Merlin scattering models for the HL-LHC collimation system

S. Tygier[12], R.B. Appleby[12], R.J.Barlow[3], H. Rafique[12], and S.Rowan[3]

[1]University of Manchester
[2]Cockcroft Institute
[3]University of Huddesfield

October 6, 2017

Abstract

The high stored energy of the High Luminosity LHC beam requires an effective multilevel collimation system to in order to protect the machine from damage. We use MERLIN to perform simulations of the collimation system and to predict the locations of beam losses around the ring. MERLIN contains advanced scattering algorithms to model the scattering of protons in matter, as well as a simpler model based on Sixtrack/K2. In this article we compare the scattering algorithms for loss maps in the HL-LHC.

1 Introduction

The High Luminosity upgrade to the LHC [1] will increase the data taking rate by an order of magnitude in order to further its physics reach. This is achieved by a number of measures including increasing the beam intensity and reducing the β^* value at the interaction points.

To protect the machine a multilevel collimation system is needed. At such high beam intensities it takes only a small fraction of the beam impacting accelerator components to cause excess heating, irradiation or damage. Simulation code that can accurately model the scattering of particles in the collimator jaws and the transport of the particles that scatter back in to the beam pipe, is used to design and validate this collimation system.

Merlin [2, 3] is a particle accelerator simulation library that can model particles in high energy synchrotrons such as the LHC. It is designed to be modular, allowing the simple addition of new physics processes. It has a scattering process for modelling collimators, which includes the proton's interactions with the electrons (Multiple coulomb scattering and ionization energy loss) and nuclei (elastic, diffracting and inelastic) in the collimator jaw. To ensure high performance on the leading proton is considered, i.e. all secondary are assumed to be lost within a small distance.

2 HL-LHC lattice

The HL-LHC introduces new inner triplet magnets with higher gradients and larger apertures which allow smaller β^* values to be achieved at IP1 (ATLAS) and IP5 (CMS). It also makes use of the Achromatic Telescopic Squeeze (ATS) scheme, which uses magnets outside the IR to contribute to the squeeze while controlling chromatic aberrations [4].

For these simulations the HL-LHC 1.2 lattice was used with β^* at IP1/5 set to 15 cm in each plane (round optics). In fig. 1 the larger β function in the arcs adjacent to the IP1/5 are visible. Figure 2 shows a zoomed section around IP1.

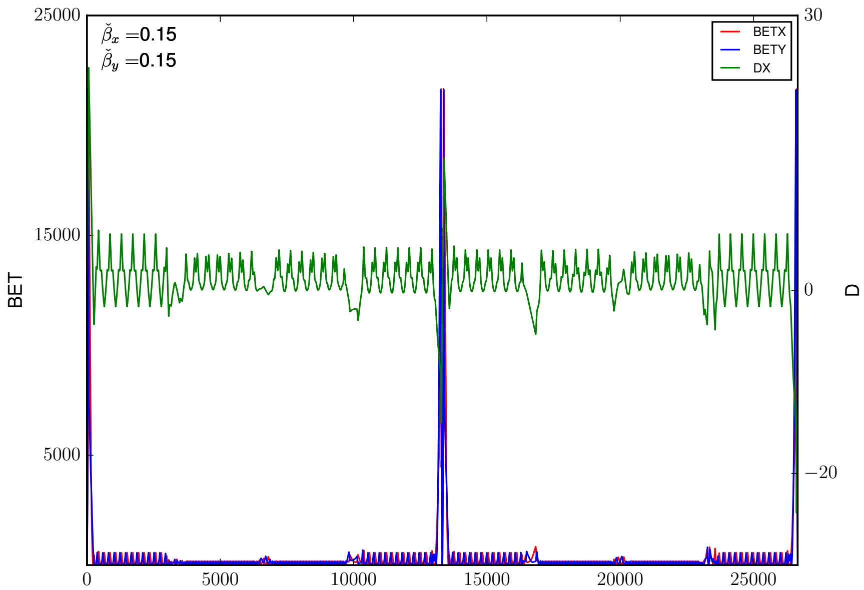

Figure 1: Squeezed HL-LHC optics.

The HL-LHC contains a new aperture shape compared to the LHC, octagonal apertures in the inner triplets. These have been implemented in MERLIN in the `OctagonalAperture` and `InterpolatedOctagonalAperture` classes. The octagon is specified by its width, height and the angles to the corners of the diagonal part.

3 Scattering models

The high energy protons in the LHC can often survive passing through a collimator jaw, with only a small energy loss and deflection. This allows them to carry on circulating in the beam pipe, potentially for many tens of turns. Accurate modelling of the energy loss and deflection is vital to predict the final

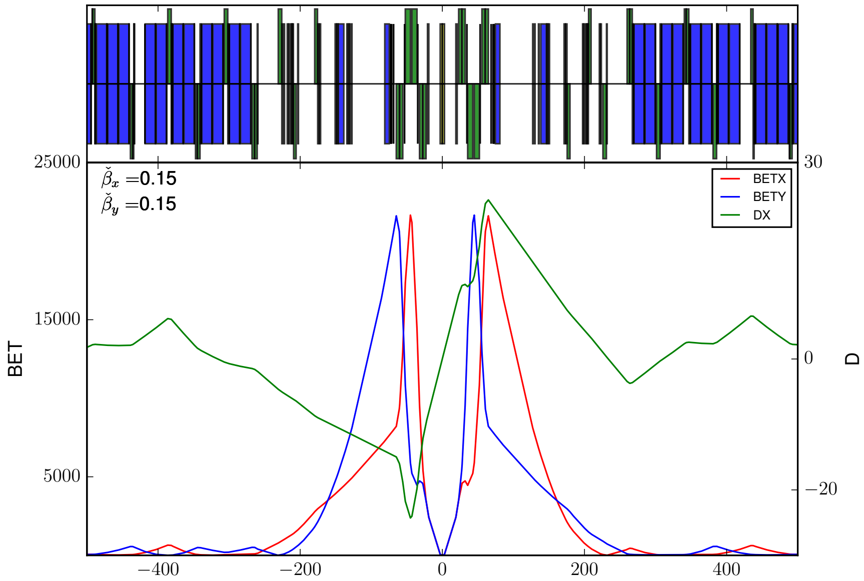

Figure 2: Squeezed HL-LHC optics around IR1 (ATLAS).

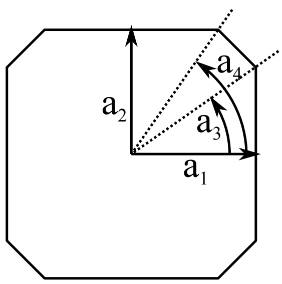

Figure 3: Parameters for octagonal aperture.

loss locations of these scattered particles.

MERLIN implements two versions of the scattering model. The first based on the K2 [5] model from SixTrack [6], the current leading code for LHC collimation studies, and a new model based on the Donnachie and Landshoff description of Pomeron and Reggeon exchange [7]. MERLIN also features three in between models, using K2 plus the new version of each of elastic, single diffractive and ionzation, which can be used to assess the relative contributions of the upgrades.

The implementation of the advanced scattering models is discussed in detail in [8]. Since then there has been some re-factoring of the code. The collimation process in the class `CollimateProtonProcess` is called at each element, and checks whether particles are within the aperture (which may be interpolated), using the `PointInside()` method of the relevant geometry derived from the `Aperture` class. If the particle is within a collimator jaw the `CollimateProtonProcess::DoScatter()` method is used perform the scattering by making calls to a `ScatteringModel` class. Each of the above scattering models is implemented as a derived class, as shown in fig. 4. Each derived model is composed of a set of `ScatteringProcess` classes, as shown in fig. 5.

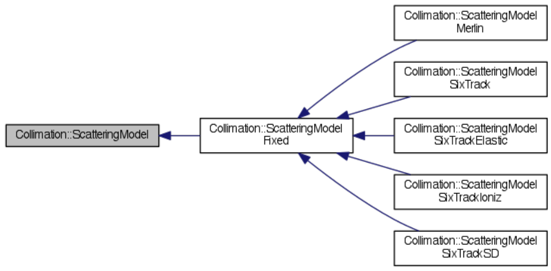

Figure 4: Class hierarchy for ScatteringModel.

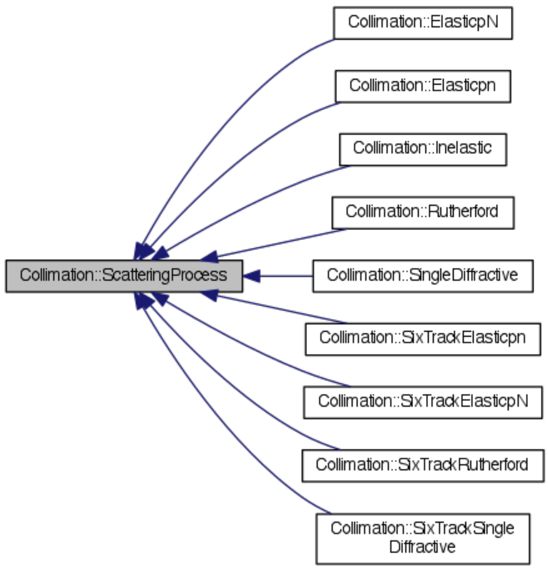

Figure 5: Class hierarchy for ScatteringProcess.

If the proton is not absorbed in the collimator, i.e. it energy has not fallen below a threshold energy it is placed back in to the beam and will continue to be tracked around the ring. Lost particles are recorded using the `CollimationOutput` class.

4 Loss maps

Loss maps show the distribution of proton losses around the ring. For simulation we assume that losses are dominated by particles slowly drifting from the halo on to the primary collimators. We therefore start the simulation with a halo beam that hits the primary collimator with a small impact factor of 1 µm. The scattered particles are then tracked for 200 turns. We use the scattering models described above when a proton interacts with a collimator jaw and consider other sections of beam pipe to be black absorbers. Losses are recorded in 10 cm bins.

Figure 6 shows the losses for a horizontal halo in beam 1 using the MERLIN scattering model. Losses in collimators are shown in black, warm magnets in red and cold magnets in blue. The same simulation using the Sixtrack/K2 style scattering is shown in fig. 7.

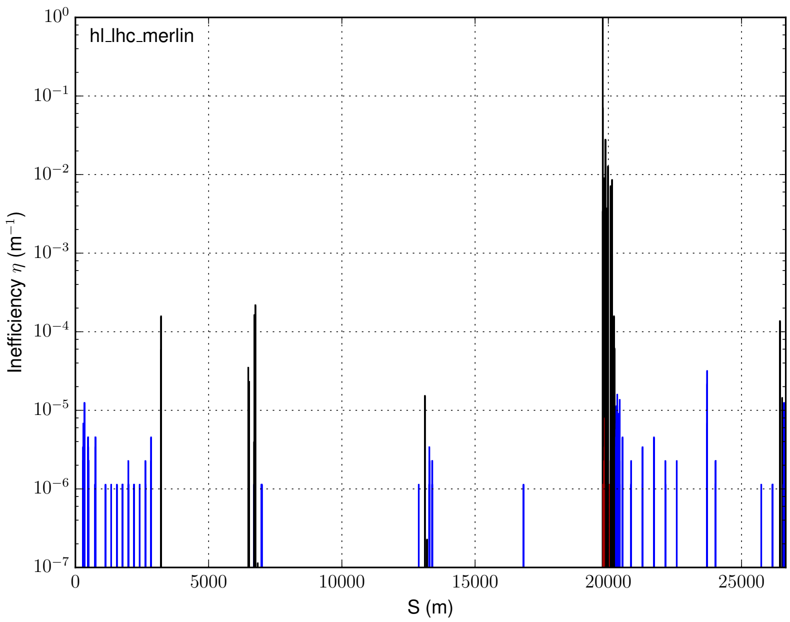

Figure 6: Loss map with MELRIN scattering.

Table 1 shows the absolute number of particles lost in the collimators, warm and cold regions, for each of the scattering models.

We can see in greater detail the difference between scattering models by zooming in to the main collimation region IR7. Figures 8 and 9 show the loss maps for the MERLIN and Sixtrack/K2 models. The MERLIN scattering shows increased losses in the warm region after the primary collimators and slightly reduced losses in the dispersion suppressor region.

Figures 10 through 12 show the intermediate models, with Sixtrack/K2 plus each of the new ionisation, elastic and single diffractive models. It can be seen that the greatest change comes from the new single diffractive modelling.

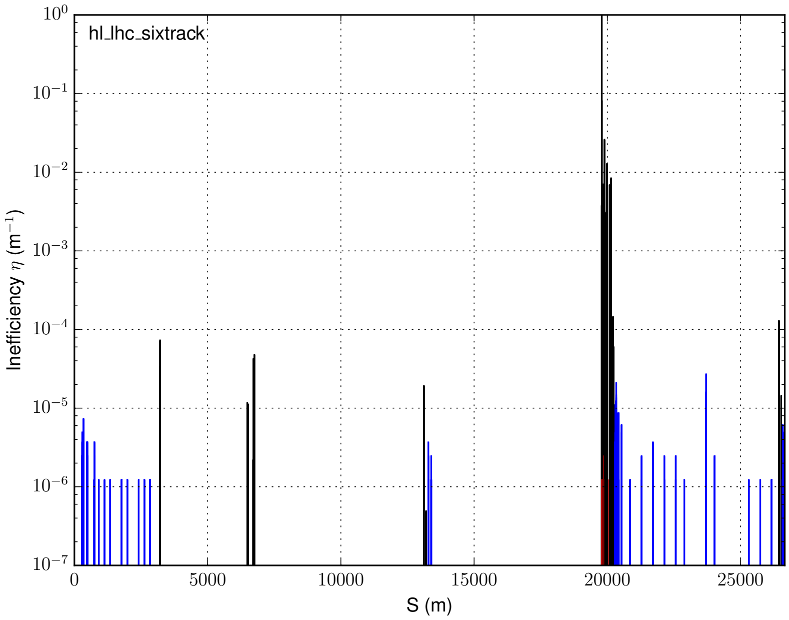

Figure 7: Loss map with sixtrack style scattering.

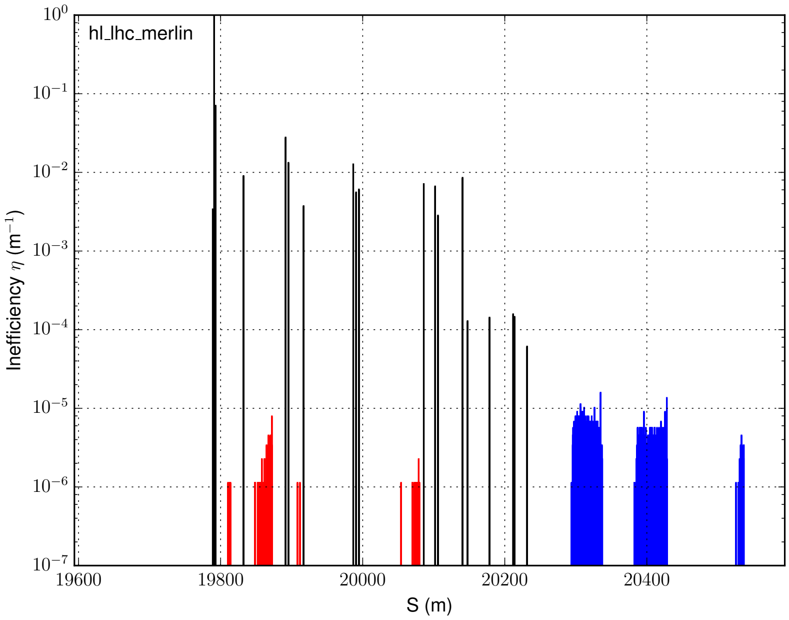

Figure 8: Loss map for IR7 with MERLIN scattering.

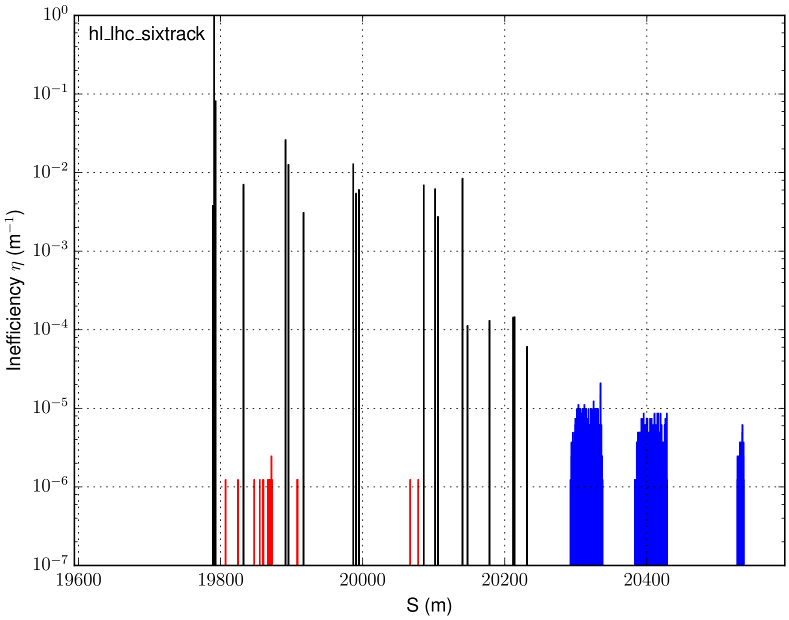

Figure 9: Loss map for IR7 with sixtrack style scattering.

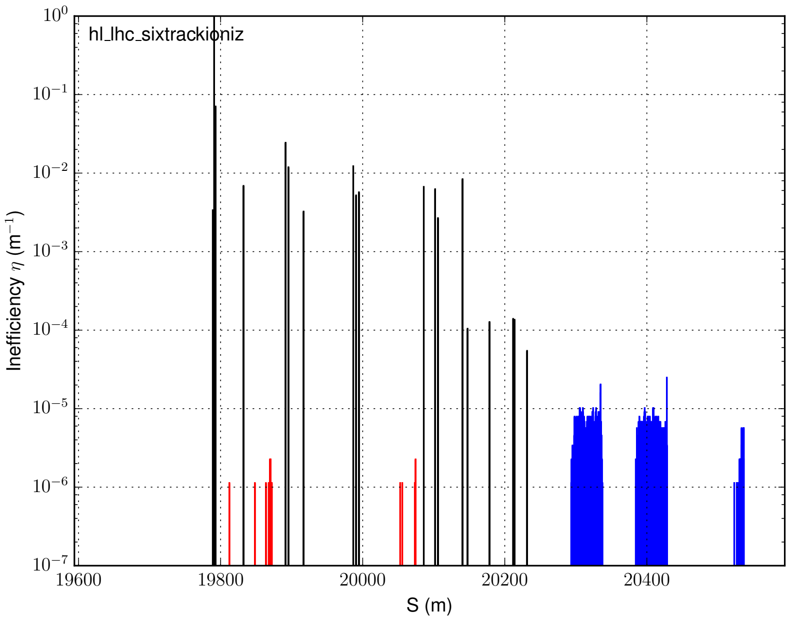

Figure 10: Loss map for IR7 with sixtrack style scattering plus new ionisation.

Table 1: Lost particles by mode and region for 100 million initial particles.

Mode	Total	Collimator	Warm	Cold
MERLIN	8847280	8844089	123	3068
Sixtrack/K2	8163140	8160076	23	3041
Sixtrack/K2 + Ionization	8844076	8839994	23	4059
Sixtrack/K2 + Elastic	8164313	8161303	14	2996
Sixtrack/K2 + SD	8164245	8162058	116	2071

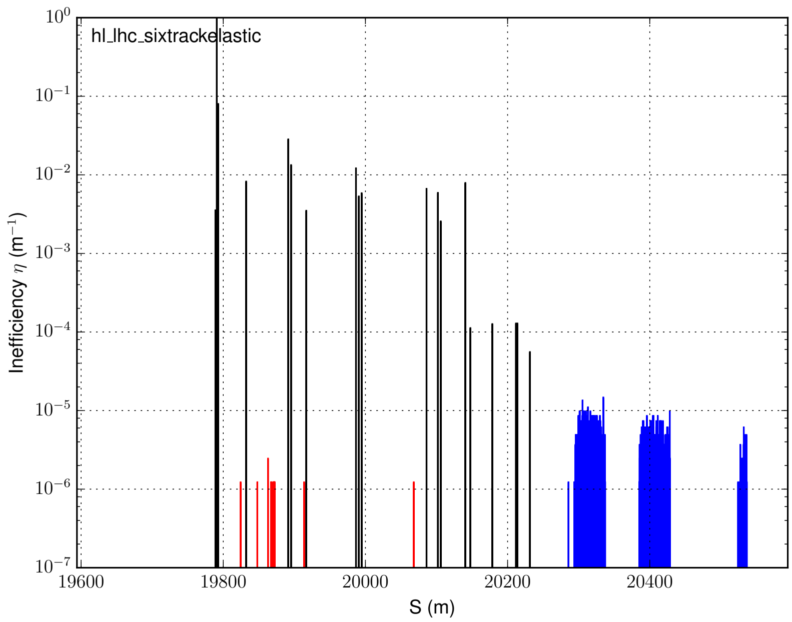

Figure 11: Loss map for IR7 with sixtrack style scattering plus new elastic.

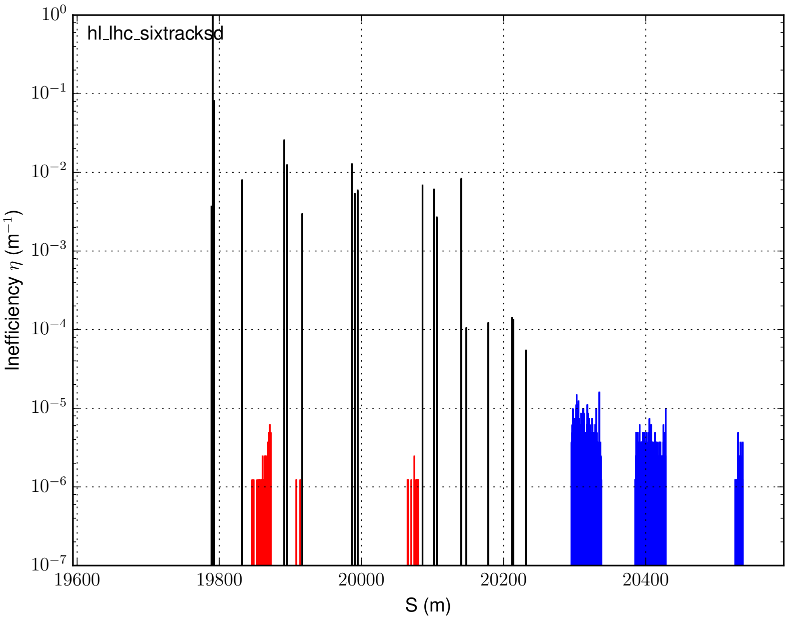

Figure 12: Loss map for IR7 with sixtrack style scattering plus new single diffractive.

5 Conclusion

The MERLIN accelerator simulation code contains advanced models of proton scattering in materials. It can be used for modelling the HL-LHC collimation system to make predictions of the losses around the ring. We demonstrate the effect on the loss maps due to the scattering model used. The more detailed scattering model is important for accurately predicting the loss rates.

MERLIN is open source and the code is released on the GitHub website: `https://github.com/MERLIN-Collaboration`.

References

[1] The HL-LHC Collaboration, HL-LHC Preliminary Design Report, CERN-ACC-2014-0300, 2014.

[2] An ILC Main Linac Simulation Package Based on MERLIN, F. Poirier et al., EPAC, 2006.

[3] MERLIN on GitHub, MERLIN Collaboration, `https://github.com/MERLIN-Collaboration`.

[4] Achromatic telescopic squeezing scheme and application to the LHC and its luminosity upgrade, S. Fartoukh, Phys. Rev. Special topics Accelerators and Beams, 2013, `http://dx.doi.org/10.1103/PhysRevSTAB.16.111002`.

[5] K2, a software package evaluating collimation systems in circular colliders (manual), T. Trenkler and J. B. Jeanneret, CERNSL/94105 (AP), 1994.

[6] CERN BE-ABP, SixTrack: Single Particle Tracking Code, `http://frs.home.cern.ch/frs/`.

[7] The practical Pomeron for high energy proton collimation, R. B. Appleby, R. J. Barlow, J. G. Molson, M. Serluca, A. Toader, Eur. Phys. J. C (2016) 76:520, `http://doi.org/10.1140/epjc/s10052-016-4363-7`

[8] Proton scattering and collimation for the LHC and LHC luminosity upgrade, James Molson, PhD Thesis, University of Manchester, 2014.

[9] A New Version of SixTrack with Collimation and Aperture Interface, G. Robert-Demolaize et al., PAC, 2005.