig. 6: Poincaré section for an AC HEL in the nominal LHC, comparing SixTrack (left), and MERLIN (right). Particles are initially populated between 0 - $10\sigma_x$.

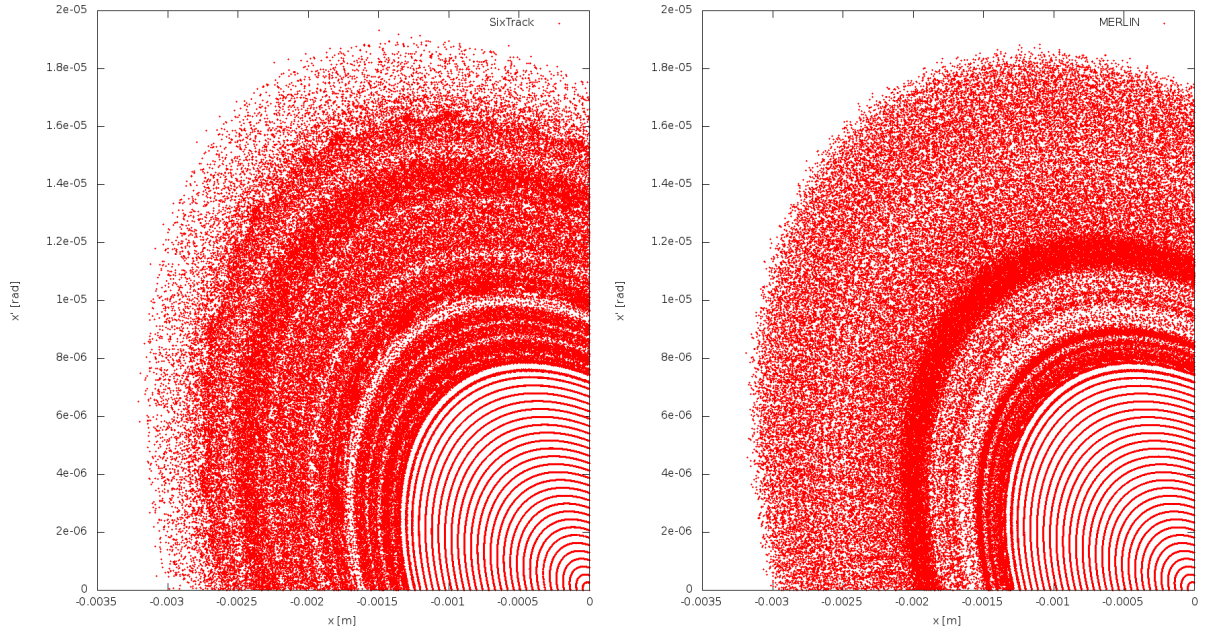


Fig. 7: Poincaré section for a diffusive HEL in the nominal LHC, comparing SixTrack (left), and MERLIN (right). Particles are initially populated between 0 - $10\sigma_x$.

LIN and SixTrack to be similar.

By directly comparing the DC, AC, and diffusive operation modes of the HEL in MERLIN and SixTrack, we may conclude that SixTracks additional physics processes and thin lens tracking cause only a small difference between the two codes, there is also a small effect from the different optics used. In reality the variation in HEL Poincaré sections due to the differences between MERLIN and SixTrack are

small, and for a large number of particles and turns are likely to be negligible.

3.4 Simulated Bunch

A MERLIN HELHaloDistribution was used to create two LHC bunches of 10^3 protons, one populated between $0 - 4 \sigma$ which will be referred to as the core, and the other populated between $4 - 6 \sigma$ which will be referred to as the halo. These initial distributions are shown in green in Fig. 8, together with their Poincaré sections for 100 turns in purple. The HEL R_{min} and R_{max} are indicated in red.

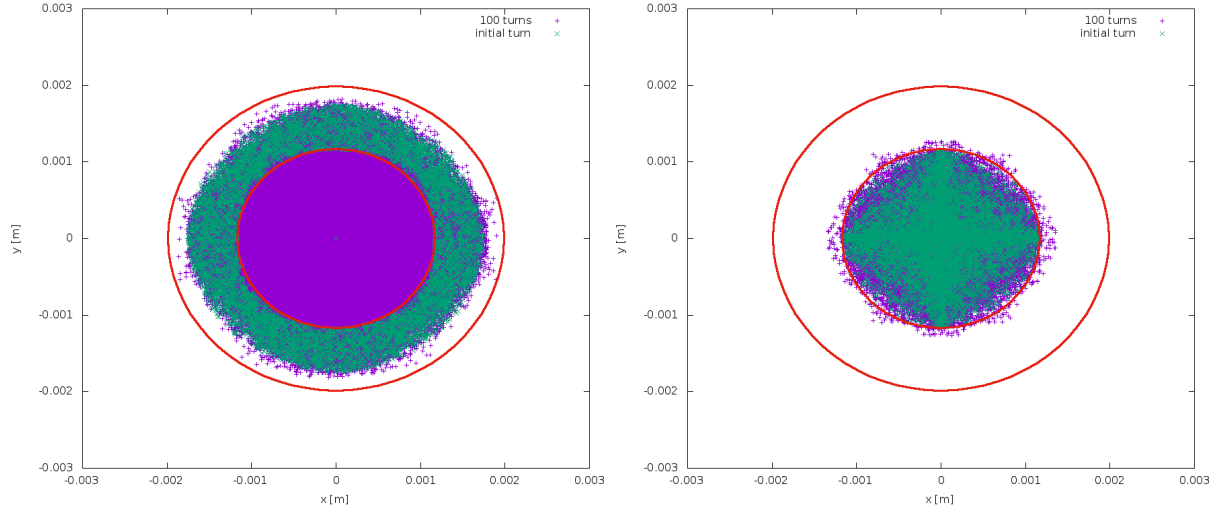


Fig. 8: HEL footprint at the position of the HEL in the nominal LHC; red lines show R_{min} and R_{max} , green points are the initial distribution, purple are a Poincaré section of this bunch over 100 turns. The left plot shows an initial halo distribution between $4 - 6 \sigma_x$, the right plot shows an initial core distribution between $0 - 4 \sigma_x$.

From Fig. 8 we see that the beam is round, which is one of the reasons for the selection of this position in the nominal LHC, we also observe that despite initially populating the core up to a maximum of 4σ , a very small percentage of particles may have a larger transverse displacement for a few turns, and thus interact with the HEL. As the HEL kick is very small, interaction for a small percentage of particles for only a few turns will have a negligible effect, therefore we are not concerned with long term tracking of the core. It is clear that for a halo distribution, particles traverse the area inside $0 - 4 \sigma$ despite being initially populated outside this range, thus the HEL does not necessarily operate on a halo particle at every turn.

We must also note that if simulating a halo bunch populated between $4 - 6 \sigma$, in the presence of a collimator with an insertion of 6σ , a small percentage of the halo will impact upon the collimator without the presence of a HEL. Therefore it is more prudent to simulate a halo between $4 - 5.8 \sigma$ so that we may have negligible losses in the case where no HEL is present in the lattice, and thus compare the HEL cleaning enhancement to that of no enhancement when no HEL is present.

3.5 Collimation Enhancement Comparison

The initial distribution for HEL simulations is a HELHaloDistribution from $4 - 5.8 [\sigma]$ with no longitudinal component. Figure 9 shows particle survival when using a HEL in the nominal LHC at $s = 10037 m$. Collimator jaw openings are shown in Table 3. We observe that the diffusive mode is the only one to enhance collimation in these simulations. MERLIN excels over SixTrack in that it can perform full collimation scattering and on-line aperture checking with a HEL for many turns within a reasonable

simulation time, whereas previous simulations using the HEL in SixTrack omitted scattering in order to minimise run time.

Figure 10 shows the effect of doubling the diffusive HEL current; particle survival is roughly halved. The same investigation performed using SixTrack [7] is compared in Fig. 10, both codes agree despite the use of full collimation scattering in MERLIN. The impact distribution on the primary collimator is shown for these two currents in Fig. 11, we observe that the doubled current increases the likelihood of larger impact parameters.

Insertion Region	Collimator Family	Setting [σ]
7	Primary	6
	Secondary	7
	Absorber	10
3	Primary	15
	Secondary	18
	Absorber	20
1	Tertiary	8.3
	Absorber	10
5	Tertiary	8.3
	Absorber	10
6	Dump Protection	8
	Secondary	7.5
2	Tertiary	30
8	Tertiary	30

Table 3: LHC collimation settings used for HEL simulations. An emittance of $3.5 \cdot 10^{-6} m$ is used.

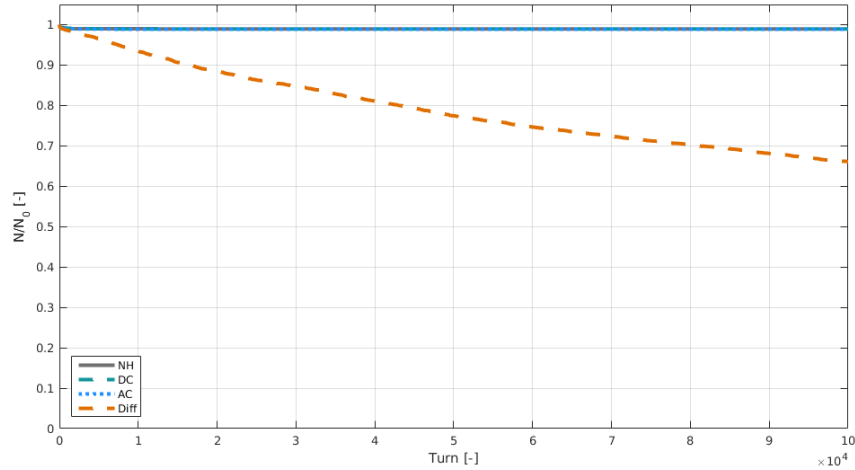


Fig. 9: Normalised particle survival for 10^5 turns in the nominal LHC, with the HEL using various operation modes, and without a HEL. The initial halo distribution is populated between $4 - 5.8 \sigma$, and the HEL operates between $4 - 6.8 \sigma$. In this simulation full collimation scattering is performed.

4 HL-LHC

For the HL-LHC the HEL is a promising tool for active halo control. The HEL is not currently part of the baseline, this will be decided after operational experience of the LHC at nominal settings (*i.e.* post 2016). In order to inform the decision, we use numerical simulations to estimate the cleaning enhancement of

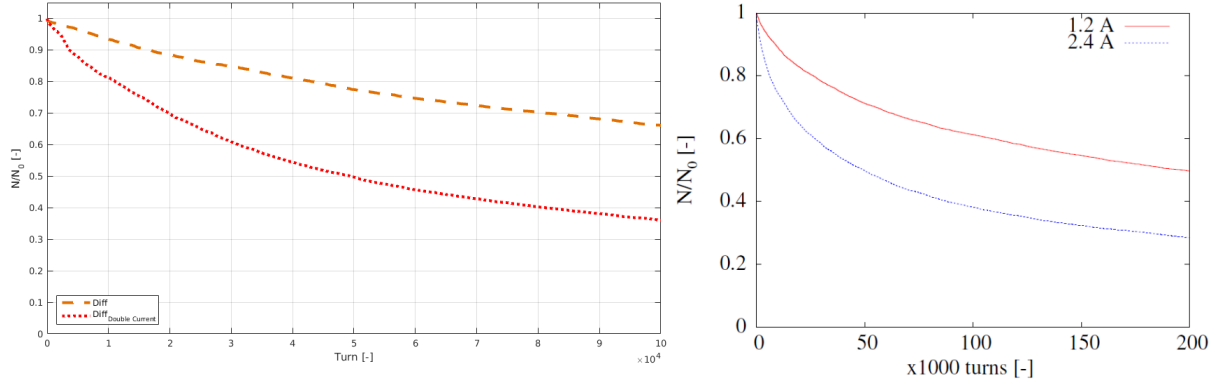


Fig. 10: Normalised particle survival for 10^5 turns in the nominal LHC, with a diffusive HEL, for an initial halo distribution populated between 4 - 5.8σ , and a HEL operating between 4 - 6.8σ (left). Compared to a similar simulation performed with SixTrack (from [7]) for $2 \cdot 10^5$ turns on the right. In both cases two sets of data are shown, one with a diffusive HEL current of 1.2 A, and the second with double the current, 2.4 A.

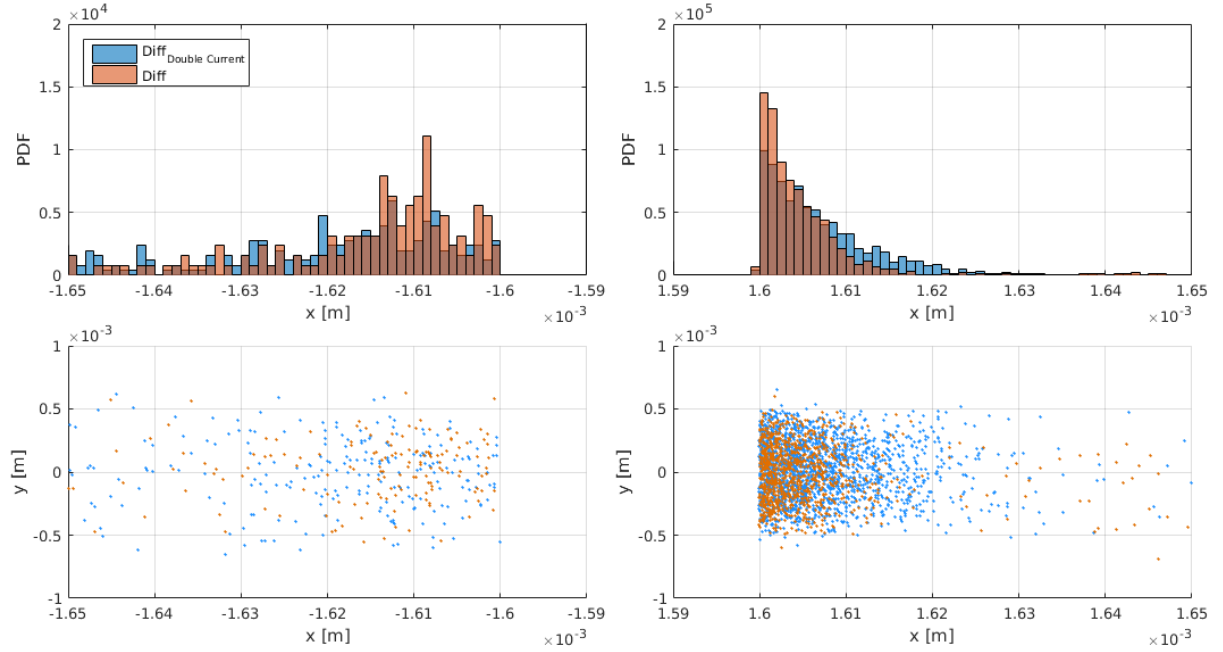


Fig. 11: The jaw impact distribution for the primary collimator for 10^5 turns in the nominal LHC, with a diffusive HEL, for an initial halo distribution populated between 4 - 5.8σ , and a HEL operating between 4 - 6.8σ . The top two plots are histograms of the distributions on the left and right collimator jaws respectively. The bottom two plots show the impacts on the jaw in transverse space. Orange shows data when using a diffusive HEL current of 1.2 A, and the blue with double the current, 2.4 A. It is evident that a larger HEL current produces larger impact parameters.

a HEL in the HL-LHC. Inevitably this depends on many factors: the hardware used, as well as the capability of the magnet systems to shape, translate, and maintain the electron beam. The sensitivity of cleaning enhancement to HEL operational parameters (active length, electron current etc.) can be assessed using numerical simulations. First it is important to identify the sensitivity to the position of the HEL in the HL-LHC. We will use beam 1 as the two beams offer similar optics.

4.1 HEL Integration

The preliminary HEL position in RB46, 88.6 m upstream of IP4, was decided because of the available space, and the fact that there are currently no hardware conflicts at this position. The transverse beam shape in real space (x, y) is not round at this position. As discussed earlier the HEL beam is considered to be perfectly round. Due to the beam shape at IP4 - 88.6 m, this will be referred to as the ‘non-round’ position.

In order to gauge the sensitivity of HEL cleaning enhancement to the HEL position, two subsequent HEL integration points have been chosen for comparison. The first position offers a round beam ($\beta_x \approx \beta_y$), and is located at $(s = IP4 - 30 \text{ m})$. This will be referred to as the ‘round’ position. The final identified position offers a more elliptical beam than the non-round position, and is located at $(s = IP4 - 119 \text{ m})$, this will be referred to as the ‘oval’ position.

Figure 12 shows the beta functions at the three identified positions, from which the difference in beam roundness is clear. The dispersion in this region is low as it is an interaction point.

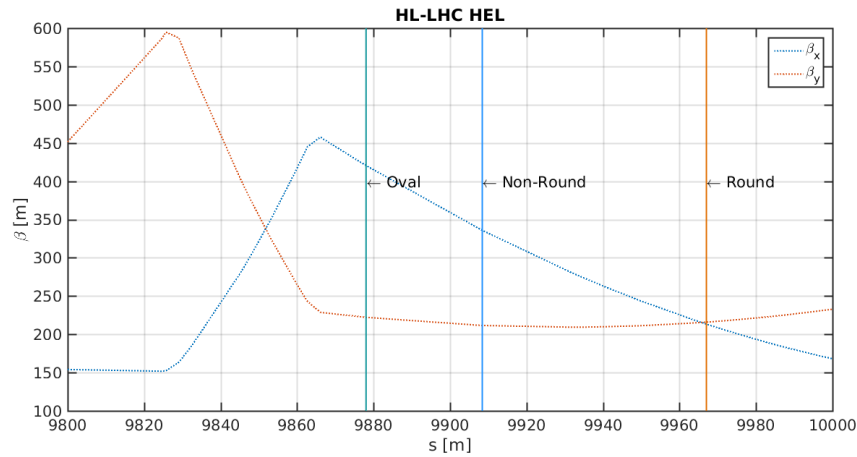


Fig. 12: Beta functions at the positions of the three identified HEL locations for this study.

Having benchmarked the MERLIN HEL process against that in SixTrack, we will use it to compare the effect of HEL operation modes at the round and non-round HEL positions in the HL-LHC, and confirm that HEL cleaning enhancement is reduced as the machine beam becomes less round, using the oval position. Subsequently the most successful existing operation mode at each of the three chosen integration positions will be ascertained. MERLIN will then be used to propose and investigate novel geometrical operation modes to improve the cleaning enhancement at the non-round and oval positions, in order to negate the expected decrease in collimation enhancement at non-round positions. In following this plan of research not only will we inform the design of the HL-LHC, but we will demonstrate the use of MERLIN as a complete tool for collimation with additional physics processes.

4.2 HEL Profile

We see from equation 2 that the maximum HEL kick is inversely proportional to the HEL maximum radius R_{max} . Thus at positions where the beam size is larger, the maximum HEL kick is smaller. This is shown in Fig. 13, where θ_{max} is larger for the round beam due to the smaller beam size in x .

Figure 13 also shows us that the radial model is no longer applicable, and gives a θ_{max} larger than the theoretical maximum. The radial profile is a parameterisation of the prototype cathode which had $g = \frac{R_{max}}{R_{min}} = 1.7$, whereas the LHC cathode is defined to have $g = 2$ [11]. By using a radial profile, numerical studies are brought closer to reality, thus the parameterisation for the radial profile was empirically adjusted until the expected profile was obtained. The result is shown in Fig. 14.

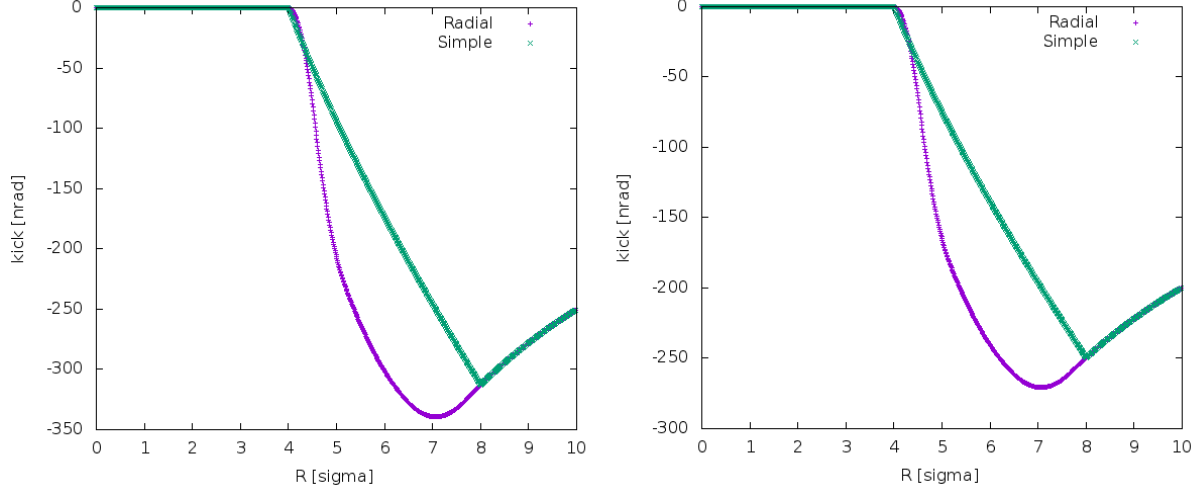


Fig. 13: HEL radial kick profiles taken along the transverse horizontal axis, comparing the round (left) and non-round (right) positions for the LHC HEL in the HL-LHC, where $R_{min} = 4 \sigma_x$ and $R_{max} = 8 \sigma_x$. Both the perfect (simple) and radial models are shown. It is clear that the radial model produces a maximum kick larger than the theoretical maximum.

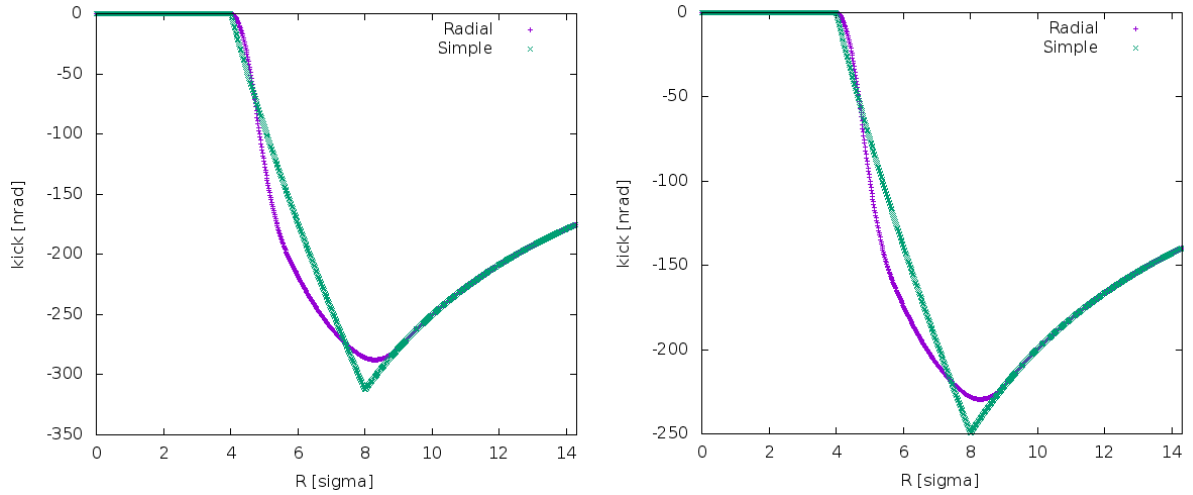


Fig. 14: HEL radial kick profiles taken along the transverse horizontal axis, comparing the round (left) and non-round (right) positions for the LHC HEL in the HL-LHC, where $R_{min} = 4 \sigma_x$ and $R_{max} = 8 \sigma_x$. Both the perfect (simple) and empirically adjusted radial models are shown. The adjusted radial model now provides a more realistic kick profile.

The corrected radial profile will be used for all simulations that follow.

4.3 Initial Bunch Distribution

A MERLIN HELHaloDistribution was used to create two HL-LHC bunches, one for the core populated between $0 - 4 \sigma$, and the other for the halo, populated between $4 - 6 \sigma$. The initial distributions, the footprint of these bunches for 100 turns, and an indication of R_{min} and R_{max} are shown in Fig. 15 at the non-round position, and Fig. 16 for the round position.

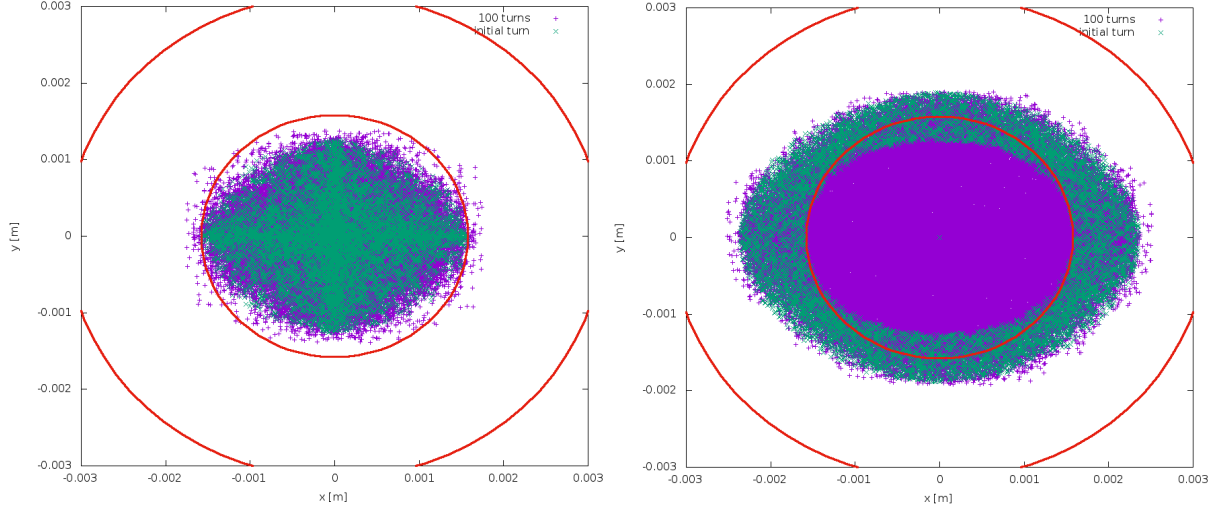


Fig. 15: HEL footprint at the non-round position of the HEL in the HL-LHC; red lines show R_{min} and R_{max} , green points are the initial distribution, purple are a Poincaré section of this bunch over 100 turns. The left plot shows an initial core distribution between 0 - 4 σ_x , the right plot shows an initial halo distribution between 4 - 6 σ_x .

At the non-round position we observe the larger HEL radii, and the fact that R_{min} touches the extremities of the core only in the horizontal. This results in a smaller overlap of the beam halo and the HEL, as shown in the halo footprint (right plot).

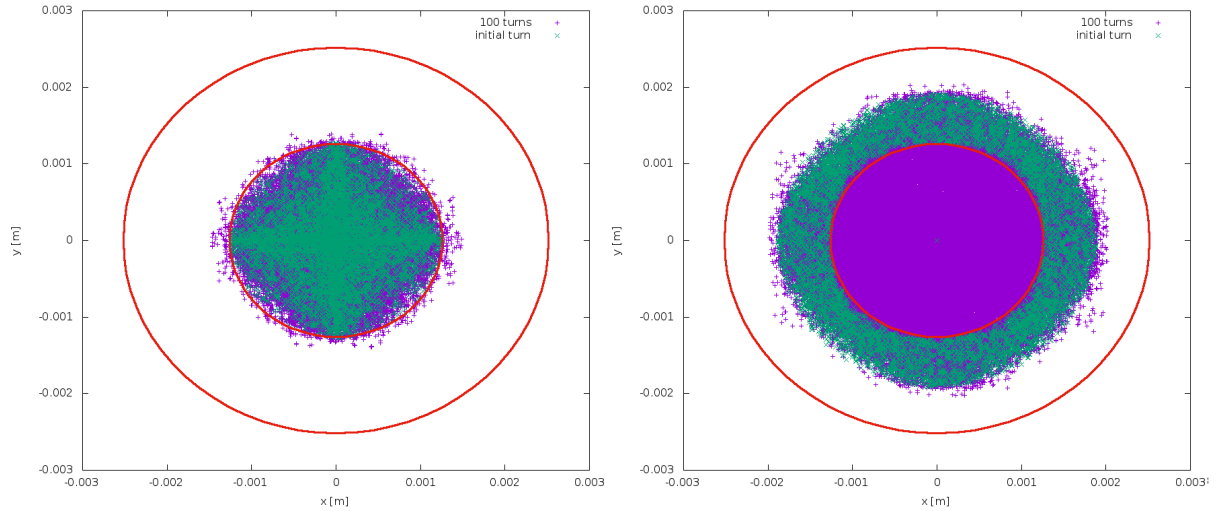


Fig. 16: HEL footprint at the round position of the HEL in the HL-LHC; red lines show R_{min} and R_{max} , green points are the initial distribution, purple are a Poincaré section of this bunch over 100 turns. The left plot shows an initial core distribution between 0 - 4 σ_x , the right plot shows an initial halo distribution between 4 - 6 σ_x .

At the round position the HEL radii are smaller in real space, and as seen for the nominal LHC case, a small percentage of the core has a transverse displacement larger than 4 σ for a small number of turns. As the beam is nearly round ($\frac{\beta_x}{\beta_y} \approx 1.01$), R_{min} encloses the extremities of the core, meaning a complete overlap with the initial halo. Due to the reduced overlap and thus interaction with the halo at the non-round position, we expect the HEL cleaning enhancement to be reduced when compared to the

round position.

4.4 Current Modulating Operation

We begin by comparing the current modulating operation modes at the three integration points. The following figures display particle survival as a function of turns in the machine. Survival when using a HEL at the round position is shown in Fig. 17, at the non-round position in Fig. 18, and at the oval position in Fig. 19.

As observed previously the diffusive mode offers the greatest collimation enhancement. The AC mode (which has not been optimised in terms of defining parameters) only kicks those particles nearest to the collimator on to it, performing an almost instant cut of these halo protons. We confirm that, as expected, the round position offers greatest enhancement, with a 54.31% halo removal after 10^5 turns, 20.85% at the non-round, and 18.53% at the oval position.

It is interesting to note that at the non-round and oval positions the AC mode offers comparable cleaning enhancement to the diffusive mode after 10^5 turns, though it is clear that the diffusive mode would cause greater enhancement over a longer period of time whereas the AC mode appears to plateau.

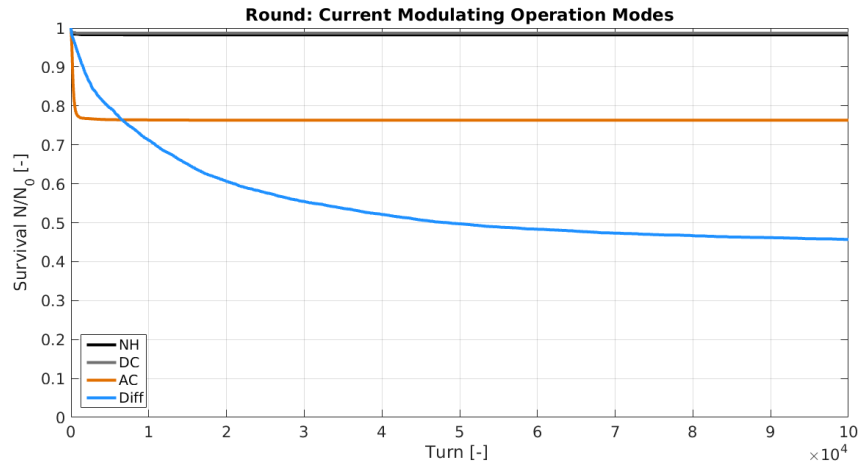


Fig. 17: Survival fraction of a HELHalo bunch populated between $4 - 5.8 \sigma$ for 10^5 turns in the HL-LHC with no HEL (NH), and DC, AC, and diffusive current modulating HEL operation modes, at the round beam position ($s = IP4 - 30$ m).

As seen from Figures. 17 - 19, the DC mode offers almost no collimation enhancement in the HL-LHC, with a halo reduction of 1.3%, 1.98%, and 2.24% at the round, non-round, and oval positions respectively. This is likely due to the optics being used, which has low chromaticity and octupole current. Octupoles provide resonances which can be exploited by the HEL, and particles are more likely to experience such resonances with increased chromaticity. Thus the DC operation mode shall be ignored.

There is a small improvement at the round position when using the AC mode (not optimised), though collimation enhancement between the non-round and oval positions are similar. The halo is reduced by 23.7%, 18.84%, and 17.89% at the round, non-round, and oval positions respectively.

We clarify the significant improvement at the round position when using the diffusive mode in Fig. 20. We aim to bring the non-round collimation enhancement closer to that of the round position, which is the motivation for attempting novel HEL operation modes.

Halo survival for the existing HEL operation modes in the HL-LHC are summarised in Table 4.4.

In practical operation the AC mode requires good knowledge of the machine tune, and numerical simulations have shown that a number of AC parameters must be optimised for each machine lattice [7].

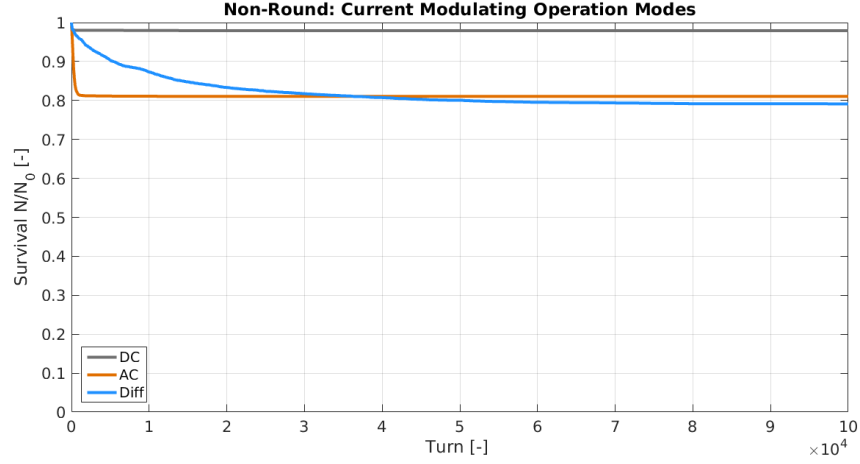


Fig. 18: Survival fraction of a HELHalo bunch populated between $4 - 5.8 \sigma$ for 10^5 turns in the HL-LHC with no HEL (NH), and DC, AC, and diffusive current modulating HEL operation modes, at the non-round beam position ($s = IP4 - 88.6$ m).

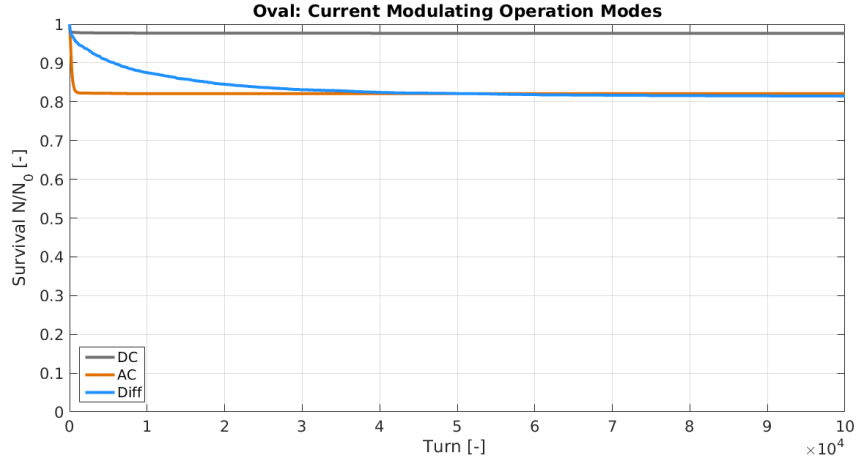


Fig. 19: Survival fraction of a HELHalo bunch populated between $4 - 5.8 \sigma$ for 10^5 turns in the HL-LHC with no HEL (NH), and DC, AC, and diffusive current modulating HEL operation modes, at the oval beam position ($s = IP4 - 119$ [m]).

The diffusive mode offers much simpler operation as the HEL may either be switched on or off or the current may be modulated randomly, on a random turn-by-turn basis. Though this may appear to be a boon, it may not be practical for the cathode to operate in the diffusive mode when compared to the AC, as it is more demanding to switch the cathode and corresponding electronics and magnets of the HEL continuously from zero to maximum current. The AC mode allows comparatively gentle ramping of currents when the parameters are set.

As no investigation of AC parameters was performed the AC mode in these simulations may not be enhancing collimation as well is possible. This provides another direction for future investigation using MERLIN.

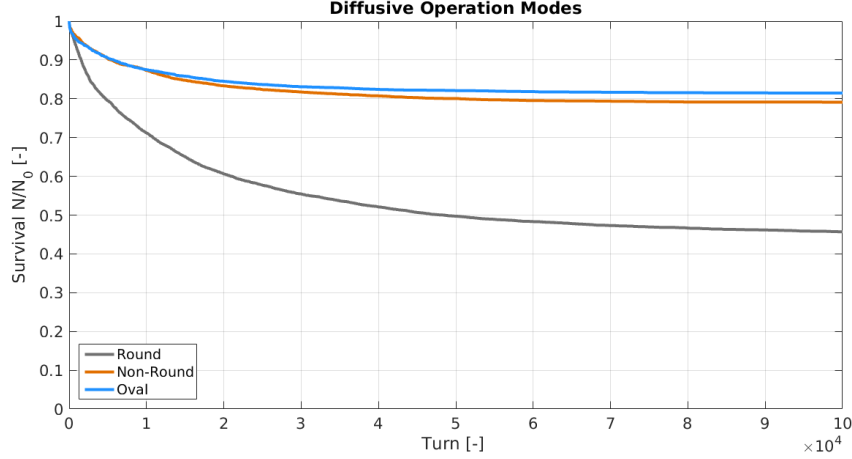


Fig. 20: Survival fraction of a HELHalo bunch populated between 4 - 5.8 σ for 10^5 turns in the HL-LHC using a HEL in diffusive mode, at the round, non-round, and oval beam positions.

Operation Mode	Round	Non-Round	Oval
DC	98.7	98.02	97.76
AC	76.3	81.16	82.11
Diffusive	45.69	79.15	81.47

Table 4: Particle survival $\frac{n}{n_o} \%$ for an initial halo of 10^4 particles between 4 - 5.8 σ after 10^5 turns in the HL-LHC with a HEL in the corresponding position using existing operation modes.

4.5 Novel Elliptical Matching

For operation of a HEL on a ‘non-round’ beam (*i.e.* where $\beta_x \neq \beta_y$) an ‘elliptical’ matching has been devised to attempt to maximise the transverse overlap of the HEL with the non-round beam halo.

Assuming dispersion to be negligible, we may illustrate the transverse footprint of the beam at a given position in the accelerator using an ellipse with semi-major and semi-minor axes (a, b respectively) being proportional to the beta functions in x and y . That is to say an ellipse with $a = n\sigma_x$ and $b = n\sigma_y$, where n is the number of sigma that denotes the minimum HEL radius. When $\beta_x \approx \beta_y$, this ellipse is a circle. For the non-round position in the HL-LHC, $\beta_x \approx 1.3 \cdot \beta_y$, and thus the beam footprint may be depicted as an ellipse with $a = 1.3 \cdot b$. As the HEL minimum radius R_{min} is set using σ_x , this results in a radius that is $\sqrt{1.3} = 1.14$ times too large in the y plane (as $\sigma = \sqrt{\beta\epsilon}$), as shown in Fig. 21. Thus the halo in one plane does not fully overlap with the HEL, resulting in a diminished collimation enhancement.

By taking an extreme case we may derive an expression to modify the radii and offset of the HEL, in order for it to overlap with more of the halo in both planes. This is shown schematically in Fig. 22.

By setting R_{min} to meet the beam core ellipse at its semi-minor extremity (in this case the maximum y), and crossing both semi-major extremities (in this case the maximum x values), we may use simple trigonometry to find the magnitude of the required HEL inner radius, which we will label $R_{min} (elliptical)$:

$$R_{min} (elliptical) = \frac{a^2 + b^2}{2b}. \quad (5)$$

We must also express the shift in co-ordinates (in this case in y), y_{shift} , as:

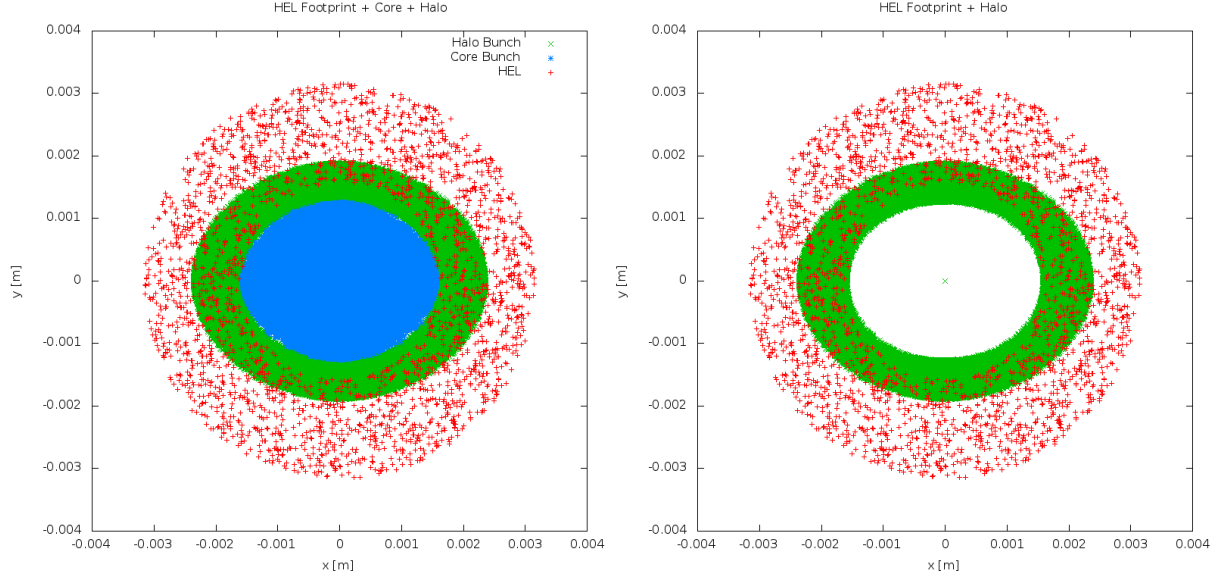


Fig. 21: Diagram of the HEL intersection with a non-round beam footprint. The red points indicate the electrons in the HEL, the green points indicate the position of the halo ($4-6 \sigma$ in x and y), and the blue points the core of the beam ($0-4 \sigma$ in x and y). Here the HEL minimum and maximum radii are set to 4 and $8 \sigma_x$ respectively. The left plot shows the core, halo and HEL beams, the right plot omits the core for clarity.

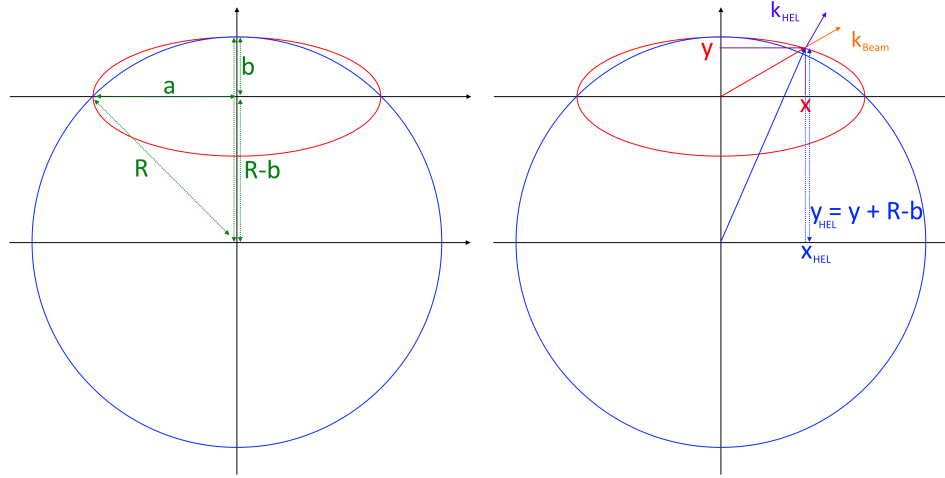


Fig. 22: Diagram of an offset HEL minimum radius (blue), with a non-round beam core envelope (red) where the semi-major is much greater than the semi-minor axis. The left figure illustrates variables used to derive the shifted HEL centre and radii. The right figure shows the co-ordinates and kick in the beam frame and the HEL frame.

$$y_{shift} = y - R_{min(elliptical)} + b. \quad (6)$$

Thus we may use a HEL with minimum radius $R_{min(elliptical)}$, shifted to be centred at (x, y_{shift}) , to maximise the overlap of the round HEL on a non-round accelerator beam where $\beta_x > \beta_y$. In order to set the HEL maximum radius R_{max} we use the fact that the ratio g :

$$g = \frac{R_{max}}{R_{min}}, \quad (7)$$

is a constant that depends only on the hardware (i.e. the cathode geometry), such that:

$$R_{max (elliptical)} = g \cdot R_{min (elliptical)}. \quad (8)$$

We also note from Fig. 22, that $R_{min (elliptical)}$ overlaps with the beam core. This is undesirable, and is mitigated by using a scaling factor of $\sqrt{\frac{a}{b}}$, modifying equation 5:

$$R_{min (elliptical)} = \sqrt{\frac{a}{b}} \cdot \left(\frac{a^2 + b^2}{2b} \right). \quad (9)$$

The resulting ‘matched’ HEL is shown in Fig. 23.

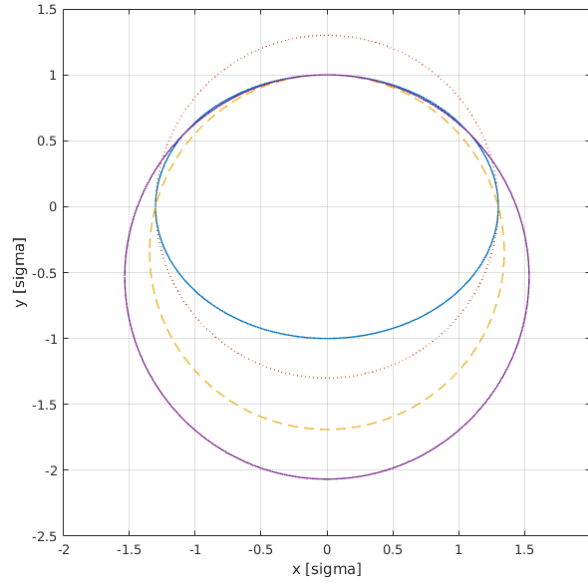


Fig. 23: Beam core at non-round position (blue), original HEL R_{min} (red dotted), offset HEL $R_{min \text{ elliptical}}$ (yellow dashed), and corrected and offset $R_{min \text{ elliptical}}$ (purple), all shown in units of normalised σ where $\sigma_x = 1.3\sigma_y$, and $\sigma_y = \frac{1}{4}$.

In reality the beam footprint is cumulative over many turns, particles trace an ellipse in phase space and thus any space in the phase space ellipse may contain a particle at a given turn. This is simply an approach to improve the performance of the HEL as a collimation enhancer when forced to use it at a position in the accelerator where the beam is non-round.

In order to improve the cleaning enhancement where the beam is not round, the elliptical matching method is used. In both the non-round and oval positions we choose to match the HEL inner radius to the vertical maximum of the bunch in real space. We note that the HEL radii are larger at the oval position because of the need to avoid any overlap with the beam core.

4.6 Elliptical Matching Halo Depletion

Figure 24 compares the use of the elliptical matching at the non-round and oval positions with the diffusive modes at all three integration positions. This method appears to reduce the collimation enhancement, which is not surprising as the non-round and oval positions have beams that are larger in x than y . This

means that the elliptical matching attempts to reconcile coverage of the vertical halo, which reduces the enhancement in the horizontal plane. It is interesting to see that in the non-round position the survival is similar after 10^5 turns, attempting simulations with many more turns could indicate an improvement in collimation enhancement over a longer time period.

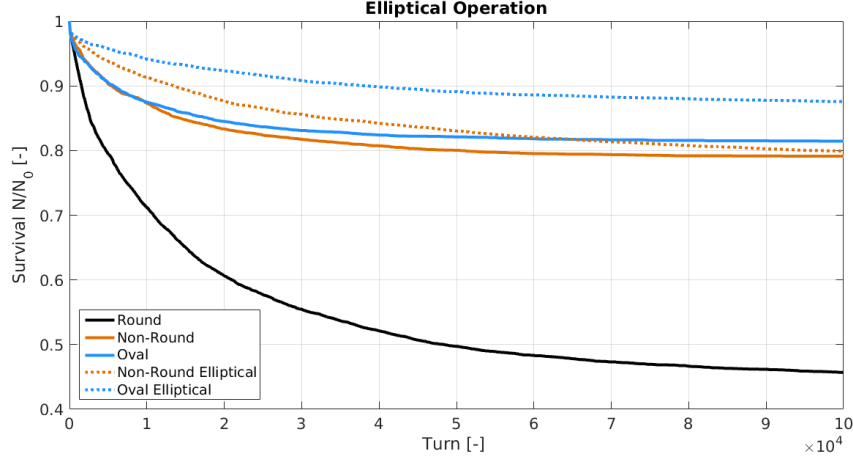


Fig. 24: Survival fraction of a HELHalo bunch populated between 4 - 5.8 σ for 10^5 turns in the HL-LHC using a HEL in diffusive mode with elliptical matching, at the non-round, and oval beam positions. For comparison the survival for a HEL in diffusive mode with no geometrical enhancement at the round, non-round, and oval beam positions are also shown.

The first attempt at dynamic operation, the pogo mode translates the elliptically matched HEL beam such that it alternately touches the top and bottom transverse extremities of the beam core. The pogo operation mode for the non-round and oval positions are compared in Fig. 25. In the case we are interested in, *i.e.* where the beam is larger in x , the pogo operation alternates the elliptical matching between the vertical maxima.

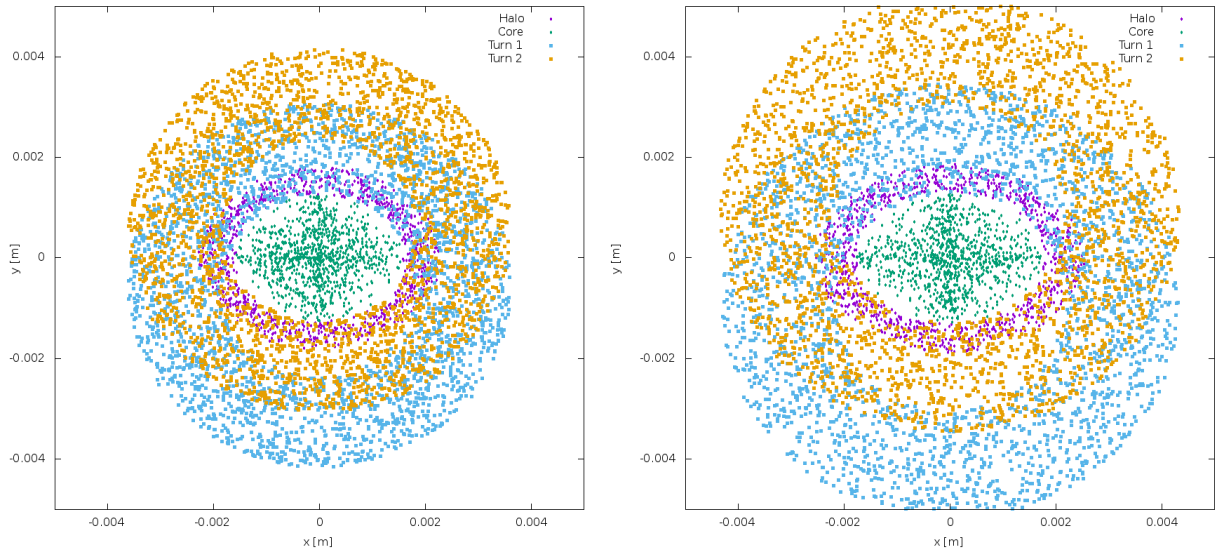


Fig. 25: HEL footprint for pogo operation of the HEL in HL-LHC; blue points indicate the HEL footprint odd turns, yellow points indicate the HEL footprint on even turns, green points are the core protons, purple are the halo protons. The left plot shows the non-round position, the right plot shows the oval position.

Halo survival when using the pogo HEL is shown in Fig. 26 and compared with normal diffusive operation, we observe that this first attempt at dynamic operation offers very similar collimation enhancement to the static elliptical matching. Particles in the beam halo do not maintain their position each turn, they undergo betatron and synchrotron oscillations that cause them to trace an elliptical Poincaré section at the HEL. This means that as well as the particle movement, the HEL is randomly switched on and off, and the HEL may or may not cover the particle in question as it is being translated vertically back and forth. It is likely the vertical translation negates any possible improvement due to increased halo coverage.

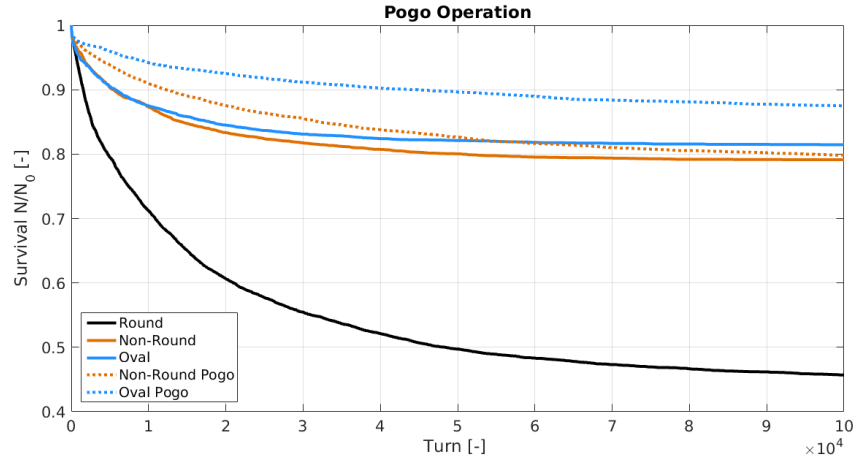


Fig. 26: Survival fraction of a HELHalo bunch populated between 4 - 5.8 σ for 10^5 turns in the HL-LHC using a HEL in diffusive mode with dynamic Pogo matching, at the non-round, and oval beam positions. For comparison the survival for a HEL in diffusive mode with no geometrical enhancement at the round, non-round, and oval beam positions are also shown.

The next approach to improving cleaning enhancement at non-round beam positions when using the HEL, is the hula mode of operation. In this second dynamic HEL operation mode, we choose to elliptically match the HEL to the vertical maxima, and then translate this HEL around the bunch such that the HEL inner radius touches each horizontal and vertical extremity. This is done in the order shown in Fig. 27 in order to imitate a clockwise rotation in real space. This is an approximation as in reality it may not be possible to re-size and translate the HEL beam on a turn by turn basis, it is more likely that many more steps may be required between these maxima, which is likely to reduce the increase in collimation enhancement. Thus we take the hula operation mode as a best case scenario.

The major concern when using the dynamic pogo HEL operation is that the HEL does not overlap sufficiently with the horizontal halo as it attempts to cover the vertical halo. The hula mode is an optimisation of the pogo mode. The result of this operation mode in the HL-LHC is shown in Fig. 28, we observe that hula operation reduces the collimation enhancement. This is likely due to the fact that when the HEL is translated to touch the horizontal extremities of the bunch core, the minimum radius is too large due to the elliptical matching, and halo coverage is reduced.

In order to improve upon the hula operation, the close-hula operation mode maintains the minimum HEL radius as well as translating it around the beam core. We observe from Fig. 29 that this method offers a very small improvement on cleaning enhancement after 10^5 turns at the non-round position, though this is not enough to approach that of the round position.

Table 5 summarises halo survival for novel dynamic operation modes of the HEL in the HL-LHC.

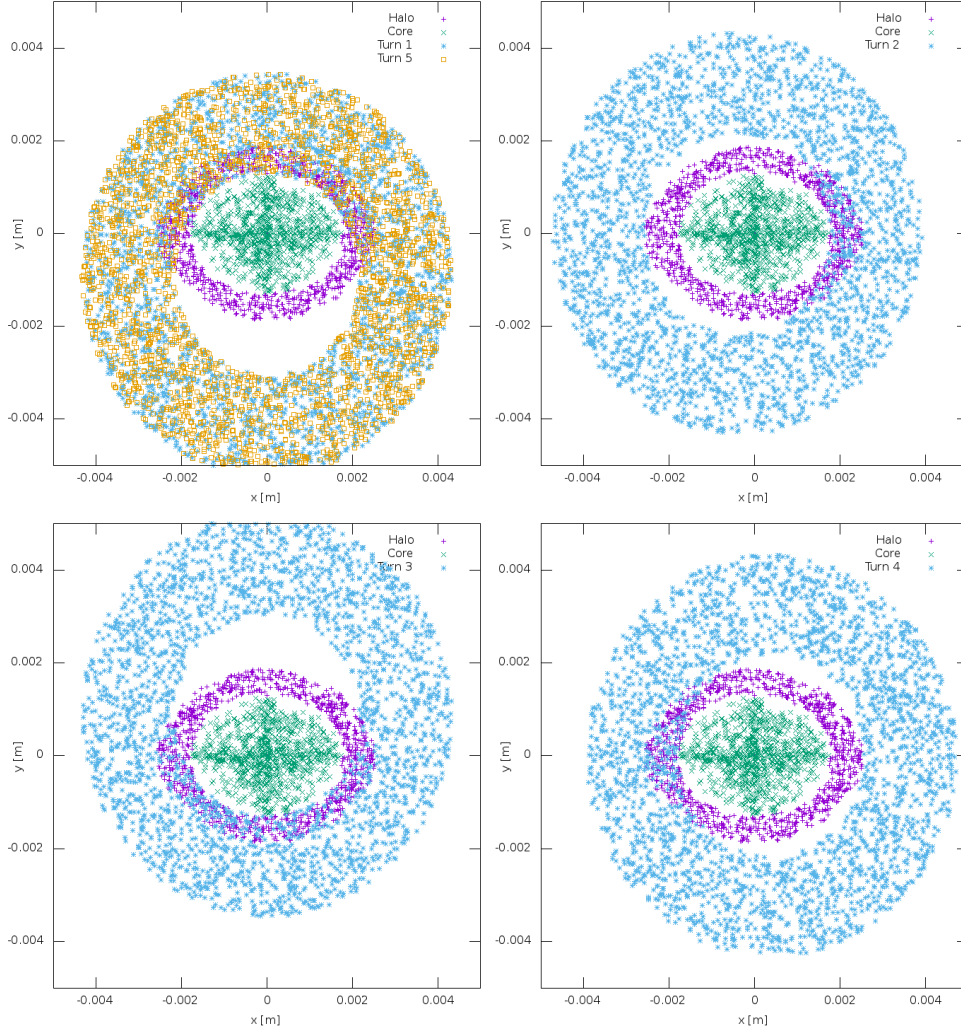


Fig. 27: HEL footprint for hula operation of the oval HEL in the HL-LHC; blue and yellow points indicate the HEL footprint, green points are the core protons, purple are the halo protons. The top left plot shows the first and fifth turn, the top right plot shows the second turn, the bottom left plot shows the third turn, and the bottom right plot shows the fourth turn.

4.7 HEL Parameters

In order to gauge the response, we perform a brief investigation of the main HEL parameters: the active length, electron current, and electron energy.

First we increase only the electron beam energy from 10 KeV to 15 KeV and 20 KeV. The resulting particle survival at the non-round position is shown in Table 6. We observe that increasing the energy decreases the collimation enhancement. To understand this we must recall the kick given by the HEL, equation 4. The electron energy is present in this equation in the form of the normalised energy β_e . A β_e term is present on both the numerator and denominator, however the rigidity term in the denominator is much larger than the remaining numerator. Increasing the electron energy results in a decreased kick. In reality a sufficiently high electron energy must be used in order to reduce the force from the proton beam which could distort the electron beam greatly, thus negating its effectiveness, and likely interfering with the proton beam core. Figure 30 confirms that this reduction occurs at all chose integration positions, and doubling the electron beam energy results in a $\approx 20\%$ decrease halo cleaning after 10^5 turns.

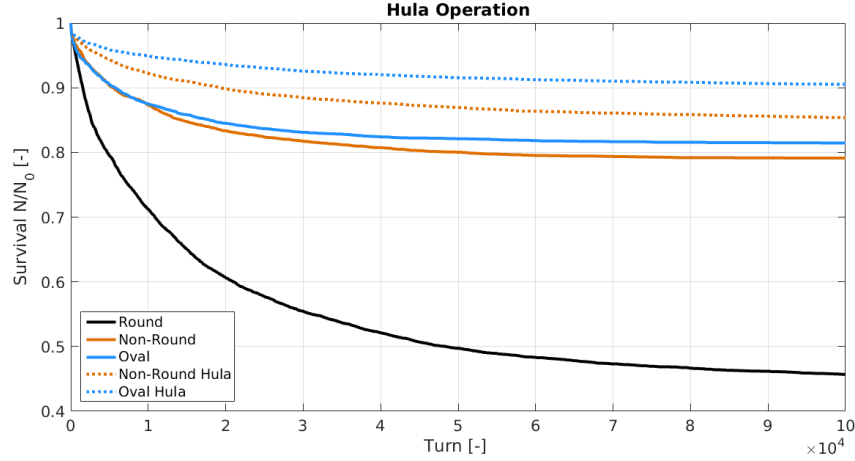


Fig. 28: Survival fraction of a HELHalo bunch populated between $4 - 5.8 \sigma$ for 10^5 turns in the HL-LHC using a HEL in diffusive mode with dynamic Hula matching, at the non-round, and oval beam positions. For comparison the survival for a HEL in diffusive mode with no geometrical enhancement at the round, non-round, and oval beam positions are also shown.

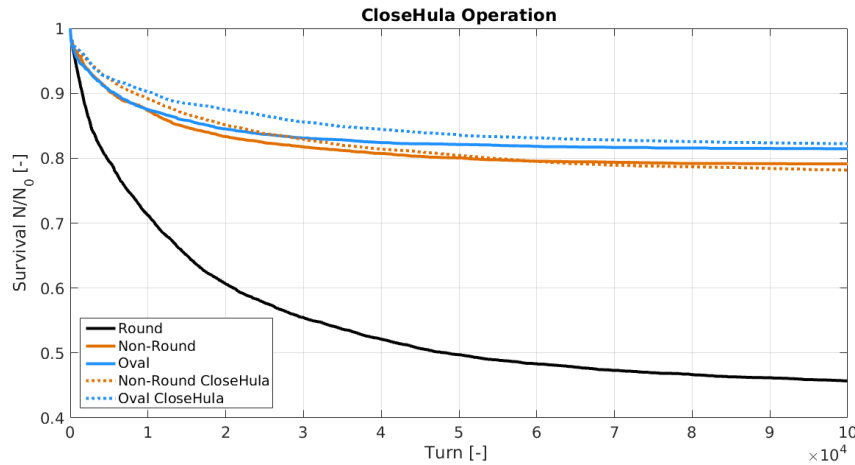


Fig. 29: Survival fraction of a HELHalo bunch populated between $4 - 5.8 \sigma$ for 10^5 turns in the HL-LHC using a HEL in diffusive mode with dynamic CloseHula matching, at the non-round, and oval beam positions. For comparison the survival for a HEL in diffusive mode with no geometrical enhancement at the round, non-round, and oval beam positions are also shown.

Next we increase only the electron beam current. From the kick equation 4 we expect a linear increase in collimation efficiency as the current is increased. The resulting survival is shown in Table 6. This shows the expected behaviour, an increase in HEL beam current results in an increased kick and therefore improved halo removal. This is the highest achieved improvement on collimation enhancement at the non-round position.

Figure 31 shows the survival at the three integration points compared to that when doubling the beam current. We observe the improvement in all cases, but note that the improvement is larger at the oval position when compared to the non-round. This implies that the position of the kick, or rather the phase advance between the HEL and the collimators, is of great importance. Drawing the conclusion that an increase in current equates to an increase in collimation enhancement up to a certain limit is naive.

Operation Mode	Non-Round	Oval
Diffusive	79.15	81.47
Elliptical	79.9	87.61
Pogo	79.76	88.2
Hula	85.37	90.52
Close hula	78.18	82.25

Table 5: Particle survival $\frac{n}{n_0}$ % for an initial halo of 10^4 particles between $4 - 5.8 \sigma$ after 10^5 turns in the HL-LHC with a HEL in the corresponding position and operation mode. The aim is to approach the collimation enhancement shown at the round HEL position, which is 45.69%, it is clear that these dynamic modes do not rectify the decreased halo removal due to the beam not being round.

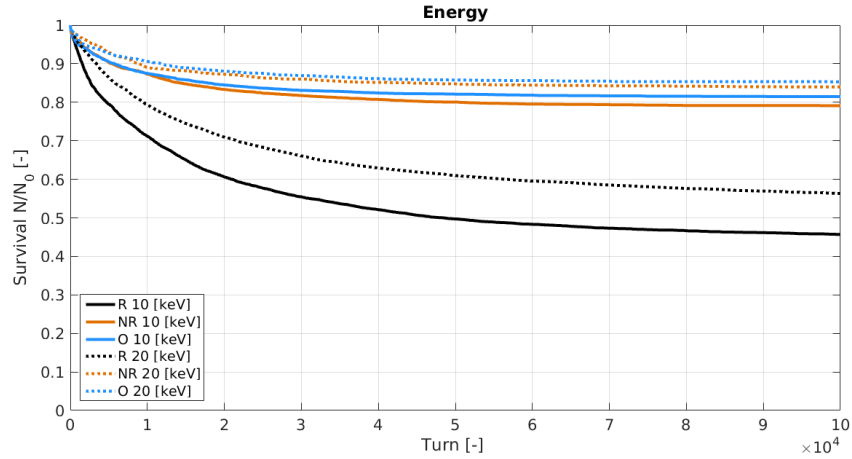


Fig. 30: Survival fraction of a HELHalo bunch populated between $4 - 5.8 \sigma$ for 10^5 turns in the HL-LHC using a HEL in diffusive mode for various HEL energies, at all three HEL positions.

The collimation enhancement is dependent on many factors, and conclusions drawn here may only be valid for the specific simulated set up. For this reason we attempt to draw conclusions on behaviour rather than quantify possible achievable collimation enhancement.

Finally we observe the effect of increasing the active length of the HEL. As expected there is a linear increase in collimation enhancement as the length is increased, as shown in Table 6. The halo cleaning does not approach the limit previously identified in the non-round position. We observe the effect of increasing the active length from 3 m to 5 m at each of the integration points in Fig. 32. In this case the increase in length results in improved halo cleaning similar to that when increasing the current.

We also attempt to increase all three of the HEL parameters that directly impact upon the kick, this is shown in Fig. 33 where a HEL current of 10 A, length of 5 m, and energy of 20 keV is used at all three integration points. We observe that particle survival is more than halved (from 45.69 % to 21.58 %) after 10^5 turns at the round position, whereas the reduction is 17.92 % and 13.14 % at the non-round and oval positions respectively. This shows that any improvement is reduced because of the reduced HEL overlap with the halo at non-round positions.

5 Conclusion

The hollow electron lens has been included in MERLIN as a physics process, building on the elens subroutine from SixTrack. This process has been benchmarked against SixTrack and is in good agreement

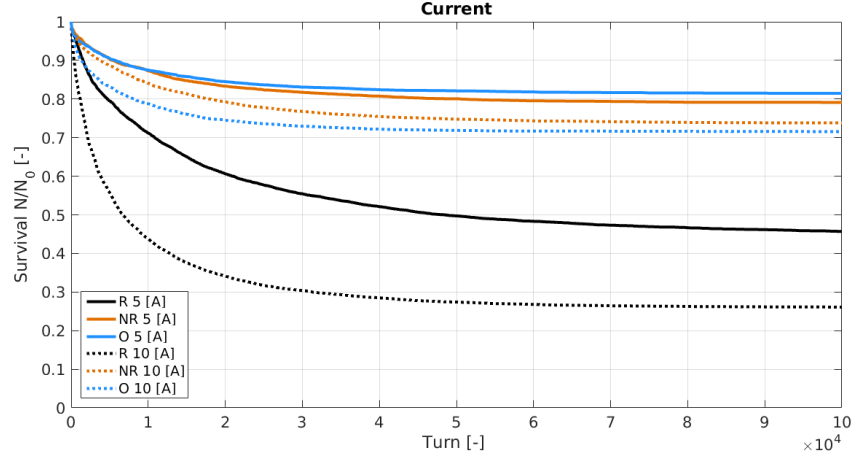


Fig. 31: Survival fraction of a HELHalo bunch populated between $4 - 5.8 \sigma$ for 10^5 turns in the HL-LHC using a HEL in diffusive mode for various HEL currents, at all three HEL positions.

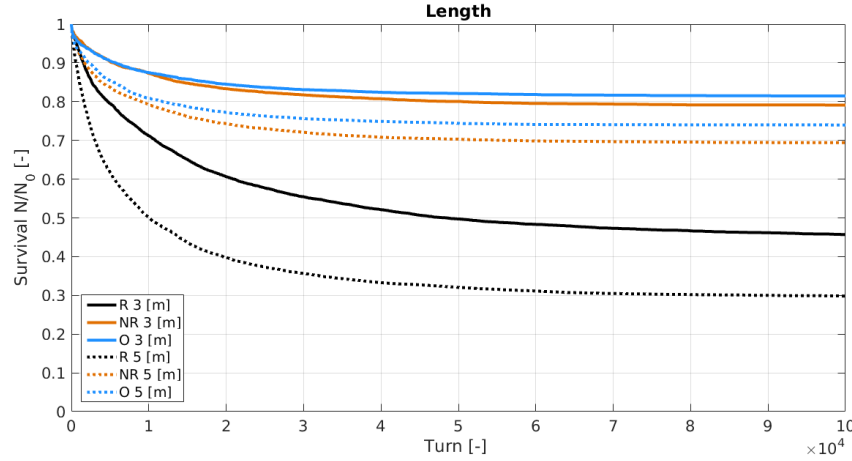


Fig. 32: Survival fraction of a HELHalo bunch populated between $4 - 5.8 \sigma$ for 10^5 turns in the HL-LHC using a HEL in diffusive mode with a length of 5 m at all three HEL positions. For comparison the standard setting of 3 m is shown for all three positions.

when comparing halo survival, though there are small differences in halo behaviour deemed to be due to differences in beam optics being used. Using the Modularity of MERLIN, a number of novel approaches to HEL collimation have been used as test cases to demonstrate the efficacy of MERLIN for collimation, and general accelerator simulation.

The suggested location for the HEL in the HL-LHC offers a proton beam that is not round. Using three integration positions; the suggested one (non-round), and two nearby that offer a round beam, and a less round beam (oval), simulations of halo survival with a HEL in the respective positions have been performed. We have observed the expected reduction in collimation enhancement when integrating the HEL here when compared to a round beam position. Collimation enhancement at the oval position is further reduced, confirming our initial hypothesis that HEL collimation enhancement is reduced as the beam becomes less round.

Upon comparing the current modulating operation modes we observe that the DC mode offers no noticeable collimation enhancement, the (unoptimised) AC mode causes an almost instant cut in the

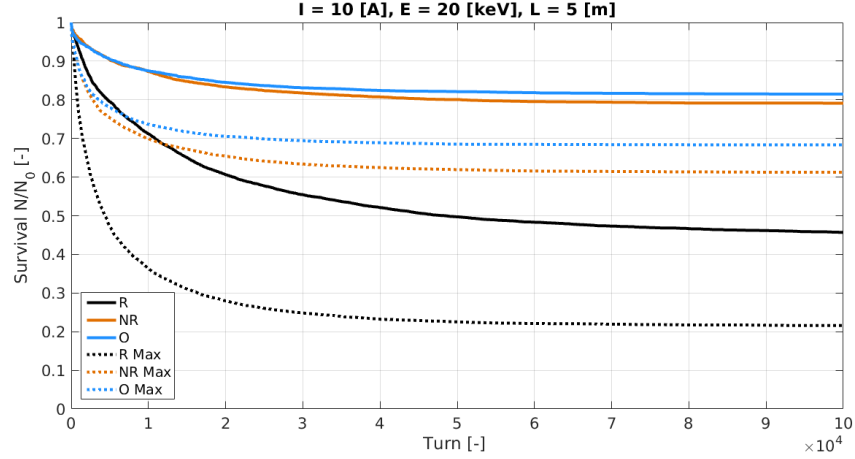


Fig. 33: Survival fraction of a HELHalo bunch populated between 4 - 5.8 σ for 10^5 turns in the HL-LHC using a HEL in diffusive mode with a current of 10 A, length of 5 m, and energy of 20 [keV] at all three HEL positions. For comparison the standard setting of 5 A, 3 m, and 10 keV is shown for all three positions.

Operation Mode	Round	Non-Round	Oval
Diffusive	45.69	79.15	81.47
HEL Active Length			
3 [m]	45.69	79.15	81.47
4 [m]	-	73.8	-
5 [m]	29.83	69.42	73.98
HEL Beam Current			
5 [A]	45.69	79.15	81.47
6 [A]	-	76.14	-
7 [A]	-	72.06	-
8 [A]	-	71.35	-
10 [A]	26.09	66.01	76.12
HEL Beam Energy			
10 [keV]	45.69	79.15	81.47
15 [keV]	-	82.97	-
20 [keV]	56.35	83.68	84.74
5 [m] 10 [A] 20 [keV]	21.58	61.23	68.33

Table 6: Particle survival $\frac{n}{n_o}$ % for an initial halo of 10^4 particles between 4 - 5.8 σ after 10^5 turns in the HL-LHC with a HEL in the corresponding position and operation mode.

halo, and the diffusive mode causes a continuous diffusion of halo particles onto the collimators. This results in 54.31% halo removal after 10^5 turns at the round, 20.85% at the non-round, and 18.53% at the oval positions.

Whereas the diffusive mode appears to be simple as it does not require good knowledge of the tune like the AC mode, from a hardware perspective random modulation of the HEL current requires rapid rise times in power supplies and the cathode, which may be very demanding. The AC mode may be preferable as it uses a relatively gentle modulation. Though the diffusive mode offers the greatest collimation enhancement. We note that the AC mode has not been optimised for these simulations, this may be a useful step in further studies. The requirement for AC optimisation makes the diffusive mode

the obvious choice for an investigation of novel geometrical operation.

A number of novel operation modes are proposed in order to mitigate this reduction in halo cleaning at non-round beam positions. Novel elliptical matching is used to increase coverage of the halo, and novel dynamic operation modes have been combined with the diffusive current modulation in order to attempt an increase in collimation efficiency at the non-round position. Through multiple iterations of dynamic operation the collimation efficiency was increased, however this improvement is small over 10^5 turns and does not compare to that at the round position. Practically an elliptical matching may be possible, but the dynamic modes are less likely, as translation of the HEL on a turn-by-turn basis may be a slow process. The EM field felt by the beam core may be of concern as the HEL is offset from the centre of the beam, this could amplify the effect of any instabilities in the HEL beam.

It is preferable to optimise the HEL properties; active length, and beam current, to which the kick HEL kick on a proton is proportional, rather than use dynamic or geometric operation. When the kick is increased we improve the collimation enhancement. When increasing the HEL beam energy it is observed that the collimation enhancement is reduced, this is due to the kick decreasing.

6 Acknowledgements

This work was performed under the guidance of Stefano Redaelli and Adriana Rossi, in collaboration with Joschka Wagner and Daniele Mirarchi, all of CERN. This research is supported by FP7 HiLumi LHC (Grant agreement 284404).

References

- [1] G. Stancari et al., “Collimation with Hollow Electron Beams”, Phys. Rev. Lett., 107 084802, 2011.
- [2] V. Shiltsev et al., “Experimental demonstration of compensation of beam-beam effects by electron lenses”, Phys. Rev. Lett., 99 244801, 2007.
- [3] V. Shiltsev, “On Possible Use of Electron Lenses in LHC”, FERMILAB-CONF-06-505-AD, 2006.
- [4] Zhang et al., “The Origination and Diagnostics of Uncaptured Beam in the Tevatron and Its Control by Electron Lenses”, Phys. Rev. ST Accel. Beams, 11 051002, 2008.
- [5] V. D. Shiltsev, “Electron Lenses for Particle Colliders”, Springer, 2015.
- [6] “Review of the needs for a hollow e-lens for the HL-LHC”, <https://indico.cern.ch/event/567839>, October 2016.
- [7] V. Previtali et al., “Numerical simulations of a proposed hollow electron beam collimator for the LHC upgrade at CERN”, FERMILAB-TM-2560-APC, 2013.
- [8] M. Fitterer et al., “Implementation of Hollow Electron Lenses in SixTrack and First Simulation Results for the HL-LHC”, in Proc. IPAC’17, Copenhagen, Denmark, May 2017, paper TH-PAB041.
- [9] SixTrack - 6D Tracking Code, <http://sixtrack.web.cern.ch/SixTrack/>.
- [10] The HL-LHC Collaboration, “HL-LHC Preliminary Design Report”, CERN-ACC-2014-0300, 2014.
- [11] G. Stancari et al., “Conceptual design of hollow electron lenses for beam halo control in the Large Hadron Collider”, FERMILAB-TM-2572-APC, 2014.
- [12] H. Rafique, “MERLIN for High Luminosity Large Hadron Collider Collimation”, PhD Thesis, University of Huddersfield, 2016.
- [13] G. Robert-Demolaize, “Design and performance optimisation of the LHC collimation system”, PhD Thesis, Ecole Nationale Supérieure de Physique de Grenoble, 2006.
- [14] D. Mirarchi, “Crystal Collimation for LHC”, PhD Thesis, Imperial College London, 2015.

Simulating off-momentum loss maps using SixTrack

H. Garcia-Morales^{1,2}, R. Bruce², D. Mirarchi², S. Redaelli², B. Salvachua², K.N. Sjobak²

¹Royal Holloway University of London, Egham, United Kingdom

²CERN, Geneva, Switzerland

Abstract

The Large Hadron Collider was designed to collide proton beams with unprecedented energy in order to extend the frontiers of the high-energy physics. The beam is not monochromatic and each particle deviates slightly from the nominal energy. Some particles follow an orbit which differs from the design orbit and, in some cases, they can reach the aperture and may quench the superconducting magnets. Therefore, these particles must be safely removed from the machine before losses in cold sections occur. A dedicated off-momentum collimation to capture these particles. In this paper we present a new set of tools to simulate off-momentum cleaning in order to better understand the dynamics of such particles and to try to improve the off-momentum collimation cleaning efficiency.

Keywords

LHC, collimation, off-momentum, simulations, Sixtrack.

The Large Hadron Collider (LHC) is designed to collide protons up to an energy of 7 TeV with a total stored energy of about 362 MJ per beam. This huge amount of energy presents a serious hazard to the superconducting magnets, which must be adequately protected to ensure reliable operation. Even a small fraction of particles lost in the superconducting aperture could quench the magnet. For this purpose, a multistage collimation system [1] was installed, in order to intercept unavoidable beam losses in a safe way. Two out of the eight Interaction Regions (IRs) of the LHC are devoted to beam collimation (Fig. 1). In IR7, the betatron cleaning section intercepts particles with large amplitudes while in IR3, the momentum cleaning section captures particles with large energy deviations.

Betatron cleaning has been already studied in detail since it is quite straightforward to implement in simulations. On the other hand, momentum cleaning has been studied with less detail due to that

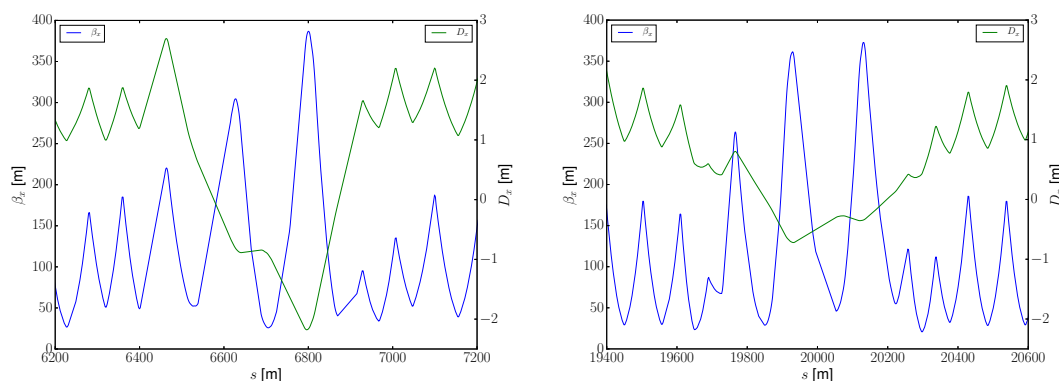


Fig. 1: Optical functions β_x and D_x in both collimation sections: momentum cleaning section (left) and betatron cleaning section (right).

the simulation setup requires a dynamical manipulation of the elements of the machine and tools were not available until recently. In addition, a large number of turns need to be simulated with respect to the betatron cleaning simulations, which makes it more CPU demanding. In this paper, we show how off-momentum simulations can be carried out using SixTrack [3].

1 Betatron cleaning

The efficiency of the betatron cleaning is evaluated by means of the so called loss maps. Betatron loss maps are acquired during LHC regular or dedicated operation by using the electric transverse damper (ADT) to blow up the beam in such a way that the beam tails become more populated and eventually reach the betatron collimator cut. From the loss pattern obtained, we can study the performance of the collimation system. To simulate the betatron cleaning of the LHC collimation system we use SixTrack, which performs a thin lens element-by-element tracking. When a particle enters a collimator, a built-in Monte Carlo code is used to simulate the particle-matter interaction [2]. A particle is considered lost when it interacts inelastically inside a collimator (excluding single diffractive events) or if it hits the aperture. The simulation output contains the coordinates of all loss locations that allow us to reconstruct the loss pattern along the ring and compare it to measurements.

2 Momentum cleaning

The momentum cleaning section in IR3 is designed to intercept particles which deviate enough from the design energy with sufficient margin to the off-momentum aperture bottleneck of the machine to fit an efficient multi-stage system. To do that we take into account that the linear motion of a particle is the solution of the inhomogeneous Hill's equation that has two different contributions: the betatron component and the dispersive term. The horizontal spatial transverse coordinate of single particle, x , is given by,

$$x = x_\beta + x_\delta = \sqrt{\epsilon\beta_x(s)} \cos(\mu(s) - \mu_0) + \delta D(s), \quad (1)$$

where x_β is the pure betatron oscillation given by the β -function at a given s location, the beam emittance ϵ and the phase advance $(\mu(s) - \mu_0)$ and x_δ is the orbit deviation due to the particle energy offset $\delta = \Delta p/p$ and the dispersion D at this location. Particles with large enough energy deviation δ can be lost in the magnet aperture if no dedicated protection system is included. Therefore, in the momentum primary collimator, dispersion is by design relatively large in order to intercept particles with even small $|\delta|$.

Off-momentum loss maps are also acquired during regular or dedicated operation at the LHC. To do so, the RF frequency is trimmed to bring the beam to higher or lower energies towards the primary collimators in IR3. The frequency shift induces a phase shift in the RF voltage seen by the synchronous particle. The synchronous particle is not synchronous any longer for the new RF configuration and starts oscillating around the new bucket. If the frequency trim is applied adiabatically, the phase space from one turn to the next changes slowly enough to always keep the full beam captured the full beam and brings the full distribution towards the collimator cut. The loss pattern recorded during this process gives information about the efficiency of the momentum cleaning. Although off-momentum loss maps are important, they have never been simulated before. In the next sections we explain how these simulations can be implemented now using new options available in SixTrack.

3 RF-trim model

To simulate realistic off-momentum dynamics we need to take into account the longitudinal dynamics in the LHC. The RF frequency trim can be modeled by modifying the longitudinal mapping equations to

take into account a phase shift term $\Delta\varphi$ that is added to every turn to the phase,

$$\delta_{n+1} = \delta_n + \frac{eV}{\beta^2 E} (\sin \phi_n - \sin \phi_s) \quad (2)$$

$$\phi_{n+1} = \phi_n + \Delta\varphi + 2\pi h \eta \delta_{n+1} \quad (3)$$

where η the slip factor, h the harmonic number, $\delta = \frac{\Delta p}{p}$ is the energy error of the particle, β is the particle speed in units of c , E is the energy, ϕ is the phase coordinate of the particle and ϕ_s is the synchronous phase that we take equal to zero for simplicity. For small shift per turn, this formalism is suitable to model adiabatic RF trims. This adiabatic phase shift is given by a quadratic function of time in order to represent a linear shift in the frequency using the simple relation,

$$\varphi = \omega t \rightarrow \varphi + \Delta\varphi = (\omega + \Delta\omega)t \quad (4)$$

$$\Delta\omega \propto t \Rightarrow \Delta\varphi \propto t^2. \quad (5)$$

The exact coefficient of the second order term is given by the maximum phase shift $\Delta\varphi_{\max}$ and the number of turns N_{turns} in which the trim will be applied. For instance, an RF frequency shift $\Delta\omega$ applied in two consecutive turns gives a phase shift $\Delta\varphi$ of,

$$\Delta\varphi(t = \tau) = \Delta\omega_{\text{RF}}\tau = \Delta\omega_{\text{RF}} \frac{2\pi}{\omega_0}. \quad (6)$$

where ω_{RF} is the angular frequency of the RF and ω_0 is the angular frequency of the particles around the ring. We can express it in terms of $f = \frac{\omega}{2\pi}$ and express the phase shift as a function of the frequency trim as,

$$\Delta\varphi = 2\pi \frac{\Delta f_{\text{RF}}}{f_0} \quad (7)$$

If one wants to apply this shift repeatedly during several turns, as it is the case for an adiabatic RF trim, an expression of the phases shift as a function of the number of turns is required. If we consider the function in Eq. (2) a quadratic increase of the phase shift is achieved by a linear model given by,

$$\Delta\varphi_{\text{map}}(n) = \frac{\Delta\varphi_{\max}}{N_{\text{turns}}} n, \quad (8)$$

where $\Delta\varphi_{\max}$ is the maximum phase shift given by the total frequency shift and N_{turns} is the total number of turns during the RF trim. However if one wants to apply the phase shift straight to the RF cavity phase where the reference phase stays constant an actual quadratic model is needed in order to take into account the cumulative behavior of the shift,

$$\Delta\varphi_{\text{cav}}(n) = \frac{\Delta\varphi_{\max}}{2N_{\text{turns}}} n^2. \quad (9)$$

Equation (9) is used in tracking simulations to dynamically change the RF cavity frequency as it is explained in following sections.

4 Off-momentum cleaning simulations

The RF trim is simulated trim using SixTrack tracking code with a collimation module to take into account impacts on the magnet aperture and the hits in the collimators. The next sections describe how to prepare the SixTrack input (`fort.2` and `fort.3`) to perform off-momentum cleaning simulations.

4.1 Lattice input (`fort.2`)

The lattice input file `fort.2` must contain the RF cavities. To include them, one must add the option `cavall` to the `sixtrack` conversion command in the MADX file. Once the file `fort.2` is generated, the voltage value must be set to a non-zero value. Otherwise SixTrack ignores the element.

4.2 Configuration input (fort.3)

DYNK [4, 5] is a module of the SixTrack code that allows the user to dynamically change some of the parameters of the simulation such as quadrupole strengths, RF cavity phase and voltage among many others. In this case the DYNK module is used to simulate the RF cavity phase shift. DYNK input requires to specify the parameter to be modified and its value at each turn. Although the DYNK module contains built-in functions, a simple script to generate the input phase for the simulations has been developed where the main input is the maximum frequency shift f_{\max} that one wants to apply and the number of turns required to go from 0 to the f_{\max} . As a typical example, in Fig. 2 a frequency trim of +500 Hz has been applied for 10000 turns (around 1 second) and the resulting phase to be applied to the cavities is shown as a function of the turn number. As an example, the following lines describe the DYNK module to be incorporated in the `fort.3` file to apply the effective RF trim to the cavities and to simulate the off-momentum loss map. The rest of the settings of the file are the same used for standard betatron cleaning simulations. The initial beam distribution instead of the typical annular halo distribution, is that of a nominal bunch with a Gaussian distribution in dp/p with $\sigma_{dp/p} = 1.14 \cdot 10^{-4}$ standard deviation.

```
DYNK
FUN volt1 CONST 2.0
FUN harm0 CONST 35640.0
FUN lag1 FILELIN lag_values.dat
SET acsca.d5l4.b1 voltage volt1 1 10000 0
SET acsca.d5l4.b1 harmonic harm0 1 10000 0
SET acsca.d5l4.b1 lag_angle lag1 1 10000 0
NEXT
```

In the second and third line we define the constant voltage and the harmonic number of the cavity. In the fourth line we call the file `lag_values.dat` which contains the values of the cavity phase at each turn of the simulation represented in Fig.2. Since the calculation of the phase value is relatively complex, we use an external script to generate the input file and to have full control of the output. Nevertheless, we can define different functions within the DYNK module to perform the same calculation.

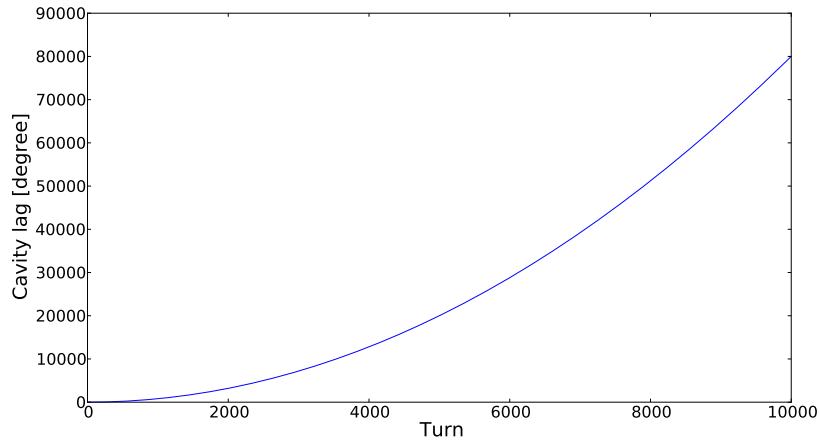


Fig. 2: Example of the phase applied to the RF cavity in the SixTrack simulation for 10^4 turns

4.3 Full simulation

It is important to note that off-momentum loss maps simulations are carried out separately for B1 and B2 while it is usual to have both beams in the machine when off-momentum loss maps are acquired. There-

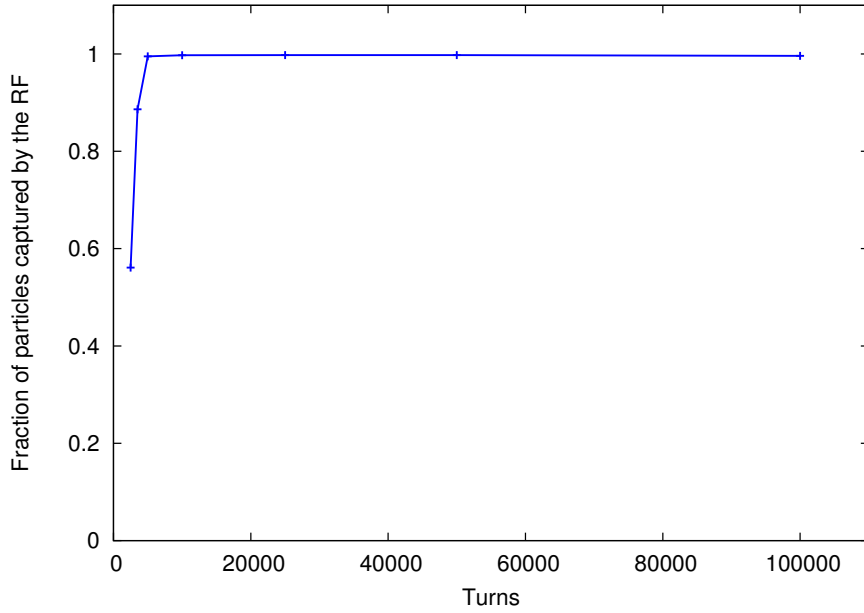


Fig. 3: Fraction of particles captured by the RF bucket and lost in collimators or aperture after RF frequency trim simulation as a function of number of turns simulated. For low number of turns the RF bucket moves fast and some particles of the bunch jump out of the bucket.

fore, to have a full representation of the off-momentum loss maps for comparison with the measured data, one must simulate both beams and combine the result.

5 Simulation speed

The off-momentum loss map acquisition in the LHC takes about 15 seconds. During this time the protons in the LHC perform more than $1.5 \cdot 10^5$ turns. If one wants to simulate the full procedure a large amount of CPU time and disk space is required. In order to reduce the simulation time simulations we take just 10^4 turns, this is less than one second real machine time and the current maximum number of turns allowed by the software. We risk to spoil the results due to the high speed of the frequency trim. To check that this speed reduction does not affect the final result of the loss pattern, different simulations using different number of turns have been performed.

In Fig. 3 the total number of particles lost in the aperture or collimators is plotted as a function of the total number of turns of the RF frequency trim. A clear reduction of the number of lost particles is observed for fast simulations i.e. for a frequency trim with performed during a small number of turns.

6 Benchmarking simulations

A simulated off-momentum loss map for the 2016 LHC optics is shown in Fig. 4 for B1 and B2. The frequency trim is -500 Hz for 10^4 turns. These simulations have been benchmarked by comparing the results with the acquired off-momentum loss maps during LHC operation as can be seen in the results published in [6]. Main losses occur in IR3 primary collimator as expected. The ratio between IR3 losses and IR7 losses is also in agreement with the measurements. One can also observe the losses in the DS around IR3 and some losses in IR4 which were also observed in the machine and reported in [7].

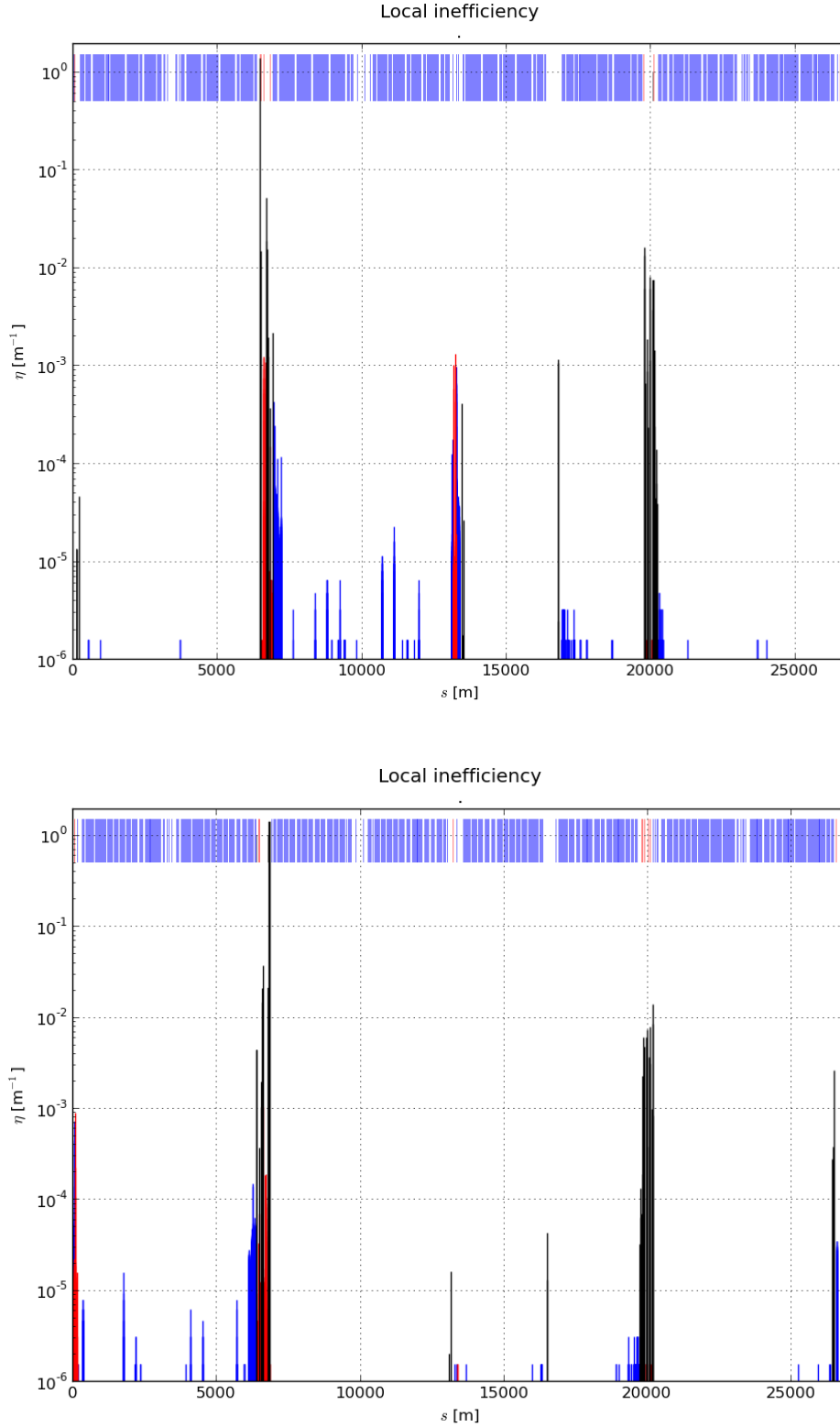


Fig. 4: Simulated off-momentum loss map for B1 (top) and B2 (top) with a -500 Hz frequency trim for 10^4 turns

7 Conclusions

A novel set of simulations of the off-momentum cleaning in the LHC have been presented. With these new implementation we are able to simulate the momentum cleaning not only of the LHC but also of

any other machine such as HL-LHC and FCC-hh. These simulations would allow us to better optimize the momentum cleaning insertion of any hadron collider as well as to establish the basis for other future simulations related to off-momentum dynamics.

References

- [1] R. Bruce et al., "Simulations and measurements of beam loss patterns at the CERN Large Hadron Collider", Phys. Rev. STAB17, 081004 (2014).
- [2] G. Demolaize *et al.*, "A New Version of SixTrack with Collimation and Aperture interfaces", PAC05 Proc., USA, (2005).
- [3] G. Ripken and F. Schmidt, "A symplectic six-dimensional thin-lens formalism for tracking", DESY 95-63 and CERN/SL/95-12(AP), (1995).
- [4] K. Sjobak, H. Burkhardt, R. De Maria, A. Mereghetti, and A. Santamaría García, "General functionality for turn-dependent element properties in SixTrack", IPAC'15, MOPJE069 (2015).
- [5] K. Sjobak, "Dynamics simulations in SixTrack", these proceedings.
- [6] H. Garcia, "Off-momentum loss maps with one beam", CERN-ACC-NOTE-2016-0011, (2016)
- [7] H. Garcia, "Modelling beam halos", Collimation tracking workshop, (2015)

First simulations of collimation cleaning performance for the FCC-hh

*M. Fiascaris**, *R. Bruce*, *S. Redaelli*
CERN, Geneva, Switzerland

Abstract

In the context of the Future Circular Collider Study (FCC), the FCC-hh is designed to collide protons at the unprecedented energy of 50 TeV and a total stored energy of 8500 MJ per beam. This puts extremely tight requirements on the collimation system to efficiently intercept unavoidable beam losses. In this paper the first FCC-hh cleaning simulations at injection and collision energy are presented. The collimation layout includes betatron and momentum cleaning, as well as dump protection collimators and collimators in the experimental insertions for protection of the inner triplets. The simulation setup and the developments that were necessary to achieve these first simulation results are described. The performance of the system is then assessed through the analysis of complete loss maps for a perfect machine.

Keywords

FCC-hh; collimation; tracking simulations.

1 Introduction

The hadron-hadron option of the Future Circular Collider study, FCC-hh [1], is designed to provide pp collisions at a centre of mass energy of 100 TeV. For a nominal beam intensity of 10^{15} protons, the total stored energy per beam will be about 8500 MJ, a factor of 20 above that of the LHC. The high energies and intensities required at the FCC-hh pose daunting challenges on the control of beam losses. Its beams are highly destructive: uncontrolled losses of even just a small fraction of the beam can cause a magnet to quench or damage to accelerator components.

A collimation system is needed to intercept and safely dispose of unavoidable beam losses. Its main functionalities are: efficient cleaning of the beam halo ensuring an operation safely below quench limits at injection and top energy, passive protection of the machine aperture against abnormal losses and minimisation of the halo-induced experimental backgrounds. In this paper, only beam halo cleaning is considered since this is expected to be the driving constraint for the FCC-hh collimation system. At the LHC this is achieved through a multi-stage cleaning [2] with two dedicated insertions for betatron and momentum collimation. Given the excellent performance of the LHC collimation system, validated up to energies of 6.5 TeV and stored beam energy of about 250 MJ, the first conceptual solution for the FCC-hh collimation is a scaled-up system derived from the present LHC. This conservative, but solid approach allow us to evaluate the performance that we can achieve with the current state-of-the-art and to start a mechanical design based on a first assessment of powerloads. Further improvements and technological developments will be considered at a later stage.

In this paper, we present the first simulation setup developed to study the cleaning performance of the FCC-hh collimation system. After a description of the system layout and optics, the parameters used to define the system cleaning performance are described. The tracking tools used to achieve the simulations are then presented, with emphasis on the developments needed to adapt the existing tools to the FCC-hh case. Finally, the results of detailed particle tracking simulations are shown and the performance of the system at injection and collision energy is assessed through the analysis of proton loss maps.

*maria.fiascaris@cern.ch

2 Baseline for the FCC-hh collimation system

In the baseline FCC-hh layout [3], described in Table 1 and shown in Fig. 1 [4], momentum and betatron cleaning are performed in two separate insertions. The so called Extended Straight Section D (ESS-D) is dedicated to the beam extraction followed by betatron collimation on one beam (B1), while momentum collimation is performed on the counter-rotating beam (B2). Similarly, in the ESS-J momentum cleaning is performed on B1 and extraction followed by betatron collimation on B2. In the rest of this paper we will consider B1 only, therefore it is assumed that betatron cleaning is implemented in ESS-D and momentum cleaning in ESS-J.

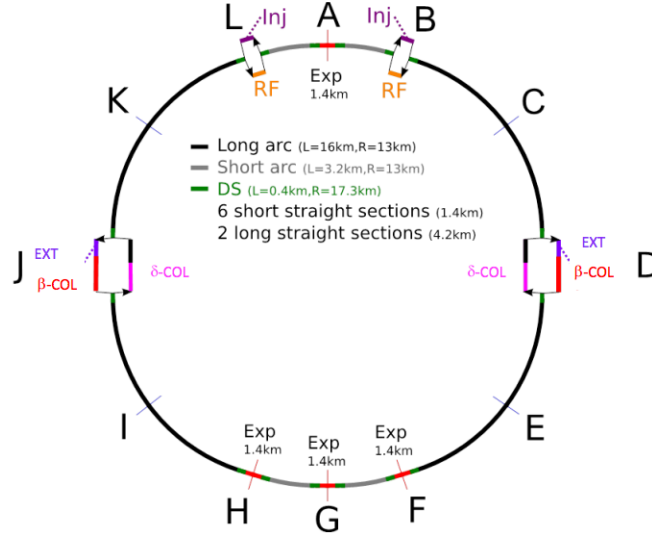


Fig. 1: Schematic layout of the FCC-hh with the configuration of the extraction, momentum and betatron collimation in the extended straight sections D and J.

Table 1: Parameters of the FCC-hh ring. The lattice v7 [5] was used. The scale factor k for the cleaning insertions refers to the factor used to scale the betatron functions from the LHC to the FCC-hh. For the interaction region, the ultimate β^* value at 50 TeV energy is indicated.

Parameter	Value
Circumference	100.171 km
Betatron cleaning insertion:	
Scale factor k	5
Length	2.7 km
Momentum cleaning insertion:	
Scale factor k	2.7
Length	1.4 km
Interaction regions:	
β^*	0.3 m
L^*	45 m
Normalised emittance	2.2 μm

The optics functions for betatron cleaning are shown in Fig. 2 (left). They are similar to the ones used in LHC, except that they are scaled up by a factor $k = 5$, resulting in an insertion length of 2.7 km. The scaling factor was chosen to achieve collimator gaps that are similar to the LHC ones both in units of the beam standard deviation σ and mm, in order to avoid excessive impedance and to guarantee

mechanical stability while keeping the σ -setting small enough to protect the aperture. The number of collimators and their phase advances are the same as in the LHC and were optimised for three-stage cleaning [6]. Primary collimators (TCP), closest to the beam, intercept primary proton losses and give rise to a secondary halo that is intercepted by secondary collimators (TCS). Active absorbers (TCLA), placed outside TCS, catch showers from upstream collimators.

A similar three stage cleaning is installed in ESS-J for momentum collimation, with the difference that the horizontal dispersion in ESS-J is much higher than in ESS-D (see Fig. 2 right). The dispersion functions of the momentum cleaning insertion are the same as at the LHC. The betatron functions are scaled with a factor $k = \sqrt{50/7} = 2.7$, derived from the ratio of the centre-of-mass energies of the FCC-hh to the LHC. This scaling factor is smaller than that used for the betatron cleaning insertion allowing to shorten the length to 1.4 km. Constraints from impedance and mechanical stability are in fact less tight in the momentum cleaning insertion because primary collimators for momentum cleaning can be kept more opened in σ than the ones for betatron cleaning. The momentum cleaning system is designed to capture losses only in the horizontal plane, while the betatron cleaning system ensures coverage in the whole transverse space.

In addition, tertiary collimators (TCT) are installed in the low-beta insertions, about 220 m upstream the interaction point, to provide local protection of the inner triplets. At the beam extraction, a dump protection collimator (TCDQ) is installed as a protection against miskicked beams in the case of extraction failures.

The same collimator jaw materials and lengths as at the LHC are assumed. These are summarised in Table 3. It is planned to modify the collimator materials in the future, based on the outcome of new material studies that are ongoing in the context of the HL-LHC upgrade [7]. Further improvements of the system, such as addition of more collimators and optimisation of phase advances are also foreseen.

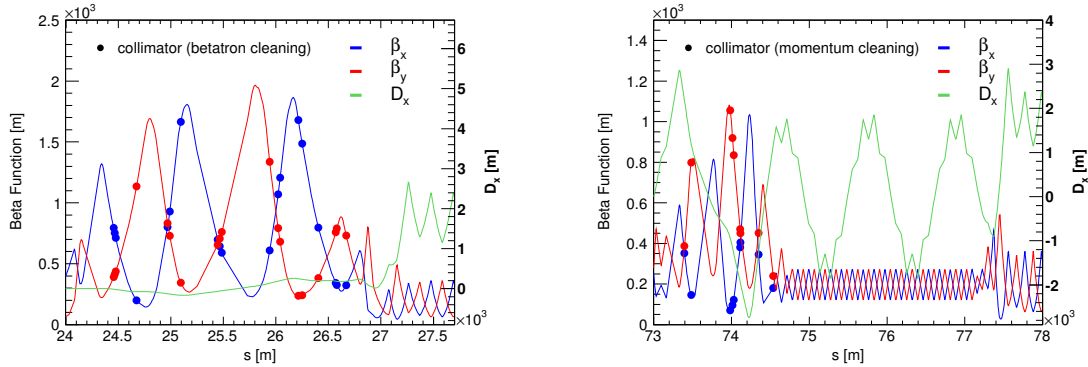


Fig. 2: Optical functions for the betatron (left) and momentum (right) cleaning insertions. The markers show the location of collimators.

3 Simulation of collimation performance

3.1 Cleaning performance

The performance of the collimation system is described by the cleaning inefficiency η_c , defined as the ratio of surviving protons above a given amplitude A_0 to the number of protons absorbed by the collimation system (N_{abs}):

$$\eta_c(A_0) = \frac{N_p(A > A_0)}{N_{\text{abs}}}. \quad (1)$$

Losses are distributed longitudinally over some dilution length. It is therefore useful to define a local cleaning inefficiency $\tilde{\eta}_c$ which is a function of the longitudinal coordinate s and has units of 1/m. Detailed

tracking simulations are used to predict $\tilde{\eta}_c$ along the accelerator with a resolution determined by the bin size Δs :

$$\tilde{\eta}_c(s) = \frac{1}{\Delta s} \cdot \frac{N_{\text{loss}}(s \rightarrow s + \Delta s)}{N_{\text{abs}}}, \quad (2)$$

where $N_{\text{loss}}(s \rightarrow s + \Delta s)$ is the number of particles lost in the aperture between position s and $s + \Delta s$.

The key parameters that determine the target performance of a collimation system are the beam intensity N_{tot} , the minimum allowed beam lifetime τ_b and the quench level R_q . These are related to collimation cleaning inefficiency by the following equation, which describes the condition for operating the machine ensuring that losses in the cold magnets remain below the quench limit:

$$\frac{N_{\text{tot}}}{\tau_b} \times \tilde{\eta}_c < R_q. \quad (3)$$

The relevant parameters for the FCC-hh are summarised in Table 2. The minimum allowed beam lifetimes at injection and top energy are assumed to be the same as at the LHC. Quench limits for the FCC-hh are not yet available. However using simple scaling assumptions from the LHC value of $R_q = 7.6 \times 10^6$ p/m/s at 7 TeV [8], the quench level for the FCC-hh at 50 TeV was estimated at $R_q = 0.5 \times 10^6$ p/m/s [9]. Substituting this values into Eqn. 3 and assuming $\tau_b = 0.2$ h, the required cleaning inefficiency at 50 TeV is calculated to be $3 \times 10^{-7} \text{ m}^{-1}$. In the absence of more accurate estimates, this target value will be taken as a benchmark to asses to the performance of the system in Sec. 5.2. Given that the performance requirements are most stringent at collision energy, a solution for 50 TeV energy also fulfills the requirements at injection energy.

Table 2: Beam parameters and assumptions on the minimum allowed beam lifetime for the FCC-hh at injection and collision energy.

FCC-hh	Injection	Collision
Beam energy [TeV]	3.3	50
Beam intensity (N_{tot})	$1 \cdot 10^{15}$	$1 \cdot 10^{15}$
Stored energy [MJ]	560	8500
Assumed minimum beam lifetime (τ_b)	0.1 h	0.2 h
Power load ($t = \tau_b$) [MW]	1.6	11.8

3.2 Simulation tools for FCC-hh cleaning studies

The cleaning performance of the system was assessed with simulations performed with SixTrack [11–15], a six-dimensional symplectic tracking code. When a particles hits a collimator, a Monte Carlo scattering routine simulates the particle-matter interactions, including Multiple Coulomb scattering, Rutherford scattering, ionisation, as well as point-like interactions such as nuclear elastic, inelastic and single diffractive interactions. The cross-sections of the scattering routine were recently updated taking into account experimental data from the LHC [17]. Since the process of interest is the impact of a high energetic proton on a fixed target (the collimator jaw), the centre-of-mass energy under study is $\sqrt{s} = 306$ GeV for a 50 TeV beam, only 2.7 times higher than at the LHC collision energy. It is therefore expected that the extrapolation to the FCC-hh energies does not significantly affect the reliability of the simulations. In order to validate this assumption, different tracking codes were compared and the level of agreement was found to be satisfactory [18, 19].

The tracking code records the trajectories of halo particles along the accelerators, which are then used as input to an aperture programme [20] to predict the longitudinal distribution of losses, producing so called loss maps. This aperture programme interpolates the saved trajectories and finds the location

where a given trajectory intercepts the machine aperture with a resolution of 10 cm. For the FCC ring, this effectively implies checking the transverse position of each particles for 1 million points.

The SixTrack code and aperture programme were developed for LHC studies. Several developments were needed to achieve the cleaning performance simulations presented in this paper. The existing code had to be adapted to account for the longer ring and the large number of elements in the accelerator structure. Furthermore a first aperture module for the FCC-hh had to be defined [21]. The geometrical aperture of the arc was given by the beam screen, assumed to be of rect-ellipse shape and dimensions 2×15 mm (width) and 2×13.2 mm (height). In the collimation insertion and matching sections, the same mechanical aperture as in the LHC was assumed. The mechanical aperture in the experimental insertion was designed as described in [22].

A major challenge for the FCC-hh simulations was the computing performance. The size of the ring resulted not only in much higher CPU usage, but also in very large size of the intermediate files containing the particle trajectories used for the aperture checks. As a result, the simulations had to be split into a large number of small size jobs, each job tracking about 2 thousands particles. In order to maximize the efficiency of the simulations, the initial particle distribution was chosen such that the majority of particles impacted the primary collimator at the first turn. Details of the simulation parameters and the initial distributions are described in Section 5.

4 Baseline collimator settings

Collimator settings have to be defined in a way that ensures protection of the minimum machine aperture with sufficient margin. Furthermore, a strict hierarchy between collimator families must be respected for optimal cleaning performance and machine protection.

By design at collision energy the aperture limitation typically lies in or close to the inner triplet, since at this location the available aperture is reduced by the large beta functions required to achieve small beam sizes at the interaction points and by the crossing and separation schemes. This constrains the opening of the tertiary collimators (TCT). In the present LHC system, TCTs are not robust and have to be placed outside protection devices, such as the TCDQ, which in turn should be at larger aperture than TCPs and TCSs. For FCC-hh, we define settings in a similar way to minimise the risk of major losses close to the experiments. At injection energy the aperture bottleneck is instead in the arc, which has to be protected by the betatron and momentum cleaning systems, while the TCTs can be more open.

In Table 3 we present the baseline collimator settings for the FCC-hh at injection and collision energy that provide a minimum protected aperture of 15.5σ . These settings correspond to the HL-LHC baseline settings [24] scaled to the FCC-hh normalised emittance of $2.2 \mu\text{m}$ and result in collimator gaps (in mm) that are comparable to the LHC ones.

Detailed aperture studies that take into account optical and mechanical tolerances will be necessary to validate and, if necessary update, the settings at injection and collision energy. In the meanwhile, this first baseline for collimator settings represent a realistic scenario and allows for a first evaluation of the system performance.

5 Tracking simulations

For the simulations presented in this paper, the initial particle distribution was a direct halo [15] at 7.2σ and a thickness $\delta\sigma = 0.06$ in the horizontal plane, a normal distribution cut at 3σ in the vertical plane and no energy errors. The average impact parameter b on the TCP was about $16 \mu\text{m}$ and $4 \mu\text{m}$ at injection and collision energy, respectively. The effect of the assumed initial distribution was assessed by varying the impact parameter b on the primary collimator between $1 \mu\text{m}$ and $12 \mu\text{m}$. The cleaning inefficiency was found to change by about 20% between the extreme values of b .

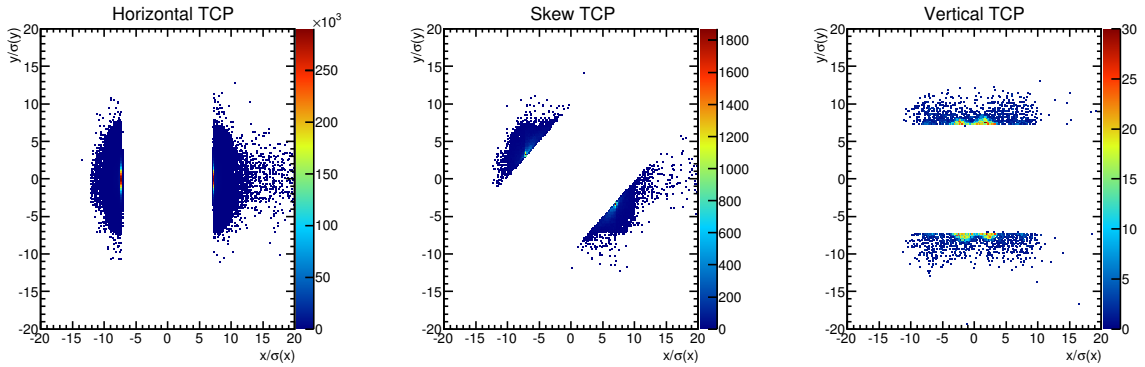
Large statistics is needed to reach a precision level for loss spikes which is below the assumed

Table 3: Baseline collimator settings at injection (3.3 TeV) and collision (50 TeV) energy, expressed in units of σ for a normalised emittance of $2.2 \mu\text{m}$.

Description	Name	Length [m]	Material	Settings [σ]	
				Injection (3.3 TeV)	Collision (50 TeV)
Betatron cleaning	TCP	0.6	C	7.2	7.2
	TCS	1.0	C	9.7	9.7
	TCLA	1.0	W	12.0	12.0
Momentum cleaning	TCP	0.6	C	11.0	21.4
	TCS	1.0	C	12.6	25.2
	TCLA	1.0	W	16.0	27.7
Extraction	TCDQ	9.0	C	11.4	11.4
Tertiaries	TCT	1.0	W	30	13.7

quench levels. Therefore, for the simulations presented here, 8.9 million particles were tracked for 200 turns for the case of a perfect machine and using collimator settings from Table 3.

Figure 3 shows the distribution of inelastic impacts (including nuclear inelastic scattering and single diffractive events) on the three primary collimators of the betatron cleaning insertion ESS-D. This is an example of distributions that can be used as input for further studies of detailed energy deposition.

**Fig. 3:** Distribution of inelastic impacts on the three primary collimators of the betatron cleaning insertion ESS-D.

5.1 Cleaning inefficiency

A first assessment of the collimation system performance can be made studying the cleaning inefficiency η_c , as explained in Sec. 3. η_c has the advantage of being independent of the longitudinal coordinate s and can be studied without an aperture model. Eqn. 1 defines η_c as a function of the radial amplitude A_0 . Similarly, to study the off-momentum population, we can define a cleaning inefficiency as a function of the relative momentum deviation $\delta p/p$, by substituting into Eqn. 1 the amplitude A with $\delta p/p$. The cleaning inefficiency of the system is shown in Figure 4 at both injection and collision energy. At an aperture of 15.5σ , corresponding to the minimum protected aperture for the baseline collimator settings, the cleaning inefficiency is below 10^{-4} , while at $\delta p/p = 0.7\%$, corresponding to the momentum acceptance of the arc, the cleaning inefficiency is below 10^{-6} .

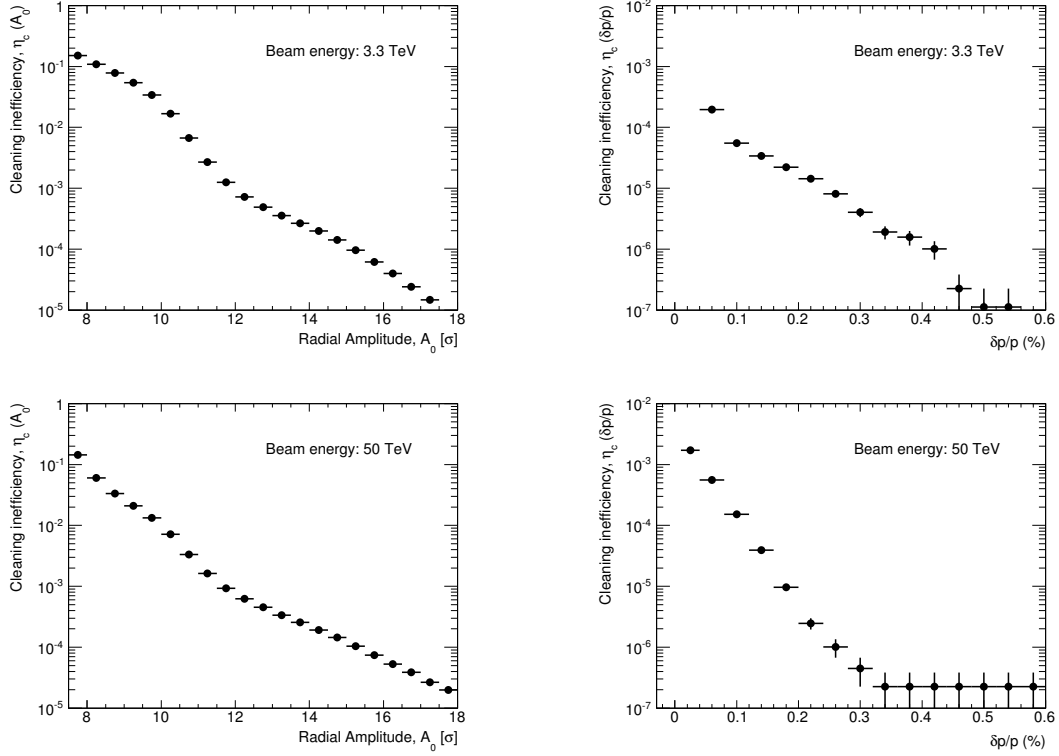


Fig. 4: Cleaning inefficiency as a function of radial amplitude A_0 (left) [23] and momentum off-set $\delta p/p$ (right) at injection (top) and collision (bottom) energy. The horizontal error bars represent the bin width, while the vertical ones are the statistical errors on the efficiency and are calculated using the binomial approach [16] as $\sqrt{\eta_c(1 - \eta_c)/N_{abs}}$, where η_c and N_{abs} are defined in Eqn. 1.

5.2 Loss maps for a perfect machine

Loss maps allow to study the local cleaning inefficiency $\tilde{\eta}_c$, that is how particles escaping the collimation system are lost longitudinally around the ring. Horizontal loss maps are shown in Fig. 5 and Fig. 6 [23] for injection and collision energy, respectively, for a resolution of 10 cm in s . A zoom is shown in locations with large losses: the betatron and the momentum cleaning insertions and one of the two interaction points. It should be noticed that the loss maps shown in this paper consider betatron losses only and the only off-momentum component arises from interactions with the collimators. As a result losses in the betatron cleaning insertion are much larger than in the momentum cleaning insertion. Complementary studies of off-momentum losses are ongoing and are not treated here.

The most significant cold losses occur in the dispersion suppressor (DS) downstream of the cleaning insertions, where two clusters of losses are observed in correspondence to local peaks in the dispersion function. The peak cleaning inefficiency is reached in the second cluster of ESS-D with a value of $(1.1 \pm 0.4) \times 10^{-5} \text{ m}^{-1}$ and $(3.7 \pm 0.6) \times 10^{-5} \text{ m}^{-1}$, at injection and collision energy, respectively. At collision energy, this value is about two orders of magnitude above the assumed quench level of $3 \times 10^{-7} \text{ m}^{-1}$. These losses are due to particles from single diffractive interactions at the primary collimators.

Other important losses occur around the interaction point (IP) in correspondence of aperture restrictions at the location of the recombination dipoles. At injection energy, these losses are higher than those in the DS regions downstream of ESS-D. If the magnet aperture cannot be increased, these losses can be cured by carefully positioning tertiary collimators for local protection of the dipole magnets.

At collision energy, losses at the level of 6×10^{-6} are also observed in the dispersion suppression

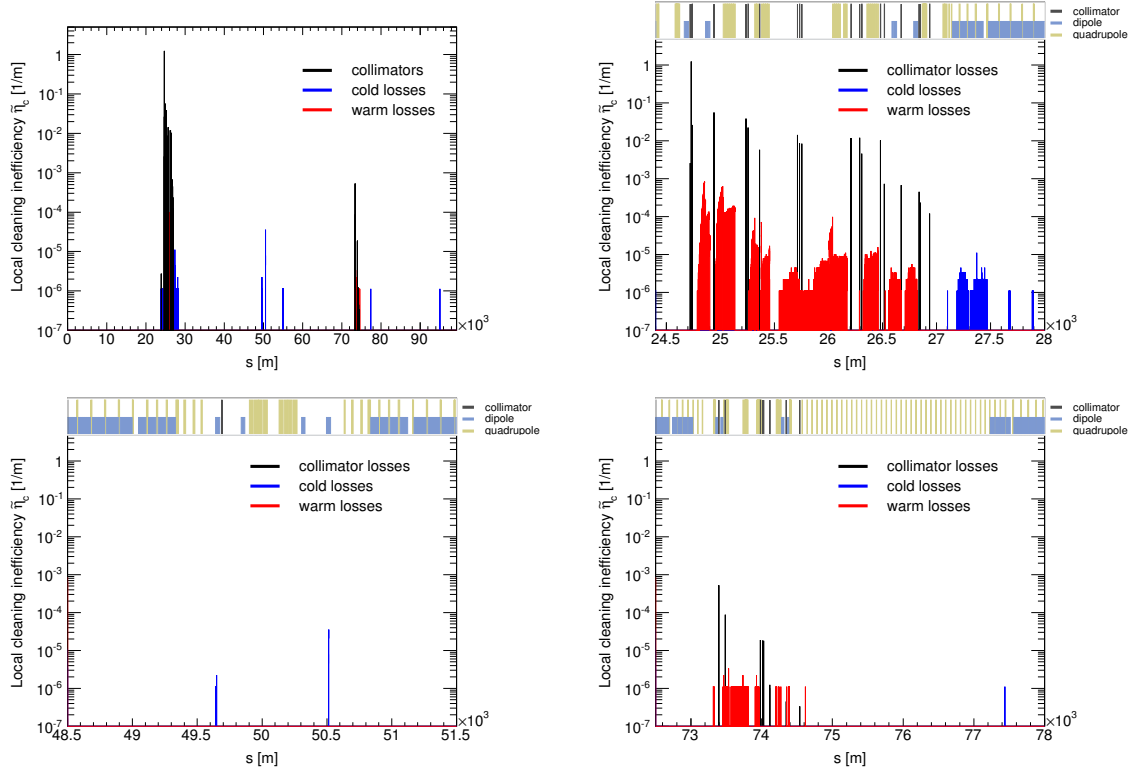


Fig. 5: Horizontal loss map at injection energy of 3.3 TeV for the entire ring (top left), zoom in the betatron cleaning insertion ESS-D (top right), in the interaction point G (bottom left) and in the momentum cleaning insertion ESS-J (bottom right) [23]. The interaction points A and G are located at $s = 0$ and 50×10^3 m respectively, while ESS-D and ESS-J start at about $s = 23 \times 10^3$ and 73×10^3 m respectively.

region after the IP, in correspondence to local peaks in the dispersion function. Such losses are not observed at injection energy: tertiary collimators are in fact retracted and protons are intercepted earlier by the aperture restriction in correspondence of the recombination dipole.

These studies allow to identify the main performance limitation of the system in the losses in the DS after the betatron cleaning insertion. Similar performance limitations are observed at the LHC and have been extensively studied in the context of the HL-LHC upgrade [25]. The LHC solution consists in the installation of new collimators in the DS upstream of the critical locations. This solution intercepts locally beam halos with a significant off-momentum component and has also the advantage to reduce losses in other locations around the ring. At the FCC-hh an analogous solution can be applied and the design of the DS can be optimised to improve the performance of the system with additional local protection collimators. Details of such implementation are being studied for different FCC layout options.

6 Outlook and conclusions

A conceptual design for the FCC-hh collimation system was presented. This first complete baseline comprised betatron and momentum cleaning, with optics scaled from the LHC, dump protection and tertiary collimators. Baseline collimator settings were proposed and the cleaning performance of the system was investigated through detailed tracking simulations. In order to achieve a complete performance evaluation of the FCC-hh collimation system, the simulation tools already used for LHC studies had to be setup for the FCC-hh machine for the first time. The existing tracking code had to be adapted to the FCC case and a first aperture model had to be developed. Achieving detailed loss maps for a ring of the size of

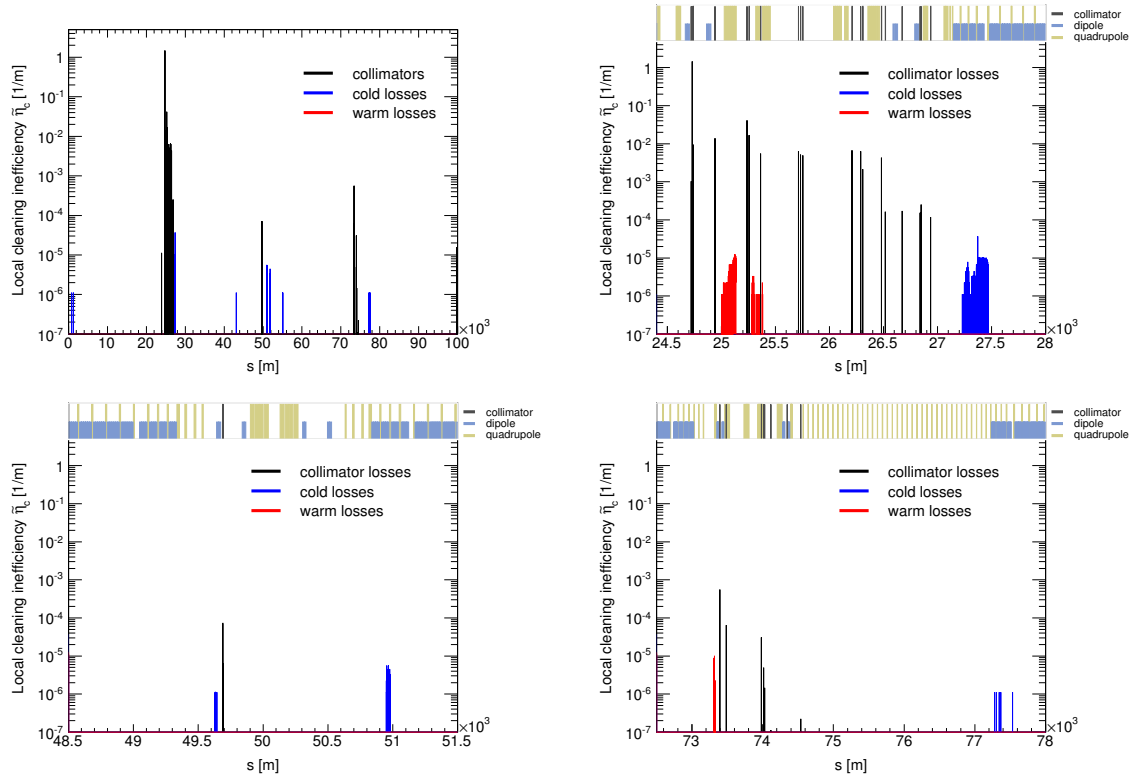


Fig. 6: Horizontal loss map at collision energy of 50 TeV for the entire ring (top left), zoom in the betatron cleaning insertion ESS-D (top right), in the interaction point G (bottom left) and in the momentum cleaning insertion ESS-J (bottom right) [23]. The interaction points A and G are located at $s = 0$ and 50×10^3 m respectively, while ESS-D and ESS-J start at about $s = 23 \times 10^3$ and 73×10^3 m respectively.

the FCC-hh demonstrated to be a challenge in terms of computing performance. This aspect is already being addressed and future FCC studies will profit from the continuous developments on the tracking code [26] aimed to improve its efficiency and capabilities for collimation cleaning.

Similarly to the LHC, the performance limitation of the system was identified to be in the cold losses in the dispersion suppressor downstream of the betatron cleaning insertion. The addition of collimators in the dispersion suppressor and a layout optimisation will be investigated in the future and are expected to provide significant improvements.

Detailed performance estimates can only be assessed once the quench limits of FCC-hh magnets will be known, but this first implementation of a multi-stage system already provides encouraging results. The results of these tracking simulations will provide a first set of input for the collimator hardware design, which should be compatible with the loss scenario of the FCC-hh.

Acknowledgments

References

- [1] A. Ball *et al.*, EDMS doc. 134202 (2014).
- [2] O. S. Bruening *et al.*, CERN-2004-003-V-1.
- [3] A. Chance *et al.*, TUMPW020, Proc. IPAC 2016.
- [4] D. Schulte *et al.*, CERN-ACC-2016-0082.
- [5] FCC-hh lattice repository, <https://gitlab.cern.ch/fcc-optics/FCC-hh-lattice.git>.

- [6] J. B. Jeanneret, LHC-Project-Report-243 (1998).
- [7] E. Quaranta *et al.*, WEPMW031, IPAC 2016.
- [8] C. Bracco, "Commissioning Scenarios and Tests for the LHC Collimation System", PhD thesis, EPFL Lausanne, 2008.
- [9] D. Schulte, FCC Week 2016, <https://indico.cern.ch/event/438866/contributions/1085167>.
- [10] J. B. Jeanneret, LHC Project Report 1007 (2007).
- [11] SixTrack, <http://sixtrack.web.cern.ch/SixTrack/>.
- [12] F. Schmidt, Report No. CERN/SL/94-56-AP (1994).
- [13] E. McIntosh, R. De Maria, CERN-ATS-Note-2012-089 TECH.
- [14] G. Robert-Demolaize *et al.*, FPAT081, Proc. PAC 2005.
- [15] R. Bruce *et al.*, Phys. Rev. ST Accel. Beams 17, 081004 (2014).
- [16] T. Ulrich and Z. Xu, arXiv:physics/0701199v2, 2012.
- [17] C. Tambasco, CERN-THESIS-2014-014.
- [18] M. Fiascaris, Collimation Upgrade Specification Meeting, June 2016, https://indico.cern.ch/event/540629/contributions/2195910/attachments/1289126/1919162/ColUSM73_FlukaCoupling.pdf.
- [19] J. Molson *et al.*, EuroCirCol Meeting, ALBA, November 2016, <https://indico.cern.ch/event/567230/contributions/2346538/attachments/1367641/2072506/JM-EUROCIRCOL-2016-11-08.pdf>.
- [20] S. Redaelli *et al.*, LHC Project Workshop, Chamonix XIV, 2005.
- [21] M. Fiascaris *et al.*, WEPMW006, Proc. IPAC 2016.
- [22] R. Martin *et al.*, TUPTY001, Proc. IPAC 2015.
- [23] M. Fiascaris, R. Bruce, S. Redaelli, "First conceptual design of a collimation system for FCC-hh", submitted to Nucl. Instr. Meth. Phys. Res. Sec. A, November 2017.
- [24] R. Bruce *et al.*, CERN-ACC-2014-0044.
- [25] R. Bruce *et al.*, MOPRO042, Proc. IPAC 2014.
- [26] A. Mereghetti *et al.*, "SixTrack for cleaning studies: 2017 updates", Proc. IPAC 2017.

Simulations of collimation losses at RHIC

G. Robert-Demolaize, A. Drees

Brookhaven National Laboratory (BNL), Upton, NY, USA

Abstract

State-of-the-art tracking tools were developed at CERN to determine the cleaning efficiency of the Large Hadron Collider (LHC) collimation system. In order to properly assess the accuracy of such simulations, benchmarking studies were required using actual beam loss measurements from a machine featuring a similar multi-stage collimation system. The following reviews the results obtained from the analysis of specific data collected from operations at the Relativistic Heavy Ion Collider (RHIC) and compares them to the simulation of these beam conditions in the new SixTrack collimation module after its first official release.

Keywords

CERN report; collimation; tracking.

1 Introduction

RHIC is a circular accelerator made of two individual beam lines (Blue and Yellow) with six common interaction regions (IR's), two of which are currently dedicated to the STAR (IR6) and PHENIX (IR8) experiments. Physics programs include 100 GeV heavy ions collisions, either symmetric ($A-A$) or anti-symmetric (A_1-A_2), and 100 GeV and 255 GeV polarized protons collisions. While it is used solely for machine protection during the ramp, the primary purpose of the RHIC collimation system at store is to control the background for the detectors: the layout is therefore not designed as a machine protection system, a major difference from the LHC case. In total, there are four collimators per beamline: one primary, dual-plane jaw ($H0$, $V0$) and three secondary one-sided scrapers (two horizontal $H1$, $H2$ and one vertical $V1$). Figure 1 gives a schematic view of how this equipment is installed around IR8, as well as a front view (from the beam perspective) of the primary collimator jaws.

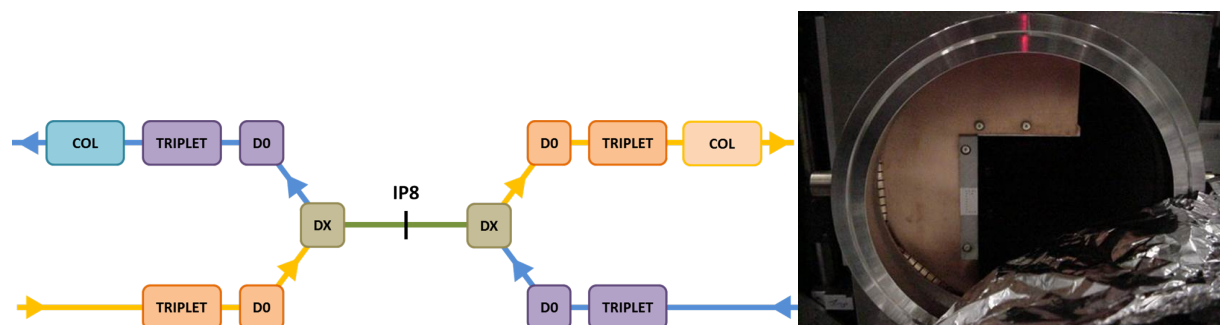


Fig. 1: Left: Schematic layout of the RHIC Collimation System layout in IR8 for both Blue and Yellow beamlines. - Right: Front view of a RHIC primary collimator jaw.

Pin diodes are installed downstream of each collimator to get a direct loss signal when setting its position. This allows controlling experimental background signals while maintaining low levels of local beam losses. The RHIC collimation system setup is overall similar enough to that of the LHC (despite different intended purposes) to allow using data collected during routine physics operation for SixTrack benchmarking purposes. The data considered in the following was taken during the 2005 polarized proton Run (Run5), whose parameters are listed in Table 1.

Table 1: Parameters for 100 GeV p^+p^+ Run5 operations.

Number of bunches	111
Protons per bunch	2.0×10^{11}
E_{store} [GeV]	100
Working point Q_x, Q_y	0.690/0.685
ϵ_N [π mm.mrad]	20.0
L_{peak} [10^{30} cm ² .s ⁻¹]	7.0
<hr/>	
β^* STAR // PHENIX [m]	0.94 // 0.92
β^* IR10, IR4 [m]	10.0
β^* IR12 [m]	5.0
β^* IR2 [m]	3.0

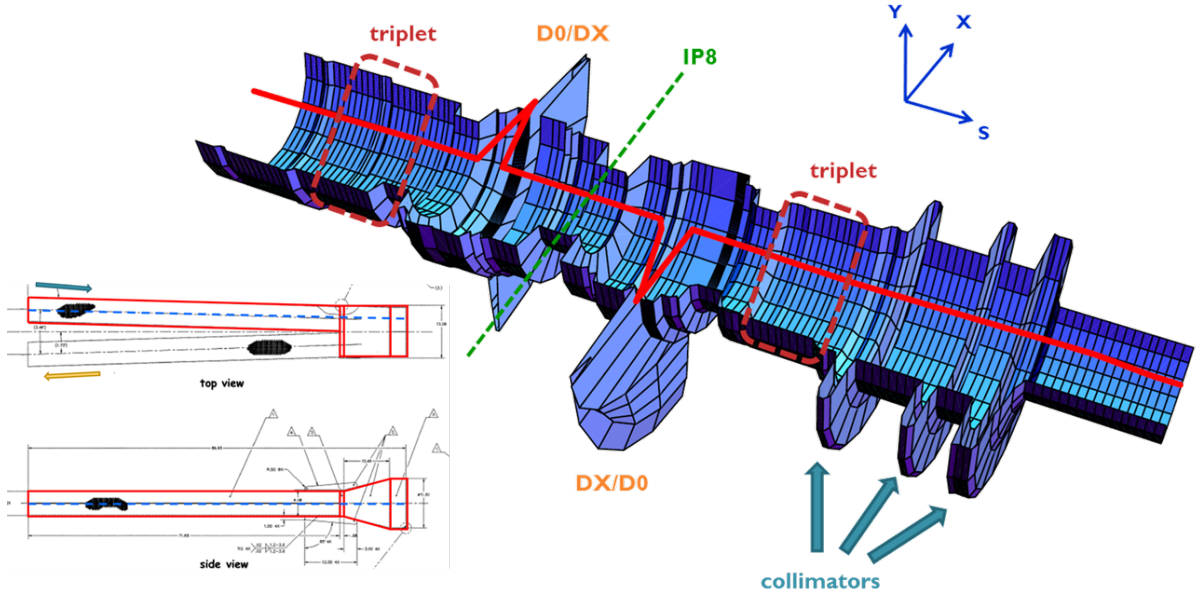


Fig. 2: 3D model of the mechanical aperture in the IR8 insertion. The solid red line represents the closed orbit. The two regions with a larger transverse opening and a large orbit offset correspond to the DX separation magnets. The three “discs” on the right hand side are the location of the RHIC collimators.

Benchmarking studies are aimed at reproducing realistic beam conditions from RHIC operations: this is done by generating a lattice with MAD-X (using the offline model) and convert it to SixTrack for tracking through the collimation system. The transverse coordinates of tracked particles are then processed through a model of the mechanical aperture, and the resulting loss maps are then compared with measurements from beam loss monitors (BLM’s). Figure 2 presents an example of the 3D modeling of the RHIC mechanical aperture. Similar to what was created for LHC studies, the RHIC aperture model is split into 10 cm bins to be as close as possible to the real shape of all elements. On the beam operations side, dedicated data sets were taken by moving the RHIC collimators close to the beam, with all relevant informations (jaw positions, closed orbit, BLM’s signal) being logged during the entire experiment.

2 Results from tracking simulations

The datasets used for benchmarking were collected over the Fill #06981 for the Blue beam during Run5. Figure 3 shows the positions of the collimator jaws and the signal from local BLM's. One can clearly establish when a given jaw is scraping the beam, which corresponds to a spike in the loss signal. For the benchmarking studies, collimator positions are reproduced from their value at 12:27:50 (dashed blue line on Figure 3), when the secondary jaw V1 is the closest to the beam. The resulting simulated beam loss map is then compared to the longitudinal loss locations given by the BLM's.

Results from simulations are shown in Figure 4. The impact parameter on the V1 collimator jaw was taken as $5\ \mu\text{m}$. The BLM data is shown for comparison and corresponds to the difference in the intensity of beam losses between the collimator positions “all out” and “all in”. The predicted loss locations are given by the solid red lines and match with most of the BLM peaks, meaning that the tracking tools developed for LHC collimation studies have reached a satisfying level of accuracy. One should note that while the simulation tools allow locating proton losses with a 10 cm resolution, the live signal from the BLM's is only given at predetermined positions around the machine.

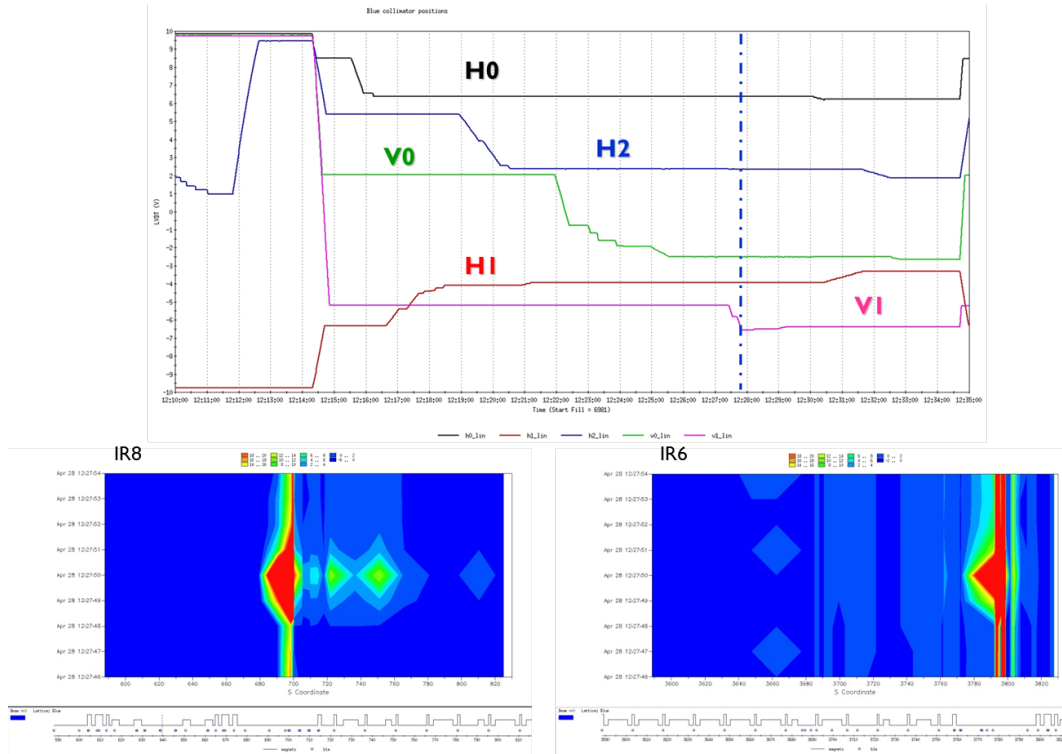


Fig. 3: Top: positions of the Blue collimator jaws in LVDT units versus time during Fill #06981 of the RHIC 2005 p^+-p^+ Run. - Bottom: BLM signal in the PHENIX and collimation insertion (left) and the STAR insertion (right). There is a clear increase in the loss signal at the time when the V1 jaw scrapes the beam.

Figure 5 shows details of the simulations around the collimation region (IR8) and the STAR experiment (IR6). Losses seen at the triplet magnet upstream of the collimation system ($s = 600\text{ m}$) and around IP6 ($s = 0\text{ m}$) are due to some of the halo protons that were scattered by the collimators and managed to escape further downstream. These protons face an aperture bottleneck at these quadrupoles since β^* in both experimental insertions is squeezed down to 0.9 m for higher luminosity. The high level of losses seen by the BLM's in IR8 around $s = 700\text{ m}$ is due to showers coming from inelastic interactions in the collimator jaws. This is shown by the green dashed lines in the simulated loss maps, giving the statistics of inelastic interactions in each jaw.

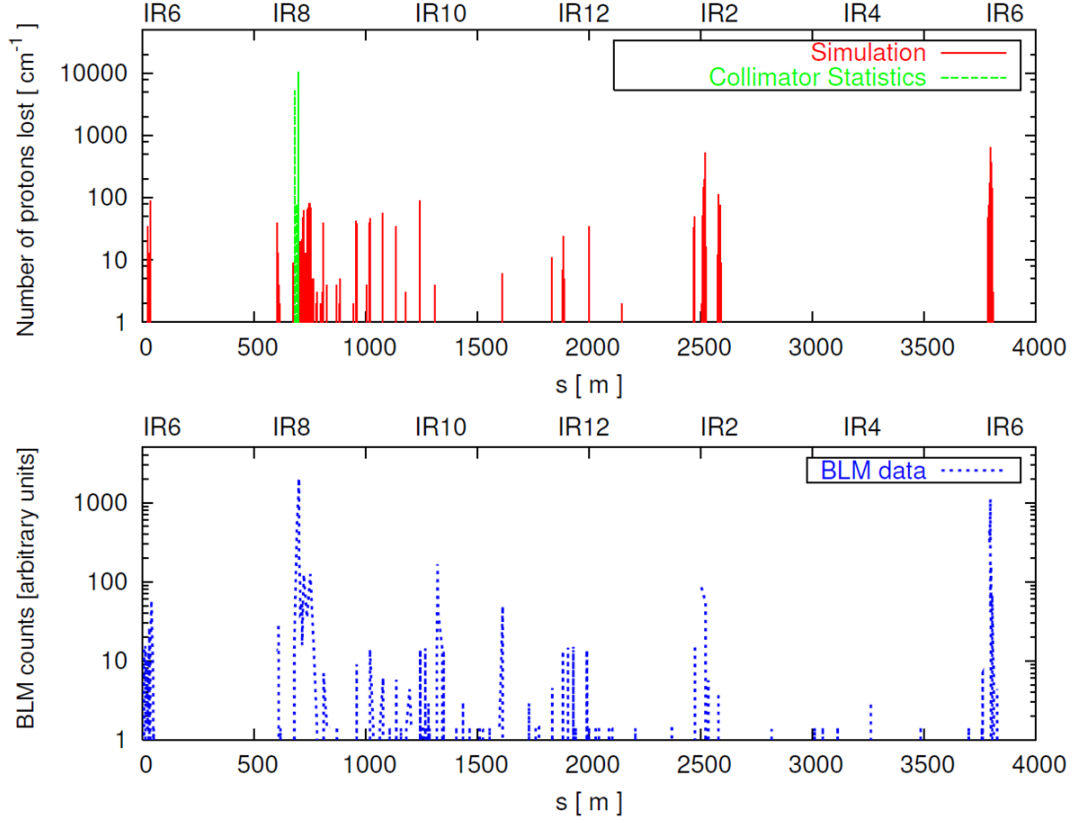


Fig. 4: Comparison between simulated loss maps (top) and BLM measurements (bottom) due to beam impacts on the V1 collimator jaw for the Blue beam circulating from left to right.

The relative height of the peaks in the simulations can be compared with the live measurements too. When studying the statistics of the predicted losses, one can consider that each BLM can only “see” beam losses up to a certain distance upstream of it, taken as $d = 5$ m here. Table 2 presents a quantitative comparison between simulations and measurements for the region close to the collimators (between $s = 700$ m and $s = 750$ m as seen in Figure 5). It shows that the overall variation of loss amplitudes at the various BLM locations is to the first order reproduced by the simulations.

Table 2: Comparison between simulations and live measurements for the statistics shown in Figure 5 of beam losses induced by the RHIC collimation system.

BLM location (from IP6)	BLM signal [rad.h ⁻¹]	N_{losses} upstream over $d = 5$ m
(1) $s = 705.5$ m	38.021	89 ± 9
(2) $s = 710$ m	50.979	132 ± 11
(3) $s = 714.2$ m	55.743	127 ± 11
(4) $s = 721.7$ m	124.600	496 ± 22
(5) $s = 736.5$ m	35.136	97 ± 10
(6) $s = 751.2$ m	125.331	495 ± 22

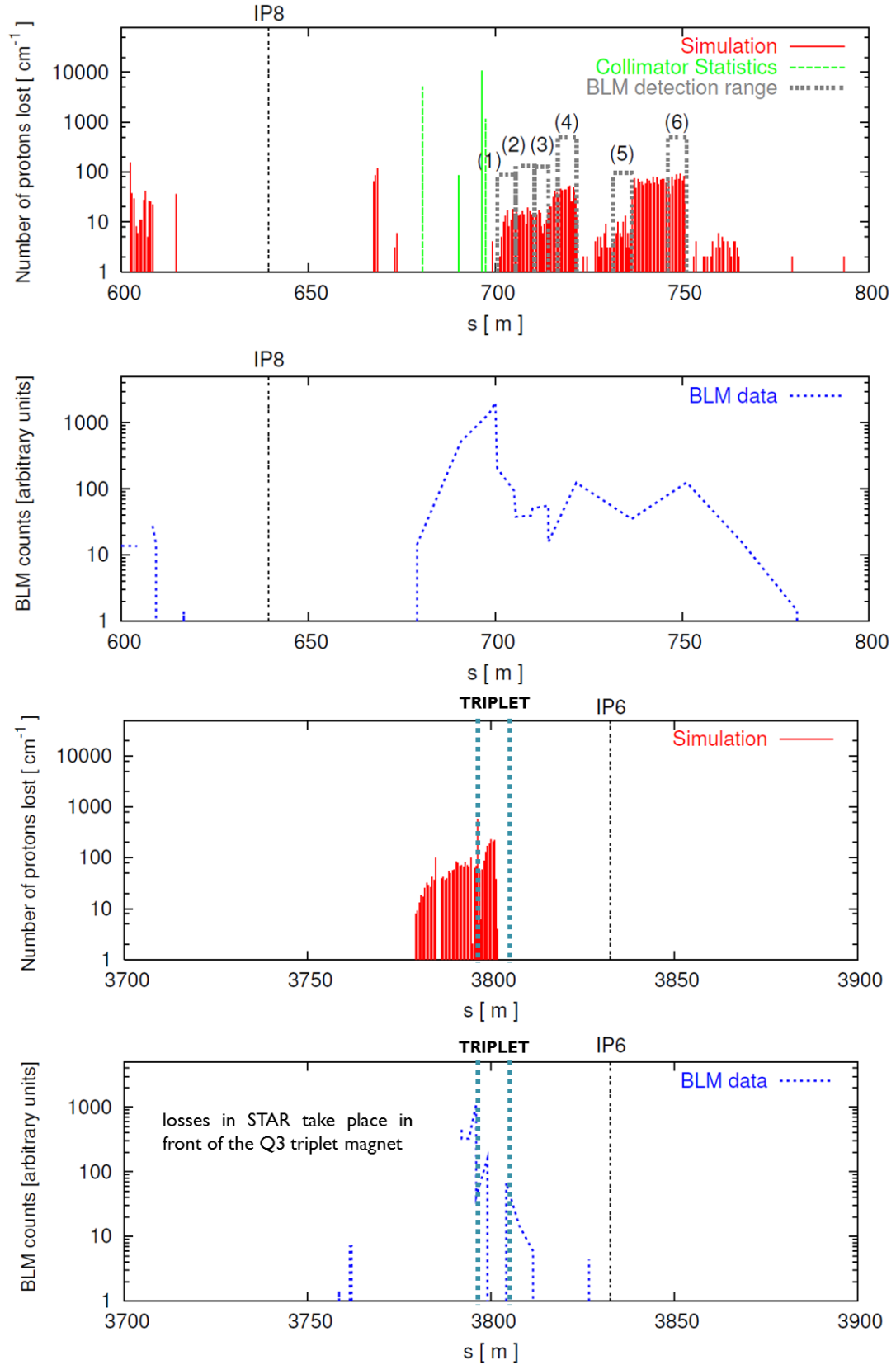


Fig. 5: Zoom of the simulated loss maps and BLM signal around the collimation region (top) and the STAR experiment (bottom) following the Blue beam. Beam losses can be spotted at the triplet magnet upstream of the collimators.

Deriving a scaling law from the loss location statistics is complicated, since the tracking tools were at the time designed to show only the locations where the protons scattered by the collimation system are lost, while particle showers induced by the proton-matter inelastic interactions in each collimator are the source of signal seen by the BLM's. Recent upgrades to SixTrack now include in-tracking checks on the mechanical aperture limitation, as well as a better interfacing to FLUKA for energy deposition studies. Repeating the simulations described in this section with up-to-date tracking tools would therefore allow for refining the statistics in Table 2.

3 Machine Protection Studies

With the stored beam intensity reaching record highs every Run, protecting RHIC's detectors as well as RHIC's superconducting magnets from excessive losses became more and more important. The most prominent reason for massive instantaneous beam losses are, notably, RHIC abort kicker prefires as described in [2]. A prefire is an event in which one of the five RHIC abort kickers self-triggers asynchronously. Given the location of the abort system (IR10) with reference to the location of the two low β^* experimental insertions (IR6 and IR8) it is obvious that the location of the collimators on the outgoing sides of IP8 are in no position to protect experiments from one-turn type losses such as occurring during a prefire event. While prefires of the blue abort system pose the highest risk to the STAR detector at IP6, prefires of the yellow abort system are most dangerous to the PHENIX detector at IP8, one arc downstream of the abort kicker location. Therefore, two additional titanium masks were installed in RHIC, each one arc downstream of the respective abort system, i.e. just upstream of the triplets on the incoming side in IR8 (yellow) and IR12 (blue). Figure 3 shows a design drawing of one the installed masks.

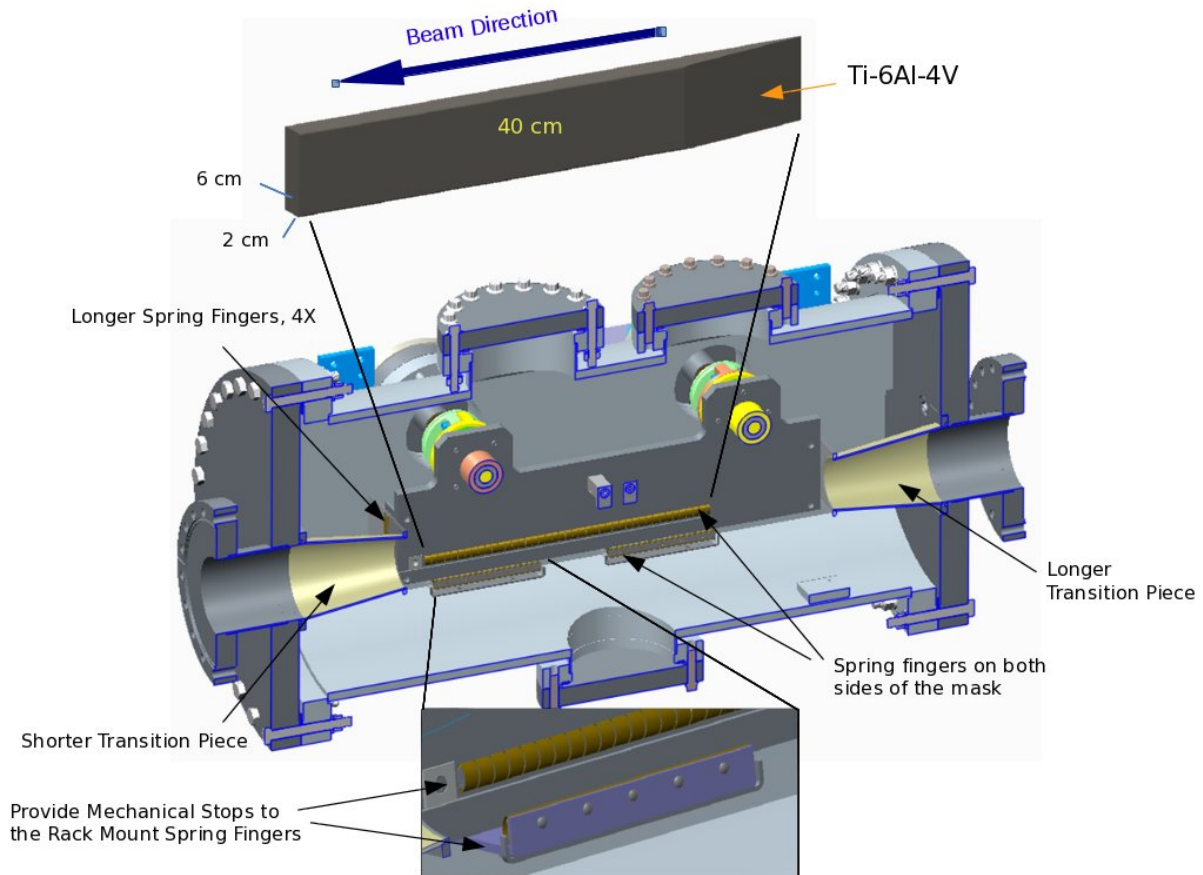


Fig. 6: Design drawing of titanium mask

Operational experience with these masks during the 2015 RHIC operations (Run15) showed them to be insufficient to protect in particular the PHENIX detector from damage in a prefire. One subdetector's preamplifiers got almost entirely destroyed in two prefire events during the p-Au mode of Run15. Two aspects turned out to eliminate the effectiveness of the masks: the tight restrictions to the phase advance between the abort system and the mask location and, in the yellow ring, the unfortunate location of the mask within the same straight section and thus immediately upstream of the experiment itself. Therefore, a protection scheme with large orbit protection bumps as shown in Figure 3 got reinstated. Currently studies are underway to prevent prefire events entirely with a focus on the abort kicker trigger circuits.

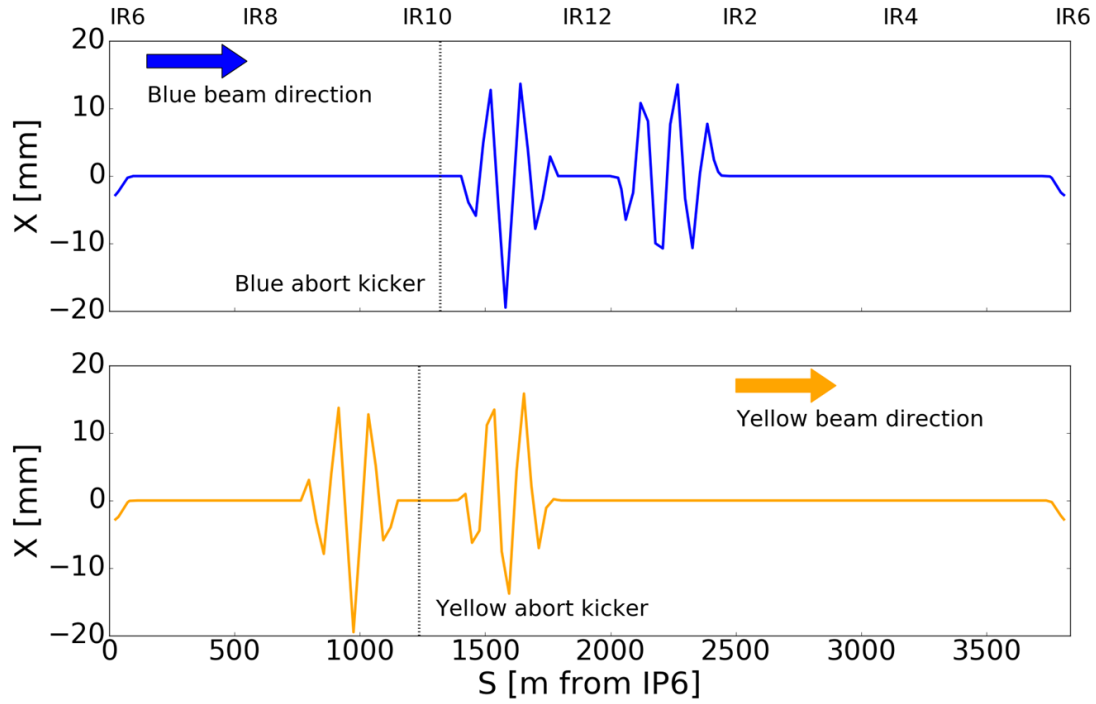


Fig. 7: Closed orbit bump for the Blue (top) and Yellow (bottom) beam in RHIC for abort kicker prefire protection

References

- [1] W. Herr, B. Muratori, *CERN Accelerator School: 5th General accelerator physics course*, CERN 94-01 vol. 1 (1994).
- [2] A. Drees et al., *RHIC Prefire protection masks*, C-A/AP/533, January 2015.

Simulation and optimization of beam losses during continuous transfer extraction at the CERN Proton Synchrotron*

J. Barranco García and S. Gilardoni
CERN, Geneva, Switzerland

Abstract

The proton beams used for the fixed target physics at the Super Proton Synchrotron (SPS) are extracted from the Proton Synchrotron (PS) by a multiturn technique called continuous transfer (CT). During the CT extraction, large losses are observed in locations where the machine aperture should be large enough to accommodate the circulating beam. This limits the maximum intensity deliverable due to the induced stray radiation outside the PS tunnel. Scattered particles from the interaction with the electrostatic septum are identified as the possible source of these losses. This article presents a detailed study aiming to understand the origin of losses and propose possible cures. The simulations could reproduce accurately the beam loss pattern measured in real machine operation and determine the beam *shaving*, intrinsic to the extraction process, as the cause for the unexpected losses. Since these losses are unavoidable, the proposed solution implies a new optics scheme displacing the losses to a region with better shielding. New simulations demonstrate the satisfactory performance of the new extraction optics and its suitability to be implemented in the machine. Finally, beam loss measurements in these new operation conditions confirmed the previous simulation results.

Keywords

CERN PS, Continuous Transfer, CT Extraction, beam losses, electrostatic septum

1 Introduction

The Proton Synchrotron (PS) is the oldest element in the CERN accelerator complex, however capable of delivering a wide variety of beams for different purposes [1]. According to the characteristics of the beam the extraction procedure varies. Three different extraction techniques are used in the PS: fast, multiturn or continuous transfer (CT), and slow or resonant extraction. On a yearly basis, radiation surveys are carried out to monitor radiation levels in the specific areas outside the tunnel where large losses occur (typically injection and extraction regions), in addition to the surroundings of office buildings. During the 2006 survey, two hot areas were identified (Fig. 1): the injection region from PS booster (stations 4 to 5) and building 151 (stations 22 to 23 and 29 to 35, where the first two are located just outside building 151 while the other seven are measured inside [2]). Building 151 is also known as PS bridge because the PS tunnel traverses it, as depicted in Fig. 2. Losses in the injection region can be explained by the injection process itself; however, losses in building 151 are unexpected as there are no aperture limitations in that region. The radiation surveys do not specify which specific process is causing the losses. An analysis of beam loss monitor (BLM) measurements during all PS cycles revealed that only during CT extraction the beam is lost in that particular region. Figure 3 shows the BLM readings (i.e. cumulative losses) for a complete CERN neutrinos to Gran Sasso (CNGS)¹ beam cycle. There are three

*This work, published in [Phys. Rev. ST Accel. Beams 14, 030101 \(2011\)](#), is licensed under a [Creative Commons Attribution 3.0 License](#).

¹The CNGS beam is delivered to the SPS for neutrino production as a part of the CERN neutrinos to Gran Sasso experiment [3].

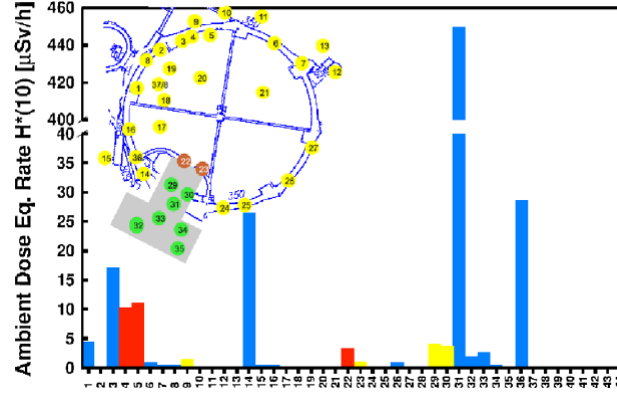


Fig. 1: Result of the 2006 PS radiation survey in $\mu\text{Sv/h}$ for a circulating intensity of $0.8 \times 10^{13} \text{ p}^+/\text{sec}$ [2]. Measurements were taken from 44 different stations spread all along the PS tunnel (top left). The color code in the histogram represents the ratio between the measured dose and the guideline value specified in [15]. In blue, stations where the dose is below 30% of the guideline value; in yellow, stations where the measured dose is between 30 and 100%; in red, where the measured dose is above 100%. The stations measuring in building 151 are 22-23 (brown) and 29-35 (green).

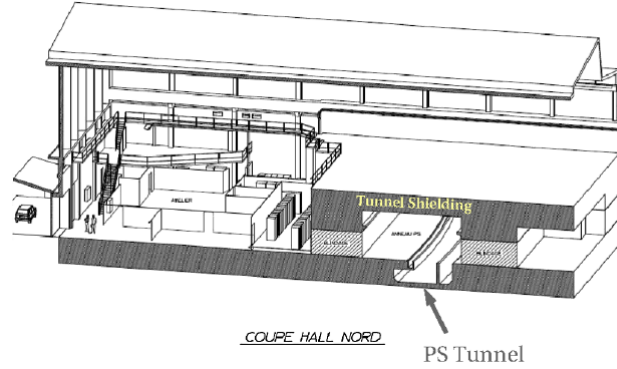


Fig. 2: Drawing of the PS bridge building 151. The tunnel PS traverses it (bottom right) being not possible to increase the shielding in that region.

main loss locations after the extraction process starts: first BLMs 5 to 10, where the saturation of BLM number 9 matches up with the high radiation dose in building 151 that appeared during the 2006 survey (Fig. 1); second, BLM 16 where the magnetic septum to extract the beam towards SPS is located; third, BLMs 30 to 35 where the saturation of BLM 31 indicates the location of the electrostatic septum used to slice the beam during the CT extraction. Studies were launched [4] to explain why these losses arose, considering that there are no aperture limitations in the PS ring in that area. The outcome of these studies pointed out particles scattered by the electrostatic septum blade as the source of the losses. This study confirms the assumptions regarding the beam loss mechanism and proposes and implements a solution to reduce the losses in building 151.

2 Principle of the CT Extraction

The PS ring is composed of 100 combined-function magnets arranged in a focusing-defocusing-defocusing- focusing lattice and interleaved by 100 straight sections (SS, numbered from SS00 to SS99).

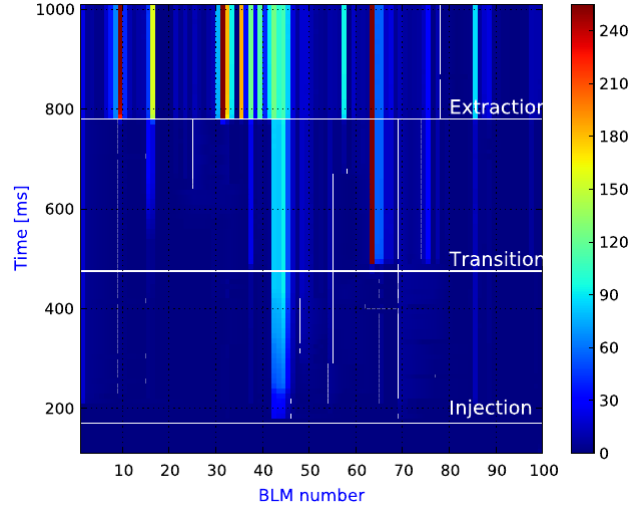


Fig. 3: BLMs integrated measurement for the CNGS beam. The BLM signal is divided into 255 bits, beyond that value the BLM saturates. The timings for injection, transition crossing, and extraction are shown. Losses below the PS bridge are measured by BLMs 5-9.

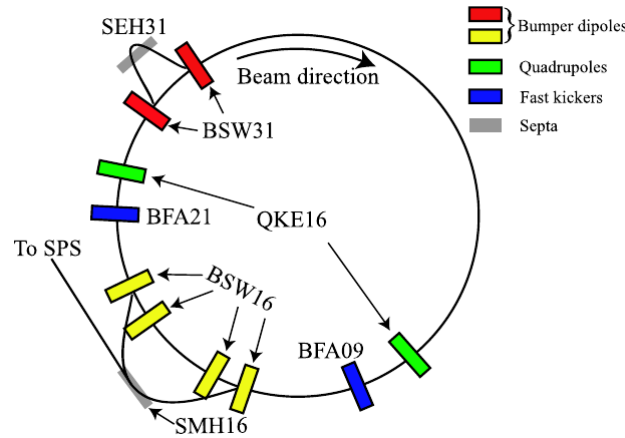


Fig. 4: Layout of all elements used during the CT extraction.

All the machine elements are named with reference to the straight section where they are located. The CT extraction is used to fill the SPS in two consecutive PS runs ($\text{circumference}_{\text{SPS}} = 11 \times \text{circumference}_{\text{PS}}$). All elements used in this process are displayed in Fig. 4.

When the 14 GeV/c flattop is reached, the PS horizontal tune is set to $Q_x = 6.25$ which provides a phase advance per turn of 90° , while the vertical one remains fixed at $Q_y = 6.3$. A few milliseconds before the extraction, two slow bumps, BSW31 and BSW16, are powered. The first one approaches the beam to the electrostatic septum in SS31 (SEH31), whereas the second does it to the magnetic septum in SS16 (SMH16). Simultaneously, two quadrupoles located in SS25 and SS05, which form the so-called QKE16 circuit, are used to alter the optics between them (Fig. 5), in such a way that large β_x and almost zero D_x values are obtained in the SEH31 location while the optics remain unchanged in the SMH16 location. These changes increase the beam size and reduce the energy dependence in SEH31, minimizing the losses at the septum while keeping the beam size small during the extraction in SMH16. When the beam is close enough to SEH31 and the first losses appear, two fast magnets with rise time

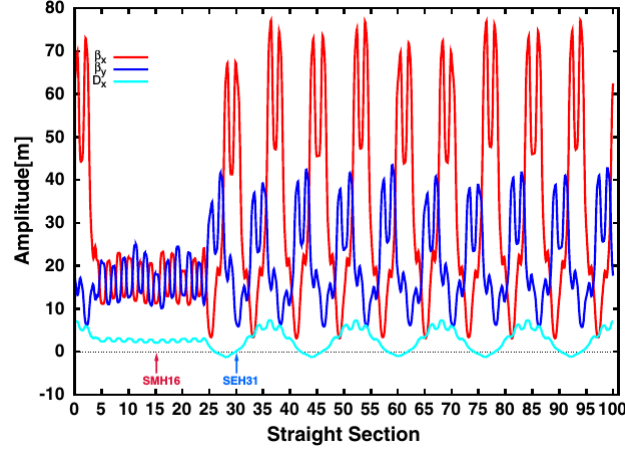


Fig. 5: PS optics distorted by the QKE16. The region at small horizontal and vertical β is between SS05 and SS25. The locations of SMH16 and SEH31 are pointed out.

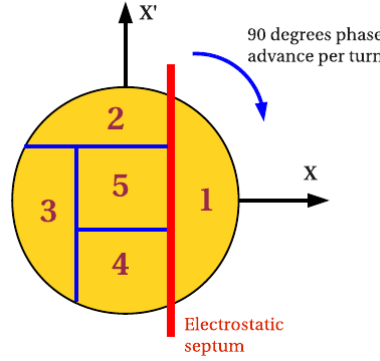


Fig. 6: Five slices in normalized phase space. $Q_x = 6.25$ corresponds to 90° phase advance per turn. As the $C_{SPS} = 11 \times C_{PS}$, two consecutive PS extraction of five slices each are needed to fill the SPS completely.

much shorter than the revolution period, namely, BFA21 and BFA09, will be used to adjust the relative position between beam and septum blade on a turn-by-turn basis, to obtain an equal intensity extraction per turn (Fig. 6). The portion of the beam above the blade will receive an additional kick that along with BSW16 will drive the particles above SMH16 and will extract them to the transfer line towards the SPS. The strengths of the fast kickers are increased during the five turns to assure the equal population between the different slices. A typical time structure of the BFAs strength is shown in Fig. 7. The total kick is divided into a constant strength or pedestal (red), and an increasing one or staircase (blue) to refine the adjustment. As described in [5], it is not possible to generate slices with same emittances and same intensities at the same time: the condition of equal intensity is the most suitable to reduce beam instabilities in the SPS.

The typical five turns long spill structure measured by the current transformer in the transfer line from PS to SPS is shown in Fig. 8. By integrating the signal the extraction efficiency for each turn is obtained and shown in Table 1. According to this, the two main parameters taken into account in order to validate the simulations are an equal intensity extraction ($\sim 20\%$ per slice) and an overall average efficiency of the process of $\sim 93\%$. The simulations described in the following sections include tracking

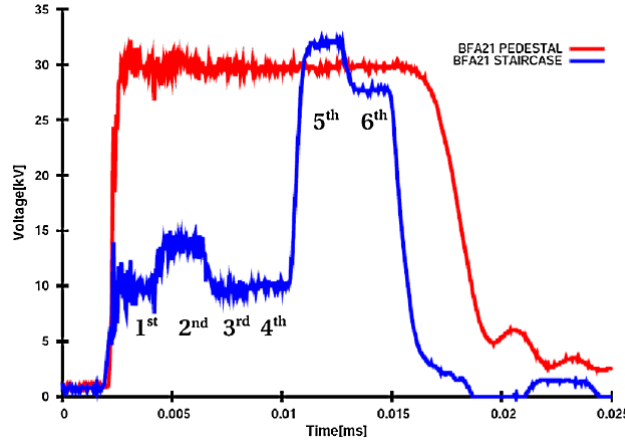


Fig. 7: BFA21 kicker strength in kV during the five extraction turns. The pulse length corresponds to six machine turns. The last part (6th) turn is used to empty the machine. During the fifth turn the remaining central part of the beam is placed above the septum to be extracted and a higher voltage is needed.

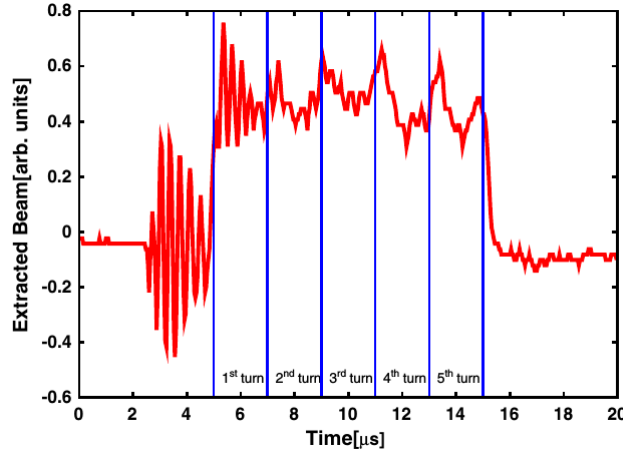


Fig. 8: Five turn spill structure measured by a current transformer in the transfer line between the PS and the SPS. Some noise on the reading is observed before the extraction and during the first turn.

through the PS lattice with the fast bumps changing turn by turn, particles interaction with the septum blade, and finally trajectory comparison with the PS aperture model to determine the loss locations.

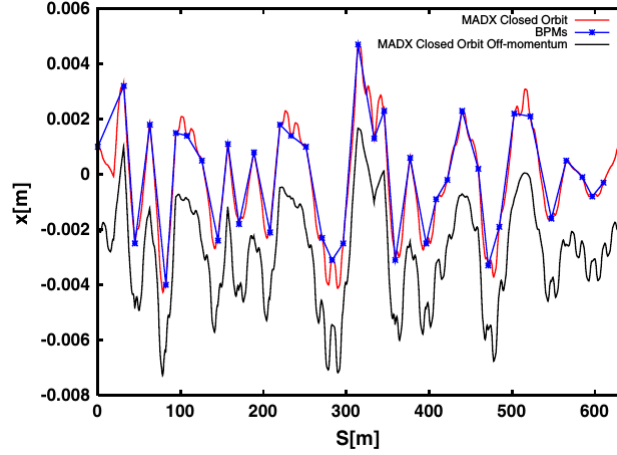
2.1 Optics of the CT extraction

During operation the radial position of the beam, thus the energy, is adjusted to match the injection energy of the SPS. A typical radial displacement implemented in the PS for this reason is of the order of $\Delta x = -3$ mm giving a momentum error of about $\delta p/p = -0.001$. In addition, there is a peak to peak value of $\Delta x = 8$ mm in the closed orbit measured just before the extraction starts. Figure 9 shows the MADX [6] orbit model used in the simulations which includes the real closed orbit measured by the BPMs and the radial displacement.

Particle tracking around the ring is performed with a modified version of the SIXTRACK code [7] based on the MADX model provided. In order to simulate the change in the closed orbit during the five turns extraction two dipoles, varying strengths turn by turn, are included in the code. The kick provided

Table 1: Extraction efficiency measurements for each of the five turns, with an overall average efficiency of $\sim 93\%$.

Beam	
Turn	Extracted(%)
1st	18.1
2nd	18.5
3rd	20.5
4th	18.1
5th	17.8

**Fig. 9:** PS closed orbit simulated and measured before the pulsing of the slow bumps.

by the fast bumps is calculated by

$$\begin{aligned}
 \theta_{SS21} &= -U_{S,N} \frac{\int B dl_{st}}{U_{max,st}} \frac{1}{B\rho} - U_{P21} \frac{\int B dl_{pd}}{U_{max,pd}} \frac{1}{B\rho} \\
 \theta_{SS9} &= -U_{S,N} \frac{\int B dl_{st}}{U_{max,st}} \frac{1}{B\rho} - U_{P9} \frac{\int B dl_{pd}}{U_{max,pd}} \frac{1}{B\rho}
 \end{aligned} \tag{1}$$

with $U_{S,N}$ the voltage of the staircase for the turn N , $U_{max,st}$ maximum voltage of the staircase, U_{P21} and U_{P9} voltage for fast bumpers in 21 and 9, $U_{max,pd}$ maximum voltage of the pedestal, $B\rho$ the magnetic rigidity, and $\int B dl$ the integrated magnetic field for the maximum voltages $U_{max,pd}$ and $U_{max,st}$. In real operation the fast bumps are adjusted manually until the equal extraction is achieved. Normalizing the coordinates, $X = \frac{x}{\sqrt{\beta}}$ and $X' = \frac{\alpha}{\sqrt{\beta}}x + \sqrt{\beta}x'$, and integrating over the distribution (bi-Gaussian in phase space) to obtain 1/5 of intensity in each slice as shown in Fig. 6, the relative position between beam and septum blade can be calculated. From these values, the needed closed orbit per each extraction turn can be calculated and with Eq. (1) the coefficients for the pedestal and staircase are then adjusted. Figure 10 shows the simulated closed orbits for the slow bumps and the five consecutive extraction turns. While the closed orbit peak in SS16 is kept constant, the one in SS31 increases according to the values calculated before. The zoomed area in Fig. 10 shows the relative position between septum blade and closed orbit during the five turn process. Only in the fifth turn the closed orbit goes above the septum as the central part of beam has to be extracted in the last turn. Table 2 summarizes the parameters for each turn used in the simulations to set up the equal intensity extraction. These values are included in the SIXTRACK model of the PS lattice.

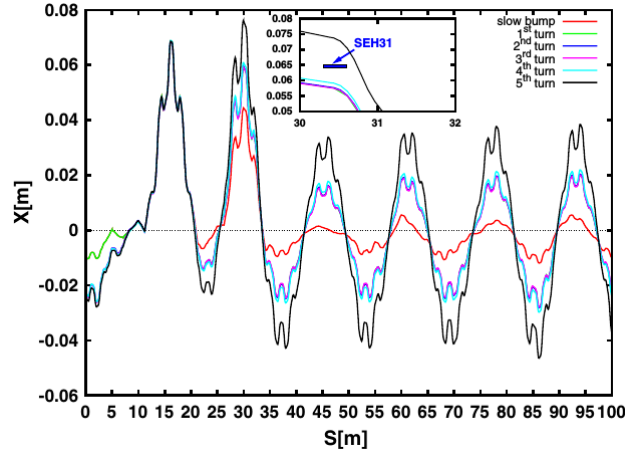


Fig. 10: Simulated closed orbit in the tracking code and zoom of the position between septum and closed orbit. The necessary fast-bump values for an equal intensity extraction turn by turn are calculated considering a bi-Gaussian beam distribution. The zoom indicates the position of SEH31 with respect to the closed orbits for each turn.

Table 2: Parameters used in the simulations to achieve the equal intensity extraction. The second column is the relative position of the beam and septum blade for a bi-Gaussian beam. The third and fourth columns show BFA kicks in mrad calculated from the needed closed orbit. The last column shows extracted efficiency for each of the five turns.

Turn	Relative Position	Kick BFA21	Kick BFA9	Beam Extracted [%]
1st	0.84	-0.5225	-0.5070	20.0
2nd	0.67	-0.4793	-0.4638	18.5
3rd	0.62	-0.4953	-0.4798	19.2
4th	0.32	-0.5864	-0.5709	19.0
1st	...	-1.0236	-1.0081	19.5

2.2 Model of the electrostatic septum

The interaction of particles with the septum blade is included in the simulations with the K2 code [8]. The electrostatic septum in SS31 is composed of a 1.8 m long molybdenum blade creating an electrostatic field with the anode as shown in Fig. 11. The vacuum pipe is earthed so the circulating beam is not affected. The main septum parameters used in the simulation are presented in Table 3.

The kick given by an electrostatic septum is calculated with [9]

$$\theta = \arctan \left(\frac{eE_0 l}{pc\beta} \right) \quad (2)$$

where e is the electronic charge, E_0 the electric field gradient, l the electrode length, p the particle momentum, c the speed of light, and finally, β the relativistic normalized speed. For the values in Table III, the maximum kick from SEH31 is $\theta_{\max} = 0.86$ mrad. The K2 model used in the simulations takes into account all possible cases when a particle reaches the septum (Fig. 12). If a particle arrives above the septum (case A in Fig. 12), it can either traverse the whole electrostatic field receiving the maximum kick or, if its divergence is negative, it might end up hitting the septum blade. On the other hand, if a particle arrives below the septum (case C in Fig. 12), it can either continue circulating freely in the vacuum chamber or, if its divergence is positive, it could hit the septum blade at some point. Finally, a

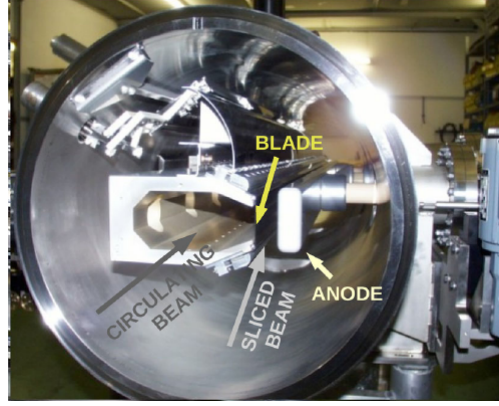


Fig. 11: Electrostatic septum 31. The circulating beam passes in the left vacuum pipe, whereas the electrical field is present at the right of the septum blade.

Table 3: Main SEH31 septum parameters.

Parameter	Value	Unit
Electrode length	1850	mm
Septum length	2293	mm
Septum half gap	65	mm
Blade nominal thickness	0.1	mm
Operative gap width	27	mm
Angle with respect to beam	-0.0006	rad
Voltage(DC)	170	kV

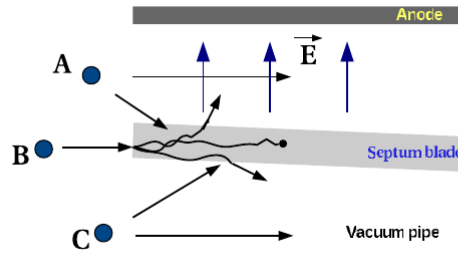


Fig. 12: Sketch of all possible cases when a particle reaches the septum.

particle could hit the septum in the front (case B in Fig. 12). In all the cases that the particle traverses a certain length of material, it will undergo all the processes included in K2, i.e., multiple Coulomb scattering, ionization energy loss, and nuclear interactions [10]. In the event of a particle outscattered into the electro- static field region, the kick applied is proportional to the remaining length traversed inside it.

2.3 PS aperture model

An accurate aperture model is essential to reproduce the loss pattern measured in real operation by the BLMs. Because of this, special effort was put in order to measure and establish it, based on dimensions from mechanical drawings for each element and measurements made in situ [11]. Figure 13 shows

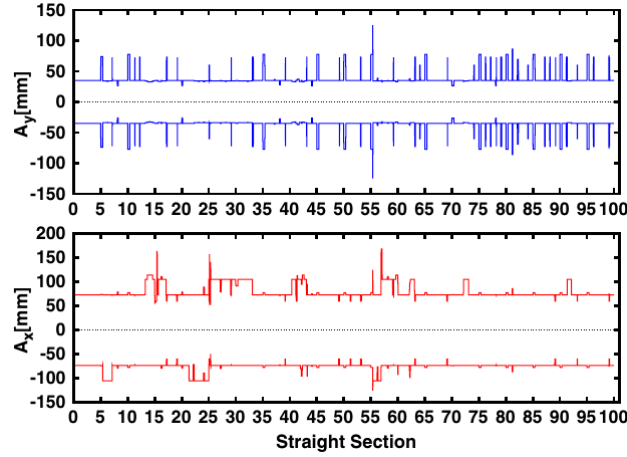


Fig. 13: PS aperture model in mm. Vertical (top), horizontal (bottom).

the horizontal (bottom) and vertical (top) aperture model used to compare with particle trajectories to determine the loss locations.

3 Beam Loss Monitor System

In order to be able to correlate simulations with measurements, it is necessary to understand how losses are measured in the PS. The PS beam loss monitor system is composed by 100 aluminum cathode electron multiplier (ACEM) detectors [12], mounted on top of the main magnets. Because of the position chosen in the past (the system dates back to the middle 1980s) and the large variation of the signal with the primary proton energy, it is practically impossible to correlate the amount of protons lost in the machine with the signal recorded by the BLMs, even if all the BLMs are calibrated every year to assure a uniform response for a given loss [13]. Moreover, most of the losses in the PS are in the horizontal plane, whereas the BLMs are mounted on top of the bulky iron yoke of the main magnets just after each SS as shown in Fig. 14. In addition, the BLMs are installed on different sides of the main magnets, sometimes facing the inside (as in Fig. 14) of the ring, sometimes outside, following the pattern of the four magnet types which compose the PS lattice. According to their location, they cover a different solid angle directly affecting the measurements. For these reasons, the BLM system is used during operation to detect malfunctioning of the machine, and not as a real protection system, obtained by comparing online a given loss pattern with the reference one.

The pattern of the losses observed with the system can tell something about the region where the losses occur, whereas it is not possible, for example, to deduce the ratio of beam lost between two different sections of the machine. It is not even possible to deduce if the loss is produced in one of the main magnets or in the straight sections, as the BLM is mounted at the entrance of the magnet unit. The aim of the simulations presented in the following section is to reproduce qualitatively the observed loss pattern but not to quantify the beam loss detected by the BLMs. This would require a detailed simulation of the secondary shower developing in a large fraction of the main magnets, and goes beyond the purpose of this study.

4 Simulation Results

The average extraction efficiency achieved during machine operation is $\eta=93.0\%$ with approximately equal intensity for the different beam slices as presented in Table I, in comparison with simulations which give an overall extraction efficiency of $\eta=96.3\%$ with an intensity per slice quoted in Table II. This discrepancy can be explained from the fact that in the simulations the real alignment of the septum

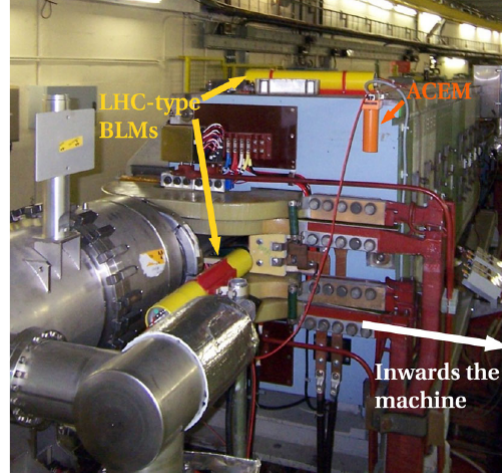


Fig. 14: ACEM detector mounted on the main magnet unit 16. The yellow cylinders on top of the magnet and parallel to the extraction magnetic septum are new BLMs-LHC type under tests.

Table 4: Extraction efficiency measurements (%) simulated for different relative angles between beam and septum blade and different septum widths.

Angle [rad]	Width [mm]		
	0.1	0.2	0.3
-0.0002	96.1	94.5	93.4
-0.0006	96.3	95.3	94.2
-0.001	95.0	94.0	92.6

with the rest of the machine elements considering all of them centered with respect to the closed orbit. In addition, simulations have been performed with the nominal width of the septum; however, the impact of the beam in the septum and the energy deposited could have lead to a deformation of the blade increasing the width. A larger impacting surface can certainly lead to a decrease in the extraction efficiency. The average extraction efficiency evaluated for different alignments and septum widths is presented in Table IV. More realistic scenarios reduce the simulated extraction efficiency to values in good agreement with the measurements during machine operation.

Figure 15 (top) compares beam loss histograms from measurements and simulations considering the nominal parameters, showing a very good qualitative agreement as all the hot locations were found in the latter. Within the $\sim 4\%$ of losses, 95% are found to be caused by the scattering in the septum blade. Specifically this is the reason for losses in SS05-09, where the quadrupole located in SS05 (as a part of the QKE16), defocuses the scattered particles and cause the losses. The maximum energy loss simulated for the particles lost in the SS05-10 region is 0.5% which gives a dispersive contribution to the closed orbit of 15 mm still far from the available aperture confirming the kick provided by the quadrupole SS05 as the main reason for the losses. The other 5% of losses are particles which have been not scattered by SEH31 but due to a wrong closed orbit are either not kicked properly in SEH31 or extracted in SMH16. However, the particles can be either outscattered towards the anode (sliced beam) thus receiving an additional kick, or towards the center of the vacuum chamber (circulating beam) as shown in Fig. 12. Figure 15 (bottom) compares simulated losses for the two cases, showing that one-third of the particles lost were scattered towards the vacuum chamber and lost in the sections immediately after the septum while the rest were outscattered into the field region spreading more the losses and producing the ones in SS05-09 after being deflected by a quadrupole in SS05. These three results (overall

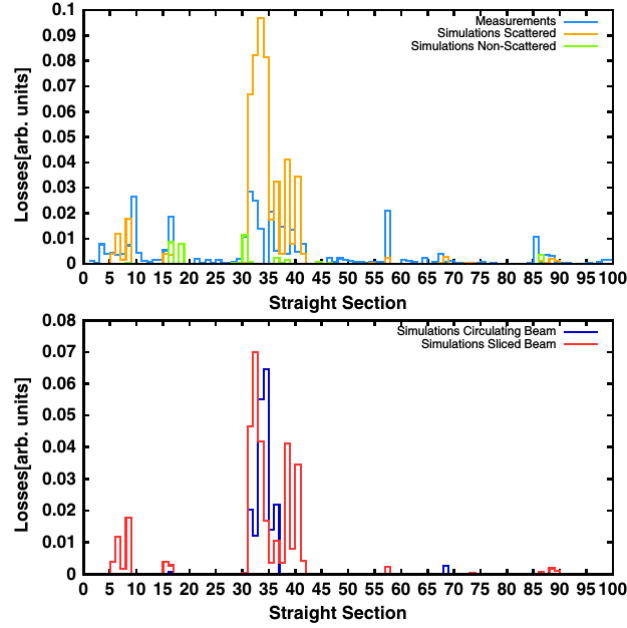


Fig. 15: Beam loss distribution for normal CT operation. Top, BLM measurements (blue) are compared with simulations (green, lost particles nonscattered, and orange, particles scattered by the blade). Bottom, simulation results for scattered particles differentiating into sliced particles (red), i.e., outscattered into the field region, and circulating particles (blue), i.e., outscattered into the vacuum pipe.

efficiency, turn-by-turn efficiency, and beam loss distribution) confirm the accuracy of the model used for simulations. However, as already mentioned in Sec. 3, there are certain issues that prevent from a one-to-one comparison between simulations and real measurements when analyzing the results.

Losses in sections SS05 to SS09 are clearly identified, but in relative terms the loss peaks are smaller in the simulations than in the measurements. This is probably because the simulations do not take into account secondary showers from the particles hitting the machine aperture, counting only direct losses. Another significant difference appears in section 57, where the magnetic septum for slow extraction (SMH57) is placed. Those losses come probably again, from secondary particles produced in the septum and measured by the BLM in that location.

The BLMs are mounted, as mentioned before, at the end of each straight section inside or outside depending on the magnet, while in the simulations all particles are grouped and counted considering the whole section producing which can lead to some inconsistency between measurements and simulations.

In Fig. 16 it can be seen that most of the losses occur inwards the machine while in the septum (SS16 and SS31) areas occur outwards, which is consistent with the closed orbit in Fig. 10. It is important to highlight that the BLM in SS09 is located inwards and, thus, measures correctly. That would explain why it is the only one that saturates in that area. Dose measurements on contact done in several points along SS06 to SS10 [14] show that higher doses are found inwards (Fig. 17) in agreement with the simulations performed. The maximum dose appears in SS08 but, as mentioned before, in this case the BLM is placed outwards which could explain why it is not reflected in the BLM measurement. Once more the secondary particles produced by losses in SS08 seem the most probable cause for the saturation of BLM in SS09.

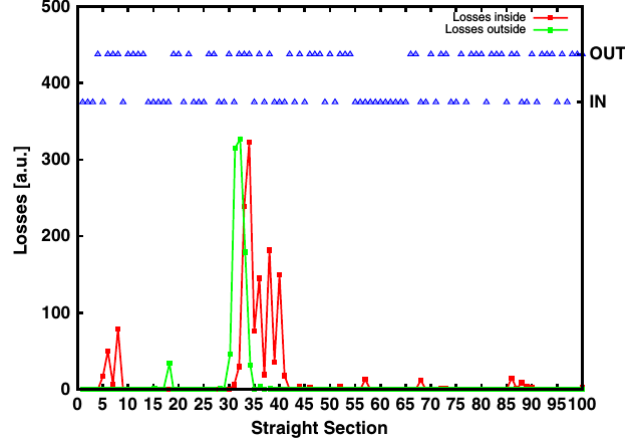


Fig. 16: Beam loss pattern simulation differentiating losses occurred inwards (red) and outwards (green) the machine. In addition, BLMs positions are displayed indicating whether they were installed inside or outside the vacuum chamber. There is a discrepancy in the BLM location and where the losses occurred in SS05-09. BLM in SS09 is located inwards the machine which explains the signal saturation due to the secondary particles produced in SS08.

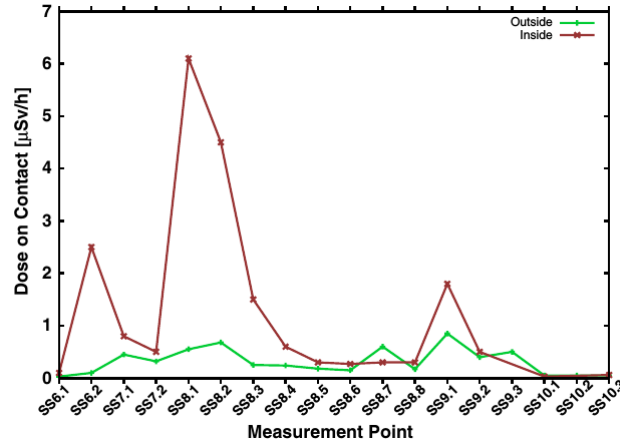


Fig. 17: Dose measurements on contact in several points along SS06 to SS10. In red, doses measured towards inside the ring; in green, towards outside. Losses occur mainly inwards the machine in consistency with simulations in Fig. 16.

4.1 New optics scheme for loss displacement

Previous simulations confirm particle interaction with the septum blade as the source of losses in SS05 to SS10, making them unavoidable unless the extraction process is changed [5]. In the meantime, the only viable solution to reduce the radiological impact of the extraction losses is to displace them in a better-shielded part of the machine, where the tunnel radiation shield thickness is larger. This can be done by installing a new quadrupole 4π in phase advance upstream the quadrupole in SS05 and removing the latter. Then the new extraction optics would be unchanged in the location of the electrostatic and magnetic septa, as shown in Fig. 18. The quadrupole defocusing the scattered particles producing the losses would be now in SS73, bringing the losses into the SS73 to SS78 region. Simulations for this new configuration show a clear displacement of losses from building 151 to the SS73 area as shown in Fig. 19 (top). These losses, as predicted, are scattered particles into the sliced beam (Fig. 19, bottom).

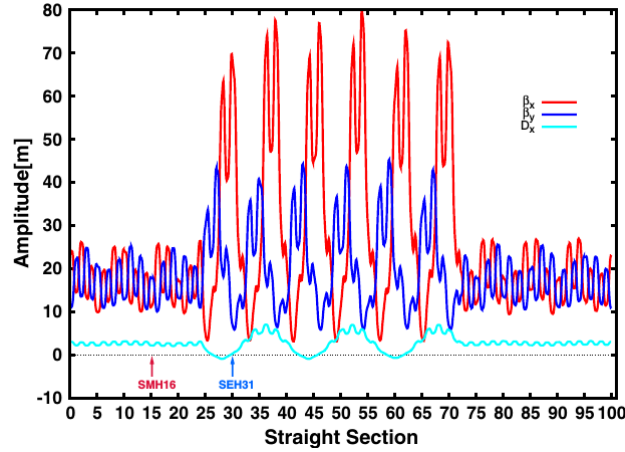


Fig. 18: CT extraction optics for the new QKE16 formed now by quadrupoles in SS25 and SS73. The locations of SMH16 and SEH31 are indicated.

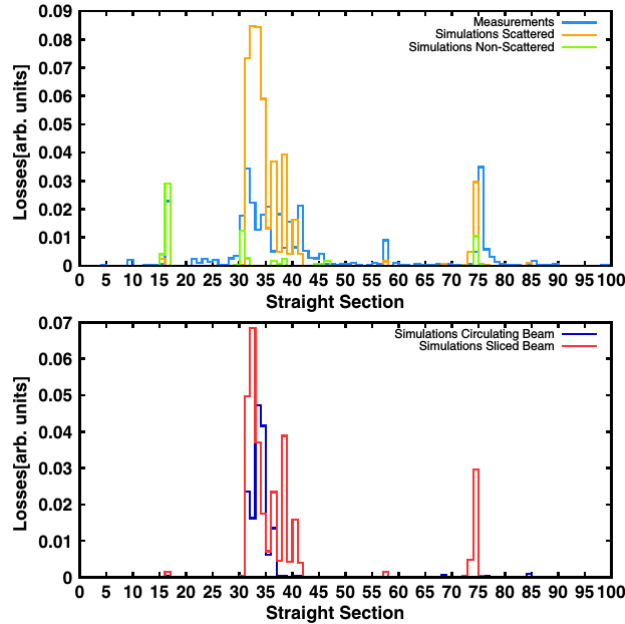


Fig. 19: Beam loss distribution for new CT optics scheme (color legend as in Fig. 15). Beam losses have been displaced from SS05-09 to the SS73 area.

This solution was implemented during the 2007 PS run obtaining the results expected from the simulations. Figure 20 shows the BLMs measurements where the ones in the PS bridge measure almost zero losses, but the one in SS75 saturates; nevertheless the shielding around this region should decrease the dose outside the tunnel. Though a mismatch of one section in SS74 between simulations and measurements can be seen, due to the location of BLM at the entrance of the main magnet that corresponds to the middle of the straight section in the simulations, it is possible that losses occurring at the end of SS74 are measured forward in SS75.

A new radiation survey, similar to the one carried out in 2006, showed that actually the irradiation outside the tunnel has been decreased in the region of the experimental building, whereas a practically

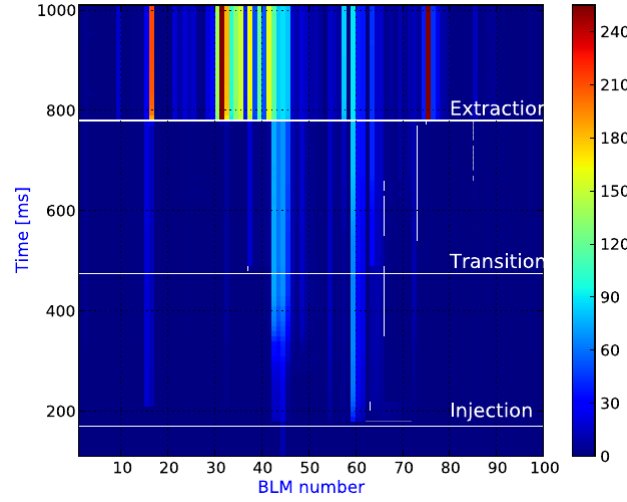


Fig. 20: BLM integrated measurement for the CNGS beam with the new optics scheme. Losses in SS05-09 have been displaced to a better-shielded region in SS75.

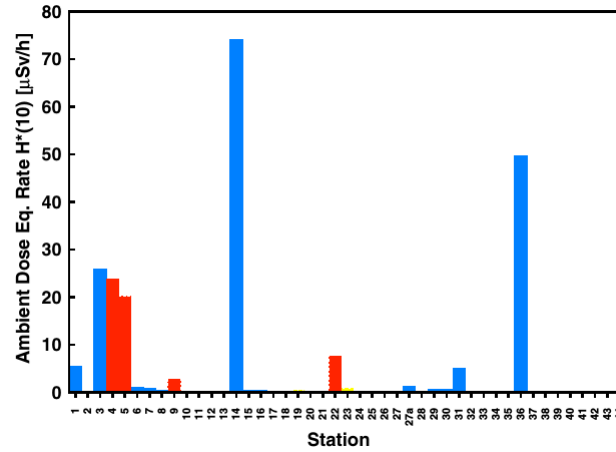


Fig. 21: PS 2007 radiation survey in $\mu\text{Sv/h}$ [2]. The color code is as in Fig. 1. The dose in stations 29 and 30 now fulfills the value guidelines.

negligible increase was observed outside the tunnel near the zone on top of SS75 (station 27a), where most of the losses were concentrated after the quadrupole displacement. The results of the new radiation survey are shown in Fig. 21, where the dose in the inside of the experimental building (stations 29 and 30) was reduced below the guideline values.

5 Conclusions

The present study succeeded to explain the beam loss mechanism producing high radiation doses in the surroundings of the CERN PS building 151 observed during the continuous transfer extraction. The detailed model built for the simulations included five turn tracking with time varying fast kickers, interaction of particles with septum, and loss location by comparison with the PS aperture model. The parameters used to validate the simulations were overall extraction efficiency, turn-by-turn efficiency, and beam loss distributions. However, several issues were mentioned to be taken into account when

comparing measurements and simulations. As these losses are unavoidable, a new optics is proposed to displace them to a better-shielded area. Simulations confirmed the expected results leading to an implementation during the 2007 PS run. A radiation survey in 2007 showed a great reduction in the radiation dose in building 151, changing its classification to supervised radiation area. High radiation doses in stations 22 and 23 are planned to be solved by increasing the shielding in that particular area.

6 Acknowledgements

The authors would like to thank G. Arduini, M. Delrieux, M. Giovannozzi, E. Métral, G. Métral, Y. Papaphilippou, and R. Steerenberg for all the useful discussions, M. Widorski, I. Floret, and the other CERN/RP colleagues for providing the data of the radiation survey, O. Berrig and G. Robert-Demolaize for developing and adapting the PS aperture model used in the simulations, R. Tomás for his help adapting the SIXTRACK code, and the members of the CERN TE/ABT/SE section for the help, expertise, and patience during the loss pattern studies.

References

- [1] <http://op-webtools.web.cern.ch/op-webtools/dokuwiki/doku.php?id=cps-beams>.
- [2] M. Widorski, Accelerator Performance Committee Meeting (APC), 2007.
- [3] K. Elsener *et al.*, CERN AC Note No. (2000-03).
- [4] S. Gilardoni, Accelerator Performance Committee Meeting (APC), 2006.
- [5] M. Giovannozzi *et al.*, Report No. CERN-2006-01.
- [6] <http://mad.web.cern.ch/mad/>.
- [7] G. Robert-Demolaize *et al.*, Report No. CERN-AB-2005-033.
- [8] T. Trenkler and J. B. Jeanneret, Report No. SL/Note 94-105 (AP).
- [9] B. Goddard, in *Handbook of Accelerator Physics and Engineering*, edited by A. W. Chao and M. Tigner (World Scientific, Singapore, 1999), pp. 431–434.
- [10] N. Catalan-Lasheras, Report No. CERN-THESIS-2000-019.
- [11] <http://cern-accelerators-optics.web.cern.ch/cern-accelerators-optics/PSoptics/defaultPS.htm>.
- [12] V. Agoritsas *et al.*, Nucl. Instrum. Methods Phys. Res., Sect. A 247, 44 (1986).
- [13] V. Prieto, Report No. CERN PS/BD/Note 2001-006, in French.
- [14] I. Floret (private communication).
- [15] CERN Safety Code F, Protection Against Ionizing Radiation. Radiation Safety Manual.

FLUKA-SIXTRACK Coupling for the SPS Scrapers

R.B. Appleby¹, F. Cerutti² and A. Mereghetti^{2}*

¹UniMAN, Manchester, UK

²CERN, Geneva, Switzerland

Abstract

Recently, the FLUKA-SIXTRACK coupling has been set up to simulate with great detail the performance of cleaning systems in circular, ultra-relativistic accelerators. In a coupled simulation, the two codes run simultaneously while exchanging particle information; SIXTRACK tracks the beam through the accelerator lattice, whereas FLUKA simulates the interaction of the tracked beam with intercepting devices. Hence, the coupling combines the best of the two codes involved, characterised by a long history of development and benchmarking. The present work summarises the main outcomes of the first extensive application of the coupling to a study case – the SPS scrapers and their upgrade as proposed in the framework of the LIU project.

Keywords

SPS scrapers; collimation system; LIU SPS upgrade; Fluka-SixTrack coupling.

1 Introduction

The performance of collimation systems in circular accelerators can be predicted by numerical simulations capable of combining particle tracking through accelerator lattices and particle-matter interactions. In the last fifteen years, SIXTRACK [1–4] functionalities were extended and constantly updated with an engine for describing the main beam-matter interactions of relevance for simulating collimation systems of ultra-relativistic storage rings [5,6]. In this way, it was possible to establish a solid simulation framework for designing and optimising cleaning systems, e.g. for the Large Hadron Collider (LHC) [7] and for the Relativistic Heavy Ion Collider (RHIC) [8]. Recent developments and applications are extensively reported in these proceedings, giving a complete overview of the ongoing work and key results achieved.

Recently a new simulation tool has been developed, combining the accuracy of SIXTRACK in tracking particles through accelerator lattices and the detailed scattering models of a full Monte Carlo code like FLUKA [9,10]; this is the FLUKA-SIXTRACK coupling [11–14]. The present work gives an overview of the essential functionalities of the new tool (see Sec. 2), the technical aspects of which are described in more details in Ref. [11]. Afterwards, the first extensive application of the coupling is summarised; the studied case is the scraping system [15,16] installed in the Super Proton Synchrotron (SPS), and its upgrade proposed in the framework of the LHC Injectors Upgrade (LIU) project [17,18] (see Secs. 3 and 4). Finally, the robustness and maturity of the tool is probed with an extensive benchmark against measurements from a test of the scraper blades carried out with beam in 2013 (see Sec. 5).

*alessio.mereghetti@cern.ch.

2 The Coupling Between FLUKA and SIXTRACK

In a coupled simulation, FLUKA and SIXTRACK run simultaneously, and exchange particle information through a network port via a TCP/IP protocol managed by the FLUKAIO Application Programming Interface (API) [19].

The most relevant advantage of the FLUKA–SIXTRACK coupling is the deployment of two codes with a long history of development and bench-marking. The coupling allows to combine the refined physics models of particle–matter interactions of FLUKA with the accuracy of SIXTRACK in tracking particles through the accelerator lattice. In this way, multi–turn effects, relevant for collimation systems in circular machines, can be simulated with great detail and accuracy. For such systems, beam particles may need to go through the intercepting devices more than once before being definitively lost, either in the machine aperture or in the intercepting devices; hence, the dynamics by which particles are affected is a combination of single particle beam dynamics and interaction with matter. In addition, the shorter the effective length of the intercepting devices seen by beam particles, the larger the interplay between beam dynamics and physics of scattering. The flow of information is automatic, with the human intervention limited to the simulation set-up; moreover, the use of a network port for exchanging particle data avoids the use of files.

In a coupled simulation, transport through the accelerator lattice is regularly carried out by SIXTRACK until a portion flagged for transport in FLUKA is reached; at that point, beam particles are transferred to FLUKA, transported in its 3D geometry for simulating the interaction with the accelerator components of interest, and given back to SIXTRACK, to continue tracking. The user can flag as many FLUKA insertions as desired. The interfaces to FLUKAIO library on both the FLUKA and SIXTRACK side take care of the change of reference system and units of measurements between the two codes. The FLUKA–SIXTRACK coupling can be enabled with any form of tracking, i.e. in case of a thin or thick lens description of the accelerator, and in case of 4–dimensional or 6–dimensional tracking, including acceleration.

The original FLUKA and SIXTRACK codes have been extended by means of user routines and functions:

- An on–line aperture checking was implemented in SIXTRACK to stop tracking beam particles scattered out of beam–intercepting devices on trajectories not fitting into the mechanical aperture of the machine, avoiding to double count them at subsequent turns. This functionality is at the basis of the estimation of the distribution of losses along the ring, and it is crucial for the correct evaluation of the energy deposition in the beam–intercepting devices (and downstream elements). The check is performed during tracking (“on–line”), element by element and turn by turn;
- In order to take into account the varying distance between the beam and the scraper blades during their sweep (see Sec. 3), FLUKA user routines were coded to deal with moving bodies; similarly, dynamic kicks were implemented in SIXTRACK, to reproduce the change with time of the corrector strengths responsible for the magnetic bump foreseen by the LIU design (see Sec. 3). This implementation has been ported to the release version of SIXTRACK and extended in capability and flexibility [20].

3 The SPS Scrapers and their Upgrade

The SPS is equipped with scrapers, routinely used to ensure a clean injection into the LHC. They are installed in the Long Straight Section 1 (LSS1) of the SPS, which also hosts the beam dumping system¹. The scrapers are composed by movable graphite blades, 1 cm in length (see Fig. 1, left frame); one per plane, they are swept through the beam to remove tails at the desired transverse position. Scraping is performed at the end of the ramp, immediately before beam transfer to the LHC, in order to minimise

¹As it will seen later (see Sec. 5.1), this fact has consequences on loss measurements in the region.

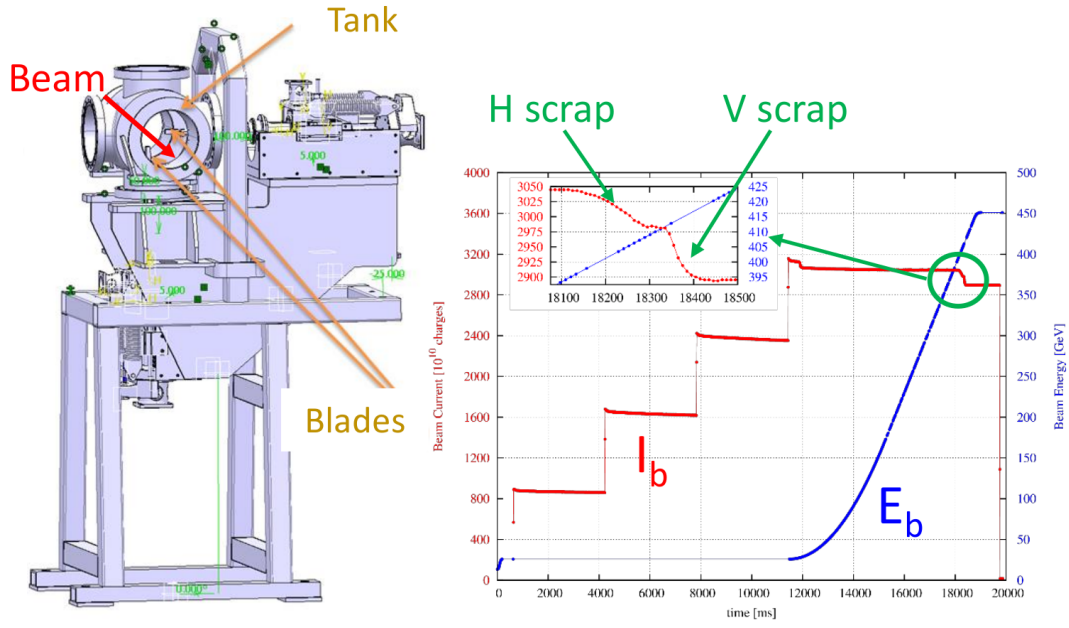


Fig. 1: Left frame: 3D view of the scraper assembly. Right frame: typical patterns of beam current and energy during an SPS cycle. The zoom shows the tiny intensity decrease due to tail scraping on both the horizontal and the vertical plane as done during regular operation.

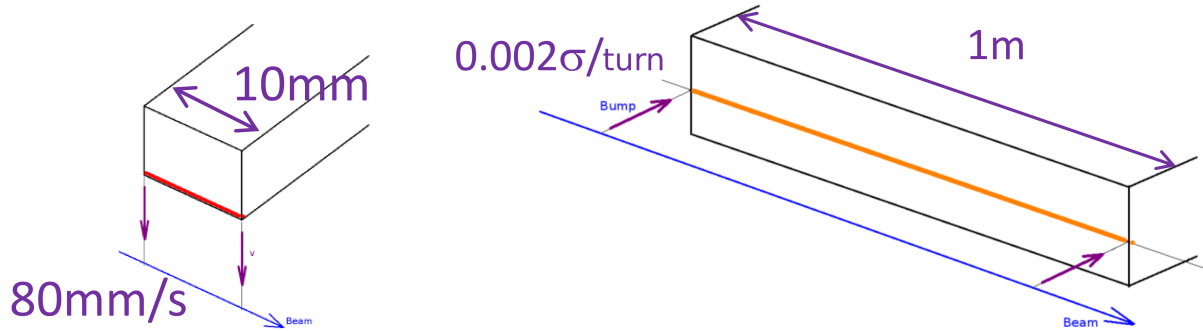


Fig. 2: Schematics of the present mechanical scrapers (left frame) and those proposed as upgrade (right frame). The sketch shows the beam trajectory, the relative beam–absorber movement with its speed, and essential dimensions of the devices.

tail re–population. Hence, they are kept off the beam for most of the SPS cycle and only at its end they perform the necessary cleaning action (see Fig. 1, right frame). As such, they are fast betatron scraping devices.

In the context of the LIU project, it was proposed to study a possible upgrade of the SPS scraping system from a thin blade mechanically swept through the beam to a bulk absorber block against which the beam is steered by means of a magnetic bump (see Fig. 2). The study was triggered not to overcome a specific limitation of the present system, but as a possible alternative to improve flexibility and limit secondary radiation escaping the beam–intercepting device, and hence induced radioactivity. In fact, though being more complex than the present one, the new design would overcome some limitations intrinsic in the design of the system presently in operation:

Table 1: Beam parameters used in simulation: normalised emittance (ϵ_N), standard deviation of the momentum distribution (σ_δ), bunch population (N_b), total beam intensity (N_p), total stored energy (E_T), number of bunches (n_b), and beam energy (E_b).

Scrapper Design	ϵ_N [μm]	σ_δ [10^{-4}]	N_b [10^{11}]	N_p [10^{13}]	E_T [MJ]	n_b []	E_b [GeV]
Present	1	1	1.15	3.31	2.38	288	450
LIU	2.5	5.25	2.5	7.2	5.52		

1. it would allow to better control the beam–impact conditions at the absorber;
2. it would imply no mechanical movements, with consequent no mechanical wear of components.
3. it would make cleaning more efficient, since a longer absorber increases the probability for a proton to undergo a nuclear inelastic interaction at the first encounter, and hence being ultimately absorbed by the system;
4. it would allow to have softer spectra of particles escaping the absorber.

The first two bullets come from the choice of deploying a magnetic bump, whereas the other two come from the use of a long absorber block. As for the present scrapers, the new system would be triggered at the end of the SPS cycle, raising the bump until scraping is accomplished. The new system would be installed in the SPS LSS6.

4 Comparison of Systems

In order to characterise the two systems, a large simulation campaign was carried out [12]. Numerical simulations were performed with the FLUKA–SIXTRACK coupling (see Sec. 2), set up on purpose for these studies. The relevant observables are the density of energy deposition in the material interacting with the beam (see Sec. 4.1), the evolution with time of the beam current during scraping (see Sec. 4.2), and the distribution of losses along the ring (see Sec. 4.3). Results refer to the case of 0σ scraping, i.e. when the whole beam is scraped away (including the core) with the blade swept at the beam closed orbit. This is taken as design case, even if it is an extreme one, since it maximises the energy deposition per beam proton [12] even though it is neither an operational case, as the scrapers are meant to scrape away tails, nor an accidental scenario. Hence, the simulated case represent a potential accident scenario, where by operational mistake the full beam is totally scraped at full intensity. Scraping on the horizontal plane is considered.

Simulations consider full beam intensity (i.e. 288 bunches), with Gaussian transverse and momentum distributions. The considered beam parameters are reported in Table 1. It should be noted that, while the number of bunches always remains 288 per injection into the LHC, the highest bunch population at SPS extraction considered for the LIU design is more than twice the one of nominal LHC beams available in the SPS at the time of the study [18].

4.1 Energy Deposition

Figure 3 shows a 3D view of the energy deposition maps superimposed to the geometry of the scraper blade and of the LIU scraper absorber. The distribution of the energy deposition is collapsed on the very first layers of material directly impacted by the beam on the plane of movement; on the other plane, the distribution keeps memory of the original beam size, implying a dependence on machine optics and beam parameters.

Table 2 reports the maximum values of expected energy deposition. Values per impacting proton for the LIU system are more than a factor 3 lower than those for the present system, mainly because of the different designs [12]. The values in kJ/cm^3 are obtained from those in GeV/cm^3 taking into

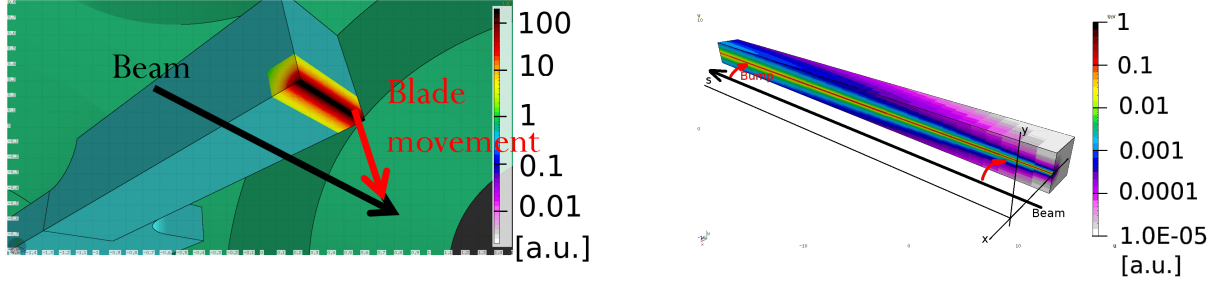


Fig. 3: 3D view of the energy deposition maps superimposed to the geometry of the scraper blade (left frame) and of the LIU scraper absorber (right frame) proposed as upgrade. The views have been produced with FLAIR [21]. The relative movement between the beam closed orbit and the absorbing devices is also shown.

Table 2: Peak energy deposition in the absorbing material of the scrapers in case of 0σ scraping at full beam intensity. Values are given per impacting proton (second column) and per per bunch train and the respective bunch populations (Tab. 1).

Scraper Design	E_{\max}	
	[GeV/cm ³]	[kJ/cm ³]
Present	30–35	160–185
LIU	10–11	115–130

account 288 bunches and the respective bunch populations, reported in Tab. 1; hence, the factor 3 of difference is decreased to 40 %. The absolute numbers should be compared with the reference value of 12.8 kJ/cm³, equivalent to the energy density necessary to locally reach the sublimation temperature of graphite of 3600°C². The values of maximum energy deposition show the severity of the simulated scenarios, implying a local damage of the absorbing material. The maximum value found for the LIU scraper increases by 50 % when the normalised emittance is decreased to 1 μ m.

4.2 Evolution of Beam Intensity with Time

Figure 4 shows the time profile of the beam intensity during scraping at 0σ for the two systems. The profile of the present system is dominated by the beam dimensions and the blade speed for a substantial fraction of time; once the blade reaches the centre of the beam, an exponential attenuation is seen and the material density and the blade thickness become the key parameters. In fact, with an inelastic interaction length of 45 cm in graphite at SPS top energy (i.e. 450 GeV, see Tab. 1), a beam proton requires on average 45 turns before undergoing an inelastic event and being hence absorbed.

The profile of the upgraded system shows a linear dependence on the speed with which the magnetic bump is raised. This is expected, because with a 1 m–long graphite absorber, a proton at SPS flat top energy (i.e. 450 GeV, see Tab. 1) has a high chance to undergo a nuclear inelastic event with a single passage through the absorber.

As a matter of fact, with the present operational parameters of the two systems (see Fig. 2), there is no difference on the time required to fully accomplish scraping, as the beam is reduced to a few % in ~ 1000 –1500 turns for both systems. For the present system, this time scale is the result of a combination of two factors: on the one hand, the time required by the blade to reach the centre of the beam (i.e. to have all beam protons at reach) determined by the speed of the blade; on the other hand, the average time

²This value was calculated simply integrating the specific heat as a function of temperature.

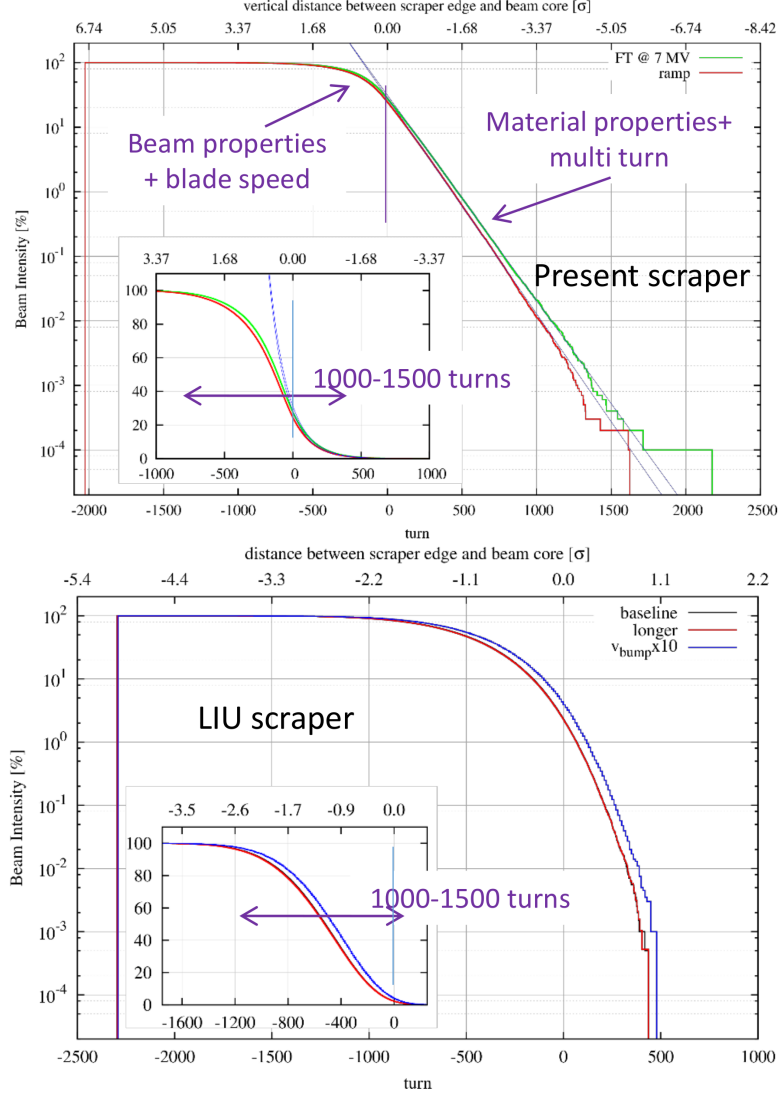


Fig. 4: Time profile of the beam current during scraping in case of the present scraping system (upper frame) and in case of the LIU design (lower frame) for 0σ scraping. The origin of the horizontal axes marks the moment when the cutting edge of the absorber reaches the position of the closed orbit. The green curve in the upper frame refers to simulations run at flat top energy with the nominal 7 MV RF voltage, whereas the red curve was obtained simulating the ramp, at around 400 GeV. The black curve in the lower frame refers to the baseline upgraded system (see Sec. 3), whereas the red curve has been obtained with the absorber twice in length. The blue curve in the same frame refers to the case with the magnetic bump 10 times faster; for the sake of clarity, the horizontal scale has been expanded by the same factor.

necessary to a proton to undergo a scattering event leading to its loss. For the upgraded system, the time scale is given by the speed with which the bump is raised.

The upper frame of Fig. 4 also shows the case of scraping towards the end of the ramp, as it happens in reality; the curve was obtained taking into account acceleration in SIXTRACK. Scraping starts earlier, due to the slightly larger spot size at lower beam energy; moreover, the exponential part is slightly steeper, in accordance to a slightly larger beam emittance, implying trajectories closer to the machine aperture and larger impact angles (and hence longer paths in the blade), and to larger scattering angles. The lower frame also shows that a longer absorber for the LIU design does not reduce the time

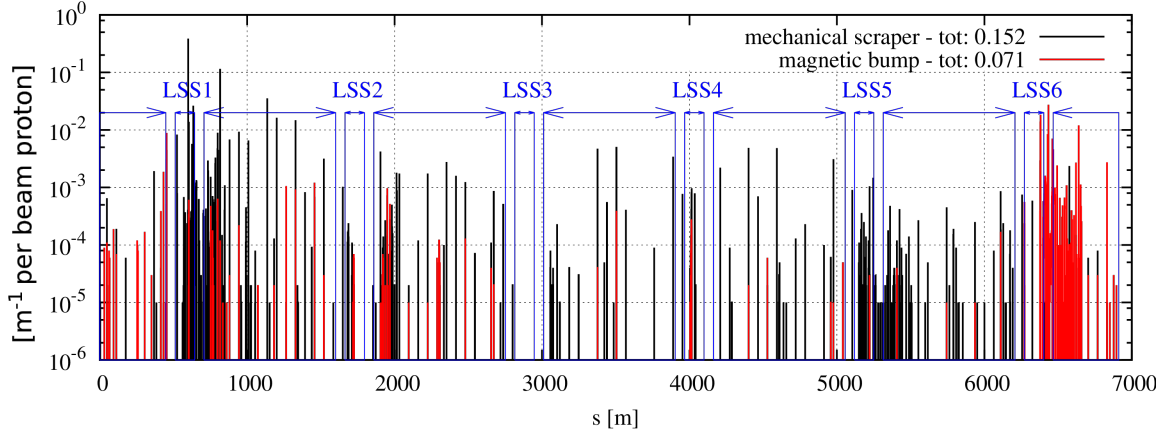


Fig. 5: Distribution of losses along the SPS ring during scraping for the two systems; only protons lost with $\delta < 2\%$ are shown. The key in the plot reports the fraction of protons lost in the ring.

for scraping.

4.3 Loss Distribution

Figure 5 shows the distribution of losses along the SPS ring during scraping for the two systems; the shown distribution takes into account only lost protons with $\delta < 2\%$. Losses concentrate in the cells immediately downstream of the scraping devices, including the first cells of the downstream arcs. Local peaks are also found further into the arcs. As a matter of fact, the LIU design decreases the overall losses by a factor 2.

5 The Robustness Test and the Benchmark of Simulation Results

The very harsh conditions of energy deposition predicted by simulations (see Sec. 4) triggered the request of a test with beam where these conditions could be reproduced and values verified (see Sec. 5.1). After the test, the blades were dismantled for detecting possible damage.

The test was also the occasion for an extensive benchmark of the simulation tool (see Sec. 5.2); the main observables are the time evolution of the beam current as recorded by the Beam Current Transformer (BCT) and the loss pattern measured by the Beam Loss Monitors (BLMs) during set up of the test.

5.1 The Robustness Test

The test took place on 16th February 2013, at the end of the SPS activity during the LHC Run I. To probe their endurance, the blades were probed with scraping the whole beam (i.e. 0σ scraping) at full intensity (see Tab. 1, “present” scraper design), since this is the scenario for which the peak energy deposition predicted by simulation is the highest (see Sec. 4). The test was performed at 450 GeV, with a total beam intensity of $3.1 \cdot 10^{10}$ charges. The spare scrapers were used in order to more easily disentangle the damage due to the test from possible damage due to regular operation. Both planes were probed independently, one after the other one.

At the very moment of the test, extremely high vacuum spikes were recorded at the vacuum gauges immediately upstream of the scrapers; the spikes were three orders of magnitude larger than the regular values with circulating beam. Such a vacuum deterioration is sign of material emission by the scraper blades. Optical scans of the surface of the blades [22] revealed an increase in the porosity of the material, from a value of $\sim 10\%$ far from the blade edge up to $\sim 30\%$ in the region nearby the edge.

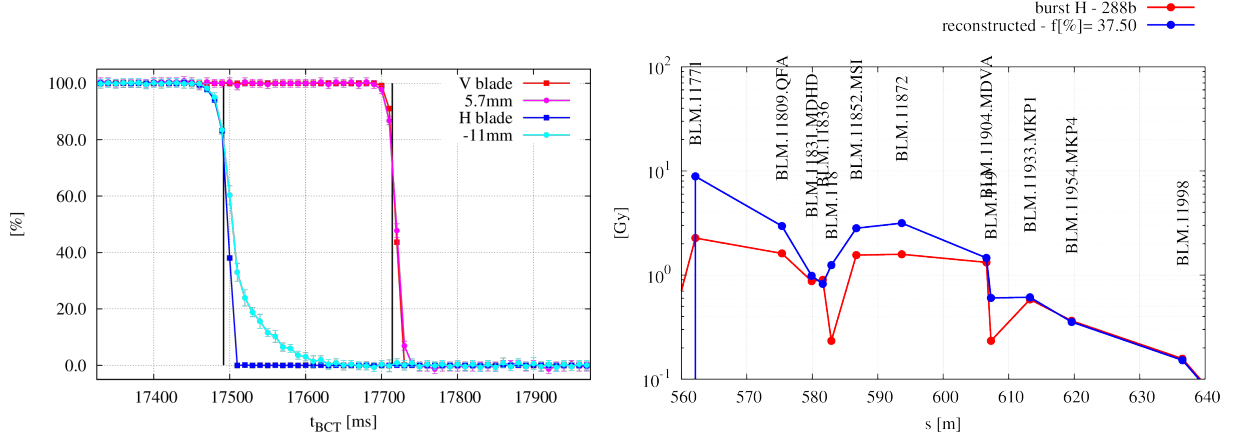


Fig. 6: Left frame: time profile of the normalised BCT signal for full beam scraping during the test of each blade (red and blue curves) and in case of scraping at the same transverse position with low-intensity beams (magenta and cyan curves). The vertical black lines mark the timing of the dump as from logging. Right frame: BLM pattern recorded during the robustness test of the horizontal blade (red curve) and the one reconstructed with Eq. 1 (with signals from pure scraping and pure dumping events, blue curve), which matches the last three BLMs in LSS1. The fraction reported in the legend of the blue curve reports the amount of scraped beam necessary for the matching.

Further analyses [12] showed that the beam was prematurely dumped while testing the blades by the BLM at the scraper. Secondary particle showers generated by proton inelastic interactions in the blades were so intense that the dump threshold was reached before scraping was completed. In fact, the time profiles of the normalised BCT signal while scraping during the tests are different from those taken with low-intensity beams and the same scraping settings (see Fig. 6, left frame); the difference is clearly visible for the horizontal blade, whereas for the vertical blade the difference is less evident, though still present.

The same conclusion can be drawn from the signals of the LSS1 BLMs (see Fig. 6, right frame). In fact, the patterns from a pure dump event and a pure scraping event can be linearly combined to reconstruct the pattern measured during the tests. The linear combination can be written as

$$R_{i,\text{test}} = sR_{i,\text{scrp}} + (1 - s)R_{i,\text{dump}}, \quad (1)$$

where R_i represents the signal from the i -th BLM, s is the amount of scraped beam, and the subscripts “test”, “scrp” and “dump” refer to the signals recorded during the test, a pure scraping event and a pure dump event, respectively. The linear combination is necessary since the values logged by the SPS BLMs electronics are integrated over the entire SPS cycle; hence, in case of partial scraping, the BLM pattern in LSS1 reflects not only the action of scraping, but also the action of dumping the circulating beam, since LSS1 hosts also the beam dumping system (see Sec. 3). Moreover, the electronics of most of the LSS1 BLMs were at saturation during the test; therefore, the proper matching between the recorded pattern and the one reconstructed via Eq. 1 is established only for those BLMs surely out of saturation, i.e. the last three in LSS1.

A premature dump implies a load on the blades lower than the one predicted by simulations. Table 3 reports the amount of beam scraped during the robustness test of each blade as estimated with BCT measurements and BLM measurements. In the case of the former set of measurements, the amount of scraped beam corresponds to the value of the normalised BCT profile with low-intensity beams at the time of dump; for the latter set of measurements, the amount of scraped beam is found when the readouts of the last three BLMs in LSS1 are matched by the profile reconstructed with Eq. 1. The two methods give quite different estimations; while the estimation via the BCT is affected by the limited BCT

Table 3: Amount of beam scraped during the robustness test of each blade as estimated via BCT measurements and BLM measurements (see Fig. 6). The consequent energy density in the most loaded point of the beam is reported as well.

Tested Blade	BCT-based		BLM-based	
	[%]	[kJ/cm ³]	[%]	[kJ/cm ³]
H	~20	20–24	38	37–46
V	~30	27–37	44	38–53

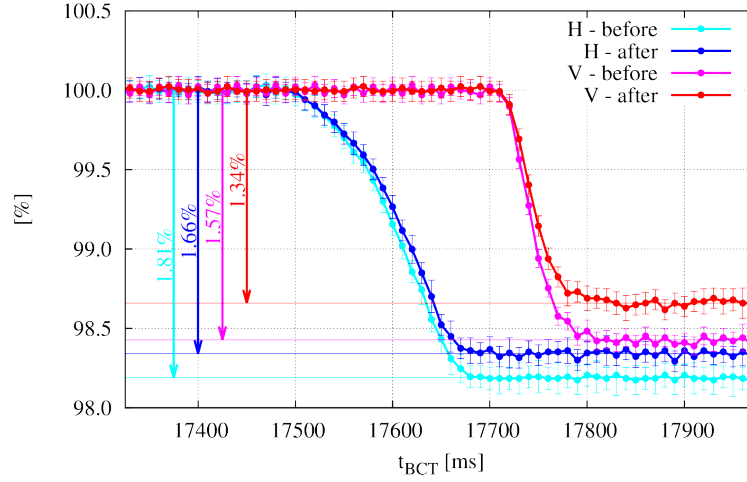


Fig. 7: Time profiles of the normalised BCT signal for regular scraping before and after the robustness test.

bandwidth (3 dB at 50 Hz) and the poor sampling rate (10 Hz) with respect to the timing of scraping, the one via the BLMs seems to be more reliable, since it is not affected by any timing issue and it is based on integral quantities measured by not saturated monitors. It can be concluded that not even half of the beam was scraped away before dumping. The energy density in the most loaded point of the blade is reported as well in Tab. 3. Even in the case of the earliest dump (i.e. for the horizontal blade), values are much larger than the heat required to locally sublimate graphite, in accordance with the vacuum spikes registered during the tests and the aforementioned crystallographic analyses.

The cleaning performance of the blades for regular scraping was checked to spot any loss of performance following the induced damage. Therefore, the time profiles of the normalised BCT signal for regular scraping before and after the robustness test were compared (see Fig. 7); very minor differences were found, most probably due to a change in beam profile or emittance, or a drift in the closed orbit [12].

5.2 The Benchmark of Simulation Results

The robustness test was also taken as occasion for an extensive benchmark of the simulation tool. BCT readouts and BLM measurements during the set up of the test were used, recorded when scraping with each tested blade at different transverse positions. Measurements were taken with low-intensity beams in order not to deal with saturated BLM readouts.

The benchmark against the normalised BCT signals allowed to reconstruct the actual speed and angle of the blades during the tests; in fact, these parameters affect the time profile of the beam current during scraping (see Fig. 8). Different values were tested in simulation, until a reasonable matching of the profiles with all scraping settings was found. Table 4 summarises the range of values used in

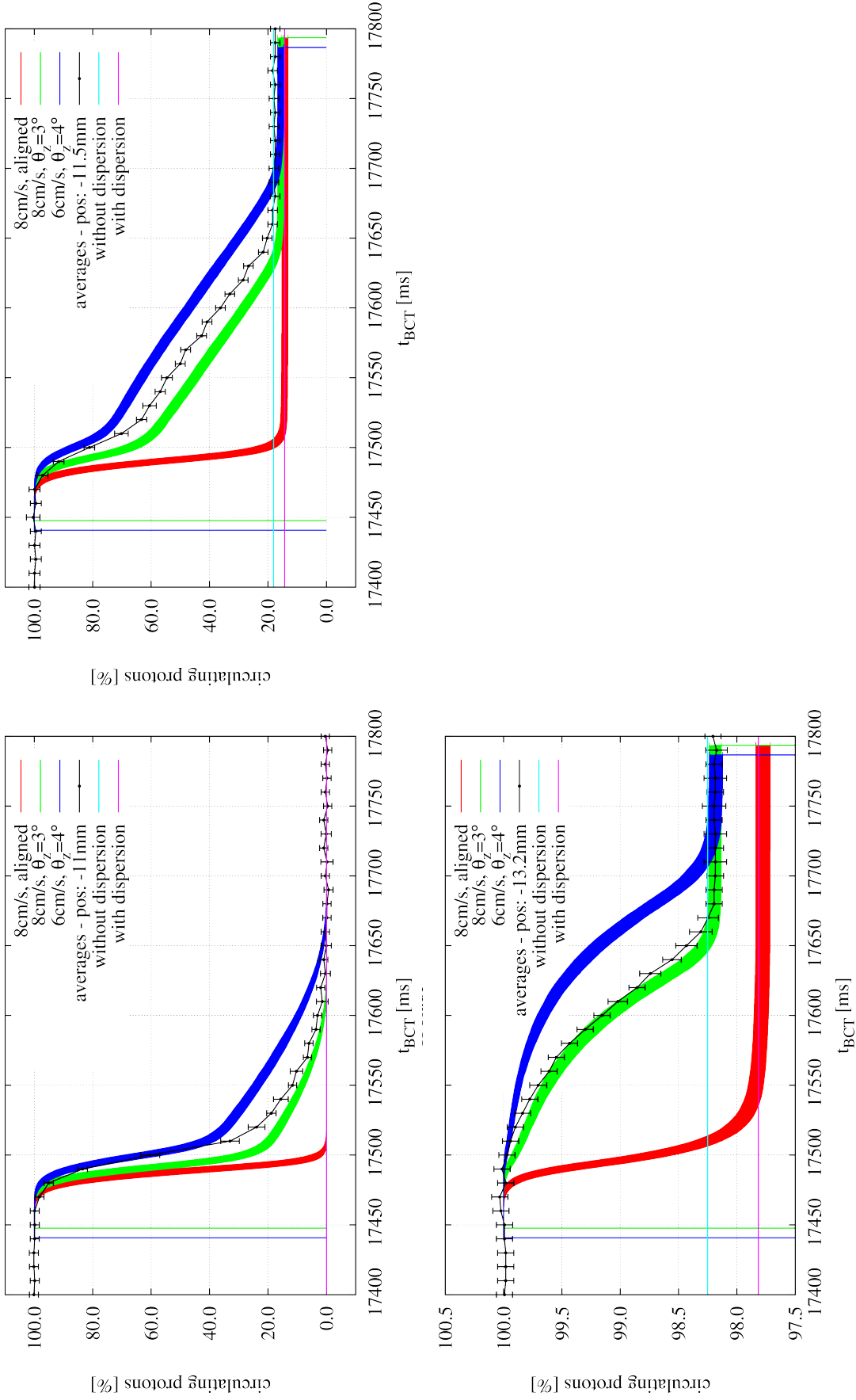


Fig. 8: Time profiles of the normalised beam intensity measured by the BCT (black points) and as predicted by simulation for full beam scraping (0 σ scraping, upper-left frame) intermediate scraping (i.e. at 500 μm from the beam centre, upper-right frame) and regular scraping (i.e. at 2.2 mm, lower-left frame). The case of the horizontal blade is shown, whereas the one of the vertical blade can be found in Ref. [12]. In each frame, the black curves show averages over consecutive fills with the same beam intensity and scraping settings, and the error bars show the 1 σ dispersion of data. Three combinations of blade speed and tilt are shown, and the error bars show the statistical error only. The magenta and cyan lines show the beam intensity after scraping predicted analytically in case dispersion effects are and are not taken into account, respectively.

Table 4: Speed and angle of the tested scraper blades as reconstructed with the benchmark against normalised BCT profiles (see Fig. 8). The nominal values are reported as well.

	Speed [cm/s]	Angle [°]
H	6–8	3–4
V	<6	<0.5
nominal	8	0

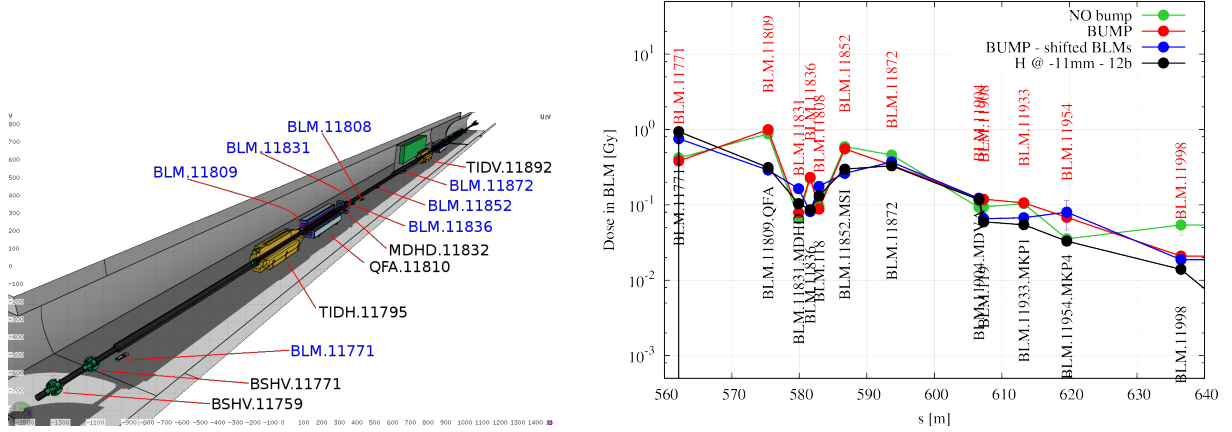


Fig. 9: Left frame: 3D view of the FLUKA model of the SPS LSS1, downstream of the scrapers. The view was generated with FLAIR [21]. Right frame: BLM signals as measured during full beam scraping at low intensity (black line) and as reproduced by simulations in case the permanent magnetic bump in the SPS LSS1 is taken into account (red curve) or not (green curve). Simulation results with transverse offsets applied to BLMs are shown as well (blue curve).

simulation giving the best matching to measurements, along with the nominal values.

The benchmark against BLM readouts required to describe in FLUKA the SPS LSS1 downstream of the scrapers (see Fig. 9, left frame). In fact, the readouts reflect the development of secondary particle cascades, started by inelastic events taking place in the scraper blades or by beam protons hitting the machine aperture. Therefore, details regarding geometry and materials of the elements nearby the monitors reached by the cascades are relevant for the purpose of the bench-marking. As it can be seen (see Fig. 9, right frame), simulation results dramatically depend on the transverse position of the BLMs. Because of the non-perfect agreement between technical drawings, the BLMs were simulated also with transverse offsets from their nominal positions, in order to give a hint of the sensitivity (red vs blue curves in the same figure). Moreover, the effect of a permanent magnetic bump³ on BLM readouts has been addressed as well (red vs green curves in the same figure).

6 Conclusions

The FLUKA-SIXTRACK coupling allows to simulate with great detail the performance of cleaning systems in circular machines in a large variety of applications. This is possible thanks to the accuracy of the two codes involved, characterised by a long history of development and bench-marking. The present work has summarised the main outcomes of the first extensive application of the coupling to a study case,

³ A permanent magnetic bump is present in the SPS LSS1 to ease dumping. The bump is achieved offsetting three consecutive main quadrupoles.

i.e. the SPS scrapers and their upgrade as proposed in the framework of the LIU project.

The two systems were compared in terms of energy deposition in the absorbing medium, evolution of beam intensity with time during scraping, and losses induced around the ring. The focus was in particular on the case of scraping at 0σ , as this scenario is characterised by the highest density of energy deposition in the absorbing material. Despite its complexity, the upgraded system is characterised by lower loads on the absorbing material and lower losses around the ring; moreover, thanks to its conceptual design, it intrinsically offers more control on the impact conditions of the beam on the absorber during scraping.

A robustness test of the system presently installed was carried out at the end of the SPS activity in 2013, to verify damage levels in the blades. The test also gave the opportunity for an extended benchmark of the simulation tool against BCT readouts during scraping and BLM signals downstream of the scrapers. The positive outcomes of the benchmark indicate the maturity and accuracy of the simulation tool.

Acknowledgements

We wish to thank K. Cornelis, O. Mete, V. Kain and the SPS operation team for setting up and conducting the robustness test.

References

- [1] F. Schmidt, SIXTRACK Version 4.2.16: Single Particle Tracking Code Treating Transverse Motion with Synchrotron Oscillations in a Symplectic Manner; User's Reference Manual, CERN-SL-94-56(-AP), CERN, Geneva, Switzerland (1994).
- [2] G. Ripken and F. Schmidt, A Symplectic Six-Dimensional Thin-Lens Formalism for Tracking, CERN-SL-95-12(-AP), DESY-95-063, CERN, Geneva, Switzerland (1995).
- [3] SIXTRACK website: <http://sixtrack.web.cern.ch/SixTrack>.
- [4] R. De Maria *et al.*, these proceedings.
- [5] G. Robert-Demolaize, R.W. Assmann, S. Redaelli, F. Schmidt, Proc. Particle Accelerator Conf. (PAC'05), Knoxville, TN, USA, 2005, paper FPAT081, pp. 4084-4087.
- [6] R. Bruce *et al.*, these proceedings.
- [7] R.W. Assmann *et al.*, in Proc. 10th European Particle Accelerator Conf. (EPAC'06), Edinburgh, Scotland, UK, Jun 2006, paper TUODFI01, pp. 986-988.
- [8] G. Robert-Demolaize and A. Drees, these proceedings.
- [9] T.T. Bohlen *et al.*, The FLUKA Code: Developments and Challenges for High Energy and Medical Applications, Nuclear Data Sheets 120, 211-214 (2014)
- [10] A. Ferrari *et al.*, FLUKA: a multi-particle transport code, CERN-2005-10 (2005), INFN/TC_05/11, SLAC-R-773.
- [11] V. Vlachoudis *et al.*, these proceedings.
- [12] A. Mereghetti, Performance Evaluation of the SPS Scraping System in View of the High Luminosity LHC, Ph. D. thesis, UniMAN, Manchester, UK (2015).
- [13] Twiki of the FLUKA-SIXTRACK coupling: <https://twiki.cern.ch/twiki/bin/view/FlukaTeam/CouplingSVNRepositories>.
- [14] SVN repository of the FLUKA-SIXTRACK coupling code (CERN credentials required): http://svn.cern.ch/repos/fluka_coupling.
- [15] H. Burkhardt and R. Schmidt, Intensity and Luminosity after Beam Scraping, CERN-AB-2004-032-ABP (2004).

- [16] G. Arduini and H. Burkhardt, Transverse Tail Scraping in the SPS for Clean LHC Injection, EDMS doc. 772782 (2006).
- [17] LIU website: espace.cern.ch/liu-project.
- [18] J. Coupard *et al.* (Eds.), LHC Injectors Upgrade – Technical Design Report – Volume I: Protons, CERN-ACC-2014-0337 (2014).
- [19] Twiki of the FLUKAIO protocol: twiki.cern.ch/twiki/bin/view/FlukaTeam/FlukaIO
- [20] K. Sjobaek, these proceedings.
- [21] V. Vlachoudis, in Proceedings of the Conf. on Mathematics, Computational Methods & Reactor Physics (M&C 2009), Saratoga Springs, New York (2009).
- [22] F. Léaux, Microscopic Observations of Irradiated Graphite Scraper Blades, EDMS. 1339153, CERN, Geneva, Switzerland (2014).

