SIGNAL ANALYSIS FOR AUTOMATED DIAGNOSTIC APPLIED TO LHC CRYOGENICS AT CERN

K. O. E. Martensson, B. Bradu, G. Ferlin, CERN, Geneva, Switzerland

Abstract

The operation of the LHC at CERN is highly dependent on its associated infrastructure to operate properly, such as its cryogenic system where many conditions must be fulfilled for superconducting magnets and RF cavities. In 2018, the LHC cryogenic system caused 172 hours of accelerator downtime (out of 5760 running hours). Since the cryogenics recovery acts as a time amplifier, it is important to identify not optimized processes and malfunctioning systems at an early stage to anticipate losses of availability. The LHC cryogenic control systems embeds about 60,000 I/O whereof more than 20,000 analog signals which have to be monitored by operators. It is therefore crucial to select only the relevant and necessary information to be presented. This paper presents a signal analysis system created to automatically generate adequate daily reports on potential problems in the LHC cryogenic system which are not covered by conventional alarms, and examples of real issues that have been found and treated during the 2018 physics run. The analysis system, which is written in Python, is generic and can be applied to many different systems.

INTRODUCTION

The LHC (Large Hadron Collider) at CERN (European Organization for Nuclear Research) consists of eight 3.3 km long cryogenically independent sectors with shafts to the surface in between them. These shafts between two sectors and associated infrastructure are denoted as *points*. Cryogenic plants are installed in five out of nine points as illustrated in Fig. 1 and together they cool down a mass of 36,000 tonnes to a temperature of 1.9 K, making it the largest cryogenic system in the world. This paper presents an analysis system constructed with the purpose of detecting non-urgent problems in the cryogenic system that are not caught by conventional alarms and are difficult for operators to detect manually. Archived data are analyzed daily by eleven different analysis algorithms, after which reports for operators are generated and published online.

The cryogenic plants are using similar setups which makes automatic and coherent checks of the systems advisable. The analysis is therefore distributed among eight VMs (Virtual Machines), each analyzing one sector. Approximately 40 analysis jobs, together analyzing about 1000 signals, is defined for each sector. The system is not replacing the daily operator surveillance, but is helping the cryogenic operations team fulfilling its main objective of maximizing the availability and performance of the LHC's cryogenic system by:

- Anticipating potential failures or loss of availability.
- Detecting unoptimized processes and malfunctioning sensors.
- Prolonging the life span of equipment.
- Reducing the need for repetitive manual operator checking and thereby freeing operator to other tasks.

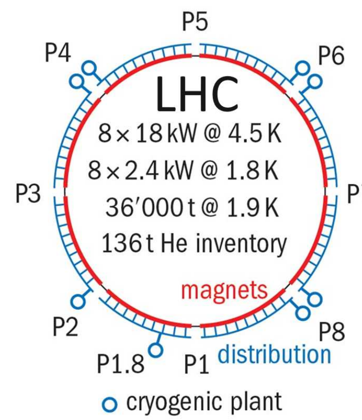

Figure 1: Distribution of cryogenic plants in the LHC [1].

INFRASTRUCTURE

Each sector is set up to be analyzed by a dedicated VM to ensure parallel and independent execution of new and updating analyses. These VMs are Linux machines that are set up on CERN's OpenStack. They are connected to a local machine using the software PuTTY©. New analyses are started automatically through cron scheduling every night and are performed on data spanning from the latest already available results up to the most recent midnight. It is also possible to redo a previous analysis with new parameters, and in that case only the explicitly queued time frame will be analyzed. This is useful when applying new or edited job specifications to dates which have already been analyzed. Triggered warnings are ranked by how severely the thresholds have been violated and are presented on a web site.

Python was chosen as programming language due to its versatility. Its many available modules and wide functionality gives great possibilities for easy expansion of the software to other types of systems and analyses. It is of great significance that the constructed analysis system can be adjusted with small effort to other systems at CERN, such as the computing cluster. At the time of construction, the necessary signals from the cryogenic system are not accessible from the computing cluster in a manner that it would make computations faster or more reliable. This is set to change in the future when the the currently undergoing change of primary logging service will be completed. Python is very commonly used at CERN and is therefore very probable to maintain

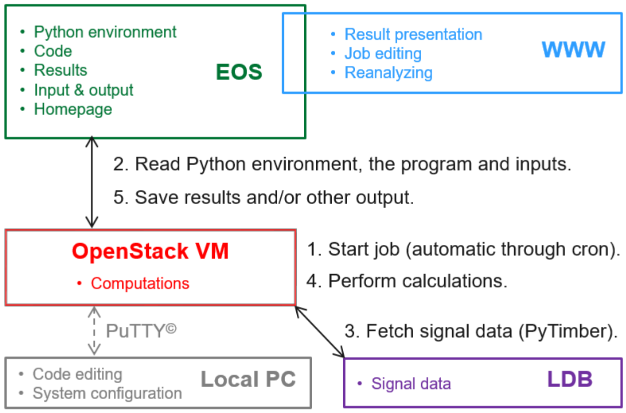

Figure 2: Schematic over the technical infrastructure set up to for the project and its work flow.

compatibility between system upgrades and to get early module updates. At time of writing this paper, the service used to obtain signal data is the CALS (CERN Accelerator Logging Service) API (Application programming interface) wrapper for Python called PyTimber [2]. CALS enables fetching of signal data from CERNS's LDB (Logging Database) to a local machine. UTC (Coordinated Universal Time) is used to avoid complications imposed by daylight savings time and to simplify conversions to Unix time.

The results of the analysis are presented on a homepage from which one can, depending on access level, perform one or more of the following actions if one has a CERN account:

- View performed analyses and corresponding results.
- Hide warnings.
- Edit existing analysis jobs and create new ones.
- Queue reanalyzing.

The access level of the user is determined by using CERN's SSO (Single Sign On) service and looking up the account name in a lists of access levels. The homepage is hosted from CERN's EOS (Elastic Organic Storage) file system, which is a distributed file system accessible from all VMs. EOS was therefore chosen as data storage to keep the complexity of the system down. The EOS file system has however shown to be unable to reliably handle sufficiently many simultaneous requests and will ignore some of them if overloaded. To avoid this issue, the analyses of the different sectors are launched with a delay of fifteen minutes in between. The technical infrastructure and its internal relations are illustrated in Fig. 2. The analysis jobs are categorized by where in the cryogenic system they are applied. In the result presentation, the categories are sorted by the number of new warnings and each warning shows a history of previous occurrences to make it easy to spot new events.

ANALYSIS ALGORITHMS

The analysis algorithms are generic and can be applied to any system that is generating signals which should relate to each other or themselves in a manner similar to what is described below. The generated warnings are sorted by their urgency $U = w * R$, where w is the weight of the job (default is 1) and R is a rank value which is computed differently depending on algorithm, but always in a manner so that a higher rank value correspond to a more severe violation of thresholds. Unless stated otherwise, warnings are triggered if $R > 1$, and threshold values are denoted as lim and are set by input parameters.

For some of the algorithms, a triggering signal behavior might not be an issue if there exists a non-problematic physical reason for the observed behaviour. In these cases, it is possible to cross-check triggered warnings with *causing signals*. If a behavior that is expected to generate the triggered warning is found in the causing signal, the warning will be disregarded.

When something is stated to be *given*, it means that it is set by input parameters. Most input parameters have default values corresponding to its most common use case, implying that they do not have to be specified unless exceptional settings are desired.

Family Analysis

Signals that are expected to have similar behaviour under proper operating conditions are denoted as a *family of signals*. The signals in a family analysis are analyzed by comparison to other signals in the family. The same operations are performed on all signals. The specific signal that is analyzed in a certain execution is referred to as *the tested signal* and uses the subscript T. The remaining signals in the family are referred to as *the other signals* and have the subscript O.

Correlation The purpose of this algorithm is to detect if a signal that is expected to follow others signals does not do that. The algorithm checks this by comparing local derivatives and assess if they differ significantly for a significant amount of time. The local derivatives are compared within a partial time window of given length and if the relative difference between the local derivative of the tested signal and the average local derivative of the other signals is larger than a given limit, a violation count is increased by one and a primary rank is assigned to the partial time window according to Eq. (1), where V is the value of the signals and t is time. If no violation is found for the partial time window, a tolerance count is increased by one. The partial time window is then shifted forward a given amount of time and the comparison is made over. If the total length of the violating time windows is longer than a given time limit, a warning will be triggered. If however, the tolerance count gets above a given limit, the violation count and tolerance count will be reset. If a warning is triggered, the final rank will be calculated through Eq. (2) and Eq. (3) where n is the largest number of partial time windows in which the conditions have been violated without a reset in between. The threshold lim_n is derived by calculating how many window shifts has to be done and yield a violating partial time window for the total violating time to be larger than a given limit. Equation (1) shows how the primary rank for a specific partial time window is calculated by dividing the difference between local derivative of the tested signal and the average local derivative of the others, by the average of the same two quantities. If $n > lim_n$, a secondary rank is calculated by taking the average fraction by which the primary rank condition was violated, over all primary ranks in the set of partial time windows that triggered the warning, N, as shown in Eq. (2).

$$R'_{V_T,i} = 2\left|\frac{\frac{dV_T}{dt} - \overline{\frac{dV_O}{dt}}}{\frac{dV_T}{dt} + \overline{\frac{dV_O}{dt}}}\right| \tag{1}$$

$$R_{V_T} = \sum_{i \in N} \frac{R'_{V_T,i}}{n \cdot lim_{R'}} \tag{2}$$

$$R_1 = \frac{1}{2}\left(R_{V_T} + \frac{n}{lim_n}\right) \tag{3}$$

The final rank is then calculated by taking the average of the secondary rank and the fraction by which the number of violating partial time windows has breached its limit, as described by Eq. (3). This analysis is applied to magnet temperature regulating valves for detection of malfunction.

Amplitude Comparison This algorithm compares the amplitudes within a family by their average value and optionally their median value. The rank value is given by

$$R_2 = \begin{cases} \frac{\left(\frac{A_T}{A_O} + \frac{MA_T}{MA_O}\right)}{2*(1+lim_{diff})} & \text{if } median = True \\ \frac{A_T}{A_O*(1+lim_{diff})} & \text{if } median = False \end{cases} \tag{4}$$

where A is the average amplitude and MA is the median amplitude. The amplitudes in this algorithm are the y-axis distance between *every* extrema and its neighbouring extrema. High frequency oscillations will thus produce many smaller amplitudes between all macro scale extrema and will therefore make the macro behaviour of the signal irrelevant. These oscillations can be adjusted for by calibrating the sampling time. There is also the possibility to set a minimum number of extrema detected for a warning to be triggered, if slowly oscillating signals are not of interest. This algorithm is applied to magnet temperatures as well as the valve positions and cold end temperatures of the DFB (Distribution Feed Box) current leads.

Offset The function of this algorithm is to checks if a signal is significantly distanced from other signals in its family. This algorithm includes two methods to chose from. Either it compares the average values of the signals, or it compares the average distance to the other signals. The rank is given by

$$R_3 = \frac{max(V_T, V_O)}{min(V_T, V_O) * (1 + lim_{Offset})}$$

where V_T is the either the average value of the tested signal or its average distance to the other signals, depending on which method is utilized, and V_O is either average of the other signals average value, or the average distance in between them. This algorithm is used to make sure that pressure and temperature sensors that should have similar values are coherent together. It is typically applied along cryogenic distribution lines and series of magnets.

Span Comparison This algorithm checks if the span, i.e. the difference between the largest and smallest value, of a signal is significantly larger than the average span of the other signals in its family. The rank value is given by

$$R_4 = \frac{S_T}{S_O * (1 + lim_{diff})}$$

where S is the span. The algorithm is applied to magnet temperature sensors and magnet temperature regulating valve positions to detect abnormal behaviour.

Individual Analysis

These algorithms use thresholds depending on the values of one single signal but warnings can be discarded depending on data from other signals.

Slow Deviation This algorithm checks if a signal that should have a generally flat appearance over time has a consistent divergence in the evolution of the signal value. This is done by fitting regressions to the data. A linear regression is fit to the data initially. If the relative change δ is larger than a limit lim_δ and the coefficient of determination r^2 is larger then a limit lim_{r^2}, a warning with a rank value given by

$$R_5 = \frac{\delta}{lim_\delta} \tag{5}$$

will be triggered. If the linear fit does not trigger any warning, the check is made again with a quadratic fit. The quadratic fit is only performed in the case of no warning from the linear fit to reduce the execution time. If a warning is triggered, the signals can be cross checked with a causing signal to see if the deviation is expected. E.g. if the pressure difference over a filter slowly increases, it could be due to a clogging (which would be a problem) but it could also be the consequence of an increase in flow (which would not be a problem). The comparison to a causing signal has a variable sensitivity that is set by providing a maximal relative difference in the relative deviations of the analyzed signal and the causing signal. If a deviation of a size within the limits are found in the causing signal, the warning will be suppressed. Beside filters, this algorithm is applied to vacuum gauges (without cross checking against any possible causing signal) to detect vacuum degradations. The algorithm can be set to only detect deviations in a specific direction or in both.

Integral This algorithm determines whether or not a signal significantly deviates from its mode (i.e. its most frequent value). It does so by subtracting the mode value from the signal and integrating the resulting curve. The rank value is given by

$$R_6 = \frac{|I|}{lim_I} \tag{6}$$

where I is the integral value and lim_I is the largest allowed integral size. The algorithm is applied to positions of magnet filling valves and to flow meters to detect significant leaks.

Count This algorithm counts the number of value switches, N, in a Boolean signal and checks if it is larger than a limit lim_N. Its rank value is given by

$$R_7 = \frac{N}{lim_N} \tag{7}$$

and it is used to check how frequent the cryogenic conditions for the magnet powering are lost and how often OnOff valves are opened to re-pressurize the helium guards (the helium guards are pressurized helium volumes located around cryogenic systems of sub-atmospheric pressure to avoid air pollution in helium circuits, so if these valves open too often, it means that there is a significant leak).

Average This algorithm will trigger a warning if the average signal value A_s is larger than a limit lim_u or smaller than a limit lim_l. At least one of the limits has to be defined and the rank value is given by

$$R_8 = \begin{cases} \frac{A_s}{lim_u}, & if\, A_s > lim_u \\ \frac{lim_l}{A_s}, & if\, A_s < lim_l. \end{cases} \tag{8}$$

The algorithm is applied to thermal shield valves and temperatures to detect reoccurring divergences from the design value.

Daily Average This algorithm checks if the ratio between a signal's daily average on two consecutive days is above a limit lim_r. It is applied to vacuum gauges to detect vacuum deteriorations. The rank value is given by

$$R_9 = \frac{A_d}{A_{d-1} * lim_r},$$

where A_d is the average signal value on date d.

Oscillation The purpose of this algorithm is to find undesired cyclic behaviour that might occur, for example, as a consequence of a feedback loop. Work on this topic has been conducted at CERN before [3], but tuning for the previously developed method has been shown difficult. Proper tuning requires extensive insight in the working principles of the measured quantities (e.g. typical valve movements, noise levels, etc.). This new method is meant to optimize the performance by limiting the scope of the algorithm. The limiting of the scope is done by excluding the probing of undesired high frequency oscillations. Since the system is not designed to make high frequency changes, signals with high frequency oscillations are deviating compared to its family's average behaviour and are therefore better probed by the Amplitude Comparison algorithm which does not trigger warnings for noise like oscillations present in the family.

This analysis is done by performing a fast Fourier transform and disregarding all frequencies of irrelevant amplitude or wavelength. The longest relevant wavelength is considered to be the one resulting in two complete periods over the analyzed time window and the shortest relevant wavelength is zero by default but can be changed through an input parameter. Either an absolute or a relative (compared to the signal average) amplitude limit has to be specified. The type of limit is specified by its naming. After the selection of relevant frequencies, signals corresponding to each of the remaining frequencies are then generated and one at the time cross-correlated to the original signal. If the magnitude of the correlation coefficient between the pure oscillation and original signal is greater than a given limit, a warning with the rank value determined by

$$R_{10} = \frac{A}{lim_A},$$

where A is the amplitude of the found cyclic behaviour and lim_A is the tolerated amplitude, will be triggered. This algorithm has the possibility to cross-check against a causing signal as well. If a sufficiently similar cyclic behaviour (relative frequency difference below a given limit) is found in a causing signal, the warning will be suppressed. There is also a possibility to set a minimum number of mean crossings for a warning to be triggered, to avoid triggers by quasi-cyclic signals triggering a warning (e.g. $y = x + sin(x)$). This algorithm is used to monitor the stability of helium pressures in cryogenic distribution lines.

PID Performance Analysis

This analysis is neither an individual nor a family analysis. It includes a set of signals that have a specific relation to each other but are not expected to have similar behaviour under optimal operating conditions, i.e. they are related but are of different families. It is used to detect poorly tuned PID (Proportional–Integral–Derivative) controllers and can be applied to all of them as long as signals for the controller output, the setpoint and a measurement of the controlled variable is available. It does not suggest how to tune it properly, it is only pointed out that it might need re-tuning. The algorithm is largely an adaptation of a concept already applied to LHC's cryogenic system, but through a different infrastructure [4]. The analysis is performed on partial time windows of a given size, and if the number of partial time windows in which the conditions are violated, n, is larger than a limit lim_n when the full time window has been analyzed, a warning ranked according to

$$R_{11} = \frac{n}{lim_n}$$

will be triggered. The main difference between the original function and this adaptation is that the former only counts the consecutively violated partial time windows whereas the latter counts all violations. The only other change is that the PI (Predictability Index) is capped at one by setting all values larger than one to one. In the original function a value larger than one was improbable but possible. This algorithm has a dedicated plotting function to show all relevant data.

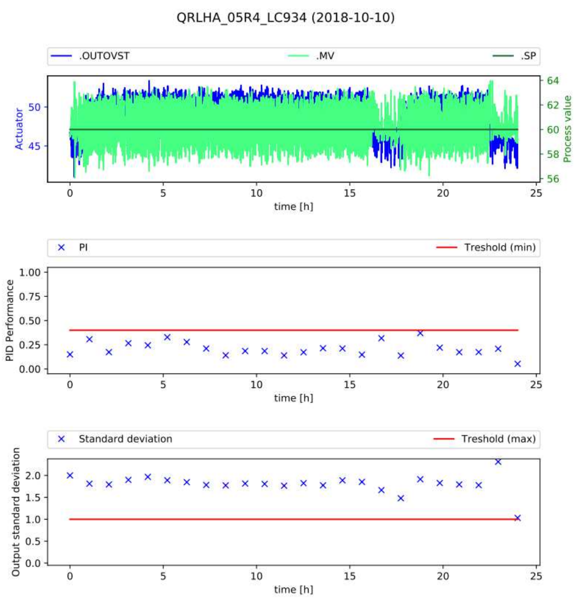

Figure 3: PID controller data that have triggered a warning, plotted by the dedicated plot function.

An example output of the dedicated plot function is shown in Fig. 3. The shown example is a benchmarking case with a clear oscillation of MV (Measured Value) and OUTOVST (controller output) around SP (the setpoint) as illustrated in the top subplot. From these signals, the PI values (presented as *PID performance*) in the middle subplot are derived, which all are below the threshold indicated by the horizontal line. This, together with the fact that the actuator in question is constantly requested to move, which is easiest seen in the bottom sub plot, suggests that the tuning of the controller is poorly done.

BENCHMARKING

To benchmark these different analysis algorithms, archived data from 2018 has been analyzed. When a date is stated to be benchmarked for, it means that data running up to the last second of previous day is analyzed. All of the benchmarking issues were found by the software and previously undetected issues were detected as well. The Correlation algorithm lacked a real case to compare to and was therefore benchmarked against synthetic cases. The *Correlation*, *Daily Average* and *Oscillation* algorithms were shown to be difficult to tune to avoid an abundance of false triggers. Since false triggers bloat the presented data and lowers the perceived significance of warnings, and the purpose of this software is not to warn about urgent issues (since these should be covered by conventional alarms), these three algorithms are currently tuned in a manner which yields no warnings when running on the benchmarking data. I.e. the cases which they have been tested against are not distinct enough to be found by settings that does not also yield many false triggers.

Previously Undetected Issues

Most prominent of the previously undetected issues were problems regarding valves with broken electronic components.

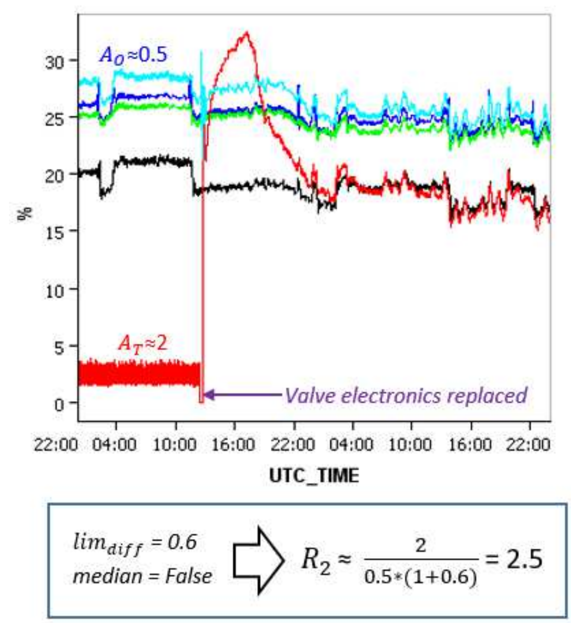

Figure 4: Broken valve electronics was replaced on July 18th 2018. The Issue was detected for the first time by the Amplitude Comparison algorithm during benchmarking and was ranked through Eq. (4).

A total of six instances of this issue were found, all of which were from similar families (valves in DFB current leads) but in different sectors. They were detected by the Amplitude Comparison algorithm because the valves were constantly going up and down with length significantly larger than the noise that tend to be present in these signals, thus yielding a significantly larger average amplitude compared to its family members. An example of this is shown in Fig. 4 where the average amplitude of the tested signal (the initially thick one), A_T, was calculated to 2 and the average amplitude of the other signals, A_O, was calculated to 0.5. The limit value $lim_{diff} = 0.6$ corresponds to allowing an A_T 60% larger than A_O, thus implying an upper threshold of 0.5*(1+0.6)=0.8 for A_T. Since the calculated value for

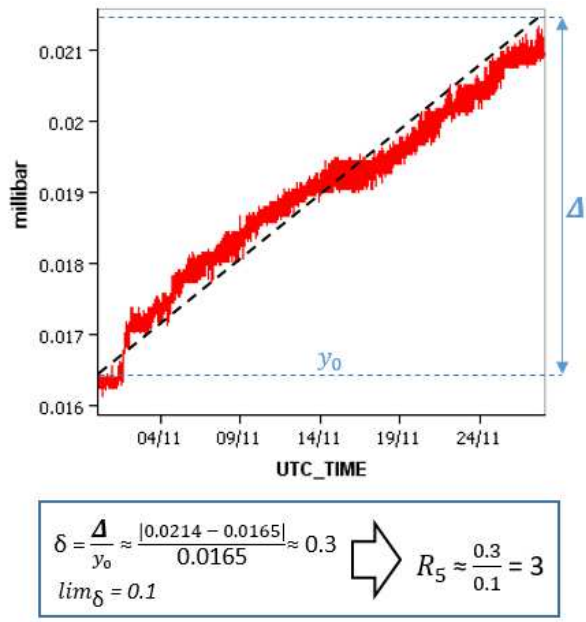

Figure 5: Degradation of primary vacuum found during benchmarking of the Slow Deviation algorithm. The issue was ranked through Eq. (5).

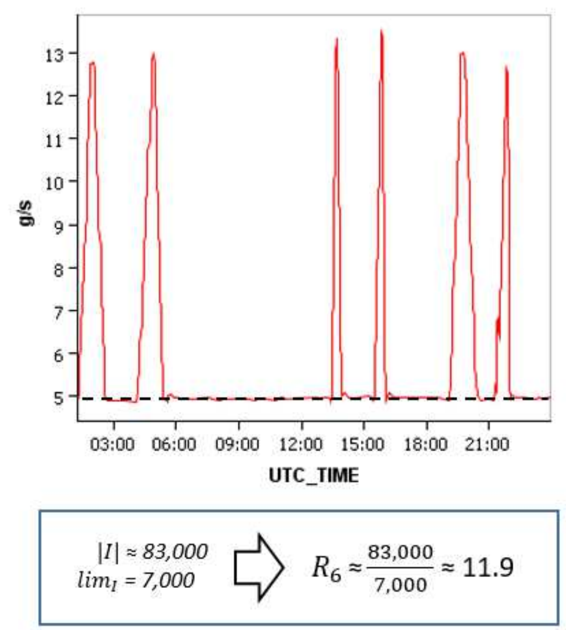

Figure 6: Internal leak on magnet bath detected by application of the Integral algorithm on a flow meter. The rank of the warning was calculated by using Eq. (6).

A_T was 2.5 times larger than the threshold, the job yielded a rank value of 2.5 and triggered a warning. The effect of replacing the broken electronics can be seen clearly. After some overshoot, the valve converged towards its family and started to regulate properly with an average amplitude similar to the other family members.

Figure 5 illustrates an example of an issue found by the Slow Deviation algorithm when analyzing for November 29th 2018. The pressure in the primary vacuum is not at an alarming level in any specific moment and neither is the local rate of change, but when observing at a longer time scale, a linear regression (dashed diagonal line) can be well fit to the data. The linear regressions show a consistent degradation of the vacuum (i.e. a leak) that has caused a pressure increase of approximately 30% over the span of 28 days. This increase is three times larger than the limit $lim_\delta = 0.1$ and thus yields a warning with the rank value 3.

Figure 6 shows an issue detected by the Integral algorithm when analyzing for November 25th 2018. The warning was triggered from an instance in which it was applied to helium flow meters to detect potential leaks in magnet baths. The y-axis represents grams per second and the x-axis represent seconds, which means that integration will give the accumulated mass flow in grams over the integrated time frame. The mode value is marked by a dashed horizontal line and is subtracted from the signal values before integration. The limit was set to be 7 kg of helium over 24 hours, so when the integration showed that the accumulated helium flow over the day was approximately 83 kg larger than what the mode value would have given, a warning of rank value 11.9 was generated.

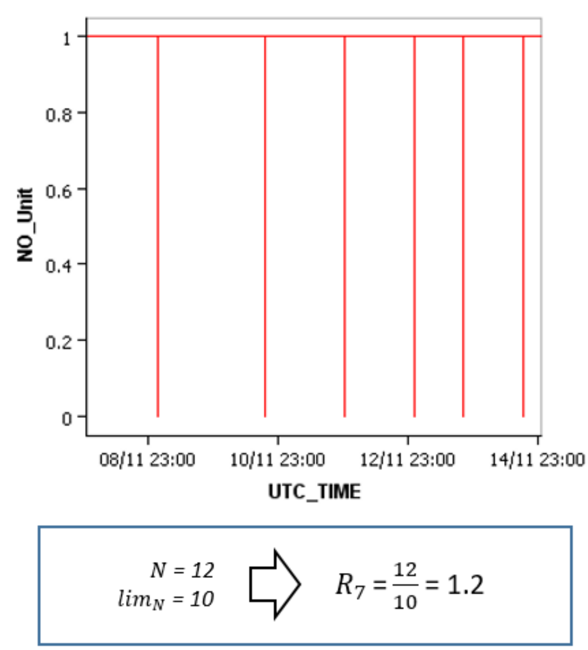

Figure 7: Leak issue in the helium guards detected by the Count algorithm. The warning was assigned a rank value according to Eq. (7).

A leak in a helium guard was detected by the Count algorithm when applied to the guards' OnOff valves in analysis for November 15th 2018. The triggering valve signal is shown in Fig. 7 where it can be seen to open and close six times within a week (thus changing value twelve times) which is alarmingly often.

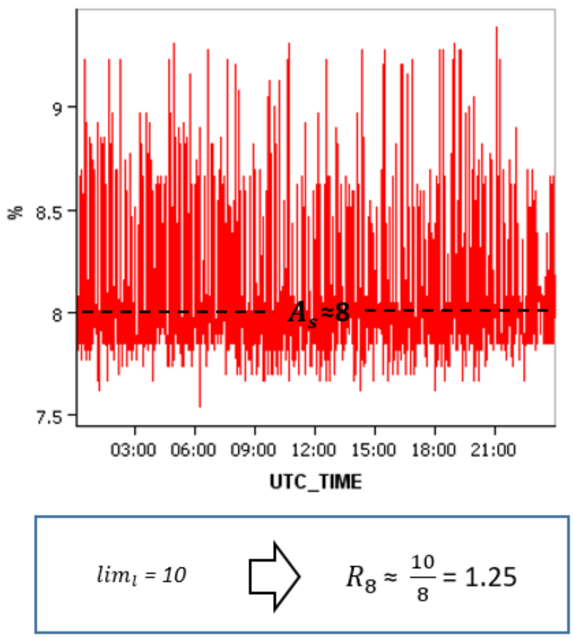

Figure 8: Thermal shield valve operating close to saturation after a reconfiguration and therefore not regulating properly. The issue was detected by the Average algorithm and the rank value was calculated through Eq. (8).

A poorly regulating thermal shield valve was detected by the Average algorithm when analyzing for November 18th 2018. The issue is shown in Fig. 8 where the average signal value is observed to be at the saturation level of 8%. The lower limit for the average was set to 10%, which is 25% more than the observed value, a warning with a rank value of 1.25 was therefore triggered.

CONCLUSION

The software is ready to be used for the LHC run 3, but tuning of settings will be an ongoing process since more data to tune against is gathered as the system is running.

All of the eleven algorithms have been conceptually validated, three of which have shown to be difficult to tune in an adequate manner. In the cases of tuning difficulty, removal of false triggers have been prioritized over detection of real issues since bloating of the result data would disintegrate the purpose of the software.

Oscillations are present in the data and they can be found, but when cross-checking with causing signals all potential warnings get suppressed. The hardship in tuning the Oscillation algorithm could thus be a consequence of there being no problematic oscillations in the data (except the high frequency ones that are covered by the Amplitude Comparison algorithm).

One of the two methods of the Offset algorithm is currently unused (the one comparing averages), but it is kept due to potential future applications.

A typical day generated on average 40 warnings per sector, of which 3 were new. The analysis of one sector took on average approximately 48 minutes, of which 30 minutes (63% of the elapsed time) for the fetching of signal data.

Among the valves with broken electronics, one had been oscillating in the manner shown in Fig. 4 since the year 2014, which is a good indicator of how hard it can be to detect this kind of problem through the use of conventional alarms and manual inspection. The fact that all six valves with broken electronics were of the same type suggests that it should be investigated if a systemic problem is causing the issues.

A new system for fetching of signal data is being developed at CERN and PyTimber will therefore be phased out. A separate version of the software has been constructed to utilize the new system. When the new system is ready, shifting to the adjusted version of the software is done by doing minuscule updates to the cron jobs.

ACKNOWLEDGEMENTS

Many thanks to Filippo Tilaro for help with setup of infrastructure.

REFERENCES

[1] L. Tavian, “The challenge of keeping cool,” *CERN Courier*, vol. 53, no. 7, pp. 29–31, Sep. 2013.

[2] PyTimber, `https://github.com/rdemaria/pytimber`

[3] F. Tilaro, B. Bradu, M. Gonzalez-Berges, M. Roshchin, “An expert knowledge based methodology for online detection of signal oscillations,” in *Proc. 2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, Annecy, France, Jun. 2017. `doi:10.1109/CIVEMSA.2017.7995293`

[4] B. Bradu, E. Blanco Vinuela, R. Marti, and F. M. Tilaro, “Automatic PID Performance Monitoring Applied to LHC Cryogenics”, in *Proc. ICALEPCS’17*, Barcelona, Spain, Oct. 2017, pp. 1017–1023. `doi:10.18429/JACoW-ICALEPCS2017-WEAPL02`