ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-THPF098

SPS-TO-LHC TRANSFER LINES LOSS MAP GENERATION USING PYCOLLIMATE

F. M. Velotti[1,2], W. Bartmann[1], C. Bracco[1], M. A. Fraser[1], B. Goddard[1], V. Kain[1], M. Meddahi[1],
[1]CERN, Geneva, [2]EPFL, Lausanne, Switzerland

Abstract

The Transfer Lines (TL) linking the Super Proton Synchrotron (SPS) to the Large Hadron Collider (LHC) are both equipped with a complete collimation system to protect the LHC against mis-steered beams. During the setting up of these collimators, their gaps are positioned to nominal values and the phase-space coverage of the whole system is checked using a manual validation procedure. In order to perform this setting-up more efficiently and more reliably, the simulated loss maps of the TLs will be used to validate the collimator positions and settings. In this paper, the simulation procedure for the generation of TL loss maps is described, and a detailed overview of the new scattering routine (pycollimate) is given. Finally, the results of simulations benchmark with another scattering routine are presented.

INTRODUCTION

The SPS is the last accelerator of the LHC injector chain. The SPS and LHC are directly connected through two ≈3 km long TLs which allow the injection of the required 450 GeV beam.

Both transfer lines (TI2 and TI8) are equipped with a complete passive protection system; it is composed of three graphite (R4550 in 1.2 m of active length) collimators per transverse plane (TCDIs). TI2 differs from TI8 because it is also equipped with a momentum collimator installed at the beginning of the line.

Such collimation system is designed to protect the LHC aperture and the LHC injection septa, in case of any kind of failures of the TLs active elements, as well as the SPS extraction systems.

The complete phase-space coverage is ensured by placing the three collimators, on each plane, at about $\pi/3 + n\pi$ phase-advance from each other. They are located at the end of the TLs and are single-stage protection devices. In case of failure in the upstream part of the TL, only one TCDI will be completely hit by the beam; therefore, every collimator has to guarantee an attenuation of the beam intensity to a safe level for accelerator components, i.e. 2×10^{12} [1].

Phase-space Coverage

The main aim of the TL collimators is to ensure adequate protection of the LHC cold apertures. From the LHC Design Report [2], the minimum available aperture in the arc is $7\,\sigma$, hence this represents the target protection for the TL collimation system.

In order to define the collimator jaws aperture needed to guarantee the above cited protection, all possible sources of error have to be taken into account. All the considered errors are listed in Table 1; summing these contributions linearly, considering a typical beam size of 0.5 mm, the total error is $\approx 1.4\,\sigma$ [1]. The maximum escaping amplitude in a "three-phase" collimation system is given by pure geometrical considerations, i.e. $A_{max} = A_{jaw}/\cos(\pi/6)$; where A_{jaw} is the required jaw position, including errors. For the LHC $A_{max} = 7\,\sigma$, so the collimator half-gap has to be $A_{jaw} = 4.5\,\sigma$.

As an example, Fig. 1 shows the horizontal Poincaré portrait at the septum at the end of TI2 (MSI2). Here, all three collimators are sketched at their ideal position, i.e. $4.5\,\sigma$, together with the surviving particles at the MSI2 for two different amplitudes, $4.6/4.9\,\sigma$.

Table 1: Errors for the TL Collimator Jaws [1]

Error type	Unit	Value
Inter-jaw parallelism	μm	50
Jaw axis wrt tank	μm	100
Tank axis wrt beam size	μm	180
Surface flatness	μm	50
Knowledge of bema position	μm	44
Beam size errors	σ	0.5

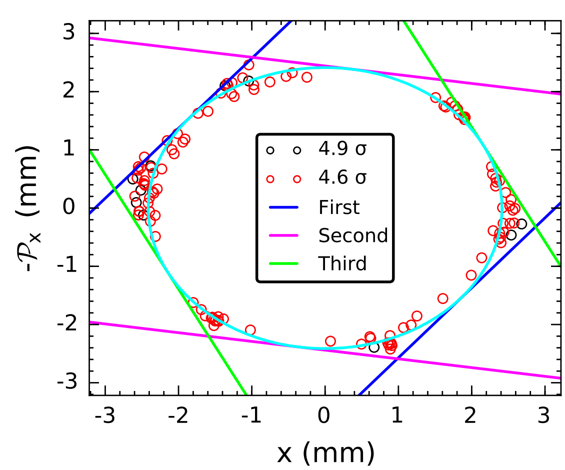

Figure 1: Horizontal phase-space at the LHC injection septum (MSI) downstream TI2.

Validation Methodology

In order to ensure the required protection, after setting up the TL collimators at their nominal apertures and centring them with respect to the measured beam trajectory,

ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-THPF098

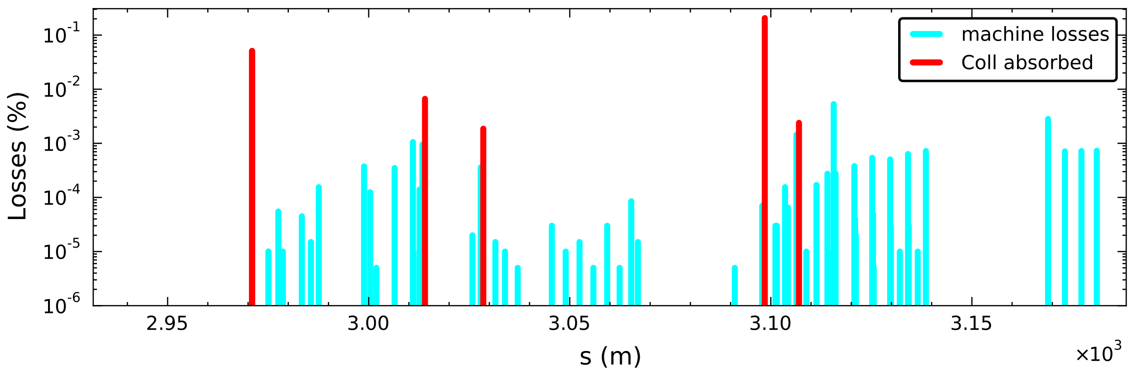

Figure 2: Simulation of TI2 loss map for phase zero oscillation with 5 σ amplitude. Losses normalised to number of particle tracked - 2×10^5.

an additional procedure is required to check the effective phase-space coverage. The validation process comprises exciting trajectory oscillations at all phases, sampled at 30° phase-advance, and amplitudes between 4 and 5 σ. After an accurate calibration of the Beam Loss Monitor (BLM), it is possible to reconstruct the maximum amplitude of the beam escaping the collimation system, at each phase.

This is a very long and tedious procedure, prone to errors, since it relies on the assumption that the beam is transversely perfectly Gaussian. An improved method comprises simulate loss patterns for every required oscillation and compare them with the measurements. To do so, a new scattering routine has been written in Python (`pycollimate`) and successfully interfaced with PTC (Polymorphic Tracking Code [3]) trough MADX.

In this paper, the characteristics and possibilities of `pycollimate` will be explored. The way to produce TL loss maps will be described, together with a first benchmark with *SixTrack extended for collimation* [4].

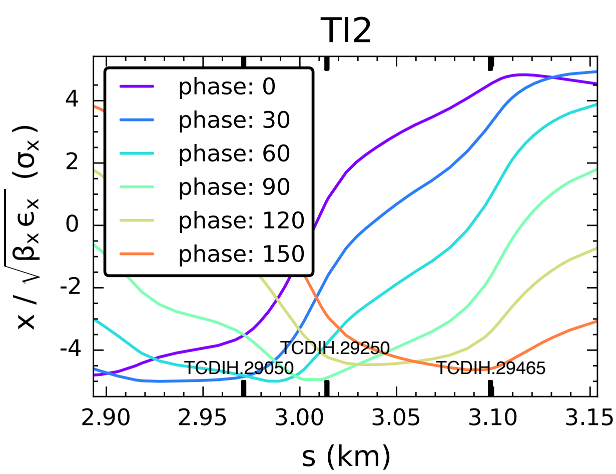

Figure 3: Four horizontal oscillations in TI2 with 5 sigma amplitude used for the TCDI setup validation. The phase 0 trajectory corresponds to the loss map in Fig. 2. The plot ends at the exit of the LHC injection kicker (MKI).

SIMULATION TOOLS

In order to produce realistic loss maps originated by the oscillations used for the TCDI validation procedure, two type of simulation codes are necessary: one tracking particles inside accelerator active elements (e.g. MADX, PTC, SixTrack, etc.) and one tracking particles inside matter (i.e. scattering routine). Different simulation tools were investigated (*SixTrack extended for collimation* or ORBIT [6]), which already provides the combination of the required tracking codes, but none of them could be directly used to produce a realistic loss map of the transfer lines (not suitable for non-circular machines, or completely different target energy). This led to the development of a new tool to simulate the validation process; that is `pycollimate` (scattering routine) interfaced with MAD-X and MADX-PTC (for the active elements tracking).

Pycollimate

The interaction processes of protons with matter, relevant for collimation studies, have been already investigated in literature [5, 7]. The scattering routine implemented in *SixTrack extended for collimation* (based on K2) is a Monte Carlo routine for tracking particles inside matter. The relevant processes considered, can be divided in four groups: nuclear interaction with finite cross section, Coulomb scattering, ionisation losses and hard electro-magnetic processes. The latter are actually marginal below proton energies of 10 TeV, hence they can be neglected for the energy range of interest, i.e. 450 GeV to 7 TeV.

Starting from the same assumptions, the new scattering routine mainly uses the same physical model implemented in *SixTrack extended for collimation* (also taking into account updated cross-sections [8] thanks to new available experimental data), although different features are also in-

ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-THPF098

cluded. For instance, the possibility to apply a very wide set of imperfections to each collimator jaw independently is implemented. It can be interfaced with MADX-PTC and perform thick lens tracking; particles escaping the collimator jaw before its end are tracked using the exact Hamiltonian in the remaining drift space.

The scattering routine lives as a function in the Python package `pycollimate`, which also includes all the tools to communicate with MADX, as well as classes and functions to characterise the scenario under analysis. This lets the user customise the simulation environment very easily and offers also the possibility to extend it without needing to modify the library source code directly; this is only one of the many advantages carried by a reusable generic library. Also, the complete independence of the scattering routine makes it usable with virtually any other accelerator tracking codes. The flexibility of the Python language allows a user of the `pycollimate` library to provide his/her own implementation of the scattering process and have it automatically invoked by the library scattering routine itself.

Loss Map Generation

In Fig. 2 an example of TL loss map is shown; this corresponds to the horizontal oscillation at "phase 0" (maximum of the oscillation at $\mu_x = \mu_x^{MSI} + n2\pi$, where μ_x^{MSI} is the horizontal phase-advance at the MSI) with 5σ amplitude. To obtain such a loss pattern, a bivariate distribution is tracked from the SPS extraction to the MSI entrance, on the trajectory shown in Fig. 3. The tracking is performed having the collimator half-gaps set at $5\,\sigma$.

The two different histograms (colour coded) in the loss maps (Fig. 2) refer to two different kind of losses: the red one refers to protons that have undergone to an inelastic scattering inside the collimator jaws, the cyan one refers to protons lost inside the machine elements that are not collimators.

Code Benchmark

To benchmark the scattering routine of `pycollimate` with SixTrack, a sample initial distribution of 6400 particles was tracked inside a 2 m graphite block. The distribution was generated as Gaussian in all phase-space transverse planes and at all particles had the same initial momentum, i.e. 450 GeV. The results are shown in Fig. 4. Moreover, Fig. 5 shows the difference of the number of particles absorbed at the same horizontal coordinate inside the Graphite block, normalised to the initial number of particles.

CONCLUSION

A new scattering routine, written in Python, has been developed and successfully interfaced with MADX-PTC to produce loss maps for the single-pass collimator setup validation. Such simulations will be used in the ongoing LHC Run 2 commissioning to improve and speed up the validation process for the TI2 and TI8 lines.

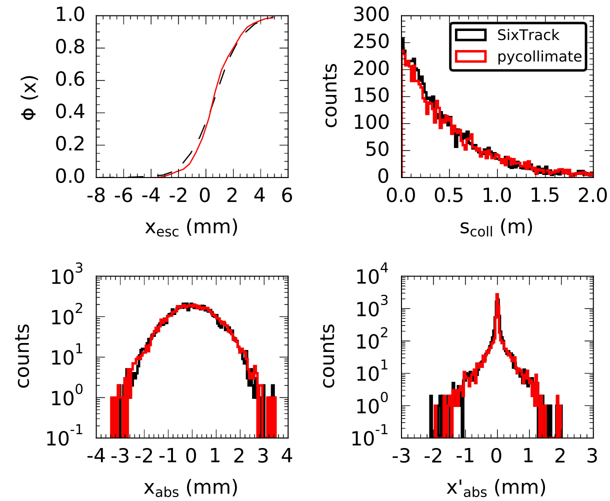

Figure 4: Comparison of *SixTrack extended for collimation* and `pycollimate` Monte Carlo scattering routines. The top left plot shows the cumulative density functions of the escaping proton amplitudes. The top right one shows the histograms of the longitudinal position of the absorbed particles inside the graphite block. The two bottom plots show the comparison of the horizontal transverse phase-space coordinates of the protons undergone an inelastic scattering.

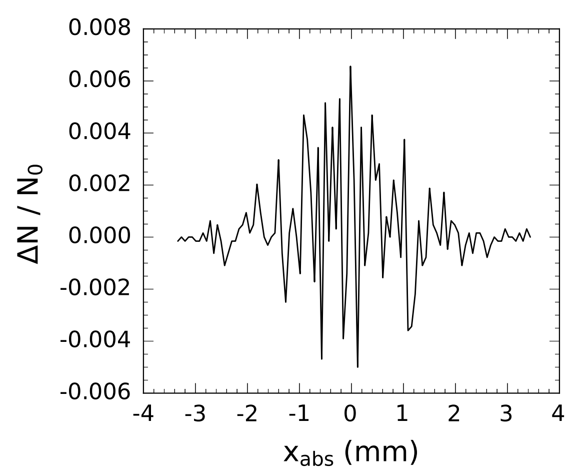

Figure 5: Difference of the histograms in the bottom left plot of Fig. 4 when using the same binning.

A first code benchmark has been done using SixTrack. The agreement shown is below the 10 % level, which is considered sufficient for the purpose of the tool.

Further benchmarks with different simulation tools are foreseen, and detailed comparisons with measurements will be performed before its release.

ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-THPF098

REFERENCES

[1] V. Kain et al., "The New Transfer Line Collimation System for the LHC High Luminosity Era," IPAC2014, June 2014, MOPRI096, p. 839.

[2] O. Brüning et al., "LHC Design Report", CERN, Geneva, 2004, http://cds.cern.ch/record/782076

[3] E. Forest, F. Schmidt and E. McIntosh, "Introduction to the Polymorphic Tracking Code", CERN-SL-2002-044-AP, KEK report 2002-3, July 2002.

[4] "Tracking Code for Collimation Studies extended version of SixTrack for collimation", http://lhc-collimation-project.web.cern.ch/, CERN, 2012.

[5] N. Catalán Lasheras, "Transverse and Longitudinal Beam Collimation in a High-Energy Proton Collider (LHC)", PhD Thesis, Universidad de Zaragoza, CERN, Geneva.

[6] J.A. Holmes et al., "Objective Ring Beam Injection and Tracking Code", http://web.ornl.gov/ jzh/JHolmes/ORBIT.html

[7] G. Robert-Demolaize, "Design and Performance Optimization of the LHC Collimation System", PhD Thesis, UNIVERSITE JOSEPH FOURIER, CERN, Geneva, 2006.

[8] C. Tabasco et al., "An improved scattering routine for collimation tracking studies at LHC", Master thesis, Università di Roma Sapienza, CERN, Geneva.