AN OFF-MOMENTUM BEAM LOSS FEEDBACK CONTROLLER AND GRAPHICAL USER INTERFACE FOR THE LHC

B. Salvachua*, D. Alves, G. Azzopardi, S. Jackson,
D. Mirarchi, M. Pojer, CERN, Geneva, Switzerland
G. Valentino, University of Malta, Msida, Malta

Abstract

During LHC operation, a campaign to validate the configuration of the LHC collimation system is conducted every few months. This is performed by means of loss maps, where specific beam losses are voluntarily generated with the resulting loss patterns compared to expectations. The LHC collimators have to protect the machine from both betatron and off-momentum losses. In order to validate the off-momentum protection, beam losses are generated by shifting the RF frequency using a low intensity beam. This is a delicate process that, in the past, often led to the beam being dumped due to excessive losses. To avoid this, a feedback system based on the 100 Hz data stream from the LHC Beam Loss system has been implemented. When given a target RF frequency, the feedback system approaches this frequency in steps while monitoring the losses until the selected loss pattern conditions are reached, so avoiding the excessive losses that lead to a beam dump. This paper will describe the LHC off-momentum beam loss feedback system and the results achieved.

INTRODUCTION

In the Large Hadron Collider two counter-rotating beams collide in 4 experiments at a centre-of-mass energy of up to 14 TeV [1, 2]. In order to keep the beams on their desired trajectory, super-conducting dipoles with a nominal field of 8.33 T are used. A collimation system protects the machine against beam losses. The loss of a very small fraction of the circulating beam is a concern because this may induce a magnet quench, where a superconducting magnet enters the normal conducting state, initiating a beam dump or, in extreme cases even damaging machine components.

More than 100 collimators are located in the LHC ring in order to concentrate beam losses in these dedicated areas. The main betatron cleaning occurs in LHC Insertion Region 7 (IR7) and off-momentum particles are cleaned in LHC Insertion Region 3 (IR3). A collimation hierarchy is defined with robust primary collimators with smaller gaps catching primary beam losses, secondary collimators more retracted than the primaries intercepting secondary showers, and absorbers to intercept and absorb the remaining showers. At both sides of each colliding interaction point, tertiary collimators provide local protection to the strong focusing triplet magnets while additional absorbers are used to intercept the physics collision debris.

The hierarchy of the collimation system has to be validated and verified at least every three months during machine operation in order to ensure that any possible drift does not affect the beam cleaning efficiency. The validation is done by means of beam loss maps. Betatron cleaning is validated by transversely exciting a low intensity beam with white noise in a controlled way, generating slow beam losses that are used to verify the collimation hierarchy in IR7 and the residual leakage into the cold magnets. Off-momentum cleaning is verified by shifting the Radio-Frequency (RF) by a fixed amount resulting in an orbit shift for both beams. This shift is more pronounced were the machine dispersion is highest, leading to higher beam losses in the off-momentum collimation system in IR3.

The validation of the LHC configuration through loss maps is performed at every stage in the LHC cycle: at injection energy with injection protection collimators inserted and retracted, at the end of the energy ramp, at the end of the beam squeeze, when the beams are colliding, as well as for special experimental configurations. Off-momentum validation [3] of a single configuration requires two dedicated fills for each configuration (to probe particles with both higher and lower than nominal momentum) where the beams are completely lost on the IR3 collimators. A minimum of 5 to 8 configurations need to be validated in this way. For the betatron cleaning a procedure was developed that allows loss maps to be performed without dumping the beam. However this was not initially the case for off-momentum loss maps, whose time significantly impacted the machine availability for physics. In order to perform the off-momentum loss maps in a controlled way without losing the beam a feedback based on the signal from beam loss monitors located downstream from each collimator was developed, together with a dedicated graphical user interface.

BEAM LOSS MONITORING DATA

The LHC is equipped with more than 3 600 ionisation chambers [4] distributed along the ring and covering the regions were high losses are expected. Downstream from each collimator there is at least one such ionisation chamber.

The beam loss monitors detect secondary shower particles the number of which, for a given energy, are proportional to the number of initiating protons. The signal is read-out in 12 different running sums (RS01 - RS12) ranging from 40 μs to 83.9 s calculated every 40 μs. A Java concentrator publishes the BLM data every 1 Hz for each of the running sums. A special fast data stream of RS06 (integration time of 10.06 ms) is published via FESA (Front-End Software

_____
* belen.salvachua@cern.ch

Architecture) [5] software at a higher rate, 100 Hz and down-sampled to 25 Hz. This is the stream of data used for the automatic collimation alignment [6, 7]. It is only available for the monitors downstream of the collimators and a few additional monitors in the dispersion suppressor regions of IR3 and IR7.

RADIO FREQUENCY TRIM

The technique used in LHC Run 1 (2010 - 2013) to validate the off-momentum configuration was to change the RF frequency by an amount large enough to move the beam onto the primary collimators in IR3. This was done with low intensities in the machine, below 3×10^{11} protons per beam. The RF frequency shift used was very large, typically ±500 Hz. While this allowed the beam halo leaking into the cold magnets to be accurately measured, the beam was irreversibly lost, making the validation lengthy, in particular at top energy.

Several studies were performed in order to asses the minimum RF frequency shift that is needed to generate enough losses to validate the collimation hierarchy [8], but this shift depended on the optics and on the collimator settings.

In LHC Run 2 a feedback that trimmed the RF frequency while monitoring the beam losses via the fast stream of BLM data was implemented. The feedback worked correctly in many cases but, due to the way it was implemented, the system was not optimal. A target shift of RF frequency was set with the frequency then changed at a rate of 110 Hz/s. The RF setting was immediately reverted if the losses exceeded a pre-defined threshold. This meant that if the acquisition of the BLM data was delayed or lost for any reason the controller could not revert the RF setting quickly enough. In order to explore ways to better control the frequency trims a new feedback controller together with a graphical user interface was developed.

NEW FEEDBACK CONTROLLER

The new feedback controller is fully based on Python, although low level processes in FESA still control the RF trim and provide the fast BLM data. Active trims to the hardware and acquisition actions are performed via a Java API for Parameter Control (JAPC) [9].

The system architecture is presented in Fig. 1. The RF frequency is trimmed via UDP packges sent to the RF Gateway from the LHC Feedback Controller FESA class, also used for the LHC position, tune and energy feedbacks. Monitoring of the BLM signals is done via one of the LHC collimator FESA classes that acts as a data concentrator, gathering the fast stream of BLM data at 100 Hz and making it available to the collimator controller at 25 Hz. The beam current is also monitored to inhibit the feedback if the beam intensity is higher than a pre-defined threshold. The project is organized in three layers:

- Hardware: involving beam current transformers (BCT) for monitoring of total beam current, the low level RF

system for the frequency change and the BLM system for measuring the losses.

- FESA Middleware: the reading from and writing to hardware is done via the FESA classes shown in Fig. 1.

- Python Top Level: interaction with the user is performed using a Graphical User Interface (GUI) written in Python, whilst monitoring of the signals (BCT, BLM and RF frequency) and the actual off-momentum feedback controller implementation is also done with Python code.

The frequency trim sent to the global LHC feedback controller needs be done in momentum units and not in frequency. However it is more intuitive for the user to select the frequency in Hz directly. The software therefore converts internally the user selected value using Eq. 1

$$\Delta p = -1 \times 8.3333 \times 10^{-6} \Delta f \qquad (1)$$

where Δp is the value in momentum units and Δf the value in Hz. Valid values of Δf are between ±500 Hz and is constrained by the application.

Additional checks are performed before the off-momentum feedback is active, including a check for machine protection purposes, where the beam intensity in the machine needs to be smaller than the predefined value of 5×10^{11} p.

The feedback can be started in three different configurations defined by the user via the GUI:

1. Loop mode using 1 Hz BLM data.

2. Feedback mode using 1 Hz BLM data.

3. Feedback mode using 100 Hz BLM data.

All of them are initiated in the same way, with the user specifying the frequency shift parameters: start ($\Delta f_{initial}$), end (Δf_{final}) and step (Δf_{step}).

The input parameters used for the comparison of the BLM data are:

- BLM maximum ratio limit: the maximum allowed ratio of the measured BLM value to the BLM dump threshold.

- Dispersion Suppressor (DS) relative limit: the maximum ratio of the current BLM value to the initial BLM value in the selected dispersion suppressor BLM monitors. Only the monitors in the dispersion suppressors left and right of the off-momentum cleaning region in IR3 are taken into account.
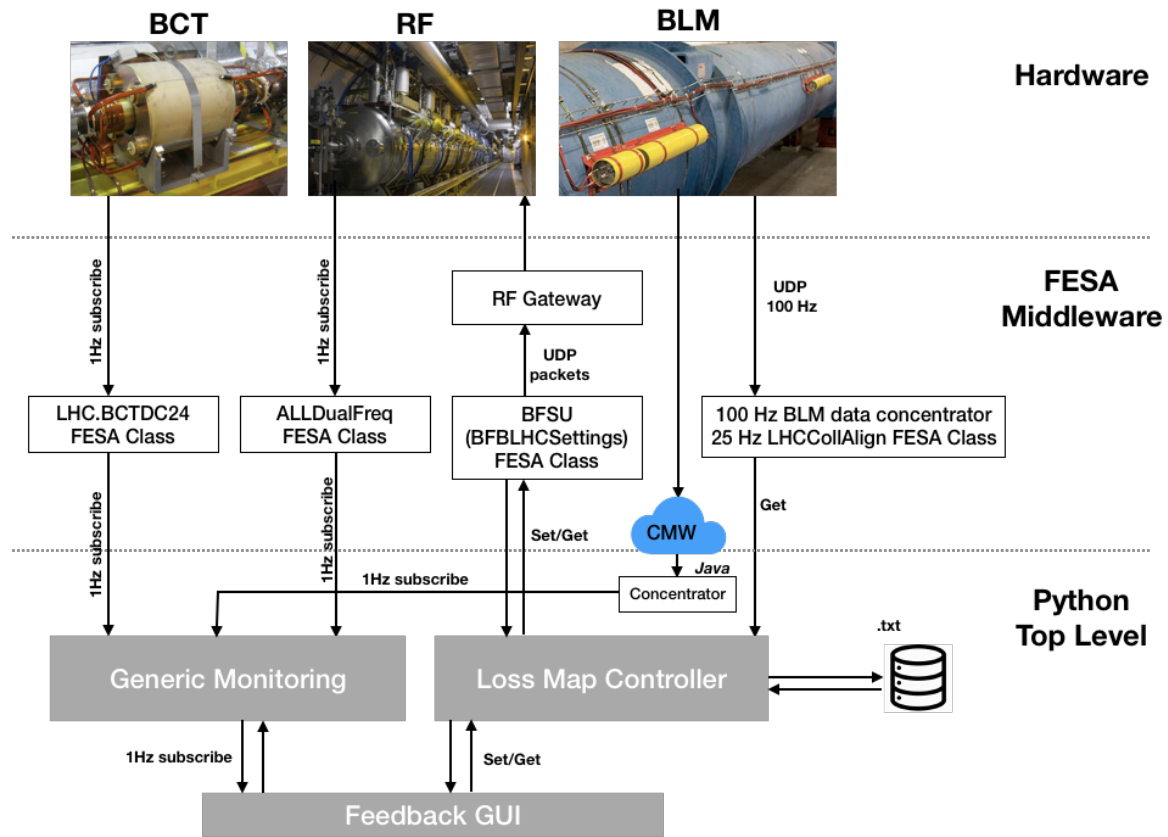
- BLM maximum value limit for 100 Hz.

Figure 1: Architecture diagram of the off-momentum feedback controller project.

Loop Mode 1 Hz BLM Data

This is called loop mode because the final frequency shift is achieved in iterative steps of a user selected frequency change, with the trim reverted each time between steps. The loop starts by setting the initial frequency shift ($\Delta f_{\text{initial}}$). Normally this would be zero but the GUI allows for a different starting point if it is smaller in absolute value than the allowed (Δf_{final}). Before evaluating the amount of beam loss observed the algorithm waits for the expected time needed to reach the target frequency (assuming 110 Hz/s). After this time the losses acquired at 1 Hz are evaluated, reading the values from the BLM concentrator (LHC_BLM_ACQ class). In this case all BLM monitors (values and dump thresholds) are read-out, but only the unmasked channels (channels that can initiate a beam dump) are taken into account. The check will return True if any BLM monitor of any running sum is above the selected *BLM maximum ratio limit*. Note that in this case the loop takes into account information of all BLM monitors in the LHC ring and any running sum. The trim is fully reverted after each step independently of the previous result but the result is kept in memory. The next step is to evaluate if the losses in the dispersion suppressor area were high enough to measure the halo cleaning leakage. The increase of signal (ratio of the current signal to the initial signal) for the BLM monitors in the dispersion suppressors of IR3 are used for this. These cells correspond to magnets where the losses are expected to be highest. If the signal

is higher than the pre-selected limit *DS relative limit* the check returns True and the loop stops. If the signal from the dispersion suppressors in IR3 is still lower than the limit, but the losses in the ring were higher then the same frequency shift is tried once again before stopping the loop. If both the dispersion suppressor signals in IR3 and those in the ring are below the selected limits the loop continues by increasing the trim to $\Delta f_{\text{initial}} + n\Delta f_{\text{step}}$ until Δf_{final} is reached.

Figure 2 (top) shows how the total frequency trim is increases over time until one of the stop conditions is reached. In this case, as the last step is not repeated, this corresponds to the case where the condition in the dispersion suppressors of IR3 is reached but ring losses are below selected threshold.

Feedback Mode 1 Hz BLM Data

In this case the feedback also starts with the $\Delta f_{\text{initial}}$ frequency shift but the frequency is constantly increasing (or decreasing, depending of the sign of the trim) in steps of Δf_{step} until Δf_{final} is reached. At each step the feedback evaluates the losses in the ring and in the dispersion suppressor using the same criteria as in the Loop mode. The feedback keeps trimming until any BLM in any running sum is above the selected margin to the dump limit or the signal in the dispersion suppressor is above threshold. Figure 2 (bottom) shows the evolution of the frequency over time in this mode. The losses are evaluated at every point.
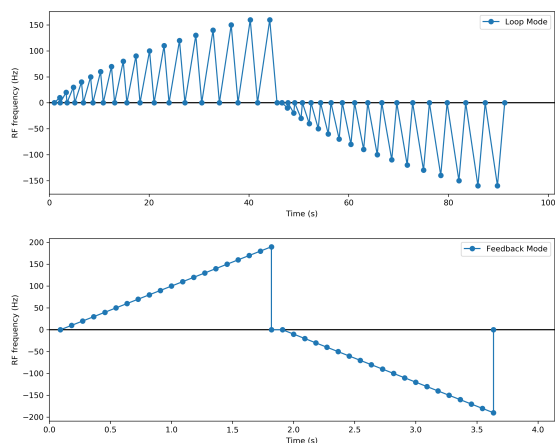
Figure 2: Simulated RF frequency change over time for the *Loop Mode* (top) and for the *Feedback Mode* (bottom).

Feedback Mode 100 Hz BLM Data

This feedback uses the fast stream of BLM data. However, the data is resampled at 25 Hz and is only available for the BLM monitors downstream of the collimators and 2 BLMs in the dispersion suppressors of IR3. For this reason, the check of losses in this mode is done with the BLMs at: the 2 primary skew collimators in IP7 (TCP.B6L7.B1 and TCP.B6R7.B2); the BLM downstream of a tertiary collimator in IP1 (TCTPV.4R1.B2) that will detect high losses from Beam 2 leaking from IR3; one BLM downstream of a tertiary collimator in IP5 (TCTPV.4L5.B1) that will detect high losses from Beam 1 leaking from IR3. In this mode this total of 6 monitors in representative locations replaces the data from the full ring used in the other two modes.

In addition to the check on the signal to dump ratio for the 4 BLM monitors selected, an additional check on the absolute value of the signal is also performed. For the dispersion suppressor signals the checks are the same as for the 1 Hz data, based on the increase of signal from the start of the loss map.

The final frequency shift is reached in steps and the losses described above are evaluated at every step, as shown in Figure 2 (bottom). The user can select only one of the three modes. For proton beams, the feedback mode at 100 Hz is the most performing option. However, for ion beams, the loss distribution along the dispersion suppressor it is very different than for protons and since the feedback mode at 100 Hz only checks 2 pre-selected BLM monitors in the DS, it is not the best option for ions. For ions, it is recommended the usage of the feedback mode at 1 Hz or the loop mode at 1 Hz as the check is done in all monitors in the ring.

SOFTWARE DESCRIPTION AND GUI

The active part of the controller was built in a dedicated and independent Python class - the Loss Map Controller; *offMomentumLossmap*. The off-momentum feedback can either be called from the Python interpreter of a custom Python script or from a dedicated GUI. The interface is sim-

ple. When the class is initiated default values of frequency shifts and beam loss limits are configured. These parameters can be changed by the user afterwards. The controller can then be called with these values, just indicating if the frequency shift is positive or negative (as shown below, in code line 2). The user can also specify as input parameters the minimum and maximum frequency shift and the step (see below in code line 3). The three modes of controller have the same interface.

```
lm = OffMomentumLossmap()
lm.lossmap_freq_loop_pos_dp()
lm.lossmap_freq_loop( min_freq = 0, max_freq =
    200, step_freq = 5)
```

Decorators are used to limit the maximum and minimum range of default parameters. For example, the maximum and minimum values for the dispersion suppressor relative limit are set to 50 and 1.

Another class covers the monitoring and publication of measured beam loss, RF frequency and beam intensity, *GenericMonitoring*. This is also a generic configurable class. When using it in standalone mode, the user can specify the FESA (hardware) properties that are monitored, the graph line names and the maximum number of points for each line. The class accesses the FESA properties via JAPC and creates an array with a rolling buffer with the number points specified and emits a pyQt signal when the buffer is updated. In the code lines below, one can see an example for the configuration of the beam current signals.

```
class BctMonitoring(GenericMonitoring):
    def __init__ ( self , japc = None):
        this_fesa_properties =
            ("LHC.BCTDC24.A/Acquisition#totalIntensityB1",
        "LHC.BCTDC24.A/Acquisition#totalIntensityB2")
        line_names = ("Beam1","Beam2")
        super(BctMonitoring, self ). __init__ ( fesaProperties
            = this_fesa_properties , lineNames =
            line_names)
        self . lineColors = ((0,0,204) ,(204,0,0) )
        self .signalName = "BCT"
```

A Graphical User Interface was built in Python using the Qt Designer tool. This is used to generate the layout of the GUI and provides an XML file that can then be automatically converted to a Python class. There is one tab for each controller mode, but monitor graphs, default settings and start-stop actions are common in the GUI for the 3 modes. Figure 3 shows a picture of the final GUI. On the top-left of the GUI the three tabs for the selection of the feedback mode are displayed; on the middle-left there is the configuration of maximum frequency shift and the step size; on the bottom-left there are the two panels for the default values selection (injection or top energy); on the right side of the GUI, the four graphs dedicated to the monitoring are displayed: RF frequency (top-left), beam currents (top-right), relevant BLM signals for Beam 1 (bottom-left) and relevant

BLM signals for Beam 2 (bottom-right). The configuration of the monitoring part is done using the *GenericMonitoring* class.
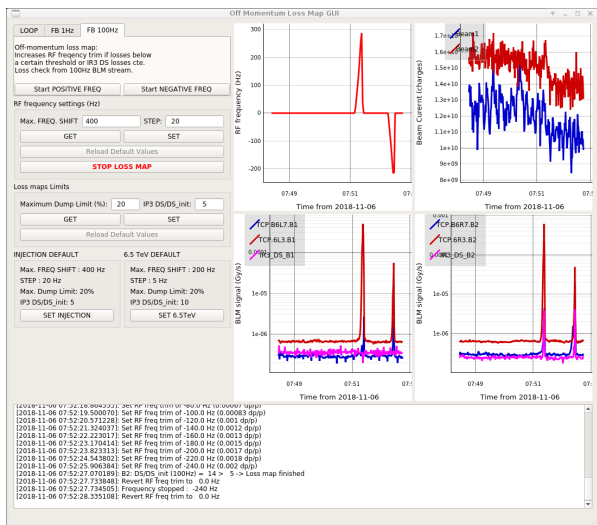

Figure 3: Graphical user interface for the beam loss off-momentum feedback.

The connection between the GUI layout and the off-momentum controller is done in an additional class called *AppOffMomentumLossmap*. This class is responsible for the threading of activities and defines controller settings at injection and top energy. The logs are displayed to the user in the GUI and are also saved locally in log files. Figure 4 shows the relationship between the main classes of the Python Top Level.
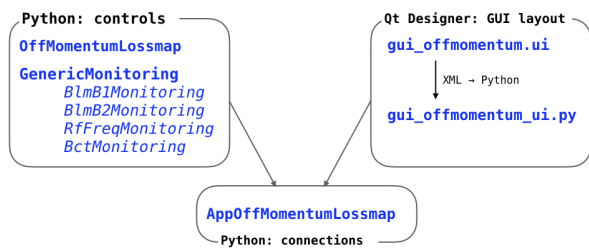

Figure 4: Description of the main classes used in the project.

RESULTS WITH LHC BEAMS

During 2018, several off-momentum loss maps were successfully done using the described feedback. Figure 5 shows as example of the distribution of losses (blue and red lines) while the RF frequency is changing (black line). The data is displayed at 1 Hz although the 100Hz mode was used as feedback.

Figure 6 shows as example of one of the loss maps achieved using this feedback. From August 2018, this off-momentum feedback was the default software used on the LHC to validate the off-momentum configuration.
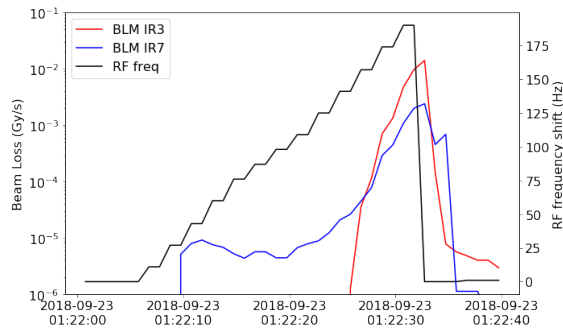

Figure 5: Distribution of beam losses after a primary collimator in IR3 (red) and in IR7 (blue) together with the change of RF frequency (black) as function of time.
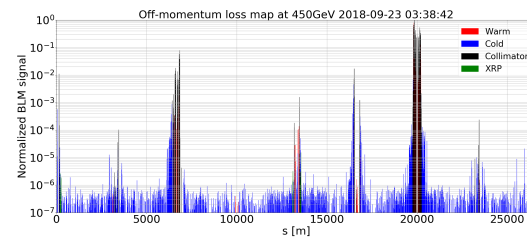

Figure 6: Off-momentum loss map done with the feedback on.

CONCLUSIONS

During 2018, a feedback controller to perform the off-momentum beam loss protection validation of the LHC was deployed in operation. The feedback software is based on a 2-tier structure, using FESA Middleware to access the hardware and a Top Level Python project to control and configure the feedback with a Python Graphical User Interface. The feedback was shown to be very effective, allowing several off-momentum loss maps to be performed with the same beam without triggering a beam abort, thus optimizing the machine time required to validate the collimation protection hierarchy.

ACKNOWLEDGEMENTS

REFERENCES

[1] L. Evans, "The Large Hadron Collider", in *New J. Phys.*, vol. 9, no. 9, art. no. 335 (22 pages), 2007. doi:10.1088/1367-2630/9/9/335

[2] L. Evans, P. Bryant, "LHC machine", in *J. Instrum.*, vol. 3, no. 8, art. no. S08001 (158 pages), 2008.

[3] B. Salvachua *et al.*, "Validation of off-momentum cleaning performance of the LHC collimation system", in *Proc. 7th Int. Particle Accelerator Conf. (IPAC'16)*, Busan, Korea, May 2016, pp. 2427–2429. doi:10.18429/JACoW-IPAC2016-WEPMW007

[4] E. B. Holzer, *et al.*, "Beam loss monitoring system for the LHC", in *IEEE Nuclear Science Symposium Conference Record, 2005*, vol. 2, pp. 1052–1056, 2005. `doi:10.1109/NSSMIC.2005.1596433`

[5] L. Fernandez *et al.*, "Front-end software architecture", in *Proc. 11th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'07)*, Oak Ridge, TN, USA, Oct. 2007, paper WOPA04, pp. 310–312.

[6] G. Azzopardi, A. Muscat, S. Redaelli, B. Salvachua, and G. Valentino, "Operational results of LHC collimator alignment using machine learning", in *Proc. 10th Int. Particle Accelerator Conf. (IPAC'19)*, Melbourne, Australia, May 2019, pp. 1208–1211. `doi:10.18429/JACoW-IPAC2019-TUZZPLM1`

[7] G. Azzopardi, B. Salvachua, A. Muscat, and G. Valentino, "Software architecture for automatic LHC collimator align-ment using machine learning", presented at the 17th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'19), New York, NY, USA, Oct. 2019, paper MOCPL04.

[8] B. Salvachua *et al.*, "An off-momentum beam loss feedback controller and graphical user interface for the LHC", presented at the 17th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'19), New York, NY, USA, Oct. 2019, paper MOPHA064.

[9] V. Baggiolini *et al.*, "JAPC - the Java API for parameter control (designing for smooth evolution)", presented at the 10th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'05), Geneva, Switzerland, Oct. 2005.