ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUSH201

ONLINE LUMINOSITY CONTROL AND STEERING AT THE LHC

M. Hostettler, R. Alemany, A. Calia, F. Follin, K. Fuchsberger, G. H. Hemelsoet, M. Gabriel, A. Gorzawski, M. Hruska, D. Jacquet, G. Papotti (CERN, Geneva, Switzerland)

Abstract

This contribution reviews the novel LHC luminosity control software stack. All luminosity-related manipulations and scans in the LHC interaction points are managed by the LHC luminosity server, which enforces concurrency correctness and transactionality. Operational features include luminosity optimization scans to find the head-on position, luminosity levelling, and the execution of arbitrary scan patterns defined by the LHC experiments in a domain specific language. The LHC luminosity server also provides full built-in simulation capabilities for testing and development without affecting the real hardware. The performance of the software in 2016 and 2017 LHC operation is discussed and plans for further upgrades are presented.

INTRODUCTION

The luminosity at an interaction point of a circular collider with Gaussian beams is given by [1]

$$\mathcal{L} = \frac{f_{\mathrm{rev}} N_1 N_2 n_{\mathrm{bunch}} \gamma}{4\pi\beta^* \varepsilon} \mathcal{G}\,\mathcal{S} \tag{1}$$

where f_{rev} is the revolution frequency, $N_{1,2}$ are the average bunch intensities in beam 1 and beam 2, respectively, n_{bunch} is the number of colliding bunches, β^* is the β-function at the interaction point, γ is the relativistic factor and ε is the normalized transverse emittance. $\mathcal{S}$ and $\mathcal{G}$ are luminosity reduction factors due to separation and crossing angle.

For beams crossing at an angle, the *geometric factor* $\mathcal{G}$ is

$$\mathcal{G} = \left(1 + \left(\frac{\sigma_z}{\sigma}\frac{\alpha}{2}\right)^2\right)^{-0.5} \tag{2}$$

for a crossing angle α, a transverse beam size of σ and a bunch length of σ_z. The plane (horizontal or vertical) in which the crossing angle is applied is commonly referred to as the “crossing plane”, while the other plane is called the “separation plane”.

The *separation factor* $\mathcal{S}$ is given by

$$\mathcal{S} = \exp\left(\frac{-d^2}{4\sigma^2}\right) \tag{3}$$

for a total separation of d between the two beams and a transverse beam size of σ (Fig. 1). If the beams are separated in the crossing plane, an additional correction for the combined effect of separation and crossing angle has to be made [1].

At the LHC, the luminosity control software primarily controls the luminosity by displacing one or both beams using a closed orbit bump, effectively introducing a separation. However, a functionality for changing the crossing angle in collisions has been implemented for the 2017 LHC proton physics run, and using β^* for luminosity control is foreseen for the future.

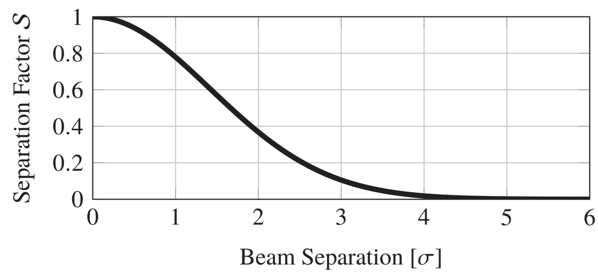


Figure 1: Separation factor as a function of the beam separation (in units of the beam size σ).

LHC Experiments

The LHC consists of two vacuum pipes where two particle beams travel in opposite directions. The two beams enter a common beam pipe in the four LHC interaction regions and collide at the Interaction Points (IP) inside the detectors of the experiments: IP1 (ATLAS experiment), IP2 (ALICE experiment), IP5 (CMS experiment) and IP8 (LHCb experiment). ATLAS and CMS are high-luminosity experiments designed to take data at the full LHC design luminosity of 10^{34} cm^{-2}s^{-1}. ALICE and LHCb take data at lower rates, and require to lower the luminosities in their IPs by partially separating the beams.

Use Cases

The operational use cases for the LHC luminosity control software stack include:

- **IP steering.** Let the operator directly modify the beam positions at any IP, e.g. for machine tests or at the request of the experiments.
- **Optimization.** Scan the beam separation while acquiring the luminosity signal to find the beam head-on position giving maximum luminosity [2].
- **Luminosity Calibration.** Perform separation scans synchronized with the experiments to calibrate the absolute luminosity measurement using the van-der-Meer method [3, 4].
- **Luminosity Levelling.** At a given IP, keep the luminosity constant around a target value, provided by the experiment or by the operator, by adjusting the beam separation [5].
- **Orchestration of setting changes.** Smoothly adjust machine parameters like the crossing angle and/or the β^* [6, 7] while the beams are in collision to optimally use margins while improving the luminosity (e.g. with decreasing beam intensities, the crossing angle can be decreased).

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUSH201

ARCHITECTURE

A new software stack for online luminosity control was developed at the LHC starting in 2015. As most other top-level control software for the LHC, it is entirely written in Java. Several fundamental choices were taken at the design stage to make the new software extensible and to ensure its reliability. These are explained in the following sections.

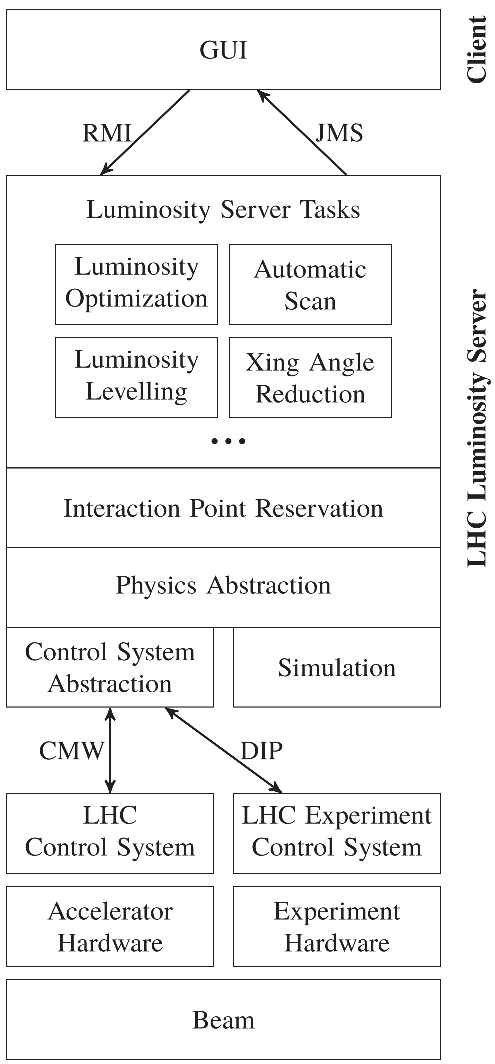


Figure 2: The new LHC luminosity software stack.

Client-Server Architecture

All the logic and processing code is separated from the GUI and encapsulated in a dedicated server (LHC Luminosity Server). The graphical client (LHC Luminosity Client) communicates remotely with this server over the network using JMS and RMI.

This client-server separation allows the server to keep full control of all the functionality and all publications made e.g. to the LHC experiments, making them independent of any open GUI. Therefore, multiple instances of the GUI can be open on the operational consoles and closed at any time, while still guaranteeing a coherent behaviour (e.g. the accidental closure of a GUI does not affect an ongoing scan). A defined state is guaranteed at all times.

Luminosity Control Tasks

All operations in the Luminosity Server are implemented as *Tasks*. A task does one thing once (e.g. optimizing a selected set of IPs). Once the task completed, the results can be reviewed for reference, but can not be restarted or reused. If the same operation needs to be done again, a new task must be created.

The framework of the Luminosity server allows tasks to be implemented in a transactional way, so the settings are reverted in case the task fails due to a problem or is aborted by the operator. This behaviour can be changed based on the needs of the actual task implementation. For example, an automated luminosity scan task is transactional, while a luminosity levelling or IP steering task is not.

Interaction Point Reservations

To ensure that only one task modifies the settings of a particular IP at a time, the Luminosity Server implements a reservation[1] system for the IPs. A task can request a set of IP reservations, and it is guaranteed that only one task will get a reservation for a particular IP at any given time.

If a task requests reservations for multiple IPs, they will be granted in IP order to prevent a deadlock situation (an example is shown in Fig. 3).

A task will not execute until the reservations are granted, and when a task finishes, fails, is cancelled or paused, the reservations it held are freed.

Figure 3: The "Task Manager" in the LHC Luminosity Client, with an overview of all active tasks along with their reservations. Granted reservations are shown in yellow; requested reservations are shown in green if they are available or in red if they are in use by another task.

Integration with the LHC Control System

The LHC Luminosity Server was designed to contain an abstraction layer for accessing the lower layers of the LHC control system, both for data acquisition and controlling the beams. The LHC Luminosity Client acquires the data to display exclusively through the LHC Luminosity Server it is connected to, and never subscribes to any data source directly. This allows replacing data sources or the acquisition logic at the server level transparently without modifying the client code, e.g. by simulation loops for testing purposes.

In the operational configuration, the LHC Luminosity Server acquires data e.g. from the power converters, the

[1] In information technology, this concept is commonly referred to as "Locking" or "synchronization", and reservations as "Mutexes"

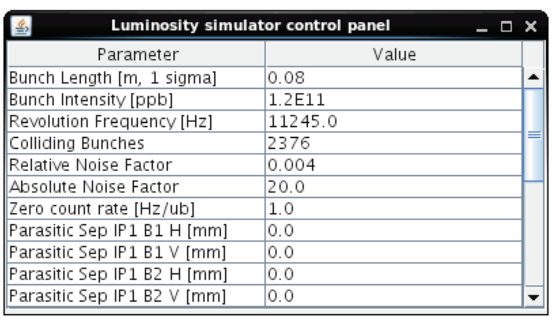
ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUSH201

LHC experiments, and the timing systems through the CERN Common Middleware (CMW) framework. Publications to the experiments are done over the Data Interchange Protocol (DIP), accessed through CMW. The beam displacements at the IPs are controlled through "Knobs" defined in the LHC Software Architecture (LSA) framework to generate 4-corrector closed orbit bumps around the IP concerned. In general the two beams are moved symmetrically in opposite directions unless requested otherwise to reduce the absolute displacement with respect to the common beam pipe.

Simulation Mode

By replacing the interface layer between the LHC control system and the Luminosity Server by a simulation loop, the server can run in simulation mode without accessing the LHC control system. In this mode, the server can even run outside the CERN network.

In simulation mode, the separations requested for each IP are recorded, and an appropriate luminosity signal is simulated according to Eqn. 1. The ramping time of the corrector magnet currents is also simulated. The beam conditions and various inputs to the simulation can be controlled by the user through a simple GUI. Any publication a task would make to the experiments is displayed instead.



Parameter	Value
Bunch Length [m, 1 sigma]	0.08
Bunch Intensity [ppb]	1.2E11
Revolution Frequency [Hz]	11245.0
Colliding Bunches	2376
Relative Noise Factor	0.004
Absolute Noise Factor	20.0
Zero count rate [Hz/ub]	1.0
Parasitic Sep IP1 B1 H [mm]	0.0
Parasitic Sep IP1 B1 V [mm]	0.0
Parasitic Sep IP1 B2 H [mm]	0.0
Parasitic Sep IP1 B2 V [mm]	0.0

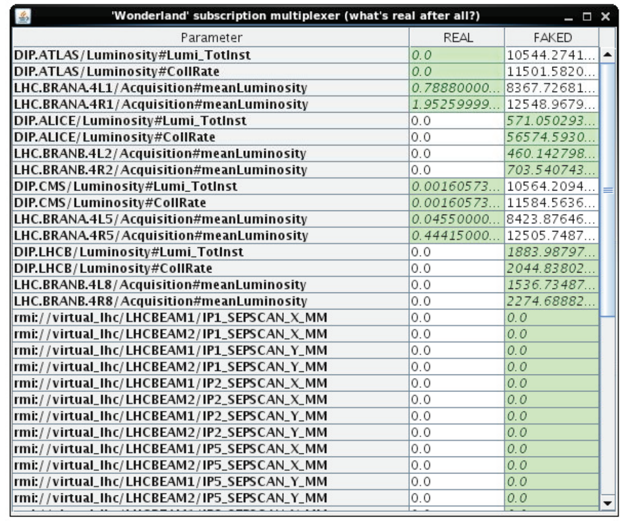
Figure 4: The control panel of the luminosity simulation loop of the LHC Luminosity Server. Any beam displacements that would be done are taken into account for the simulation.

The simulation mode allows to test any task and almost all aspects of the Luminosity Client and Server without accessing the LHC [8]. It is also used to dry-run various scan sequences, and to predict the resulting luminosity patterns.

Partial Simulation

It is also possible to start the server in partial simulation mode (dubbed "wonderland mode"). This allows to select either the simulated or the real implementation of every sub-system of the Luminosity Server, and provides a special subscription multiplexer to dynamically switch every input signal between the real acquisition and the simulated counterpart (Fig. 5).

This mode can be used to test certain interactions of the server with the control system with the least impact on operations. For example, it is possible to test the communication with the LHC experiments over DIP without actually moving the beams in the LHC.



Parameter	REAL	FAKED
DIP.ATLAS/Luminosity#Lumi_TotInst	0.0	10544.2741...
DIP.ATLAS/Luminosity#CollRate	0.0	11501.5820...
LHC.BRANA.4L1/Acquisition#meanLuminosity	0.78880000...	8367.72681...
LHC.BRANA.4R1/Acquisition#meanLuminosity	1.95259999...	12548.9679...
DIP.ALICE/Luminosity#Lumi_TotInst	0.0	571.050293...
DIP.ALICE/Luminosity#CollRate	0.0	56574.5930...
LHC.BRANB.4L2/Acquisition#meanLuminosity	0.0	460.142798...
LHC.BRANB.4R2/Acquisition#meanLuminosity	0.0	703.540743...
DIP.CMS/Luminosity#Lumi_TotInst	0.00160573...	10564.2094...
DIP.CMS/Luminosity#CollRate	0.00160573...	11584.5636...
LHC.BRANA.4L5/Acquisition#meanLuminosity	0.04550000...	8423.87646...
LHC.BRANA.4R5/Acquisition#meanLuminosity	0.44415000...	12505.7487...
DIP.LHCB/Luminosity#Lumi_TotInst	0.0	1883.98797...
DIP.LHCB/Luminosity#CollRate	0.0	2044.83802...
LHC.BRANB.4L8/Acquisition#meanLuminosity	0.0	1536.73487...
LHC.BRANB.4R8/Acquisition#meanLuminosity	0.0	2274.68882...
rmi://virtual_lhc/LHCBEAM1/IP1_SEPSCAN_X_MM	0.0	0.0
rmi://virtual_lhc/LHCBEAM2/IP1_SEPSCAN_X_MM	0.0	0.0
rmi://virtual_lhc/LHCBEAM1/IP1_SEPSCAN_Y_MM	0.0	0.0
rmi://virtual_lhc/LHCBEAM2/IP1_SEPSCAN_Y_MM	0.0	0.0
rmi://virtual_lhc/LHCBEAM1/IP2_SEPSCAN_X_MM	0.0	0.0
rmi://virtual_lhc/LHCBEAM2/IP2_SEPSCAN_X_MM	0.0	0.0
rmi://virtual_lhc/LHCBEAM1/IP2_SEPSCAN_Y_MM	0.0	0.0
rmi://virtual_lhc/LHCBEAM2/IP2_SEPSCAN_Y_MM	0.0	0.0
rmi://virtual_lhc/LHCBEAM1/IP5_SEPSCAN_X_MM	0.0	0.0
rmi://virtual_lhc/LHCBEAM2/IP5_SEPSCAN_X_MM	0.0	0.0
rmi://virtual_lhc/LHCBEAM1/IP5_SEPSCAN_Y_MM	0.0	0.0
rmi://virtual_lhc/LHCBEAM2/IP5_SEPSCAN_Y_MM	0.0	0.0

Figure 5: The subscription data source selector of the LHC Luminosity Server running in *wonderland mode*. At runtime, every subscription can be switched between the real acquisition and the internal simulation loop.

IMPLEMENTATION AND OPERATIONAL EXPERIENCE

Following the development in 2015, the new luminosity control software stack was commissioned in 2016 for LHC operation. Since then, it was used for luminosity setup and optimization in regular operation, as well as for the special scan sessions to calibrate the absolute luminosity measurements. As of 2017, the functionalities for luminosity levelling by separation and for reducing the crossing angle in collisions were introduced and used operationally.

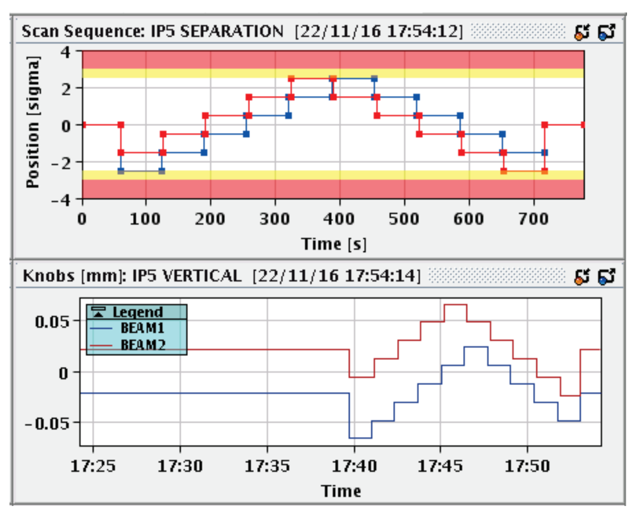


Figure 6: The CMS length-scale calibration scan run during the 2016 luminosity calibration session. The top panel shows the sequence as read from the file. The bottom panel shows the beam displacements sent to the control system while running the scan (the zero position is arbitrary).

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUSH201

Luminosity Calibration Scans

One of the first novel use cases for the new luminosity control stack was to allow running arbitrary scan sequences provided by the LHC experiments.

The *Automatic Scan* task reads sequences specified in a Domain Specific Language (DSL) from external text files [9]. While running the scan, it publishes the scan progress and actual beam separation to the experiments via DIP.

This task greatly simplified the luminosity calibration scan sessions in 2016 and 2017 compared to the past. In particular, the experiment specific scan sequences for length-scale calibration (see Fig. 6) had to be followed manually by the operators in previous years. The new software can run them in fully automated way, which saves ~20% of beam time per calibration session [10], and lowers the risk of operational mistakes.

Luminosity Optimization

At the start of each LHC fill, and regularly while in collisions, the positions of the beams at the IPs need to be optimized to ensure the beams are colliding head-on, so to deliver the maximum luminosity to the experiments.

The *Optimization* task achieves this by scanning the beam separation at an IP to maximize the luminosity signal. The two planes are optimized independently. For each plane, the beam separation is scanned while acquiring any luminosity signal selected by the operator until a peak (increase in luminosity followed by a decrease, considering statistical errors) is found. Thereafter, a parabola is fitted to the top 3 points acquired during the scan to refine the peak position (Fig. 7). Multiple IPs can be scanned in parallel.

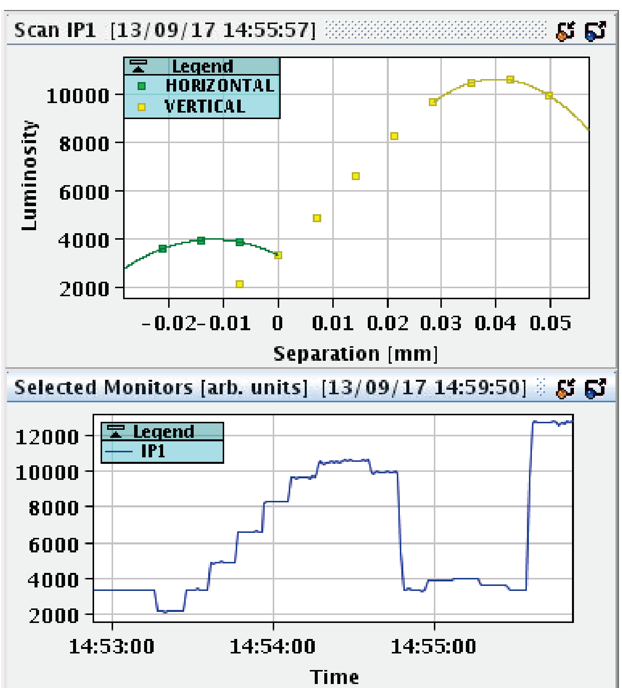


Figure 7: An example luminosity optimization in IP1. The top panel shows the acquired data points and fits. The bottom panel displays the raw luminosity signal; note the increase after applying the corrections.

Sanity checks are applied automatically both on the selected luminosity signal before a scan is launched, and on the results after a scan finishes. If potential problems are found for the selected luminosity signals (e.g. not publishing or publishing zeroes), a warning is issued before the scan is launched. In case a problem is found with the result (e.g. a bad fit, non-reproducible luminosity at the initial point or an unusually large correction), a warning is issued along with the scan result.

The results for all scanned IPs are shown to the operator via the LHC Luminosity Client along with the acquired luminosity points and the used fits; the operator is given the choice of which results to apply. After applying, the optimization can be undone (and the choice of which results to apply reconsidered) within a short time in case the luminosity did not increase.

The automatic optimization task has been used throughout 2016 and 2017 LHC operation. Compared to the previous years, the improved diagnostics and sanity checks allowed to quickly track down problems caused e.g. by missing or delayed luminosity data publications from the experiments.

Separation Levelling

The ALICE (IP2) and LHCb (IP8) experiments at the LHC are not designed to record data at the maximum luminosity the LHC can deliver. To lower the luminosity according to their needs, the beams are partially separated. As the beam intensity decreases over the course of a fill, the separation is decreased in order to keep the luminosity constant.

The *Separation Levelling* task implements a simple feedback loop on the luminosity signal published by the levelled experiment. The experiment also provides the target luminosity and acceptable tolerance. In a loop, the task acquires the luminosity signal, checks if it is outside of the target band, and adjusts the beam separation if necessary (Fig. 8).

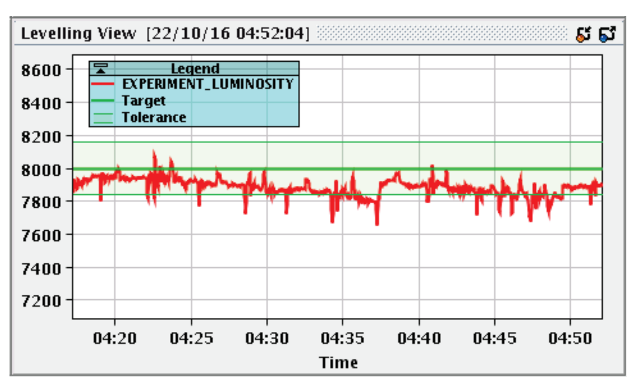


Figure 8: An example separation levelling status. At ~04:37 and at ~04:50, the beam separation was adjusted to bring the luminosity back into the target band.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUSH201

Crossing Angle Reduction

As the two beams share a common vacuum chamber around the interactions regions and the bunch distance is shorter than this common chamber, they have to collide at an angle to avoid parasitic collisions. Still, the beams are affected by each other's electromagnetic field. The crossing angle needs to be large enough to keep these beam-beam long-range effects to an acceptable level. However, the crossing angle decreases the luminosity due to the geometric factor (Eqn. 2). As the beam-beam effect decreases with the decreasing beam intensity over the course of the fill, the crossing angle can be reduced to increase the luminosity.

This is implemented in the *Angle Steering* task of the luminosity server. As changing the crossing angle (unlike changing the local separation at an IP) implies a significant orbit change, both the orbit feedback and the collimation system need to follow this change. The task orchestrates the crossing angle change by generating the settings, arming the systems involved, and finally sending a message to the common timing system to start the transition on all systems simultaneously.

This novel technique has been commissioned at the LHC in 2017, and was used during physics production ever since. Over 8 hours in collisions, the crossing angle was reduced from 150 μrad to 120 μrad in steps of 10 μrad. This increases the integrated luminosity for ATLAS and CMS by $3-5$ % depending on the fill length (Figs. 9, 10).

CONCLUSIONS

The new software stack for online LHC luminosity control was successfully commissioned and used in operation as of 2016. Compared to previous approaches, it improved the stability and reliability, as well as adding new functionalities. The whole framework is testable outside of the LHC control system using the built-in simulation mode. Mixing of acquired and simulated signals is also possible.

The typical use cases in LHC operation include automated luminosity optimizations scans, luminosity levelling by separation, and (as of 2017) "anti-levelling" by dynamically reducing the crossing angle in collisions, improving the integrated luminosity by up to ~5 % per LHC fill.

In the future, the functionality will be further extended to allow luminosity levelling by β^*, i.e. by squeezing or de-squeezing the beams at the interaction points. Also, the functionality for reducing the crossing angle will be further extended to reduce the crossing angle more continuously in smaller steps over the course of a fill.

REFERENCES

[1] W. Herr, "Particle Colliders and Concept of Luminosity", at CERN Accelerator School, Granada, Spain, 2012.

[2] R. Alemany-Fernandez, M. Lamont, S. White, "Separation Scan Application Requirements", EDMS specification, https://edms.cern.ch/document/894460/1

[3] S. Van Der Meer, ISR-PO/68-31, KEK68-64.

[4] S. White, "Determination of the Absolute Luminosity at the LHC", PhD thesis, University of Paris, France, 2010.

[5] R. Alemany-Fernandez *et al.*, "The LHCb Online Luminosity Control and Monitoring", Proceedings of IPAC2013, paper TUPFI010, Shanghai, China, 2013.

[6] A. Gorzawski, "Luminosity control and beam orbit stability with beta star leveling at LHC and HL-LHC", PhD thesis, Ecole Polytechnique Lausanne, Switzerland, 2016.

[7] J. Wenninger *et al.*, "Implementation of Luminosity Leveling by Betatron Function Adjustment at the LHC Interaction Points", Proceedings of IPAC2014, paper TUPRO022, Dresden, Germany, 2014.

[8] A. Calia *et al.*, "Testing the Untestable: A Realistic Vision of Fearlessly Testing (Almost) Every Single Accelerator Component Without Beam and Continuous Deployment Thereof", Proceedings of IBIC2016, paper TUPG30, Barcelona, Spain, 2016.

[9] R. Alemany-Fernandez *et al.*, "Automatic scan application around the LHC interaction points", EDMS specification, https://edms.cern.ch/document/1367878/7

[10] R. Alemany-Fernandez *et al.*, "Report about new lumi calibration program", Talk at the LPC Meeting of 23 October 2015.

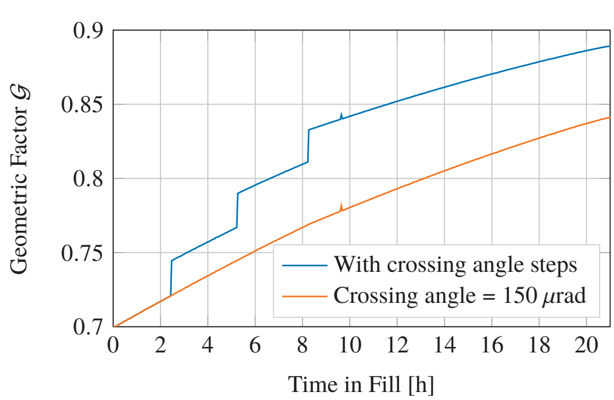


Figure 9: Evolution of the geometric factor $\mathcal{G}$ with crossing angle anti-levelling (LHC fill 5849). At each step, the crossing angle was reduced by 10 μrad.

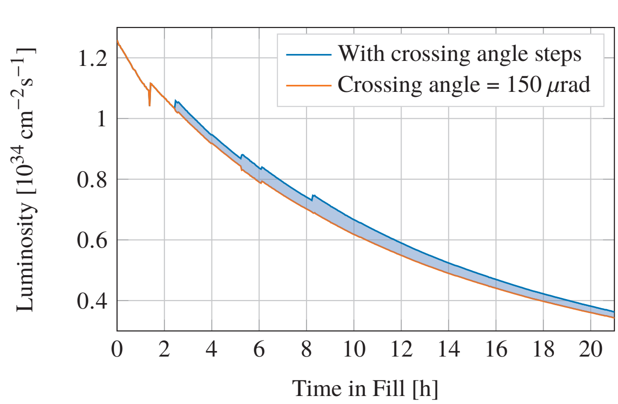


Figure 10: Luminosity evolution with crossing angle anti-levelling (LHC fill 5849). The gain in integrated luminosity (shaded area) is ~4 %.