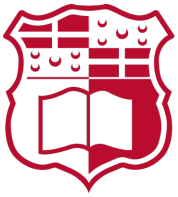
Automation of the LHC Collimator Beam-Based Alignment Procedure for Nominal Operation

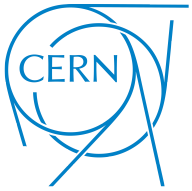
by

Gabriella Azzopardi

Supervisors: Dr. Gianluca Valentino (UM)
Dr. M. Belen Salvachua Ferrando (CERN)
Prof. Adrian Muscat (UM)



L-Università ta' Malta



Department of Communications and Computer Engineering

Faculty of ICT

University of Malta

September 2019

A dissertation submitted to the University of Malta for the degree of Doctor of Philosophy

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Student Name

Signature

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Declaration

Plagiarism is defined as "the unacknowledged use, as one's own, of work of another person, whether or not such work has been published, and as may be further elaborated in Faculty or University guidelines" (University Assessment Regulations, 2009, Regulation 39 (b)(i), University of Malta).

I, the undersigned, declare that the dissertation submitted is my work, except where acknowledged and referenced.

I understand that the penalties for committing a breach of the regulations include loss of marks; cancellation of examination results; enforced suspension of studies; or expulsion from the degree programme.

Work submitted without this signed declaration will not be corrected, and will be given zero marks.

_____ _____

Student Name Signature

_____

Title of work submitted

_____

Date

If you can't explain it simply, you don't understand it well enough.

- *Albert Einstein*

Abstract

The CERN Large Hadron Collider (LHC) is the largest particle accelerator in the world, built to accelerate and collide two counter-rotating beams. The LHC is susceptible to unavoidable beam losses, therefore a complex collimation system, made up of around 100 collimators, is installed in the LHC to protect its superconducting magnets and sensitive equipment.

The collimators are positioned around the beam following a multi-stage hierarchy. These settings are calculated following a beam-based alignment (BBA) technique, to determine the local beam position and beam size at each collimator. This procedure is currently semi-automated such that a collimation expert must continuously analyse the signal from the Beam Loss Monitoring (BLM) device positioned downstream of the collimator. Additionally, angular alignment are carried out to determine the most optimal angle for enhanced performance.

The human element, in both the standard and angular BBA, is a major bottleneck in speeding up the alignment. This limits the frequency at which alignments can be performed to the bare minimum, therefore this dissertation seeks to improve the process by fully-automating the BBA.

This work proposes to automate the human task of spike detection by using machine learning models. A data set was collated from previous alignment campaigns and fourteen manually engineered features were extracted. Six machine learning models were trained, analysed in-depth and thoroughly tested, obtaining a precision of over 95%.

To automate the threshold selection task, data from previous alignment campaigns was analysed to define an algorithm to execute in real-time, as the threshold needs to be updated dynamically, corresponding to the changes in the beam losses. The thresholds selected by the algorithm were consistent with the user selections whereby all automatically selected thresholds were suitable selections.

Finally, this work seeks to identify the losses generated by each collimator, such that any cross-talk across BLM devices is avoided. This involves building a cross-talk model to automate the parallel selection of collimators, and seeks to determine the actual beam loss signals generated by their corresponding collimators.

Manual, expert control of the alignment procedure was replaced by these dedicated algorithms, such that the software was re-designed to achieve fully-automatic collimator alignments. This software is developed in a real-time environment, such that the fully-automatic BBA is implemented on top of the semi-automatic BBA, thus allowing for both alignment tools to be available together and maintaining backward-compatibility with all previous functionality. This new software was used for collimator alignments in 2018, for both standard and angular alignments.

Automatically aligning the collimators decreased the alignment time by 70%, whilst maintaining the accuracy of the results. The work described in this dissertation was successfully adopted by CERN for LHC operation in 2018, and will continue to be used in the future as the default collimator alignment software for the LHC.

Sommarju

Il-"Large Hadron Collider" (LHC) li jinsab CERN, huwa l-ikbar aċċelleratur talpartiċelli fid-dinja, mibni biex jaċċellera u jħabbat partiċelli f'żewg faxex tad-dawl li jduru kontra xulxin. L-LHC huwa suxxettibbli għal ħsara li tista' tiġi kkawżata mit-telf tal-faxex tad-dawl li ma jistax jiġi evitat, u għalhekk sistema kumplessa ta' kollimazzjoni, li tikkonsisti minn madwar 100 kollimatur, hija nstallata fl-LHC biex tipproteġi l-kalamiti super-konduttivi u apparat ieħor sensittiv.

Il-kollimaturi huma pożizzjonati madwar il-faxex tad-dawl wara herarkija f'diversi stadji. Dawn l-issettjar huma kkalkulati wara proċedura t'allinjament (BBA), biex tiddetermina l-valuri tal-pożizzjoni u l-wisa' locali tal-faxex tad-dawl f'kull post tal-kollimatur. Din il-proċedura bħalissa hija semi-awtomatika b'tali mod li l-operatur għandu kontinwament janalizza r-rekordjar tal-"Beam Loss Monitoring (BLM)" ditettur, pożizzjonat 'l isfel mill-kollimatur. Barra minn hekk, l-allinjament angolari jitwettaq biex jiddetermina l-iktar angolu ottimali għal prestazzjoni aħjar.

L-element uman, kemm fil-proċedura normali kif ukoll l-angolari, huwa fattur kbir li jippreveni it-tħaffif tal-proċedura t'allinjament. Dan il-fatt jillimita lfrekwenza li biha dawn l-allinjamenti jistgħu jitwettqu għall-minimu, għalhekk din it-teżi tfittex li ssolvi dan billi tawtomatizza kompletament il-BBA.

Din ir-riċerka tipproponi li tawtomatizza x-xogħol meħtieġ biex tiskopri sinjali t'allinjament fl-irrekordjar tal-BLM, bl-użu ta' "machine learning". Sett ta' data nġabar minn kampanji t'allinjament preċedenti, li minnhom ġew estratti manwalment erbatax-il karatteristika. Sitt mudelli tal-"machine learning" ġew imħarrġa, analizzati fil-fond u ttestjati, u kisbu preċiżjoni aktar minn 95%.

Għall-awtomatizzar tas-selezzjoni tal-limitu, sett ta' data nġabar minn kampanji t'allinjament preċedenti u ġie analizzat biex jiġi definit algoritmu li jkun eżegwit f'"real time", peress li l-limitu jeħtieġ li jiġi aġġornat b'mod dinamiku, li jikkorrispondi għall-bidliet fit-telf tal-faxex tad-dawl. Il-limiti magħżula mill-algoritmu huma consistenti mal-limiti magħżula mill-operaturi, talli li l-limiti magħżula awtomatikament kienu kollha selezzjonijiet xierqa.

Fl-aħħarnett, dan ix-xogħol ifittex li jiddentifika t-telf iġġenerat minn kull kollimatur, b'mod li jiġi evitat kwalunkwe "cross-talk" madwar id-ditetturi BLM. Dan jinvolvi li jinbena mudell tal-"cross-talk" biex jawtomatizza s-selezzjoni ta' kollimaturi li jistgħu jibdew l-allinjament tagħhom fl-istess ħin, u jfittex li jikkwantifika ssinjali attwali tat-telf tal-faxex tad-dawl iġġenerati mill-kollimaturi korrispondenti tagħhom.

Il-kontroll manwalment u espert tal-proċedura tal-allinjament ġie sostitwit b'dawn l-algoritmi ddedikati, b'tali mod li s-"software" ġie ddisinjat mill-ġdid biex jinkiseb allinjament tal-kollimaturi kompletament awtomatiku. Dan is-"software" huwa żviluppat f'ambjent "real-time", b'tali mod li l-BBA kompletament awtomatiku huwa implimentat fuq il-BBA semi-awtomatiku, sabiex jippermetti li ż-żewġ għodod t'allinjament ikunu disponibbli flimkien u jżommu "backward compatibility" mal-funzjonalita' kollha preċedenti. Dan is-"software" il-ġdid intuża

għall-allinjamenti tal-kollimaturi fl-2018, kemm għall-proċedura normali kif ukoll l-angolari.

L-allinjament kompletament awtomatiku tal-kollimaturi naqqas il-ħin tal-allinjament b'70%, filwaqt li żamm il-preċiżjoni tar-riżultati. Ix-xogħol deskritt f'din it-teżi ġie adottat minn CERN għall-operazzjoni tal-LHC fl-2018, u se jibqa' jintuża fil-futur bħala s-"software" primarju għall-allinjament tal-kollimaturi fl-LHC.

Acknowledgements

This work would not have been possible without the guidance of my supervisors Gianluca Valentino, Belen Salvachua and Adrian Muscat. I am thankful to them for giving me the opportunity to join the collimation project and for placing their trust and confidence in my abilities. Over the past 3 years they have supported and guided me in every possible way.

A special thank you to Stefano Redaelli for welcoming me into the LHC collimation team at CERN and for regularly overseeing and participating in my work. I would also like to thank my present and past colleagues in the team for helping me settle into the team and for always discussing and answering any questions that I had, especially Nuria, Alex, Andrey, Daniele, James, Alessio, Roderik and others.

Thank you to Jörg Wenninger for welcoming me into the LHC Operations team and for providing any assistance required as the section leader. Thanks to all my colleagues in the operations team acting as LHC Engineers in Charge, for preparing the beams for collimator alignments, both during commissioning and beam tests over the years.

I would like to express my gratitude to my parents for their endless support, love and encouragement and I am thankful to my sister Kristina, for always understanding and advising me.

I would like to thank the many friends that I made at CERN who have definitely been a big factor of my PhD life, especially Nima, Dimitris, Pinelopi, Christina, Leander and Pablo. I would also like to thank my friends in Malta who have constantly supported each of my decisions, especially Petra and Carlo.

Finally, I am deeply thankful to Aleks for providing constant patience, support and understanding, for joining me on this journey and for always helping me whenever possible.

Contents

List of Figures

List of Tables

Nomenclature

Symbols

α_i	angular offset of collimator i
$\beta*$	amplitude of the β function in the experimental insertion point
Δx_i	beam offset at collimator i with respect to an ideal orbit trajectory, measured from BLM-based alignment
δ	angular step size
ε	beam emittance
a_{ni}	cross-talk factor quantifying the effect of a closed upstream collimator i on collimator n
b_n	cross-talk factor in protons/Gy of collimator n, calculated from the change in beam intensity
BLM_i	BLM device positioned downstream from collimator i
I	beam intensity
J_L	left collimator jaw position
J_R	right collimator jaw position
N_i	half gap opening in units of beam σ specific to a collimator family
P_i	impact rate for any collimator i, in protons / second
S_i'	signal recorded by $\mathrm{BLM_i}$ without cross-talk
S_i^{Thres}	BLM stop threshold
S_i	signal recorded by $\mathrm{BLM_i}$
t_i^s	time interval in seconds
$x_i^{L,m}$	left jaw setup position determined from BLM-based alignment

$x_i^{R,m}$	right jaw setup position determined from BLM-based alignment
σ_i	one standard deviation of the beam size at element i

Acronyms

ALICE	A Large Ion Collider Experiment
ANN	Artificial Neural Network
ATLAS	A Toroidal LHC ApparatuS
B1	LHC beam 1
B2	LHC beam 2
BBA	Beam-based Alignment
BCI	Brain Computer Interface
BLM	Beam Loss Monitoring
BPM	Beam Position Monitor
CCC	CERN Control Centre
CERN	European Organization for Nuclear Research
CFC	Carbon Fibre Composite
CMS	Compact Muon Solenoid
DOROS	Diode ORbit and OScillation
DS	Dispersion Suppressor
DT	Decision Tree
ECoG	Electrocorticography
EEG	Electroencephalography
EiC	Engineer in Charge
EMG	Electromyography
EWMRMS	Exponentially Weighted Moving RMS
FEC	Front-End Computer
FESA	Front-End Software Architecture

GB	Gradient Boost
GUI	Graphical User Interface
HMM	Hidden Markov Model
HSD	Honest Significant Difference
IC	Ionization Chamber
iEEG	Intracranial Electroencephalography
IP	Interaction Point
IR	Insertion Region
JAPC	Java API for Parameter Control
LD	Left Downstream Jaw Corner
LHCb	LHC beauty
LHC	Large Hadron Collider
LR	Logistic Regression
LSA	LHC Software Architecture
LU	Left Upstream Jaw Corner
LVDT	Linear Variable Differential Transformer
MD	Machine Development
PS	Proton Synchrotron
RD	Right Downstream Jaw Corner
RF	Random Forest
RMS	Root Mean Square
RS	Running Sum
RU	Right Upstream Jaw Corner
SFS	Sequential Forward Selection
SPS	Super Proton Synchrotron
SVM	Support Vector Machine

TCDQ	Target Collimator Dump Quadrupole
TCLA	Target Collimator Long Absorber
TCLI	Target Collimator Long Injection Protection
TCP	Target Collimator Primary
TCSG	Target Collimator Secondary Graphite
TCT	Target Collimator Tertiary
TDI	Target Dump Injection

List of Publications

The following publications are the result of the work of this dissertation.

Journals

1. G. Azzopardi, G. Valentino, A. Muscat, B. M. Salvachua Ferrando, "Automatic spike detection in beam loss signals for LHC collimator alignment", Nuclear Instruments and Methods in Physics Research Section A, 934:10-19, 2019.

2. G. Azzopardi, B. M. Salvachua Ferrando, G. Valentino, "Data-driven cross-talk modeling of beam losses in LHC collimators", Physical Review Accelerators and Beams, 22(8), 2019.

3. G. Azzopardi, B. M. Salvachua Ferrando, G. Valentino, S. Redaelli, A. Muscat, "Operational Results on the Fully-Automatic LHC Collimator Alignment", Physical Review Accelerators and Beams, 22(9), 2019.

Conferences

1. G. Azzopardi, G. Valentino, B. M. Salvachua Ferrando, S. Redaelli, A. Mereghetti, A. Muscat, "Automatic Angular Alignment of LHC Collimators". In Proceedings of ICALEPCS'17, Barcelona, Spain, pp. 928-933, 2017.

2. G. Azzopardi, B. M. Salvachua Ferrando, G. Valentino, S. Redaelli, A. Muscat, "Operational Results of LHC collimator alignment using machine learning". In Proceedings of IPAC'19, Melbourne, Australia, pp. 1208-1211, 2019 (contributed talk).

3. G. Azzopardi, G. Valentino, B. M. Salvachua Ferrando, S. Redaelli, A. Muscat, "Software Architecture for Automatic LHC Collimator Alignment using Machine Learning". Accepted for publication at ICALEPCS'19, New York, NY, USA, 2019 (contributed talk).

4. G. Azzopardi, G. Valentino, B. M. Salvachua Ferrando, S. Redaelli, A. Muscat, "Beam Loss Threshold Selection for Automatic LHC Collimator Alignment". Accepted for publication at ICALEPCS'19, New York, NY, USA, 2019.

1. Introduction

1.1 The Large Hadron Collider at CERN

CERN, the European Organization for Nuclear Research, uses the world's largest and most complex scientific instruments to study particle physics, bringing together physicists, engineers and computer scientists from around the world. CERN is situated on the Franco-Swiss border near Geneva, and was founded in 1954.

The Large Hadron Collider (LHC) at CERN is the largest particle accelerator in the world. It is used to accelerate and collide two counter-rotating beams, each having a nominal energy of 6.5 TeV, with a design luminosity of 10^{34} $\mathrm{cm}^{-2}\mathrm{s}^{-1}$ at a bunch collision rate of 40 MHz [1]. The LHC is situated in a tunnel approximately 100 m underground, and has a circumference of 27 km.

It is made up of eight arcs containing superconducting magnets and eight straight sections which are referred to as insertion regions (IRs). Four IRs house the four main experiments; ATLAS, ALICE, CMS and LHCb, which are located at the points where the beams are brought into collisions. These points are referred to as Interaction Points (IPs) and Figure 1.1 shows the four main experiments located at IP 1, 2, 5 and 8 [2].

The LHC is susceptible to beam losses from normal and abnormal conditions [4, 5], which may generate a transition from superconducting to normal conducting

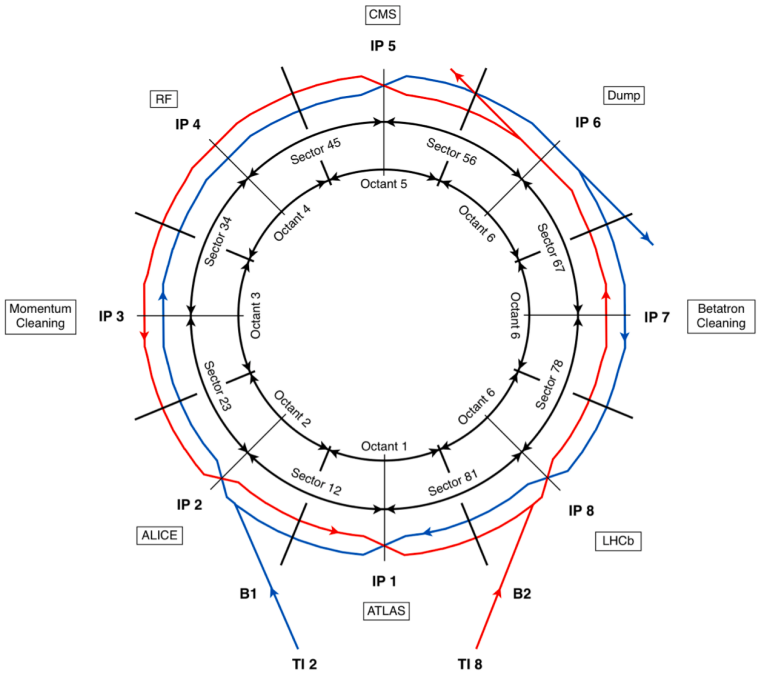


Figure 1.1: The LHC layout with beam 1 and beam 2 circulating in opposite directions and are brought into collisions at four interaction points: ATLAS (IP1), ALICE (IP2), CMS (IP5), LHCb (IP8), from [3].

state (quench) of the superconducting magnets. A robust collimation system is used to safely dispose of beam losses by concentrating them in the collimation regions. This protects the superconducting magnets and other sensitive equipment from any damage by effectively absorbing halo beam losses. To perform this, the collimation system follows a multi-stage hierarchy, that needs to be controlled and preserved by aligning the collimators towards the beam with a precision of less than 50 μm.

Particle beams are generally considered to follow a Gaussian distribution of particles in the transverse plane. One standard deviation of the Gaussian beam, 1 σ, consists of ∼68% of all particles. The beam core is usually defined as 0 - 3 σ (99.7% of all particles), while the region > 3 σ is known as the beam halo. The collimators are mainly concentrated in two dedicated cleaning insertions, IR3 and IR7 [4], and provide the cleaning of particles with large momentum and betatron offsets, respectively. The nominal intensity (3×10^{14} protons per beam) requires a collimation inefficiency of 2×10^{-5} m^{-1} [6], which is only possible by placing the collimators precisely around the beam.

Proton beams were first injected into the LHC on 10th September 2008. During a given LHC run (order of a few years), each year of operation ends with a winter shutdown, enabling equipment to be upgraded if needed. Therefore, due to the changes and period of inactivity, each year of operation begins with a commissioning phase to ensure the correct operation of all the machine sub-systems and fine-tuning accelerator devices to allow the LHC to achieve nominal operation [7]. The LHC and its injector chain are operated remotely from the CERN Control Centre (CCC).

1.2 LHC Machine Operation

The operation of the LHC follows a machine cycle which is made up of well-established stages, as shown in Figure 1.2. The *injection* (1) stage is when the LHC receives two 450 GeV beams from the Super Proton Synchrotron (SPS). The beams arrive in bunch trains of 288 bunches, whereby the number of bunches depends on the use of the beam (up to a nominal maximum of 2808 per beam). Once the filling procedure is complete, both beams are *ramped up* (2) until they reach top energy. Between 2015-2018 the LHC was operating at a maximum of 6.5 TeV for proton beams, and this stage is referred to as *flat top* (3). The procedure for shrinking the β^* i.e. the beam size at the experimental IPs, is initiated. This is required to increase the rate of collisions, and this process is referred to as the *squeeze* (4). The final step is then to bring the beams into collisions to collide in stable conditions, referred to as *stable beams* (5). Due to the high energy present in the LHC at all times, collimation is required throughout the machine cycle.

At any moment during the fill, the beams may be extracted or *dumped* (6). This could be a programmed dump due to operational requirements or a dump triggered by one of the machine protection systems to avoid damage of the LHC. At this point the machine is *ramped down* (7) bringing the magnets to their injection current, in preparation for the next fill.

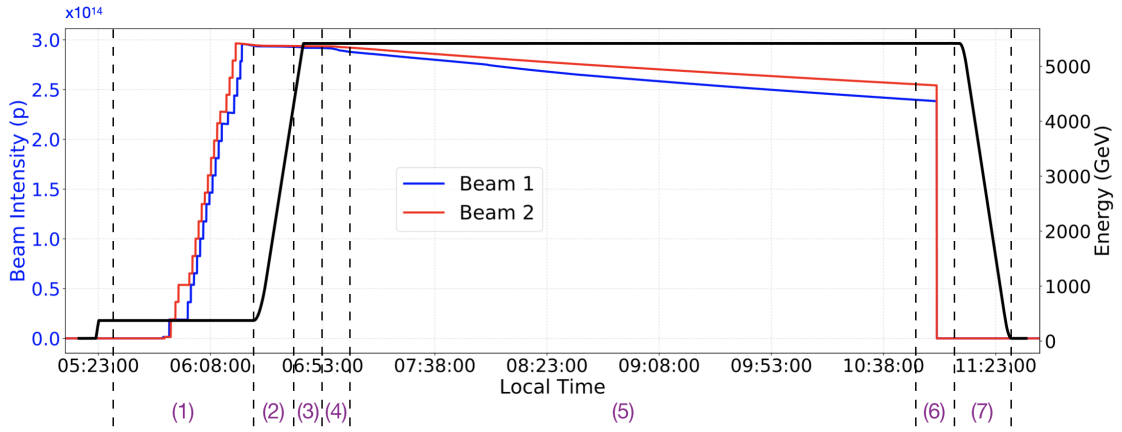


Figure 1.2: The seven stages of the LHC machine cycle [3], illustrated by beam energy and beam intensities: (1) injection, (2) ramp, (3) flat top, (4) squeeze, (5) stable beams, (6) beam dump, (7) ramp down.

1.3 Motivation for Automated Alignment

To achieve highly effective beam cleaning performance, the LHC collimators need to be aligned by precisely positioning them around the beam, in terms of offset and jaw gap. The major collimator alignment campaigns are performed at the start of each operational year and when the machine configuration is changed; optics, crossing angle, etc. Such alignment campaigns are required to set up the collimation hierarchy. Various standalone alignments are also frequently performed during machine development (MD) tests with beam. Each major alignment campaign is performed with the aim of preserving the hierarchy obtained until the next major alignment campaign, however in case of degradation in performance or if the beam orbit, the path taken by the beams, drifts over time, then the collimators would need to be realigned.

The current alignment software used to align collimators around the beam is semi-automated, and the high involvement of a human factor places the following restrictions:

1. Alignment campaigns require a number of collimation experts to be available during the whole period of the alignment.

2. The alignments take a considerable amount of hours and the required time is not constant, as it depends on human performance.

3. Decisions on the quality of alignment for each collimator are performed by experts, therefore the alignment precision depends on human performance.

Due to this, the semi-automation limits the results and the frequency and speed at which alignments can be performed. This therefore motivated the development of a fully-automated tool for performing alignments. Having the process fully-automated would allow anyone in the CCC to initiate the alignment procedure, and a collimation expert would only be required to analyse the results.

Moreover, fully-automating the alignment will aid in paving the way for other alignment techniques such as angular alignments. Collimators have always been aligned assuming no tilt between the collimator and the beam, as the current setting hierarchy has always had enough margins to safely absorb possible tilt angles. Tank misalignments or beam envelope angles at large-divergence locations could introduce a tilt, which would limit the collimation performance. Therefore in order to push the performance of the LHC, tighter collimator settings are required to improve halo cleaning. This can be achieved by applying the necessary tilts in the collimator jaws, which can only be determined after performing angular alignments.

At present, angular alignments must be done by manually setting the collimators at different angles to determine the most optimal angle. Exploring a suitable angular range is equivalent to 6 - 15 standard alignments, in time. Due to this, angular alignments are only performed when absolutely necessary. Having novel angular alignment procedures that could be run automatically at regular intervals (e.g.: on a yearly basis during commissioning), would be useful to identify and correct any tilts early on, to allow for tighter collimator settings.

This further motivates having a fully-automatic alignment tool which can be extended to also efficiently align collimators at different angles, as this would save

a significant amount of time and human resources. Considering the importance of delivering physics data to the LHC experiments, beam time for these activities is costly, therefore any time saved by fully automating the alignment is of vital importance.

1.4 Project Aim

The aim is to explore, design and implement an approach to transform the semi-automatic collimator alignment into a fully-automated one. Collimators need to be aligned frequently, therefore having the alignment fully-automated would allow alignments to be done by anyone at any time, as no human intervention will be required during the process.

The initial aim is to implement software able to automatically align all collimators by replacing the main decisions made by the user related to collimator selection, spike detection, and threshold selection. This dissertation seeks to analyse the data associated with each of these tasks to close the loop between the feedback received and the decision making process, thus fully-automating the alignment process. The software will apply the same collimator alignment procedure already in use, with the difference being that no human intervention would be required.

This fully-automated alignment can be taken a step further by exploring and introducing novel angular alignment procedures and automating them such that they can also be applied to any collimator upon request. This will allow finding the best alignment angle for each collimator individually, and will significantly decrease the angular alignment time required compared to the current manual procedure.

1.5 Objectives

The objectives are as follows:

1. Review other work done in analysing time series data, mostly related to;

- Extracting defining characteristics and to identify patterns in a data set.
- Spike detection techniques and in particular whether machine learning is a good candidate.

2. Build Dataset - Gather various sets of data from previous alignment campaigns. This is done by selecting instances during which collimators were moved closer to the beam.

3. Develop Spike Recognition System - A spike in the data implies a collimator is aligned, therefore spike recognition needs to be applied to ensure that a collimator is actually aligned with the beam. Machine learning models are trained and compared to automatically recognize such spikes.

4. Automate Threshold Selection - In order to guarantee that the beam is not totally scraped and that the same beam can be used to align all collimators, a beam loss threshold has to be selected. This threshold needs to be changed according to the beam loss conditions, therefore this must also be automated by analysing the data gathered.

5. Cross-talk Model Analysis - This must be studied to determine how collimators effect each other. This study can be based on a data-driven model or using simulated data, depending on the data available and the level of detail that will be studied.

6. Fully-Automate Alignment - The alignment of all collimators can be automated using the three results; automatic spike recognition model, automatic threshold selection algorithm, and cross-talk analysis.

7. Fully-Automate Angular Alignment - Novel angular alignment procedures, to align collimators at different angles, are required and these can also be fully automated using the three dedicated studies.

1.6 Thesis Organisation

This dissertation is structured as follows. An introduction of the Large Hadron Collider was provided in Chapter 1. An overview was given on LHC operation and the collimation system and how automating the alignment of collimators is vital to enhance operational efficiency. This was followed by the aim of this dissertation and the approach to achieve this goal, including how three forms of data analysis are key to replacing the user tasks with automatic algorithms.

A detailed background of the collimation system is expanded upon in Chapter 2, which explains the software architecture and the beam instrumentation available for collimator alignments. This chapter then gives an overview on supervised machine learning models and simulation software used at CERN.

Chapter 3 discusses work similar to the objectives of this dissertation, namely; collimator alignments in other colliders, pattern recognition in time-series data, threshold selection, automating systems that make use of time series data as feedback and cross-talk analysis.

A detailed formulation of the collimator alignment problem and its specifications and constraints are presented in Chapter 4. This chapters lists the different elements that make up the alignment task, followed by an overview of the proposed solution.

The data analysis and implementation details begin in Chapter 5, with a description of how machine learning was applied for automatic spike recognition. This chapter discusses the training and comparison of six machine learning models, which were validated during LHC operation.

Chapter 6 discusses the analysis and implementation involved in automating the threshold selection task. This includes the validation performed by comparing the results to user selected thresholds.

The cross-talk analysis for the selection of collimators to align in parallel is presented in Chapter 7. This chapter discusses an initial solution for parallel selection, followed by a more advanced preliminary analysis for predicting the cross-talk

observed at a single collimator.

The software implementation and tools used to design, implement and commission the algorithms are explained in Chapter 8. This is followed by implementation details related to the software application and the user interface available.

Three major alignment campaigns were performed over the course of this dissertation, and these are discussed in Chapter 9. This chapter begins with a summary of the results from alignment campaigns prior to this dissertation, followed by a detailed analysis of the semi-automatic alignment campaign in 2017 and the two versions of the fully-automatic alignment campaigns in 2018.

Angular alignments are presented in Chapter 10, as an additional alignment technique that can significantly benefit from a fully-automatic approach. Three novel angular alignments methods were designed as part of this dissertation, starting off as a proof of concept and evolving into fully-automatic applications that were validated during LHC operation.

Finally, Chapter 11 summarizes the achievements of this dissertation, provides conclusions obtained from the results and includes suggestions for possible future work.

2. Background

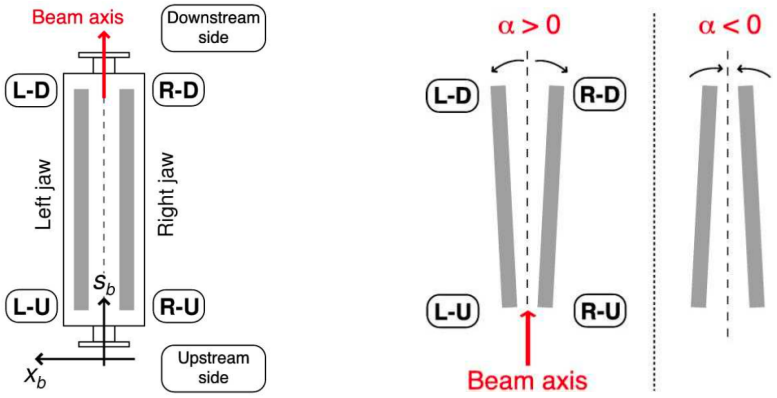
2.1 The LHC Collimation System

The collimation system provides the LHC with a 99.998% cleaning efficiency of all halo particles [4], meaning that unwanted halo particles need to be absorbed by the collimation system with a maximum leakage to the cold super-conducting magnets of 2×10^{-5} Zc. This is required in order to:

1. Protect superconducting magnets from quenching, by cleaning the beam halo throughout the beam cycle of the LHC.
2. Passively protecting the machine aperture from radiation effects as this might damage other hardware, such as electronics.
3. Reduce the halo-induced backgrounds in the experiments.
4. Serve as a diagnostic tool, such as for halo population measurements.
5. Abort gap cleaning to avoid spurious quenches after normal beam dumps.

An LHC collimator is composed of two parallel absorbing blocks, referred to as jaws, inside a vacuum tank. The jaws are made up of carbon fibre composite (CFC), tungsten (W), copper (Cu) or molybdenum carbide -graphite (MoGr), and these are

what scatter and absorb halo particles to prevent damage to the LHC. The jaws are identified as *left* or *right*, depending on their position with respect to the incoming beam. The collimators are installed with a fixed rotational angle, depending on their location and functionality, which allows to clean in either the horizontal (H), vertical (V) or skew (S) plane. The jaws must be positioned symmetrically around the beam and their coordinate system is displayed in Figure 2.1a. Each jaw can be moved individually using two stepping motors located in the jaw corners, allowing collimators to be positioned at different gaps and angles, as displayed in Figure 2.1b.



(a) Jaw coordinate system.

(b) Jaw angular tilt convention.

Figure 2.1: (a) The collimator coordinate system and (b) the jaw tilt angular convention as viewed from above, from [8].

The maximum and minimum possible operational angles are ± 1900 μrad [9]. The jaw corners are known as left-up (LU) and right-up (RU) when they are upstream of the beam and left-down (LD) and right-down (RD) when they are downstream of the beam. A fifth axis is also available to move the whole tank housing the jaws in the orthogonal plane, in case a region of the jaw surface would be damaged due to beam impacts, providing a new fresh surface. Linear Variable Differential Transformers (LVDTs) provide an independent measurement of these five settings, as well as the upstream and downstream jaw gap. Four resolvers count the steps of each motor, and ten switches are used to prevent the jaws from moving too far in or out.

2.1.1 LHC Collimation Layout

The LHC collimation system consists of around 100 movable collimators in the LHC ring. The collimators provide halo cleaning using a multi-stage hierarchy, as shown in Figure 2.2. The hierarchy is set up such that primary collimators (TCP) are placed closest to the beam to intercept the primary halo particles; the secondary collimators (TCSG) are retracted from the primary ones to clean secondary particles; the absorbers (TCLA) absorb the remaining showers; and the tertiary collimators (TCT) are further retracted from the TCSGs (but closer than TCLAs) to provide a local protection to the inner triplet superconducting magnets, which are installed around the IPs. Several injection protection collimators (TDI, TCLIA, TCLIB) follow a different hierarchy during injection and are further retracted when the injection is finished. Finally the TCDQ is a single-jaw collimator protecting the beam dump region in case of an asynchronous dump, by absorbing the beam swept over the mechanical aperture. In order to establish this cleaning hierarchy, the collimators must be aligned with a precision of less than 50 μm, which corresponds to a beam size of 0.25 σ at 6.5 TeV at the primary collimators, where σ is the Gaussian standard deviation. The TCDQ and TDI collimators are not considered for alignment with the rest of the collimators as their motorization is different from a normal collimator.

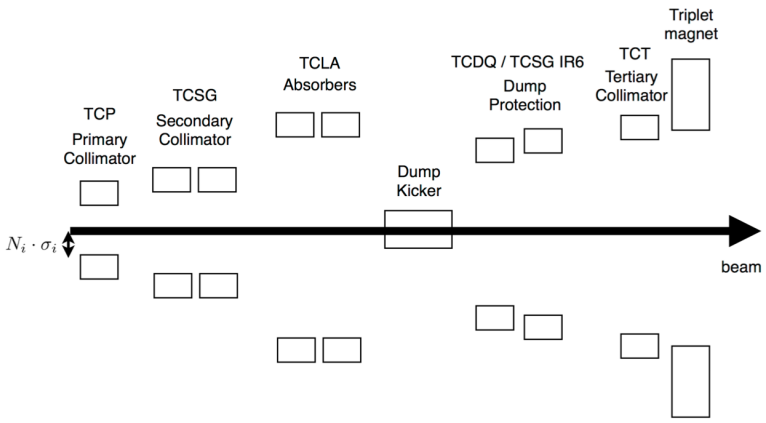


Figure 2.2: Collimation system hierarchy in the LHC, from [3].

The collimators are mainly concentrated in two dedicated cleaning insertion regions; IR7 enhances betatron cleaning, and IR3, where the dispersion is larger, enhancing off-momentum cleaning [6]. Figure 2.3 shows the collimators around the LHC whereby the official layout collimator names are dependent on a collimator's; cell number, position within IP (left or right), IP number and beam.

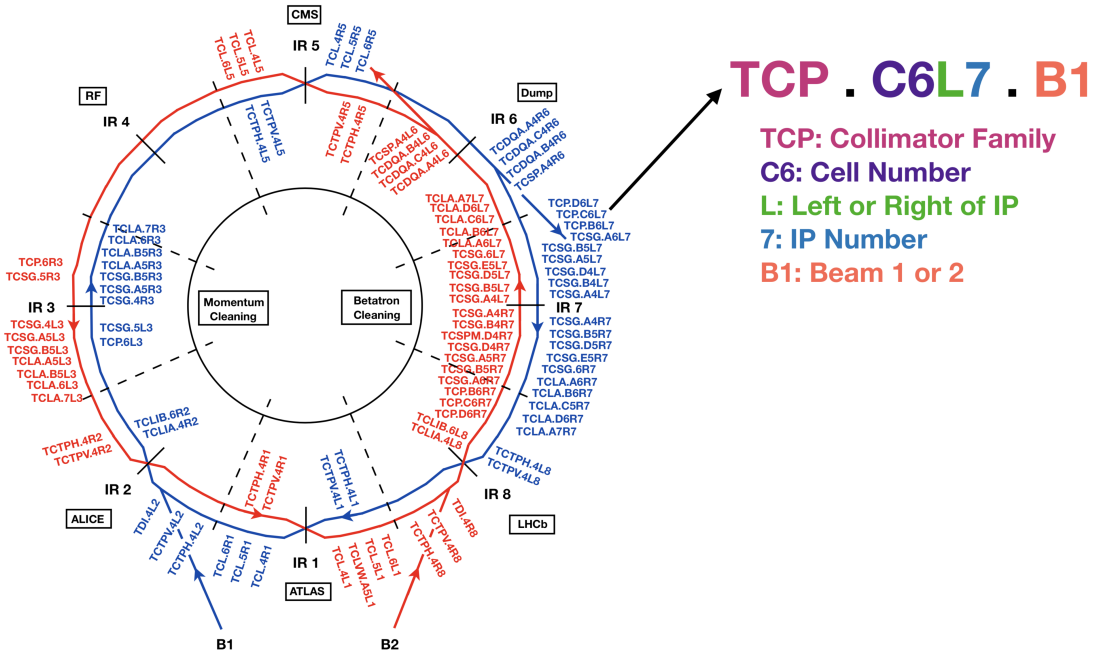


Figure 2.3: The present collimation system layout with the majority of the collimators in IR3 and IR7. Other collimators are placed in the dump and transfer regions and in the experimental IPs.

2.1.2 LHC Software Architecture

The software architecture designed for the collimation system is implemented via a 3-tier structure as shown in Figure 2.4. The bottom level consists of actuators, sensors and measurement devices, which allow for adjusting a number of parameters. These include the collimators left and right jaw positions and angles, with a minimum step size of 5 μm, which can reach up to 2.5% of the 1 σ beam size at 6.5 TeV, at the primary collimators.

The hardware is abstracted and controlled in real-time through the use of FESA

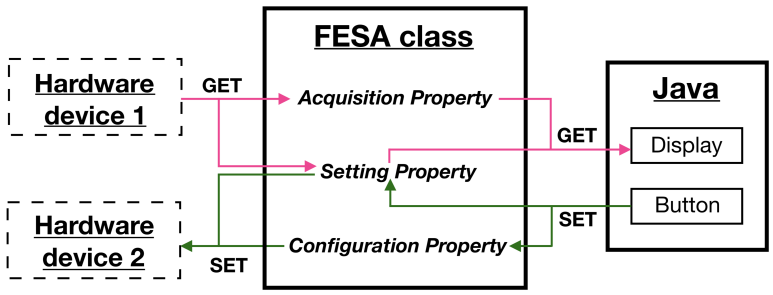


Figure 2.4: Software architecture diagram showing an example of FESA acting as the middleware between the Java application and hardware devices, from [10].

(Front-End Software Architecture) [11], a C/C++ framework developed in-house at CERN. It is used to develop LHC ring front-end equipment software which sends motor step commands [12]. FESA is a complete environment for equipment specialists to design, develop, test and deploy real-time control software run on front-end computers (FECs). The main objective of this framework is to standardise, simplify and optimise writing front-end software.

A hardware device is abstracted as a *FESA device* by exposing a public interface made up of *properties*. A property is made up of *value-items* which can be scalars, arrays of int, strings, and other simple types. A client can *get*, *set* and *subscribe* to a property in order to read, write and be notified on change, respectively. FESA provides *Acquisition properties* and *Configuration properties* for getting and setting, respectively, and *Setting properties* allow both getting and setting.

FESA devices are grouped into a *FESA class* which defines the; property interface, private data and real-time behaviour, for all devices belonging to that class. The FESA class handles hardware access through the use of *RTActions* which are triggered by an interrupt to execute in real-time and are subject to tight timing constraints.

The top level consists of Java Swing GUI applications which interact with the FESA middleware framework through the Java API for Parameter Control (JAPC) [13]. JAPC is an API used to build Java applications that control accelerator devices by interacting with the get, set and subscription device properties.

2.2 Beam Monitoring Systems

During LHC operation small amounts of particles are continuously lost, such that their location and loss signature give information about the conditions of the circulating beams. In order to monitor such losses and the actual beam orbit, the LHC is equipped with beam monitoring systems. Two types of beam monitoring systems are available to align collimators; the Beam Loss Monitoring (BLM) and the Beam Position Monitoring (BPM) systems.

2.2.1 LHC Beam Loss Monitoring System

The Beam Loss Monitoring system uses approximately 3600 Ionization Chambers (ICs) installed around the LHC ring, to detect ionizing radiation resulting from particle losses [14, 15]. If the detected losses exceed a pre-defined threshold, the BLM system is able to trigger a beam extraction to prevent damage to the machine.

Each collimator has a dedicated BLM device positioned outside the beam vacuum, immediately downstream. Such devices are used to detect beam losses generated when halo particles impact the collimator jaws, as shown in Figure 2.5. Recorded losses are proportional to the amount of beam intercepted by the collimator jaws, measured in units of Gy/s. Dispersion suppressors (DS) aim to reduce the machine dispersion inside the insertions, therefore the losses in the DS of IR7 and IR3 are key observables to evaluate the cleaning efficiency of the system.

The BLM detectors integrate the measured signals in 12 different time intervals, listed in Table 2.1. The integration time periods, or running sums (RS) range from 40 μs, which represents half an LHC turn, to 84 s which is equivalent to ∼9 million turns. The first RS is one step of 40 μs, and each n^{th} RS is a sliding window containing data from the previous 2^n windows. Each RS has a threshold associated, such that if the recorded losses exceed the threshold then the beam is dumped. The peak loss is published at a 1 Hz rate by RS01 to RS08, while the integrated loss is

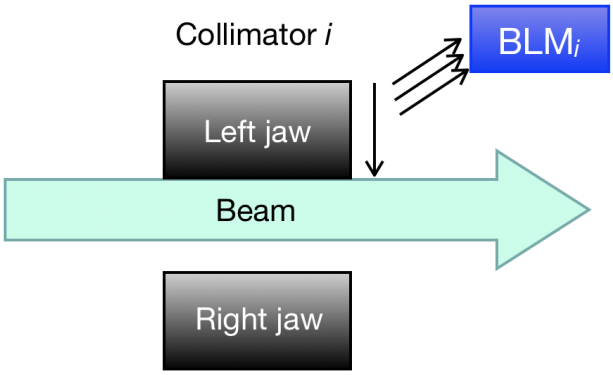


Figure 2.5: The collimator jaw scraping the beam halo and the scattered particles are detected by the BLM detector downstream, from [16].

published by RS09 to RS12. The collimation system uses the 1.3 s RS09 to analyse its performance, as it contains information about steady-state losses over thousands of turns [3]. The BLM data for all running sums is available at a rate of 1 Hz and as of 2016 the BLM data for RS06 has also been made available at a rate of 100 Hz.

Table 2.1: Running sum time windows, from [17].

Signal Name	**No. of 40 μs steps**	**Duration (ms)**
RS01	1	0.04
RS02	2	0.08
RS03	8	0.32
RS04	16	0.64
RS05	64	2.56
RS06	256	10.24
RS07	2048	81.92
RS08	16384	655.36
RS09	32768	1310.72
RS10	131072	5242.88
RS12	524288	20971.52
RS13	2097152	83886.08

2.2.2 Beam Position Monitoring at Collimators

The Beam Position Monitoring system consists of a set of electromagnetic pick-ups which are able to measure the beam position in the horizontal and vertical planes. A total of 1032 BPMs are installed in the LHC, corresponding to almost one BPM per quadrupole magnet. These devices are used to measure the beam trajectory within the machine, however are not located close enough to collimators to be used directly for collimator alignments.

During the long shutdown of the LHC in 2013, 20 collimators were replaced/introduced with a new design in which BPM pick-up buttons were embedded in the collimator jaws [18]. The BPM data is acquired from the button pick-ups via the Diode ORbit and OScillation (DOROS) system [19]. These embedded BPMs are able to provide a direct measurement of the beam orbit at the collimators by analysing the electrode signals, allowing for a safer and faster alignment without needing to touch the beam. However, BPMs are only available for a fifth of the collimators, therefore the remaining collimators must be aligned with BLM devices by touching the beam.

2.3 Beam-Based Alignment

Collimators aligned with BLM devices use a beam-based alignment (BBA) procedure, to determine the beam centre (Δx_i) and beam size (σ_i^{m}) at each collimator i. These calculations are necessary to be able to position the jaws within a certain number of standard deviations (beam σ) from the beam centre, as the actual beam orbit is susceptible to ground motion, thermal effects and machine sources such as multipole field errors [20]. Collimator alignments with BLMs are performed for 79 collimators at injection and flat top and 12 collimators in collisions, whereas 16 collimators aligned during the squeeze are aligned with embedded BPMs.

Collimators must be aligned before nominal operation with beam intensities above the damage limit, in order to ensure that the correct collimator hierarchy

is used and that the halo particles are cleaned. Figure 2.6 shows how the collimator jaws are placed symmetrically around the beam, and how Δx_i and σ_i^{m} are calculated.

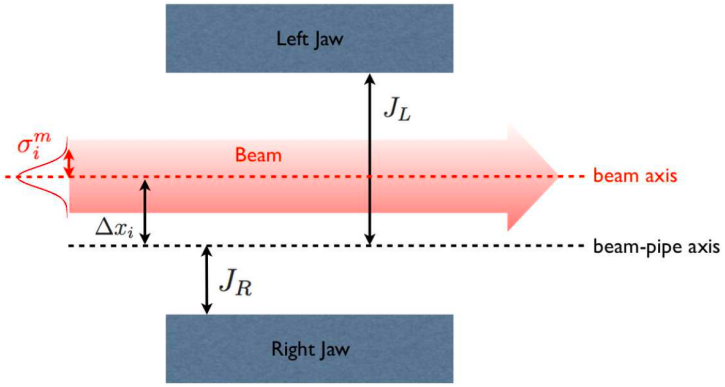
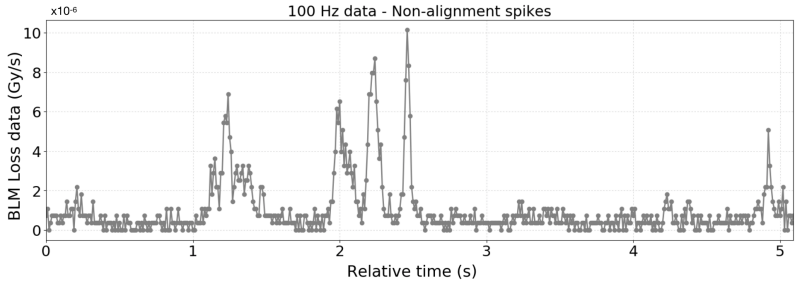


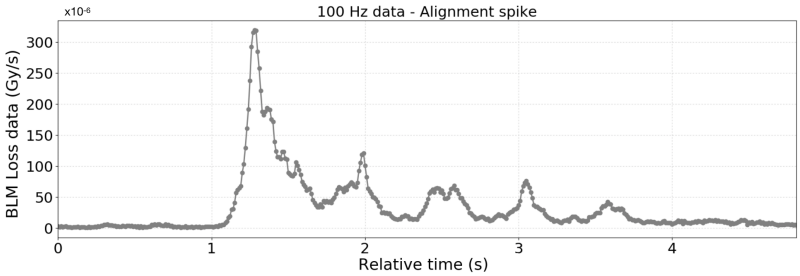
Figure 2.6: The beam surrounded by the collimator jaws where J_L and J_R are the left and right jaw positions, respectively, with respect to the beam-pipe axis, from [3].

The alignment procedure with BLM devices is referred to as beam-based as a collimator's jaws are moved towards the beam whilst observing the presence of any spikes in the beam loss signal of its respective BLM. The BBA procedure involves moving the jaws towards the beam in steps of 5 - 20 μm by following the beam loss signal of the collimator's respective BLM. The alignment of any collimator relies on being able to classify between alignment spikes and non-alignment spikes, such that a collimator must continuously move towards the beam ignoring any non-alignment spikes, until a clear alignment spike is observed. Figure 2.7a shows an example of non-alignment spikes indicating that the collimator has not yet touched the beam halo and must resume its alignment, whilst Figure 2.7b shows an example of a clear alignment spike indicating the collimator in question is aligned.

The beam-based alignment is performed via a four-step procedure established in [21]. The procedure was tested with a prototype collimator in the SPS [22, 23] and was used in the LHC from the start-up in 2010 onwards [24]. The standard sequence involves aligning a reference collimator in addition to the collimator in question (i), as is depicted in Figure 2.8. The reference collimator is taken to be the



(a) Non-alignment Spikes - The corresponding collimator is far from the beam.



(b) Alignment Spike - The corresponding collimator is aligned with the beam.

Figure 2.7: Typical BLM signals at 100 Hz showing (a) non-alignment spikes and (b) a clear alignment spike, after inward collimator movements at $t = 1$ *s*.

primary collimator in the same plane (horizontal, vertical or skew) as collimator i. This creates a "reference halo" that extends into the aperture of collimator i. The alignment of the collimators is beam-based as a collimator's jaws are moved towards the beam whilst observing the spikes in the beam loss signal of its respective BLM device. A jaw is classified as aligned when a signature spike pattern is detected in the losses, indicating that it reached a transverse position closer to the beam than the reference collimator.

1) The first step is to align the reference collimator in steps towards the beam to form a reference cut in the beam halo. 2) The second step is to align collimator i with respect to the reference halo generated. 3) The reference collimator is then realigned and 4) collimator i is retracted to its position in the hierarchy [25]. At each alignment step, the left and right jaws are aligned separately, in order to be able to associate the spike in the BLM signal to the particular jaw that was moving.

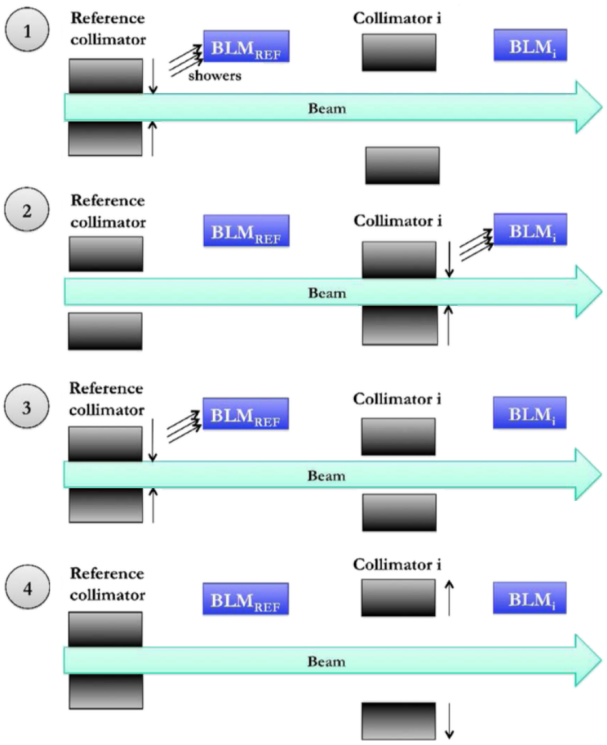


Figure 2.8: The four-stage beam-based alignment procedure for collimator i, using a primary collimator as a reference, adapted from [3].

The beam centre can then be calculated from the aligned jaw positions of collimator i using:

$$\Delta x_i = \frac{x_i^{L,m} + x_i^{R,m}}{2}, \tag{2.1}$$

where ${x_i}^{L,m}$ and ${x_i}^{R,m}$ are the measured left and right jaw setup positions. The jaw gap can also be calculated from these values using:

$$G_i = x_i^{L,m} - x_i^{R,m}, \tag{2.2}$$

The half gap is then used to calculate the measured beam size using:

$$\sigma_i^m = \frac{G_i}{(N\sigma_{ref_1} + M\sigma_{ref_2})/2}, \tag{2.3}$$

where $N\sigma_{ref_1}$ is the gap expressed in the number of standard deviations ($N\sigma$) of the reference collimator before aligning collimator i and $M\sigma_{ref_2}$ is the gap expressed

in the number of standard deviations ($M\sigma$) of the reference collimator after aligning collimator i. The gap at any collimator j is calculated using:

$$\sigma_{\mathrm{j}} = \sqrt{\varepsilon_{\mathrm{geom}} \cdot \beta(\mathrm{j})}, \tag{2.4}$$

where β is the nominal β-function defined by the magnets' lattice at the position of collimator j, and $\varepsilon_{\mathrm{geom}}$ is the nominal geometrical emittance that is inversely proportional to the Lorentz factor γ_{rel}:

$$\varepsilon_{\mathrm{geom}} \quad = \quad \frac{\varepsilon_{\mathrm{norm}}}{\beta_{\mathrm{rel}} \cdot \gamma_{\mathrm{rel}}}, \tag{2.5}$$

where β_{rel} and γ_{rel} are the relativistic factors, such that $\beta = v/c \approx 1$ for the case of the LHC, and $\gamma = E/E_0$.

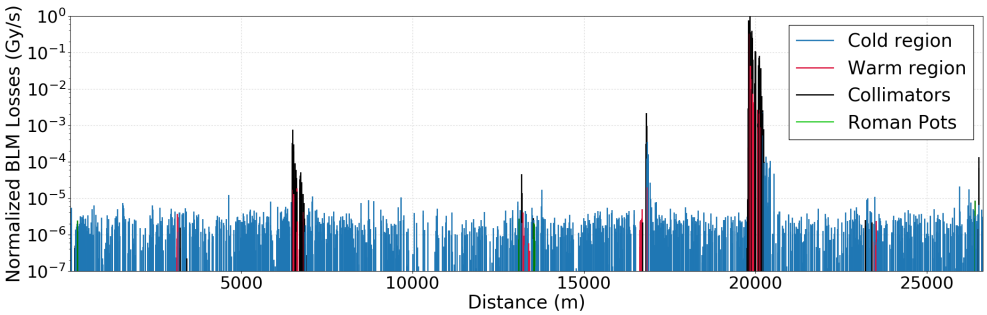
The final step is to then set the left and right jaws of collimator i to their respective positions in the collimation hierarchy, using the values obtained for the beam centre and beam size, using:

$$\mathrm{x_i^{L,set}} \quad = \quad \Delta \mathrm{x_i} + \mathrm{N_i}\sigma_{\mathrm{i}}^{\mathrm{m}}, \qquad\qquad \mathrm{x_i^{R,set}} \quad = \quad \Delta \mathrm{x_i} - \mathrm{N_i}\sigma_{\mathrm{i}}^{\mathrm{m}}, \tag{2.6}$$

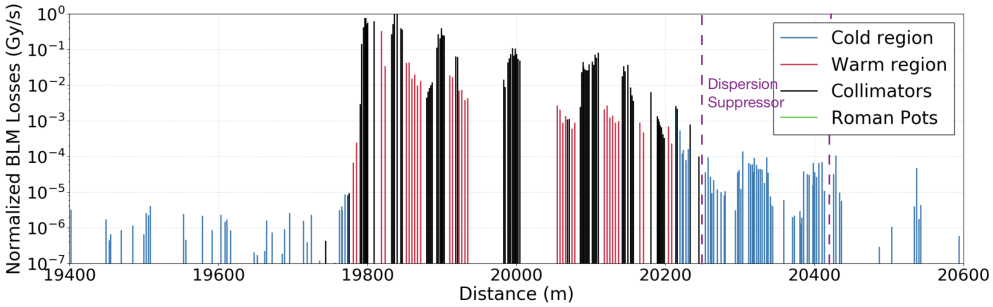
2.4 Collimation Hierarchy Validation

Over time, typically several months of operation, the beam orbit may shift, therefore the collimation system performance is monitored every three months of operation, or when the configuration is changed. This is done by inducing slow (multi-turn) beam losses, such that a large number of particles are purposely sent onto primary collimators in a controlled way, and the resulting electromagnetic showers can be detected by BLM devices around the LHC ring. A beam loss map, showing the spatial distribution of the measured losses along the LHC ring, can then be generated, as shown in Figure 2.9a, to validate the collimator setup. A zoom into IR7 is shown in Figure 2.9b, giving information on the collimation hierarchy and

halo cleaning performance.



(a) Beam 1 loss map for the full LHC ring.



(b) Beam 1 loss map zoomed in IR7.

Figure 2.9: An example of a beam 1 betatron horizontal loss map at 6.5 TeV, where the data points are readings taken from BLM detectors in (a) all IRs, and (b) IR7, highlighting the dispersion suppressor, from [26].

In case of a degradation in performance, the collimators might need to be re-aligned to generate a new setup for the current beam state. This could be observed as a degradation of the cleaning in the dispersion suppressor of IR7, or a difference in the pattern of the beam losses at the collimators. Collimator reconfiguration is also required when machine parameters are changed, such as the β^*, which defines the colliding beam size in the experimental points where the beams are brought into collisions, or the local orbit configurations around the IP [27].

2.5 Machine Learning Models

The proposed automatic alignment will make use of machine learning to automatically detect alignment spikes in the BLM data. The goal of the machine

learning model would be to automatically classify between alignment spikes and non-alignment spikes. A well labelled training data set can be extracted form previous alignment campaigns, therefore supervised classification learning will be used. Following is an overview of a number of machine learning models in this category.

1. Logistic Regression (LR) - Logistic regression [28] is a mathematical modelling approach used to describe the relationship of several inputs to a dichotomous dependent variable. The logistic model is based on the sigmoidal function:

$$\mathrm{f}(\mathbf{z}) \quad = \quad \frac{1}{1+\mathrm{e}^{-\mathbf{z}}} \tag{2.7}$$

   The logistic function, *f(**z**)*, ranges between 0 and 1 and is used to describe the probability and for the logistic model, $\boldsymbol{z}$ is written as the linear sum expression:

$$\mathbf{z} \quad = \quad \mathrm{w}_0 + \mathrm{w}_1\mathrm{x}_1 + ... + \mathrm{w}_\mathrm{D}\mathrm{x}_\mathrm{D}, \tag{2.8}$$

   where $\boldsymbol{x}$ is the input feature vector and $\boldsymbol{w}$ is the parameter vector with w_0 being the bias. Therefore the final model for logistic regression is:

$$\mathrm{f}(\mathbf{z}) \quad = \quad \frac{1}{1+\mathrm{e}^{-(\mathrm{w}_0+\sum \mathbf{w}_\mathrm{i}\mathbf{x}_\mathrm{i})}} \tag{2.9}$$

2. Artificial Neural Network (ANN) - An artificial neural network [29] is a collection of connected artificial neurons which loosely model the neurons in a biological brain, whereby each connection can transmit a signal from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it. It can be described as a series of functional transformations by first constructing M

linear combinations of input variables:

$$\mathbf{a}_\mathrm{j} \quad = \quad \mathrm{w}_0 + \sum_{\mathrm{i}=1}^{\mathrm{D}} \mathbf{w}_{\mathrm{ji}}{}^{(1)} \mathbf{x}_\mathrm{i}, \tag{2.10}$$

where $j = 1, ..., M$, the superscript *(1)* indicates the first layer of the network and $\boldsymbol{w}$ is the parameter vector with w_0 being the bias. The quantities a_j are known as *activations*, which are transformed using a differentiable nonlinear activation function:

$$\mathbf{z}_\mathrm{j} = \mathrm{h}(\mathbf{a}_\mathrm{j}) \tag{2.11}$$

These are referred to as *hidden units* and are linearly combined to give the output of activation units:

$$\mathbf{a}_\mathrm{k} \quad = \quad \mathrm{w}_0 + \sum_{\mathrm{j}=1}^{\mathrm{M}} \mathbf{w}_{\mathrm{kj}}{}^{(2)} \mathbf{z}_\mathrm{j}, \tag{2.12}$$

where $k = 1, ..., K$ such that K is the total number of outputs. This transformation corresponds to the second layer of the network (2). Finally the nonlinear functions $h(\cdot)$ for classification problems are generally sigmoidal functions, $f(\cdot)$, as defined in Equation 2.7, such that each output unit activation is transformed to generate the network outputs:

$$\mathbf{y}_\mathrm{k} \quad = \quad \mathrm{f}(\mathbf{a}_\mathrm{k}) \tag{2.13}$$

3. Support Vector Machine (SVM) - The support vector machine [30] algorithm works by maximising the margin between the training data points and the decision boundary. It typically makes use of a kernel function to directly perform all necessary computations in the input space. This involves mapping non-separable data onto a higher dimensional space in order to be able to use

a linear classifier. SVM classifiers are based on the class of hyperplanes:

$$(\mathbf{w} \cdot \mathbf{x}) + \mathrm{b} = 0, \quad \mathrm{w}\,\varepsilon\,\mathrm{R}^{\mathrm{N}}, \mathrm{b}\,\varepsilon\,\mathrm{R}, \tag{2.14}$$

where $\boldsymbol{w}$ is the weight vector, $\boldsymbol{x}$ is the input vector and b is the threshold. This corresponds to the decision functions:

$$\mathrm{f}(\mathbf{x}) \quad = \quad \mathrm{sgn}((\mathbf{w} \cdot \mathbf{x}) + \mathrm{b}) \tag{2.15}$$

The optimal hyperplane is defined as the one with maximal margin separation between the two classes. Since SVM makes use of kernels, its nonlinear decision function is of the form:

$$\mathrm{f}(\mathbf{x}) \quad = \quad \mathrm{sgn}(\sum_{\mathrm{n}=1}^{\mathrm{N}} \mathrm{v_i} \cdot \mathrm{k}(\mathbf{x}, \mathbf{x_i}) + \mathrm{b}) \tag{2.16}$$

Examples of kernels include the linear, polynomial and Radial Basis Function (RBF) kernels. The RBF is the most commonly used kernel as it is perceived to classify data with the best performance [31].

4. Decision Tree (DT) - A decision tree [32] makes decisions by assigning a probability to each of the possible choices based on the context of the decision, *P(f|h)*. A decision includes; f representing the set of choices and h the context of the decision. The probability *P(f|h)* is determined by asking a sequence of questions about the context, such that each question is uniquely determined by the answers to all previous questions. Therefore the selection process is a sequence of binary selections, corresponding to the traversal of the tree structure.

5. Random Forest (RF) - A random forest [33] is a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. Random

forests work by generating a random vector Θ_k for the kth tree, independent of the past random vectors, $\Theta_1, ..., \Theta_{k-1}$, but with the same distribution. A tree is then grown using the training set, resulting in a classifier $h(\mathbf{x},\Theta_k)$ where $\boldsymbol{x}$ is the input vector. The generalization error of random forest classifiers depends on the strength of the individual trees in the forest and the correlation between them. The number of features used in the splitting is determined by internal estimates to measure variable importance. After a number of trees are generated, each one votes for the most popular class.

6. Gradient Boost (GB) - Gradient boost [34] can be interpreted as an optimization algorithm on a suitable cost function. It is built in a forward stage-wise fashion, allowing for optimising arbitrary loss functions. At each stage, regression trees are fit on the negative gradient of the binomial or multinomial deviance loss function. A frequently employed loss function for classification is the negative binomial log-likelihood, $log(1 + e^{-2yF})$ where $y\,\varepsilon\,\{-1, 1\}$. Binary classification is a special case where only one regression tree is induced.

7. Hidden Markov Model (HMM) - A hidden markov model [35, 36] is a finite statistical Markov model whereby the system being modelled is assumed to be a Markov process with hidden states. It is composed of a number of states interconnected by state-transition probabilities and each state emits a residue according to the state's emission probability distribution. The probability that an HMM with parameters ϑ generates a state path π and an observed sequence S, is the product of all the emission probabilities and transition probabilities used:

$$\mathrm{P}(\mathrm{S}, \pi \,|\, \mathrm{HMM}, \vartheta) \tag{2.17}$$

Therefore the sequence of states is a Markov chain, as the choice of the next state depends on the identity of the current state. This state sequence is hidden and only the output is visible, hence the term 'hidden Markov' model.

2.6 Shower Propagation in the LHC

In the case of LHC beams, when a high energy beam particle interacts with the active region of a collimator jaw, the interaction effects are:

- The particle is lost, or survives with a different direction and energy. In the latter case the particles continue their path following a different orbit and are ultimately lost downstream along the ring.
- Secondary particle showers are generated, that propagate inducing thermal loads, mechanical stresses and radioactivity in the impacted area.

Secondary particle showers are generated by an incoming particle interacting with a material, thus producing multiple new particles with lesser energy, such that each of these then interacts in the same way, similar to a chain reaction. This process terminates once the generated particles do not have enough energy to interact inelastically with the medium to produce new particles [17]. Secondary particle showers have two main components:

1. The hadronic shower - This is generated when a hadron travelling through a material interacts inelastically and produces new hadrons at a lower energy.
2. The electromagnetic shower - Its main event is an electron or positron losing its energy by radiating a photon, which will then radiate other photons.

2.7 Particle Tracking with SixTrack

SixTrack is a particle tracking code written in Fortran-77, and is the main tool used for tracking particles, collimation studies in the LHC and beam cleaning studies [37]. It allows for tracking millions of particles for different halo types, taking into account the halo interaction with arbitrarily placed collimators of various collimator materials. As part of the collimation library, there is a scattering routine based on a Monte Carlo method that simulates all the physical interactions

between the hitting particle and the matter of the collimator jaw.

To perform SixTrack simulations, the following information is required as input:

- Machine lattice - Magnetic strength and sequence of the machine elements (including collimators).
- Collimator database - The details of collimator geometry, material and settings.
- Tracking parameters - Number of particles and turns, type of beam, type of halo.

The machine optics layout is defined by the MAD-X program [38], which is the standard tool to describe particle accelerators, simulate beam dynamics and optimise beam optics. The output of SixTrack for collimation is the list of the number of impacts on the jaw and absorbed protons for each collimator.

The LHC has energies up to 6.5 TeV therefore a small fraction of this energy is already sufficient to induce quenches of the superconducting magnets as well as material damage. As a result, it is vital to have a dependable and properly bench-marked simulation tool able to accurately predict the cleaning efficiency of the collimation system in future scenarios, to ensure the sufficient protection of magnets.

3. Literature Review

3.1 Collimator Alignments in other Colliders

The LHC currently has a nominal stored energy of 6.5 TeV per beam, giving it two orders of magnitude beyond other hadron colliders, as shown in Figure 3.1.

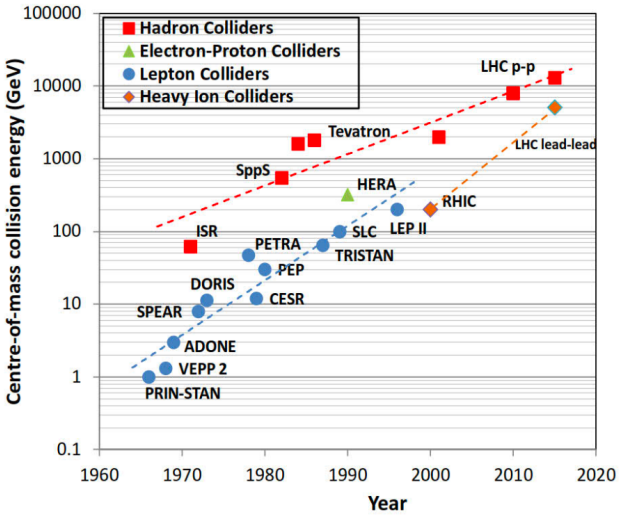


Figure 3.1: Centre-of-mass energy of particle colliders per year, from [39].

Any synchrotron is susceptible to beam losses and three colliders which operated or operate in high energy regimes, are equipped with beam collimation systems in order to protect the machine.

3.1.1 Tevatron

The Tevatron [40] was a high-energy proton-proton collider situated at Fermilab, Illinois, USA. In terms of beam energy, it was the closest machine to the LHC with 980 GeV. The Tevatron collimation system consisted of 4 horizontal and vertical primary collimators, and 8 secondary collimators, each of which had a single-jaw. The collimation system was used in a halo removal process which was applied at the start of every fill at 980 GeV. This involved moving the jaws into the beam halo using feedback from BLM detectors, and once the halo was removed, the jaws were retracted. For halo removal the Tevatron had shown to function optimally with the primary collimators positioned at 5 σ and the secondary collimators positioned at 6 σ. In addition, the minimum step size of the jaws was 25 μm (LHC is 5 μm) and the minimum time required for the halo removal process was 7 minutes [41].

3.1.2 RHIC

The Relativistic Heavy Ion Collider (RHIC) is a heavy ion collider situated at the Brookhaven National Laboratory in New York, USA. Its collimation system consists of 12 single-jaw L-shaped scrapers which are used for demands related to high experimental backgrounds and abort gap cleaning. Each beam has 2 primary collimators, and 2 horizontal and 2 vertical secondary collimators. The alignment is performed using loss monitors and the jaw is moved into the halo with step sizes of approximately 0.5 μm [42].

3.1.3 HERA

The Hadron-Elektron-Ring-Anlage (HERA) was an electron-proton collider situated at the Deutsches Elektronen-SYnchrotron (DESY) laboratory in Hamburg, Germany. Its collimation system was equipped with 6 double-jaw collimators, such that 2 were primary collimators and 4 were secondary collimators. These collimators were used to reduce the proton halo-induced background at the ZEUS and

H1 detectors. The aligmnent was done using pre-defined values, irrespective of the beam loss rates, or using an automatic procedure which moves the jaws in until the losses exceed a pre-defined threshold. In total, the alignment would take 20 minutes [43].

The LHC has higher energy and smaller beam sizes therefore the alignment precision required is much larger. Moreover, after the alignment of LHC collimators, the measured beam centres at the collimator locations are used by the LHC orbit feedback correction system, and the beam size measurements are incorporated into the collimator settings, both of which are not done for other colliders. In addition, the LHC has around 100 movable collimators, therefore an automatic approach for aligning them would definitely be beneficial.

3.2 LHC Semi-Automatic Beam-Based Alignment

The alignment procedure for the LHC collimators previously implemented in [3], uses a semi-automatic approach. This system makes use of a BLM-based algorithm [25, 44] which requires the user to make decisions as the alignment progresses. A typical semi-automatic alignment involves the following steps:

1. **The user** selects the collimator i to be aligned.
2. **The user** sets four input parameters:
    - Δx_i^L - Left jaw step size in µm, from a pre-defined list between 5 µm and 200 µm.
    - Δx_i^R - Right jaw step size in µm, from a pre-defined list between 5 µm and 200 µm.
    - S_i^{Thres} - Stop threshold in Gy/s, from a list of pre-defined thresholds between 1×10^{-7} and 2×10^{-4}.

- t_i^s - Time interval between each step in seconds.

3. **The user** selects which collimator jaw(s) to move towards the beam (left or right or both), and starts the alignment.

4. The jaw(s) of collimator i are automatically moved towards the beam in steps of Δx_i every t_i^s seconds.

5. The jaw movement automatically stops if the BLM losses exceed the threshold, S_i^{Thres} , by obtaining the BLM data S_i associated with collimator i after each jaw step.

6. Once the jaw(s) stop moving, **the user** is required to analyse the associated BLM losses in order to determine whether the collimator jaw(s) are aligned or not.

7. **The user** first moves both jaws towards the beam simultaneously, then separately aligns each jaw twice, until a clear alignment spike is observed in each case, by repeating steps 2-6.

8. Once a collimator is considered aligned, **the user** must select to save the position of collimator i so that the beam centre and beam size can be automatically calculated.

This semi-automatic alignment algorithm is depicted in Figure 3.2 as a flowchart. A clear difference from the full-automatic alignment proposed is that the current system only automates the movement of collimators towards the beam, and requires collimation experts to control the alignment, hence the term semi-automated.

Attempts were made to fully automate this alignment, however this was done from the user application level which was therefore not optimised for real-time behaviour, in contrast to the proposed solution which will be developed in a real-time framework (FESA). In addition, with the increase in energy from 3.5 TeV to 6.5 TeV, alignment spikes, thresholds selected and cross-talk between collimators,

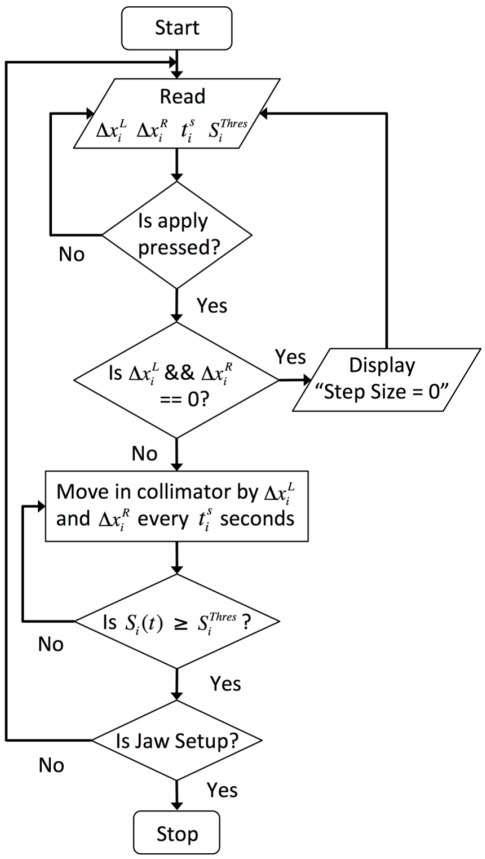


Figure 3.2: Semi-automatic beam-based alignment flowchart, from [25].

have changed therefore any related models need to be revised. Moreover, higher energies require tighter collimator settings therefore the automation will need to be extended to also include any required tilts in the jaws when aligning.

3.3 Pattern Recognition in Time Series Data

In order to automatically align a collimator, pattern recognition needs to be applied to detect spikes in the BLM signal, which will then determine how to proceed with the alignment. Methods working with time series data, such as EEG (Electroencephalography) and EMG (Electromyography), commonly apply pattern recognition to extract useful information.

3.3.1 Spike Detection in EEG

Electroencephalography is a method used to record brain activity through electrodes placed on the surface of the scalp. The signal recorded in EEG is a cumulative electrical activity of the brain in time. A second type of brain activity monitoring is electrocorticography (ECoG), or intracranial electroencephalography (iEEG), which uses electrodes placed directly on the exposed surface of the brain to record electrical activity from the cerebral cortex, thus making it more accurate [45].

EEG has many uses, however it is most often used to diagnose epilepsy, which causes abnormalities in EEG readings. Epilepsy heavily depends on the detection of interictal (between seizures) paroxysmal epileptic discharges (IPED), the so called spikes, in the EEG recordings. An EEG spike varies from the background activity as it has a pointed peak and lasts between 20-70 ms [46], as shown in Figure 3.3.

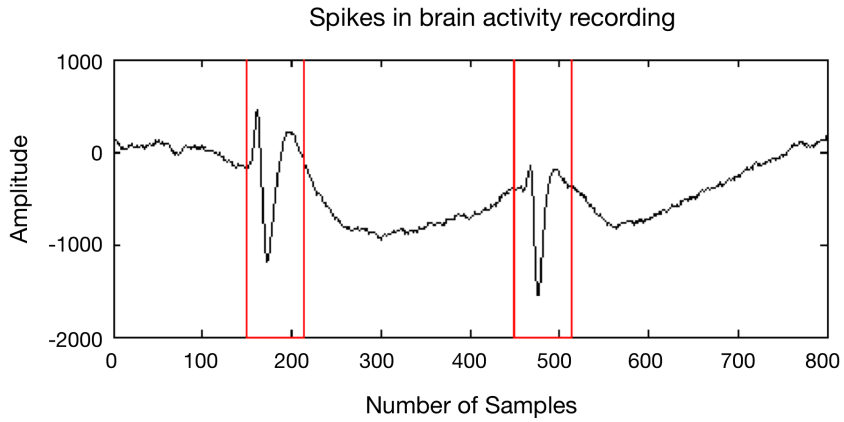


Figure 3.3: Spikes highlighted in EEG recording, from [47].

Manually detecting spikes is a time-consuming task, due to the large number of signals recorded, and requires training and experience. In order to enable a time-efficient evaluation of such recordings, various research has been done in search for algorithms to automatically detect spikes in EEG recording. Following is a number of machine learning models used in novel methods presented in recent papers, which seek to automate spike detection in EEG.

1. Random Forest - The algorithm presented in [48] makes use of a random forest binary classifier to detect spikes in both EEG and iEEG recordings. The data was first processed to remove influence from external perturbations using notch and bandpass filters, then the data set was checked to discard any non-optimal spike candidates. The extracted features were based on discrete wavelet transform (DWT) which allows for a multi-resolution analysis of the signal, providing localization in both time and frequency domains. The DWT resulted in 5 detail levels and the coefficients at levels 3, 4 and 5 (frequencies between 4 Hz and 32 Hz) were selected for the classification. This procedure resulted in 37 coefficients at level 3, 21 coefficients at level 4 and 13 coefficients at level 5, which made up the feature vector for each spike candidate. These feature vectors were inputted directly into the random forest classifier. The algorithm was trained and cross validated using 17 subjects with surface EEG and 10 subjects with iEEG. The method resulted in 62% recall and 26% precision for surface EEG subjects and 63% recall and 53% precision for iEEG subjects. These results indicate limited precision, however this can possibly be explained through a number of phenomena, therefore it was concluded that the proposed method has potential for diagnosis support in clinical environments.

2. Artificial Neural Network - An ANN model is used in the algorithm presented in [49]. This algorithm first processes the data by applying PureEEG algorithm [50] to remove common scalp EEG artifacts such as; muscle, movement, line noise and loose electrode artifacts. The signal is then segmented into single-channel segments which are then combined into multi-channel segments. For each multi-channel segment, features are then calculated from the wave characteristics, such as; the potential distribution, local context and signal morphology. The data set was built from 12 EEG recordings with an average duration of 10.9 hours, which resulted in 5582 spikes and 264193 non-spikes. The averaged performance of the ANN achieved a 44.1% detec-

tion sensitivity, 56.2% positive predictive value and a false positive rate of 19.8 per hour. These results were considered favourable results for a first implementation.

3. Support Vector Machine - The algorithm presented in [47] inputs frequency-band amplitude features into an SVM in order to detect spikes in iEEG signals. These features are based on the rhythms of brain waves, whereby different rhythms (delta: 1-4 Hz, theta: 4-8 Hz, alpha: 8-13 Hz, beta: 13-30 Hz, gamma: 30+ Hz) correspond to different brain states. These are extracted by first applying the Fast Fourier Transform (FFT) algorithm [51] to convert the data from its time domain to the frequency domain, then the features of delta, theta, alpha, beta and gamma frequency-bands are extracted. The data set used consists of 875 spikes and 2125 non-spikes, and 50% of each class was randomly selected as the test set. The averaged performance of this algorithm achieved 98.44% sensitivity, 100% selectivity and 99.54% accuracy, however this algorithm is limited, as it only uses single-channel (univariate) information for spike detection.

3.3.2 Motion Recognition in EMG

Electromyography is a technique used to evaluate and record electrical activity produced by skeletal muscles, i.e. the motors which allow the body to move. When a muscle contracts, electrical activity is generated within the muscle and this is recorded using electrodes placed on the surface of the skin. Muscle movement can also be recorded using invasive EMG, whereby a needle electrode is inserted directly into a muscle to record the electrical activity in that muscle [52].

EMG use cases include anything related to muscle movement, such as; diagnoses of Parkinsons disease, introducing prosthetic limbs and any robotic devices used to restore motor abilities. Figure 3.4 shows a number of forearm gestures which can be recognized through EMG. Such systems require an accurate recognition algorithm

for biological signal classification, therefore researchers have proposed a number of methods for recognising different gestures using machine learning.

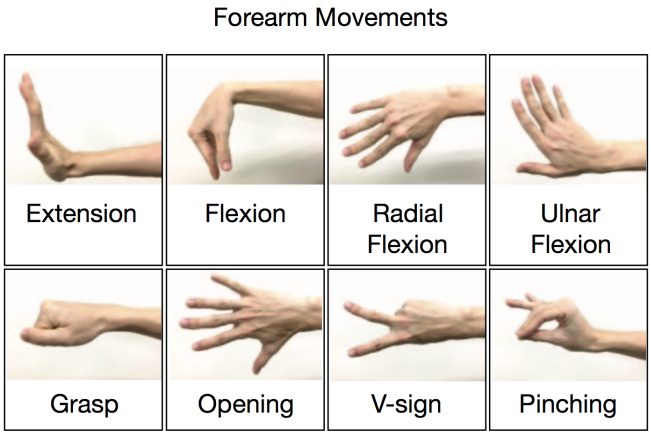


Figure 3.4: Forearm movements for motion recognition, from [53].

An important step when working with EMG signals is preprocessing to extract the relevant information. When analysing EMG signal data, one of the most common signal attributes extracted from the time-domain is the amplitude. This is done by calculating the Root Mean Square (RMS), as can be seen in a number of papers [54, 55, 56]. On the other hand, when applying pattern recognition, researchers have favored preprocessing using various filters, then input the filtered data into the machine learning model. Following is an overview of a number of machine learning models used for gesture recognition algorithms, presented in various papers.

1. HMM Forearm Motion Detection - The novel algorithm in [53] proposes to use an HMM based sequential pattern recognition to recognize unlearned classes on time-series data. Three right-handed males were used for the experiment, such that they performed eight forearm motions, four learned; wrist flexion, wrist extension, radial flexion, ulnar flexion, and four unlearned; grasp, opening, V-sign and pinching. The training data was made up of 1500 samples which were gathered from the learned motions performed three times each. The testing data was made up of all motions performed twice for approx-

imately three seconds. Feature vectors were extracted from the measured EMG signals based on previous work in [57]. The EMG signals were filtered and normalized before inputting them into the HMMs for learning and evaluation. The force information estimated from the patterns in the EMG channels [57] was used to determine the subject's motion timing. The model achieved high classification performance with both classes for EMG signals; 90.13%±3.80 (learned) and 91.25%±10.40 (unlearned). These results demonstrate the effectiveness of the technique for classification of unlearned classes.

2. SVM Knee Motion Detection - An SVM is used in the study presented in [58] with the aim of classifying different types of lower-limb motion. Fourteen healthy untrained subjects were required for the experiment in order to investigate three types of knee exercises; flexion of the leg up (standing), hip extension from a sitting position (sitting) and gait (walking). The raw EMG signals were first processed using a high-pass filter and the features were extracted using a WT-based Singular Value Decomposition (SVD) approach to build the feature vectors. The data set was then randomly split into 80% for training and 20% for testing. An SVM was used to build a multiple-subject classifier for the three types of lower-limb motions independent of the subject and the results indicate a classification accuracy of 91.85% ±0.88%. This method was compared to classifications done using features commonly extracted from time-domain (e.g.: Root Mean Square (RMS) and Zero Crossing (ZC)) and frequency-domain (e.g.: Mean Frequency (MNF) and Median Frequency (MDF)) and the proposed WT-based SVD approach obtained the best results.

3. NN Grasp Detection - The algorithm presented in [59] proposes a new EMG-based learning approach which makes use of a neural network to decode the grasping intention of the user at an early stage, i.e. before the final hand shape takes place. Fourteen right-handed healthy young subjects participated

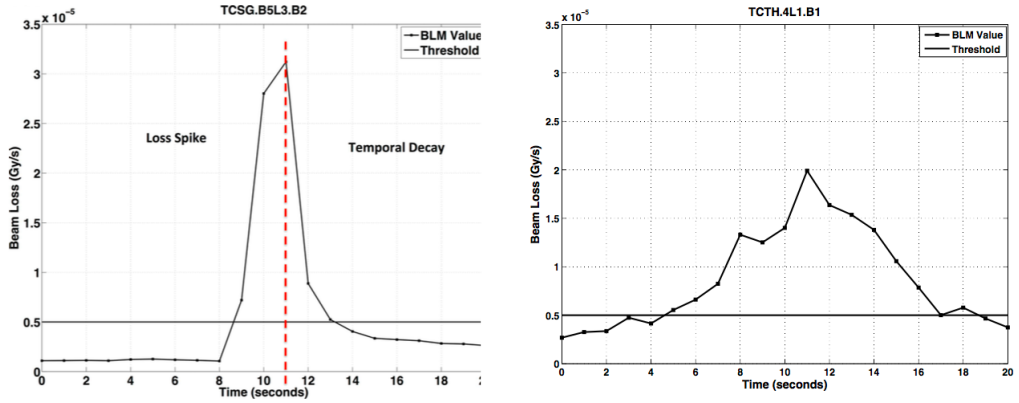
in the experiment to perform five different grasp types; precision disk, tripod, thumb-2 fingers, thumb-4 fingers, and ulnar pinch. Each subject performed 20 trials for each of the five grasp types and the recorded signals were filtered with a band-pass Butterworth filter and normalized. The Majority Vote criterion was applied to each time window from the motion onset, as suggested in [60], in order to assign a class label to the class that gathers the most votes. Finally, without extracting any features, the data was inputted into an Echo State Network (ESN) [61], a constructive learning algorithm for recurrent neural networks, that only modifies the weights to the output units. The data set was split into 75% training data and 25% testing data, and a number of classifiers were trained. Precision disk, tripod and ulnar pinch were distinguishable already at half motion (89.5%, 92.7% and 98.3% respectively), while thumb-2 fingers and thumb-4 fingers were distinct later in the reaching motion when the hand was closer to the object (85.1% for both of the classes). These results indicate that it is possible to decode the grasping intention in the early stages of the reaching motion and as expected misclassifications were observed between similar hand configurations.

3.3.3 Spike Detection in LHC Collimation

Applying pattern recognition in time series data brings us to LHC collimator alignments which will apply machine learning for spike detection in the time-series BLM signal, and this was studied in the past in [3].

The beam loss signal when a jaw touches the beam shows a sharp increase due to the scraping of particles from the beam halo, then the losses can be seen to gradually decay to a steady-state signal, as shown in Figure 3.5a. Any other patterns are referred to as non-optimal spikes, as shown in Figure 3.5b, and these would usually arise due to beam instabilities or mechanical vibrations.

Previous work done involved fitting a Gaussian function to the loss spike and a power function to the temporal decay component, as shown in Figure 3.6. Six



(a) Optimal Loss Spike

(b) Non-Optimal Loss Spike

Figure 3.5: 1 Hz BLM losses when aligning LHC collimators at 3.5 TeV, from [3].

features were extracted from the data, two of which were from the Gaussian fit and two from the power fit:

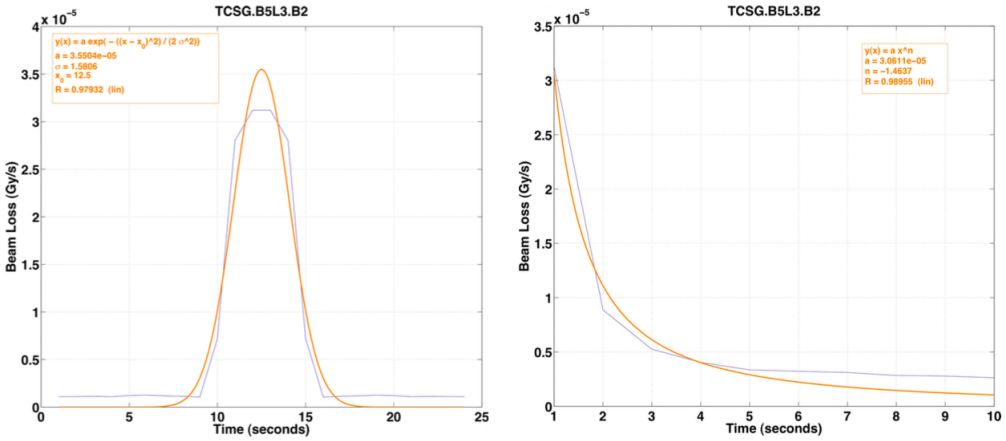
1. Maximum Value - The maximum of the ten BLM values observed after the jaws stopped moving.

2. Minimum Average - The average of the three smallest points taken from the seven loss points immediately preceding the maximum value. These smallest values allow for eliminating any spikes from a previous movement. An optimal spike generally has a high maximum value relative to the minimum average.

3. Variance - The width of the Gaussian fit, whereby an optimal spike would have a smaller width as this would reflect a sharp increase and a quick decrease of the losses. This value is equivalent to σ in the example shown in Figure 3.6a.

4. Gaussian Correlation Coefficient - The proximity of the loss pattern to the Gaussian fit, such that the closer this value is to unity, the sharper the loss spike. This value corresponds to R in Figure 3.6a.

5. Power Coefficient - A steep temporal decay indicates an optimal spike, equivalent to n in the example in Figure 3.6b.



(a) Gaussian Fit (b) Power Fit

Figure 3.6: (a) A Gaussian function fit to the loss spike folded about the maximum value, (b) A power function fit to the temporal decay starting from the maximum loss value, from [3].

6. Power Correlation Coefficient - The temporal decay becomes smoother as this value approaches unity, and this value indicates how well the power fit represents the loss pattern. This value corresponds to R in Figure 3.6b.

These features were scaled and inputted into a Support Vector Machine using the RBF kernel. A total of 444 samples were obtained from tertiary collimators and IR3 collimators at 3.5 TeV, which were split into training and testing sets. The final classification of the SVM on the test set achieved an accuracy of 82.4%, and it was decided that this was not adequate to use in operation.

This pattern recognition approach needs to be reviewed as this work was done at an energy of 3.5 TeV, and since then the energy has doubled, thus forming new spike patterns in the BLM loss signal. In addition, BLM data was acquired at a rate of 1 Hz in 2010 and 2011, and at 12 Hz from 2012 onwards, whereas nowadays data is available at 25 Hz and 100 Hz, therefore allowing for a more in-depth analyses of the loss spikes. Finally the goal is to achieve a higher classification rate so that the model can be incorporated into the proposed fully automatic alignment algorithm.

3.4 Threshold Selection

Automatically aligning a collimator involves automatically selecting the BLM threshold, such that the collimator stops moving when the losses recorded by a nearby BLM exceed this threshold.

3.4.1 Thresholding in EMG

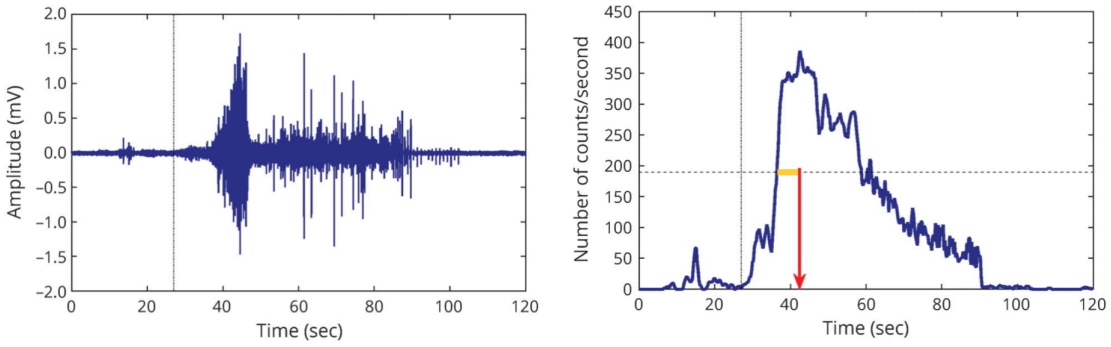
Thresholding is commonly used in EMG to provide feedback on the recorded signal. A use case example is the HMM forearm motion detection algorithm presented in Section 3.3.2. This algorithm determines the subject's motion timing using force information estimated from the EMG patterns, and if the force was larger than the predefined threshold, then the motion recognition was implemented.

Thresholding also plays vital roles in EMG studies, including:

- Subject Training - Threshold markers can be used in a number of ways to accurately perceive how tense or active a patient's muscle really is. This is required when patients have lost touch with whether a muscle is tensed or not due to problems with muscle-oriented pain and tension. In such cases a threshold is required as a target level during an EMG activity, whereby the patient tries to reach various thresholds multiple times. The aim of such activities is to train the patient to more accurately perceive whether his or her muscle activity is elevated above or relaxed below this threshold [52].

- Seizure Detection - Generalized tonic-clonic seizures (GTCS) may lead to injuries, and to sudden unexpected death in epilepsy (SUDEP), especially in unattended patients. The unpredictability of seizure occurrence is distressing therefore it is suggested to wear seizure detection EMG devices to prevent SUDEP. An example of such a wearable device is the Epileptic seizure Detector Developed by IctalCare (EDDI) which measures surface EMG signals.

Research presented in [62] implements an algorithm into this device which filters the raw EMG signal using a high-pass filter, and if the number of zero-crossings exceed a predefined threshold, then the alarm is triggered, as shown in Figure 3.7. The device is able to give real-time seizure alarms and has a sensitivity of 93.8% as it detected 30 out of the 32 GTCS that were conducted in the experiment.



(a) EDDI-EMG

(b) Zero-crossing

Figure 3.7: (a) The EMG signals recorded by Epileptic seizure Detector Developed by IctalCare (EDDI) during a generalized tonic-clonic seizure (high-pass filter: 150 Hz), (b) The number of zero-crossings calculated from the EMG signal, such that if they exceed the threshold for the prespecified time window, then the seizure alarm is triggered, from [62].

3.4.2 LHC Collimator Threshold Selection

Automating the selection of the threshold for the LHC collimators was previously studied in [63]. When selecting the threshold at time t = 0 the most recent values ($BLM_{t=-5}$ to $BLM_{t=-0}$) are given the most importance as these would indicate the latest level of losses. At the same time, the previous values till $BLM_{t=-19}$ must also be considered as these indicate the previous situation of the losses, in case the previous spike would still be decaying to its steady-state value. This can be expressed mathematically using an exponentially weighted moving average

(EWMA) over a 20 second window:

$$\mathrm{EWMA_{BLM}} = \frac{\sum_{i=1}^{20} e^{i} \times \mathrm{BLM_{t=i-20}}}{\sum_{i=1}^{20} e^{i}} \tag{3.1}$$

In total 475 samples were gathered, each consisting of a 20 second window of 1 Hz BLM data, and the corresponding threshold selected by the user. A power function was fit to the user thresholds as a function of the EWMA, generating a correlation coefficient of 0.96611. Based on this fit, the threshold was automatically calculated using:

$$S_i^{\mathrm{Thres}} = 0.53584 \times (\mathrm{EWMA_{BLM}})^{0.85916} \tag{3.2}$$

The data used was based on semi-automatic alignments performed in 2011 at 3.5 TeV at flat top. Since then, the energy has double and the data is now acquired and logged at a frequency of 100 Hz. As a result, the EWMA and user thresholds are no longer correlated, as shown in Figure 3.8, therefore this algorithm must be revised.

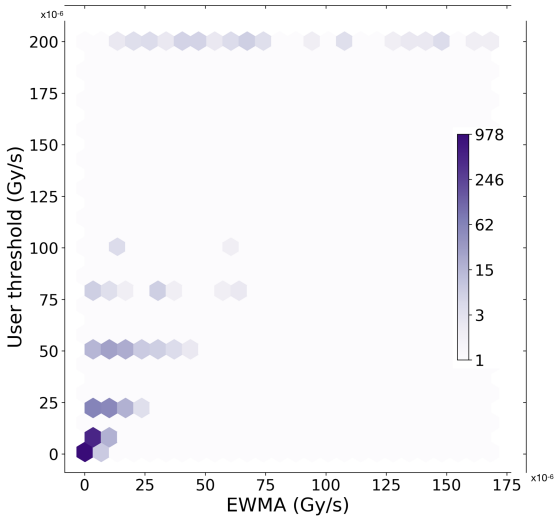


Figure 3.8: Loss thresholds applied before the start of jaw movements in 2018, as a function of the exponentially weighted moving average of the BLM signal, from [64].

Moreover, the waiting time between alignments was fixed at ∼10 seconds, ideally this should be decreased as much as possible to increase efficiency. Finally, this threshold selection algorithm was implemented at the user application level, whereas moving it to the server level would allow for faster execution.

3.5 Automating Time Series Data Systems

This dissertation proposes to transform the semi-automatic beam-based alignment into a fully-automatic one by closing the loop between the collimator stopping its movement after its losses exceed the threshold, and resuming the alignment based on the BLM loss signal. This involves using the feedback from the BLM devices in real-time to automatically select a threshold, and to automatically detect alignment spikes to proceed accordingly. Examples of such closed systems are Brain Computer Interfaces (BCIs) which use feedback from signals to automatically resume operation.

3.5.1 BCI Automatic Feedback Systems

Brain-computer interface systems use signals acquired from the brain to directly control external devices. The goal of BCI systems focuses on restoring motor function lost due to injury or disease of the nervous system, and improving functional independence for individuals with severe motor impairments, including providing tools for communication and mobility [65]. An example of a closed loop BCI application is depicted in Figure 3.9.

Following is a number of use cases of BCIs presented in recent papers, which seek to close the loop between neural feedback and system control.

1. Automatic Tremor Control System - Essential Tremor (ET) is the most common neurological disorder which causes uncontrollable rhythmic motions, thus making it difficult to perform everyday tasks. Current treatment involves deep brain stimulation (DBS) using a battery-powered implantable

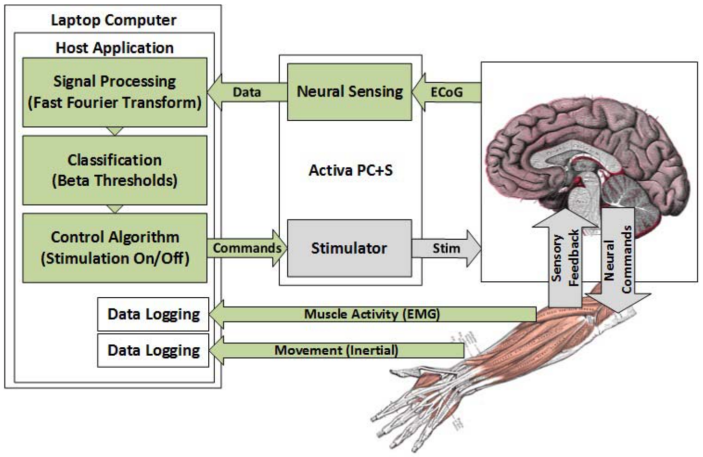


Figure 3.9: BCI system designed for analysing neural feedback data, whereby if the signal exceeds the predefined threshold, the stimulation is automatically turned on and off, from [66].

pulse generator (IPG). This system runs in an open-loop manner whereby the stimulation is delivered constantly at a steady state for as long as the battery lasts. The work presented in [66] seeks to close this loop by applying stimulation such that IPGs use sensors to collect data from the patient and automatically make stimulation adjustments as required. This system works by reading motor cortex ECoG readings in real-time, and if the signal exceeds the predefined threshold, the stimulation is automatically enabled and disabled accordingly. This was successful as the system accurately applied therapeutic stimulation to treat the tremor only when the patient was actively using the limb, thus can be used to extend the lifetime of surgically-implanted batteries.

2. Automatic Absence Seizure Control System - Seizure prediction in absence epilepsy remains unresolved due to the sudden onset of seizures. The research presented in [67] seeks to develop a real-time absence seizure prediction algorithm, implemented in a real-time closed-loop brain stimulation system for a genetic rat model. This research is aimed at preventing spike-wave-discharges (SWDs) which are typical for absence epilepsy. This system in-

volves recording the EEG readings of freely moving rats, and analysing the signals in real-time using the prediction algorithm, in terms of synchrony between the activity of brain structures. If the level of synchrony exceeds a predefined threshold, a stimulator is triggered to interrupt ongoing SWDs. Overall the algorithm correctly predicted 88% of the SWDs while the remaining were quickly detected. These results demonstrate that it is possible to use closed-loop SWD systems to detect local seizure activity in real-time and automatically interfere by stimulation before seizure-related clinical signs become manifest in the animal or patient.

3. Automatic Prosthetic Finger Control System - Patients with severe paralysis depend on prosthetic devices to provide a level of hand dexterity, to perform certain daily activities. At present, finger movement control has only been achieved in the context of coordinated movements of multiple fingers. A recent study presented in [68] seeks to provide instant online control of individual prosthetic fingers. The study involved analysing the recordings from an ECoG grid placed over a subject's sensorimotor cortex. The subject was asked to perform finger tapping tasks to collect data for training a linear discriminant analysis (LDA) classifier to automatically identify which finger was moved. The results achieved a peak online classification accuracy of 92% and 96.5% during peak offline classifications. Overall this study demonstrated that it is possible for a patient implanted with ECoG over sensorimotor areas to independently control all five individual fingers of a robotic prosthetic hand.

3.6 Cross-talk Analysis

Once the fully-automatic alignment is implemented, the final step would be to parallelise it by aligning multiple collimators at once, in order to gain time. This is not a straightforward task, as the cross-talk effect must be taken into consideration,

whereby losses generated by a collimator are not only detected by its corresponding BLM, but also by other BLM detectors around the LHC. As mentioned in Section 2.6, when a particle reaches the primary collimator, it can either have an inelastic interaction and get absorbed, thus starting a hadronic or electro-magnetic shower that can be detected by several BLM devices (including BLM devices in the other beam), or interact elastically, thus changing its momentum and continue its trajectory. Therefore before attempting to parallelise the alignment, the cross-talk across various collimators must be studied, by tracking the loss propagation in the IRs.

3.6.1 Cross-talk in EMG

Cross-talk is commonly observed in surface EMG recordings, whereby an electrode positioned over a specific muscle detects signals from neighboring muscles, as shown in Figure 3.10. Cross-talk limits surface EMG applications, especially in cases where it is important to know the timings different muscles are activated, such as in movement analysis. Therefore reducing cross-talk will help in identifying activities of specific muscles [69].

Attempts have been made to investigate cross-talk across muscles, such as the work presented in [69]. This work seeks to asses the cross-talk between two muscles, by placing surface electrodes over one and checking for action potentials generated by motor units of the other muscle, and vice versa. A total of 66 motor units from both muscles were studied using both invasive and surface EMGs. Visual inspection of the data revealed it was not possible to identify which motor units triggered which spikes. The data was analysed using the RMS amplitude, for all combinations of inter-electrode distances and pair of electrodes and muscles. As explained in Section 3.3.2, RMS is commonly used to extract useful information from EMG recordings, and in fact qualitative differences can indeed be observed between the surface potentials of the two muscles by comparing their RMS amplitude.

On the other hand, there are other research cases which observe muscle activity

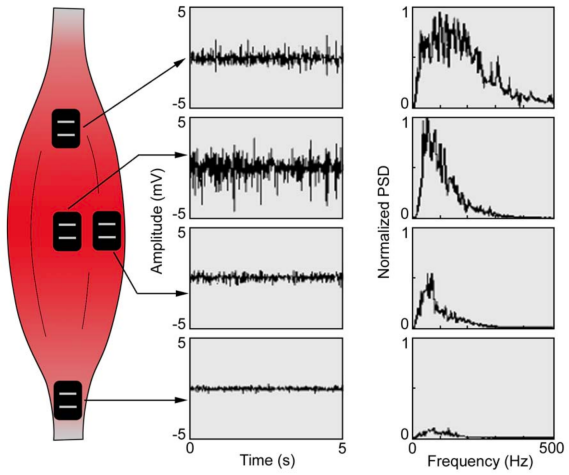


Figure 3.10: The amplitude and frequency spectrum of the EMG signal is effected by the location of the electrode. One can clearly see that the two electodes placed at the lateral edge of the muscle are effected by cross-talk from the electrode above them, from [70].

using surface EMG, and if any signals are suspected of suffering from cross-talk, then they are discarded. An example of such work is presented in [71], which involves analysing thigh-muscle activity in children. In this study, if two muscles showed similar activity simultaneously, then additional differential probes were added to confirm if cross-talk is present, and if cross-talk is confirmed, then the signal is discarded.

3.6.2 Cross-talk in LHC Collimator Alignment

The ratios of the spikes in BLM signals of stationary collimators to the spikes in BLM signals of an aligned collimator, were used to construct a model based on alignments done in 2011 and 2012. 620 spike samples of 21 seconds each were extracted from the alignment data, whereby only one collimator was moving during the time interval and the BLM data of all collimators was extracted, as shown in the example in Figure 3.11.

The relative position and jaw half gap of the primary collimator and ratios were used in RapidMiner, a software suite which allows for quick data import, processing

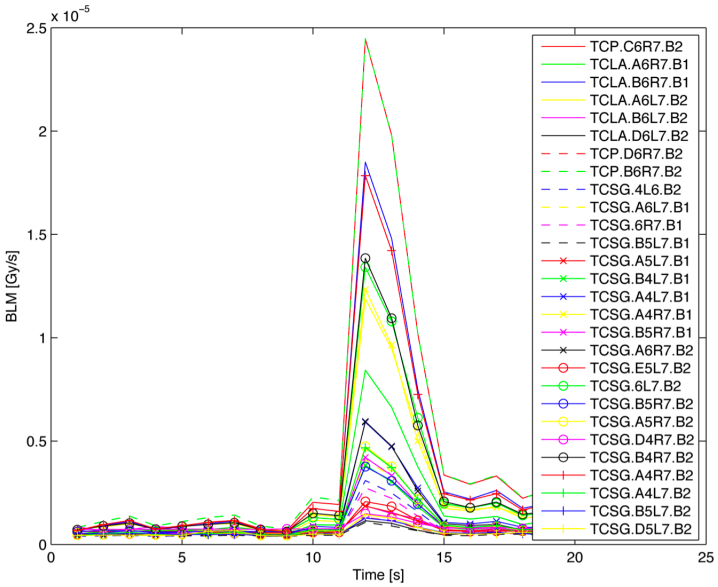


Figure 3.11: BLM signals observed with aligned collimator: TCP.C6R7.B2, from [3].

and output [72]. The aim was to use the k-NN algorithm to develop a model which can predict the factor that needs to be applied to the BLM signal of an aligned collimator, for another collimator's BLM detector at a given relative distance and with a given jaw half gap. This algorithm performs a regression which predicts a value depending on the closest class and a test-to-train ratio of 50/50 was used. This resulted in a model for each collimator BLM detector.

The aim of this cross-talk analysis was for simulation purposes, whereas the proposed automatic alignment requires cross-talk analysis for alignment parallelisation. Furthermore, new cross-talk analysis is required as the collimation layout, orbit and beam parameters have been changed.

3.7 Parallel Alignment of LHC Collimators

Parallelising the alignment of LHC collimators was previously studied in [3], to optimise the setup time by simultaneously aligning any number of collimators.

The algorithm used for parallel setup makes use of a timer task *CheckColls* to check if any collimators have stopped moving. Once any collimator has stopped moving due to its BLM losses exceeding the threshold, another timer task *Check-*

CollsT is started to check if any other collimators also stop moving within the pre-defined time period *T = 2 seconds*. This time period was selected to be twice the BLM data readout rate at the time (1 Hz). In this case, all other collimators moving in parallel are stopped and the algorithm focuses only on the first collimator. This collimator resumes its alignment, and the user can increase the threshold as required. Once the collimator is aligned, the algorithm terminates and allows the user to start the sequential alignment. The parallel collimator setup algorithm is depicted as a flowchart in Figure 3.12, and the variables it uses are described in Table 3.1. This algorithm was tested during re-commissioning in 2011 [73].

The main difference between this method and the new parallel alignment proposed, is that the former algorithm seeks to find a solution after cross-talk has potentially been observed, whilst the latter algorithm seeks to avoid any cross-talk altogether, thus aiming for prevention rather than cure.

Table 3.1: Variables used in the parallel algorithm in Figure 3.12, from [25].

Variable Name	**Description**
CheckColls	Thread that polls the collimator status every second.
S_i^{Thres}	The user-specified loss threshold.
colls	An array of references to the stopped collimators.
CheckCollsT	Checks if other collimators stopped within time interval *T*.
T	*colls* lists collimators stopped within time interval *T*.
Δx_i	The jaw step size in μm.
steps	The number of steps taken by a collimator.
$steps_{max}$	Max steps taken by a collimator until exceeding threshold.
incrThreshold	If true, if losses are too high, the threshold is increased.
S_{step}^{Thres}	The threshold increment value.
S_{max}^{Thres}	The maximum threshold that can be set.

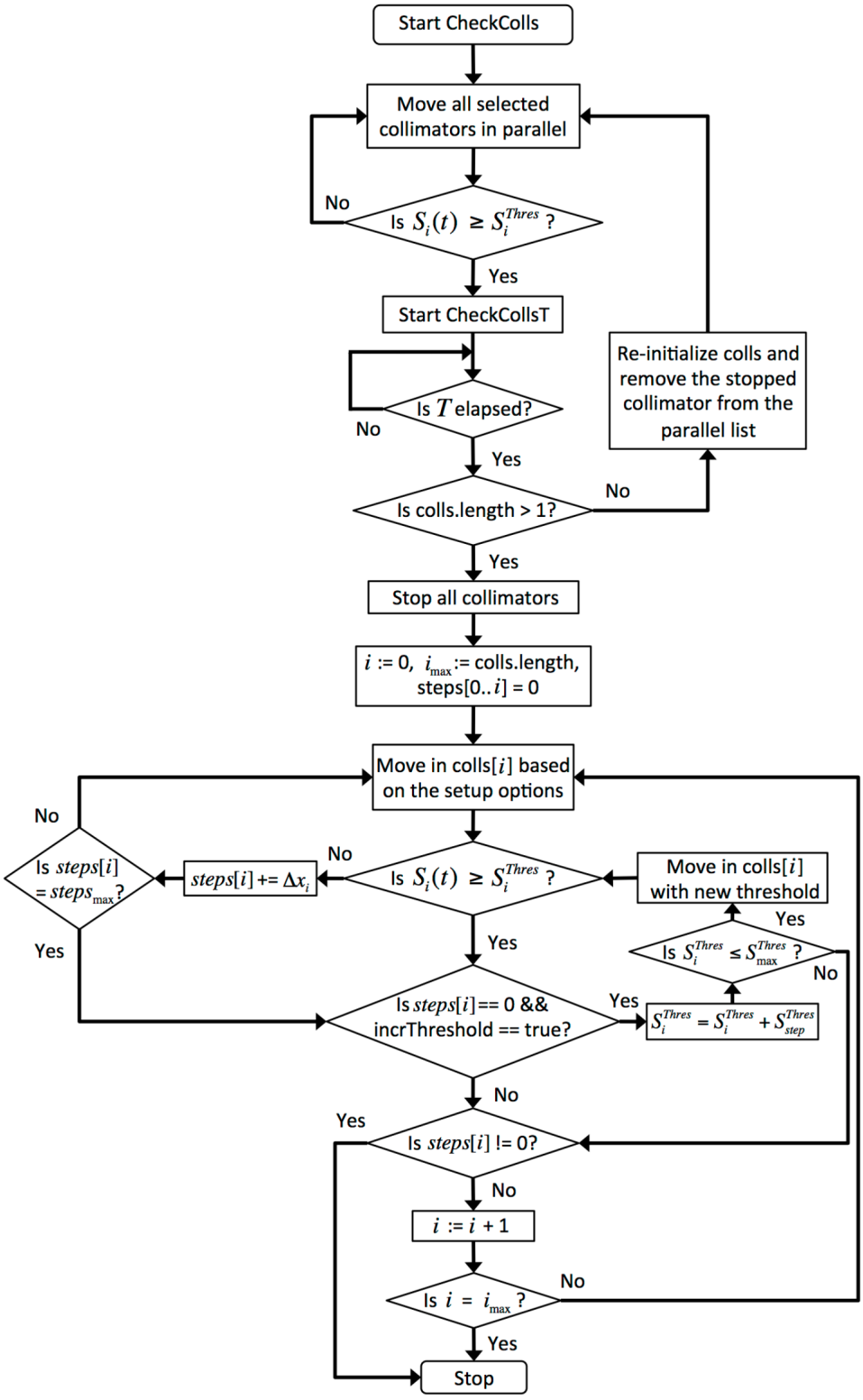


Figure 3.12: A flowchart of the algorithm able to identify which collimator jaw touched the beam after multiple collimators stop moving due to cross-talk, from [25].

4. LHC Collimator Alignment Problem Formulation and Constraints

---

This chapter defines the goals and constraints of the alignment task. This will aid in quantifying the objectives which in theory can be met by fully-automating the alignment, as well as the constraints that restrict the alignment performance. Moreover, the associations between different components of the alignment problem are defined, thus providing a fundamental understanding of the basic alignment.

4.1 Beam-based Alignment Overview

Aligning a collimator starts off by moving both jaws towards the beam simultaneously, to save time in cases where the jaws are far out. Following this, the collimator jaws must be aligned one at a time to be able to associate the BLM losses with the particular jaw. Therefore the left jaw is first aligned until an alignment spike is observed, followed by the right jaw until an alignment spike is observed. This is repeated for each jaw to obtain a second alignment spike, to ensure that the colli-

mator has indeed touched the beam. This process is depicted in the state machine in Figure 4.1.

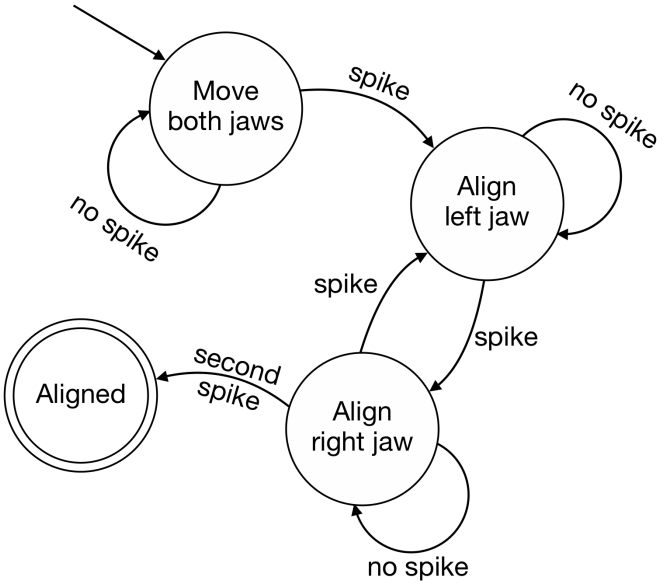


Figure 4.1: State machine of the jaw movements for aligning an individual collimator, from [64].

Aligning a collimator i in plane p, unless it is a primary collimator, requires the primary collimator in its respective plane to be aligned before and after as explained in Section 2.3. Once this is done the settings of the collimators are saved, and this is depicted in the state machine in Figure 4.2.

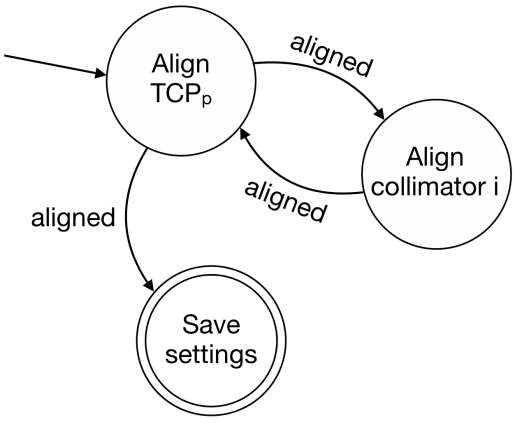


Figure 4.2: State machine of BBA with primary collimator.

4.2 Review of Time Performance

The principal assumption is that having a fully-automatic approach will allow for achieving the minimum time possible required for aligning collimators. In theory, the minimum time required for an alignment campaign with BLMs, excluding additional time required for misalignments, would be at injection, as aligning at flat top requires more caution. The alignment time is composed of:

- The time to move the collimator jaw from the initial transverse position until it finds/touches the beam. This is driven by the maximum jaw movement speed that is 1 mm/s.
- The time to measure the beam loss spike, including the decay time which depends on the beam energy. The rate of diffusion of particles at flat top (6.5 TeV) results in longer loss signal decays when compared to the decays at injection (450 GeV) [74]. From operational experience, we assume that beam losses would have decayed within 6 seconds at 6.5 TeV and within 4 seconds at 450 GeV.
- The time to classify the spike, by applying pattern recognition to identify whether there is an alignment spike.
- The time to align the two collimator jaws individually, by observing 2 spikes per jaw. An average movement of 100 μm per jaw is required (50 μm per spike).
- The time to re-align the primary collimator jaws. This collimator is already defining the beam halo, therefore only one step per jaw is required.

Table 4.1 quantitatively summarises the alignment and indicates that the theoretical minimum time required to align a single collimator at injection is 53.8 seconds, assuming that every spike is an alignment spike and the collimator is not tilted. Therefore aligning all 79 collimators at injection needs at least 1 hour 11

Table 4.1: Theoretical minimum time to align a collimator at injection in steps of 10 μm with a time interval of 0.02 seconds, and starting from 8 mm position.

Action	**Time (s)**	**Comments**
Move both jaws to 4 mm	8	Assume clear alignment spike first try
Wait for losses to decay	x	Take x as arbitrary wait time
Classify spike	1	Time to classify spike through FESA
Align Left Jaw Align Right Jaw	$2(0.1 + x + 1)$ $2(0.1 + x + 1)$	Assume no tilt and beam centre $\sim= 0$ => 2 clear spikes after $\sim$10 steps each
TCP before (Left Jaw) TCP before (Right Jaw)	$0.1 + x + 1$ $0.1 + x + 1$	Assume primary was aligned before => both jaws with instant clear spikes
TCP after	$2(0.1 + x + 1)$	Left and Right jaws
Total	$\mathbf{17.8 + 9x}$	At injection $x >= 4$, flat top $x >= 6$

minutes. On the other hand, if the collimator jaws encounter one non-alignment spike at each alignment, then each unique jaw alignment would require an extra 5.1 $(0.1 + x + 1)$ seconds, resulting in an additional 15.3 seconds per collimator. Therefore the minimum time required at injection in this case would increase to 1 hour 31 minutes.

4.3 Alignment Time Optimization

Ideally, a collimator jaw only stops moving when it has touched the beam and a clear alignment spike can be observed. The collimator jaw stops moving when its losses exceed the threshold, therefore if the threshold is set high enough to ignore all noisy spikes, then this would avoid; unnecessarily stopping, waiting for the losses to decay and triggering the machine learning model. This would save a significant amount of time from having to classify multiple noisy spikes before finally achieving an alignment spike.

The minimum time calculated in Section 4.2 assumes all collimators are aligned sequentially. Parallelising the fully-automated alignment would further decrease the alignment time, however not all collimators can be aligned in parallel due to

the cross-talk between them. Therefore the aim is to align as many collimators as possible in parallel, whilst at the same avoiding any cross-talk, as otherwise any collimators involved in cross-talk would need to be re-aligned sequentially, thus taking more time.

4.4 Control of Beam Losses

When performing alignments, a beam intensity limit is set by machine protection to prevent from damaging the machine. This limit is 3×10^{11} p/beam at top energy, which corresponds to $\sim$0.1% of the nominal LHC beam. Figure 4.3 shows the beam intensity decreasing with each collimator alignment, therefore although the alignment should be done as fast as possible, each alignment must preserve the beam intensity to align all collimators with the same beam. The larger the step size taken by each jaw, the faster they move towards the beam, however the amount of beam particles scraped would be larger. The beam lifetime depends on the losses generated by the alignment, and the alignment cannot take longer than the lifetime of the beam. Therefore the aim is to find a compromise between moving fast towards the beam, and scraping away as little beam as possible. In fact, at flat top, alignments are done more cautiously by using a smaller step size and a slower time interval, as the beam size is smaller due to higher energy and one cannot afford to scrape away extra beam.

Although selecting a higher threshold results in a faster alignment, the primary reason for using a threshold is to stop the jaw movement before it scrapes away too much beam, and as a result dumps the beam. Therefore if the threshold is set too high, and the jaw moves too close to the beam at a very fast rate, the losses observed by the BLM devices may exceed their dump threshold and as a result trigger a beam dump. If the beam is dumped, it would take time to re-inject the beam, and if the alignment was being done at flat top, it would take over an hour to bring the beam back to flat top.

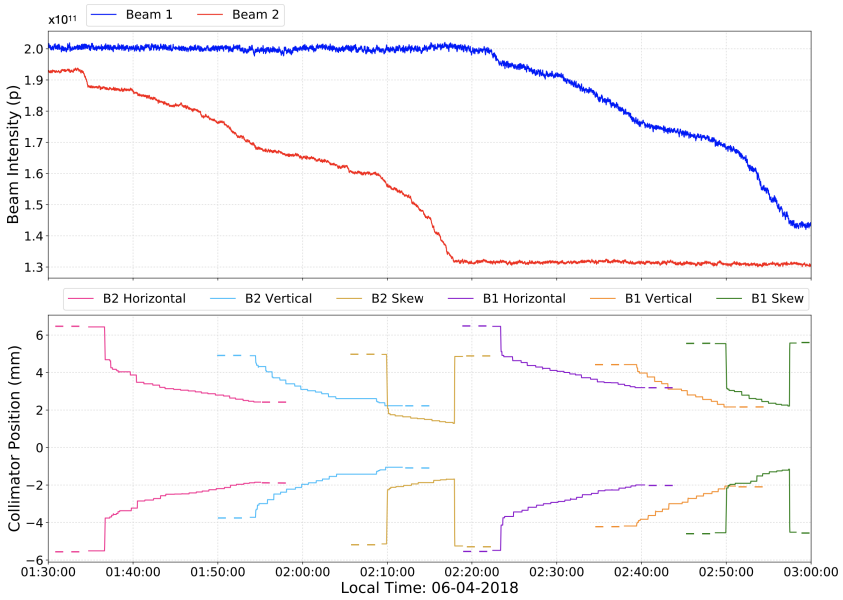


Figure 4.3: Beam intensities decrease as collimators are aligned (only TCP collimators are shown for simplicity), from [75].

4.5 Proposed Solution

In order to satisfy these requirements and fully automate the alignment, three dedicated algorithms are implemented as part of this dissertation, based on their own respective data analysis:

1. Automating the process of spike recognition can be cast as a classification problem, by training a machine learning model to distinguish between alignment and non-alignment spikes.

2. The threshold selection process can be automated by mimicking the BLM signal analysis performed by collimation experts when manually selecting the threshold.

3. The cross-talk effect across collimator BLM devices can be analysed to automatically select collimators to align in parallel.

5. Automatic Spike Recognition

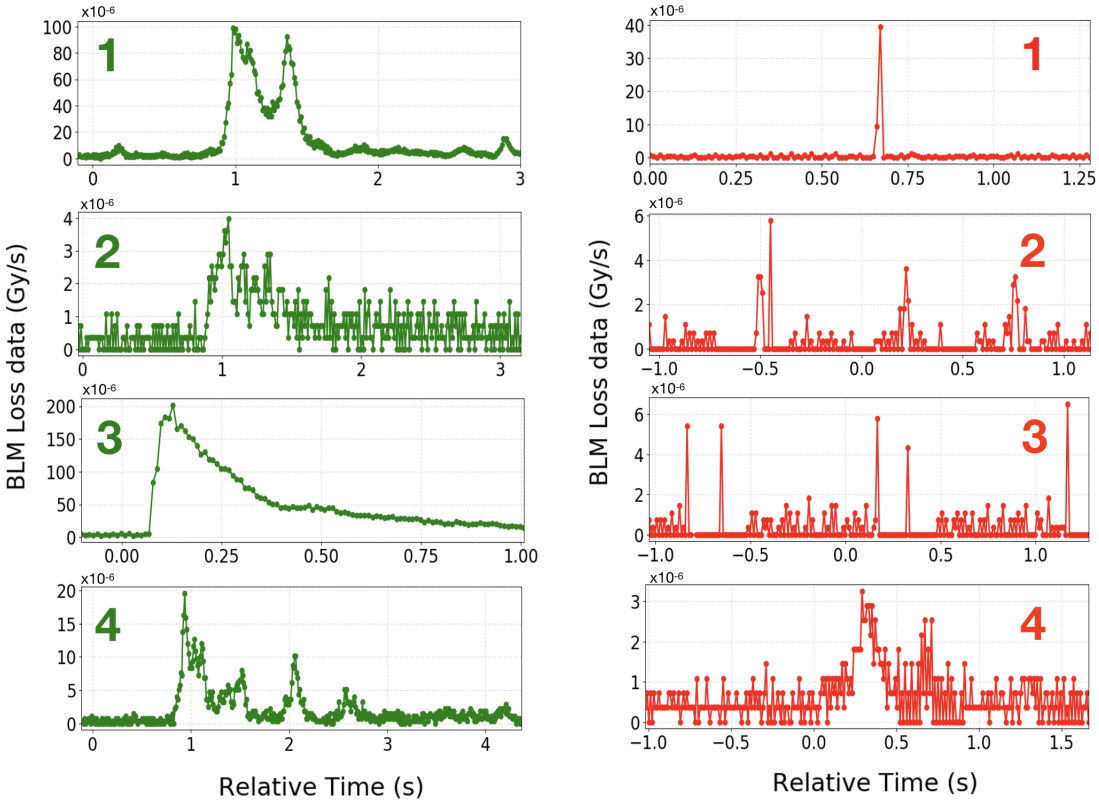
The semi-automatic BBA requires a collimation expert to visually analyse the BLM loss signal to determine if a collimator is aligned or not. A collimator stops moving when its losses exceed a predefined threshold, and is classified as aligned when both its jaws are touching the beam, by individually generating alignment spikes in the BLM loss signals.

5.1 Beam Loss Signals in Collimator Alignments

To fully-automate the BBA, distinguishing between the different spikes, as shown in Figure 5.1, must be automated. In each of these cases, the BLM devices recorded a rapid increase in beam loss, which is high enough to exceed the threshold. However, the signals in Figure 5.1a were generated by a collimator movement touching the beam, indicating that the collimator is aligned, whereas the signals in Figure 5.1b are not generated by the collimator, indicating that it has not yet touched the beam and must continue moving.

As shown in Figure 5.1a, alignment spikes can have various shapes, including;

- The maximum value can be high (as in graphs 1,3,4) or low (as in graph 2)
- The decay of losses can be short (as in graphs 1,2,3) or long (as in graph 4)



(a) Alignment spike class.

(b) Non-alignment spike class.

Figure 5.1: Examples of various beam loss shapes classified as (a) alignment spikes and (b) non-alignment spikes, from [76].

- The decay of losses can be noisy (as in graphs 1,2,4) or smooth (graph 3)

On the other hand, non-alignment spikes don't have a fixed structure and can contain spurious high spikes (as in graphs 1,2,3). These can arise due to various reasons, such as; beam instabilities, mechanical vibrations of the opposite jaw when close to the beam, or drifts of the beam position.

Taking a closer look at the alignment spikes (recall Figure 2.7b), a clear alignment spike consists of; a steady-state signal before the spike, the loss spike itself, temporal decay of losses, and a steady-state signal after the spike. The steady-state is a result of continuous scraping of halo particles when the jaw positions are fixed. The further a jaw cuts into the beam halo the more the steady-state signal increases as the density of the particles near the jaw increases. Other spikes which

do not follow this pattern are classified as non-alignment spikes.

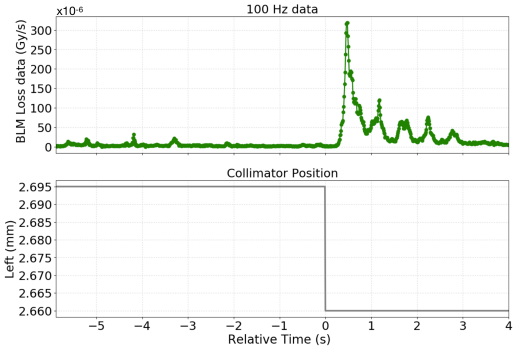
5.2 Dataset Construction

Data was gathered from five semi-automatic collimator alignments campaigns performed in 2016, both at injection (450 GeV) and at flat top (6.5 TeV). A total of 1652 samples were extracted, 467 positive (alignment spikes) and 1185 negative (non-alignment spikes). This data set was used to engineer useful features, as inputs to machine learning models for spike detection.

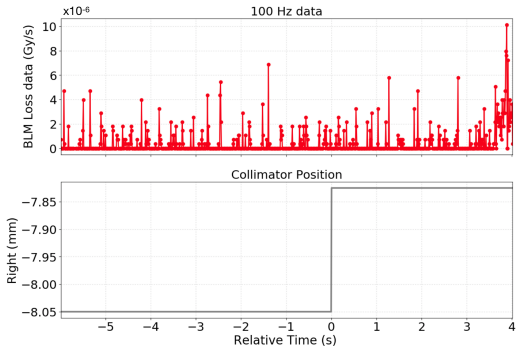
5.2.1 Data Acquisition

The data logged during alignment campaigns consists of the 100 Hz BLM signals and the collimator jaw positions logged at a frequency of 1 Hz. The BLM signal is analysed for alignment spikes the moment the collimator jaw(s) stop moving, when the losses exceed the predefined threshold. Therefore a data set is generated from these moments and the BLM signal in each data sample was individually analysed and labelled by experts into the two classification classes, as per the example shown in Figure 5.2. The data was extracted by first collecting all the jaw movements towards the beam that were closer than ± 10 mm from the beam axis. These movements were then divided into left and right jaw movements, and the times of these movements were extracted to obtain the corresponding BLM signals.

The time window to extract the BLM signal at each collimator movement starts and ends a maximum of 20 s before and after the current movement, as this contains all the required information. The time window was set to keep a 2 s gap between the previous and the next collimator movements, in order to ensure that there is no overlap between the decay of the first sample and the steady state of the next sample. If the next movement starts within 2 s after the current movement, or the BLM losses for the entire window are less than 1×10^{-6} Gy/s, then the current data sample is discarded.



(a) Positive data sample extracted when a left jaw movement towards the beam created an alignment spike.



(b) Negative data sample extracted during a right jaw movement towards the beam.

Figure 5.2: Two data samples extracted for spike classification, whereby the left and right jaw movements towards the beam generated an alignment and non-alignment spikes, respectively. The convention is to have the left jaw on the positive side of the beam axis, and the right jaw on the negative, from [76].

5.2.2 Feature Engineering

Fourteen features were initially extracted for spike detection from the alignment data set as shown in Table 5.1. The first five features were taken from [77], explained in Section 3.3.3. This feature set was augmented using the three coefficients from the Power function, whilst seven features were newly engineered for this study.

5.2.3 Feature Analysis

To select the most relevant features, the strength of association between each pair of variables was first analysed using the Spearman Correlation, Figure 5.3. This is a non-parametric test used to check for a statistically significant relationship without assuming the distribution of the data. This test is necessary to avoid selecting correlated features. The Maximum value, Height and Factor, were found to be correlated, whilst Exp_b and Pow_b are not correlated to any other features except with each other.

A feature selection analysis was then performed using five different machine learning models to see how they order the importance of each of the features. The

Table 5.1: A list of the fourteen features extracted from the data set, from [76].

Name	**Description**	**Numerical Range**
Maximum value	The highest point taken to be within a two second range after the collimator stopped moving, from [77].	0.7 : 26000
Variance	The width of the Gaussian fit, whereby an optimal spike would have a smaller width as this would reflect a sharp increase and a quick decrease of the losses, from [77].	0.002 : 0.133
Gaussian Correlation Coefficient	The proximity of the loss pattern to the Gaussian fit using least squares, such that the closer this value is to unity, the sharper the loss spike from [77].	0 : 1
Power function	*Function to fit the decay using* $\mathbf{a}x^{\mathbf{b}} + \mathbf{c}$	-
a) Gradient (*Pow_a*)	This is proportional to the decay steepness.	-15000 : 80
b) Power decay (*Pow_b*)	This is the rate of decay, from [77].	0 : 200
c) Horizontal Asymptote (*Pow_c*)	This is proportionate to the new steady state.	-0.033 : 15000
Power Correlation Coefficient	The temporal decay becomes smoother as this value approaches unity, and this value indicates how well the Power fit represents the loss pattern using least squares, from [77].	0 : 1
Height	This is calculated by subtracting the average steady state from the Maximum value. The average steady state is calculated from the BLM signal after the decay of the previous alignment, until the current collimator was stopped.	0.456 : 26000
Position in sigma (*Pos*)	A beam size invariant way of expressing the fraction of the normally distributed beam interrupted by the jaw, as the beam size in mm varies across locations in the accelerator.	0 : 80

Name	**Description**	**Numerical Range**
Factor	This is calculated by taking the Maximum value as a fraction of the average steady state. The average steady state is calculated from the BLM signal after the decay of the previous alignment, until the current collimator was stopped.	1.46 : 1130
Exponential function	*Function to fit the decay using* $\mathbf{a}e^{-\mathbf{b}x} + \mathbf{c}$	-
a) y-Intercept (*Exp_a*)	This is proportional to the height of the spike.	-5800 : 15600
b) Gradient (*Exp_b*)	This is the rate of decay.	-20 : 2200
c) Horizontal Asymptote (*Exp_c*)	This is proportionate to the new steady state.	-3500 : 6000
Exponential correlation coefficient	Indicates how well the exponential fit represents the decay using least squares.	0 : 1

models used, specifically designed for feature selection are; Logistic Regression, Linear Support Vector Machine, Extra Trees Classifier [78], Random Forest Classifier and Gradient Boosting Classifier. The models were individually trained using all features and outputted the features ranked in ascending order, according to their importance. The outputs from the models are stacked in Figure 5.4, which displays the overall rank of all the features in ascending order.

The Power, Gaussian and Exponential Correlation Coefficients are used in this section to measure the reliability of the functions' fit to the data. The results in Figure 5.4 clearly indicate that the exponential fit to the decay is the first most important feature. Moreover, the Gaussian and Power functions did not manage to fit the decay of all the data samples, as the least-squares minimization used when fitting the functions, failed. From these observations it can be concluded that the exponential function is a better and more reliable fit to the decay.

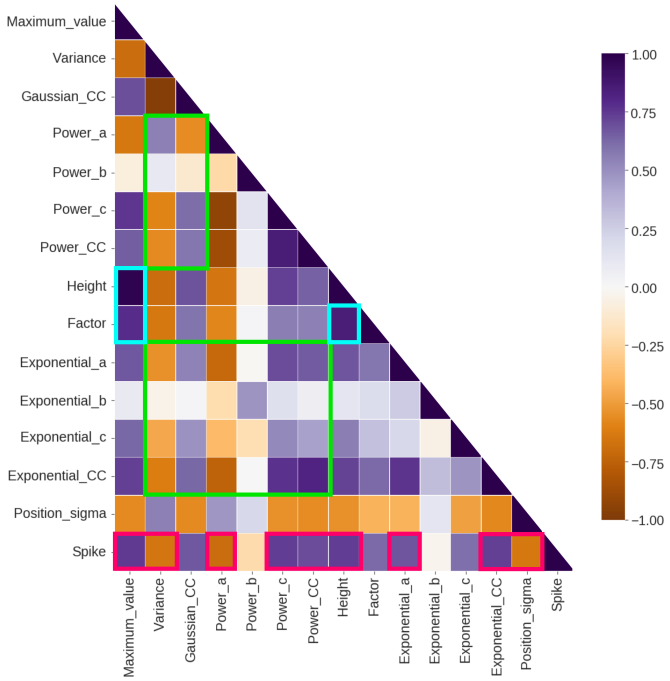


Figure 5.3: Heatmap of feature pairwise Spearman correlation indicating; Maximum value, Height and Factor are correlated (blue highlight), the decay fits do not seem to be fully correlated (green highlight) and most features have some correlation with the spike class (pink highlight), from [76].

5.3 Model Training

This section focuses on training, optimising and comparing the performance of six different classifiers using the engineered features. The exponential function has already been selected as the best function for fitting the decay, therefore after removing the Power and Gaussian functions' features, the nine remaining features are further analyzed in this section, whilst optimising the machine learning models accordingly. The models used in this section are; Logistic Regression, Neural Network, Support Vector Machine, Decision Tree, Random Forest and Gradient Boost. The scikit-learn [79] implementations were used.

When aligning any collimator it is vital that after the spike detection declares a collimator to be aligned, then the collimator is actually aligned. As a result, false detection of an alignment spike is more grievous than not detecting an alignment spike, therefore precision, i.e. the ratio of correct spike classifications to the

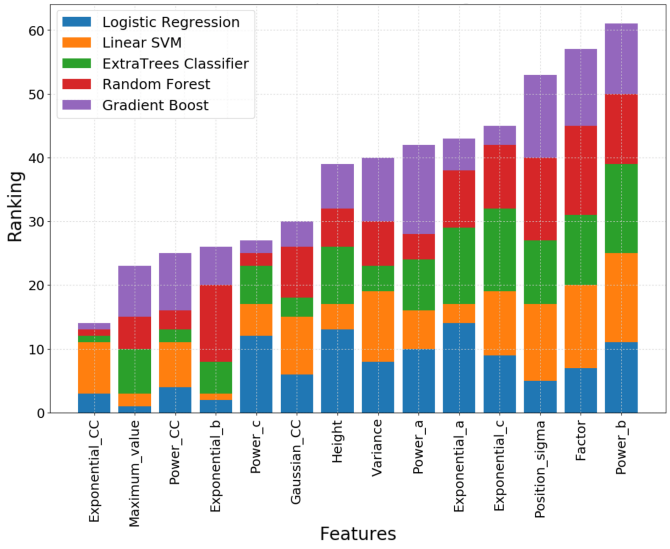


Figure 5.4: The list of features ranked in overall importance according to the individual machine learning models, from [76].

total number of spike classifications, will be used as the main performance metric throughout this section.

5.3.1 Hyper-Parameter Optimization

The hyper-parameters were optimised for each model. An exhaustive grid search algorithm, guided by precision as a performance metric, was applied on the training set using a 10-fold cross-validation randomly stratified 30 times, to handle lucky splits.

The data set was split using stratified sampling into an 85% training set and a 15% held-out testing set. The following parameters were tested for the corresponding models and the best parameters (in bold) were selected according to the highest precision obtained on the held-out testing set.

- Logistic Regression:
  - Regularization - 1×10^{-4}, $\mathbf{1 \times 10^{-3}}$, 1×10^{-2}, 1×10^{-1}, 1, 10, 100
- Neural Network:

  Default Architecture - 1 hidden layer with 100 units

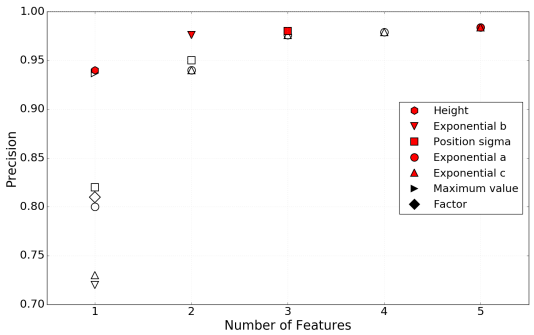
  - Hidden layer activation - **tanh**, relu
  - Weight optimization solver - **L-BFGS** [80], Stochastic gradient descent, Adam [81]
  - Regularization - 1×10^{-4}, 1×10^{-3}, 1×10^{-2}, $\mathbf{1 \times 10^{-1}}$, 1, 10

- Support Vector Machine:
  - **Linear** kernel - Penalty - 1×10^{-4}, $\mathbf{1 \times 10^{-3}}$, 1×10^{-2}, 1×10^{-1}
  - RBF kernel - Gamma - 1×10^{-4}, 1×10^{-3}, 1×10^{-2}, 1×10^{-1}
    - Penalty - 1×10^{-3}, 1×10^{-2}, 1×10^{-1}, 1, 10

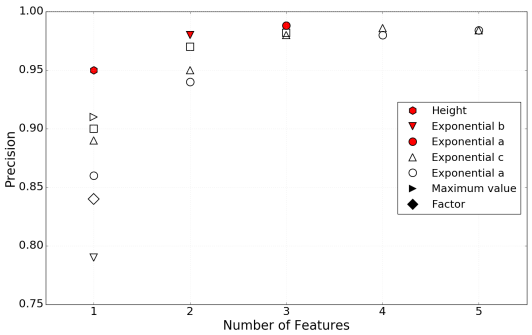
- Decision Tree:
  - Split criterion - Gini Impurity, **Entropy**
  - Splitter at each node - **best**, random
  - Maximum tree depth - **2**, 3
  - Minimum samples at leaves - **1**, 5, 10, 20, 40

- Random Forest:
  - Number of trees - 10, 50, **100**, 150
  - Split criterion - Gini Impurity, **Entropy**
  - Maximum tree depth - **2**, 3
  - Minimum samples at leaves - **1**, 5, 10, 20, 40

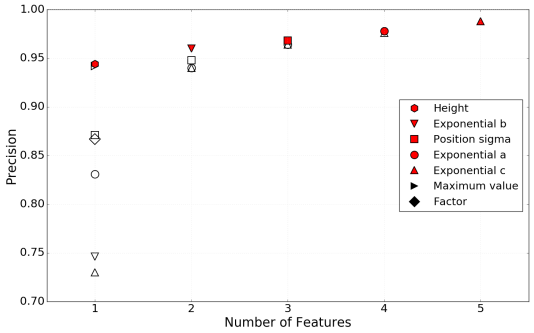
- Gradient Boost:
  - Learning rate - 1×10^{-4}, 1×10^{-3}, 1×10^{-2}, $\mathbf{1 \times 10^{-1}}$, 1, 10
  - Number of trees - 10, 50, 100, **150**, 200, 250
  - Maximum tree depth - 2, **3**
  - Minimum samples at leaves - **1**, 5, 10, 20, 40



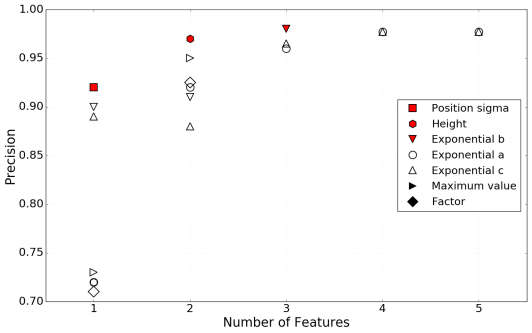
(a) Logistic Regression precision results.



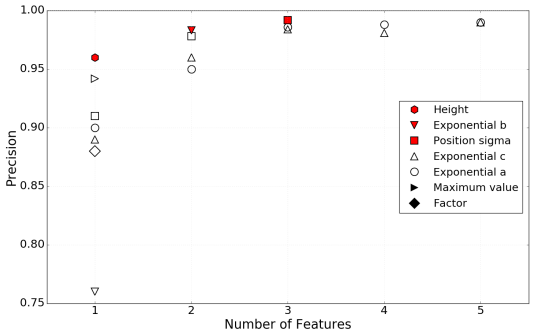
(b) Neural Network precision results.



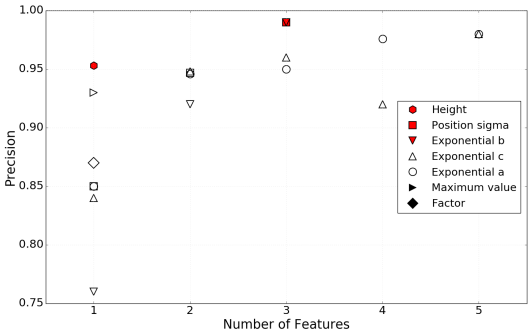
(c) SVM precision results.



(d) Decision Tree precision results.



(e) Random Forest precision results.



(f) Gradient Boost precision results.

Figure 5.5: The precision obtained by each of the models using the different feature combinations from the SFS with nested grid search. Each feature is indicated by a unique shape, such that the selected features are marked in red and the features ignored due to correlation are marked in black, from [76].

The grid search was nested within the sequential forward selection algorithm (SFS) [82], to select the best features with the best hyper-parameters. The SFS is a greedy algorithm that tries all feature combinations by introducing one feature at a time and keeping the best feature for future combinations. Moreover, three correlated features (Maximum value, Height and Factor) were identified from Section 5.2.3, therefore if a model selects any of these three, the SFS algorithm used will remove the other two features from future combinations, to avoid selecting correlated features. The precision results from the various feature combinations using the best performing hyper-parameters for each combination are plotted in Figure 5.5, and the selected best features are listed in Table 5.2.

Table 5.2: Ranking of best features selected by each model, from [76].

	Best feature ranking				
ML Model	**#1**	**#2**	**#3**	**#4**	**#5**
Logistic Regression	Height	Exp_b	Pos	Exp_a	Exp_c
Neural Network	Height	Exp_b	Exp_a		
SVM	Height	Exp_b	Pos	Exp_a	Exp_c
Decision Tree	Pos	Height	Exp_b		
Random Forest	Height	Exp_b	Pos		
Gradient Boost	Height	Pos	Exp_b		

A closer look at the training of each model using the selected features and hyper-parameters is shown in Figure 5.6, which shows the learning curves over various data set sizes using a 10-fold cross-validation randomly stratified 30 times. The results show that the Gradient Boost model is suffering from high variance on the training set, whilst the learning curves for the rest of the models on the two sets converge, indicating that the models are successfully learning. Overall the Logistic Regression model obtained the best results, followed by the Support Vector Machine.

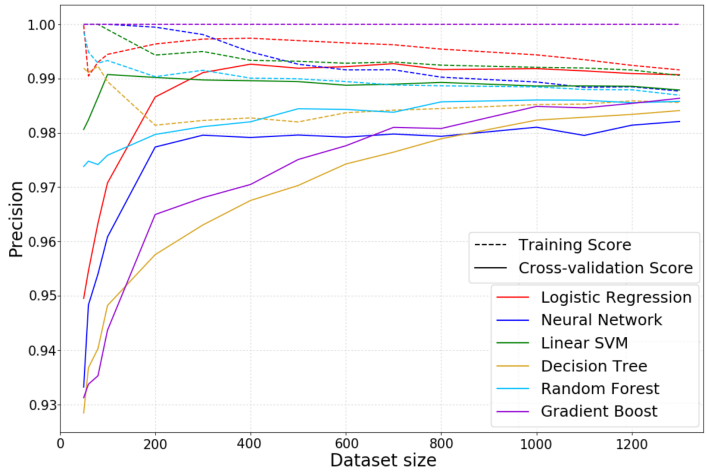


Figure 5.6: The learning curves based on the precision obtained by each model on the training and testing sets, using the best selected features and hyper-parameters, on various data set sizes, from [76].

5.3.2 Model Robustness

The six machine learning models were trained on 2016 data, achieving high precision. Figure 5.7 plots the precision results obtained by each of the models and their Ensemble.

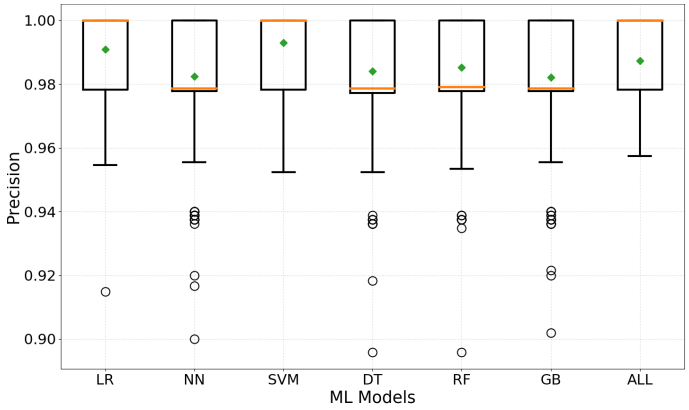


Figure 5.7: The precision obtained by each model and their Ensemble, using the 2016 data set, from [76].

Throughout this section the results are obtained using 10-fold cross-validation randomly sampled 30 times. Moreover, Tukey's HSD test [83] was used to determine whether the means of the results obtained by the models are significantly

different. This test is based on the Student t-test while adjusting the p-values for multiple comparison tests, therefore allowing to compare all possible pairs of means. Due to this, each box plot comparing means in this section has a corresponding table of the results obtained using post-hoc Tukey's test.

In the case of Figure 5.7, Tukey's test in Table 5.3 indicates that the mean precision obtained by the Support Vector Machine is not significantly different from that of the Logistic Regression, but significantly greater than the rest of the models. In addition, the mean of the Ensemble model is also similar to the Logistic Regression, whereas the other models obtained lower and similar results.

Table 5.3: 2016 data set: Tukey HSD Test - Models with the same letter are not significantly different, Alpha=0.05, from [76].

ML Model	**Precision**	**Groups**
SVM	0.9930386	a
LR	0.9909449	ab
ALL	0.9873208	bc
RF	0.9853248	cd
DT	0.9840476	cd
NN	0.9824100	d
GB	0.9821891	d

To test the robustness of the machine learning models, new data was gathered from six alignment campaigns performed in 2018, both at injection and flat top. A total of 4794 samples were extracted, 3912 positive (alignment spikes) and 882 negative (non-alignment spikes).

Each model was first tested on the entire 2018 data set after being trained on the 2016 data set, therefore cross-validation was not applied as the training and testing sets are fixed in this case. Table 5.4 lists the precision results obtained, indicating that each model obtained high precision, with the Neural Network obtaining the lowest precision at 0.944.

Figure 5.8 plots the precision results obtained using solely 2018 data for training and testing, and the Support Vector Machine obtained the best results, followed by the Ensemble and Neural Network (refer to Table 5.5). These results indicate

Table 5.4: The precision obtained by each model and their Ensemble, using the 2016 data set for training and the 2018 data set for testing, from [76].

ML Model	Precision
ALL	0.9913721
SVM	0.9905603
RF	0.9860101
DT	0.9813967
LR	0.9788605
GB	0.9683479
NN	0.9443495

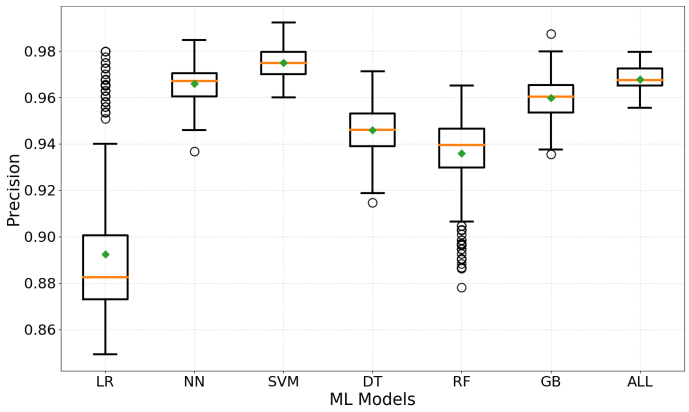


Figure 5.8: The precision obtained by each model and their Ensemble, using the 2018 data set, from [76].

Table 5.5: 2018 data set: Tukey HSD Test - Models with the same letter are not significantly different, Alpha=0.05, from [76].

ML Model	Precision	Groups
SVM	0.9749714	a
ALL	0.9678411	b
NN	0.9659314	b
GB	0.9598743	c
DT	0.9458548	d
RF	0.9358178	e
LR	0.8924207	f

that using the 2018 data set for training can further improve the predictions of the models. Due to this, the models were trained and tested on both data sets combined, and Figure 5.9 plots the precision results obtained. These results and

Table 5.6 indicate that the Support Vector Machine obtained the best precision, with a mean similar to the Ensemble, which has a more stable range of results. The Logistic Regression obtained the lowest precision results overall, with similar results to the Decision Tree.

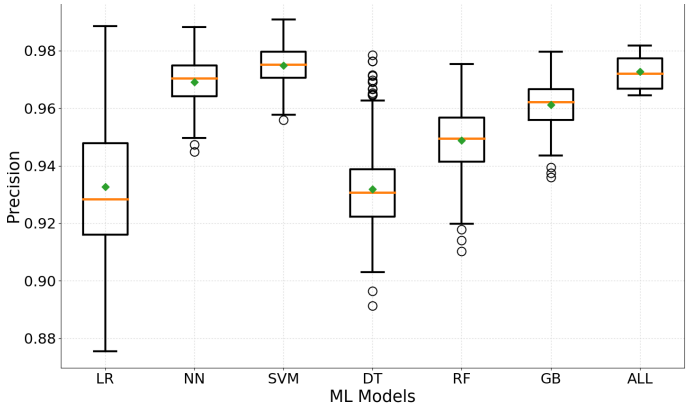


Figure 5.9: The precision obtained by each model and their Ensemble, using the combined 2016 and 2018 data sets, from [76].

Table 5.6: 2016+2018 data set: Tukey HSD Test - Models with the same letter are not significantly different, Alpha=0.05, from [76].

ML Model	Precision	Groups
SVM	0.9749064	a
ALL	0.9726233	a
NN	0.9690865	b
GB	0.9612389	c
RF	0.9488177	d
LR	0.9326096	e
DT	0.9319209	e

For the case combining 2016 and 2018 data, the bootstrap method [84] was used to estimate the performance of each of the machine learning models. This method is a re-sampling technique used to estimate summary statistics of the models when making predictions. In contrast to cross-validation, this method provides confidence intervals together with the precision of each machine learning model. Table 5.7 lists the 95% confidence intervals of the precision that can be obtained

by each model, indicating that the better models are the Support Vector Machine and Neural Network.

Table 5.7: 95% confidence interval on precision for each model, from [76].

ML Model	Precision Interval
Logistic Regression	89.0 - 98.0
Neural Network	97.0 - 98.0
SVM	97.0 - 98.0
Decision Tree	93.0 - 95.0
Random Forest	94.0 - 96.0
Gradient Boost	96.0 - 98.0

5.4 Model Validation in LHC Operation

The machine learning models have been successfully deployed in operation throughout 2018, with the aim of transforming the semi-automatic alignment into a fully-automatic one. At first, tests were carried out in the LHC at injection [85], which involved having a collimation expert align collimators using the semi-automatic procedure, whilst the Ensemble model was run in parallel as an independent component. The performance of the machine learning model was determined by comparing its classifications to those of the user as the collimators were being aligned in real-time.

A total of 52 alignments were performed, such that each collimator was first aligned using both jaws, then left, then right until a spike in each case was achieved. A full alignment example is that of the TCLA.D6L7.B2 shown in Figure 5.10. Moreover, a subset of the alignments performed are presented in more detail in Figure 5.11.

Overall, the machine learning model correctly classified 50/52 alignments, achieving an accuracy of 96% on automatic spike pattern recognition during alignment. This is higher than the previous work discussed in Section 3.3.3, which achieved a score of 82.4% on unseen data. Moreover, this study has focused on more recent

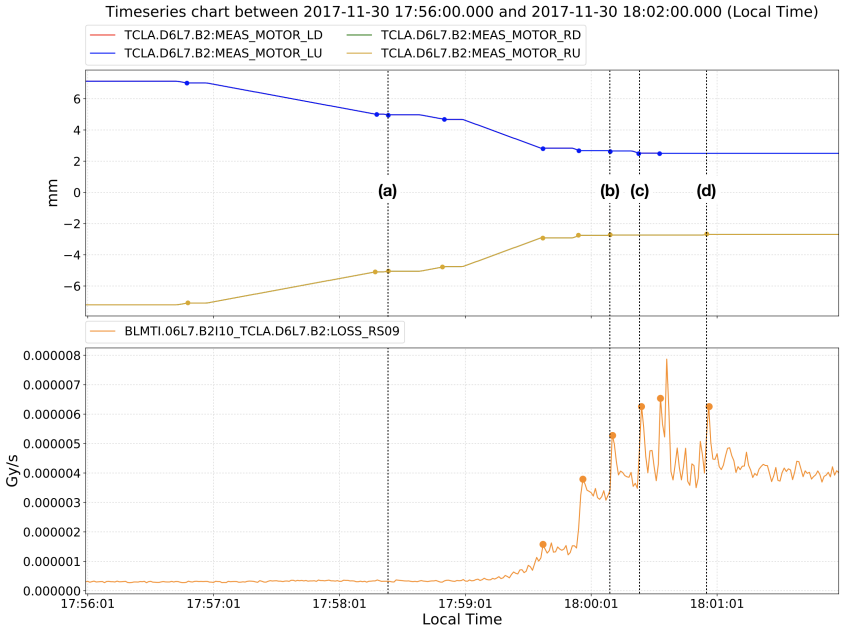
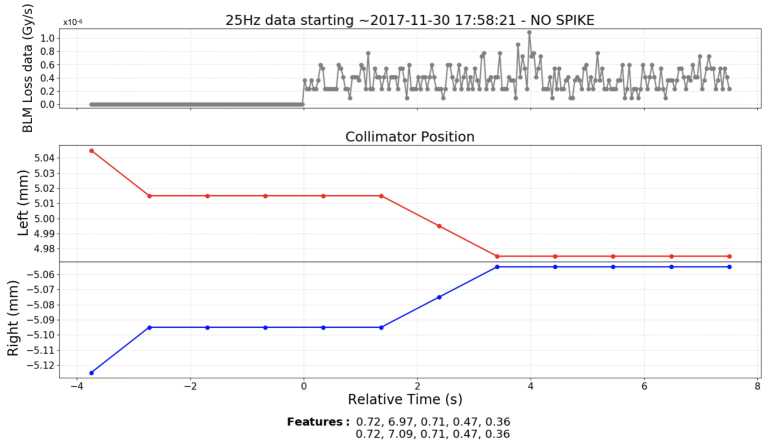
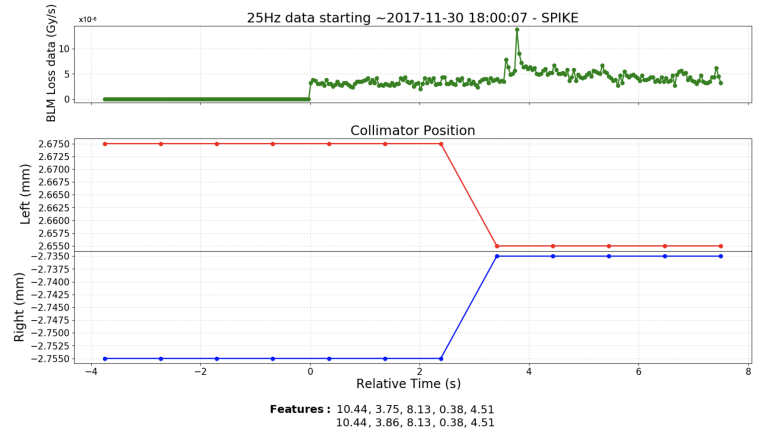


Figure 5.10: TCLA.D6L7.B2 full alignment.

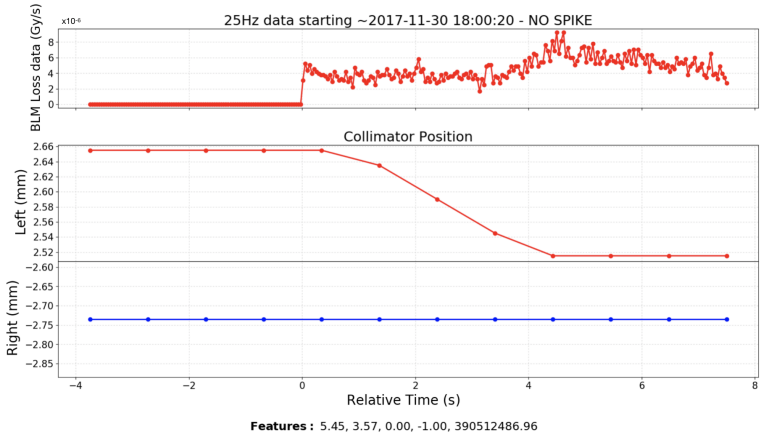
LHC data, at new higher energies at 6.5 TeV, thus making it suitable for present operational use. The two incorrectly classified spikes were false negatives therefore the only repercussion was a re-alignment, i.e. moving the jaw further in to obtain another spike. These tests confirm the reliability of introducing machine learning for aligning collimators, therefore this can be used for all future alignment campaigns.



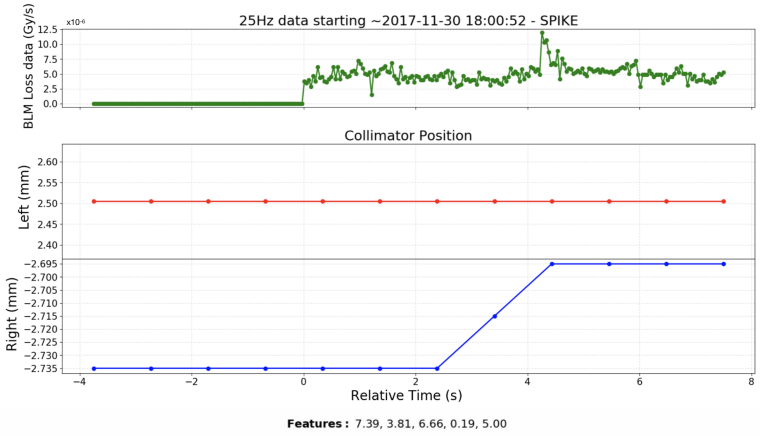
(a) No spike on both jaws.



(b) Spike achieved on both jaws.



(c) ML says no spike, users say spike.



(d) Spike achieved on right jaw.

Figure 5.11: A closer look at a subset of the alignments in Figure 5.10.

6. Threshold Algorithm for Automatic Selection

The semi-automatic BBA automatically moves a collimators toward the beam, until the losses recorded by its corresponding BLM device exceed a predefined threshold (S_i^{Thres}). The user application was designed to provide a fixed list of thresholds in Gy/s:

$$1 \times 10^{-6}, 2 \times 10^{-6}, 5 \times 10^{-6}, 8 \times 10^{-6}, 1 \times 10^{-5}, 2 \times 10^{-5},$$
$$4 \times 10^{-5}, 5 \times 10^{-5}, 6 \times 10^{-5}, 8 \times 10^{-5}, 1 \times 10^{-4}, 2 \times 10^{-4}$$

These thresholds were developed over time following experience from manual alignments. They are available for the collimation expert to select the one which corresponds best to the current BLM signal. The ideal threshold must be:

- High enough to ignore any noisy spikes and touch the beam without interrupting the movement.
- Low enough to immediately stop the jaws and generate minimal losses when the collimator actually touches the beam.

Fully-automating the BBA, requires automatically selecting the BLM threshold, by finding a compromise between the two requirements.

6.1 Implementation

The automatic threshold selection algorithm must attempt to mimic the BLM signal analysis made by the collimation expert when selecting the threshold, namely:

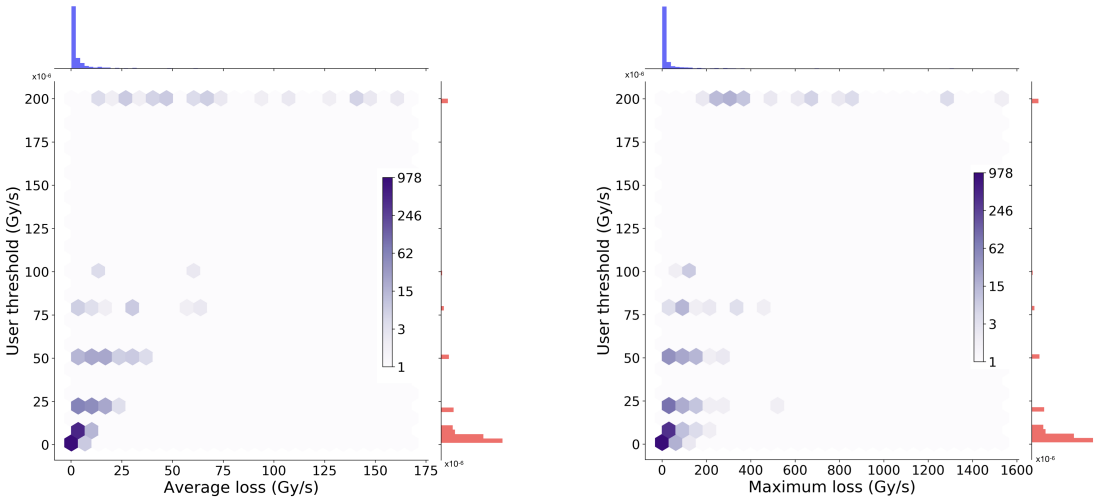
- Give higher importance to the most recent BLM values
- Partially factor out high spikes as the threshold does not need to be as high
- Assume long decays will eventually reach the steady state level
- Introduce a gap above the steady state high enough to allow for alignment spikes

6.1.1 Data Analysis

Alignments in 2016 were performed using the semi-automatic alignment tool requiring experts to manually select the thresholds at the start of each jaw movement towards the beam. This provided a data set of 1778 samples at injection and flat top, which were studied to find an automatic technique for selecting a reasonable threshold. The samples consist of 7.5 second windows of BLM signals, as the maximum decay time is 6 seconds at flat top, as explained in Section 4.2, and each sample was studied at 25 Hz, as this is what is available at the time of threshold selection. In addition, the thresholds selected by the user at the time were also extracted.

To determine whether there is a pattern between the BLM losses and the user selected thresholds, the thresholds are analysed against the maximum and average losses in Figure 6.1. The majority of the samples required thresholds below 2.5×10^{-5} Gy/s, and as expected, the larger thresholds are selected both when the average loss is larger as well as the maximum loss. Therefore RMS-smoothing will be applied to examine both and to be immune to outliers, due to it being a verified approach in time-series data (recall Section 3.3.2). In order to assign

different priorities to the data depending on their occurrence in time, exponential weights can be combined with the RMS by applying EWMRMS.



(a) User selected threshold against the average BLM losses.

(b) User selected threshold against the maximum BLM losses.

Figure 6.1: The threshold selected by the user against the (a) average BLM loss, (b) maximum BLM loss, 7.5 seconds before the user selected the threshold. Marginal histograms are included to show the distribution of each measure, from [64].

6.1.2 Automatic Threshold Selection Algorithm

The threshold selection algorithm was implemented in FESA and is shown in Algorithm 1. This algorithm is provided with the latest BLM data of length 188 (7.5 s at 25 Hz) to apply an EWMRMS on windows of fixed length (*length = 50*). This is used to automatically select a reasonable threshold (*auto threshold*). The first step is to create an array of length 188, of evenly spaced numbers between -1 and 0 and use them as power coefficients to the exponential function (*weights[size=188]*). A window for each BLM data element i is taken with the element itself and the 49 elements (or less) which precede it (*window[i-50, i]*). The RMS is then applied to the window and is multiplied by the exponential factor corresponding to the data

element (*weights[i]*):

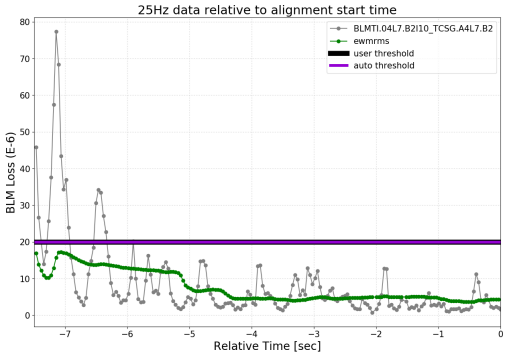
$$\text{EWMRMS} \quad = \quad \text{weights[i]} \sqrt{\frac{\sum_{\text{j=1}}^{\text{length}} \text{window[j]}^2}{|\text{window}|}} \tag{6.1}$$

Once the EWMRMS is calculated for all data elements, the maximum EWMRMS value is used to select the first threshold higher. If the threshold selected is also the first threshold above the mean value of the data, then the next higher threshold is selected. Finally, if the selected threshold is less than or equal to the threshold selected for the preceding alignment of the same collimator, and if that alignment was classified as no alignment spike, then the next higher threshold is selected.

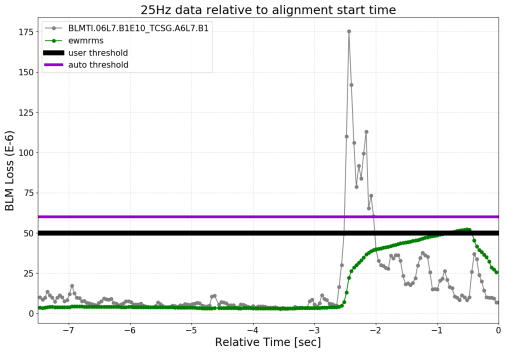
```
int window_size = 50;
double T[] = threshold options;
double weights[] = power exponentials;

//INPUT: previous_threshold (T[prev_t])
mean = data_sum / data_length;
for i = 0; i <data_length; i++ do
    start_position = i - window_size;
    if start_position <0 then
        start_position = 0;
    end
    window = data[start_position, i];
    ewmrms = Calculate EWMRMS using Equation 6.1;
end
auto_threshold = T[t] >max(ewmrms);
if auto_threshold >mean and T[t-1] <mean then
    auto_threshold = T[t+1];
end
if t <= prev_t then
    auto_threshold = T[prev_t+1];
end
```

Algorithm 1: Automatic threshold selection algorithm.

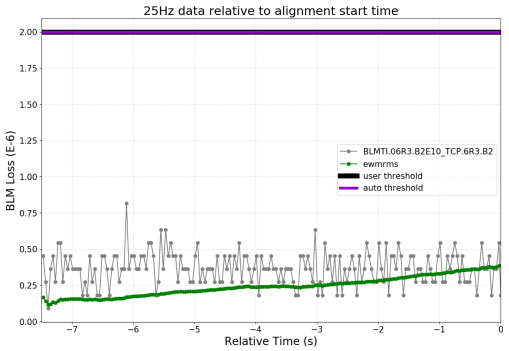
6.2 Results Analysis



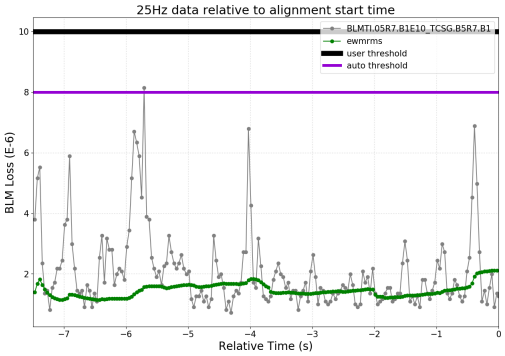
(a) BLM signal starting with a spike, from [64].



(b) BLM signal ending with a spike, from [64].



(c) BLM signal stable $\sim 4 \times 10^{-7}$ Gy/s.



(d) BLM signal with noisy spikes.

Figure 6.2: Four examples of 25 Hz BLM signals with the thresholds selected by the operator and algorithm.

The implemented algorithm was tested on the data set gathered. Figure 6.2 shows four BLM signals and the thresholds selected by the user and the algorithm. Each case is analysed below:

(a) The first example displays an alignment spike, 7 seconds before the start of the alignment, and one can see that the spike completely decayed and the signal reached steady state towards the end of the signal. Therefore in this case the spike is not important and one would simply aim for a suitable gap between the steady state and the threshold. This is the approach taken by both the user and the algorithm, such that the same threshold was selected.

(b) The signal in this example includes a clear spike towards the end of the signal. In this case the spike must be considered and the steady state before the spike is ignored as now (and possibly in the future) the steady-state losses are higher. Therefore the aim would be to create a suitable gap between the new steady state and the threshold, and partially factoring in the spike. Both the user and the algorithm selected suitable thresholds, such that they are high enough to allow for clear spikes, whilst at the same time conservative of future losses.

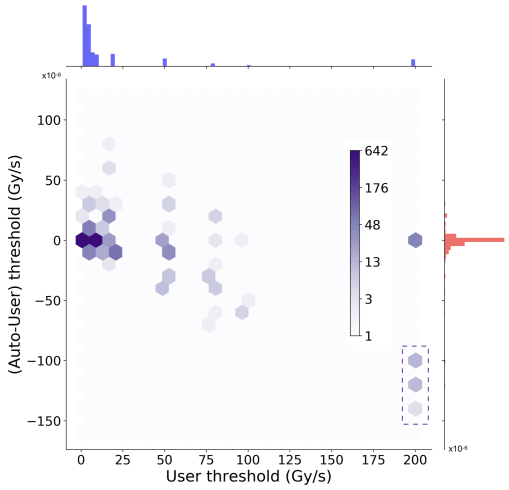
(c) This signal is stable with a steady state around 4×10^{-7} Gy/s, therefore the aim is to select a threshold creating a gap high enough to ignore any noisy spikes to avoid unnecessarily stopping the alignment. The algorithm selected the same threshold as the operator and this is a clear case where the algorithm adjusts the threshold selection according to the mean of the signal.

(d) The last example shows a noisy signal, in this case the aim is to create a gap above the steady, whilst factoring in the noisy spikes to raise the threshold slightly, in order to avoid stopping at each spike. The user and the algorithm both selected a threshold above the noisy spikes with the user's threshold selected slightly higher than that of the algorithm. Finally, this is another example where the algorithm adjusted the threshold selection using the mean of the signal.

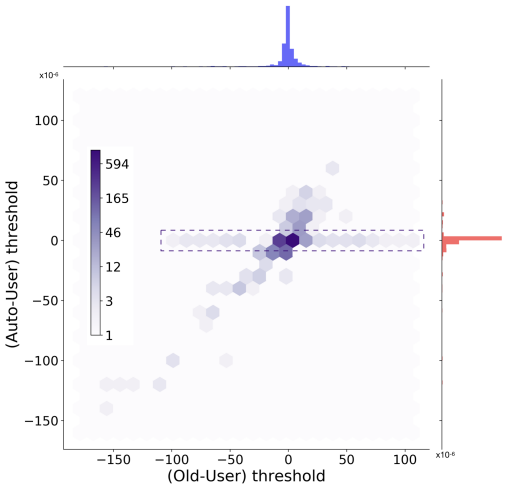
The overall performance of the algorithm is determined by comparing the thresholds selected automatically to the ones selected by the user. Figure 6.3a shows the difference between the two thresholds against the thresholds selected by the user. One can see that the difference is negligible for 90% of the cases, and the large differences occur when dealing with larger thresholds at 2×10^{-4} Gy/s, thus also making them suitable selections.

Finally, Figure 6.3b compares the thresholds selected by the new algorithm to those selected by the old algorithm previously described in Section 3.4.2, by comparing them both to the user selected thresholds. These results indicate that

the thresholds selected by the new algorithm corresponded better to those selected by the user. These promising results endorse using this algorithm, together with the automatic spike detection algorithm developed in Chapter 5, to fully-automate the alignment.



(a) The difference between the thresholds selected by the new automatic algorithm and the user, against the thresholds selected by the user. The dotted rectangle highlights that the larger differences occur only when a high threshold is necessary.



(b) The new automatic algorithm selections against the old algorithm selections, both compared to the user selections. The dotted rectangle highlights that the new algorithm corresponds better to the user.

Figure 6.3: The performance of the new automatic algorithm compared to (a) the user threshold, (b) the old automatic algorithm, including marginal histograms to show the distribution of each measure, from [64].

7. Cross-talk Analysis for Parallel Selection

As described in Section 2.6, when a particle reaches the primary collimator, it can either have an inelastic interaction, thus starting a hadronic and electro-magnetic shower that can be detected by several BLM devices, or interact elastically, thus changing its momentum and continue its trajectory. In both cases, additional signals, referred to as cross-talk, could be measured by BLM devices placed downstream from the initial interactions.

7.1 Cross-talk Analysis across Beams

Before 2018, the semi-automatic alignment software allowed for collimation experts to align beam 1 and beam 2 in parallel by having them manually handle the cross-talk. This meant that if a cross-talk signal was observed in one beam, the collimation expert would pause the alignment until the cross-talk signal disappears. However, with a fully-automatic alignment the collimation experts would no longer be able to intervene in case of cross-talk.

Due to this, the cross-talk observed between collimators must be analysed to automate the process of selecting collimators from the two beams to align in paral-

lel. An analysis was performed empirically to evaluate the effect of the propagation of electromagnetic showers after a collimator jaw touches the beam, on the BLM signals in the other beam.

7.1.1 Measurements from Beam Tests

For each collimator, the last two alignment steps for each jaw (left and right) were used to study cross-talk. These last steps are known to have touched the beam (recall Section 3.2) as shown in the alignment example in Figure 7.1. In these moments the collimator under study is the primary bottleneck in the ring and any other losses in other BLM devices are cross-talk signals.

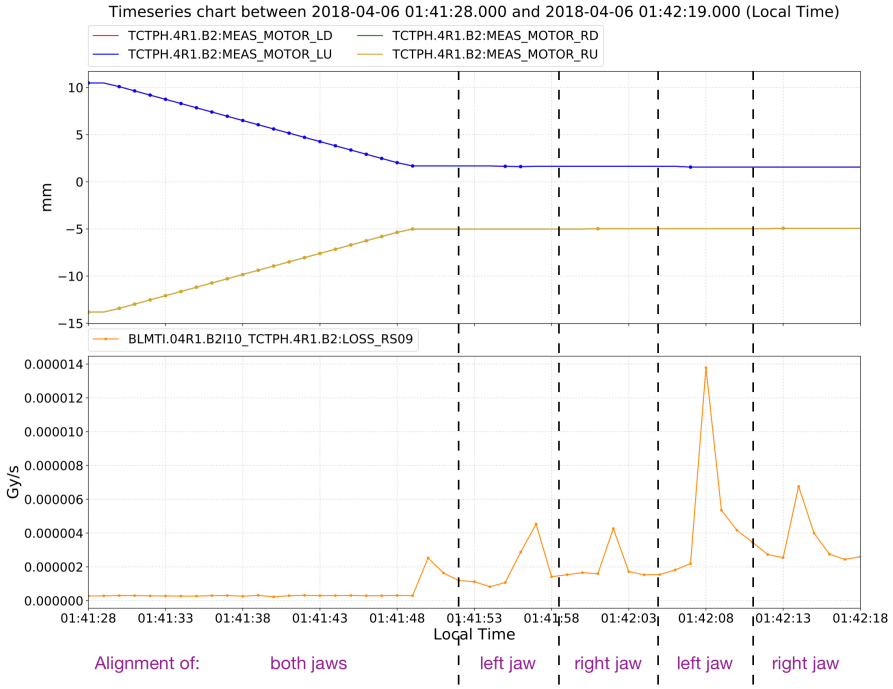


Figure 7.1: Alignment of the TCTPH.4R1.B2 during 2018 commissioning at injection, showing the BLM losses for the alignment of both jaws and the alignment spikes generated when the left and right jaws touched the beam twice.

A data set of 650 samples was gathered from alignment campaigns performed during commissioning 2018, as all alignments were done sequentially. Each data sample contains the BLM signal of the aligned collimator and the signal of all other

collimator BLM devices in the other beam, that have a signal greater than 5×10^{-6} Gy/s. The time-window around the maximum value of the aligned collimator was extracted from each of these signals, and was then smoothed using RMS:

$$\mathrm{RMS} = \sqrt{\frac{\mathbf{s}^2}{|\mathbf{s}|}}, \tag{7.1}$$

where $\boldsymbol{s}$ is a cascading subset of the time-window used. Each resulting RMS signal was used, such that if the mean of any of the four signals of that particular collimator was larger than 5% of the maximum value of the aligned collimator, then the signal was labelled as having experienced cross-talk. This process is displayed for the TCTPH.4R1.B2 in Figure 7.2.

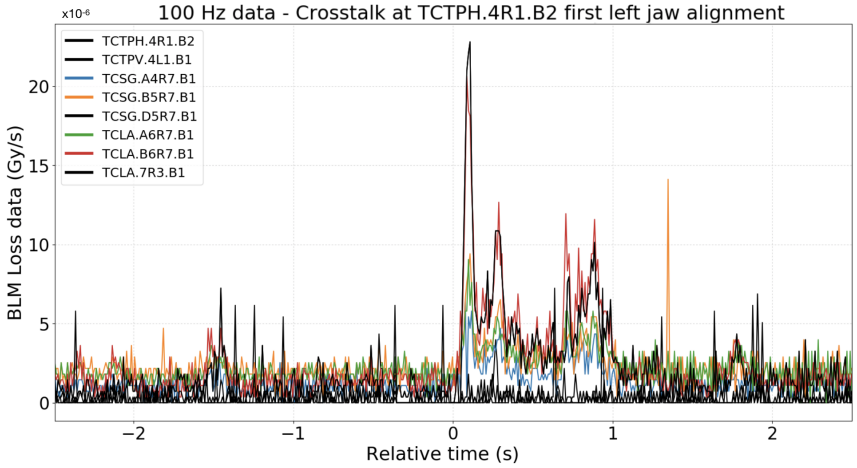
7.1.2 Results

To visualise the overall cross-talk experienced, the coefficients for all collimators marked as having experienced cross-talk were calculated using:

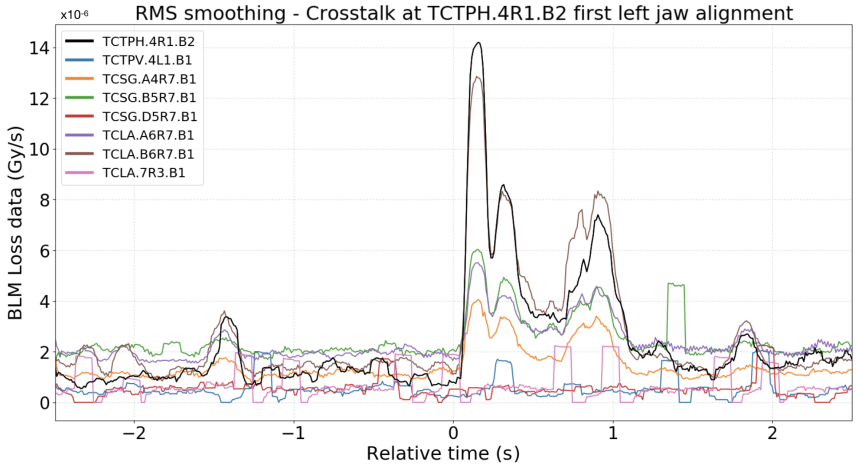
$$\frac{1}{\mathrm{N}} \sum_{\mathrm{n}=1}^{\mathrm{N}} \frac{\mathrm{BLM}_{\mathrm{n}}^{\mathrm{i}}}{\mathrm{BLM}_{\mathrm{n}}^{\mathrm{A}}}, \quad \mathrm{N} \leq 4 \tag{7.2}$$

where $\mathrm{BLM}^{\mathrm{A}}$ is the BLM signal of the collimator which was aligned, $\mathrm{BLM}^{\mathrm{i}}$ is the BLM signal of a collimator labelled as having experienced cross-talk, and N is the number of times collimator i was labelled as having experienced cross-talk. The coefficients obtained are displayed in the form of heat maps in Figure 7.3, which lists the collimators in the order they are physically positioned in the LHC.

The heat maps indicate that in both beams, the TCLA.A6 and TCLA.B6 in IR7 experience the highest amount of cross-talk. This cross-talk comes from showers generated by the primary collimators of the other beam as they generate the most showers picked up by downstream BLM devices. The reason for this cross-talk is that the primary collimators of one beam are positioned in front of the TCLAs of the other beam (recall Figure 2.3).



(a) The BLM losses of: the TCTPH.4R1.B2 at the time of alignment, the 4 collimators marked as having cross-talk, and the 3 collimators marked as having no cross-talk (in black). These were automatically determined using RMS smoothing in Figure 7.2b.

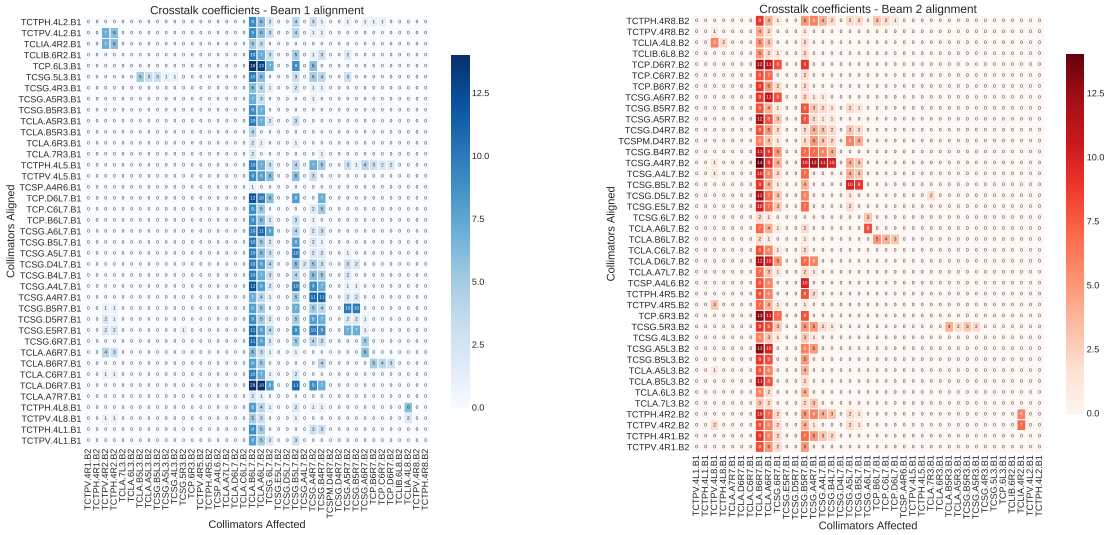


(b) RMS smoothing applied to the 100 Hz data shows a more clear distinction between the collimators actually effected by cross-talk and the collimators which are not.

Figure 7.2: The 100 Hz data of the first left jaw alignment of the TCTPH.4R1.B2 shown in Figure 7.1, was (a) extracted and (b) smoothed using RMS. The plots also include the losses of all collimators in the other beam (beam 1) which have losses greater than 5×10^{-6} Gy/s.

7.2 Cross-talk Analysis per Collimator

It would be beneficial to quantify the cross-talk experienced at a selected BLM device, in cases such as collimator alignments or predicting the main plane of losses. This motivated a more in-depth analysis of the cross-talk across collimators, to seek



(a) Effects of beam 1 collimators on beam 2 collimators.

(b) Effects of beam 2 collimators on beam 1 collimators.

Figure 7.3: Heat maps showing the amount of cross-talk observed across collimators in one beam, when collimators in the other beam were aligned during commissioning 2018.

to quantify the level of cross-talk recorded in BLM signals. A machine development (MD) study was scheduled at injection, dedicated to perform beam tests, focusing solely on collimators in IR7 [86].

7.2.1 Measurements from Beam Tests

The two beams were used, such that each beam was injected with 25 pilot bunches, each having an average bunch intensity of 6×10^9 protons/bunch. The measurement procedure was started by positioning only one collimator close to the beam, while the rest of the hierarchy was retracted (individual cases). At this point a selected bunch was excited using the transverse damper [87], until it was fully scraped away at the selected collimator. This was repeated for each collimator listed in Table 7.1, which identifies each collimator by a label and lists the corresponding cleaning plane (vertical or horizontal). Each collimator was closed by moving it to its injection setting (refer to Table 7.2), and for cases when these settings were too far from the beam (e.g.: TCLAs), the collimators were manually moved closer

to the beam. This was followed by exciting the beam with various combinations of closed collimators positioned close to the beam (combined cases). For each combined case, the closed collimators followed the injection hierarchy. Table 7.3 lists the combined cases used, by combining the IDs from Table 7.1, eg.: B1V - 1+2 indicates the combined case of collimators B1V-1 and B1V-2 positioned close to the beam, as shown in Figure 7.4.

Table 7.1: Individual cases of collimators that were set as primary bottleneck, and cleaning plane, from [16].

ID	**Collimator**	**Cleaning Plane**	**Case**
B1V-1	TCP.D6L7.B1	Vertical	1
B1V-2	TCSG.D4L7.B1	Vertical	2
B1V-3	TCLA.A6R7.B1	Vertical	3
B1V-4	TCLA.C6R7.B1	Vertical	4
B1H-1	TCP.C6L7.B1	Horizontal	1
B1H-2	TCSG.B4L7.B1	Horizontal	2
B1H-3	TCSG.6R7.B1	Horizontal	3
B1H-4	TCLA.B6R7.B1	Horizontal	4
B1H-5	TCLA.D6R7.B1	Horizontal	5
B1H-6	TCLA.A7R7.B1	Horizontal	6
B2V-1	TCP.D6R7.B2	Vertical	1
B2V-2	TCSG.D4R7.B2	Vertical	2
B2V-3	TCSPM.D4R7.B2	Vertical	3
B2V-4	TCLA.A6L7.B2	Vertical	4
B2V-5	TCLA.C6L7.B2	Vertical	5
B2H-1	TCP.C6R7.B2	Horizontal	1
B2H-2	TCSG.B4R7.B2	Horizontal	2
B2H-3	TCSG.6L7.B2	Horizontal	3
B2H-4	TCLA.B6L7.B2	Horizontal	4
B2H-5	TCLA.D6L7.B2	Horizontal	5
B2H-6	TCLA.A7L7.B2	Horizontal	6

7.2.2 Mathematical Formulation

The signal recorded by the BLM at collimator j (S_j) is composed of the beam losses generated by collimator j and upstream collimators (the latter resulting in cross-talk). Therefore, it is necessary to quantify the level of cross-talk present in S_j.

Table 7.2: List of 2018 injection settings for collimation, from [16].

Collimator	**IR**	**Orientation**	**Setting (σ)**
Primary (TCP)	7	H/V/S	5.7
Secondary (TCSG)	7	H/V/S	6.7
Absorber (TCLA)	7	H/V	10.0
Primary (TCP)	3	H	8.0
Secondary (TCSG)	3	H/V	9.3
Absorber (TCLA)	3	H	12.0
Secondary (TCSP)	6	H	7.5
Dump prot. (TCDQ)	6	H	8.0
Tertiary (TCT)	1/2/5/8	H/V	13.0

Table 7.3: Combined cases of collimators that were set as primary bottleneck with a hierarchy, and cleaning plane, from [16].

ID	**Plane**	**Case**
B1V - 1+2	Vertical	1+2
B1V - 1+2+3	Vertical	1+2+3
B1V - 1+2+3+4	Vertical	1+2+3+4
B1H - 1+2	Horizontal	1+2
B1H - 1+2+3	Horizontal	1+2+3
B1H - 1+2+3+4	Horizontal	1+2+3+4
B1H - 1+2+3+4+5	Horizontal	1+2+3+4+5
B1H - 1+2+3+4+5+6	Horizontal	1+2+3+4+5+6
B2V - 1+2	Vertical	1+2
B2V - 1+2+3	Vertical	1+2+3
B2V - 1+2+3+4	Vertical	1+2+3+4
B2V - 1+2+3+4+5	Vertical	1+2+3+4+5
B2H - 1+2	Horizontal	1+2
B2H - 1+2+3	Horizontal	1+2+3
B2H - 1+2+3+4	Horizontal	1+2+3+4
B2H - 1+2+3+4+5	Horizontal	1+2+3+4+5
B2H - 1+2+3+4+5+6	Horizontal	1+2+3+4+5+6

This can be done using the BLM signal contribution of each individual upstream collimator *i*, without the contribution from other collimators, i.e. without cross-talk (S_i').

For *n* collimators (*1*, *2*, ... *n*), the signal recorded by the BLM at the most downstream collimator *n*, can be defined using a general recursive equation:

$$S_n = \sum_{i=1}^{n} a_{ni}.S_i', \quad a_{nn} = 1, \quad \text{if } i > n: \ a_{ni} = 0 \tag{7.3}$$

Where a_{ni} is the cross-talk factor quantifying the effect of a closed upstream collimator *i* on collimator *n*. For a set of *n* collimators, this system of equations can be represented by the triangular matrix:

$$\begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_n \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ a_{21} & 1 & 0 & \dots & 0 \\ a_{31} & a_{32} & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & 1 \end{bmatrix} \begin{bmatrix} S_1' \\ S_2' \\ S_3' \\ \vdots \\ S_n' \end{bmatrix} \tag{7.4}$$

To calculate the cross-talk factors, the individual cases with each collimator closed individually must first be analysed. As an example, two individual cases (Case 1 and Case 2 shown in Figure 7.4) are used to calculate the cross-talk factors on the next open downstream collimator (Coll 3), to be able to predict the signal measured by its BLM device:

$$S_3 = \mathbf{a_{31}}.S_1' + \mathbf{a_{32}}.S_2' + a_{33}.S_3' \tag{7.5}$$

For each individual case, the contributions from other collimators are zero as they are completely retracted. For Case 1, the signal without cross-talk is 0 ($S_j' = 0$)

for $j \neq 1$, therefore all the coefficients a_{j1} can be calculated as follows:

Case 1: $S'_2 = S'_3 = 0$

$$S_1 = a_{11}.S'_1, \quad a_{11} = \frac{S_1}{S_1} = 1 \tag{7.6.1}$$

$$S_2 = a_{21}.S'_1, \quad a_{21} = \frac{S_2}{S_1} \tag{7.6.2}$$

$$S_3 = a_{31}.S'_1, \quad \mathbf{a_{31}} = \frac{S_3}{S_1} \tag{7.6.3}$$

Similarly, for Case 2, the signal without cross-talk is 0 ($S'_j = 0$) for $j \neq 2$:

Case 2: $S'_1 = S'_3 = 0$

$$S_1 = a_{11}.S'_1 = 0 \tag{7.7.1}$$

$$S_2 = a_{22}.S'_2, \quad a_{22} = \frac{S_2}{S_2} = 1 \tag{7.7.2}$$

$$S_3 = a_{32}.S'_2, \quad \mathbf{a_{32}} = \frac{S_3}{S_2} \tag{7.7.3}$$

As a result, the cross-talk factors for Equation 7.5 are calculated using Equations 7.6.3 and 7.7.3. This procedure can be applied recursively such that Cases 1, 2 and 3 can be used to predict the signal at a fourth collimator and so on.

7.2.3 BLM Coefficient Calculation

Beam tests for all of the collimator configurations were performed to determine the cross-talk factors and therefore predict the losses recorded by the first downstream collimator BLM after the last closed collimator. As an example, Figure 7.5 presents the BLM signals recorded using three collimator configurations in the vertical plane of beam 1 (B1V-1, B1V-2 and B1V - 1+2). The cross-talk factors determined for TCP.D6L7.B1 and TCSG.D4L7.B1 are then used to predict the cross-talk observed at TCLA.A6R7.B1.

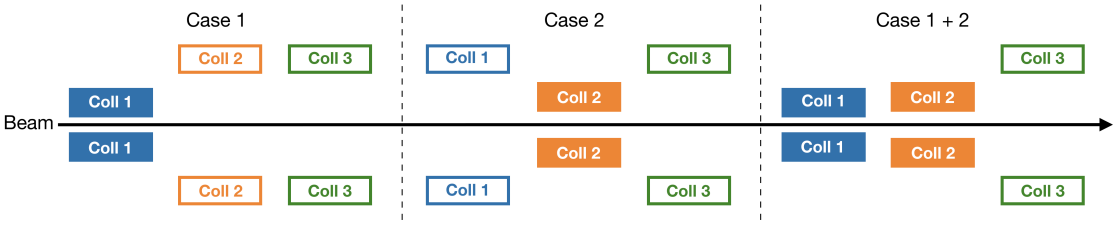


Figure 7.4: Simplified diagram of three collimators (Coll 1, Coll 2 and Coll 3), each one further downstream than the previous. Case 1 and Case 2 individually close Coll 1 and Coll 2, respectively, whilst Case 1+ 2 closes both collimators simultaneously, from [16].

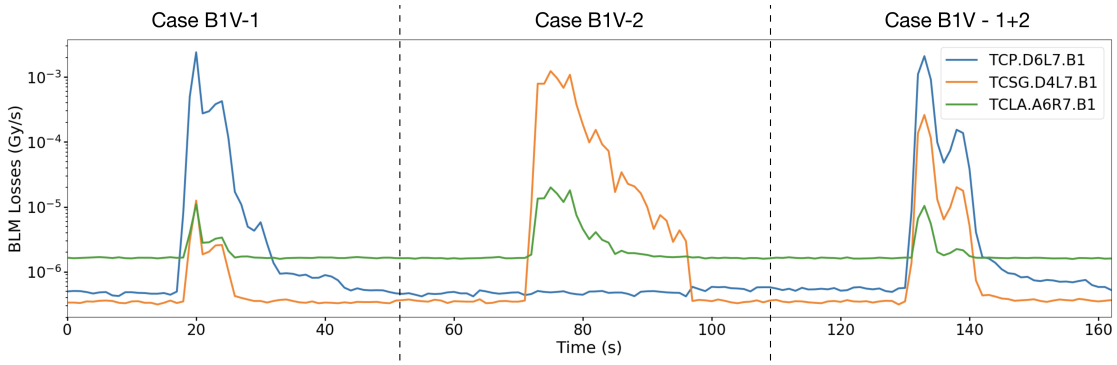


Figure 7.5: Series of beam loss signals recorded for a subset of collimators in the vertical plane in beam 1. Case B1V-1 and Case B1V-2 involve individually closing TCP.D6L7.B1 and TCSG.D4L7.B1, respectively, whilst Case B1V - 1+2 involves closing both collimators simultaneously, to ultimately predict the cross-talk experienced by the most downstream collimator, TCLA.A6R7.B1, from [16].

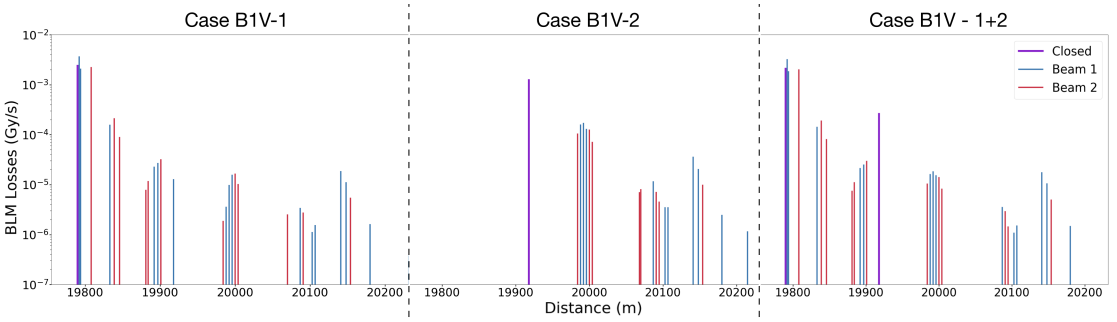


Figure 7.6: The spatial distribution of the losses measured in IR7 using the three configurations of closed collimators in Figure 7.5, from [16].

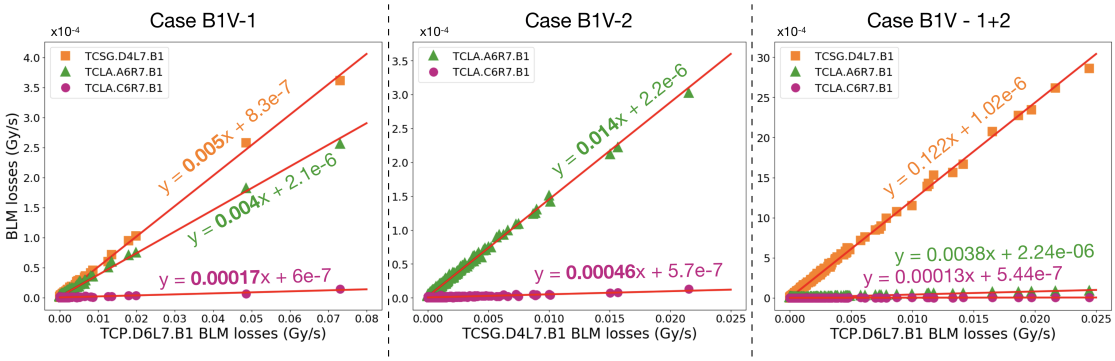


Figure 7.7: 100 Hz BLM signals recorded using the three configurations of closed collimators in Figure 7.5, against the 100 Hz BLM signal of the first upstream collimator closed within the configuration. Linear fits were applied to each set of data points, and the gradients obtained using the individual case configurations are marked in bold, from [16].

From this figure, one can qualitatively observe that:

1. Case B1V-1 shows that most of the particles are lost at TCP.D6L7.B1 when it acts as primary bottleneck. The other two monitored collimators were retracted, therefore the losses recorded by their BLM devices derive from the protons lost at the primary collimator.

2. Case B1V-2 shows the losses generated when TCSG.D4L7.B1 is moved in, therefore no losses are observed at TCP.D6L7.B1 (that is now retracted), but higher losses are recorded at TCLA.A6R7.B1.

3. Case B1V - 1+2 is a combined case where beam losses occur at TCP.D6L7.B1 and TCSG.D4L7.B1 with different ratios. The aim is to predict the signal and compare it with the measured at TCLA.A6R7.B1, for this configuration.

Figure 7.6 shows the loss maps generated using the same three configurations, focusing solely on IR7 collimators in both beam 1 and beam 2, to show that both beams can be effected by cross-talk. Finally, Figure 7.7 highlights the linear correlation between the collimators' 100 Hz BLM signals, such that the linear fits can be used to calculate the cross-talk factors. Moreover, this linearity between the

BLM signals verifies that the equations in Section 7.2.2 can be used. Therefore the BLM signals from all configurations are used in these equations to calculate the cross-talk factors, whereby each signal is integrated to eliminate any fluctuations between points. In both cases, the factors obtained are equivalent within a reasonable margin.

The resulting cross-talk factors for the vertical plane in beam 1 are displayed in Figure 7.8 and a complete table of all the cross-talk factors is presented in Appendix A. The cross-talk factors confirm that the cross-talk goes down the further downstream a collimator is located, with respect to the primary loss location.

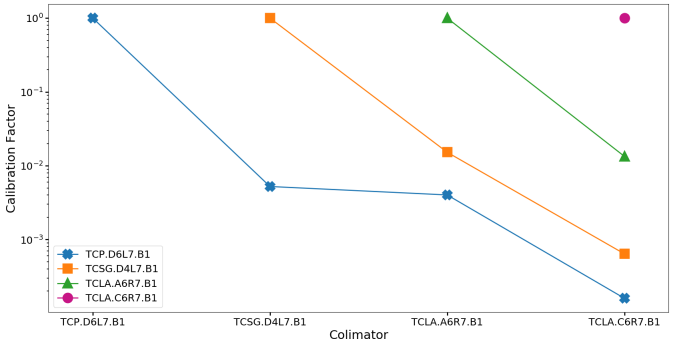


Figure 7.8: The cross-talk factors quantifying the effect of each collimator on the downstream collimators, for the set of collimators used in the vertical plane in beam 1 (Case B1V-1: 1, 0.005, 0.004, 0.00016, Case B1V-2: 1, 0.015, 0.0006, Case B1V-3: 1, 0.013, Case B1V-4: 1), from [16].

7.2.4 BLM Coefficient Validation

The cross-talk factors determined for the respective collimator configurations are validated by comparing the losses predicted using the cross-talk factors to measurements for the 13 downstream collimators used. Figure 7.9 qualitatively compares the measured and predicted BLM signals for four of these configurations, one from each plane.

The overall results are displayed in Figure 7.10 which shows all the measured signal points against their predictions. All points used are above 10^{-7} Gy/s, to

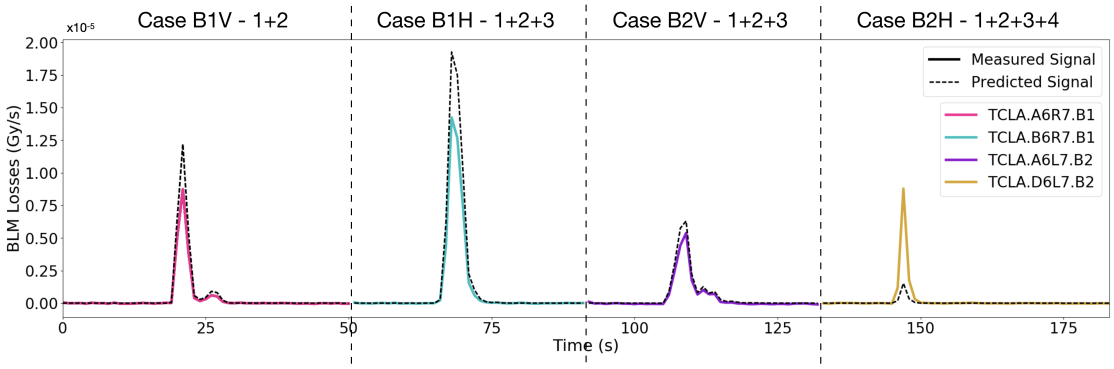


Figure 7.9: A comparison of the measured and predicted signals of four configurations of closed collimators, one from each plane; Case B1V - 1+2, Case B1H - 1+2+3, Case B2V - 1+2+3, Case B2H - 1+2+3+4, from [16].

exclude background noise. These results indicate a strong linear correlation between the two signals, especially as the signal increases. A linear fit was applied to the data with an R^2 of 98%, showing that on average the predicted signal is 11% higher than the measured one. This could possibly be used to decrease the calculation error depicted in Figure 7.11. The results are promising as the error decreases systematically towards 10% for higher measured signals, whereas a larger error is expected with lower measured signals.

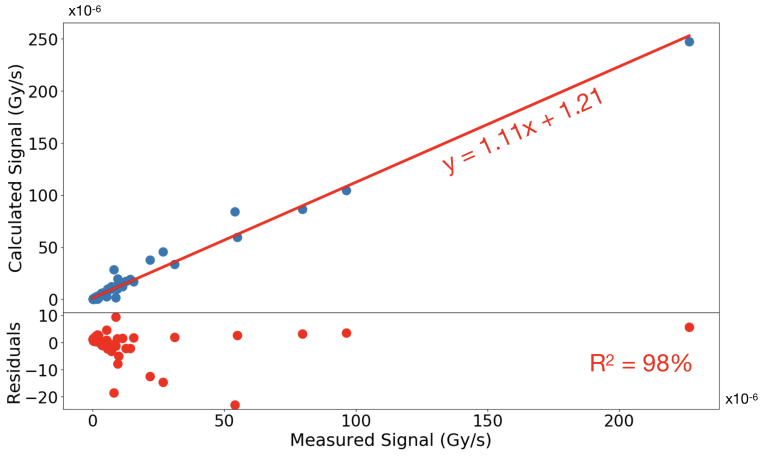


Figure 7.10: Scatter plot of the measured signal points above 10^{-7} Gy/s, against the calculated signal points. A linear fit was applied to the data and it's residuals were also plotted, from [16].

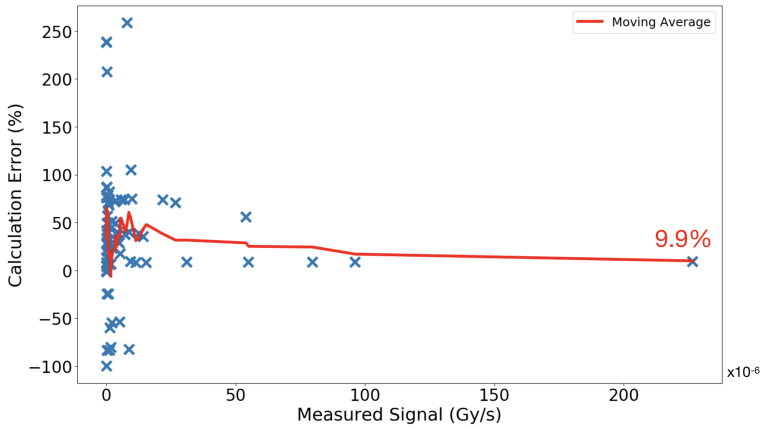


Figure 7.11: Scatter plot of the measured signal points above 10^{-7} Gy/s, against the calculation error. The moving average was calculated showing the average error is mostly below 50% for low BLM signals and as small as 10% for high BLM signals, from [16].

In the majority of the cases, the predicted signals are larger than the measured signals. This could be a result of upstream collimators shielding showers generated by impacts of particles on further upstream collimators, which are assumed to be zero in this paper. In addition, cases for which the predicted loss is smaller than the actual loss, corresponded to signals below 10^{-5} Gy/s which are more prone to errors. Therefore overall, the results clearly indicate that it is indeed possible to estimate the cross-talk that a collimator BLM will experience, within a reasonable margin.

7.2.5 Application to Loss Maps

This study seeks to predict the losses at the first open collimator positioned downstream after a series of closed collimators, therefore a case study for this analysis is an application to loss maps. During operation, two types of loss maps per beam are generated; horizontal and vertical loss maps, to examine the losses induced at the collimators by generating losses in the respective planes.

To apply this analysis to loss maps, the analysis can be used to predict the losses at the first skew collimator, as this is the plane in which no losses are generated, to

mimic the requirement of having an "open" collimator. This can be calculated by applying the cross-talk factors in Appendix A to the losses at the horizontal and vertical collimators positioned upstream. Since no measurements were performed in the skew plane (recall Section 7.2.1), no cross-talk factors were determined for that plane. According to the order the collimators are physically positioned around the beam, the collimators selected for this section are listed in Table 7.4.

Table 7.4: Collimator configurations used for skew collimator predictions in loss map case studies, from [16].

V Case	**H Case**	**Skew**
B1V - 1+2	B1H - 1+2	TCSG.A4L7.B1
B2V - 1+2+3	B2H - 1+2	TCSG.A4R7.B2

The data used in this section was gathered from three loss map campaigns during LHC operation. The campaigns were selected such that the order of planes in which the losses were generated vary, to ensure no bias in the results.

1. Technical Stop 2 in 2018 - After a technical stop, loss maps are used to validate the collimation hierarchy before resuming operation. In this case loss maps were generated at injection and flat top, with the horizontal plane performed first.

2. MD time - At the end of an MD at injection, time was dedicated to generate loss maps for this study. In this case the vertical plane was specifically performed first.

3. Dedicated MD - This was a dedicated test performed at flat top to generate losses by scraping the beam with the primary collimators. This scraping was performed to mimic the beam losses required for generating loss maps without using the standard hardware, to ensure that the order in which the losses are induced in the planes does not influence the results.

The individual configurations in Section 7.2.1 and the losses generated in the horizontal and vertical planes for the loss maps, were used to predict the losses for the two skew collimators in both planes.

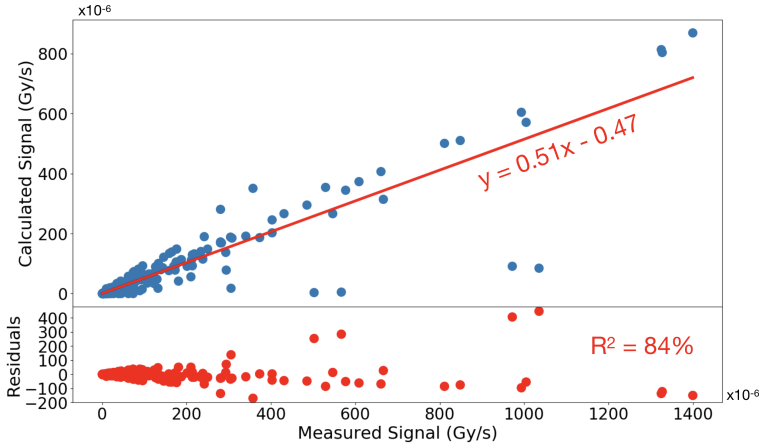


Figure 7.12: Scatter plot of the loss map measured signal points above 10^{-7} Gy/s, against the calculated signal points, for the two skew collimators. A linear fit was applied to the data, the fit residuals are presented in the lower plot, from [16].

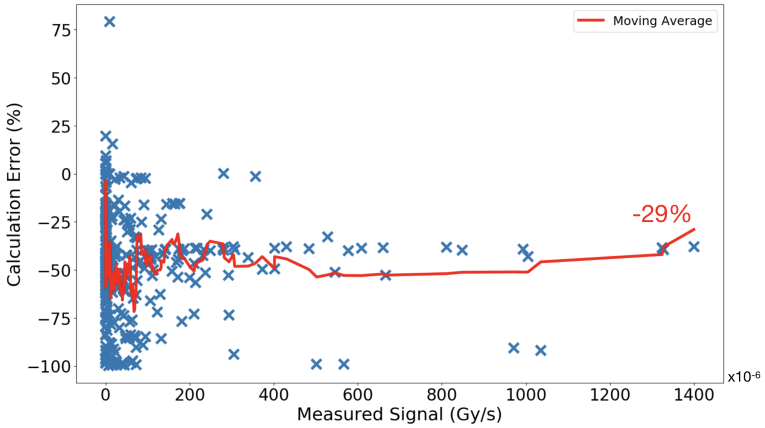


Figure 7.13: Scatter plot of the loss map measured signal points above 10^{-7} Gy/s, against the calculation error, for the two skew collimators. The moving average was calculated, showing the average error is mostly around -50% for low BLM signals and -30% for high BLM signals, from [16].

The skew collimator results from the three loss map campaigns are displayed in Figure 7.12, which plots the predicted signals against the measured signals, for the measured signal points above 10^{-7} Gy/s. A linear fit with an R^2 of 84% was

applied to the data, and the calculation error is plotted in Figure 7.13.

For the majority of the cases, the calculated cross-talk underestimates the measurements, with the average error starting off around -50%, and as the measured signal increases the error decreases to -29%. This is a result of the missing contribution from the skew collimators when calculating the cross-talk, as explained previously.

7.2.6 Calculation of Proton Impacts

Having successfully quantified the level of cross-talk observed by BLM devices in Section 7.2.4, this section seeks to calculate the equivalent impact rate for any collimator *j*, in protons / second (P_j). Following the same approach as in Section 7.2.2, the signals of all closed upstream collimators must be taken into consideration using Equation 7.3. This is transformed into Equation 7.8 to calculate the effect of a single collimator on its BLM.

$$S_n = \sum_{i=1}^{n-1} a_{ni}.S'_i + a_{nn}S'_n, \quad a_{nn} = 1$$

$$S'_n = S_n - \sum_{i=1}^{n-1} a_{ni}.S'_i \tag{7.8}$$

This is then extended into:

$$P_n = b_n.S'_n, \quad b_n = \frac{\Delta I}{S_n} \tag{7.9}$$

Where b^n is the cross-talk factor in protons/Gy of collimator *n*, calculated from the change in beam intensity (ΔI) against the BLM signal of collimator *n*, when it was the only one closed (Case n). The cross-talk factors of all collimators are collected in Table 7.5.

Table 7.5: Cross-talk factors (protons/Gy), calculated from the change in beam intensity using the individual cases, from [16].

Plane	**Collimator**	**Factor**
B1V	TCP.D6L7.B1	1.17×10^{12}
	TCSG.D4L7.B1	9.36×10^{11}
	TCLA.A6R7.B1	5.72×10^{11}
	TCLA.C6R7.B1	9.43×10^{11}
B1H	TCP.C6L7.B1	7.95×10^{11}
	TCSG.B4L7.B1	8.66×10^{11}
	TCSG.6R7.B1	6.48×10^{11}
	TCLA.B6R7.B1	4.39×10^{11}
	TCLA.D6R7.B1	6.72×10^{11}
	TCLA.A7R7.B1	6.59×10^{11}
B2V	TCP.D6R7.B2	9.54×10^{11}
	TCSG.D4R7.B2	7.32×10^{11}
	TCSPM.D4R7.B2	7.04×10^{11}
	TCLA.A6L7.B2	5.58×10^{11}
	TCLA.C6L7.B2	6.61×10^{11}
B2H	TCP.C6R7.B2	9.66×10^{11}
	TCSG.B4R7.B2	8.56×10^{11}
	TCSG.6L7.B2	7.56×10^{11}
	TCLA.B6L7.B2	5.12×10^{11}
	TCLA.D6L7.B2	1.34×10^{13}
	TCLA.A7L7.B2	2.85×10^{11}

The protons impacting each collimator i can be compared by representing them as a percentage of the total impacts per combined case (Case 1+...+n), using:

$$\mathrm{P}'_{\mathrm{i}} = \frac{\mathrm{P_i}}{\sum_{\mathrm{j=1}}^{\mathrm{n}} \mathrm{P_j}} \times 100, \quad \mathrm{i} \leq \mathrm{n} \tag{7.10}$$

The proton impacts calculated for the three combined cases in the vertical plane in beam 1 (recall Section 7.2.3) are displayed in Figure 7.14, indicating that 91% of all protons lost are assigned to the primary collimator, 9% are assigned to the secondary collimator and the rest are below 0.001%.

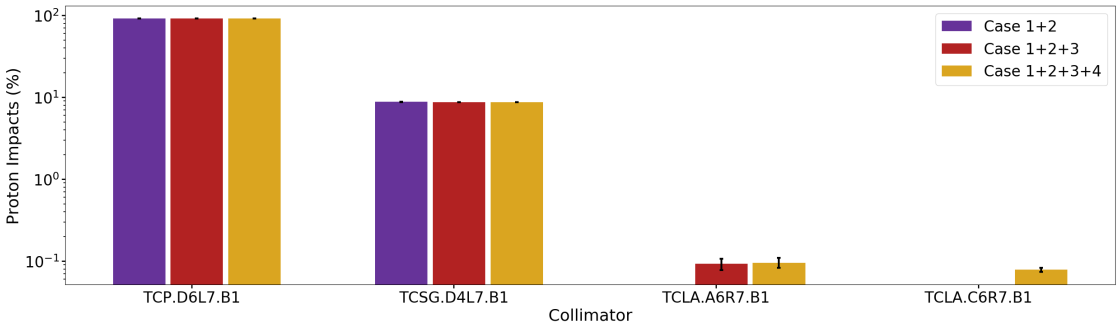


Figure 7.14: The percentage distribution of the predicted proton impacts for the three configurations of closed collimators in the vertical plane in beam 1; Case B1V - 1+2, Case B1V - 1+2+3, Case B1V - 1+2+3+4, from [16].

7.2.7 Simulated Proton Impacts

To validate the calculated proton impacts, the results are compared to simulated impacts. The simulations were run using Sixtrack and were set up using the methods described in [88], as this type of simulation has shown a good agreement with measurements in previous studies [88, 89, 90]. The simulations were set to use the collimator positions (refer to Table 7.2) and beam optics that were used during the beam tests.

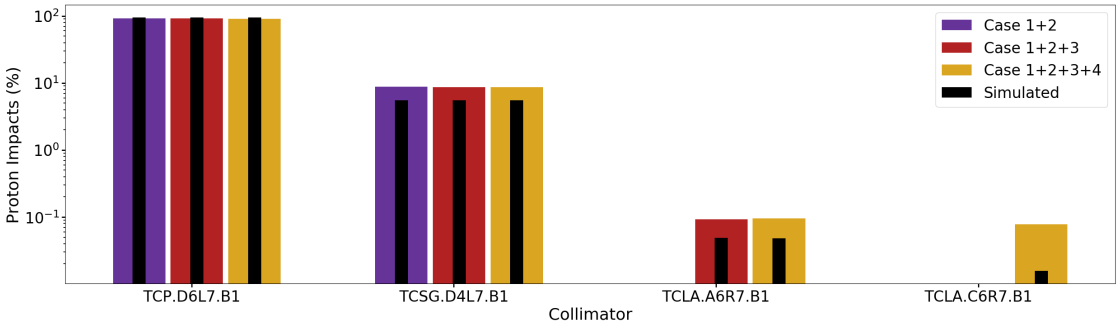


Figure 7.15: A comparison of the percentage distribution of the predicted and simulated proton impacts for the three configurations in the vertical plane in beam 1; Case B1V - 1+2, Case B1V - 1+2+3, Case B1V - 1+2+3+4, from [16].

The results for the vertical plane in beam 1 are displayed in Figure 7.15 which overlays the simulated impacts on the measured impacts (refer to Appendix B for a complete table of all the calculated impacts). From the simulations, 94% of all protons lost are assigned to the primary collimator, 6% are assigned to the secondary

collimator and the rest are below 0.001%, indicating a ±3% difference from the measured impacts. Overall the two sets of impact results are similar, however one can notice that the overestimation in the measured impacts increases the further downstream a collimator is. This is another effect of upstream collimators shielding showers before they reach the downstream collimators, which are assumed to be zero in this study.

7.3 Cross-talk Analysis Outlook

An initial cross-talk model was developed from the data gathered during commissioning 2018 (Section 7.1), to automate the collimator selection for parallel alignments. This involved making a list of the collimators that effect each other, to ensure that no two collimators effected by cross-talk are aligned in parallel. This is a first basic model which fully-prevents any cross-talk, however it limits the collimators that can be aligned in parallel.

This model can be enhanced through further analysis and can also be combined with other cross-talk studies (Section 7.2). Beam tests were performed to seek to quantify the levels of cross-talk observed by BLM devices to determine the actual beam loss signals generated by their corresponding collimators. The tests were focused on IR7 collimators and the cross-talk was successfully predicted with an error less than 10% of the measured BLM signal. In most cases, the cross-talk was overestimated as the showers shielded by upstream collimators were assumed to be zero.

These results were then applied to loss maps, to predict the losses at the first skew collimator positioned downstream, when losses are generated in the horizontal and vertical planes. The results indicate that the amount of cross-talk was underestimated for the majority of the cases, due to the contribution of the skew collimators which were not accounted for. Finally, the number of protons that impacted each collimator were then calculated. The impacts at the primary colli-

mators are conclusive as they are equal to the change in beam intensity. On the other hand, the losses at downstream collimators overestimate the amount of proton impacts, due to the showers shielded by upstream collimators assumed to be zero.

Overall this work provides a foundation for cross-talk analysis across BLM devices in LHC collimators which can be further expanded by taking measurements using more collimators, especially skew collimators. This can also be extended by repeating the measurements at flat top and analysing any energy dependent effects.

8. Software Implementation

The semi-automatic beam-based alignment procedure can now be transformed into a fully-automatic beam-based alignment procedure by integrating the three algorithms designed to replace the user tasks; automatic spike recognition model (Chapter 5), automatic threshold selection algorithm (Chapter 6) and cross-talk model for automatic parallel alignment (Chapter 7).

8.1 Fully-Automatic Alignment Procedure

Highlighting the differences from the semi-automatic BBA described in Section 3.2, the fully-automatic BBA involves the following steps:

1. **The user** can now select all collimator**s** to be aligned.
2. **The user** must then set **two** input parameters:
    - Δx_i - Jaw step size in μm, from a pre-defined list between 5 μm and 200 μm. The default step size is 10 μm at injection and 5 μm at flat top.
    - t_i^s - Time interval between each step in seconds, the minimum being 0.02 s. The default time interval is 0.02 s at injection and 0.2 s at flat top.

3. **The automated procedure** then selects; a collimator i, the jaws to move towards the beam (left or right or both), and a reasonable threshold, S_i^{Thres}. The alignment is then started automatically.

4. The jaw(s) of collimator i are automatically moved towards the beam in steps of Δx_i every t_i^s seconds.

5. The jaw movement automatically stops if the BLM losses exceed the threshold, S_i^{Thres} , by obtaining the BLM data S_i associated with collimator i after each jaw step.

6. Once the jaw(s) stop moving, **the automated procedure** uses machine learning to classify the spike exceeding the threshold, to determine whether the collimator jaw(s) are aligned or not.

7. **The automated procedure** first moves both jaws towards the beam simultaneously, then separately aligns each jaw twice, until a clear alignment spike is observed in each case. Therefore steps 3-6 must be repeated until these alignment spikes are observed, and must also be repeated for all selected collimators.

8. Once a collimator is considered aligned, **the automated procedure** will save its position in a database, such that the beam centre and beam size can be calculated.

This fully-automatic BBA algorithm is depicted in Figure 8.1 as a flowchart, and the entire process is started using the *play* button available in the GUI. This flowchart shows the overall process of selecting and aligning all collimators, using Figure 8.2 which shows the steps required to automatically align an individual collimator. Using the semi-automatic BBA the first step is to move all collimators in the current plane in parallel, to save time, therefore this was also included as the first step in the fully-automatic BBA.

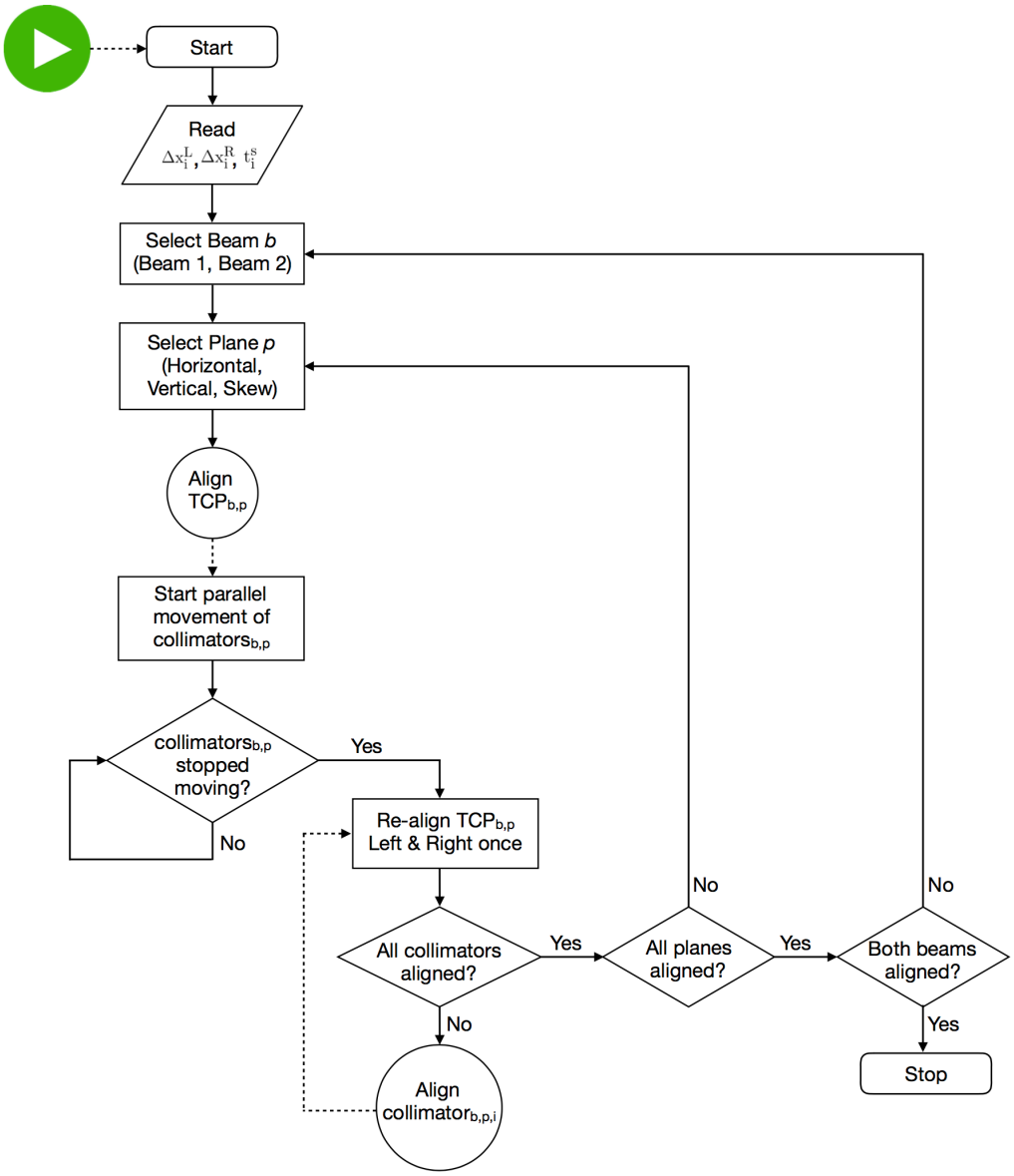


Figure 8.1: Flowchart of the fully-automatic beam-based alignment of all selected collimators, using Figure 8.2.

The alignment can also be paused or stopped at any point in time using the respective buttons, and their implementations are depicted as flowcharts in Figures 8.3, 8.4, respectively.

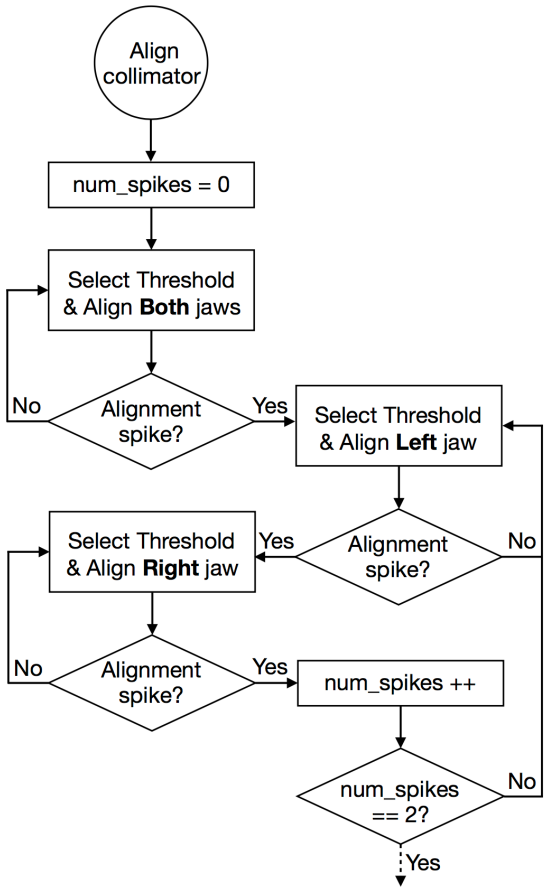


Figure 8.2: Flowchart of the fully-automatic BBA of an individual collimator.

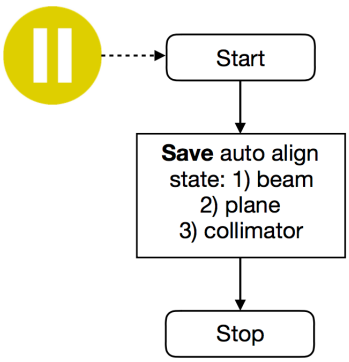


Figure 8.3: Flowchart to pause the fully-automatic BBA.

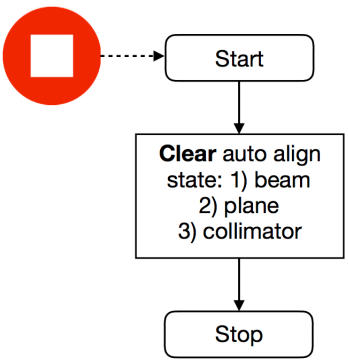


Figure 8.4: Flowchart to stop the fully-automatic BBA.

8.2 Software Development Tools

The Beams Department at CERN makes use of the following tools for software development:

- Java programming language for user application development.
- FESA real-time C/C++ environment (recall Section 2.1.2) for hardware control in the LHC.
- The Java API for Parameter Control (JAPC) for Java GUI applications to interact with the FESA middleware framework.
- The Eclipse Integrated Development Environment (IDE).
- Subversion (SVN) for version management.
- The CommonBuild Next Generation (CBNG) build and release management tool [91], based on Gradle [92].

The alignment is performed remotely from the CERN Control Centre using a top-level application implemented in Java, which allows users to move collimators whilst monitoring their BLM signal. These signals are logged at a frequency of 100 Hz, however they are shown to the user at a frequency of 25 Hz due to a limitation of the TCP (Transmission Control Protocol) communication protocol used.

8.3 Beam-based Alignment Software Architecture

The alignment application is based on the semi-automatic alignment software architecture [3] which is based on the original LHC collimator control software [93], as used in the 2010 LHC collimator alignments.

8.3.1 Semi-Automatic Software Architecture

The semi-automatic BBA is implemented in the FESA class *LHCCollAlign* [94] which runs on its own Front-End Computer (FEC) and is responsible for:

- Acting as a BLM concentrator by collecting the 100 Hz data from the BLMs of all the collimators via UDP (User Datagram Protocol), which is converted to 25 Hz to decrease the amount of processing required. The 25 Hz data is used by Java to display the signal for the user to view, and the 100 Hz data is logged for offline access.
- Running the feedback loop during collimator alignments to stop any movement when the losses in the BLM signal exceed the predefined threshold.
- Moving the collimators towards the beam until the BLM losses exceed the threshold, using the *Align Collimator* property. This process is started from the Java application by the user, who must provide the required input parameters and select the jaw(s) to align.

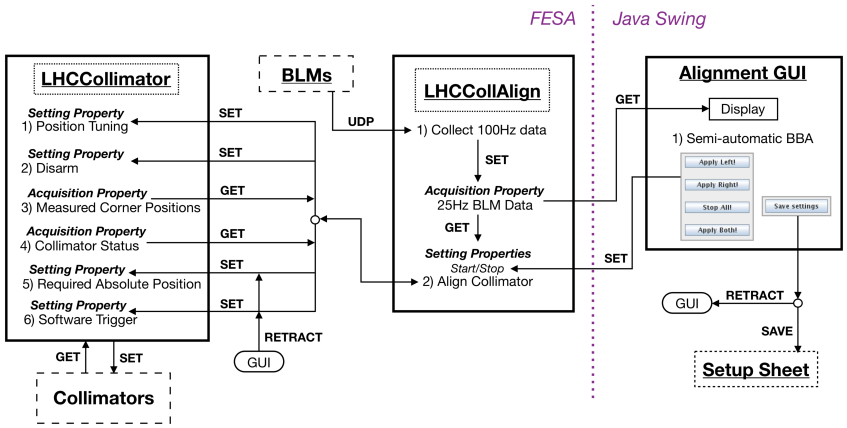


Figure 8.5: Overview of semi-automatic BBA implemented in FESA and Java, from [10].

The collimators are controlled using the *LHCCollimator* FESA class. This class is responsible for triggering and setting all collimator jaw movements, in particular

functions which move collimators during different phases of the operational cycle, and for getting any data related to the collimator positioning system [12]. An overview of the two FESA classes and the Java application is shown in Figure 8.5.

8.3.2 Fully-Automatic Software Architecture

The fully-automatic BBA is implemented on top of the semi-automatic BBA, thus allowing for both alignment tools to be available together and maintaining backward-compatibility with all previous functionality. It is implemented in the FESA class *CollAlignSupervisor* on its own FEC, and is responsible for:

- Repeatedly automatically aligning the collimators towards the beam by calling the *Align Collimator* property in the *LHCCollAlign* FESA class, as used by the semi-automatic BBA. The difference being that the user does not provide the input parameters as these are computed automatically. The full-automation also selects the jaw(s) to align and the entire procedure is performed by calling the *Automatic Alignment* property.

- Automatically selecting the collimators to align in parallel based on the offline cross-talk analysis results, by providing the FESA class with the list of collimators effected by cross-talk. This is accessed by the algorithm before selecting any collimator to align, to first check which collimator (if any), is being aligned in the other beam and which collimators it effects. This ensures that no collimators effected by cross-talk are aligned in parallel.

- Automatically selecting the threshold for aligning any collimator based on the latest BLM signal. This is implemented by accessing the latest 7.5 seconds of raw 25 Hz BLM data from the *LHCCollAlign* FESA class, and selecting the threshold accordingly. This algorithm keeps track of the previous threshold selected in the *CollAlignSupervisor* FESA class, to enhance the threshold selection process.

- Applying the machine learning model on the recorded BLM signal, to automatically detect alignment spikes and determine if the collimator is aligned. Machine learning model implementations are available in Python therefore a separate thread on the same FEC as the *CollAlignSupervisor* FESA class is dedicated to run the Python script developed for alignment classifications. This script is directly provided with the latest raw 25 Hz BLM data obtained from the *LHCCollAlign* FESA class and returns the classification result to the *CollAlignSupervisor* FESA class to resume the alignment based on the result.

- Automatically save the positions of the aligned collimators to a table and save all information related to the alignment in a file which can only be edited from the CCC. This information is used to automatically retract the collimators to their final position in the hierarchy.

The fully-automatic BBA is controlled from the same Java application, using the three new buttons; Play, Pause and Stop. The full implementation of how the new FESA class interacts with the semi-automatic alignment and the Java application, is shown in Figure 8.6.

FESA allows for executing multiple instances of the same property in different threads. Therefore in order to allow for the alignment of the collimators in the two beams to be done in parallel, two threads were assigned to the *CollAlignSupervisor* FESA class. This allows for executing two instances of the same *Automatic Alignment* property in parallel. The user can use two separate consoles to select two lists of collimators, as was the case with the semi-automatic BBA, and upon pressing play, the Java application sends the list of selected collimators to FESA on an available thread.

These two threads constantly communicate with each other before starting the alignment of any collimator, to ensure that the selected collimators to align in

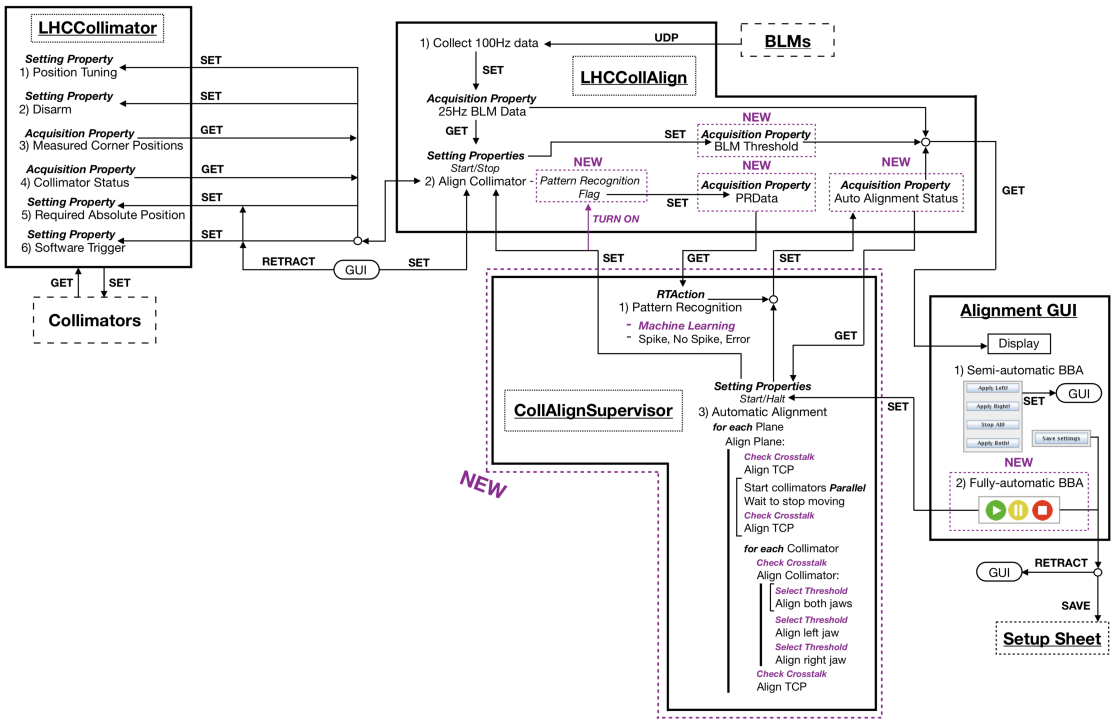


Figure 8.6: Overview of the fully-automatic BBA implemented in FESA and Java, from [10].

parallel are in accordance with the cross-talk analysis. The algorithm in FESA begins by using the cross-talk analysis to sort the selected collimators in descending order, according to the most collimators they effect through cross-talk. Therefore the collimators which generate the most cross-talk, TCLA.A6 and TCLA.B6 (recall Figure 7.3), will always be aligned first when selected. After a collimator is aligned, the respective thread checks the sorted list of remaining collimators to be aligned and checks the collimator being aligned in the other beam. These checks are used to select the first collimator on the list that does not experience cross-talk from the collimator being aligned in the other beam. If no collimator is found, the thread sleeps until the collimator in the other beam has finished its alignment. Whilst the automatic alignment is running, the user retains full control over pausing, resuming and stopping the alignment whenever required. Similarly, the alignment of the primary collimator of the corresponding plane before and after, is handled using the same cross-talk model, to check when it can be aligned according to what

is being aligned in the other beam at that moment.

Furthermore, the parallel movement of all collimators in any plane was implemented such that the software aims to start new planes simultaneously when both beams are starting a new plane. On the other hand, if only one beam (eg.: beam 1) is starting a new plane, its thread waits until beam 2 has finished aligning its current collimator, with both the primary before and after. Once the beam 2 collimator is aligned, the fully-automatic alignment of beam 2 will automatically pause, and beam 1 can then start the parallel movement of all the collimators in its current plane. Beam 2 waits for beam 1 to finish its parallel movement so that it can automatically resume its alignment as explained previously.

8.4 Safety-Critical Design

The fully-automatic BBA must ensure safe operation within the collimation software at all times, therefore this section discusses the precautions taken to avoid additional beam losses and ultimately preventing dumping the beam.

Firstly, the real-time thread which receives the BLM UDP packet is given the highest priority. This thread is responsible for decoding the BLM signal and to determine if the losses exceeded the BLM threshold, in which case the jaw movement must be stopped immediately. Secondly, the BLM thresholds available for selection are an order of magnitude lower than the losses close to the BLM thresholds that trigger a beam dump (refer to Chapter 9).

In addition, a number of default settings are applied when using the fully-automatic software. As mentioned in Sections 4.2 and 8.1, at injection the default step size is 10 µm and the time window for spike recognition is 4 seconds, whereas at flat top the default is 5 µm and the time window is 6 seconds, as the losses are much higher. Furthermore, the process of moving all plane collimators in parallel towards the beam (recall Section 8.1) is disabled at flat top due to the major losses observed during commissioning 2017, which is later explained in Section 9.2.

8.5 Graphical User Interface

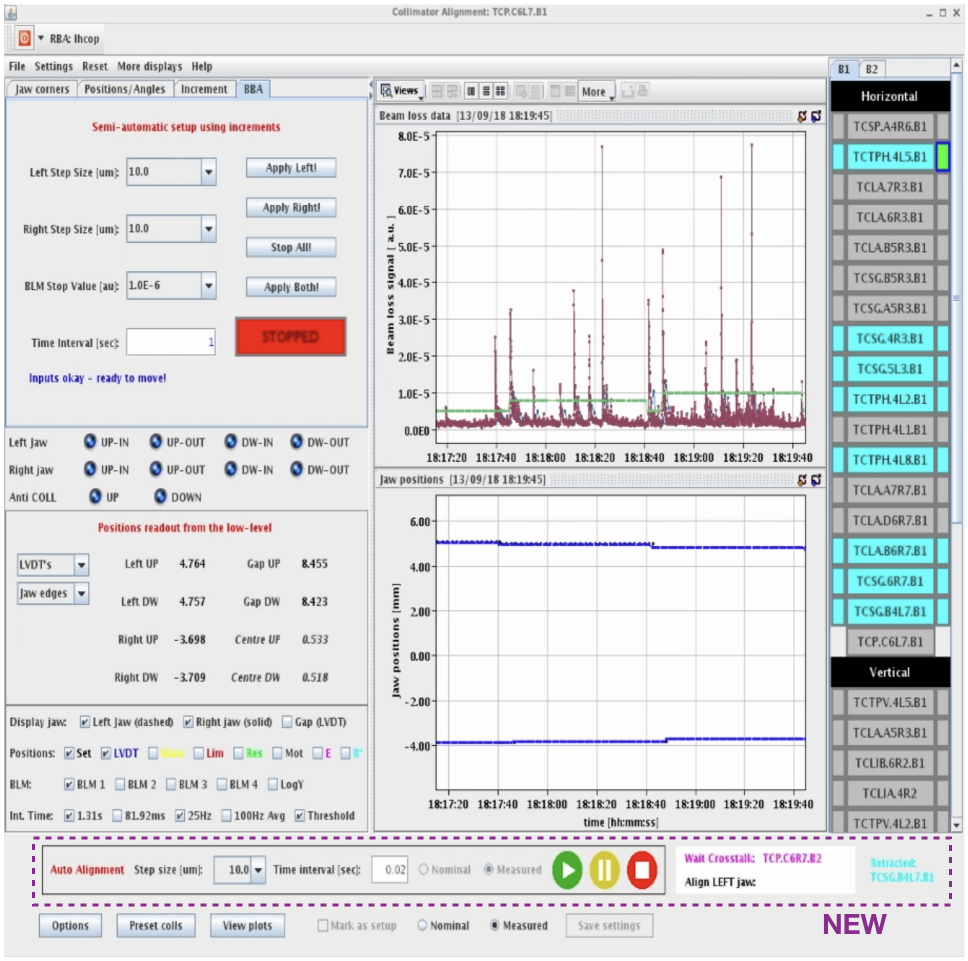


Figure 8.7: A screen shot of the collimator controller window providing fully-automatic beam-based alignments, from [10].

The fully-automatic BBA GUI extends the previous semi-automatic BBA GUI [95], by introducing a new section as shown by the screen shot in Figure 8.7. The user can input the jaw step size, the time interval between each step and select between the nominal or measured settings when retracting the collimators to their final positions. The alignment can be paused, stopped and resumed by the user at any time. The GUI subscribes to the 25 Hz BLM data and 1 Hz jaw position for display purposes. The GUI lists the collimators to be aligned on the right, grouped by their orientation, such that each group ends with the primary/reference collimator of the

plane. A collimator must be selected from this list to control it and to show its corresponding BLM signal and jaw position, with the selected collimator is marked by a blue border. The state of each collimator is indicated by the various colour codes listed in Table 8.1.

Table 8.1: Color coding used to identify the different alignment states of an individual collimator, from [10].

Color	**State**
Gray	Not yet moved
Green	Collimator jaw(s) moving
Yellow	Fully-automatic BBA paused by user when this collimator was mid-action.
Red	Fully-automatic BBA stopped by user when this collimator was mid-action.
Turquoise	Collimator aligned and jaws retracted to operational settings.
Magenta	Collimator jaw(s) waiting to align, due to cross-talk from another collimator.

The GUI is operated from the CCC and two instances of the same application can be opened simultaneously. This allows for aligning a maximum of two collimators in parallel, one from each beam.

9. LHC Collimator Alignment Campaigns

At the start of each year, the LHC goes through a commissioning phase to ensure that everything is correctly set up and ready for nominal operation. The machine requirements for commissioning the collimators are as follows:

- Establish Golden Orbit - Before starting the collimator alignment, the LHC golden reference orbit needs to be established, which must be set up in advance. Following this, 1 or 2 nominal LHC bunches with a total beam intensity below 3×10^{11} p/beam, need to be injected. The individual bunch intensity must be close to 1×10^{11} p/beam to operate the LHC BPM at low gain, in order to satisfy the requirement of the BPM measurement and bunch intensity. If beam loss maps need to be performed after the alignment, then more bunches (with a lower intensity of 10^{10} p/beam) can be injected whilst keeping the limit below the 3×10^{11} p/beam. Collimators are designed to resist such low intensity beams, foreseeing cases when they accidentally cut into the beam.

- Setup Beam Flag - During the alignment, in order to avoid an unnecessary beam dump, the setup beam flag needs to be set to allow masking hardware

interlocks at low beam intensity (3×10^{11} p/beam). The EiC (Engineer in Charge) must ensure the flag time-out does not elapse and trigger a dump.

- BLM limits - BLM devices downstream from the collimators will measure unusually high signals, that risk dumping the beam if not masked. The BLM devices protecting sensitive equipment, such as the superconducting magnets, can never be masked as the machine must always be protected. Therefore the EiC masks the Maskable BLM devices.

- Collimator Position Limits - The EiC opens all collimator jaw position thresholds and masks collimator movements.

- Measurement of Beam Emittance - It is desirable that the emittance is measured before the alignment by doing a wire-scan [96]. The emittance gives an idea of the transverse beam size and indicates if there is an issue worth dumping the beam and re-injecting for the alignment, in order to have nominal beam conditions.

- Orbit Feedback Configuration - The EiC will either switch off the orbit feedback or change the settings to have a very smooth correction, to avoid sudden orbit shifts which could induce false spikes or cause unnecessary beam loss.

During the commissioning phase, all collimators are aligned during their respective machine stages, to set up the correct hierarchy. Figure 9.1 depicts the different stages of the LHC machine cycle and the corresponding beam size (β^*) at each stage.

As of 2018, the collimators aligned at each machine stage are:

- Injection - 75 collimators and 4 injection protection collimators are aligned with BLM devices.

- Flat top - 75 collimators are aligned with BLM devices.

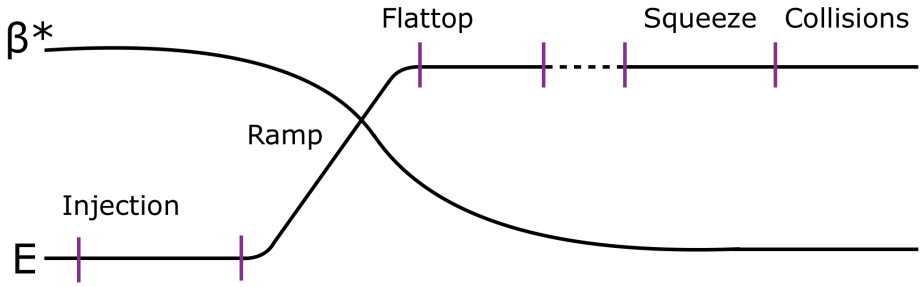


Figure 9.1: The stages of the LHC machine cycle and the corresponding beam size.

- End of Squeeze - 16 TCTs are aligned with BPM pick-ups, as only the optics are changed.
- Collisions - 16 TCTs are aligned with BPM pick-ups and 12 TCLs (physics debris collimators) aligned with BLM devices, as the orbit is only changed at the experimental IPs.

Table 9.1: List of 2018 settings for collimation. The settings are expressed in beam size values (σ), assuming a nominal beam emittance of 3.5 μm.

Coll	**IR**	**Injection**	**Flat top**	**Squeeze**	**Collisions**
TCP	7	5.7	5.0	5.0	5.0
TCSG	7	6.7	6.5	6.5	6.5
TCLA	7	10.0	10.0	10.0	10.0
TCP	3	8.0	15.0	15.0	15.0
TCSG	3	9.3	18.0	18.0	18.0
TCLA	3	12.0	20.0	20.0	20.0
TCSP	6	7.5	B1: 7.8 / B2: 7.4	7.3	7.3
TCDQ	6	8.0	B1: 7.8 / B2: 7.4	7.3	7.3
TCT	1/5	13.0	15.0	9.0	9.0
TCT	8	13.0	15.0	15.0	15.0
TCT	2	13.0	37.0	37.0	37.0
TCL	4/5	-	-	-	15.0
TCL	6	-	-	-	-

As explained in Section 2.3, collimator alignments are performed to determine the beam orbit and beam size at each collimator location, which are initially unknown as the actual beam orbit may deviate from the design orbit and the collimator tank may be misaligned. This information is required to establish the hierarchy,

such that the jaws are positioned within a certain number of standard deviations (beam σ) from the beam centre, and to generate continuous setting functions for the whole LHC cycle [97, 98]. The collimator settings used during 2018 operation at the different machine states are listed in Table 9.1.

9.1 Alignment Evolution

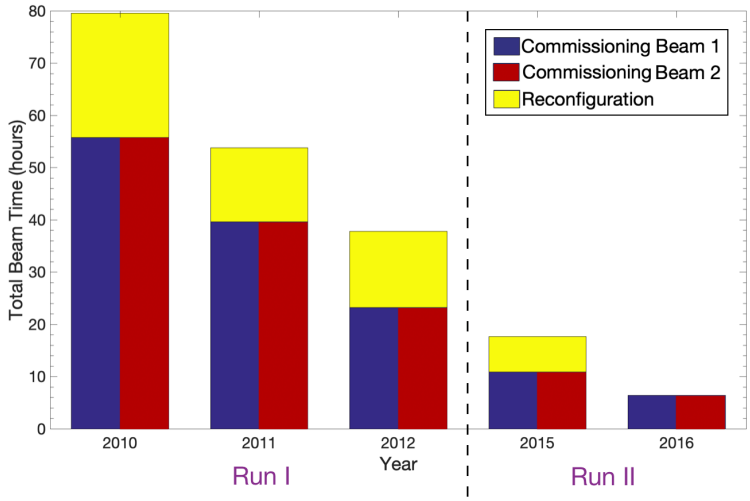


Figure 9.2: The time taken to align all collimators during commissioning and any reconfigurations, between 2010 and 2016, using the manual and semi-automatic setup. The collimators in the two beams were aligned in parallel as indicated by the vertical bars.

The cumulative net time required to align the collimators over the years during all machine states, is displayed in Figure 9.2, excluding any operational overhead for configurations and setups, prior starting the alignment. This diagram clearly indicates the consistent decrease in the alignment time each year. The major improvements leading to collimation alignment milestones over the years include:

- Run I
  - 2011: Introduction of the semi-automatic alignment software using 1 Hz BLM data.

  - 2012: Availability of 12 Hz BLM data.

- Run II
  - 2015: First collimators installed with BPM pick-ups.
  - 2016: Availability of 100 Hz BLM data.

The initial alignment of the system in 2010 took 55 hours. Since then, several software and hardware upgrades were introduced to improve the alignment time and to simplify the alignment procedure.

The majority of the collimators are aligned with BLM devices, however in Long Shutdown 1 (before the start of Run II) the first collimators were installed with BPMs. Collimators aligned with BPMs use an alignment algorithm proposed in [99], which automatically centres the collimator jaws around the beam, based on its position. The algorithm is based on successive approximation, which is required due to the non-linear BPM sensitivity to beam displacement.

9.2 Semi-Automatic Commissioning 2017

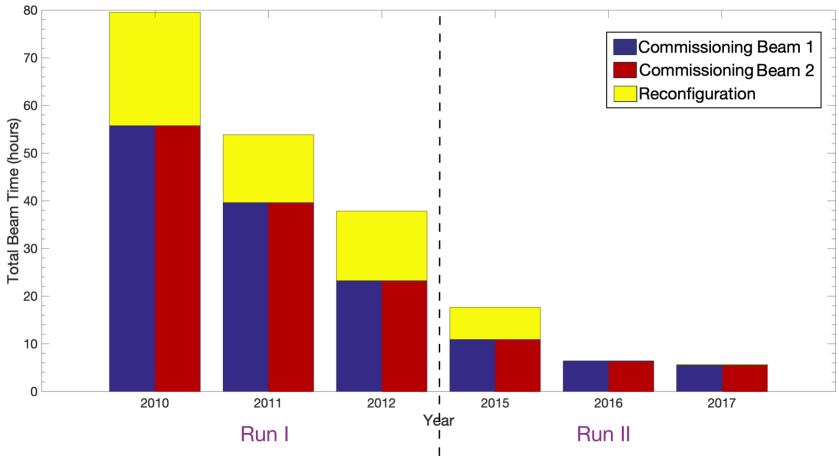


Figure 9.3: The time taken to align all collimators during commissioning and any reconfigurations, including data from 2017.

For commissioning in 2017, a minor update was added to the semi-automatic alignment algorithm. This involved stopping the jaw movement when a minimum of two points exceed the threshold, rather than just one, to decrease the number of times the jaws are stopped by noisy spikes. Additionally, the GUI was updated to show both the measured and nominal collimator values.

Figure 9.3 compares the cumulative net time required to align the collimators including 2017 data, showing a minor improvement. Overall this indicates that the semi-automatic alignment seems to have reached its peak alignment time performance at around 6 hours.

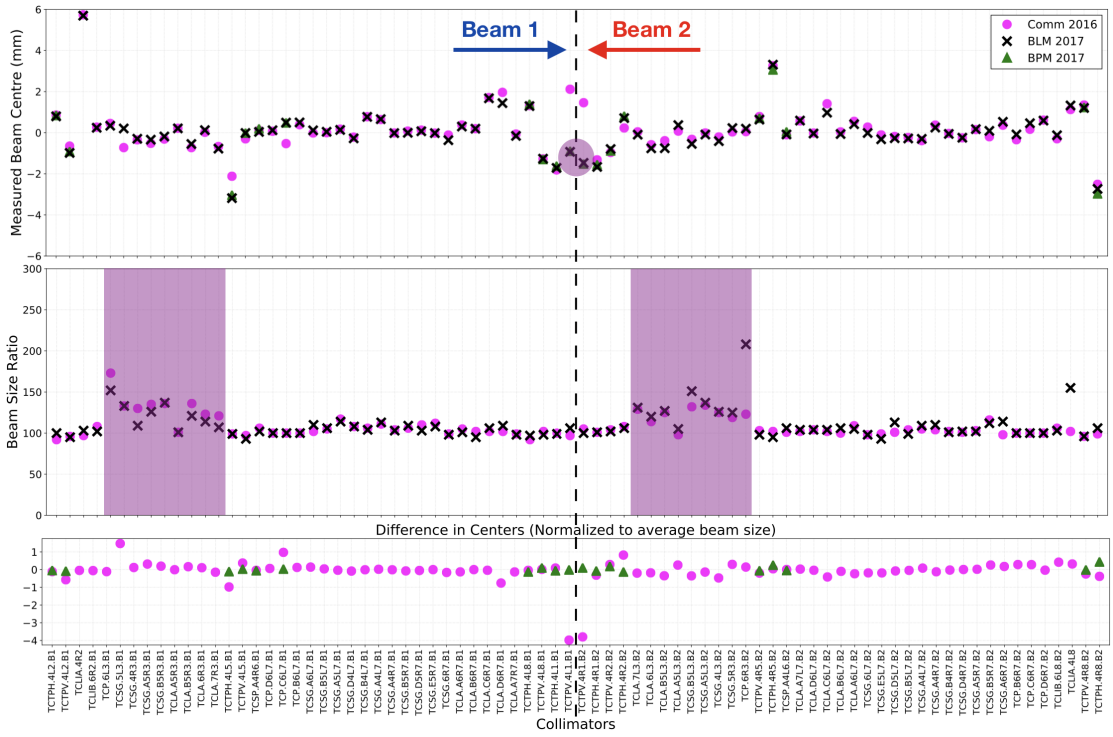


Figure 9.4: Injection commissioning 2017 results, comparing the beam centres measured with BLM and BPM devices to 2016 commissioning centres, highlighting the change in polarity of the crossing angle in IR1 (top graph). The beam size ratios are compared to 2017 (middle graph), those in IR3 are shaded as they are expected to be large due to the higher dispersion in that location. Finally the differences between the 2017 BLM centres and the rest are compared after normalising by the average beam size of 2016 and 2017 (bottom graph).

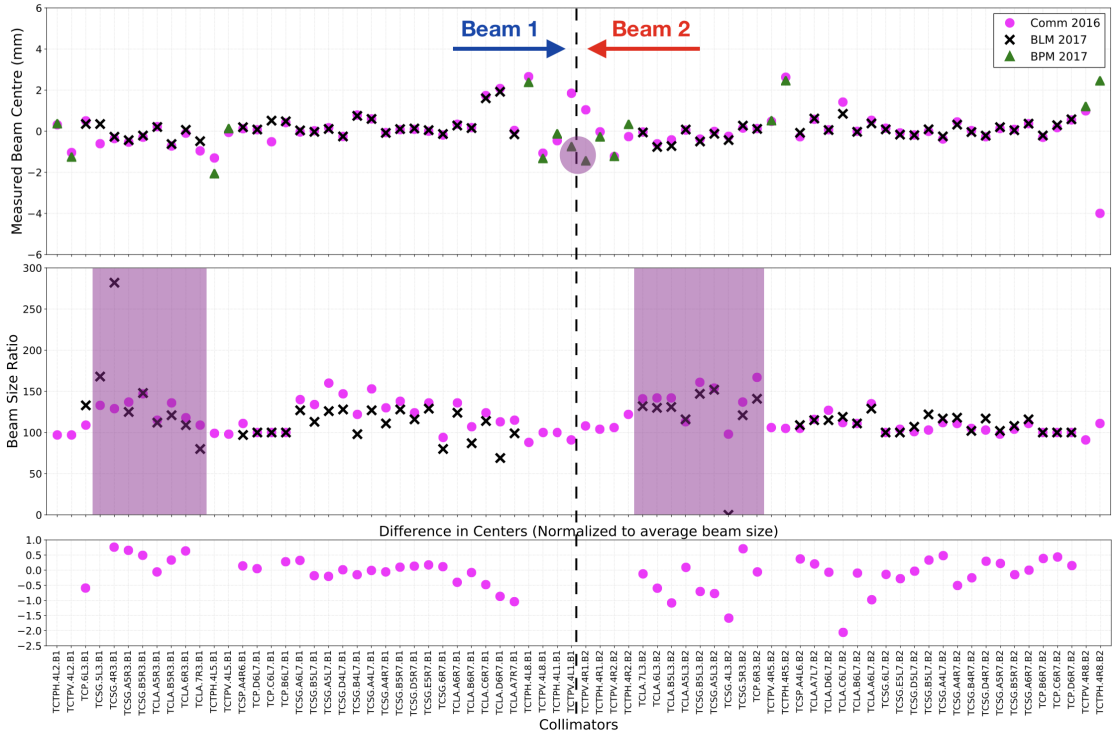


Figure 9.5: Flat top commissioning 2017 results, comparing the beam centres measured with BLM and BPM devices to 2016 commissioning centres, highlighting the change in polarity of the crossing angle in IR1 (top graph). The beam size ratios are compared to 2017 (middle graph) and the differences between the 2017 BLM centres and the rest are compared after normalising by the average beam size of 2016 and 2017 (bottom graph).

The centres measured for beam 1 and beam 2 at injection (Figure 9.4) and flat top (Figure 9.5), are similar to those found in 2016 as expected, as nothing was changed and any slight differences could be due to drifts in ground motion. The only exceptions are the TCTs in the vertical plane in IR1, which reflect the change in polarity of the crossing angle with a change in sign.

These results together with the loss maps shown in Appendix C, validate the alignment campaign and the settings were used throughout LHC operation in 2017. Overall it was a successful beam commissioning alignment campaign, where 3 collimators were introduced/upgraded and two events were encountered [100]:

1. The TCSPM.D4R7.B2 was a new collimator aligned for the first time with BLM and BPM devices at injection.

2. The TCP.C6L7.B1 and TCL.4L5.B2 were replaced and embedded with BPMs and were aligned for the first time at injection.

3. Whilst moving all the collimators in a plane in parallel at flat top, one of the BLM devices detected losses higher than its threshold as shown in Figure 9.6, which resulted in a beam dump. For this reason, parallel movements at flat top have been disabled by default.

4. Loss maps performed at a later stage indicated that the TCSG.B5L7.B2 was misaligned at flat top, as during alignment its jaws were stopped early and the centre found was off by around 300 µm. This was as a result of human error and the collimator was re-aligned.

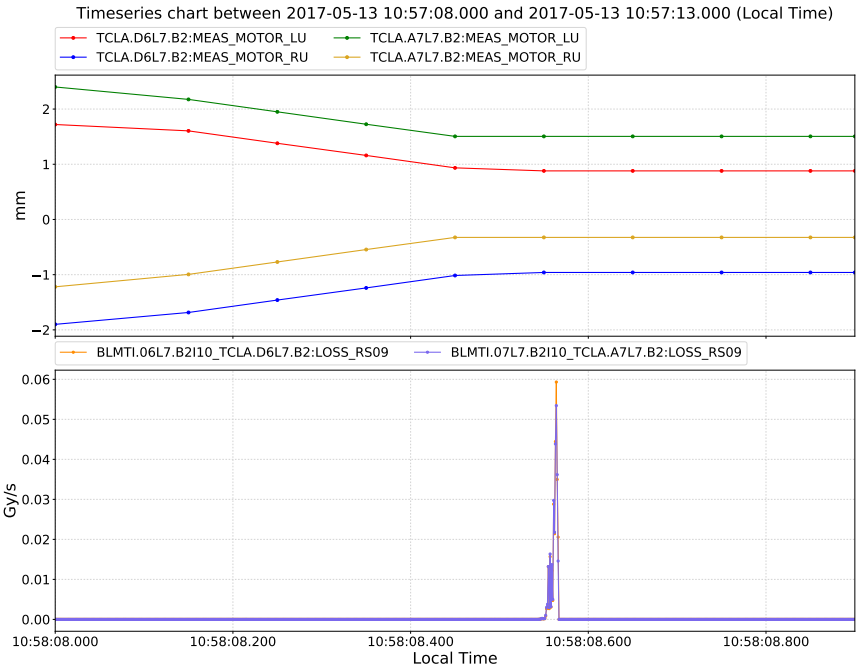


Figure 9.6: High TCLA.D6L7.B2 BLM losses, during parallel movement at flat top collimator commissioning, caused a beam dump.

9.3 Fully-Automatic Commissioning 2018

The latest alignment with BLM software upgrade is the full automation developed as part of this dissertation. In 2018, it was used for LHC collimator commissioning for the first time.

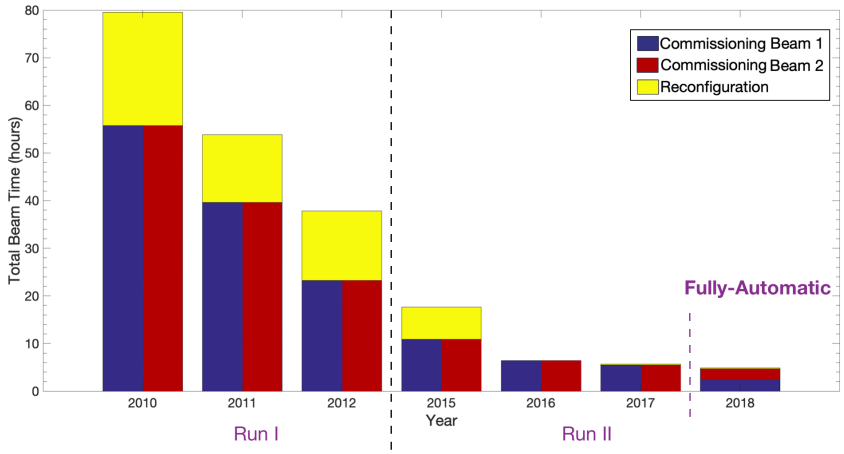


Figure 9.7: The time taken to align all collimators during commissioning and any reconfigurations, including data from 2018, from [26].

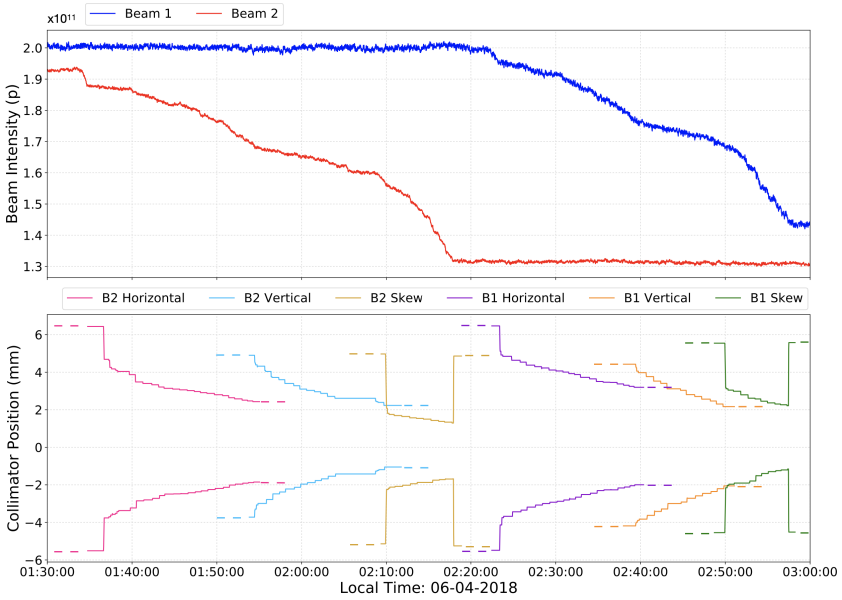


Figure 9.8: Sequential alignment of the collimators in the two beams at injection, showing the beam intensities (top graph) and only the positions of the primary (TCP) collimators (bottom graph) for simplicity, from [75].

Figure 9.7 compares the cumulative net alignment time using the full automation, indicating that the collimators in the two beams were aligned sequentially, in contrast to previous years. The reason is that, in the past, collimation experts would cautiously examine the cross-talk to select the collimators to align in parallel, however the fully-automatic did not yet account for this (recall Section 7.1). Ultimately, one can still appreciate a decrease in the alignment time required.

During commissioning at injection, the alignment software worked with the default settings, by predicting the spike class, using the trained machine learning models, after 4 seconds of spike decay and using a 0.02 second time interval between collimator movements of 10 μm.

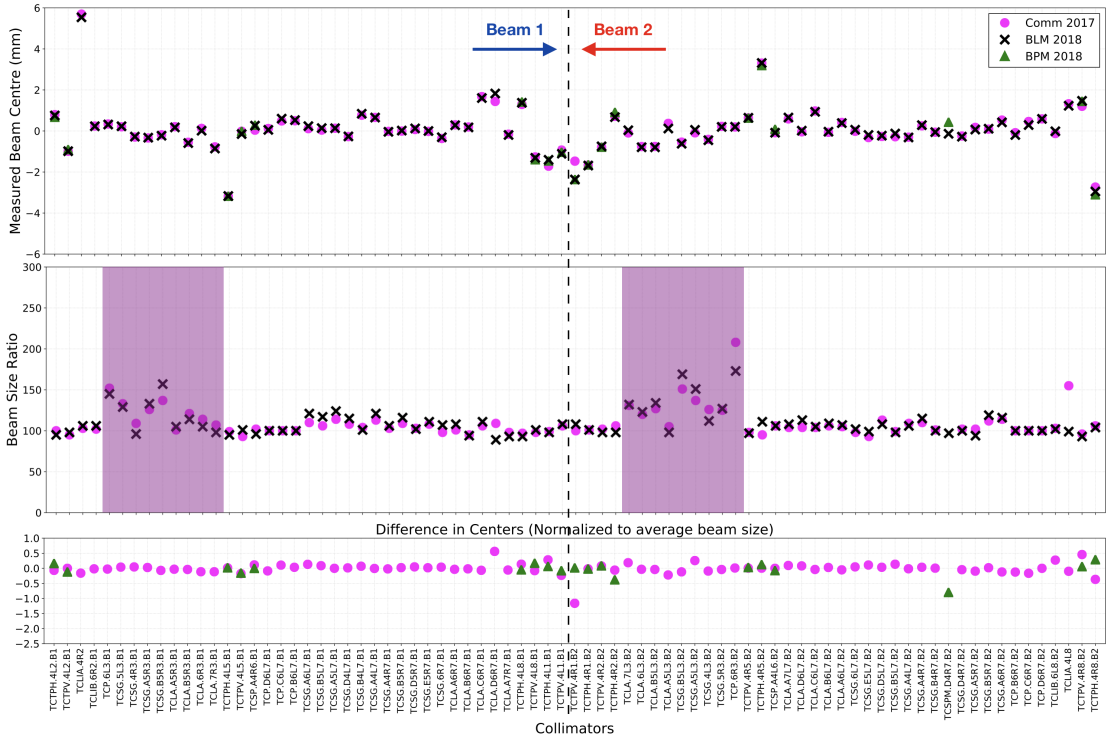


Figure 9.9: Injection commissioning 2018 results, comparing the beam centres measured with BLM and BPM devices to 2017 commissioning centres (top graph), the beam size ratios compared to 2017 (middle graph), and the difference between the 2018 BLM centres are compared to the rest after normalising by the average beam size of 2017 and 2018 (bottom graph), from [75].

As explained in Section 2.3, the standard BBA aligns the primary collimator, of the corresponding plane, before and after each collimator is aligned. Figure 9.8 summarizes the alignment campaign at injection commissioning by plotting the positions of the primary collimators in the horizontal, vertical and skew planes in each beam. The collimators in the two beams can be seen to be aligned sequentially with beam 2 followed by beam 1. The entire alignment campaign required 1.5 hours for 79 collimators, reaching the theoretical minimum time calculated in Section 4.2 in the case of each collimator encountering one non-alignment spike. On average it took 1 minute to align a single collimator using this approach.

The collimator centres measured at injection with BLM detectors are similar to the centres obtained during injection commissioning in 2017, and are consistent with the centres measured using BPMs (where available), as shown in Figure 9.9. The beam sizes are also consistent with 2017, evidently showing the reproducibility of the LHC after correction of orbit and optics. Furthermore, the difference in centres is normalized to the average beam size of 2017 and 2018, to better compare the centres across the LHC ring.

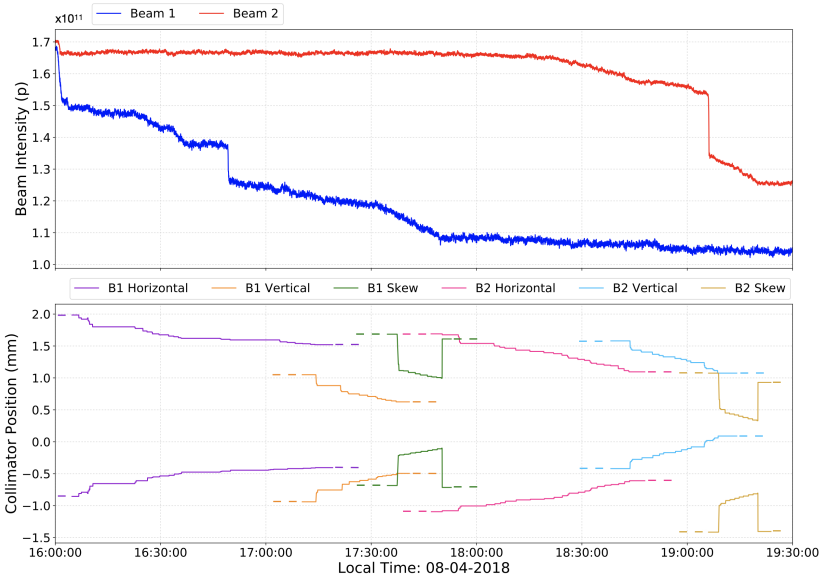


Figure 9.10: Sequential alignment of the collimators in the two beams at flat top, showing the beam intensities and only the positions of the primary (TCP) collimators for simplicity, from [75].

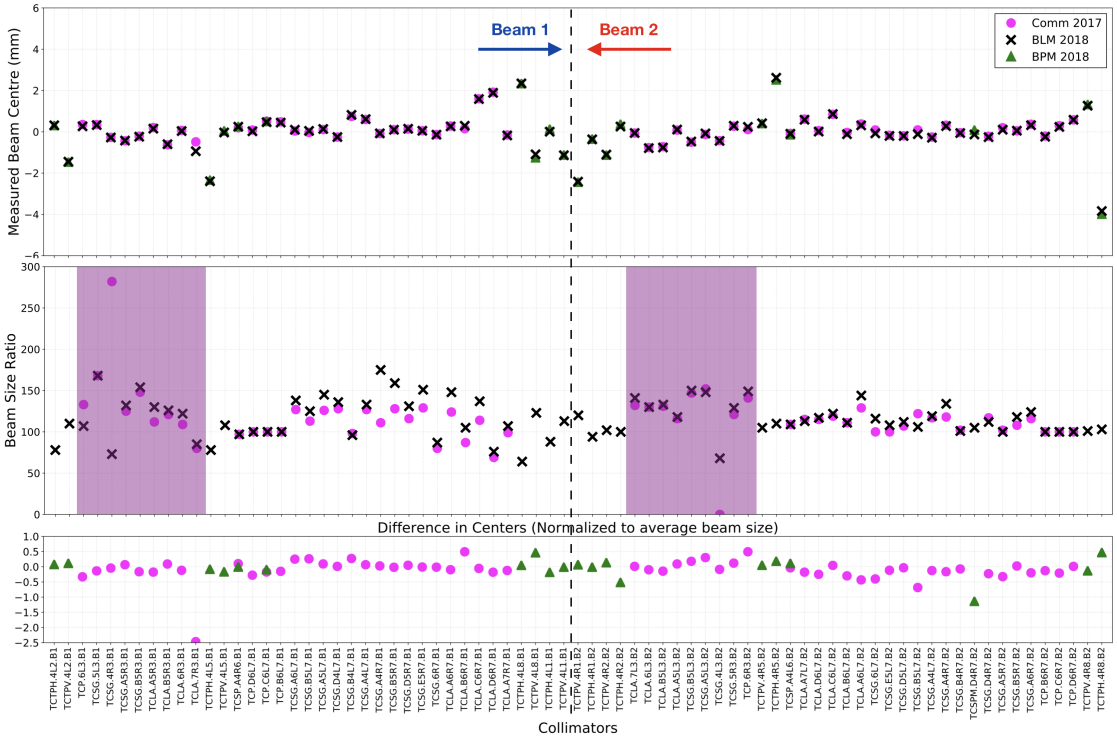


Figure 9.11: Flat top commissioning 2018 results, comparing the beam centres measured with BLM and BPM devices to 2017 commissioning centres (top graph), the beam size ratios compared to 2017 (middle graph), and the difference between the 2018 BLM centres are compared to the rest after normalising by the average beam size of 2017 and 2018 (bottom graph), from [75].

During commissioning at flat top, the alignment conditions were difficult due to beam stability issues, especially in the horizontal plane in beam 1. Due to this, the settings of the fully-automatic tool were modified to be more conservative with collimator movements to limit high losses. This was performed by predicting the spike class, using the trained machine learning models after 6 seconds of spike decay and using a 0.2 second time interval between collimator movements of 5 μm. Using these settings, the 75 collimators were aligned in 3.5 hours, therefore on average it took 2 minutes to align a single collimator using this approach. The entire alignment campaign is displayed in Figure 9.10, showing that the collimators in the horizontal plane in beam 1 took longer to align than the same number of collimators in the horizontal plane in beam 2. Similar to injection, the flat top beam centres and sizes measured with BLMs and BPMs (where available) are consistent

between 2017 and 2018, as shown in Figure 9.11.

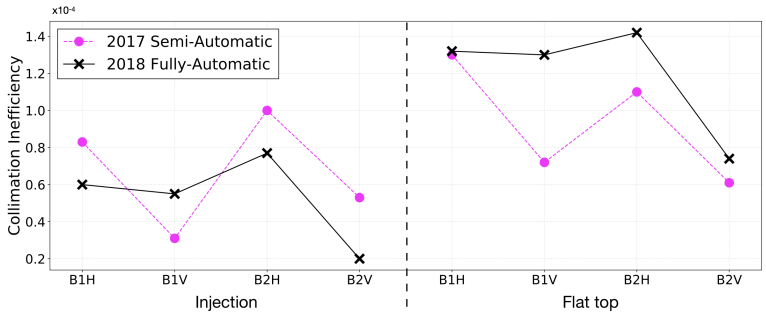


Figure 9.12: Comparison of losses in the IR7 DS after commissioning using the fully-automatic alignment in 2018 and the semi-automatic in 2017, from [75].

These commissioning campaigns at injection and flat top were validated by generating loss maps (refer to Appendix C), which are summarized by observing the losses in the dispersion suppressor in IR7. These losses determine the collimation inefficiency, and Figure 9.12 compares the inefficiency between 2017 and 2018. The difference is within an acceptable limit, as any misalignments would result in a factor of two to three difference [101]. Therefore the results are consistent, thus validating both alignment campaigns.

9.3.1 Commissioning for Ion Beams

During ion beam commissioning in 2018, the fully-automatic alignment software was used to align all IR7 collimators in collisions. The losses generated by ion beams exceed the proton losses by a factor 100 in the dispersion suppressor (DS) [102]. As a consequence, the fully-automatic tool was modified to avoid high DS losses by using a 0.1 second time interval between collimator movements of 10 μm. Table 9.2 summarises the various settings used for alignment campaigns performed at different machine states. Using ion beams the collimators in the two beams were aligned sequentially and on average it took 1.5 minutes to align a single collimator using this approach.

Since no heavy iron run took place in 2017, there is no data available to compare to the 2018 alignment. However the collimators in IR7 are not expected to have

Table 9.2: Angles obtained from BPM collimator alignments in beam 2.

Beam type	**Machine State**	**Decay time**	**Step size**	**Step rate**
Proton	Injection	4 s	10 µm	50 steps/s
Proton	Flat top	6 s	5 µm	5 steps/s
Ion	Collisions	6 s	5 µm	10 steps/s

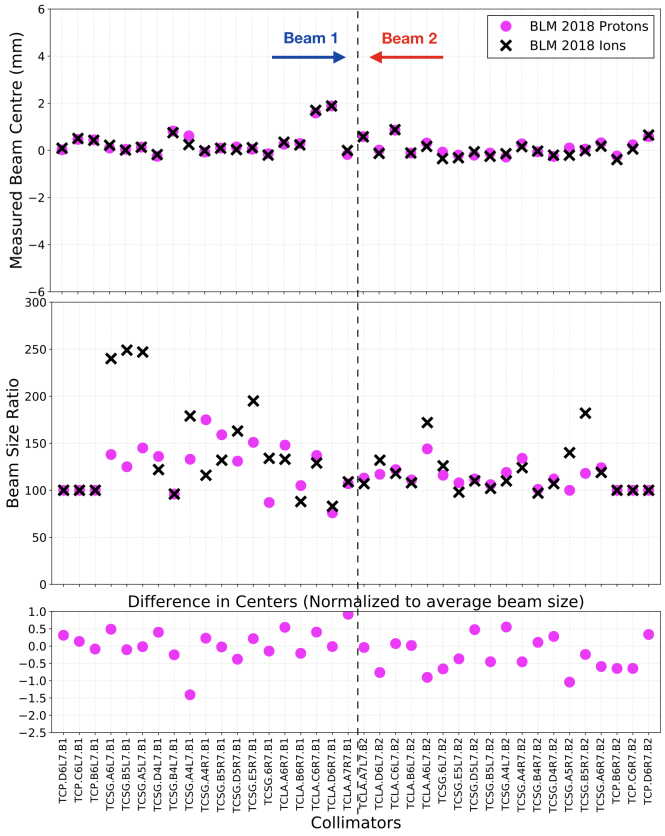


Figure 9.13: Ion commissioning IR7 BLM results in collisions comparing the beam centres and beam size ratios to flat top commissioning results, from [75].

large changes between collisions with ion beams and at flat top with proton beams. Therefore this comparison of centres is displayed in Figure 9.13, and the results are consistent for the majority of the collimators. A couple of collimators indicate a difference between ±150 µm and ±200 µm, which could be expected due to the change of reference orbit between the two beam types.

9.4 Parallel Fully-Automatic MD 2018

An MD study was scheduled at injection in 2018, to test a second version of the fully-automatic alignment software, able to align the collimators in the two beams in parallel [103]. The default injection settings were used for the alignment software, i.e. applying machine learning after 4 seconds of spike decay and using a 0.02 second time interval between collimator movements of 10 μm. The alignment of 79 collimators at injection took 50 minutes, decreasing the alignment time at injection by a factor of three, with respect to 2017 as shown in Figure 9.14. On average it took 1.5 minutes to align two collimators using this approach.

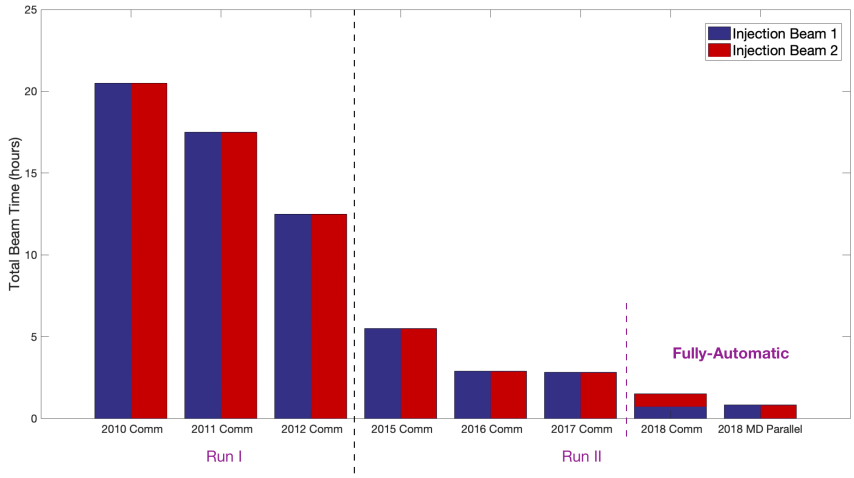


Figure 9.14: The time to align 79 collimators at injection commissioning, compared to the 2018 parallel alignment MD, from [26].

The two beams were aligned in parallel and the entire alignment campaign is displayed in Figure 9.15. The beam centres and beam sizes measured at each collimator are consistent with those from commissioning 2018, as shown in Figure 9.16. This validates the new parallel alignment software, confirming that it can be used as a baseline for future parallel alignments at injection, and should be repeated for flat top data. In the future, this software can be further upgraded based on more in-depth cross-talk analysis studies as explained in Section 7.2.

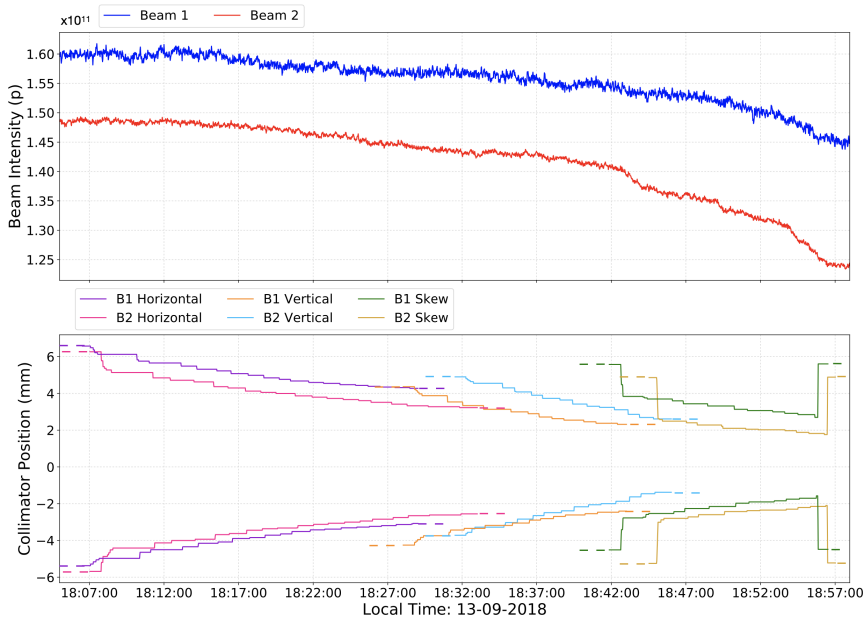


Figure 9.15: Parallel alignment of the collimators in the two beams at injection, showing the beam intensities and only the positions of the primary (TCP) collimators for simplicity, from [75].

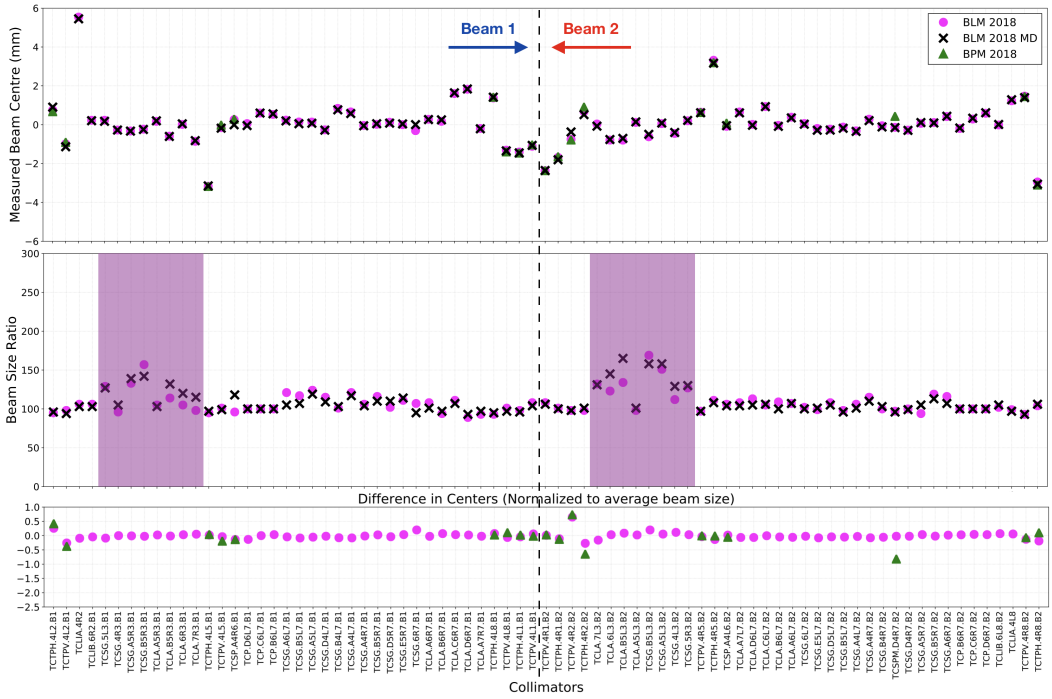


Figure 9.16: Parallel alignment MD results at injection, comparing the beam centres (top graph) and beam size ratios (middle graph) to the 2018 injection commissioning results using BLM and BPM devices. The difference in centres normalized by the average beam size is presented on the bottom graph, from [75].

10. Fully-Automatic Angular Alignment

The current operational settings for the betatron cleaning hierarchy requires a 1.5 σ retraction between primary and secondary collimators of the betatron cleaning insertion, which correspond to around 300 μm. In order to improve the LHC performance and achieve a smaller β^*, tighter collimator settings with smaller retractions are foreseen [104].

Until now, collimators aligned with BLM devices have always been aligned with a zero tilt angle with respect to the beam, however, recent beam tests indicate that this approach will not be adequate to operate the system with retractions below 1.5 σ [105]. Figure 10.1 shows a collimator i with an angular offset of α_i with respect to the beam axis, assuming that the two jaws are parallel in the collimator tank (ideal condition). Possible tank misalignments are a source of error that could jeopardize the performance of the system if not corrected. Other cases involving mounting errors within the tank can also occur, however these are less likely as the tank alignment is done with survey measurements and positioning systems which give an accuracy of 100 - 300 μm, whereas the collimator manufacturing is performed under much more stringent conditions. The proposed solution is to implement an automatic procedure to align collimators with different jaw tilts to

be able to determine the best angle.

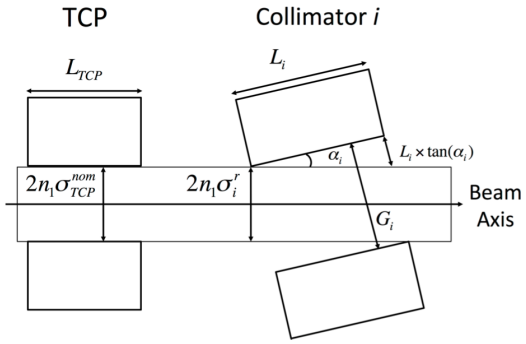


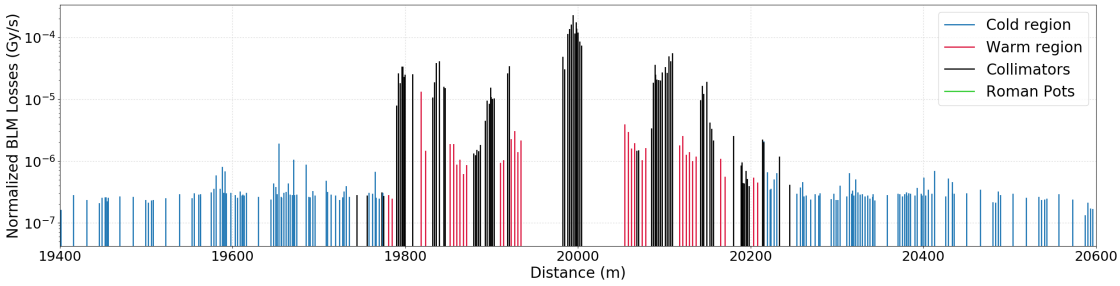
Figure 10.1: A case where collimator jaws are perfectly mounted with the vacuum tank, showing a tank misalignment w.r.t. the nominal orbit, from [25].

10.1 Use Case

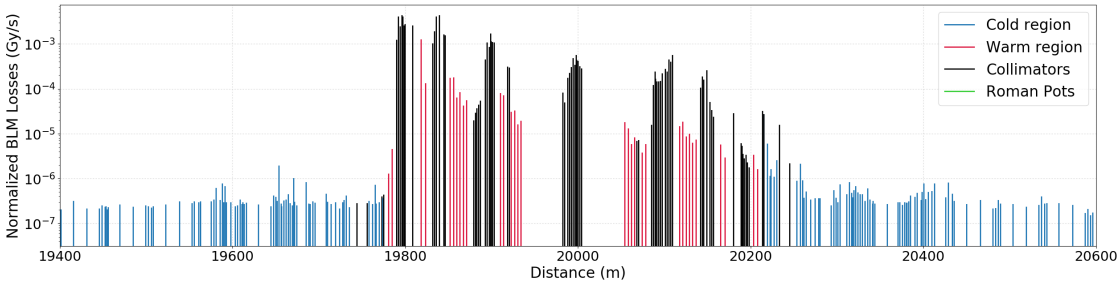
A use case example is that of the TCSG.D4L7.B1. In December 2017 the loss map displayed in Figure 10.2a was generated at flat top, showing a hierarchy breakage in IR7. This was caused by the TCSG.D4L7.B1, as it required an angle of –350 μrad. This angle was determined by performing an angular alignment using the semi-automatic tool available. This procedure required aligning one jaw corner at a time, which took around 40 minutes. Once the tilt was set, the loss map displayed in Figure 10.2b was generated, indicating the hierarchy is restored.

During commissioning in 2018, a similar breakage in the hierarchy was observed at flat top. This resulted in repeating the angular alignment for the TCSG.D4L7.B1, which indicated a –400 μrad tilt in the left jaw and a 320 μrad tilt in the right jaw. This is consistent with the previous angle found, and clearly indicates a hardware issue within the collimator tank.

As seen in this case, angular alignments are only performed when absolutely necessary, after a hierarchy breakage is observed, in order to save time. Therefore, it is possible that other collimators also require tilts, however they are not large enough to break the hierarchy, for now. This motivates having a fully-automatic



(a) Beam 1 vertical loss map indicating broken hierarchy in IR7.



(b) Beam 1 vertical loss map indicating correct hierarchy in IR7.

Figure 10.2: An example of a beam 1 betatron vertical loss map at 6.5 TeV, (a) before, and (b) after, tilting the TCSG.D4L7.B1 at the required angle.

procedure, to be able to detect any required tilts in advance and address the issue earlier, possibly even at the hardware level.

10.2 Angular Alignment Algorithms

Three novel algorithms were introduced which align collimator i at different angles, one jaw at a time, such that ultimately the angle which produces the best results could be determined and used in operation. Selecting which method to use depends on the specific error encountered and can therefore only be determined with more operational experience with angular setups.

10.2.1 METHOD 1: Angular alignment using reference collimator

This method is heavily based on the standard BBA procedure, explained in Section 2.3, which uses the primary collimators to create a reference halo for the collimators to align with. The main difference introduced is that collimator i is tilted at a pre-defined angle before alignment. To run this method, the user selects a collimator i, then sets the start and end angles (e.g.: α^a = 1000 μrad, α^z = –1000 μrad) and the angle step size (e.g.: δ = 100 μrad). The following procedure is then performed automatically:

1. Align the corresponding reference collimator and collimator i (angle: 0 μrad)
2. Tilt collimator i to α^a (angle: 1000 μrad), as shown in Figure 10.3.
3. Align collimator i at the angle set, then re-align the reference collimator (angle: 1000 μrad)
4. Tilt collimator i by δ (angle: 900 μrad)
5. Repeat steps 3 and 4 until the angle α^z is reached (final angle: -1000 μrad)
6. Tilt collimator i back to its initial angle (angle: 0 μrad)
7. Retract the reference collimator to its starting position

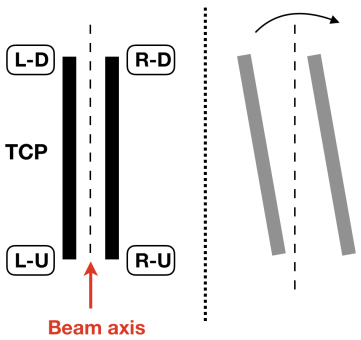


Figure 10.3: Collimator tilted at an angle with primary collimator used as reference.

This method generates the positions of both collimators after each alignment, at angles between α^a and α^z. As a result, the best angle can be extracted by

identifying the angle which generates the smallest measured beam σ. This method is ideal for cases where there is a real offset of the tank, as the requirement that the collimator jaws are parallel would be met.

10.2.2 METHOD 2: Angular alignment at maximum angles

This method does not require a reference collimator or any user inputs, it automatically applies the following steps on the selected collimator i:

1. Retract collimator i by 2 mm
2. Tilt at maximum angle (1900 μrad) to find the upstream centre as shown on the left in Figure 10.4
3. Align collimator i then retract by 2 mm
4. Tilt at opposite maximum angle (-1900 μrad) to find the downstream centre as shown on the right in Figure 10.4
5. Retract collimator i by 2 mm

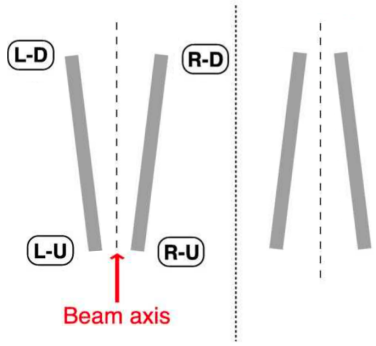


Figure 10.4: Jaws tilted at maximum angles, from [106].

This method generates the upstream and downstream centres of collimator i, which can be used to calculate the optimum angle. This is the quickest method out of the three as it simply requires two alignments to be performed.

10.2.3 METHOD 3: Angular alignment using a jaw as reference

This method does not require a reference collimator, as it treats the two jaws of collimator i separately, such that one jaw acts as the reference and the other jaw is tilted at different angles. Running this method requires the same parameters as the first method, whereby the user selects a collimator i, then sets the start and end angles (e.g.: α^{a} = 1000 μrad, α^{z} = –1000 μrad) and the angle step size (e.g.: δ = 100 μrad). The following procedure is then performed automatically:

1. Align the collimator i (collimator i angle: 0 μrad)
2. Retract the left jaw of collimator i and tilt it to α^{a} (left jaw angle: 1000 μrad) as shown on the left in Figure 10.5
3. Align the left jaw at the angle set (left jaw angle: 1000 μrad)
4. Align the right jaw as a reference (right jaw angle: 0 μrad)
5. Retract the left jaw and tilt it by δ (left jaw angle: 900 μrad)
6. Repeat the above three steps until the angle α^{z} is reached (left jaw final angle: -1000 μrad)
7. Tilt the left jaw back to its initial angle (collimator i angle: 0 μrad)
8. Retract collimator i by 1.2 mm
9. Repeat all steps 1 - 8, this time by tilting the right jaw and keeping the left jaw as a reference as shown on the right in Figure 10.5

This method generates the positions of both jaws after each alignment, at angles between α^{a} and α^{z}, for both the left and right jaws separately. As a result the most optimal angles for both jaws can be determined independently, making this method useful in cases of asymmetries within the collimator itself.

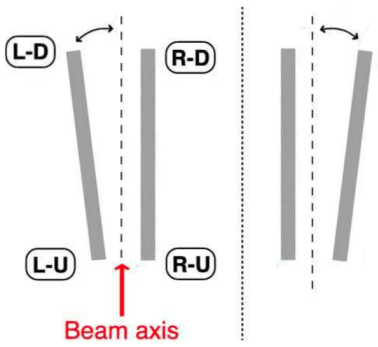


Figure 10.5: Single jaw at an angle with other jaw used as reference, from [106].

10.3 Calculations

For each alignment at an angle, the measured beam σ can be calculated, and the angle which results in the minimum value is the most optimal one. This is due to the fact that the angle resulting in the lowest measured beam σ implies that the collimator is the most parallel to the beam axis at the given angle, recall Figure 10.1.

The different methods produce different data, therefore the way the measured beam σ is calculated varies according to the method used.

- **Method 1**: This method is based on the fact that if there is a misalignment of collimator i with respect to the beam, the measured beam size of the collimator will be larger than the real one. This method gives the positions of collimator i aligned at each angle, as well as the position of the reference collimator aligned before and after collimator i. Therefore this method uses the BBA equations presented in Section 2.3, whereby the jaw gap is calculated using Equation 2.2, and the half gap is then used to calculate the measured beam size using Equation 2.3.

- **Method 2**: This method gives the left and right positions after aligning the upstream and downstream jaw corners separately. From these positions the upstream and downstream centres can be calculated using Equation 10.1, then the difference between the two positions using Equation 10.2 can be

converted to an angle using the small angle approximation.

$$\Delta x_i^{up} = \frac{x_i^{LU,m} + x_i^{RU,m}}{2}, \quad \Delta x_i^{down} = \frac{x_i^{LD,m} + x_i^{RD,m}}{2} \tag{10.1}$$

$$\vartheta = \Delta x_i^{down} - \Delta x_i^{up} \tag{10.2}$$

- **Method 3**: This method is based on the same concept of monitoring the measured beam size of collimator i as a function of the angle, this time using one jaw as a reference. This method gives the position of the aligned angular jaw at each angle and the position of the aligned reference jaw before and after. Equation 10.3 is then used to calculate the measured σ at each of the angles aligned.

$$\sigma_i^m = \frac{\Delta x_i^j - \Delta x_i^0}{(N\sigma_{jaw_1} + M\sigma_{jaw_2})/2}, \tag{10.3}$$

where Δx_i^j is the average of the up and down positions of the aligned jaw at angle j, in millimetres, Δx_i^0 is the centre of the collimator jaws when aligned with zero angle, in millimetres, $N\sigma_{jaw_1}$ is the $N\sigma$ of the reference jaw before aligning the angular jaw and $M\sigma_{jaw_2}$ is the $M\sigma$ of the reference jaw after aligning the angular jaw.

10.4 Software Architecture

The angular alignment tool was first implemented by automating the semi-automatic BBA, by selecting a high BLM threshold and keeping it constant, and assuming that any losses that exceeded such a high threshold were a result of the collimator touching the beam. The aim of this implementation was to test the angular alignment algorithms and understand whether the optimal angle of collimators can be

found using each of the methods.

Following the implementation of the dedicated algorithms to replace the user tasks, the angular semi-automatic BBA was then transformed into a fully-automatic one. The angular fully-automatic BBA tool requires a number of inputs to execute:

- Collimator name - The name of collimator i to be aligned at multiple angles.
- Starting angle (α^a) - The first angle in μrad, of the angular range to be explored.
- Ending angle (α^z) - The last angle in μrad, of the angular range to be explored.
- Angle step size (δ) - The step size in μrad, at which collimator i will be tilted between each alignment.
- Operation - A flag that identifies which of the three methods to execute.

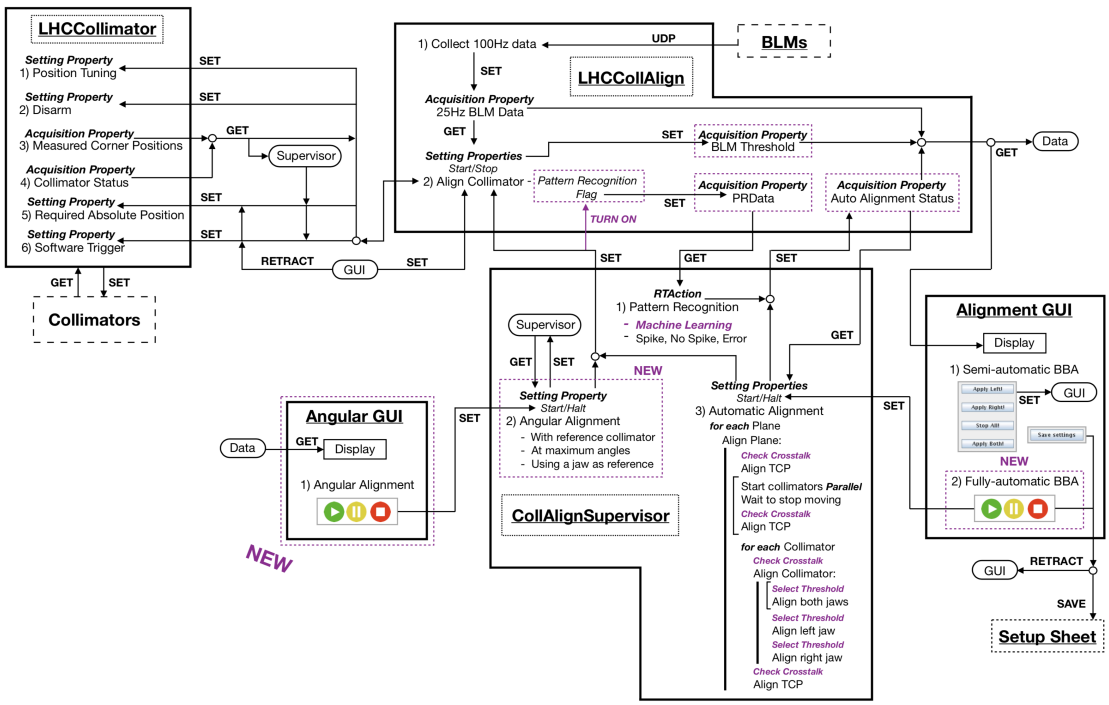


Figure 10.6: Overview of fully-automatic Angular BBA implementation in FESA and Java, from [10].

The angular fully-automatic BBA is implemented in the same *CollAlignSupervisor* FESA class, as the standard fully-automatic BBA explained in Section 8.3). This software is built on top of the semi-automatic BBA and provides the same functionality as the standard fully-automatic BBA, as depicted in Figure 10.6. The main difference is that the angular fully-automatic BBA is controlled from a separate application, however it preserves the same play, pause and stop functionality.

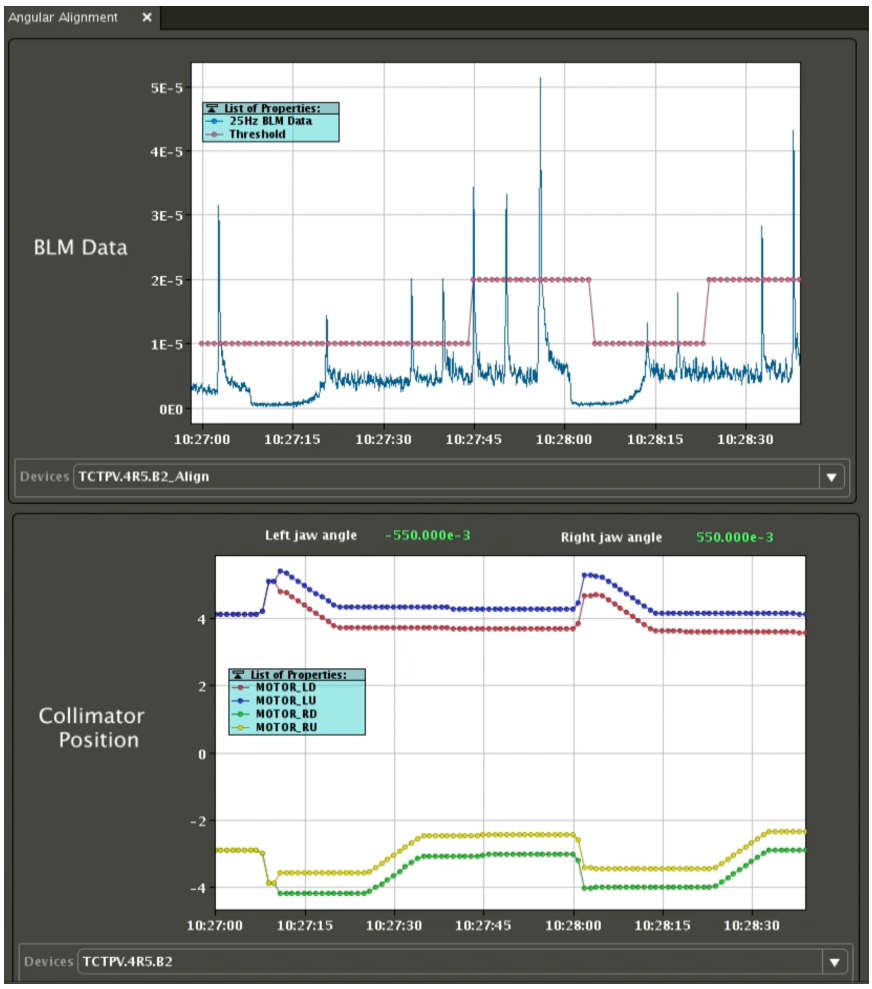


Figure 10.7: A screen shot of the GUI designed for the angular fully-automatic BBA, such that the top part monitors the 25 Hz BLM data and the automatic selection of the threshold, whilst the bottom part monitors the 1 Hz positions and angles of the collimator jaws, from [10].

A standalone GUI was designed for the angular fully-automatic BBA which serves solely as a display, as shown in Figure 10.7. In this case, all commands must be provided directly to FESA to control the angular alignments. This was designed as a temporary solution to monitor the status of the collimators aligned at different angles when performing beam tests. A more user-friendly version of this tool will eventually be incorporated with the rest of the collimation controls. Figure 10.7 shows the TCTPV.4R5.B2 being aligned using Method 1, and one can clearly see:

- The threshold selection algorithm automatically increases and decreases the threshold as required.
- The clear spikes generated when aligning the collimator.
- The BLM losses resetting back to a steady state with each collimator retraction, before aligning with a new angle.

Finally, similar to the standard fully-automatic BBA, multiple instances of the same application can be opened at the same time to align collimators in parallel. At present this is open for the user to select any collimators, as the parallel selection using the cross-talk model has not yet been incorporated in this tool.

10.5 Software Validation in LHC Operation

The three methods were tested with beam at the SPS and the LHC [107], and compared to BPM results where available.

10.6 SPS Results

The methods were first tested using the semi-automatic alignment in the SPS, using coast beam at 270 GeV, with one LHC-type nominal bunch having an intensity of 1.1×10^{11} p. The BPM prototype collimator in the SPS, TCSM.51934, was

aligned using each method and the results obtained are displayed in Figure 10.8. The results are normalized using:

$$\sigma_i^{norm} = \frac{\sigma_i^m - \min(\sigma_i^m)}{\max(\sigma_i^m) - \min(\sigma_i^m)} \tag{10.4}$$

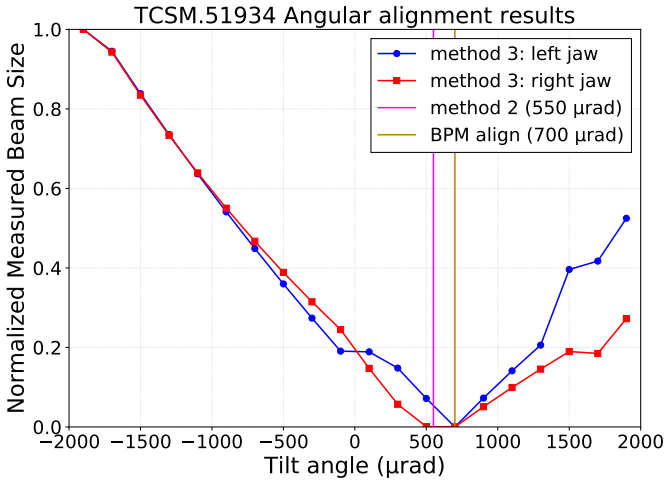


Figure 10.8: The normalized measured beam σ for different tilt angles for the SPS beam test with the TCSM.51934 collimator, from [106].

Since the SPS has only one LHC-type collimator where the position of both jaws is known with respect to a common zero position (i.e. the centre of the beam pipe), the method which requires a reference collimator had to be done using a scraper instead. The scraper is another device for scraping beam and has a different reference system. Its jaw position is the absolute one starting from zero at parking and therefore it was not possible to calculate the gap between the two jaws, which is required to calculate $N\sigma$ in Equation 2.3. The results using methods 2 and 3 indicate that the most optimal angle is approximately 600 µrad.

10.7 LHC Results

10.7.1 Semi-Automatic Angular Results

The three methods were then executed in the LHC at injection [107] using one nominal bunch of an intensity of 1.1×10^{11} p, in beam 2. The BPM collimators were first aligned to quickly determine the collimator with the largest tilt angle. The angles found for each collimator are displayed in Table 10.1. The angular range explored for each collimator was between –800 μrad and 800 μrad, and an angle step size of 100 μrad was used. The step size of each jaw movement was set to 5 μm, at a time interval of 0.1 s.

Table 10.1: Angles obtained from BPM collimator alignments in beam 2.

Collimator	**Angle (μm)**
TCSP.A4L6.B2	-345
TCTPH.4R1.B2	50
TCTPH.4R2.B2	**-490**
TCTPH.4R5.B2	55
TCTPH.4R8.B2	-165
TCTPV.4R1.B2	63
TCTPV.4R2.B2	133
TCTPV.4R5.B2	-245
TCTPV.4R8.B2	-25

The TCTPH.4R2.B2 which is a tertiary horizontal collimator in IP2 beam 2, was found to have the largest tilt, therefore this was aligned using the three angular alignment methods. The movements of each of the jaw corners during the alignments are shown in Figure 10.9, in order to better visualise the different methods. The times taken for each of the methods were: 22 minutes for method 1, 5 minutes for method 2 and 15 minutes for method 3.

When aligning the right jaw using method 3 one can observe a large number of misalignments, whereby the jaw was stopped too early due to non-alignment spikes. This issue comes from using the semi-automatic tool without spike recognition,

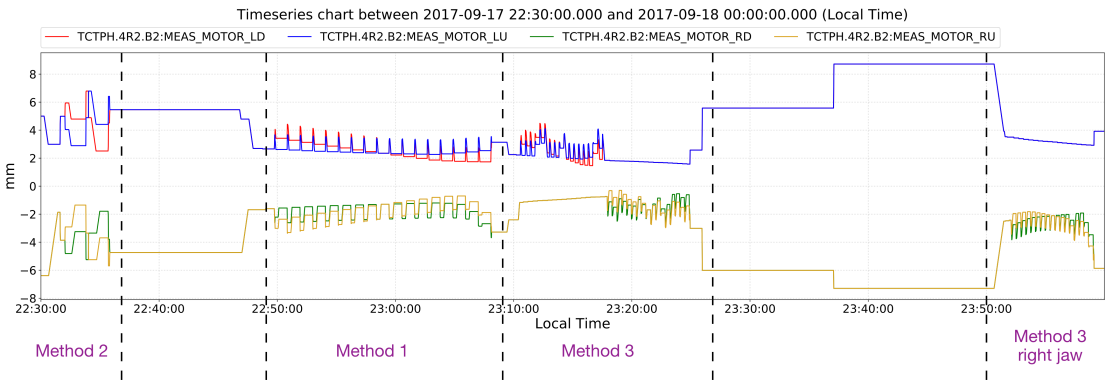


Figure 10.9: TCTPH.4R2.B2 angular alignment movements.

therefore in this case the right jaw alignment was performed a second time. The second alignment was then used to display the results in Figure 10.10. The results using each of the methods indicate that the most optimal angle is approximately -450 µrad, and the misalignments are included in the plot as outliers.

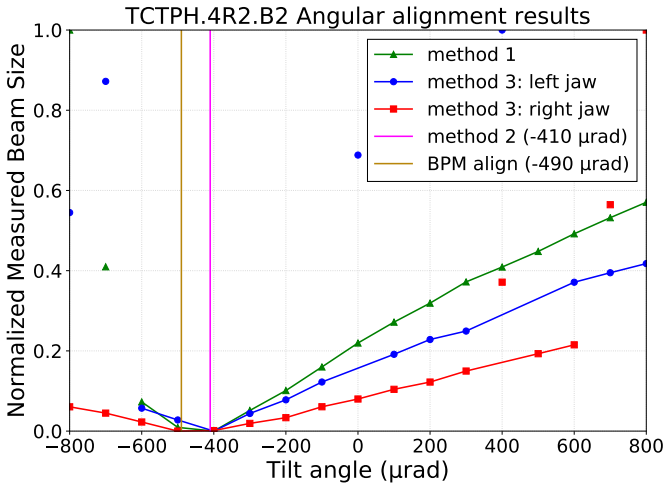


Figure 10.10: TCTPH.4R2.B2 results using the three methods and BPMs, from [106].

As expected, such misalignments were also encountered when aligning other collimators. This was especially an issue when aligning the TCTPV.4R5.B2 at different angles. Figure 10.11 shows the movements of each jaw corner during the angular alignments, such that only methods 1 and 2 were applied due to the large number of misalignments. The results in Figure 10.12 further highlights the

inconsistent results when attempting to automate the semi-automatic tool.

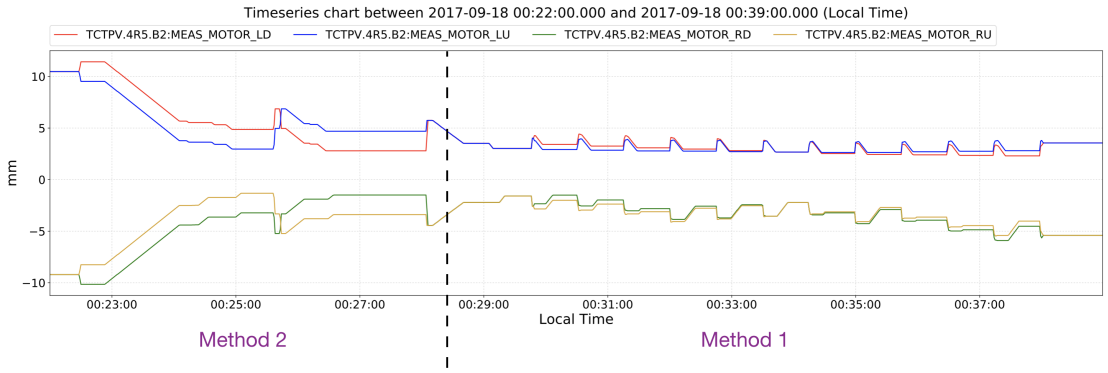


Figure 10.11: TCTPV.4R5.B2 angular alignment movements.

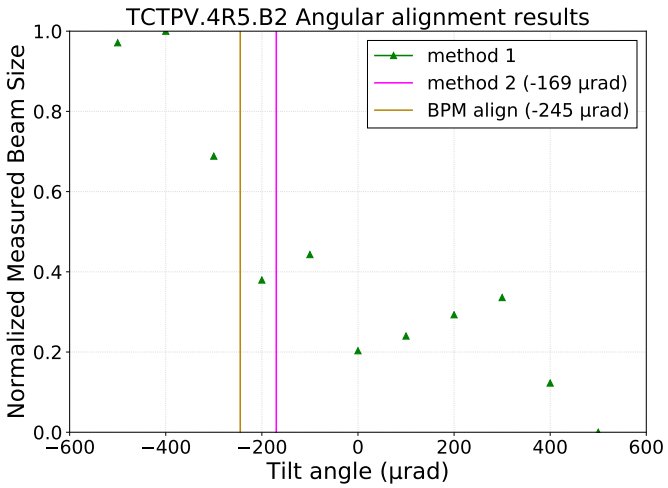


Figure 10.12: TCTPV.4R5.B2 results using methods 1, 2 and BPMs.

Overall, the results indicate that all three methods are able to converge to the same angle, therefore the most optimal angle can indeed be obtained using any of the methods. However, a major issue observed is the need for a fully-automatic tool able to automatically determine whether a collimator is aligned or not using spike recognition, and to automatically update the threshold before each alignment.

10.7.2 Fully-Automatic Angular Results

As explained in Section 10.4, the semi-automatic angular alignment was also fully-automated using the automatic threshold selection algorithm and spike recognition

models. This fully-automatic angular software was tested in the LHC at injection [108] using one nominal bunch of an intensity of 1.1×10^{11} p, in each beam. The collimators were first aligned with BPMs, then an angular range between –600 µrad and 0 µrad was explored, using an angle step size of 50 µrad. The step size of each jaw movement was kept at 5 µm, as the losses generated with 10 µm steps were not conservative of the beam. In addition, the alignments during commissioning 2018 (recall Section 9.3) showed that the time interval can be decreased to 0.02 s, thus speeding up the angular alignment.

The TCTPH.4R2.B2 was re-aligned at different angles using this tool, such that the movements of each of the jaw corners are shown in Figure 10.13. The results in Figure 10.14 show that the most optimal angle is approximately –370 µrad, indicating that there is a real offset in the collimator tank.

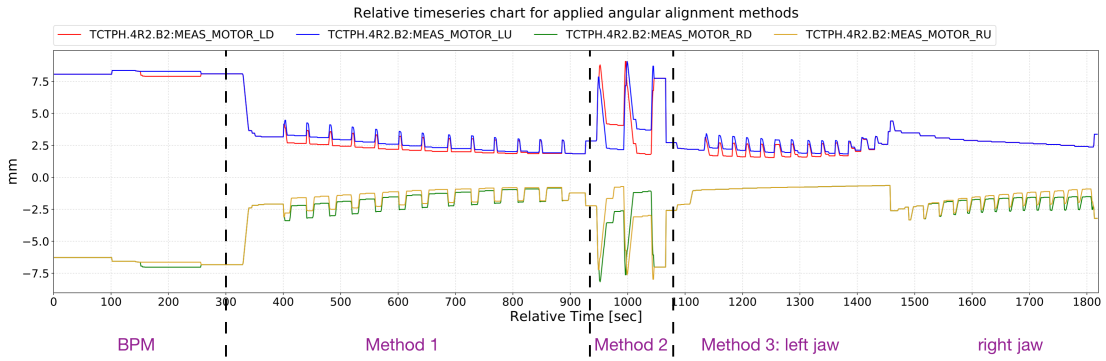


Figure 10.13: TCTPH.4R2.B2 alignment movements.

The TCTPV.4R5.B2 was then re-aligned and its alignment movements are shown in Figure 10.15. The results obtained are shown in Figure 10.16 showing that the three angular alignment methods perfectly converged at –200 µrad, whilst the alignment with BPMs indicated an optimal angle of –350 µrad. These results highlight the consistency, reliability and efficiency provided by having a fully-automatic alignment tool, in comparison to the semi-automatic approach.

Finally, these three methods were performed on the TCSG.D4L7.B1 which is known to have a tilt (recall Section 10.1), such that its alignments are shown in Figure 10.17. The results are shown in Figure 10.18, indicating that the most

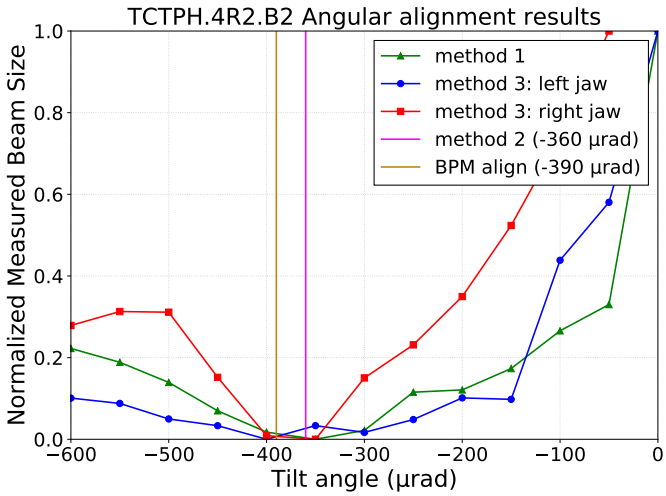


Figure 10.14: TCTPH.4R2.B2 results using the 3 methods and BPMs, from [75].

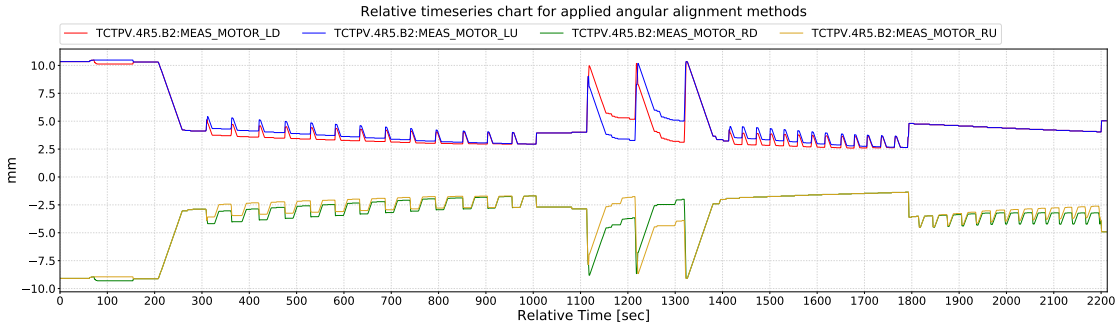


Figure 10.15: TCTPV.4R5.B2 alignment movements.

optimal angle for the right jaw is 300 μrad, whilst the most optimal angle for the left jaw is –500 μrad. This suggests an asymmetry within the collimator itself, which is consistent with previous results.

Note: According to the collimation angular tilt convention (recall Figure 2.1b), the right jaw angle is the negation of the left jaw angle when the jaws are parallel, however in order to plot them together to determine a single optimal angle, the inverse of the right jaw angles are plotted throughout this work. In the case of the TCSG.D4L7.B1 the right jaw needs to be taken individually thus when using the collimation angular tilt convention the most optimal right jaw angle is actually around 300 μrad not -300 μrad as may be understood from Figure 10.18.

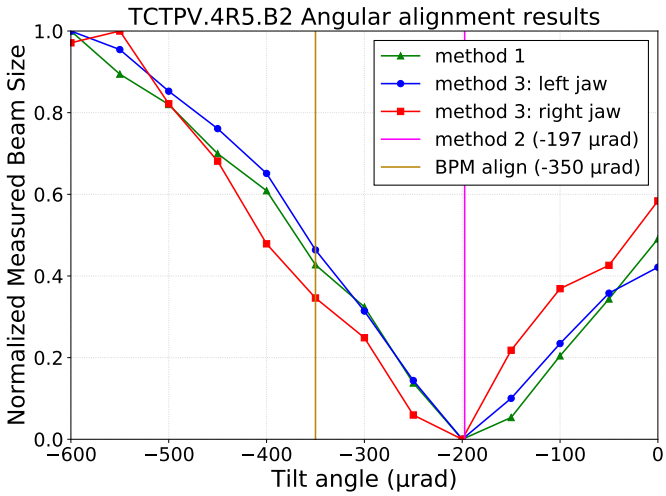


Figure 10.16: TCTPV.4R5.B2 results using the 3 methods and BPMs.

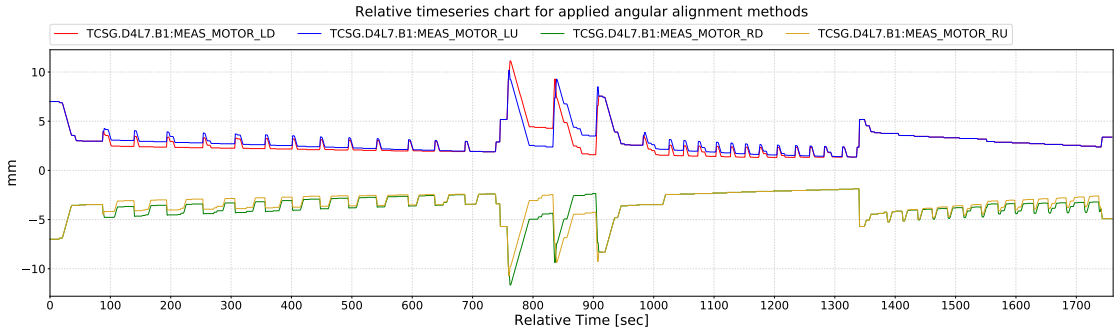


Figure 10.17: TCSG.D4L7.B1 alignment movements.

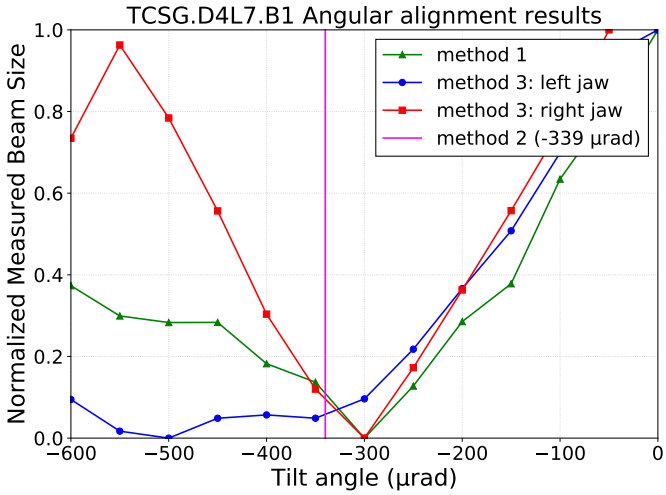


Figure 10.18: TCSG.D4L7.B1 results using the 3 methods, from [75].

The times taken to determine the most optimal angle at each collimator using the three different angular alignment methods are compared in Figure 10.19. These times indicate that methods 1 and 3 require a similar amount of time at around 13 minutes on average, whilst method 2 requires 3 minutes on average.

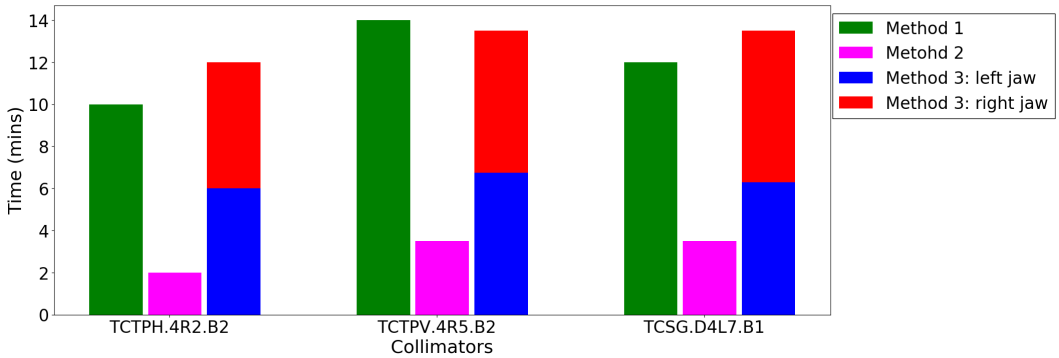


Figure 10.19: The time taken to find the optimal angle of various collimators using the three angular alignment methods.

Overall, the fully-automatic tool aligned collimators at different angles with minimal misalignments, and the angles obtained are consistent with previous alignments. These results indicate that the new angular alignment software is reliable and the results are reproducible.

Finally, this new tool allows for fully-automatically aligning a collimator at 41 different angles using 3 different methods in 25 minutes. This is 70% faster in comparison to the previous approach used in Section 10.1. This software will be used in the near future, to achieve a tighter hierarchy.

11. Conclusion

11.1 Summary

The CERN Large Hadron Collider (LHC) is the world's largest particle accelerator, built to accelerate and collide two counter-rotating beams, each currently having a nominal energy of 6.5 TeV. The LHC is equipped with a complex collimation system consisting of around 100 movable collimators, to protect sensitive equipment from unavoidable beam losses.

The collimators are aligned to the beam to calculate beam parameters, such as the beam centres and beam sizes at the collimator locations. The operational settings are determined using this beam-based alignment procedure, whereby the jaws are moved towards the beam until an alignment spike is observed in the signal of the BLM immediately downstream from the collimator.

Since 2011, beam-based alignments are performed semi-automatically such that a collimator automatically moves towards the beam and when its BLM losses exceed a predefined threshold, the collimator automatically stops moving. At this point a collimation expert must be present to analyse the BLM losses and determine if the collimator is aligned or not, and proceed accordingly. Therefore the user is responsible for; selecting the threshold, starting the alignment, analysing the BLM losses when the collimator stops moving, and restarting the alignment as

many times as required. As a result collimation experts must be present during the whole period of the alignment, therefore the frequency at which alignments can be performed is limited.

In addition to this, collimators have always been aligned assuming no tilt between the collimator and the beam, however, recent beam tests indicate that this approach will not be adequate to operate the system with tighter collimation settings. This is due to the fact that any tank misalignments or beam envelope angles at large-divergence locations could introduce a tilt, and at present this is solved by aligning collimators with a wider gap between the jaws. Therefore in order to tighten the collimator jaws, tilts must be introduced to counteract the tilts introduced by imperfections in the systems in the machine.

The aim of this dissertation is to implement software able to fully-automatically align all collimators without requiring any human intervention during the process. This can be done by replacing the three main user tasks; cross-talk analysis, spike detection and threshold selection, to close the loop between BLM feedback and decision making.

In order to automatically align a collimator, pattern recognition needs to be applied to detect spikes in the BLM signal, to determine whether a collimator is aligned or not. A machine learning model was implemented to automatically classify between the two classes of temporal beam loss patterns. Past alignment campaigns were studied by extracting all collimator alignments performed, to form a data set. Fourteen features were initially extracted from this data set, which were then analysed in detail. Six machine learning models were trained and tested, whilst selecting their best features and hyper-parameters, and achieved a precision of over 95%. Robustness testing was then applied to each of the models, and confirmed that machine learning is indeed reliable to be used during LHC operation.

Automatically aligning a collimator involves automatically selecting the BLM loss threshold, such that the collimator stops moving when its losses exceed this threshold. An algorithm was designed to calculate the exponentially weighted

moving root mean square (EWMRMS) on fixed length windows of data. The thresholds selected by the algorithm were compared to thresholds selected by users during alignment campaigns in the past. The results indicate that the difference between the two thresholds was negligible and that the algorithm selected suitable thresholds for all cases. Therefore this algorithm managed to achieve the correct balance between selecting a low threshold to decrease beam losses and selecting a high threshold to avoid stopping at noisy spikes.

Collimators are selected to align in parallel based on the cross-talk observed across their corresponding BLM devices. This cross-talk was studied to provide an initial solution for automatically aligning collimators, based on data gathered from previous sequential alignments. Furthermore, beam tests were performed to seek to quantify the levels of cross-talk observed by BLM devices, to determine the actual beam loss signals generated by their corresponding collimators. The cross-talk was successfully predicted with an error less than 10% of the measured BLM signal, thus providing a foundation for future analysis.

These three dedicated algorithms are used to transform the semi-automatic beam-based alignment into a fully-automatic one. The alignment software is developed in the real-time C/C++ FESA environment, having the semi-automatic and fully-automatic alignments implemented in their own FESA class and executing on separate FECs. The new fully-automatic FESA class automatically selects collimators to align in parallel based on the cross-talk analysis. For both collimators it then selects a threshold and starts the same beam-based alignment used for the semi-automatic alignment. Once the selected collimator stops moving, control is returned to the new FESA class to apply machine learning to automatically detect whether the collimator is aligned or not. The new FESA class then automatically continues the alignment, without requiring any human intervention. The new fully-automatic alignment was designed to work on top of the semi-automatic alignment, whilst allowing both tools to be available for use, whereby the fully-automatic BBA GUI simply extends the semi-automatic BBA GUI.

The new fully-automatic software was used in all collimator alignments throughout 2018. The first version was used during commissioning, such that the collimators in the two beams were automatically aligned sequentially, at injection and flat top. A few months later, an improved version of the software was used to align the collimators in the two beams in parallel at injection, successfully decreasing the 2017 alignment time by 71.4%.

An additional alignment technique that can significantly benefit from a fully-automatic approach is the angular alignment. Three novel angular alignment methods were introduced to align the same collimator at different angles to determine the best angle for operational use. The angular fully-automatic BBA is implemented in the same FESA class, as the standard fully-automatic BBA and provides the same functionality. The implementation successfully decreased the alignment time by 70%, no human intervention was required for the alignments, and the most optimal angle can indeed be obtained using any of the three methods. The angular BBA is controlled from a separate application and a separate interface was temporarily designed to monitor the status.

Overall, the fully-automatic software significantly decreased the time required for both standard and angular alignments, and successfully generates reproducible results without requiring any human intervention during the process. This full automation is a major step in enhancing operational efficiency and will be used as the default software for starting the LHC in 2021.

11.2 Suggestions for Future Work

In view of the envisaged LHC re-start in 2021, a few suggestions for further work are provided. The cross-talk analysis performed provides a stable foundation for expanding the analysis by taking measurements at different energies and using more collimators, especially skew collimators, to analyse any energy dependent effects. Furthermore, once this analysis is more conclusive and covers the full range of

collimators, this can be incorporated with the full automation to allow for more collimators to be aligned in parallel.

The GUI of the standard fully-automatic BBA can be upgraded by providing an additional panel to give users the option of more advanced control of the alignment. This panel can provide additional control of; number of spikes required to conclude the alignment, decay waiting time to apply the machine learning model, choice of moving all collimators in a plane in parallel, etc. Moreover an additional tab can be introduced, to allow for fully-automatically aligning a single collimator in a more straightforward manner.

On the other hand, a new GUI can be designed for the angular fully-automatic BBA such that it can be incorporated with the rest of the collimation controls. A new panel can be designed to allow for multiple collimators to be selected for angular alignments, which is only now possible following the work done for this dissertation. Moreover, this will allow to introduce fully-automatic parallel angular alignments, by making use of the cross-talk model presented.

The full-automation can be extended to also automatically analyse the beam-based settings measured during the alignment, to determine if any collimators needs to be re-aligned, and automatically re-align them. Moreover, the settings could also be stored directly in LSA database tables, rather than locally on disk, making it easier to import the measured settings directly into the beam processes. Finally, this automated setup for collimators can be extended to any movable device in an accelerator, which requires a beam-based alignment using BLMs.

A. Cross-talk Factors

This section gathers, in Table A.1, cross-talk factors measured from beam tests, which quantify the effect of a closed collimator on all downstream collimators.

Table A.1: BLM cross-talk factors calculated from the individual cases during LHC beam tests, from [16].

Case ID	**Effected Collimators**					
	B1V-1	**B1V-2**	**B1V-3**	**B1V-4**		
B1V-1	1.0	0.00523	0.00402	0.00016		
B1V-2	-	1.0	0.01522	0.00064		
B1V-3	-	-	1.0	0.01340		
B1V-4	-	-	-	1.0		
	B1H-1	**B1H-2**	**B1H-3**	**B1H-4**	**B1H-5**	**B1H-6**
B1H-1	1.0	0.00093	0.014584	0.00094	0.00039	0.00018
B1H-2	-	1.0	0.013217	0.00131	0.00284	0.00022
B1H-3	-	-	1.0	0.05542	0.02536	0.00730
B1H-4	-	-	-	1.0	0.02127	0.00831
B1H-5	-	-	-	-	1.0	0.01688
B1H-6	-	-	-	-	-	1.0
	B2V-1	**B2V-2**	**B2V-3**	**B2V-4**	**B2V-5**	
B2V-1	1.0	0.0052	0.00304	0.00257	0.00016	
B2V-2	-	1.0	0.61890	0.00761	0.00020	
B2V-3	-	-	1.0	0.00979	0.00030	
B2V-4	-	-	-	1.0	0.01237	
B2V-5	-	-	-	-	1.0	
	B2H-1	**B2H-2**	**B2H-3**	**B2H-4**	**B2H-5**	**B2H-6**
B2H-1	1.0	0.00105	0.01836	0.00122	0.00039	0.00018
B2H-2	-	1.0	0.01432	0.00165	0.00217	0.00033
B2H-3	-	-	1.0	0.08164	0.03403	0.01557
B2H-4	-	-	-	1.0	0.02399	0.01535
B2H-5	-	-	-	-	1.0	0.02145
B2H-6	-	-	-	-	-	1.0

B. Measured and Simulated Impacts

This section gathers, in Tables B.1 and B.2, the percentage distribution of proton impacts that were measured from beam tests, and obtained from Sixtrack simulation, respectively. Due to the shielding of upstream collimators assumed to be zero, the proton impacts at all downstream collimators are overestimated, thus underestimating the percentage of impacts at the primary collimators.

Table B.1: Measured proton impacts percentages calculated from the combined cases of closed collimators, from [16].

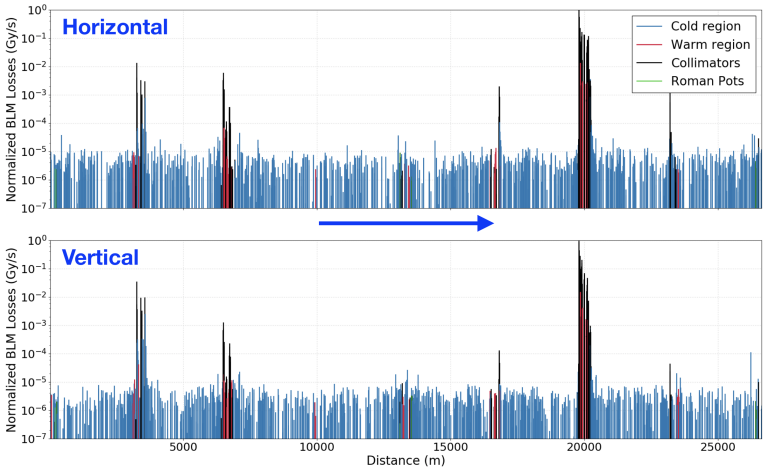
Case ID	**Measured Collimator Impacts**					
	B1V-1	**B1V-2**	**B1V-3**	**B1V-4**		
B1V - 1+2	0.91244	0.08756	-	-		
B1V - 1+2+3	0.91201	0.08706	0.00092	-		
B1V - 1+...+4	0.91121	0.08704	0.00096	0.00078		
	B1H-1	**B1H-2**	**B1H-3**	**B1H-4**	**B1H-5**	**B1H-6**
B1H - 1+2	0.78973	0.21027	-	-	-	-
B1H - 1+2+3	0.74165	0.17754	0.08081	-	-	-
B1H - 1+...+4	0.74997	0.17338	0.07406	0.00258	-	-
B1H - 1+...+5	0.74876	0.17105	0.07724	0.00261	0.00033	-
B1H - 1+...+6	0.75561	0.16990	0.07141	0.00224	0.00042	0.00042
	B2V-1	**B2V-2**	**B2V-3**	**B2V-4**	**B2V-5**	
B2V - 1+2	0.88429	0.11571	-	-	-	
B2V - 1+2+3	0.82250	0.11091	0.06659	-	-	
B2V - 1+...+4	0.82752	0.10763	0.06468	0.00017	-	
B2V - 1+...+5	0.82376	0.10759	0.06489	0.00365	0.00011	
	B2H-1	**B2H-2**	**B2H-3**	**B2H-4**	**B2H-5**	**B2H-6**
B2H - 1+2	0.75050	0.24950	-	-	-	-
B2H - 1+2+3	0.70668	0.15890	0.13441	-	-	-
B2H - 1+...+4	0.70038	0.16475	0.13472	0.00015	-	-
B2H - 1+...+5	0.69905	0.16281	0.13738	0.00076	0	-
B2H - 1+...+6	0.69675	0.16309	0.13939	0.00058	0	0.00019

Table B.2: Simulated proton impacts percentages calculated from the combined cases of closed collimators, from [16].

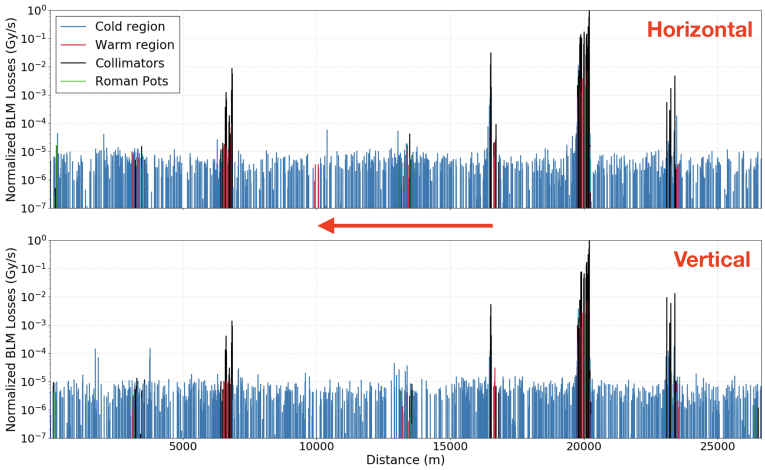
Case ID	**Simulated Collimator Impacts**					
	B1V-1	**B1V-2**	**B1V-3**	**B1V-4**		
B1V - 1+2	0.94446	0.05552	-	-		
B1V - 1+2+3	0.94420	0.05530	0.00049	-		
B1V - 1+...+4	0.94395	0.05541	0.00048	0.00016		
	B1H-1	**B1H-2**	**B1H-3**	**B1H-4**	**B1H-5**	**B1H-6**
B1H - 1+2	0.88632	0.10943	-	-	-	-
B1H - 1+2+3	0.85836	0.07169	0.06919	-	-	-
B1H - 1+...+4	0.85697	0.07164	0.06909	0.00216	-	-
B1H - 1+...+5	0.85627	0.07161	0.06894	0.00217	0.00102	-
B1H - 1+...+6	0.85592	0.07154	0.06897	0.00214	0.00101	0.00042
	B2V-1	**B2V-2**	**B2V-3**	**B2V-4**	**B2V-5**	
B2V - 1+2	0.93951	0.06049	-	-	-	
B2V - 1+2+3	0.93477	0.06004	0.00519	-	-	
B2V - 1+...+4	0.93450	0.06001	0.00518	0.00031	-	
B2V - 1+...+5	0.93447	0.06001	0.00519	0.00032	0.000004	
	B2H-1	**B2H-2**	**B2H-3**	**B2H-4**	**B2H-5**	**B2H-6**
B2H - 1+2	0.89701	0.09678	-	-	-	-
B2H - 1+2+3	0.87597	0.06427	0.05878	-	-	-
B2H - 1+...+4	0.87511	0.06415	0.05864	0.001838	-	-
B2H - 1+...+5	0.87464	0.06415	0.05877	0.001829	0.00079	-
B2H - 1+...+6	0.87426	0.06418	0.05858	0.001832	0.00079	0.00035

C. Commissioning Loss maps

The alignment campaigns during commissioning at the start of each year, are followed by loss maps to validate the collimation hierarchy defined by the alignment. The loss maps generated after injection and flat top commissioning in 2017 are displayed in Figures C.1 and C.2, respectively. Similarly, the loss maps generated after injection and flat top commissioning in 2018 are displayed in Figures C.3 and C.4, respectively. Loss maps are individually generated for the horizontal and vertical planes in each beam. One can see that in each case the collimators absorb the majority of the losses and these losses follow the direction of the beam, indicating the correct hierarchy.

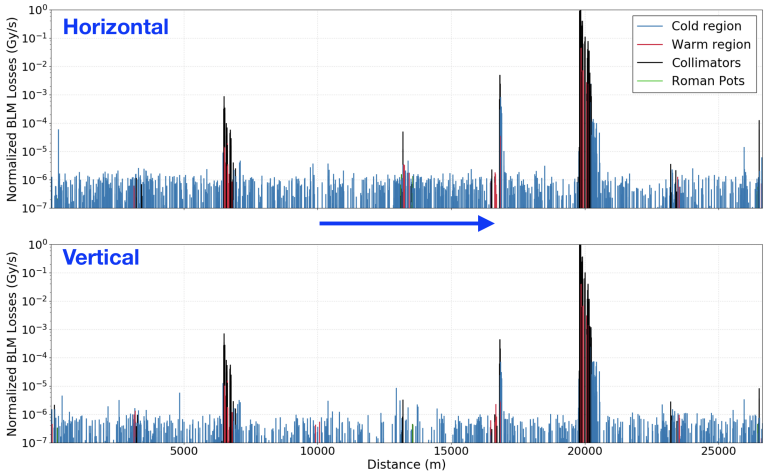


(a) Beam 1 horizontal and vertical loss maps.

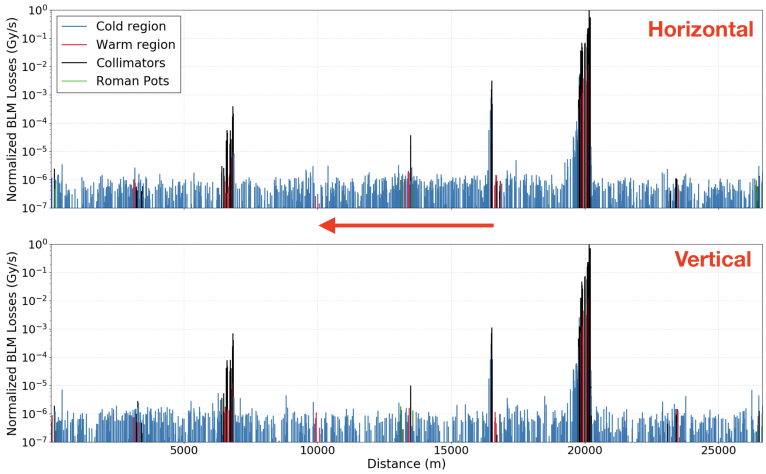


(b) Beam 2 horizontal and vertical loss maps.

Figure C.1: Loss maps generated on 13-04-2018 to validate the BLM collimator alignment at injection for (a) beam 1, and (b) beam 2, indicating the beam direction by the arrow in each case.

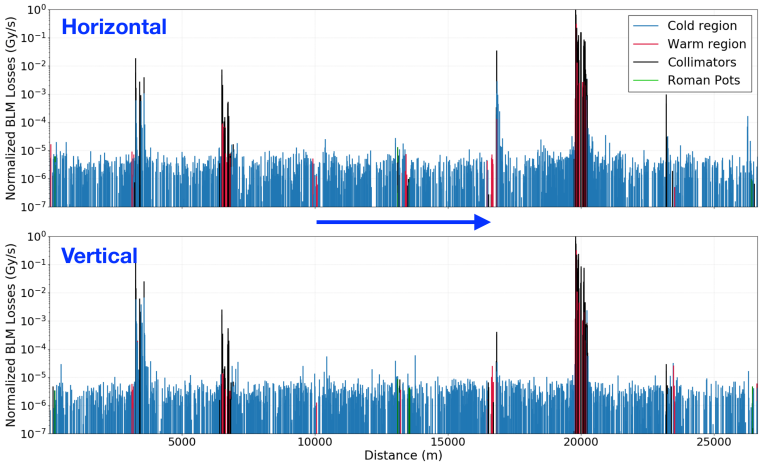


(a) Beam 1 horizontal and vertical loss maps.

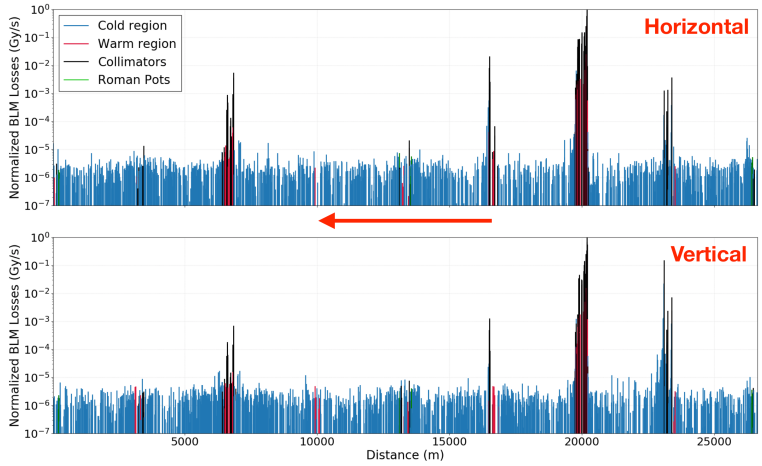


(b) Beam 2 horizontal and vertical loss maps.

Figure C.2: Loss maps generated on 13-04-2018 to validate the BLM collimator alignment at flat top for (a) beam 1, and (b) beam 2, indicating the beam direction by the arrow in each case.

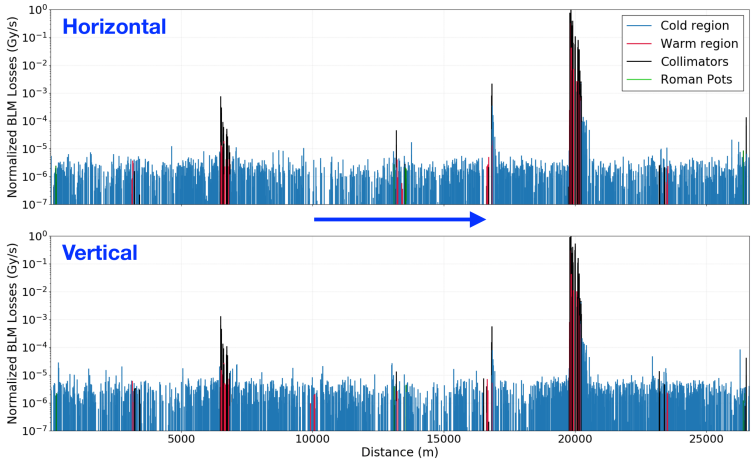


(a) Beam 1 horizontal and vertical loss maps.

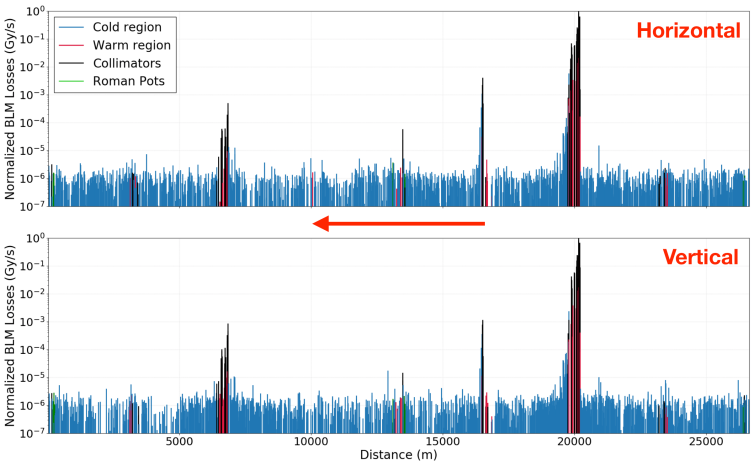


(b) Beam 2 horizontal and vertical loss maps.

Figure C.3: Loss maps generated on 13-04-2018 to validate the BLM collimator alignment at injection for (a) beam 1, and (b) beam 2, indicating the beam direction by the arrow in each case, from [75].



(a) Beam 1 horizontal and vertical loss maps.



(b) Beam 2 horizontal and vertical loss maps.

Figure C.4: Loss maps generated on 13-04-2018 to validate the BLM collimator alignment at flat top for (a) beam 1, and (b) beam 2, indicating the beam direction by the arrow in each case, from [75].

References

[1] L. R. Evans. The Large Hadron Collider. *New Journal of Physics*, 9(9):335, 2007.

[2] L. R. Evans. Beam physics at LHC. In *Proceedings of the 2003 Particle Accelerator Conference, Portland, OR*, volume 1, pages 19–23. IEEE, 2003.

[3] G. Valentino. *Fast automatic beam-based alignment of the LHC collimator jaws*. PhD thesis, University of Malta, 2013.

[4] R. W. Aßmann et al. Requirements for the LHC collimation system. Technical report, CERN, 2002.

[5] C. Bracco. *Commissioning Scenarios and Tests for the LHC Collimation system*. PhD thesis, Ecole Polytechnique Federale de Lausanne, 2009.

[6] Oliver Brüning. *LHC Design Report: The LHC Main Ring*, volume 1. European Organization for Nuclear Research, 2004.

[7] M. Lamont. The LHC from commissioning to operation. In *Proceedings of IPAC2011, San Sebastian, Spain*, pages 11–15, 2011.

[8] S. Redaelli et al. Operational performance of the LHC collimation. In *HB2010, Morschach, Switzerland*, pages 395–399, 2010.

[9] D. Missiaen, R. J. Steinhagen, and J. P. Quesnel. The alignment of the LHC. In *Proceedings of IPAC2009, Vancouver, Canada*, 2009.

[10] G. Azzopardi et al. Software Architecture for Automatic LHC Collimator Alignment using Machine Learning. In *Proceedings of ICALEPCS2019, New York, NY, USA (Accepted for publication)*, 2019.

[11] A. Guerrero et al. CERN front-end software architecture for accelerator controls. In *Proceedings of ICALEPCS2003, Gyeongju, Korea*, pages 342–344, 2003.

[12] A Masi, R Losito, and S Redaelli. Measured performance of the LHC collimator low-level control system. In *Proceedings of ICALEPCS2009, Kobe, Japan*, pages 612–614, 2009.

[13] V. Baggiolini et al. JAPC-the Java API for Parameter Control. In *Proceedings of ICALEPCS2005, Geneva, Switzerland*, volume 5, 2005.

[14] E. Holzer et al. Beam loss monitoring system for the LHC. In *Nuclear Science Symposium Conference Record, 2005 IEEE*, volume 2, pages 1052–1056. IEEE, 2005.

[15] M. Stockner et al. Classification of the LHC BLM ionization chamber. In *Proceedings of DIPAC2007, Venice, Italy*, pages 328–330, 2007.

[16] G. Azzopardi, B. M. Salvachua Ferrando, and G. Valentino. Data-driven cross-talk modeling of beam losses in LHC collimators. *Physical Review Accelerators and Beams*, 22(8), 2019.

[17] M. D'Andrea. Review of BLM thresholds at tertiary LHC collimators. Master's thesis, Universita degli studi di Padova, 2017.

[18] G. Valentino et al. Final implementation, commissioning, and performance of embedded collimator beam position monitors in the Large Hadron Collider. *Physical Review Accelerators and Beams*, 20(8):081002, 2017.

[19] M. Gasior, J. Olexa, G. Valentino, and G. Baud. First Operational Experience with the LHC Diode ORbit and OScillation (DOROS) System. In *Proceedings of IBIC2017, Barcelona, Spain*, pages 43–46, 2017.

[20] R. J. Steinhagen. *LHC Beam Stability and Feedback Control-Orbit and Energy*. PhD thesis, RWTH Aachen U., 2007.

[21] R. W. Aßmann et al. Expected performance and beam-based optimization of the LHC collimation system. In *Proceedings of the 9th European Particle Accelerator Conference, Lucerne, Switzerland*, pages 1825–1827, 2004.

[22] S. Redaelli et al. Operational experience with a LHC collimator prototype in the CERN SPS. In *Proceedings of IPAC2009, Vancouver, Canada*, pages 2835–2837, 2009.

[23] G. Robert-Demolaize. *Design and performance optimization of the LHC collimation system*. PhD thesis, Ecole Nationale Superieure de Physique de Grenoble, 2006.

[24] D. Wollmann et al. First cleaning with LHC collimators. In *Proceedings of IPAC2010, Kyoto, Japan*, pages 1237–1239, 2010.

[25] G. Valentino et al. Semiautomatic beam-based LHC collimator alignment. *Physical Review Special Topics-Accelerators and Beams*, 15(5):051002, 2012.

[26] G. Azzopardi et al. Operational Results of LHC Collimator Alignment Using Machine Learning. In *Proceedings of IPAC2019, Melbourne, Australia*, pages 1208–1211. JACOW Publishing, Geneva, Switzerland, 2019.

[27] B. M. Salvachua Ferrando et al. Observations of beam losses at the LHC during reduction of crossing angle. In *Proceedings of IPAC2017, Copenhagen, Denmark*, 2017.

[28] D. G. Kleinbaum and M. Klein. *Logistic Regression: A Self-learning Text*. Springer Science and Business Media, 2010.

[29] M. Svensén and C. M. Bishop. *Pattern recognition and machine learning*. Springer Science and Business Media, 2007.

[30] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[31] I. Steinwart and A. Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[32] D. M. Magerman. Statistical decision-tree models for parsing. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 276–283. Association for Computational Linguistics, 1995.

[33] L. Breiman. Random Forests. *Machine learning*, 45(1):5–32, 2001.

[34] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[35] S. R. Eddy. Hidden markov models. *Current opinion in structural biology*, 6(3):361–365, 1996.

[36] S. R. Eddy. What is a hidden Markov model? *Nature biotechnology*, 22(10):1315, 2004.

[37] 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. Technical report, CERN, 2012.

[38] H. Grote and F. Schmidt. MAD-X-an upgrade from MAD8. In *Proceedings of the 2003 Particle Accelerator Conference*, volume 5, pages 3497–3499. IEEE, 2003.

[39] F. Zimmermann. Future colliders for particle physics - "Big and Small". *Nuclear Instruments and Methods in Physics Research Section A*, 909:33–37, 2018.

[40] Tevatron I Group et al. Design Report Tevatron 1 Project. Technical report, FERMILAB-DESIGN-1984-01, 1984.

[41] N. Mokhov et al. Tevatron beam halo collimation system: design, operational experience and new methods1. *Journal of Instrumentation*, 6(08):T08005, 2011.

[42] A. Drees, R. Fliller, and W. Fu. RHIC loss limitations and collimation. In *AIP Conference Proceedings*, volume 773, pages 55–59. AIP, 2005.

[43] M. Seidel. *The proton collimation system of HERA*. PhD thesis, Hamburg University, 1994.

[44] G. Valentino et al. Semi-automatic beam-based alignment algorithm for the LHC collimation system. In *Proceedings of IPAC2011, San Sebastian, Spain*, pages 3768–3770, 2011.

[45] J. X. Tao, A. Ray, S. Hawes-Ebersole, and J. S. Ebersole. Intracranial EEG substrates of scalp EEG interictal spikes. *Epilepsia*, 46(5):669–676, 2005.

[46] P. Xanthopoulos et al. A novel wavelet based algorithm for spike and wave detection in absence epilepsy. In *2010 IEEE International Conference on BioInformatics and BioEngineering*, pages 14–19. IEEE, 2010.

[47] B. Yang, Y. Hu, Y. Zhu, Y. Wang, and J. Zhang. Intracranial EEG Spike Detection Based on Rhythm Information and SVM. In *2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, volume 2, pages 382–385. IEEE, 2017.

[48] J. E. Le Douget et al. Surface and intracranial EEG spike detection based on discrete wavelet decomposition and random forest classification. In *2017*

39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pages 475–478. IEEE, 2017.

[49] W Ganglberger et al. A Comparison of Rule-Based and Machine Learning Methods for Classification of Spikes in EEG. *JCM*, 12(10):589–595, 2017.

[50] M. M. Hartmann et al. PureEEG: automatic EEG artifact removal for epilepsy monitoring. *Neurophysiologie Clinique/Clinical Neurophysiology*, 44(5):479–490, 2014.

[51] H. M Ozaktas, O. Arikan, M. A. Kutay, and G. Bozdagt. Digital computation of the fractional Fourier transform. *IEEE Transactions on signal processing*, 44(9):2141–2150, 1996.

[52] R. Merletti, P. A. Parker, and P. J. Parker. *Electromyography: physiology, engineering, and non-invasive applications*, volume 11. John Wiley & Sons, 2004.

[53] T. Mukaeda and K. Shima. A novel hidden Markov model-based pattern discrimination method with the anomaly detection for EMG signals. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 921–924. IEEE, 2017.

[54] M. Point et al. Cryotherapy induces an increase in muscle stiffness. *Scandinavian journal of medicine & science in sports*, 28(1):260–266, 2018.

[55] R. M. Erskine et al. The individual and combined effects of obesity-and ageing-induced systemic inflammation on human skeletal muscle properties. *International Journal of Obesity*, 41(1):102–111, 2017.

[56] C. M. Smith et al. Differences between the Time Course of Changes in Neuromuscular Responses and Pretest versus Posttest Measurements for the Examination of Fatigue. *Journal of Nature and Science (JNSCI)*, 3(10), 2017.

[57] O. Fukuda, T. Tsuji, M. Kaneko, and A. Otsuka. A human-assisting manipulator teleoperated by EMG signals and arm motions. *IEEE Transactions on Robotics and Automation*, 19(2):210–222, 2003.

[58] Y. Zhang et al. Extracting time-frequency feature of single-channel vastus medialis EMG signals for knee exercise pattern recognition. *PloS one*, 12(7), 2017.

[59] I. Batzianoulis et al. EMG-based decoding of grasp gestures in reaching-to-grasping motions. *Robotics and Autonomous Systems*, 91:59–70, 2017.

[60] K. Englehart et al. A robust, real-time control scheme for multifunction myoelectric control. *IEEE transactions on biomedical engineering*, 50(7):848–854, 2003.

[61] H. Jaeger. The 'echo state' approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13, 2001.

[62] S. Beniczky. Automated real-time detection of tonic-clonic seizures using a wearable EMG device. *Neurology*, 90(5):428–434, 2018.

[63] G. Valentino, R. Aßmann, R. Bruce, S. Redaelli, and N. Sammut. Automatic threshold selection for BLM signals during LHC collimator beam-based alignment. In *2012 Sixth UKSim/AMSS European Symposium on Computer Modeling and Simulation*, pages 210–213. IEEE, 2012.

[64] G. Azzopardi et al. Beam Loss Threshold Selection for Automatic LHC Collimator Alignment. In *Proceedings of ICALEPCS2019, New York, NY, USA (Accepted for publication)*, 2019.

[65] D. M. Brandman, S. S. Cash, and L. R. Hochberg. Human intracortical recording and neural decoding for brain–computer interfaces. *IEEE Trans-*

actions on Neural Systems and Rehabilitation Engineering, 25(10):1687–1696, 2017.

[66] J. A. Herron et al. Cortical brain–computer interface for closed-loop deep brain stimulation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(11):2180–2187, 2017.

[67] V. A. Maksimenko et al. Absence seizure control by a brain computer interface. *Scientific reports*, 7(1), 2017.

[68] G. Hotson et al. Individual finger control of a modular prosthetic limb using high-density electrocorticography in a human subject. *Journal of neural engineering*, 13(2), 2016.

[69] T. M. Vieira, A. Botter, S. Muceli, and D. Farina. Specificity of surface EMG recordings for gastrocnemius during upright standing. *Scientific reports*, 7(1):13300, 2017.

[70] C. J. De Luca. The use of surface electromyography in biomechanics. *Journal of applied biomechanics*, 13(2):135–163, 1997.

[71] F. Di Nardo et al. Antagonist thigh-muscle activity in 6-to-8-year-old children assessed by surface EMG during walking. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3469–3472. IEEE, 2017.

[72] I. Mierswa. Yale: Rapid prototyping for complex data mining tasks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 935–940. ACM, 2006.

[73] R. Aßmann et al. Improving LHC Collimator Setup Efficiency at 3.5 TeV. Technical report, CERN, 2011.

[74] G. Valentino et al. Beam diffusion measurements using collimator scans in the LHC. *Physical Review Special Topics-Accelerators and Beams*, 16(2), 2013.

[75] G. Azzopardi, B. M. Salvachua Ferrando, G. Valentino, S. Redaelli, and A. Muscat. Operational Results on the Fully-Automatic LHC Collimator Alignment. *Physical Review Accelerators and Beams*, 22(9), 2019.

[76] G. Azzopardi et al. Automatic spike detection in beam loss signals for LHC collimator alignment. *Nuclear Instruments and Methods in Physics Research Section A*, 934:10–18, 2019.

[77] G. Valentino, R. W. Aßmann, R. Bruce, and N. Sammut. Classification of LHC beam loss spikes using Support Vector Machines. In *2012 IEEE 10th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 355–358. IEEE, 2012.

[78] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[79] F. Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12:2825–2830, 2011.

[80] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[81] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[82] S. J. Reeves and Z. Zhe. Sequential algorithms for observation selection. *IEEE Transactions on Signal Processing*, 47(1):123–132, 1999.

[83] J. W. Tukey et al. Comparing individual means in the analysis of variance. *Biometrics*, 5(2):99–114, 1949.

[84] B. Efron and R. J. Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.

[85] G. Azzopardi et al. Spike Pattern Recognition for Automatic Collimation Alignment. Technical report, CERN, 2017.

[86] G. Azzopardi, B. M. Salvachua Ferrando, and G. Valentino. MD1653-Part 1: Characterisation of BLM Response at Collimators. Technical report, CERN, 2018.

[87] W. Hofle et al. Controlled transverse blow-up of high-energy proton beams for aperture measurements and loss maps. In *Proceedings of IPAC2012, New Orleans, Louisiana, USA*, volume 12, pages 4059–4061, 2012.

[88] R. Bruce et al. Simulations and measurements of beam loss patterns at the CERN Large Hadron Collider. *Physical Review Special Topics-Accelerators and Beams*, 17, 2014.

[89] R. Bruce et al. 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*, 848:19–30, 2017.

[90] R. Bruce et al. Collimation-induced experimental background studies at the CERN Large Hadron Collider. *Physical Review Special Topics-Accelerators and Beams*, 22, 2019.

[91] L. Cseppento et al. CBNG-The new build tool used to build millions of lines of Java code at CERN. In *Proceedings of ICALEPCS2017, Barcelona, Spain*, 2017.

[92] B. Muschko. *Gradle in action*. Manning, 2014.

[93] R. W. Assmann, S. Redaelli, M. Lamont, and M. Jonker. Application software for the LHC collimators and movable elements. Technical report, CERN, 2007.

[94] G. Valentino et al. Upgraded control system for LHC beam-based collimator alignment. In *Proceedings of ICALEPCS2015, Melbourne, Australia*, 2015.

[95] G. Valentino et al. LHC Collimator Alignment Operational Tool. In *Proceedings of ICALEPCS2013, San Francisco, CA, USA*, pages 860–863, 2013.

[96] M Kuhn et al. First LHC Emittance Measurements at 6.5 TeV. In *Proceedings of IBIC2015, Melbourne, Australia*, pages 562–566, 2015.

[97] A. Mereghetti et al. Performance of the collimation system during 2016-hardware perspective. In *7th Evian Workshop on LHC beam operation, Evian, France*, pages 225–228. CERN, 2017.

[98] R. Bruce, S. Redaelli, and R. W. Aßmann. Principles for Generation of Time-dependent Collimator Settings during the LHC Cycle. In *Proceedings of IPAC2011, San Sebastian, Spain*, pages 3754–3756, 2011.

[99] G. Valentino et al. Successive approximation algorithm for beam-position-monitor-based LHC collimator alignment. *Physical Review Special Topics-Accelerators and Beams*, 17(2):021005, 2014.

[100] G. Azzopardi. *Results of Collimator Alignment*, 2017 (accessed July 7, 2019). LHC Collimation Working Group, https://indico.cern.ch/event/646289/contributions-/2624967/attachments/1479057/2293597/align res 2017.pdf.

[101] N. Fuster-Martínez et al. Run 2 Collimation Overview. In *9th Evian Workshop on LHC beam operation, Evian, France (Accepted for publication)*. CERN, 2019.

[102] D. Mirarchi et al. Collimation: experience and performance. In *7th Evian Workshop on LHC beam operation, Evian, France*. CERN, 2017.

[103] G. Azzopardi, B. M. Salvachua Ferrando, and G. Valentino. MD3343-Fully-Automatic Parallel Collimation Alignment using Machine Learning. Technical report, CERN, 2018.

[104] R. Bruce, R. W. Aßmann, and S. Redaelli. Calculations of safe collimator settings and β* at the CERN Large Hadron Collider. *Physical Review Special Topics-Accelerators and Beams*, 18(6):061001, 2015.

[105] A. Mereghetti et al. β*-Reach–IR7 Collimation Hierarchy Limit and Impedance. Technical report, CERN, 2016.

[106] G. Azzopardi et al. Automatic angular alignment of LHC collimators. In *Proceedings of ICALEPCS2017, Barcelona, Spain*, pages 928–933, 2017.

[107] G. Azzopardi et al. Automatic angular alignment of LHC collimators. Technical report, CERN, 2017.

[108] G. Azzopardi, B. M. Salvachua Ferrando, and G. Valentino. MD3344-Fully-Automatic Angular Alignment of LHC Collimators using Machine Learning. Technical report, CERN, 2018.