



UNIVERSITÄT HAMBURG



UNIVERSITÀ DEGLI STUDI
DELL'AQUILA



UNIVERSITAT AUTÒNOMA DE
BARCELONA

Erasmus Mundus Consortium “MathMods”

Joint MSc Programme in
Mathematical Modelling in Engineering: Theory, Numerics, Applications

In the framework of the
Consortium Agreement and Award of a Joint/Multiple Degree 2013-2019

Master's Thesis

Development of the Active Radiation Monitoring System of the LHCb Experiment at CERN

Supervisor

Prof. Alessandro D'Innocenzo

Candidate

Michał Mazurek
248836

and



Gloria Corti
Matthias Karacson
Luis Granado Cardoso



Academic Year 2017/2018

Laurea Magistrale in Ingegneria Matematica
Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica
Università degli Studi dell'Aquila

Abstract

The main goal of this work was to improve the LHCb Active Radiation Monitoring System (LbARMS) of the LHCb Experiment taking place at the European Organization for Nuclear Research (CERN). A new predictive algorithm was devised based on the Outlier-Robust Kalman Filter and Rauch-Tung-Striebel formula. The algorithm operates in quasi-real time in a Supervisory Control And Data Acquisition (SCADA) system. In addition, a new calibration model was suggested for RADFETs (RADiation-sensitive Field Effect Transistors) covering saturating range. A new Graphical User Interface (GUI) was created, which displays and monitors the results produced by the correction algorithm. It also assists the user in replacing the sensors if necessary. Finally, LbECSDData and LbARMS web APIs were developed in order to read and present data coming from the Experiment Control System (ECS).

Keywords:

LHCb, CERN, Radiation Monitoring, RadFET, p-i-n diode, Outlier-Robust Kalman Filter, Rauch-Tung-Striebel Smoother, WinCC OA, SCADA, RESTful API

Abstract

In questa tesi è stato applicato un miglioramento del sistema LHCb Active Radiation Monitoring System (LbARMS) dell'esperimento LHCb dell'acceleratore Large Hadron Collider (LHC) nell'Organizzazione europea per la ricerca nucleare (CERN). In dettaglio, è stato ideato un nuovo algoritmo predittivo basato sul Outlier-Robust Kalman Filter e la formula Rauch-Tung-Striebel. L'algoritmo opera in tempo reale con l'aiuto del sistema Supervisory Control And Data Acquisition (SCADA). Inoltre è stato suggerito un nuovo modello della calibrazione per i RADiation-sensitive Field Effect Transistors (RADFETs) che copre la regione di saturazione. È stata creata una nuova interfaccia grafica che mostra e controlla i risultati degli algoritmi. L'interfaccia assiste anche l'utente del sistema nella sostituzione dei sensori. Infine sono state sviluppate due web API (LbECSDData e LbARMS) per presentare i dati del sistema Experiment Control System (ECS).

Streszczenie

Celem tej pracy było ulepszenie obecnie istniejącego systemu LHCb Active Radiation Monitoring System (LbARMS) w eksperymencie LHCb Wielkiego Zderzacza Hadronów (LHC) przy Europejskiej Organizacji Badań Jądrowych (CERN). Nowy algorytm przewiduje poprawność obecnej wartości dawki promieniowania i strumienia cząstek na podstawie filtru Outlier-Robust Kalman Filter i algorytmu Rauch-Tung-Striebel. Algorytm działa w systemie Supervisory Control and Data Acquisition (SCADA), w czasie prawie rzeczywistym. Ponadto zaproponowano nowy model kalibracji dozymetrów RADiation-sensitive Field Effect Transistors (RADFETs), uwzględniający również zakres, w którym ulegają one nasyceniu. W tym celu powstał graficzny interfejs, który wyświetla i monitoruje pomiary. Pomaga on również użytkownikowi systemu w zamianie czujników na nowe. Ostatecznie powstały dwie aplikacje internetowe (LbECSDData i LbARMS), które wyświetlają dane pochodzące z systemu Experiment Control System (ECS).

Acknowledgements

I would like to express my very great appreciation to my supervisors at the LHCb Experiment: Gloria Corti, Matthias Karacson and Luis Granado Cardoso for their professional guidance and support in this project. I am really grateful that you always found some time to explain everything in detail. A big, massive ‘thank you’ to all friends I made during my internship and the whole LHCb Collaboration for making this amazing place even more welcoming. I have had an amazing time here!

I wish to acknowledge the help provided by professor Alessandro D’Innocenzo and MathMods programme coordinators: professor Bruno Rubino, Danilo Larivera and Cinzia Di Martino from the Department of Information Engineering, Computer Science and Mathematics at the University of L’Aquila. I would also like to thank all my friends I made during the MathMods programme. It has been an amazing cultural and scientific adventure with you guys!

Chciałbym wyrazić głęboką wdzięczność całemu Zakładowi Eksperymentu ATLAS przy Instytucie Fizyki Jądrowej im. Henryka Niewodniczańskiego w Krakowie. W szczególności chciałbym podziękować Pani Joli oraz Pani Eli za wprowadzenie do systemów SCADA.

Dziękuję całemu zespołowi JohnnyBros w Gdańsku za przekazanie cennej wiedzy na temat rozwoju oprogramowania w zespole oraz za kreatywne podejście w poszukiwaniu rozwiązań problemów, nawet jeśli wydaje się, że nie istnieją.

Pragnę podziękować również wszystkim nauczycielom i znajomym z Politechniki Gdańskiej za możliwość kształcenia interdyscyplinarnego. Wiedza zdobyta w trakcie studiów inżynierii materiałowej i fizyki technicznej okazała się niezbędna do pełnego zrozumienia problemów poruszanych w tej pracy.

Osobne podziękowania za nieustanne wsparcie oraz motywację chciałbym złożyć całej mojej rodzinie. W szczególności dziękuję mojej Ukochanej Klaudii, moim wspaniałym Rodzicom Marysi i Tomkowi, Cioci Bożence, Babci Klarze, Mariance, Krystianowi, Tomkowi, Magdzie, Jasiowi oraz Ani. Jesteście najlepszą Rodziną pod Słońcem!

Contents

1	Introduction	9
2	Experimental setup	11
2.1	LHCb Detector	11
2.1.1	Sub-detectors	11
2.1.2	Performance	13
2.1.3	Radiation environment	14
2.2	Dosimetry at the LHCb Experiment	15
2.2.1	Polymer-Alanine Dosimeters	15
2.2.2	RADiation-sensitive Field-Effect Transistor	15
2.2.3	High-resistivity forward biased p-i-n diodes	19
2.3	Online system	23
2.3.1	Architecture	23
2.3.2	WinCC OA	24
2.3.3	Functions of the Online System	27
2.4	LHCb Active Radiation Monitoring System	28
2.4.1	Correction algorithm	30
2.4.2	Online project	31
3	Theoretical introduction to Kalman filtering	33
3.1	Dynamic systems	33
3.2	State space estimation	35
3.2.1	Filtering	35
3.2.2	Smoothing	37
3.3	Estimating the state space without outliers	39
3.3.1	Filtering	39
3.3.2	Smoothing	40
4	New correction algorithm	42
4.1	Modeling the state space	42
4.2	Calibrating the smoother	44
4.2.1	Muting parameter μ as a signal-to-noise ratio manager	45
4.2.2	Degrees of freedom s for spike detection	46
4.2.3	Window's size N	47
4.3	Drift correction	49
4.4	Conversion of units	49
4.4.1	REM-PAD calibration	51

5	Description of software implementation	53
5.1	Offline project in C++	53
5.1.1	Installation	55
5.1.2	Running	56
5.1.3	Exception handling	59
5.1.4	MATLAB implementation of Kalman filters	60
5.2	Online project in WinCC OA	61
5.3	Django Web APIs	65
5.3.1	Generic application	65
5.3.2	LbECSDData	67
5.3.3	LbARMS	68
6	Outcome of the correction algorithm	69
7	Summary	74
	Appendix A Comparison of the results	76
	Appendix B Results for the devices with a serial no. 37	80

List of abbreviations

API – Application Programming Interface

ATLAS – A Toroidal LHC ApparatuS (LHC detector)

CERN – European Organization for Nuclear Research

CMRP – Center of Medical Radiation Physics (diode)

CMS – Compact Muon Solenoid (LHC detector)

CP violation – violation of charge conjugation parity symmetry

DAQ – Data Acquisition System

DP – datapoint

DPE – datapoint element

ECAL – Electromagnetic Calorimeter

ECS – Experiment Control System

ELMB – Embedded Local Monitoring Board

ESA – European Space Agency

GEDI – Graphical Editor (WinCC OA)

GUI – Graphical User Interface

HCAL – Hadron Calorimeter

HTTP – Hypertext Transfer Protocol

IP – Interaction Point

IT – Inner Tracker

JCOP – Joint COntrols Project

JS – JavaScript

JSON – JavaScript Object Notation

KF – Kalman Filter

LAAS – Laboratory of Analysis and Architecture of Systems (diode)

LHC – Large Hadron Collider

LHCb – Large Hadron Collider beauty (LHC detector)

M1-M5 – Muon System stations

MLE – maximum likelihood estimator

MOSFET – Metal-Oxide-Semiconductor Field-Effect Transistor

MVUE – unbiased minimum variance estimator

NIEL – Non Ionizing Energy Loss hypothesis

ORKF – Outlier-Robust Kalman Filter

ORKS – Outlier-Robust Kalman Smoother

ORM – Object-Relational Mapping

OT – Outer Tracker

PAD – Polymer-Alanine Dosimeter

PARA – Parametrization Tool (WinCC OA)

PS – Pre-Shower Detector

RadFET – RADiation-sensitive Field-Effect Transistor

REM – Radiation Experiment and Monitors (dosimeter)

REMUS – Radiation and Environment Monitoring Unified Supervision

REST – Representational State Transfer

RICH1, RICH2 – Ring Imaging Cherenkov detector stations

ROA – Resource-Oriented Approach

RTS – Rauch-Tung-Striebel algorithm

SCADA – Supervisory Control And Data Acquisition

SPD – Scintillating Pad Detectors

TFC – Time and Fast Control System

TT, T1-T3 – Tracking System stations

VELO – VERtex LOCator

WinCC OA (Open Architecture) – SCADA standard provided by SIEMENS

List of important symbols

D – absorbed dose [Gy]

Φ – particle fluence [cm^{-2}]

ΔV_{th} – threshold voltage [V]

ΔV_{th_0} – initial threshold voltage [V]

s_{ox} – SiO_2 layer length [nm]

ΔV_F – forward voltage [V]

ΔV_{F_0} – initial forward voltage [V]

C_{ox} – gate capacitance [F]

Φ_{eq} – neutron fluence equivalent of 1MeV [cm^{-2}]

$\mathcal{L}$ – instantaneous luminosity [$\text{cm}^{-2}\text{s}^{-1}$]

$\mathcal{L}_{int}$ – integrated luminosity [fb^{-1}]

S_+^n – positive semidefinite matrix

S_{++}^n – positive definite matrix

$t \in \mathbb{R}$ – time variable

$u(t), u_k$ – input vector

$x(t), x_k$ – state vector of the system

$\hat{x}_{k|i}$ – state estimate vector at timestep k with information up to i timestep

$\tilde{x}$ – estimation error vector

$\varphi(t), F_k \in \mathbb{R}^{n \times n}$ – state transition matrices

$\beta(t), B_k \in \mathbb{R}^{n \times m}$ – input model matrices

$\eta(t), H_k \in \mathbb{R}^{q \times n}$ – observation model matrices

z_k – observation vector

$\hat{P}_{k|i}$ – error covariance matrix at timestep k with information up to i timestep

K_k – Kalman gain at timestep k

1 Introduction

The LHCb Experiment is located at the Interaction Point 8 of the Large Hadron Collider (LHC) – currently the biggest particle accelerator in the world. It was designed to investigate the matter-antimatter antisymmetry puzzle by observing the decays of hadrons containing b and $\bar{b}$ quarks.

As any facility where ionizing radiation is present, LHCb is equipped with systems monitoring its intensity in order to prevent people from being exposed to high doses of radiation and to foresee potential damage to electronics. One of them, the LHCb Active Radiation Monitoring System (LbARMS) collects measurements of absorbed dose D and particle fluence Φ via RADiation-sensitive Field-Effect Transistors (RADFETs) and high-resistivity forward biased p-i-n diodes. The advantage of the active devices is that the readout of dose and fluence can be performed in quasi-real time. Moreover, the devices can be easily installed due to their small size. Although their response to radiation has already been studied, it is still necessary to implement correction algorithms that cover for negative effects during their operation.

The diodes and the dosimeters are connected to Embedded Local Monitoring Boards (ELMBs), which work as input/output devices for SCADA (Supervisory Control And Data Acquisition) – a system that is responsible for displaying, monitoring and archiving the measurements. The current standard of SCADA at CERN is WinCC OA from Siemens.

The LbARMS system so far incorporated a Graphical User Interface (GUI) written in WinCC OA that displayed raw voltage from all the devices, updated every 20 minutes. A correction algorithm script had to be executed offline and it took around 15 minutes to update the values. The inefficient algorithm and strong deviations in areas of high dose necessitated changes. Therefore, the main aims of the thesis were to:

- ensure the raw readout is correctly performed and the sensors are properly calibrated;
- provide tools transforming the raw values to physical values of dose and fluence appropriately, by applying corrections and calibration factors based on data previously collected;
- devise a predictive correction algorithm based on the readout history and data from other sources and evaluate the performance of this algorithm as applied to an online readout;
- validate the algorithm against other sources of data of similar type from unrelated systems.

In order to achieve this, the results obtained until the end of 2018 were compared with literature and other radiation systems operating in LHCb. The correction algorithm has been restructured by implementing more complex algorithms based on Outlier-Robust

Kalman Filter and Rauch-Tung-Striebel smoother. The algorithm has been completely moved to the online SCADA system in order to monitor, manage the correction procedure and visualize any results in real time. Finally, two Django APIs were created: LbECSDData and LbARMS. Both of them work as a read-only tool for the Oracle database. LbECSDData is focused on obtaining data from the Experiment Control System (ECS) of the LHCb Experiment, whereas LbARMS focuses on displaying measurements coming from the radiation sensors of the LbARMS and REMUS (Radiation and Environment Monitoring Unified Supervision) systems.

2 Experimental setup

2.1 LHCb Detector

The LHCb Detector is one of the large experiments located at the Large Hadron Collider (LHC) at CERN (The European Organization for Nuclear Research) [32]. It is looking for indirect evidence of new physics in ‘CP violation’ (violation of charge conjugation parity symmetry) where the laws of physics are not the same if a particle is interchanged with its antiparticle. ‘CP violation’ plays an important role in the matter-antimatter antisymmetry puzzle. In order to do this, the LHCb experiment focuses on capturing rare decays of beauty and charm hadrons. LHC is so far the most abundant supplier of B mesons (a pair of bottom antiquark and up, down, strange or charm quark) in the world.

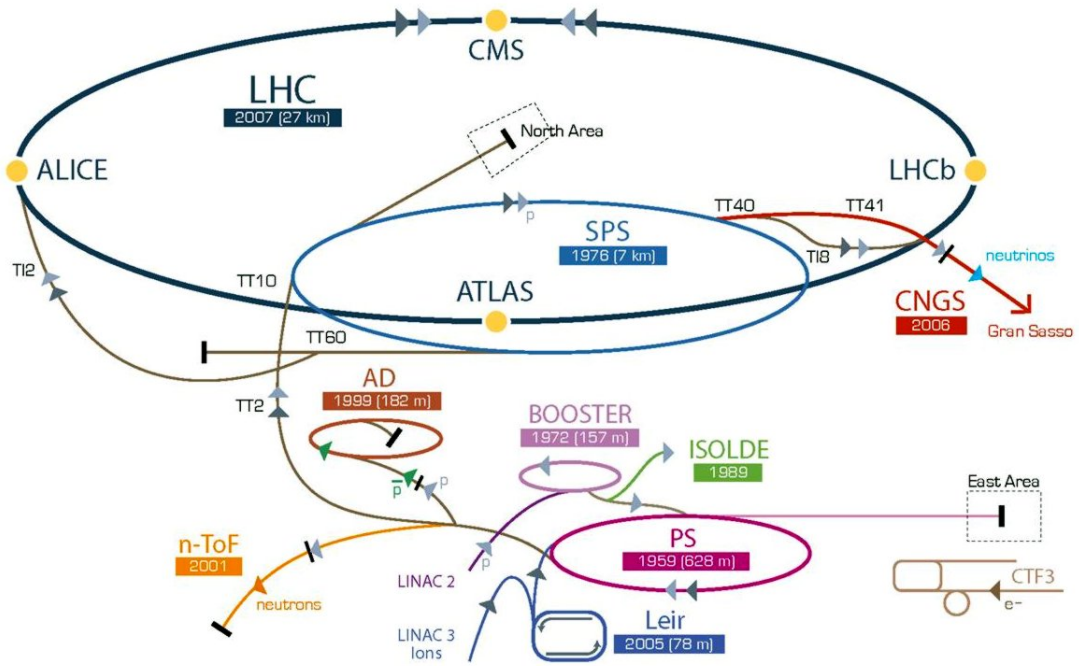


Figure 1: CERN Accelerator Complex

2.1.1 Sub-detectors

The LHCb detector [32, 33] is located at the Interaction Point 8 of the LHC (Figure 1), which was previously used by the DELPHI experiment. It is a single-arm spectrometer with a forward angular coverage. This specific shape of the detector has been chosen in order to detect hadrons containing b and $\bar{b}$ quarks, which are produced with a strong forward momentum along the beam lines. It is different than that of most collider experiments including the general-purpose detectors, CMS and ATLAS, at the LHC.

The Interaction Point (IP) of the LHCb detector is located close to the origin

of the experiment's coordinate system, with z -axis oriented downstream along the beam line (Figure 2). The beam pipe is made out of beryllium close to the interaction point and stainless steel outside this region. The choice of Be guarantees high transparency for particles traversing it. The VERtEX LOcator (VELO) sub-detector, which was built very closely around the IP, provides precise measurements of track coordinates and vertices.

Particles normally travel along the beam line. In order to measure their momentum particles of interest are required to move through a strong magnetic field. This is provided by a large dipole magnet with an integrated magnetic field of 4 Tm. The tracking system consists of four stations (TT, T1, T2 and T3) that provide track coordinates used to measure momentum of particles. Three of them (T1, T2 and T3) lie directly behind the dipole magnet and consist of two sub-detectors: the Inner Tracker (IT) and the Outer Tracker (OT). The Trigger Tracker (TT) and the IT, the latter placed close to the beam pipe are equipped with silicon microstrip detectors. On the other hand, the OT detectors use straw tube chambers and cover the majority of the sensitive area in T1-T3 stations.

Particle identities are determined using two RICH (Ring Imaging Cherenkov) sub-detectors, in which Cherenkov light generated by traversing particles is focused using spherical and flat mirrors to reflect the image out of the spectrometer acceptance.

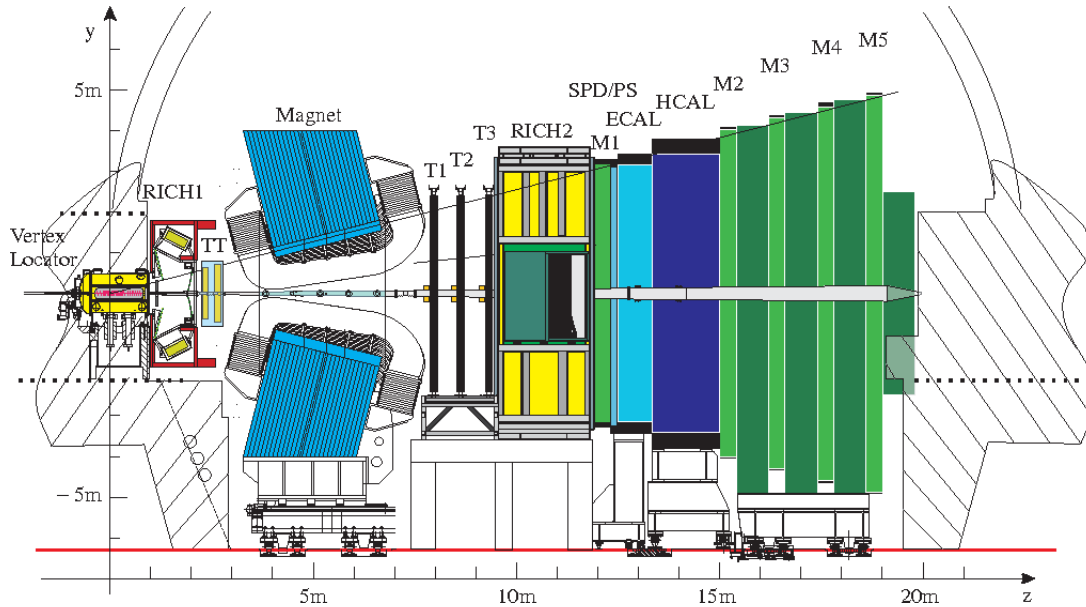


Figure 2: Side view of the LHCb detector.

Many B meson decays created at the LHCb produce muons that are detected in five rectangular stations (M1, M2, M3, M4 and M5). Whenever the muon passes through the chambers of each station, it interacts with a combination of three gases: carbon dioxide, argon and tetrafluoromethane, which induces charges that can be measured within a tightly knit wire grid within the gas chamber. The first muon station is positioned behind RICH2. The remaining four stations are located at the far end of the detector, gradually increasing in size.

Before muons reach the remaining four stations, the calorimetric system measures the amount of energy deposited when particles like electrons, photons and hadrons are stopped completely within its massive metallic bodies. It consists of several layers: the Scintillating Pad Detector (SPD), the Pre-Shower Detector (PS), the Electromagnetic Calorimeter (ECAL) and the Hadron Calorimeter (HCAL). The PS/SPD detectors allow for a fast discrimination between neutral and charged particles. The ECAL is responsible for measuring the energy of electromagnetic particles and the HCAL measures the remaining hadrons.

2.1.2 Performance

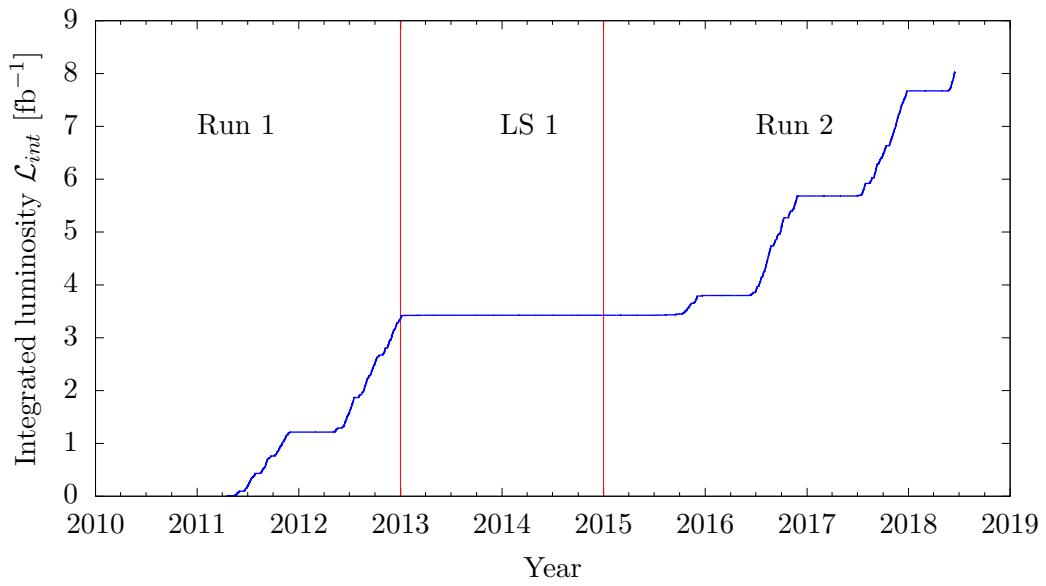


Figure 3: Integrated luminosity at the LHCb Experiment recorded in 2010-2018.

In order to measure the performance of the detector, physicists introduce a parameter called instantaneous luminosity $\mathcal{L}$ [21]. It is defined as the number of interactions N in a certain time divided by the cross section σ of the interaction

$$\mathcal{L} = \frac{1}{\sigma} \frac{dN}{dt} \quad [\text{cm}^{-2}\text{s}^{-1}]. \quad (1)$$

A more useful term for the work described in this thesis is the integrated luminosity $\mathcal{L}_{int}$, because it is proportional to the total number of events over time and helps to follow the history of the number of collisions taking place in the detector. It is defined by the integral from time 0 to time T of the instantaneous luminosity

$$\mathcal{L}_{int} = \int_0^T \mathcal{L}(\tau) d\tau \quad [\text{fb}^{-1}]. \quad (2)$$

The evolution of the integrated luminosity delivered to the LHCb Experiment in 2010-2018 is depicted in Figure 3. Its nondecreasing shape is caused by a positive instantaneous luminosity during operation (Run 1 and Run 2). During the maintenance period (LS1) luminosity remains constant since no collisions are taking place.

2.1.3 Radiation environment

The radiation conditions at the LHC are unprecedented because of the very high energy of the colliding hadrons. The mixed fields generated by the collisions consist of a large variety of particles spread over a wide energy range, and they deposit high amounts of energy on their way through the experiment. Therefore, it is crucial to measure the radiation level in the facility in order to protect people from harm and to foresee potential damage to electronics. Let E be the energy deposited within mass m by ionizing radiation. Then *absorbed dose* D is defined by

$$D = \frac{dE}{dm} \quad \left[\frac{\text{J}}{\text{kg}} \right]. \quad (3)$$

Now, let N be the number of particles passing through the cross-sectional area σ . Then the fluence Φ is defined by

$$\Phi = \frac{dN}{d\sigma} \quad [\text{cm}^{-2}]. \quad (4)$$

Each particle interacts with matter differently. Electrons and positrons lose energy either by ionization due to collisions or by the production of *Bremsstrahlung*, which is an electromagnetic radiation emitted when the charged particles are decelerated in electromagnetic fields. Photons, on the other hand, interact on the basis of three mechanisms: photoelectric effect dominating at low energies, Compton scattering at intermediate energies and finally pair production at higher energies. The behaviour of hadrons passing through matter depends on their charge [44]. At low energies, protons interact with lattice atoms through electrostatic forces. Neutrons, however, interact only with nuclei causing elastic collisions or nuclear reactions at high energies. The variety of particles and interactions can be compared by observing radiation induced changes in matter, which scale linearly with the amount of energy created by displacing collisions (*Non Ionizing Energy Loss (NIEL)* hypothesis) [11, 34]. It is useful to compare the damage caused by irradiation with the damage that would have been caused by 1 MeV neutrons. The equivalent 1 MeV neutron fluence is defined by

$$\Phi_{eq} = \kappa \Phi = \kappa \int \phi(E) dE, \quad (5)$$

where κ is a hardness factor (see [34]) and $\phi(E)$ are individual energy spectra of particles.

2.2 Dosimetry at the LHCb Experiment

2.2.1 Polymer-Alanine Dosimeters

The enantiomer L- α -alanine $\text{CH}_3\text{CH}(\text{NH}_2)\text{COOH}$ is an amino acid that releases one ammonia radical cation [37] and becomes the only stable radical of alanine at room temperature when exposed to radiation (Figure 4). Electron paramagnetic resonance (EPR) can be used in order to measure the number of alanine radicals. It has been found that the concentration of alanine radicals is proportional to the absorbed dose of radiation [55, 44].

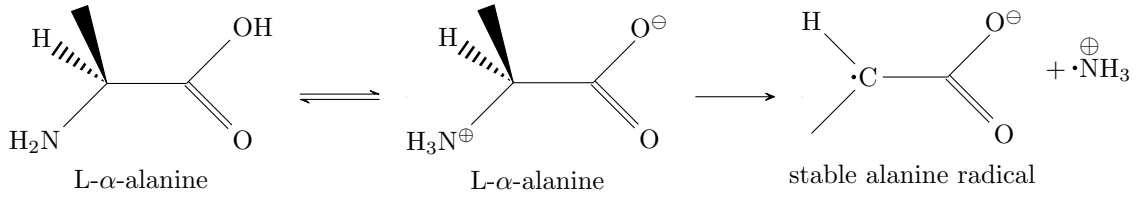


Figure 4: Creation of a stable alanine radical from L- α -alanine exposed to radiation.

The alanine dosimeters used at CERN are small tubes consisting of 67% alanine powder and 33% binding polymer. In the LHCb Experiment they are the most reliable source of information about the absorbed dose. However, the biggest obstacle is that it is impossible to collect data in real time. Each tube has to be temporarily removed and brought to the readout equipment every time the measurement takes place [29]. This limits the readout procedure to be performed once a year, as it is not possible to do it during beam operation, because the dosimeters cannot be accessed.

2.2.2 RADiation-sensitive Field-Effect Transistor

In 1970 Poch and Holmes-Siedle [22] published results presenting an electronically-measured, integrating dosimetric system based on measuring space charge in the silicon-dioxide portion of metal-oxide-semiconductor, field-effect transistor (MOSFET) (Figure 6a). Implementation of an insulator allowed to respond to doses of ionizing radiation. When irradiated, a space charge being proportional to the number of electron-hole pairs is built up. This comes from the fact that applying electric field to the insulator, which is irradiated by high-energy radiation, results in excitation of carriers that start to flow in the conduction band. Some of the electrons and holes are trapped while flowing (Figure 5a), however the number is not the same. The accumulated charge can remain for a long time and still be proportional to the number of electron-hole pairs unless the accelerating field reduces in time (Figure 5b). This new transistor has been called RadFET (RADiation-sensitive Field-Effect Transistor) and soon found to be useful in unmanned satellites run by European Space Agency [24], clinical dosimetry and finally in high-energy physics.

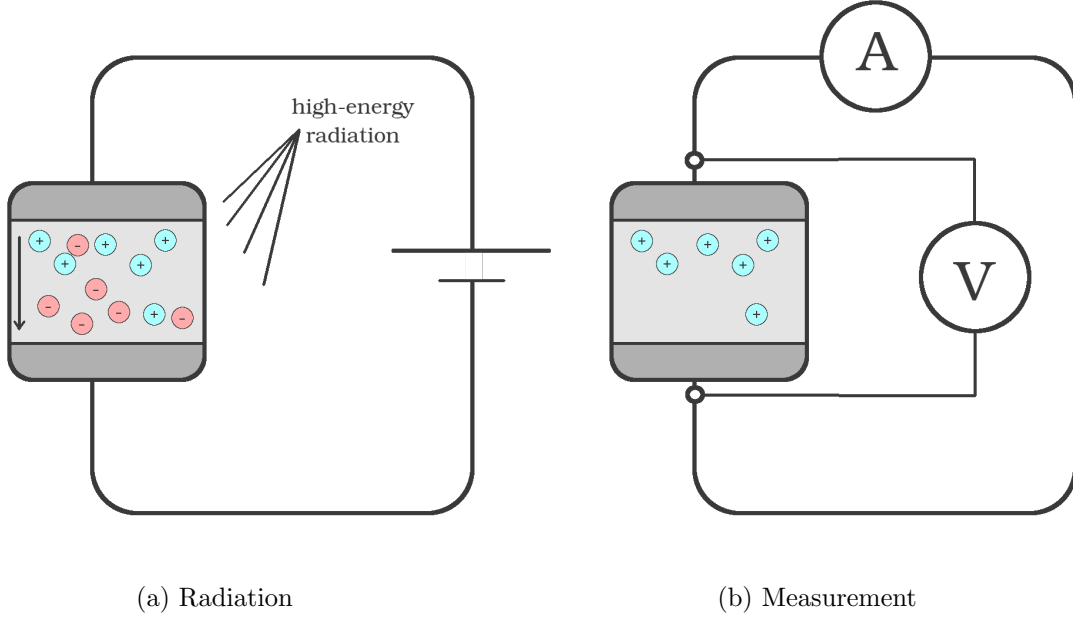


Figure 5: Space charge accumulation in an insulator. The electrodes are metal and silicon.

A trapped positive charge in RadFET alters the transistor gate *threshold voltage* V_{th}

$$V_{th} = V_{th0} - \frac{\sum Q_i}{C_{ox}} = V_{th0} + \Delta V_{th} \quad (6)$$

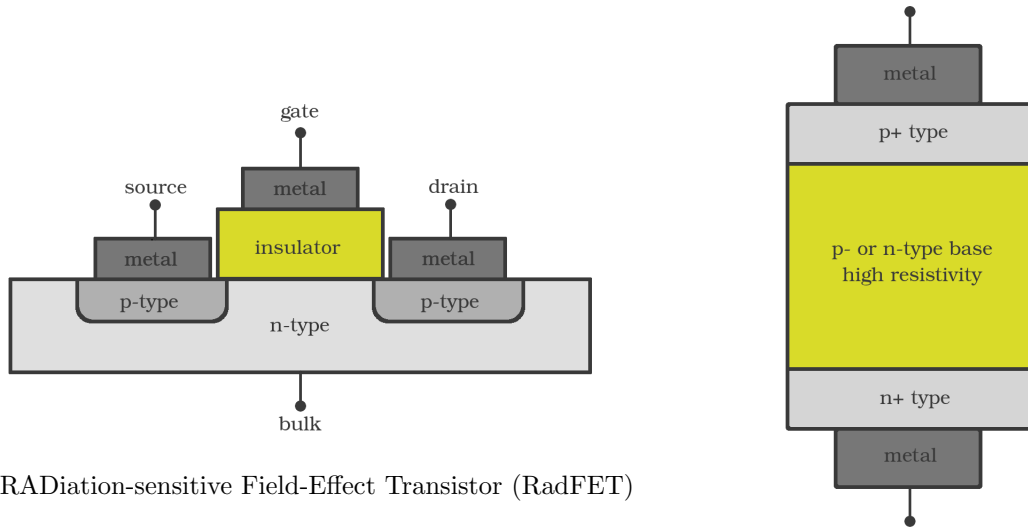
where V_{th0} is the initial threshold voltage (*offset* before irradiation), C_{ox} is the gate capacitance and $\sum Q_i$ is the sum of charges trapped in the gate oxide layer and the charge created at the oxide-semiconductor interface (see [47]). In practice, ΔV_{th} is the only measured parameter. The relation (eq. 6) strongly depends on the electric field in the oxide during irradiation and the thickness of the gate silicon oxide insulator. Thin oxide RadFET ($0.1 \mu\text{m}$) is designed to measure doses ranging from 10 Gy to 10000 Gy [42]. The thicker the oxide layer is, the more sensitive to small doses the RadFET becomes. In principle, thick layers guarantee a bigger number of interactions and capacitive effect of trapped charges. These effects can be expressed using the responsivity r , which is denoted by mV/cGy, and is roughly proportional to the square of the thickness of the oxide layer [24].

There are two regions where RadFET operates differently [23]: *dynamic* range and *saturating* range. The first one is linear on log-log plot of dose vs threshold voltage (Figure 7, 8). The empirical formula for the linear range can be represented by the power law [23, 24]:

Power law as a calibration curve for RADFETs

$$\Delta V_{th} = a \cdot D^b, \quad (7)$$

where a and b are calibration parameters. There is a smooth transition between dynamic



(a) RADiation-sensitive Field-Effect Transistor (RadFET)

(b) Forward biased p-i-n diode

Figure 6: Scheme of a RADiation-sensitive Field-Effect Transistor (RadFET) and a forward biased p-i-n diode.

and saturating range. The bigger the values of threshold voltage are, the more the increment of dose falls off turning into a saturating range. Holmes [22] suggested that this sub-linearity can be described using

$$\Delta V_{th} = \alpha V_I (1 - e^{-\beta D}), \quad (8)$$

where parameters α and β depend on the trapping parameters, thickness of the silicon oxide layer and the space charge location. V_I is the irradiation bias. If $V_I \neq 0$, then a constant electric field E is applied to the silicon oxide layer through the gate electrode. Applying positive bias results in increasing hole trapping on the other side of the metal. The closer the space charge is to the semiconductor the higher the sensitivity of the RadFET is. In the case of high dose exposure, a zero bias mode ($V_I = 0$) is preferable. Not only does that postpone the saturating range, but it also helps to avoid installing an external power supply.

A dynamic range of the dosimeter is the number of decades of dose between the lowest detectable values of response and the beginning of sub-linearity. In LHCb, two types of RadFET have been installed with different dynamic ranges. The LAAS (Laboratory of Analysis and Architecture of Systems) [29, 30, 42] is a RadFET with a thick silicon oxide layer, dedicated to measuring low doses ranging from 1 mGy to 10 Gy. On the other hand, the REM sensor (Radiation Experiment and Monitors) [16, 29, 42] is equipped with a thin silicon oxide layer able to measure doses from 1 to 10 kGy. All parameters were presented in Table 1.

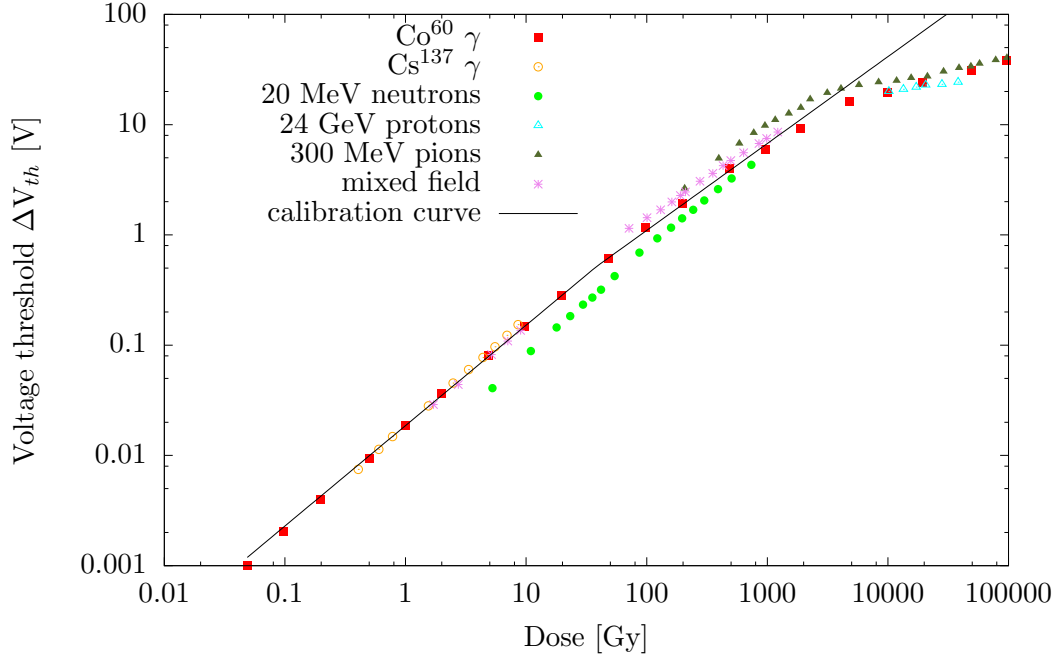


Figure 7: Reconstruction of the plot obtained by F. Ravotti [44] depicting calibration results^a for the REM RadFET.

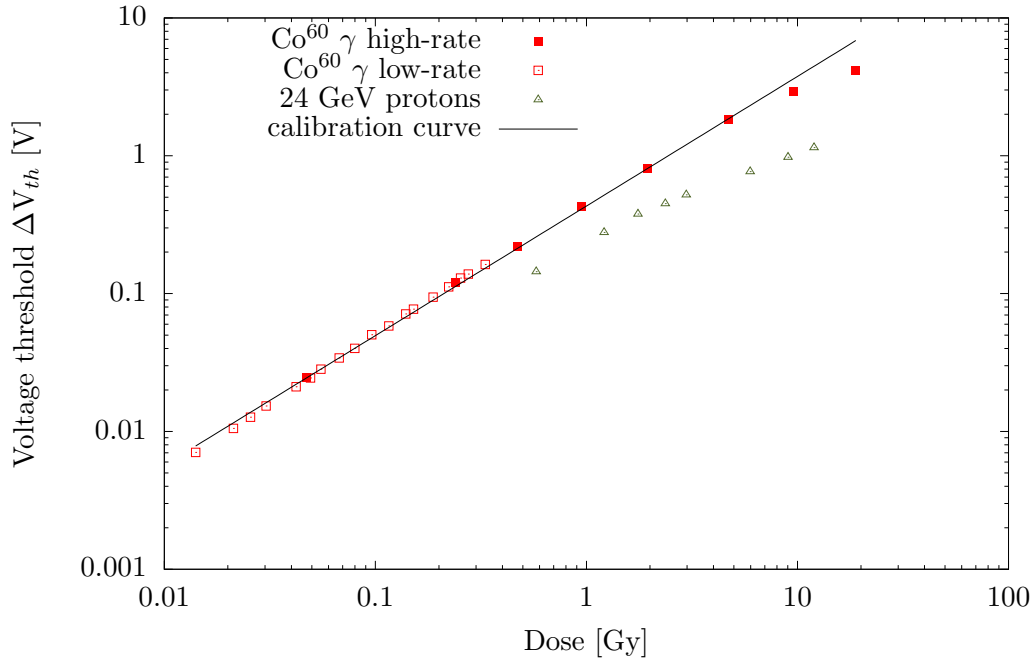


Figure 8: Reconstruction of the plot obtained by F. Ravotti [44] depicting calibration results^b for the LAAS RadFET.

^aThe data was taken using different sources of radiation: natural isotopes (10^{13} Bq Co^{60} coming from irradiation facility Hi-Activ in the UK; 740 GBq Cs^{137} in the Gamma Irradiation Facility at CERN, Geneva), neutrons and protons (cyclotron in Louvain-la-Neuve, Belgium [43] and IRRAD facilities at CERN in Geneva [19], Switzerland) and pions from the πE1 cyclotron (Villigen, Switzerland).

^b Co^{60} experimental data was taken by the producer and by the irradiation facility at the Montpellier University II. The 23 GeV proton beam was delivered by the IRRAD1 facility at CERN.

The biggest obstacle in calibrating the RADFETs is that the relation between V_{th} and a measured dose (7) is purely empirical. Moreover, in normal irradiation facilities it is very difficult to recreate the radiation environment at LHCb. Therefore, it is crucial to simulate the radiation response of RadFET sensors under conditions which reflect the original one as much as possible. F.Ravotti in [44] compared different sources of radiation for REM and LAAS and plotted them together (Figure 7, 8).

There are several mechanisms (apart from the lowered sensitivity in zero-bias mode) that may significantly disturb the measurement. *Annealing of the space charge* appears due to the neutralization of the charge in the silicon oxide layer by direct tunneling of electrons. It is mostly visible after the radiation exposure. The *build-up of oxide-semiconductor interface*, also called reverse annealing, is an increase in ΔV_{th} after irradiation. *The read-time instability* is caused by the rearrangement of charges formed during the irradiation. It is explained by the existence of slow interface states that lie in the silicon oxide layer and become very volatile when accessed by the readout system. The RADFETs are also subject to the *electronic noise* and depend on the *ambient temperature*.

	REM	LAAS
Origin	Oxford, United Kingdom	Toulouse, France
SiO ₂ layer length s_{ox} [nm]	250 (type K)	1600
Initial threshold volt. V_{th0} [V]	3.3 ± 0.3	2.2 ± 0.3
Gate to body voltage	zero bias	zero bias
Readout current [μ A]	160	100
Dynamic range [Gy]	1.0 – 10000.0	0.001 – 10.0

Table 1: Parameters of RADFETs used at the LHCb Experiment [42, 44].

The calibration curves using data from Figure 7 and 8 are piecewise linear functions of dose. For the REM dosimeter the calibration relation is

Calibration curve for the REM RadFET

$$\Delta V_{th}(D) = \begin{cases} 0.01864 \cdot D^{0.91072} & \text{if } D < 40\text{Gy} \\ 0.02921 \cdot D^{0.78778} & \text{if } D \geq 40\text{Gy} \end{cases} \quad (9)$$

In a similar manner, the calibration curve for the LAAS dosimeter is

Calibration curve for the LAAS RadFET

$$\Delta V_{th}(D) = 0.432 \cdot D^{0.9404} \quad (10)$$

2.2.3 High-resistivity forward biased p-i-n diodes

A p-i-n diode is an ordinary p-n diode with an undoped, intrinsic semiconductor placed in between p-layer and n-layer (Figure 6b) [29, 44, 49]. In general, p-i-n diodes

	CMRP	BPW	LSBD Si-2
Origin	Wollongong, AU	-	Prague, CZ
Base length [mm]	1.0	0.210	2.5
Base resistivity [$\text{k}\Omega \cdot \text{cm}$]	10.0	2.5	100.0
Initial forward volt. V_{F0} [V]	1.2 ± 0.3	0.564 ± 0.055	1.8-2.5
Exposure Mode	zero bias	zero bias	zero bias
Readout current [μA]	1000.0	1000.0	25000.0
Φ_{eq} range [cm^{-2}]	$10^8 \div 2.0 \cdot 10^{12}$	$2 \cdot 10^{12} \div 5 \cdot 10^{14}$	up to $2 \cdot 10^{11}$

Table 2: Parameters of the p-i-n diodes of the LHCb Experiment [44, 42].

have a base made out of high-resistivity Si ($\geq \text{k}\Omega \cdot \text{cm}$) or low-resistivity Si ($< \text{k}\Omega \cdot \text{cm}$). In the case of high resistivity and in the conditions of low injections (small current), the concentration of carriers in the base region and, as a result, resistivity depends on the particle fluence:

$$\rho = \rho_0 \cdot e^{\Phi/K_p}, \quad (11)$$

where ρ_0 is the initial equilibrium resistivity of Si before irradiation, Φ is the particle fluence and K_p is the coefficient ranging from 400 to 3000 cm^{-2} [6]. This clearly shows that the diodes have a hadron-dependent increase in the voltage drop (forward voltage ΔV_F) between p- and n- terminals. The calibration curve including forward voltage for diodes is given by

Calibration curve for p-i-n diodes

$$\Delta V_F = c \cdot k \cdot \Phi = c \cdot \Phi_{eq}, \quad (12)$$

where c is the calibration parameter and Φ_{eq} is an equivalent 1 MeV neutron fluence (eq. 5).

Three kinds of diodes are used in the LHCb Experiment covering different ranges of fluence. The first one, produced in the Center of Medical Radiation Physics (CMRP) at the University of Wollongong in Australia covers the range from 10^8 to $2 \cdot 10^{12} \text{ cm}^{-2}$. The CMRP diode has an n-type base of 1 mm length and a resistivity of $10 \text{ k}\Omega \cdot \text{cm}$. BPW 34FS is a diode produced by Siemens Semiconductor Group designed to cover high fluences from $2 \cdot 10^{12}$ to $4 \cdot 10^{14} \text{ cm}^{-2}$. The base of BPW 34FS has $210 \mu\text{m}$ and resistivity $2.5 \text{ k}\Omega \cdot \text{cm}$. LSBD Si-2 diodes were manufactured in the ‘Institute for Research, Production and Application of Radioisotopes’ in Prague in the Czech Republic by ion implementation of boron and phosphorus into the initial n-type silicon monocrystal using the floating zone method [39]. All parameters of the diodes are presented in Table 2.

The calibration curves for both diodes were obtained using different radiation sources listed below the plots (Figure 9, 10a and 11). The calibration parameters are:

Calibration curve for the BPW diode

$$\Phi_{eq} = 9.1 \cdot 10^{12} \cdot \Delta V_F \quad [cm^2/V] \quad (13)$$

Calibration curve for the CMRP diode

$$\Phi_{eq} = 1.7 \cdot 10^{11} \cdot \Delta V_F \quad [cm^2/V] \quad (14)$$

Calibration curve for the Si-2 diode

$$\Phi_{eq} = 2.7 \cdot 10^{10} \cdot \Delta V_F \quad [cm^2/V] \quad (15)$$

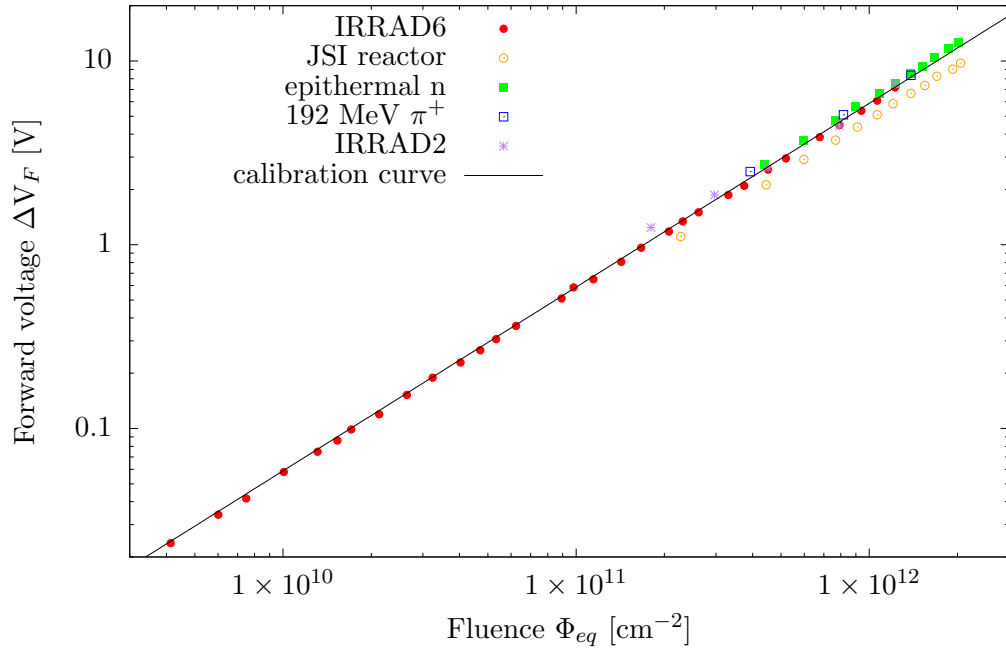


Figure 9: Reconstruction of the plot obtained by Ravotti [44] depicting calibration results^c for the CMRP diode with a readout current of 1 mA.

^cThe data was taken from JSI TRIGA Mark II reactor, IRRAD2 facility at CERN, in the 192 MeV pion beam at PSI- π E1 and in the mixed-hadron field from CERN-IRRAD6.

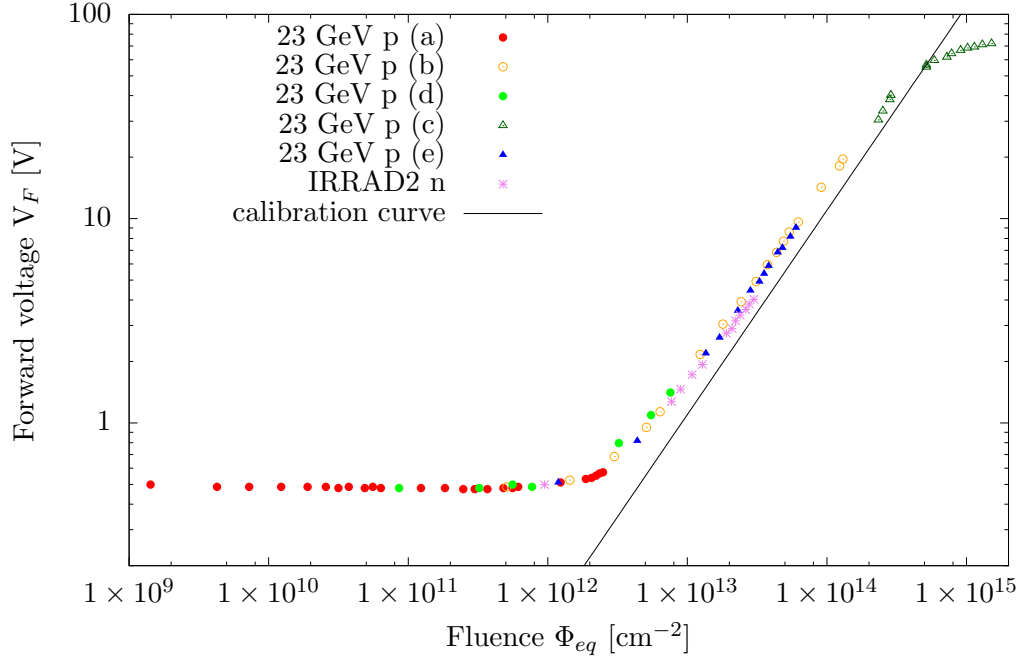


Figure 10: Reconstruction of the plot obtained by Ravotti [44] depicting calibration results^d for the BPW diode with a readout current of 1 mA.

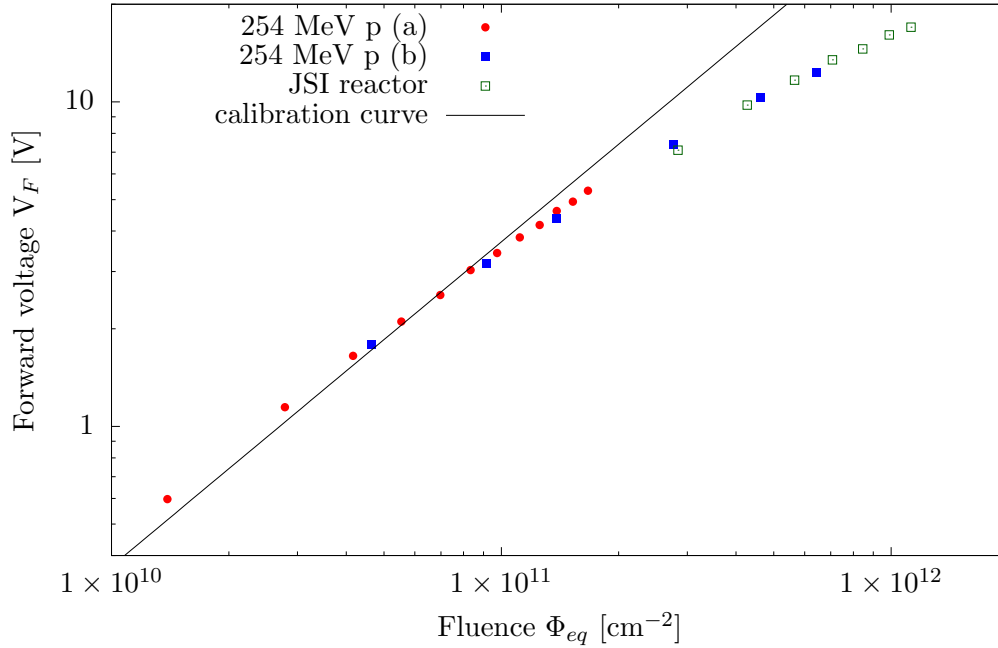


Figure 11: Reconstruction of the plot obtained by Ravotti [44] depicting calibration results^e for the LSBD Si-2 diode with a readout current of 25 mA.

^dThe experimental data was taken in the 23 GeV beam of the IRRAD1 facility at CERN with proton rates: (a) $2.4 \cdot 10^{11} \text{ cm}^{-2}\text{h}^{-1}$ (b) $3.7 \cdot 10^{12} \text{ cm}^{-2}\text{h}^{-1}$ (c) $8.1 \cdot 10^{12} \text{ cm}^{-2}\text{h}^{-1}$, (d) $2 \cdot 10^{13} \text{ cm}^{-2}\text{h}^{-1}$, (e) $2.5 \cdot 10^{13} \text{ cm}^{-2}\text{h}^{-1}$ and in the neutron beam at the IRRAD2 facility.

^eThe experimental data was taken from JSI reactor and in the 254 MeV beam of the PIF facility at PSI with proton rates: (a) $10^8 \text{ cm}^{-2}\text{s}^{-1}$ and (b) $2.5 \cdot 10^8 \text{ cm}^{-2}\text{s}^{-1}$.

2.3 Online system

2.3.1 Architecture

The Online System [1, 31, 35] in LHCb is responsible for the acquisition of data, the distribution of trigger data and the control, configuration and monitoring of the whole experiment. The architecture of the online system consists of three main components:

- Data Acquisition system (DAQ),
- Time and Fast Control System (TFC) and
- Experiment Control System (ECS).

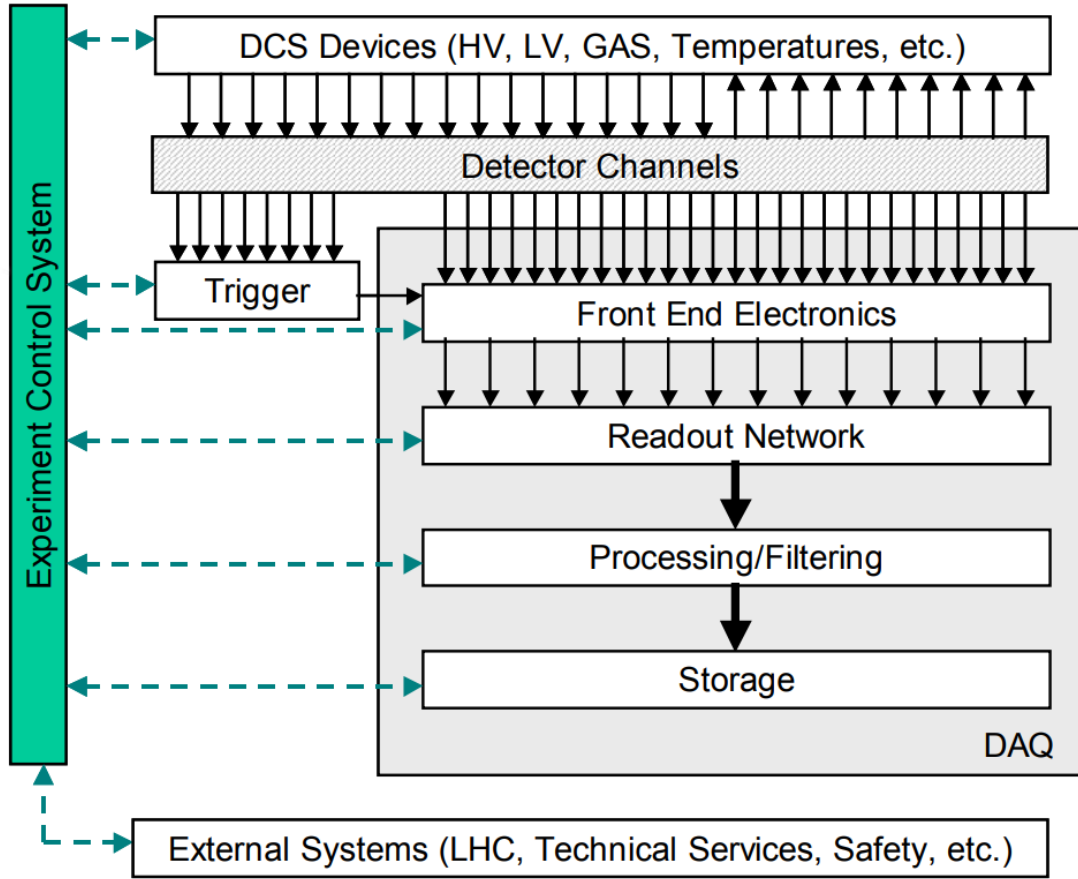


Figure 12: Scope of the Experiment Control System (ECS). ECS controls and monitors many systems: DAQ, DCS, External Systems etc.

The main purpose of the DAQ system is to collect, process and transport data to permanent storage. It is composed of many electronic boards and chips, which have to be initialized, configured, monitored and to operate under the requirements of the experiment at any given point. There are two main types of electronics: electronic boards placed close to the detector in the radiation area and boards that operate outside the radiation zone. ATLAS Embedded Local Monitoring Box (ELMB) [3] has been chosen because of its

radiation tolerance. ELMB was originally designed as an I/O device for analog and digital values. It contains micro-controllers that can be programmed to execute user's code. Two micro-controllers allow the ELMB to work under radiation by resetting each other in case of corruption of data due to Single Effect Upsets (SEUs).

The TFC system supervises all stages of the data readout from the front-end electronics, the online processing farm, synchronous resets and fast control command. The heart of the system, the Readout Supervisor, synchronizes trigger decisions and beam-synchronous commands to the LHC clock and transmits the signal provided by LHC.

Finally, the ECS system (Figure 12) controls and monitors the operational state of the LHCb detector ensuring that it is correctly configured and running properly. This includes the control of the DAQ, the detector hardware, the event-filter farm, the environment at the experimental site, the communication with systems from the outside etc. The ECS software is based on a commercial SCADA (Supervisory Control and Data Acquisition) system. Current standard of SCADA for LHCb is WinCC OA provided by Siemens. WinCC OA provides tools useful for building the ECS system itself like archiving interface, communication between distributed components, graphical libraries to build operation user interfaces and an alarm system. LHCb is a member of the Joint COntrols Project (JCOP) [7] collaboration, which aims to provide a common software framework and guidelines, as well as components for the common equipment in all the experiments and the accelerator.

2.3.2 WinCC OA

Projects in WinCC OA [15] are managed through the console panel (Figure 14). It controls the functional modules of the project called *managers*. Each manager runs as a separate process. The roles of the most important managers are shown in Figure 13.

The central processing unit in WinCC OA is called Event Manager. It holds the current image of all process variables in the memory. It is also a central data distributor and executes alert handling. The RDB Manager provides the interface to the Oracle database [36]. It handles the data parameterization of the application and archives the values and alerts.

WinCC OA provides an internal language Control (CTRL) that helps to implement user code and handles the treatment of data as well as it implements user interfaces and alert triggers. CTRL is a procedural higher-level language that has largely the same syntax as ANSI-C.

WinCC OA developed its own syntax for values and states that correspond to a particular device. They are called *datapoint elements* (DPE) and lie inside *datapoint type* structures. A datapoint type is an object definition of a structured datapoint as a computerized representation of a real device. For debug purposes and quick access, DPEs can be accessed through a PARA module (PARAMetrization tool) (Figure 15). They are

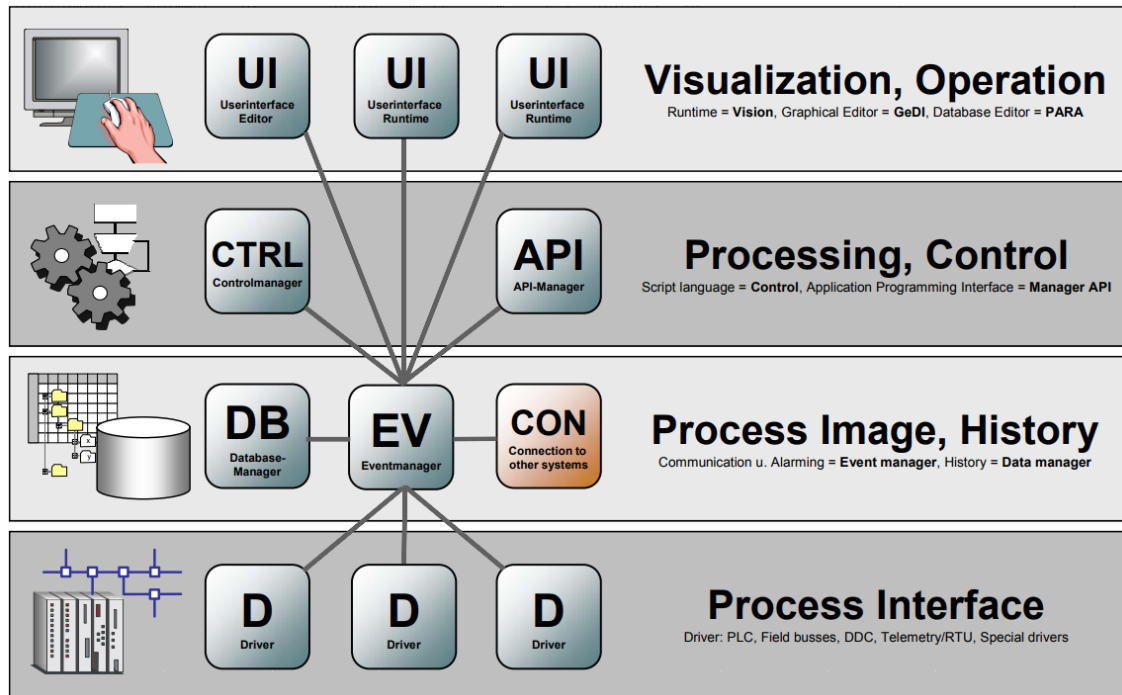


Figure 13: WinCC OA function-specific managers design. All managers are linked to the Event Manager and displayed in the console panel (Figure 14.)

displayed in a tree-like structure. The PARA tool is also a database editor that allows to create new datapoint elements, change their current type or remove them completely. It is also possible to specify which datapoint elements should be archived i.e. stored in the Oracle database. Custom process displays can be created and edited with the aid of GRaphical EDitor (GEDi).

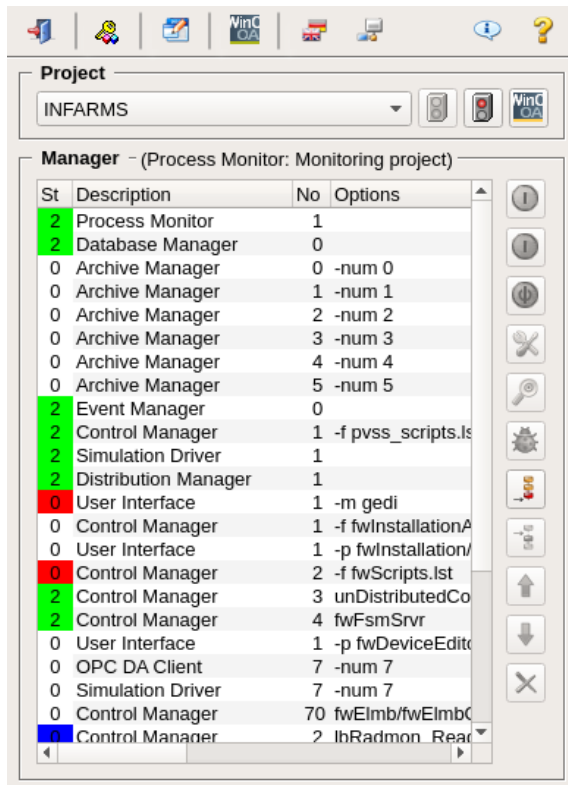


Figure 14: WinCC OA console panel. It lists all managers of the project.

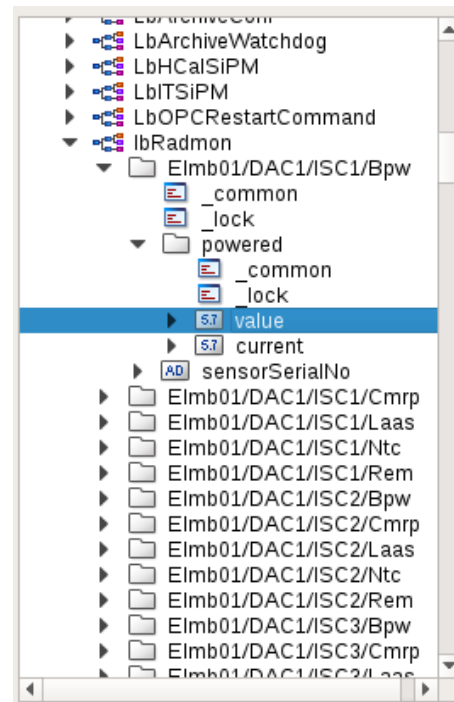


Figure 15: Part of the PARA module of the LHCb Active Radiation Monitoring System with visible tree-like structure of the datapoint elements.

2.3.3 Functions of the Online System

The Online System [31] was designed to operate in the environment depicted as a context diagram in Figure 16. It shows all external components whose requirements must be met by the Online System.

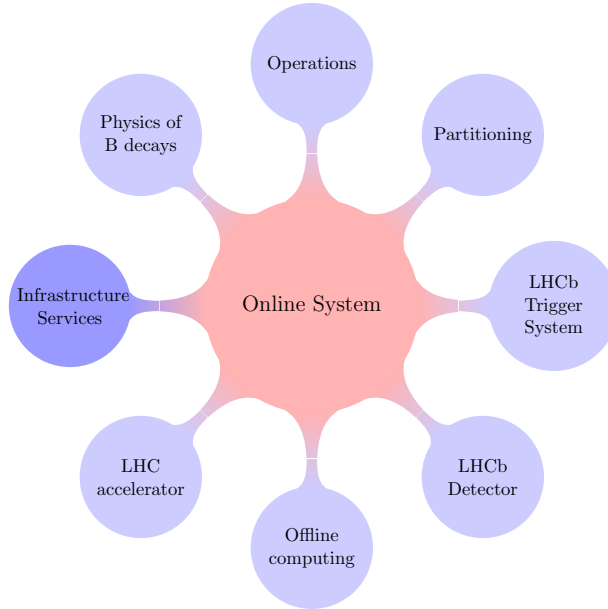


Figure 16: Context diagram showing the environmental requirements of the LHCb Online System.

‘CP violation’ in hadronic systems originating from b-quarks is studied by selecting only a very small fraction of all B mesons produced for further analysis offline. This can be done only if a relatively high level of background, originating mostly from uninteresting B decays, is preserved. Therefore, the Online System is expected to record as many events as possible (Physics requirement). An average total event size of 100kB was set, based on estimating the average detector occupancies (LHCb detector requirement). Moreover, independent sets of complex trigger algorithms, based on a pattern recognition code, are required to differentiate events containing B mesons from the background. Each of those algorithms requires a lot of CPU power to be executed after the event building stage (LHCb Trigger System requirement).

A small shift crew of two people present in the control room has to be provided with user-friendly interfaces that are also self-explanatory and network-based (Experiment Operations requirement). The Online System can also be partitioned, i.e. some parts of the experiment can be split. These partitions can be operated concurrently (Partitioning requirement).

The Online System must also ensure that the information coming from the LHC accelerator about particle intensities, currents etc. is interlocked with LHCb operations and that the detector is always in a safe operational state. It must also coordinate and configure the experiment according to the states of the LHC accelerator (LHC accelerator

requirement). Data coming from physics need to be stored on permanent media and remain accessible from any institute in the LHCb collaboration. Hence, the Online System has to be capable of sending the output to the computer center safely and provide support for offline computing and analysis (Offline computing requirement).

Finally, the Online System provides information interchange between the LHCb Experiment and the infrastructure services that comprise the power distribution system, the cooling and ventilation system, safety warnings etc. It also gives information about the environmental parameters such as: temperature, humidity and radiation levels (via radiation monitoring systems).

2.4 LHCb Active Radiation Monitoring System

LHCb Active Radiation Monitoring System (LbARMS) is a part of the Online System that provides information about radiation levels i.e. absorbed dose and particle fluence inside the the LHCb Experiment. Twenty-eight boxes (Figure 17) were placed in strategic positions throughout the whole experiment. Each box consists of four sensors: two p-i-n diodes used to measure particle fluence and two RADFETs for the absorbed dose. The initial set of sensors for each of the boxes as of 2011 was: CMRP, BPW, REM and LAAS.

The sensors of the LbARMS system were placed in different parts of the experiment. The detector layout of the sensors is depicted in Figure 18 as a projection on the xz -plane. Eight sensors were placed on the surface of the M1 station that is perpendicular to the z -axis. Half of them were placed along x -axis and another half along the y -axis starting from the center of the M1 station (Figure 19). Finally, last four sets of sensors were distributed throughout the cavern (Figure 20).



Figure 17: The Active Radiation Monitoring set of sensors with a serial number 11. The box of Polymer-Alanine Dosimeter is on the left and the box containing CMRP and BPW p-i-n diodes with REM and LAAS RADFETs is on the right.

Data acquisition in the LbARMS system takes place in the online control system project (INFARMS) written in WinCC OA. So far, the online project was only capable of reading out correctly calibrated voltages. In order to obtain the results in Gy and cm^{-2} for the RADFETs and p-i-n diodes respectively, the online results had to be downloaded and converted offline using the correction algorithm.

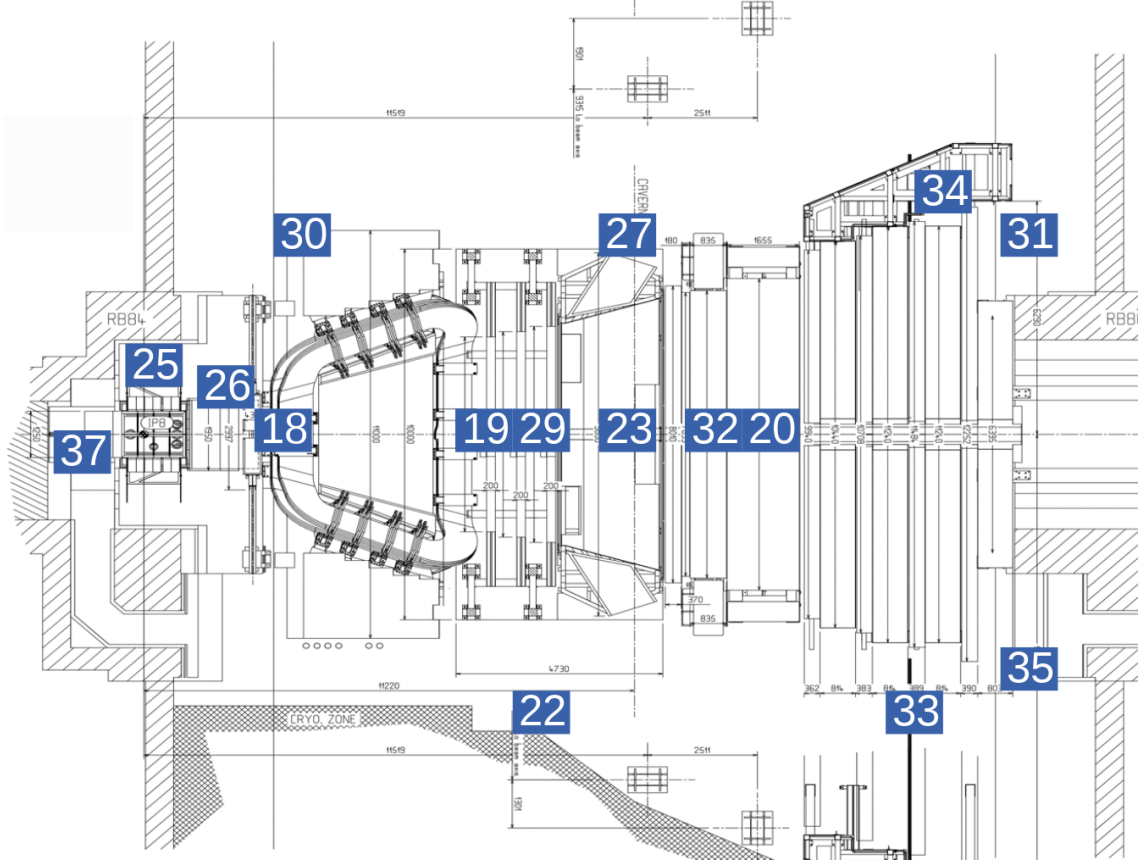


Figure 18: The detector layout of the radiation sensors. Each set is depicted as a blue rectangle with a serial number on it.

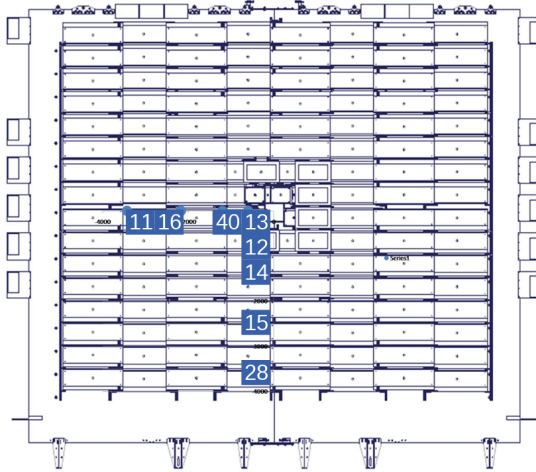


Figure 19: The M1 station layout of the radiation sensors. Each set is depicted as a blue rectangle with a serial number on it. The station is perpendicular to the z -axis that goes along the beamline in the detector layout in Figure 18.

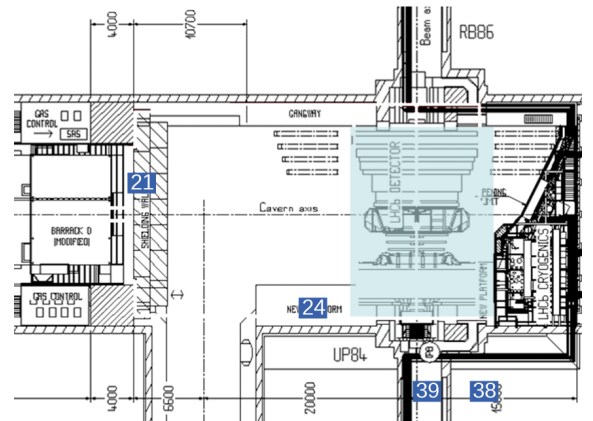


Figure 20: The cavern layout of the radiation sensors. Each set is depicted as a blue rectangle with a serial number on it. The detector layout (Figure 18) is highlighted with a light blue rectangle.

2.4.1 Correction algorithm

In 2012 V. Battista [4] devised a correction algorithm that consists of five major steps: loading and splitting the CSV files containing voltages and integrated luminosity (fetched previously from the database), evaluating the initial offset voltage, removing the negative temperature influence on the signal, correcting the noise, spikes and drifts in the signal and finally converting voltage to particle fluence and absorbed dose.

The initial offset of the sensor is estimated by taking an arithmetic mean of a number n of measurements starting with the first readout. The negative temperature influence is then incorporated in the results using the formula

$$\Delta V = V - V_0 + A \cdot (20 - T), \quad (16)$$

where: T is the temperature and A is an experimental parameter selected for each sensor individually. Afterwards, the algorithm converts voltage to dose and particle fluence using the calibration curves obtained by F. Ravotti in equations 9, 10 and 12.

The electronic noise and other environmental factors described in previous sections are subsequently filtered out using a simple formula based on first and second moments of the signal. The measurement is substituted with the former one, whenever its value does not fit in the statistical pattern. The strength of the filter is determined by the additional parameters, set for each dosimeter separately.

Eventually, the drops in the signal caused by the inevitable annealing are added back to the signal. In practice, the drops are only detectable when the beam is not present in the accelerator and hence, no increase in the signal is predicted. The no-beam periods are determined by checking as to whether the integrated luminosity reached a plateau or not.

It is very difficult to automate the algorithm in its current form due to the number of input parameters that were estimated after collecting data for a couple of years. This cannot be done, however, for a new sensor, whose statistical pattern is yet unknown. Moreover, tuning up the noise detection results in losing the original shape of the signal, whose accuracy is crucial during the drift correction step. The bigger the difference in signal is, the bigger the final error becomes due to the fact that it has a pile-up effect.

The code of the correction algorithm had to be improved in order to optimize its performance. The correction takes around 15 to 20 minutes for each new measurement. This is mainly caused by the size of the input files (especially luminosity that is measured every 3 seconds), but there are also some nested loops that could have been removed.

2.4.2 Online project

The online project of the LHCb Active Radiation Monitoring System (INFARMS) is one of the infrastructure services of the Online System. It uses the ELMBs to acquire raw voltage from p-i-n diodes and RADFETs. Raw values are collected by the Online System and archived in the Oracle database. The datapoint elements naming follows a pattern:

INFARMS:Elmb##/DAC#/ISC#/dosimeter_name.powered.value,

which matches the physical connections of the sensors on the ELMB read-out system. They can be then accessed through the PARA module (Figure 15) by their ELMB, ISC and DAC number. Another panel is responsible for displaying voltages in a table (Figure 22). There is a possibility to display the layout of the sensors distributed throughout the cavern, the LHCb detector (Figure 23) and the muon station. The measurement history can be easily viewed and plotted in the ECS, but in practice it can only be traced back to a recent date (Figure 21), because the process of loading the values from the database takes a lot of time.

In 2015 J. Faruggia [17] tried to incorporate the correction algorithm described in the previous section into the INFARMS project. As mentioned in the previous section, this could not be done for the correction algorithm of that form. Loading all values from the database is a time consuming task. The best option for the algorithm is to do the computations in close to real time, based solely on statistical information collected over time. It should be then used to filter out any unwanted flaws of the signal simultaneously with keeping its original shape intact.

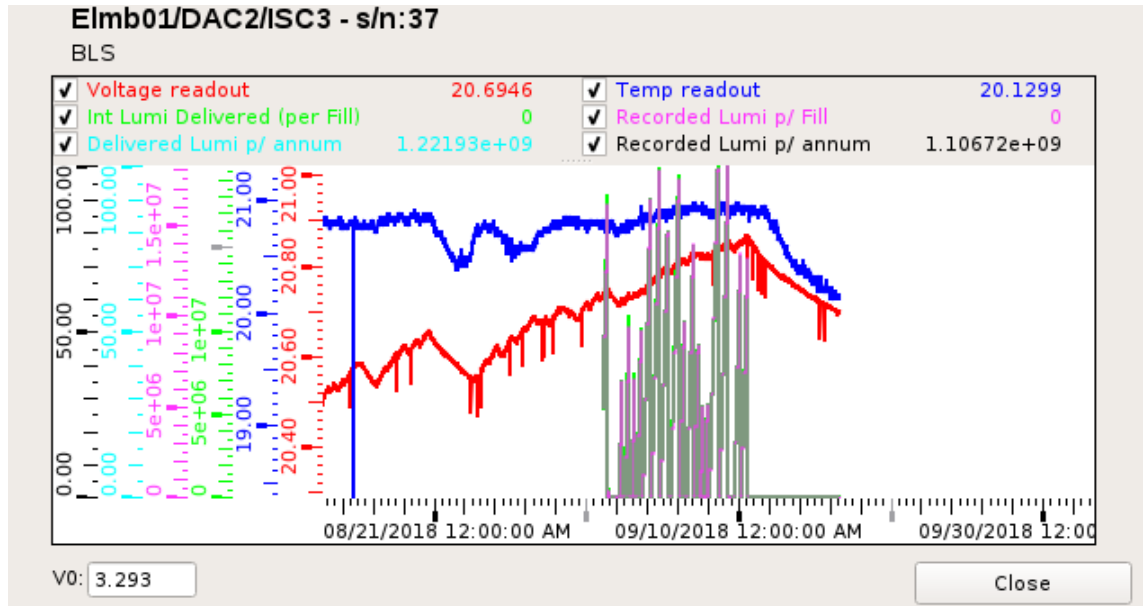


Figure 21: INFARMS panel used to display the integrated luminosity, voltage and temperature history for each device.

LHCb Active Radiation Monitoring System

Cavern				
Detector				
M1				
Table				

#1	#2	#3	#4	#5
6.789915	2.700269	26	RICH1 HPD, bottom A-side	INFARMS:Elmb01/DAC1/ISC1/Laas.powered
4.416951	1.223618	24	Balcony (VELO readout)	INFARMS:Elmb01/DAC1/ISC2/Laas.powered
7.814631	7.031618	25	VELO Repeater, V3B01	INFARMS:Elmb01/DAC1/ISC3/Laas.powered
8.82299	5.671622	18	RICH1 Exit	INFARMS:Elmb01/DAC2/ISC1/Laas.powered
6.554933	1.540847	30	TT Service Boxes, Magnet A Side	INFARMS:Elmb01/DAC2/ISC2/Laas.powered
27.5	1.845074	37	BLS	INFARMS:Elmb01/DAC2/ISC3/Laas.powered
2.267707	1.250469	38	Tunnel Calo Module (Wall Side)	INFARMS:Elmb02/DAC1/ISC1/Laas.powered
2.589149	1.393568	39	Tunnel Calo Module (Beam Side)	INFARMS:Elmb02/DAC1/ISC2/Laas.powered
4.500113	4.839584	0	INFARMS:Elmb02/DAC1/ISC3/Laas.	INFARMS:Elmb02/DAC1/ISC3/Laas.powered
0.126302	1.353704	0	INFARMS:Elmb02/DAC2/ISC1/Laas.	INFARMS:Elmb02/DAC2/ISC1/Laas.powered
27.5	27.5	0	INFARMS:Elmb02/DAC2/ISC2/Laas.	INFARMS:Elmb02/DAC2/ISC2/Laas.powered
27.5	27.5	0	INFARMS:Elmb02/DAC2/ISC3/Laas.	INFARMS:Elmb02/DAC2/ISC3/Laas.powered
0.128403	1.332298	0	INFARMS:Elmb03/DAC1/ISC1/Laas.	INFARMS:Elmb03/DAC1/ISC1/Laas.powered
27.5	17.437476	0	INFARMS:Elmb03/DAC1/ISC2/Laas.	INFARMS:Elmb03/DAC1/ISC2/Laas.powered
27.5	27.5	0	INFARMS:Elmb03/DAC1/ISC3/Laas.	INFARMS:Elmb03/DAC1/ISC3/Laas.powered

Figure 22: INFARMS main panel that lists the most recent voltage measurements.

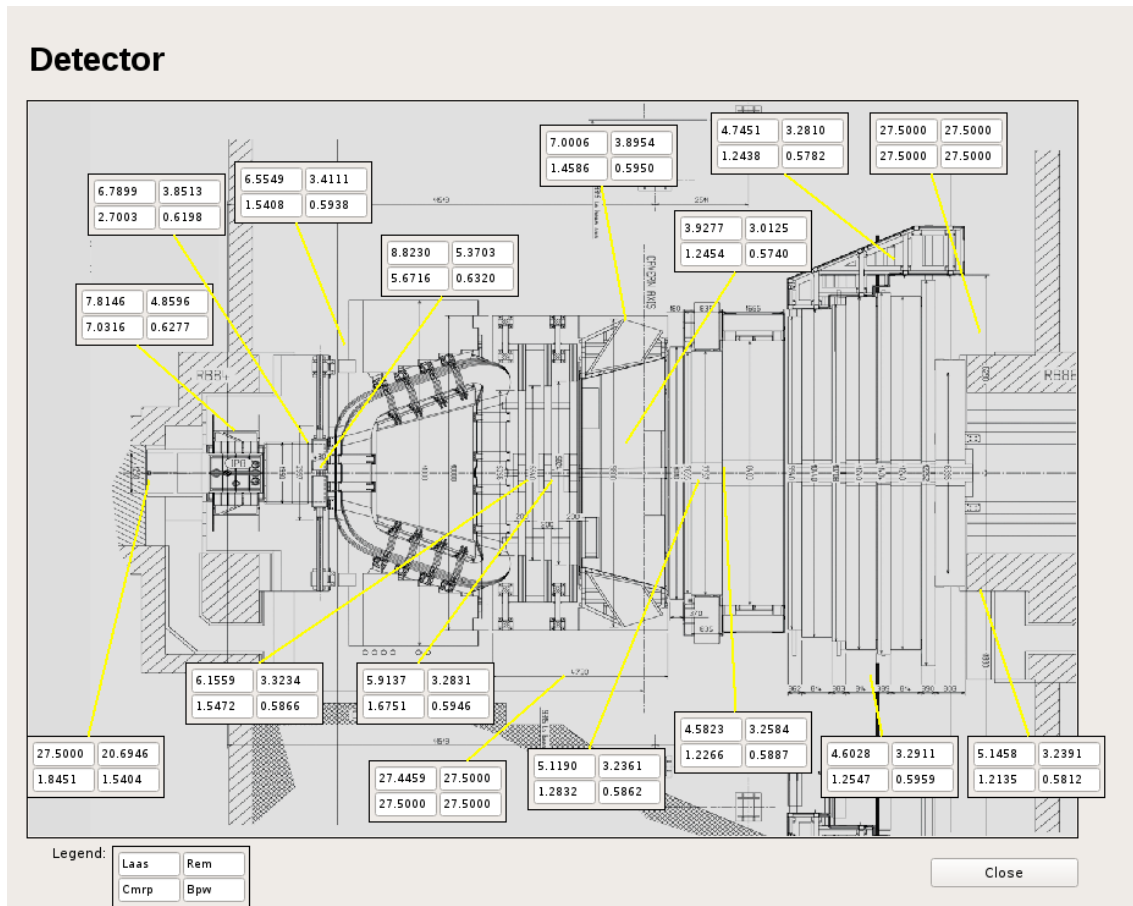


Figure 23: INFARMS panel with a detector layout used to indicate the location of some devices.

3 Theoretical introduction to Kalman filtering

3.1 Dynamic systems

Active radiation dosimeters, as other electronic devices produced nowadays, whose performance might be easily corrupted with environmental factors, need to go through post-processing algorithms to improve their accuracy. In general, it is usually possible to determine the *input* of such a device and measure its *output* [9]. The empirical formula can be usually written with the aid of differential equations. The *input equation* is usually written as follows

$$\dot{\xi}(t) = f(\xi(t), u(t), t) \quad (17)$$

where:

- $t \in \mathbb{R}$ is the *time* variable;
- $\xi(t) \in \mathbb{R}^{n \times 1}$ is the *state* of the system, i.e. the minimum amount of information needed about the past behaviour of the system in order to describe its future behaviour;
- $u(t) \in \mathbb{R}^{m \times 1}$ is the *input* variable.

Let $\zeta(t) \in \mathbb{R}^{q \times 1}$ be a measurable output variable. Then, in a similar manner, the *output equation* can be represented as

$$\zeta(t) = g(\xi(t), u(t), t) . \quad (18)$$

Both equations together form a *nonlinear deterministic system*. In the linear case, the deterministic system looks like

$$\begin{cases} \dot{\xi}(t) &= \varphi(t)\xi(t) + \beta(t)u(t) \\ \zeta(t) &= \eta(t)\xi(t) \end{cases} , \quad (19)$$

where:

- $\varphi(t) \in \mathbb{R}^{n \times n}$ is the *state transition* matrix, which describes how one state is transformed into another;
- $\beta(t) \in \mathbb{R}^{n \times m}$ is the *input model* matrix,
- $\eta(t) \in \mathbb{R}^{q \times n}$ is the *observation model* matrix.

The well-known solution to the input equation using a nonsingular *transition matrix*

$\phi(t_2, t_1) \in \mathbb{R}^{n \times n}$ that indicates the transition from time t_1 to time t_2 , is

$$\xi(t) = \phi(t, t_0)\xi(t_0) + \int_{t_0}^t \phi(t, \tau)\beta(\tau)u(\tau) \, d\tau. \quad (20)$$

Moreover, if we let φ be constant, then

$$\phi(t, t_0) = e^{\varphi \cdot (t - t_0)} = \sum_{i=0}^{\infty} \frac{(\varphi \cdot (t - t_0))^i}{i!}. \quad (21)$$

The use of the transition matrix helps to describe the system in a time-discrete way. Since the signals were observed at equally spaced points in time and the model is linear, only the discrete and linear dynamic systems are considered as

$$\begin{cases} \xi_{k+1} &= F_k \xi_k + B_k u_k \\ \zeta_k &= H_k \xi_k \end{cases} \quad (22)$$

where the matrices F , B and H are analogous to their time-continuous counterparts.

In a deterministic system the transformation follows exactly the given process model. In real life, however, one may only be able to approximate the behaviour of the underlying system. The gap between the deterministic and empirical system is usually modeled as a *process noise*. The most convenient form of the process noise variable is a zero-mean multivariate normal distribution $v_k \sim \mathcal{N}(0, Q_k)$ with covariance Q_k . In case one is incapable of acquiring a precise measurement, a similar variable called *observation noise* $w_k \sim \mathcal{N}(0, R_k)$ can be added. The choice of a normal distribution comes from the idea that all the macroscopic random effects are, in fact, a superposition of random effects observable in micro-scale [28] (according to the central limit theorem). An equation incorporating this fact is called a *stochastic system* and can be represented as

$$\begin{cases} \theta_{k+1} &= F_k \theta_k + G_k v_k \\ \psi_k &= H_k \theta_k + w_k \end{cases}. \quad (23)$$

Finally, a composition of these two systems is usually called a *linear stochastic/deterministic system* [9] and can be formulated as follows

$$\begin{cases} x_{k+1} &= F_k x_k + B_k u_k + G_k v_k \\ z_k &= H_k x_k + w_k \end{cases} \quad (24)$$

where:

- $G_k \in \mathbb{R}^{n \times p}$ – the matrix that applies process noise to the state space,
- $x_k = \xi_k + \theta_k$,
- $z_k = \zeta_k + \psi_k$.

3.2 State space estimation

In the preceding subsection the state-vector model of a randomly excited linear system was developed. In order to make the stochastic system applicable, it is necessary to determine the formula for the *estimate* $\hat{x}$, which is a computed value of the state vector x , based upon the measurements z [10]. In general, optimal estimates minimize the estimation error. There are three ways of estimating the state space:

- *filtering* in which the current state vector is estimated based upon all past measurements,
- *prediction* where the estimation of a state vector takes place in the future and
- *smoothing* that estimates the state vector based on preceding and following measurements.

3.2.1 Filtering

In 1960 Rudolf E. Kalman [27, 28] formulated a filtering problem (Problem 1) based on the system (25). The solution can be acquired with the aid of two different estimators: *unbiased minimum variance estimator* (MVUE) or *maximum likelihood estimator* (MLE), both yielding the same solution in this particular case.

Problem 1 *Optimal estimation problem*

Let $k \in \mathbb{N}$. Given

- the observation set $Z = \{z_0, z_1, z_2, \dots, z_k\}$,
- the state x_k generated by

$$\begin{cases} x_k = F_k x_{k-1} + v_k \\ z_k = H_k x_k + w_k \end{cases} \quad \text{where} \quad \begin{aligned} v_k &\sim \mathcal{N}(0, Q) \\ w_k &\sim \mathcal{N}(0, R) \\ \text{cov}[v_i, w_j] &= 0 \quad \forall i, j \in \mathbb{N} \end{aligned}, \quad (25)$$

- with the initial condition

$$\begin{aligned} E(x_0) &= \bar{x}_0 \\ \text{cov}(x_0) &= \bar{P}_0 \end{aligned},$$

find an estimate $\hat{x}_{k|k-1}$ of a state x_k with the property that the expected squared error is minimized, i.e.

$$\min E[(\hat{x}_k - x_k)A(\hat{x}_k - x_k)^T] \quad (26)$$

where the choice of $A \in S_{++}$ is arbitrary.

Kalman suggested representing the estimate $\hat{x}_k$ in its own state space called a *canonical form of the optimal filter* that generates the best linear estimate of the system:

$$\begin{cases} \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k \tilde{z}_{k|k} \\ \tilde{z}_{k|k} &= z_k - H_k \hat{x}_{k|k-1} \end{cases}, \quad (27)$$

where K_k is for now an unspecified, time-varying matrix. The difference between the estimate and its corresponding state is called an *estimation error* and can be represented using

$$\tilde{x}_{k|k} = \hat{x}_{k|k} - x_k. \quad (28)$$

Combining equations (25), (27) and (28) yields an alternative expression for the estimation error

$$\tilde{x}_{k|k} = (I - K_k H_k) \tilde{x}_{k|k-1} + K_k w_k. \quad (29)$$

It is useful now to define the *error covariance matrix* as $P_{k|k} = \text{cov}[\tilde{x}_{k|k}]$. Expanding it using (29) and then applying covariance properties from (25) gives

$$\begin{aligned} P_{k|k} &= E[\tilde{x}_{k|k} \tilde{x}_{k|k}^T] = E\left[((I - K_k H_k) \tilde{x}_{k|k-1} + K_k w_k) ((I - K_k H_k) \tilde{x}_{k|k-1} + K_k w_k)^T \right] \\ &= E\left[(I - K_k H_k) \tilde{x}_{k|k-1} (\tilde{x}_{k|k-1}^T (I - K_k H_k)^T + w_k^T K_k^T) + \right. \\ &\quad \left. + K_k w_k (\tilde{x}_{k|k-1}^T (I - K_k H_k)^T + w_k^T K_k^T) \right] \\ &= (I - K_k H_k) P_{k-1|k-1} (I - K_k H_k)^T + K_k R_k K_k^T. \end{aligned}$$

The last thing to determine is the optimum choice of K_k in order to minimize a weighted scalar sum of the trace of the error covariance matrix. Therefore, the matrix A in a cost function $E[(\hat{x}_k - x_k) A (\hat{x}_k - x_k)^T]$ is set to an identity in order to get the minimization problem

$$\min \text{tr}(P_{k|k}).$$

Proposition 1 Let $B \in S$. Then $\frac{\partial}{\partial A} \text{tr}(ABA^T) = 2AB$.

Proof: See [13]. ■

The solution of the minimization problem is obtained by taking a partial derivative with respect to K_k and making it equal to 0

$$\frac{\partial \text{tr}(P_{k|k})}{\partial K_k} = -2(I - K_k H_k) P_{k|k-1} H_k^T + 2K_k R_k = 0.$$

Solving the equation with respect to K_k yields

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1}, \quad (30)$$

which is called a *Kalman gain matrix*. Using it again in equation (29) can simplify the error covariance matrix formula

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}. \quad (31)$$

Algorithm 1 Kalman filter

```

1: procedure KF( $\hat{x}_{k-1|k-1}, P_{k-1|k-1}, R_k$ )
2:    $\hat{x}_{k|k-1} \leftarrow F_k \hat{x}_{k-1|k-1}$  ▷ state estimate a priori
3:    $P_{k|k-1} \leftarrow F_k P_{k-1|k-1} F_k^T + Q$  ▷ error covariance a priori
4:    $\tilde{z}_k \leftarrow z_k - H_k \hat{x}_{k|k-1}$  ▷ innovation
5:    $S_k \leftarrow H_k P_{k|k-1} H_k^T + R_k$  ▷ innovation covariance
6:    $K_k \leftarrow P_{k|k-1} H_k^T S_k^{-1}$  ▷ optimal Kalman gain
7:    $\hat{x}_{k|k} \leftarrow \hat{x}_{k|k-1} + K_k \tilde{z}_k$  ▷ state estimate a posteriori
8:    $P_{k|k} \leftarrow (I - K_k H_k) P_{k|k-1}$  ▷ error covariance a posteriori
9: return  $\hat{x}_{k|k}, P_{k|k}$ 

```

Algorithm 1 provides a fully applicable recursive formula for estimating a set of observations. Kalman filter (KF) is one of the greatest discoveries in the history of statistical estimation theory [20]. It has been applied in the fields where a full control of complex dynamic systems is crucial. The discrete form of Algorithm 1 makes it easier to be applied in a computer program. Moreover, the recursive formula does not need to store past measurements that is crucial in the case of big data problems. Finally, with the use of probability functions, the Kalman filter is a sort of a learning tool. Although it is a very simple one, the Kalman filter can already state via its gain matrix, whether it should trust more the observations or the predictions.

On the other hand, as every tool, the Kalman filter has its limitations. First of all, it is only a mathematical tool, in the sense that it does not solve the physics by itself. Hence, it strongly depends on given state-transition, observation and noise models.

3.2.2 Smoothing

Kalman considered only a filtering and a predictive approach in his solution. In a non-real-time procedure, called *smoothing*, it is possible to estimate the state of a system at a certain timestep $\hat{x}_{k|N}$, where $k \in [0, N] \subset \mathbb{N}$ [10]. An optimal smoother is a superposition of two filters: a *forward recursion* filter and a *backward recursion* filter. There are three types of smoothing depending on the way of obtaining it:

- *fixed-interval smoothing*, in which the estimate $\hat{x}_{k|N}$ is established using the states varying from $k = 0$ to $k = N$,

- *fixed-point smoothing* where k is fixed and N is increased,
- *fixed-lag smoothing* with increasing N and fixed Δ for $\hat{x}_{N-\Delta|N}$.

Algorithm 2 Rauch-Tung-Striebel backward recursion [41]

```

1: procedure RTS( $\hat{x}_{k+1|N}, P_{k+1|N}, \hat{x}_{k|k}, P_{k|k}$ )
2:    $P_{k+1|k} \leftarrow F P_{k|k} F^T + Q$ 
3:    $J_k \leftarrow P_{k|k} F_k^T P_{k+1|k}^{-1}$ 
4:    $\hat{x}_{k|N} \leftarrow \hat{x}_{k|k} + J_k (\hat{x}_{k+1|N} - F_k \hat{x}_{k|k})$ 
5:    $P_{k|N} \leftarrow P_{k|k} + J_k (P_{k+1|N} - P_{k+1|k}) J_k^T$ 
6:   return  $\hat{x}_{k|N}, P_{k|N}$ 

```

Rauch, Tung and Striebel in 1965 invented a fixed-interval smoother based on the maximum likelihood estimation [41]. Let $Y = \{y_1, \dots, y_N\}$ be the set of random variables that have a joint probability function denoted by $p(Y) = p(y_1, \dots, y_N)$ and let $Y_A, Y_B \subset Y$ be such that $A, B \subset \{1 \dots N\}$ and $A \cap B = \emptyset$. Then the *conditional probability* [40] is defined by

$$p(Y_A, Y_B) = \frac{p(Y_A, Y_B)}{p(Y_B)}.$$

Theorem 1 *Bayes' theorem*

$$p(Y_A, Y_B) = \frac{p(Y_B|Y_A)p(Y_A)}{p(Y_B)}$$

For the filtering problem (Problem 1) the estimate $\hat{x}_{k|k}$ of x_k could be achieved following the MLE procedure by maximizing the conditional probability density function $p(x_k|Z_k)$. The same result is given by maximizing a logarithm of the density function using Bayes' theorem 1)

$$L(x_k, Z_k) = \log p(x_k|Z_k) = \log p(x_k, Z_k) - \log p(Z_k). \quad (32)$$

As a result, the log-likelihood for the estimates $\hat{x}_{k|N}$ and $\hat{x}_{k+1|N}$ is

$$\begin{aligned}
L(x_k, x_{k+1}, Z_N) &= \log p(x_k, x_{k+1}|Z_N) \\
&= \log p(x_k, x_{k+1}, z_{k+1}, \dots, z_N|Z_k)p(Z_k) \\
&= \log [p(z_{k+1}, \dots, z_N|x_{k+1})p(x_{k+1}|x_k)p(x_k|Z_k)p(Z_k)].
\end{aligned}$$

Assuming that $\hat{x}_{k|k}$ is already known from the solution of Problem 1, taking the probability density from multivariate Gaussian and getting rid of the terms that do not depend on x_k nor x_{k+1} , the maximum likelihood estimation problem is

$$\max_{x_k, x_{k+1}} \{-\|x_k - F_k x_k\|_{Q^{-1}}^2 - \|x_k - \hat{x}_{k|k}\|_{P_{k|k}^{-1}}^2\}. \quad (33)$$

Solving the maximization problem and using the matrix inversion lemma gives

$$\hat{x}_{k|N} = \hat{x}_{k|k} + J_k(\hat{x}_{k+1|N} - F_k \hat{x}_{k|k}), \quad (34)$$

where $J_k = P_{k|k} F_k^T P_{k+1|k}^{-1}$. The estimates $\hat{x}_{k|k}$ and $\hat{P}_{k|k}$ are taken from the Kalman forward recursion. The recursive formula is represented by Algorithm 2 and a complete Rauch-Tung-Striebel smoother by Algorithm 3.

Algorithm 3 Rauch-Tung-Striebel smoother [41]

- 1: **for** $k=1:N$ **do**
 - 2: $\hat{x}_{k|k}, P_{k|k} \leftarrow \text{KF}(\hat{x}_{k-1|k-1}, P_{k-1|k-1}, R_k)$ $\triangleright$ forward recursion using Kalman filter
 - 3: **for** $k=N-1:1$ **do**
 - 4: $\hat{x}_{k|N}, P_{k|N} \leftarrow \text{RTS}(\hat{x}_{k+1|N}, P_{k+1|N}, \hat{x}_{k|k}, P_{k|k})$ $\triangleright$ backward recursion
-

3.3 Estimating the state space without outliers

3.3.1 Filtering

The Kalman filter is an optimal estimator when the observation and process noise models are Gaussian. Some devices, although already having well-defined noise characteristics, may be subject to spontaneous disturbances that differ significantly from the pattern of normal distribution. These disturbances are called *outliers* [2]. Not only does the Kalman filter perform unpredictably in their presence, but they might also change the statistical parameters drastically. The Gaussian model fails, because due to its ‘lightweight tails’ (left and right parts of the probability density function are referred to as ‘tails’) it is incapable of verifying as to whether the measurement is an outlier or not.

There already have been many proposals for extending the Kalman filter such that it would be able to recognize and eliminate outlying measurements [46, 52, 53]. Some of them were even successfully tested on real-life devices. On the International Conference on Robotics and Automation in 2011 Gabriel Agamennoni et al. [2] presented an Outlier-Robust Kalman Filter (ORKF), which is a generalization of other authors’ work. In practice the ORKF inherits a vast majority of the KF concepts. The key difference is the observation noise model. In the KF the observation noise is an additive white noise that is constant and dictated by the observation noise covariance matrix R . The ORKF, on the other hand, allows the new observation error matrix Ω_k to be neither constant nor Gaussian. In principle, the observation noise becomes a hidden variable, taken from a Wishart distribution, i.e.

$$\Omega_k^{-1} \sim \mathcal{W}\left(\frac{R^{-1}}{s}, s\right).$$

The authors claim that the observation z_t is Student t-distributed with ‘heavy’ tails, possibly capable of finding the outliers. As a result, the condition probability density

function becomes

$$p(z_k|x_k) = \int_{\Omega_k > 0} p(z_k|x_k, \Omega_k) p(\Omega_k) d\Omega_k \propto (1 + \delta_k^T R^{-1} \delta_k / s)^{-\frac{s+1}{2}}, \quad (35)$$

where $\delta_k = z_k - Hx_k$. The key difference between the outlier-robust version and the standard Kalman Filter lies in the update step. The noise covariance R_k is replaced with its expected value Γ_k . Then,

$$\Omega_k^{-1}|Z \sim \mathcal{W}(\Omega_k^{-1}/(s+1), s+1),$$

with

$$\Gamma_k = \frac{sR + \langle \delta_k \delta_k^T \rangle}{s+1},$$

where

$$\langle \delta_k \delta_k^T \rangle = (z_k - H\hat{x}_{k|k})(z_k - H\hat{x}_{k|k})^T + HP_{k|k}H^T.$$

Γ_k is a convex combination of the noise covariance R_k and the expected outer product matrix. The ORKF reduces to the traditional Kalman filter in the limit, i.e.

$$\lim_{s \rightarrow \infty} \Gamma_k = R_k. \quad (36)$$

The above equations have been summarized in algorithm 4). The convergence is assessed using the innovation likelihood in eq. 35).

Algorithm 4 Outlier-Robust Kalman Filter [2]

```

1: procedure ORKF( $\hat{x}_{k-1|k-1}, P_{k-1|k-1}$ )
2:    $\hat{x}_{k|k-1} \leftarrow F_k \hat{x}_{k-1|k-1}$  ▷ state estimate a priori
3:    $P_{k|k-1} \leftarrow F_k P_{k-1|k-1} F_k^T + Q$  ▷ error covariance a priori
4:   while not converged do
5:      $S_k \leftarrow H_k P_{k|k-1} H_k^T + \Gamma_k$  ▷ innovation covariance
6:      $K_k \leftarrow P_{k|k-1} H_k^T S_k^{-1}$  ▷ optimal Kalman gain
7:      $\tilde{z}_k \leftarrow z_k - H \hat{x}_{k|k-1}$  ▷ innovation
8:      $\hat{x}_{k|k} \leftarrow \hat{x}_{k|k-1} + K_k \tilde{z}_k$  ▷ state estimate a posteriori
9:      $P_{k|k} \leftarrow (I - K_k H_k) P_{k|k-1}$  ▷ error covariance a posteriori in a Joseph form
10:     $\delta \leftarrow z_k - H \hat{x}_{k|k}$ 
11:     $\Gamma \leftarrow \frac{sR + \delta \delta^T + H P_{k|k} H^T}{s+1}$  ▷ update expected value of noise covariance
12:  return  $\hat{x}_{k|k}, P_{k|k}$ 
    
```

3.3.2 Smoothing

A very similar trick to that of the Kalman Filter extending to the smoothing algorithm could be applied to the Outlier-Robust Kalman Filter. G. Agamennoni et al. present it implemented inside an Expectation-Maximization (EM) algorithm. It alternates between the expectation part, which in this case is a result from a smoothing algorithm, and the

maximization part, in which parameters of the algorithm are estimated by computing the partial derivative of the log-likelihood and making it equal to zero [18]. The maximization step is, however, very complex and computationally demanding. Instead, only the expectation part is taken into consideration. It is similar to Algorithm 3, however, the expected value of the noise covariance Γ_k is used instead of R_k and updated in backward recursion. This is presented in Algorithm 5.

Algorithm 5 Outlier-Robust Kalman Smoother [2]

```

1: procedure ORKS( $\hat{x}_{k-1|N-1}, P_{k-1|N-1}$ )
2:   while not converged do
3:     for  $k=1:N$  do
4:        $\hat{x}_{k|k}, P_{k|k} \leftarrow \text{KF}(\hat{x}_{k-1|k-1}, P_{k-1|k-1}, \Gamma_k)$   $\triangleright$  forward recursion using KF
5:     for  $k=N-1:1$  do
6:        $\delta \leftarrow z_k - H\hat{x}_{k|N}$ 
7:        $\Gamma_k \leftarrow \frac{sR + \delta\delta^T + HP_{k|N}H^T}{s+1}$   $\triangleright$  update expected value of noise covariance
8:        $\hat{x}_{k|N}, P_{k|N} \leftarrow \text{RTS}(\hat{x}_{k+1|N}, P_{k+1|N}, \hat{x}_{k|k}, P_{k|k})$   $\triangleright$  backward RTS
9:   return  $\hat{x}_{k|N}, P_{k|N}$ 

```

4 New correction algorithm

A new correction algorithm can be divided into three major steps (Figure 24). Firstly, the voltage is smoothed out, which results in producing an estimate of the current measurement. The estimate is then sent to the drift correction step, where the negative effects caused by annealing during beam off periods are removed. The output is then transferred to the part where the voltage is converted to dose and particle fluence.

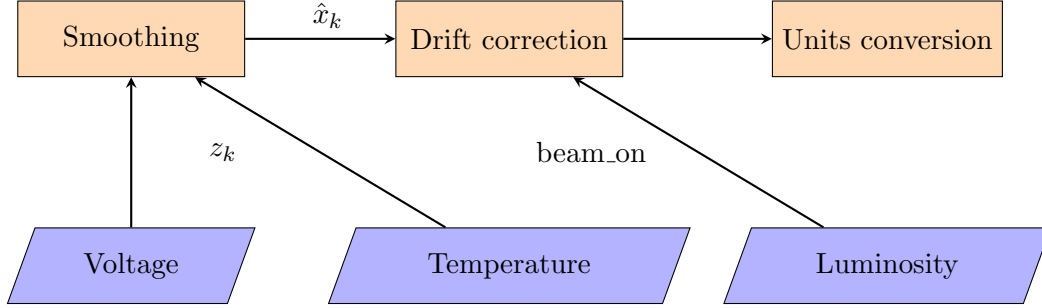


Figure 24: Correction algorithm flowchart.

In the algorithm, the Outlier-Robust Kalman Smoother (Algorithm 5) is used to smooth out the signal. However, it has to be properly modeled and calibrated in order to be useful in the LbARMS project.

4.1 Modeling the state space

Let $x(t) = (V(t) \dot{V}(t))^T$ be the state of the system, where $V(t)$ is a voltage function. The corresponding state-space representation is analogous to the system (19). The crucial part of the new model is to approximate the voltage threshold curve using a piecewise continuous function of time s.t.

$$\ddot{V}(t) = 0. \quad (37)$$

An example of a piecewise linear approximation is depicted in Figure 25. This results in having

$$\dot{x}(t) = \underbrace{\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}}_{\varphi} x(t).$$

In order to approximate the difference between the piecewise linear model and the real data, an input variable is added to the model

$$\dot{x}(t) = \varphi x(t).$$

Let Δt be a timestep between adjacent measurements. Then the solution can be acquired

using the transition matrix explained in the theoretical introduction (21)

$$x((k+1)\Delta t) = e^{\varphi(k+1)\Delta t}x(0).$$

Solving in a similar manner for $x(k\Delta t)$ and combining the equations together yields

$$x((k+1)\Delta t) = e^{\varphi\Delta t}x(k\Delta t).$$

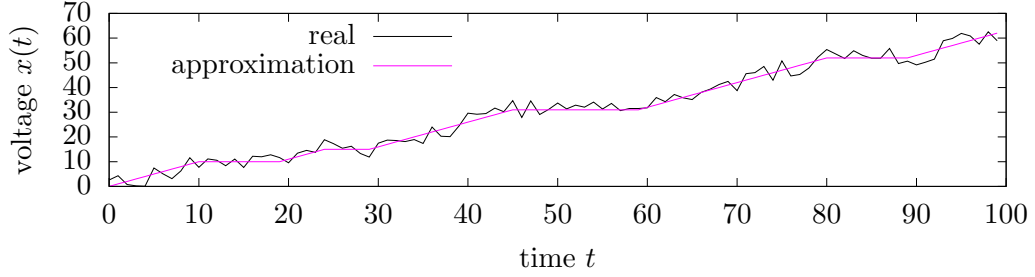


Figure 25: An example of a piecewise continuous function.

Matrix φ is nilpotent with degree 2, therefore an expansion of the exponent gives

$$e^{\varphi\Delta t} = \sum_{i=0}^{\infty} \frac{1}{i!} (\varphi\Delta t)^i = I + \varphi\Delta t = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} = F$$

The gap between the piecewise linear and real state model can be performed by approximating the second derivative $\ddot{V}(t)$ with a white-noise variable $v_k \sim \mathcal{N}(0, Q)$ as explained in the theoretical part (Problem 1). The approximation of $V((k+1)\Delta t)$ and $\dot{V}((k+1)\Delta t)$ using Taylor expansion gives

$$V((k+1)\Delta t) \approx V(k\Delta t) + \dot{V}(k\Delta t)\Delta t + \frac{1}{2}\ddot{V}(k\Delta t)(\Delta t)^2$$

$$\dot{V}((k+1)\Delta t) \approx \dot{V}(k\Delta t) + \ddot{V}(k\Delta t)\Delta t$$

The observation equation follows the same principles as in the theoretical part (Problem 1). Since only the voltage and not its derivative is measured, the observation matrix has the form $H = (1 \ 0)$. Each measurement is distorted by a zero-mean Student's t-distributed variable that models the standard measurement noise and occasionally popping outliers.

All the above assumptions can be summarized using the standard subscript notation:

1. the timestep between the measurements is constant i.e.

$$\Delta t = \text{const} > 0$$

2. the voltage is approximated by a piecewise linear model of a function, whose second

derivative with respect to time is zero i.e.

$$\ddot{V}(t) = 0,$$

- the state transformation error is approximated by sending the second derivative of voltage to a white-noise variable $v_k \sim \mathcal{N}(0, Q)$ giving

$$x_{k+1} = \underbrace{\begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}}_F x_k + v_k$$

with

$$Q = \begin{pmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{pmatrix} \begin{pmatrix} \frac{1}{2}\Delta t^2 & \Delta t \end{pmatrix} \sigma_v^2 = GG^T \sigma_v^2,$$

- voltage is the only measured variable,
- each measurement z_k is corrupted with an observation noise w_k , which is taken from a Student's t-distribution with s degrees of freedom giving

$$z_k = \underbrace{\begin{pmatrix} 1 & 0 \end{pmatrix}}_H x_k + w_k.$$

4.2 Calibrating the smoother

Kalman filter is not a fully independent filter and needs some tuning before it can be applied. It had be to made in such a way as to guarantee that it will fit each sensor's statistical pattern. In general, it can be assumed that there are two different patterns among them. The first pattern has a very a high signal-to-noise ratio due to the fact that the measurement is relatively high compared to the electronic noise. In this case, also the annealing effects are more distinct. On the other hand, there are sensors that are still below the lower limit of their dynamic range and as a result, noise and spikes influence the output significantly.

The tuning part consists of setting the initial conditions and parameters of the filter, which model its noise and precision. The initialization consists of setting:

- $\hat{x}_0 = (z_0 \ 0)^T$ – the initial state estimate,
- P_0 – the initial error covariance matrix.

The elements of the initial error covariance matrix are set to small values. The choice of the calibration parameters has been based only on the experimental performance of the filters. As a result, there exist three parameters, which have to be set:

- muting parameter* μ , which is controlled by σ_v and σ_w that describe the process and observation noise covariances and the relation between them is given by $\mu = \sigma_v/\sigma_w$;

- *degrees of freedom s* in the outlier-robust version of the Kalman filter, whose value influences the spike detection;
- *window N* or in other words a number of iterations in Rauch-Tung-Striebel forward and backward recursions, which specifies how many values from the ‘future’ influence the current measurement.

4.2.1 Muting parameter μ as a signal-to-noise ratio manager

It has been found that the relation between process and observation noise parameters itself is more important than setting their values independently. In order to give the algorithms more learning capabilities, a standard deviation of a raw voltage output is computed and put in place of σ_w parameter.

A comparison of seven different μ values was presented for both the ORKF (Figure 26) and the ORKS (Figure 27) algorithms. The degrees of freedom s was set to 2 and the window size N to 20. It is clearly visible that the bigger the muting parameter is, the more both algorithms resemble original measurements. On the other hand, low μ values smooth out the signal completely, keeping only its trend. The project’s purpose is to find a solution in the middle – the electronic noise and the spikes should be removed while keeping the signal’s shape intact. A value of $\mu = 0.075$ was chosen because it manages to meet this requirement for the majority of the sensors.

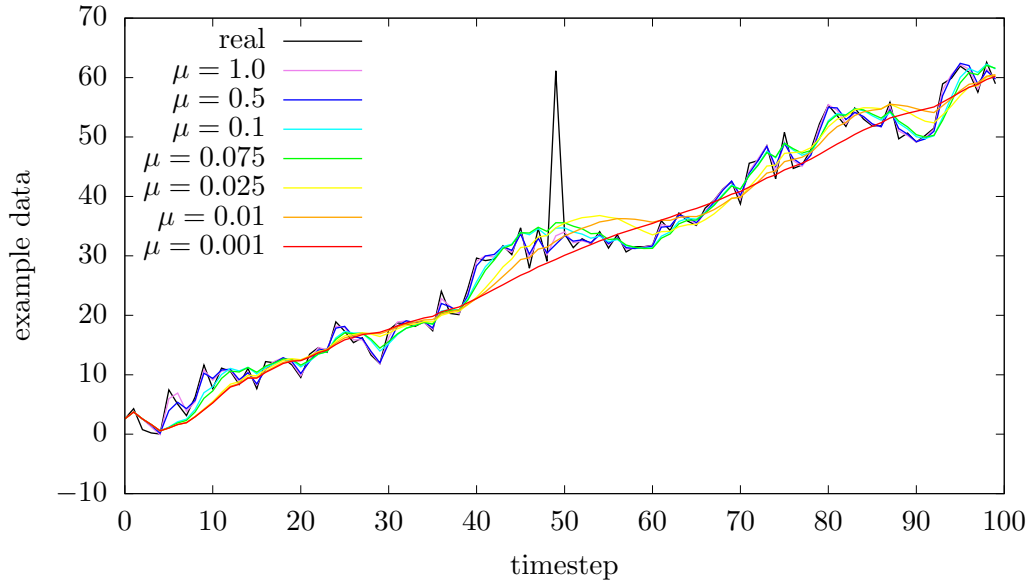


Figure 26: Testing variations of the muting parameter μ responsible for signal-to-noise ratio using Outlier-Robust Kalman Filter. The delay in the response to the signal is bigger with lower values of the muting parameter.

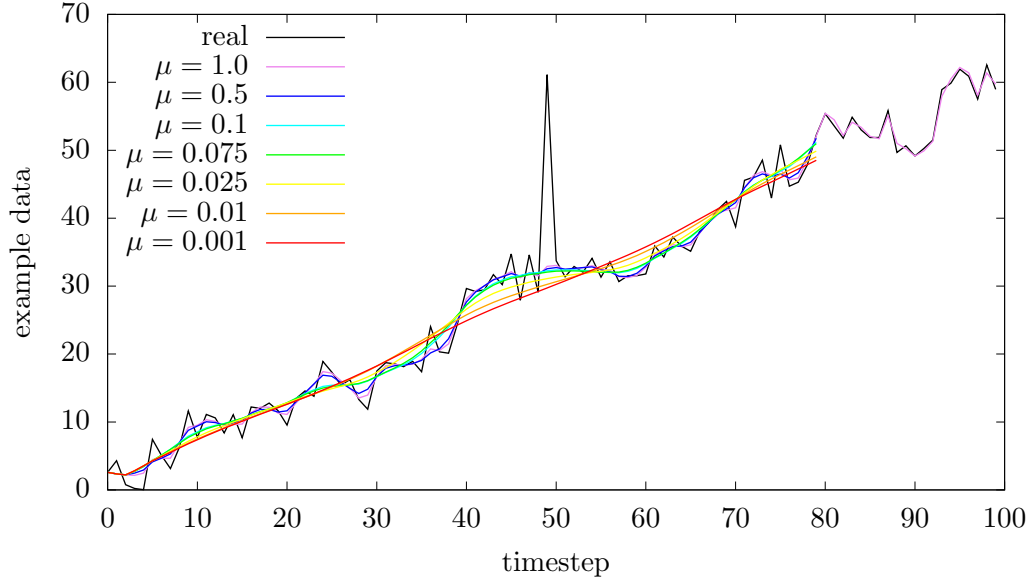


Figure 27: Testing muting parameter μ responsible for signal-to-noise ratio using Outlier-Robust Kalman Smoother. The ORKS removes the lag from the signal completely.

A drawback of this solution is that the ORKF with higher μ has a tendency to include a delay in the response to the signal that will be referred to as lag. It is due to the fact that ORKF only takes the history of the signal into account. On the other hand, the ORKS is capable of removing this lag by anticipating its behaviour from the subsequent measurements.

4.2.2 Degrees of freedom s for spike detection

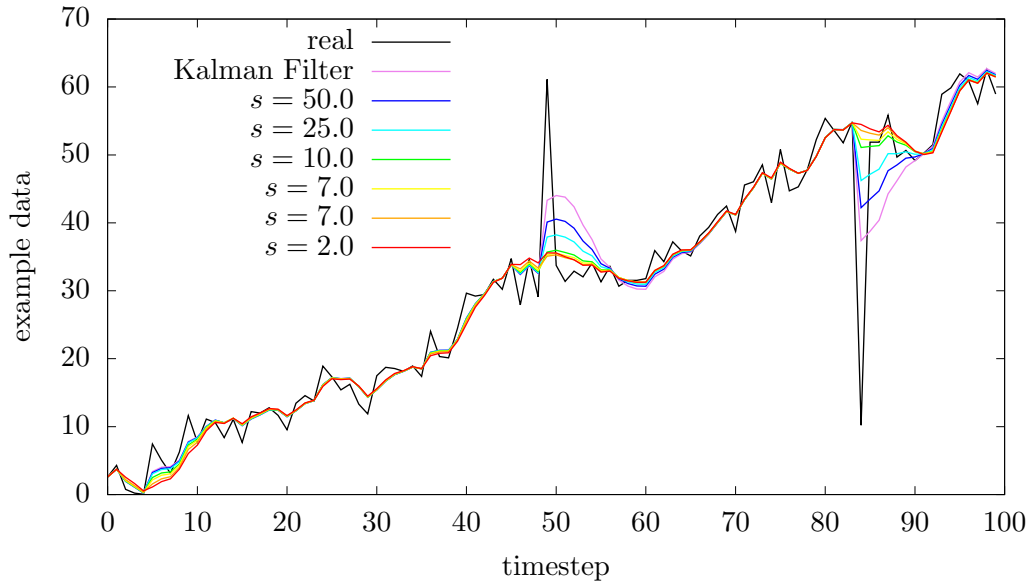


Figure 28: Testing the value of degrees of freedom s used for spike detection in the Outlier-Robust Kalman Filter.

It has been already seen that Kalman Filter itself does not model the outliers at all. The lack of a means to deal with large outliers has been the most important reason for choosing an algorithm, because a huge spike in the signal could be added permanently to the drift variable and hence it has to be detected in advance. As mentioned before, the ORKF is capable of detecting them by incorporating a new observation noise model. The most important parameter, whose value directly affects the strength of spike detection, is the degrees of freedom s . In equation (36) it was shown that the ORKF becomes a standard Kalman Filter as $s \rightarrow \infty$.

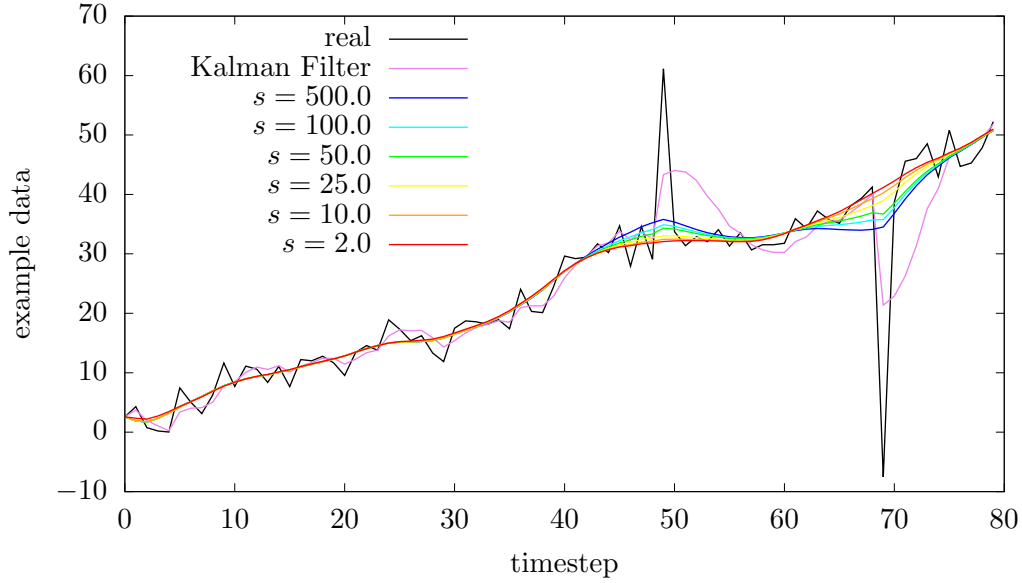


Figure 29: Testing the value of degrees of freedom s used for spike detection in the Outlier-Robust Kalman Smoother.

A set of six different values of s was chosen to find out how strong the spike detection needs to be. This was tested for both the ORKF (Figure 28) and the ORKS (Figure 29). It is clearly visible that the ORKF resembles the behaviour of the KF everywhere apart from the vicinity of the spike. There are two spikes generated in the data set. The bigger the parameter s is, the more the ORKF fails to anticipate the spike and the bigger the similarity between the ORKF and KF is. None of the single spikes are considered correct whatsoever. Hence, the value $s = 2.0$ was chosen as it is the lowest value possible with a mathematical sense behind it.

4.2.3 Window's size N

One additional parameter had to be set for the ORKS algorithm, stating how many subsequent values need to be taken into consideration. Inside the system, the ORKS operates only on values in the past in order to make it applicable in real-time. Therefore, the window's size N also implies how far in the past the actual correction performed by the ORKS algorithm is located.

The reason behind applying the smoothing algorithm is that the ORKF performs badly with spikes bigger than one measurement and with rapid changes of the dynamics. This is due to the fact that it is incapable of differentiating between real signal and an anomaly without the knowledge about the following measurements.

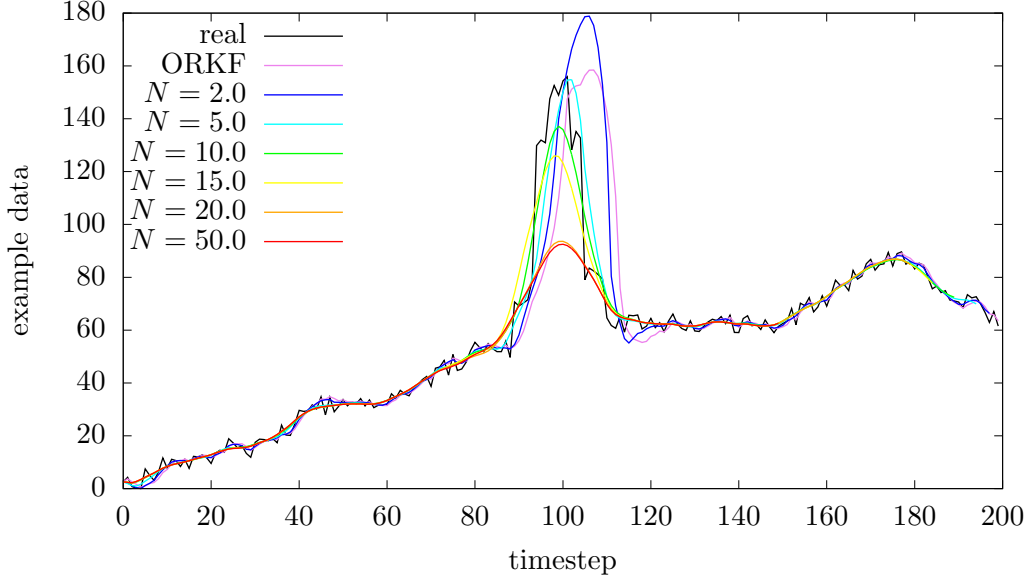


Figure 30: Testing different values of window's length N with a wide spike.

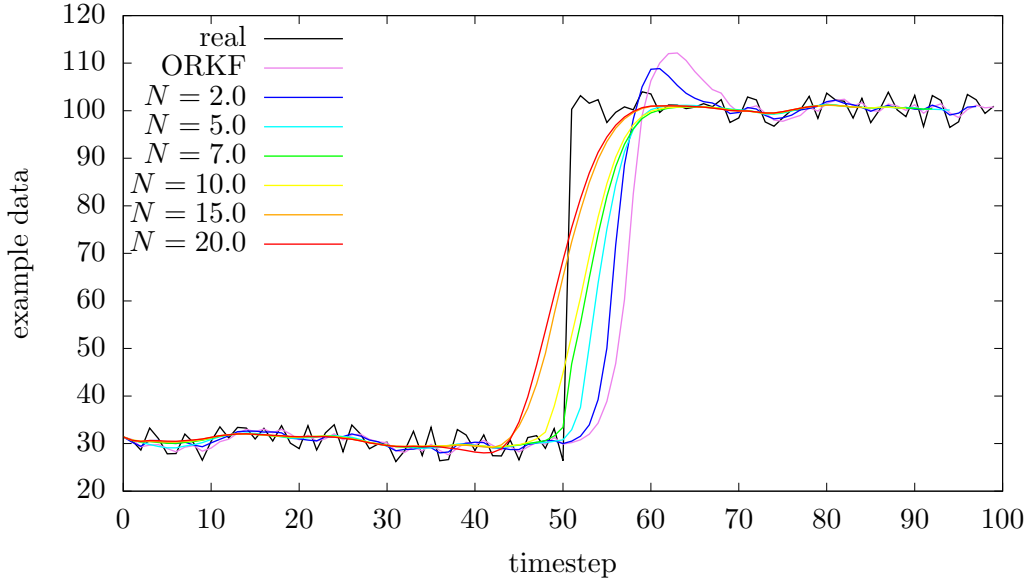


Figure 31: Testing different values of window's length N with a step function.

Six different values of N were compared in two different situations: a wide spike (Figure 30) and a step function (Figure 31). It is worth mentioning that the ORKS with $N = 0$ turns back into the ORKF algorithm. In both situations, a rapid fading of the delay 'bump' caused by the ORKF is observed with an increase in the value of N . Moreover, it is clearly visible that the signal's response moves back to reach its expected position with

high N .

The value of $N = 50$ was chosen as it models the step function accurately and fades large spikes sufficiently. The drawback of increasing this value would be a significant increase of the computation time of the algorithm.

4.3 Drift correction

All the sensors are subject to a significant amount of annealing effects as explained in the experimental setup section. Usually, the closer the voltage value is to the saturating range, the stronger the annealing effect is. It is characterized by a nonlinear drop in the value that looks similarly to an exponential decay. Whereas it is present throughout the whole timespan, it is only detectable during the beam-off period. In other words, no increase in the voltage is anticipated whenever the instantaneous luminosity drops to zero. This fact is used to assess a voltage drop, also called a *drift* of the signal, during the evaluation of the sensors.

In practice, when the integrated luminosity reaches a plateau, the difference between the last value before the plateau and the first value after beam operation resumes is added up to the following measurements. Additional conditions are added, which guarantee a nondecreasing shape of the voltage function. Thresholds for each sensor have to be set in order to skip the drift correction for voltages that do not exceed the lower limit of the dynamic range of the sensor.

The drift correction is a part of the correction algorithm, whose precise output strongly depends on the precise corrections made during the signal processing part. This is because the drift is an additive variable. The bigger the smoothing error is, the more the drift correction diverges from the correct value. This is why it is crucial to implement filters that have a high signal-to-noise ratio and do not introduce a lag to the output.

4.4 Conversion of units

The way the units are converted did not change much compared to the previous algorithm [4]. The first step is to compute the offset V_0 , which is subtracted from the voltage measurement, giving the voltage threshold ΔV_{th} for dosimeters and forward voltage ΔV_F for fluence sensors. The temperature dependence was also taken into account following suggestions of F. Ravotti [24]. This is optional, because the way the temperature influences the signal might not be the same for the sensors in the same batch. The formula for $\Delta V_{th}/\Delta V_F$ is the same as in (16).

The conversion of units to fluence and dose is performed using the calibration curves obtained by F. Ravotti in equations (7) and (12). In order to extend the range of calibration into and beyond the saturating range, calibration curves can be presented as piecewise-linear functions in the logarithmic scale, separated by breakpoints. In 1977 Chua and Kang [8] proposed a so-called canonical piecewise-linear representation given by

Theorem 4.1.

Theorem 4.1 *Canonical piecewise-linear representation* [8]

Any single-valued piecewise-linear function with at most σ breakpoints $\beta_1 < \beta_2 < \dots < \beta_\sigma$, can be represented by the expression:

$$y(x) = a + bx + \sum_{i=1}^{\sigma} c_i |x - \beta_i| \quad (38)$$

with: $b = \frac{J_1 + J_{\sigma+1}}{2}$, $c_i = \frac{J_{i+1} - J_i}{2}$, $a = y(0) - \sum_{i=1}^{\sigma} c_i |\beta_i|$ for $i = 1, 2, \dots, \sigma$ and J_i denoting the slope of the i -th constitutive linear segment in the piecewise-linear function.

In 2016 Jimenez-Fernandez V. [26] derived a general approximation of a piecewise-linear function. The formula comes from the smooth approximation using the integral of a sigmoid function presented by M. Schmidt et al. [48] in 2007 for the absolute-value function. A formal definition is expressed by Theorem 4.2.

Theorem 4.2 *Smooth-piecewise function* [26]

Any one-dimensional canonical piecewise-linear function that is characterized by L segments and σ breakpoints $\beta_1 < \beta_2 < \dots < \beta_\sigma$ can be transformed into *smooth-piecewise function* expressed as

$$y(x) = A + Bx + \sum_{i=1}^{\sigma} C_i \cdot \ln(1 + e^{-\alpha(x-\beta_i)}) \quad (39)$$

where $A = a - \sum_{i=1}^{\sigma} c_i \beta_i$, $B = b + \sum_{i=1}^{\sigma} c_i$, $C_i = \frac{2c_i}{\alpha}$ for $i = 1, 2, \dots, \sigma$ and the parameter α can be used to preserve a constant smoothness in the whole function domain or to define a specific smoothness grade α_i at any i -th breakpoint location as $\alpha_i = \frac{2c_i \ln(2)}{\delta}$ with δ being the deviation between the piecewise-linear and the smooth-piecewise functions at $x = \beta_i$.

The authors provide an example of the smooth transformation applied onto a characterization curve for a n-channel MOS transistor. Hence, the smooth-piecewise model was also chosen as a representation of calibration curves for the diodes and dosimeters described in this work, provided that enough validation data from other sources corresponding to the saturating range of the active sensors has been collected.

At least two linear segments are needed to include both the dynamic and the saturating ranges. Let a_d , a_s be the slopes and b_d , b_s be the intercepts of the linear functions fitted to the dynamic and saturating range respectively. Moreover, let s be the point where these two lines intersect each other. Then, using Theorem 4.2 a calibration

curve with a smooth transition between these two regions is given by:

$$\log(y(x)) = b_s + a_s \cdot \log(x) + \frac{(a_s - a_d)}{\alpha} \cdot \ln\left(1 + e^{-\alpha \cdot (\log(x) - s)}\right) \quad (40)$$

The smoothing parameter α can be obtained by further fitted using the nonlinear least-squares (NLLS) Marquardt-Levenberg algorithm provided by gnuplot [51].

4.4.1 REM-PAD calibration

The REM equation used for converting the units was recalibrated using the measurements of PAD dosimeters obtained by M. Karacson [29] during 2011-2018. Only those which lie in the vicinity of REM dosimeters, were presented in Table 3. The dose values obtained by the PAD dosimeters are sometimes significantly bigger than in the case of REM dosimeter calibrated as in (9), because the power law (7) is incapable of modeling the saturating range. It is clearly visible in Figure 32 that the calibration curve (purple line) diverges from the measurements in high dose values.

Instead, a smooth-piecewise model is suggested as a calibration curve for REM dosimeters. In order to achieve this, two straight lines were fitted to the dynamic and saturating range of the REM dosimeter calibrated with PAD measurements in logarithmic scale. Following the recipe (40) from the previous section, two segments are connected by a smooth transition around the breakpoint giving (41). All fitted parameters are presented in Table 4.

SN	Description	Dose [Gy] at the end of each year			
		2011	2012	2016	2017
11	M1 Xouter	8.97	27.22	47.92	86.52
12	M1 Ycenter	31.3	116.9	225.4	319.6
13	M1 Xcenter	51.65	174.5	344.6	494
14	M1 Ycenter+1	10.71	29.06	53.56	74.96
15	M1 Ycenter+2	3.9	10.845	19.045	26.745
16	M1 Xcenter+2	11.95	39.85	73.1	100.6
25	VELO repeater	12.65	38.25	71.4	99.95
27	RICH2 HPD bottom A-side	5.57	18.12	34.07	47.42
28	M1 Youter	too low	2.75	5.85	8.8
37	BLS beam entrance	3090	9871	18984.5	25086
38	ECAL test module WS	642	2589	4613.5	5893.5
39	ECAL test module BS	1137	4487	7306.5	10859.5
40	M1 Xcenter+1	20.5	78.9	148.8	209.95

Table 3: Polymer-Alanine Dosimeters (PAD) dose measurements obtained by M. Karacson [29] at the end of 2011, 2012, 2016 and 2017.

Smooth-piecewise calibration curve for the REM dosimeter

$$D(\Delta V_{th}) = 10^{0.5368 + 2.7073 \cdot \log(\Delta V_{th}) + 0.10271 \cdot \ln\left(1 + e^{-15.6919 \cdot (\log(\Delta V_{th}) - 0.8521)}\right)} \quad (41)$$

a_d	1.10	$\pm 2.98\%$
b_d	1.91	$\pm 0.98\%$
a_s	2.71	$\pm 8.22\%$
b_s	0.54	$\pm 47.46\%$
s	0.85	-
α	15.6919	$\pm 88.11\%$

Table 4: Calibration parameters for the REM dosimeter with their asymptotic standard errors provided by gnuplot software.

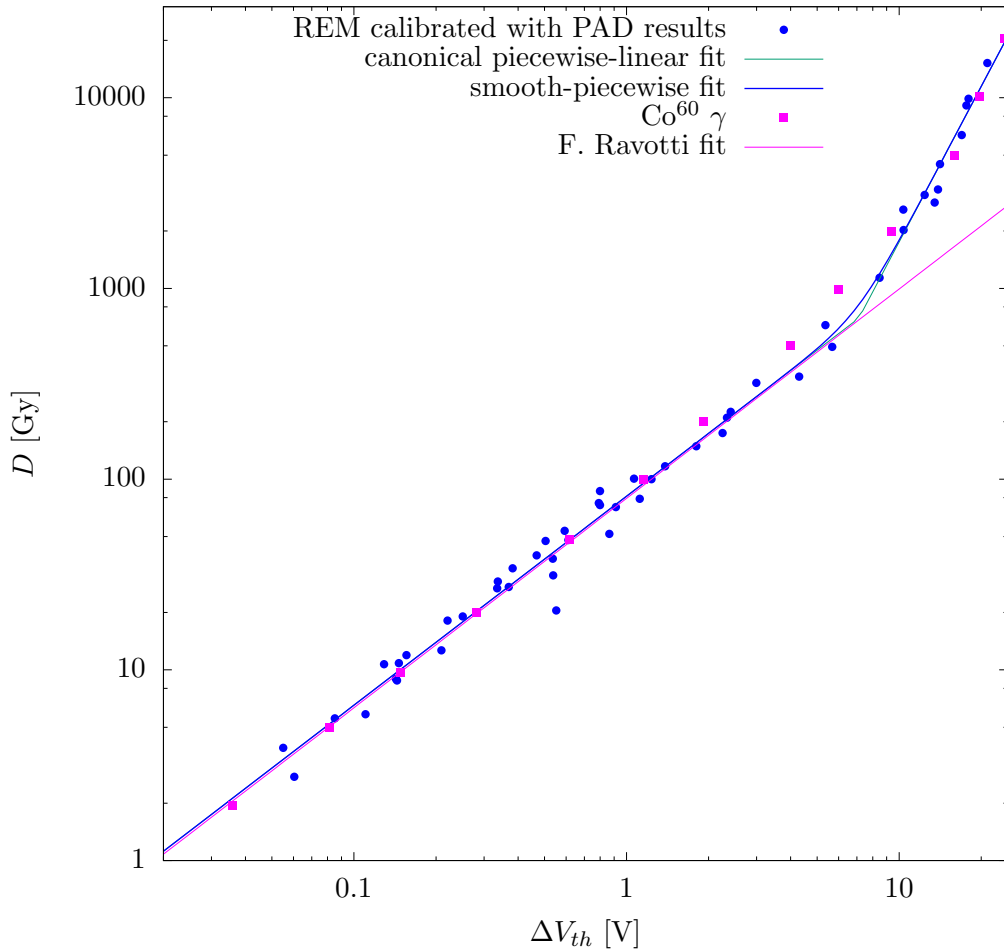


Figure 32: REM dosimeters calibrated with PAD measurements^f obtained at the end of 2011, 2012, 2016 and 2017.

^fREM-PAD measurements are presented with blue circles. A piecewise-linear function with a smooth transition was fitted to both dynamic and saturating ranges of the REM dosimeter. A calibration curve (purple) obtained by F. Ravotti was plotted to compare the results.

5 Description of software implementation

The target system for the implementation of the correction algorithm is the INFARMS control system project written in WinCC OA, because it is capable of correcting measurements and storing them in the Oracle database autonomously in quasi-real time. The correction algorithm has been designed to work as a recursion i.e. it does not require full access to the historical data each time an estimate is computed. A recursive character is also of great significance to the control system architecture that operates in real time.

Repository	Framework	Languages	Type
arms	-	C++11, MATLAB	offline computing
infarms	WinCC OA	CTRL, ANSI-C	control system
lbecsdata, lbarms	Django	Python, JavaScript	Web API

Table 5: Software components of the LHCb Active Radiation Monitoring System.

A major drawback of this approach is that it is impossible to change parameters without rerunning the whole algorithm from scratch. The data coming from the control system is stored as a time series in the database. In order to avoid corrupting data it is not allowed to write on the values with a timestamp earlier than the last archived value. In other words, the algorithm has to remain intact once launched. Whenever a user decides that some parameters are incorrect and running it again is inevitable, the only possibility to do so is to do it offline on a local machine.

Another inconvenience arises, when the historical data stored in the database is plotted using the WinCC OA. It usually takes a lot of time to obtain recent data. For this reason, a web API was developed in order to retrieve archived data from the database. The main goal of the API is to decouple the database access from the underlying database infrastructure. Additionally, it optimizes the queries sent to the database in order to obtain an equally sampled dataset. This can also be applied to other systems that store values in the same database. Hence, also a more generic RESTful API was spun off. A list of all software components is presented in Table 5.

5.1 Offline project in C++

LbARMS is the offline project created in order to obtain the results from the correction algorithm by hand. The main goal was to make the project easily adaptable to a variety of settings by changing only the configuration files. The LbARMS project's directory structure is presented in Figure 33. All header files (`.h` and `.hpp`) were placed in the `include/` directory, whereas all the source files (`.cpp`) under the `src/` directory. The user has to put in `/data` a CSV file containing voltage and timestamp columns. In the `config/` directory two configuration files are mandatory: one containing general parameters like the smoother calibration parameters and one with the parameters specified for each sensor separately. The project is equipped with two exemplary configuration files.

```

arms/
├── bin/
│   └── arms .....main binary file to execute
├── config/
│   ├── config.cfg.....exemplary general configuration file
│   ├── example_sensors_list.csv.....exemplary list of sensor-dependent
│   │   settings
│   └── lhcb_sensors_list.csv.....list of sensor-dependent settings for the
│       LHCb Experiment as of January 2019
├── data/
│   ├── example_data.csv .exemplary CSV File containing timestamp, voltage
│   │   and (optionally) temperature readout
│   └── example_lum.csv (optional) exemplary CSV File containing timestamp
│       and integrated luminosity readout
├── doc/
│   ├── images/ .....directory with images used in documentation
│   └── Doxyfile..... Doxygen config file used to generate LaTeX and HTML
│       documentation
├── include/
│   ├── data.h .....Data class responsible for i/o files
│   ├── exception_handler.h ..... ExceptionHandler
│   │   class extends Exception class in order to display more information
│   │   whenever an exception related to the project is thrown
│   ├── helper.h ... contains miscellaneous functions and global variables
│   │   used throughout the whole project
│   ├── kalman.h ..... defines Kalman filters
│   ├── matrix.hpp .....creates an NxN dimensional matrix container and
│   │   provides basic operations on matrices
│   └── signal.hpp .....Signal class provides functions used in signal
│       processing and drift correction
├── libs/ .....external libraries
│   └── VariadicTable/
│       └── VariadicTable.h
├── matlab/
├── output/
│   └── plots_to_pdf.tex .....generates one pdf file containing all .png or
│       .tex plots inside this directory
├── src/
│   ├── data.cpp
│   ├── exception_handler.cpp
│   ├── helper.cpp
│   ├── kalman.cpp
│   ├── main.cpp
│   ├── .gitignore
│   ├── CMakeLists.txt
│   └── README.md

```

Figure 33: Directory tree of the offline project

The signal processing part of the correction algorithm has been implemented us-

ing three classes, whose collaboration diagram is depicted in Figure 34. The whole C++ LbARMS offline project was designed in such a way as to simulate the performance of its WinCC OA counterpart. New containers and functions for matrices needed to be implemented, because WinCC OA supports only a limited number of mathematical functions. This has been solved in the offline program using a `Matrix` container. The container is, in fact, a vector container inside another vector container (`vector<vector<double>>`). This resembles the container used by the WinCC OA (`dyn_dyn_float`). The `Matrix` class implements also the most important operations on matrices (Listing 4). In the online project they are placed as functions in the library file (Listing 9).

5.1.1 Installation

The LbARMS offline project is maintained in a git repository at CERN's GitLab. Git is a system that enables team members to track changes in the project. The LbARMS repository can be accessed via

<https://gitlab.cern.ch/lhcb-rad/arms>.

In order to download the project on a local machine, one has to clone the git repository locally. Once downloaded on the local computer, a CMake program has to be run in order to generate a Makefile. The Makefile is then used to compile and generate a binary file `arms` for the project. The configuration file for the CMake is `CMakeLists.txt`. All needed commands are listed in Listing 1. The integrated luminosity can be obtained by adding up the value of the instantaneous luminosity multiplied by the time difference between the adjacent values.

```
1 git clone https://gitlab.cern.ch/lhcb-rad/arms
2 cd arms
3 cmake .
4 make
5 cd bin
6 ./arms -c ../config/config.cfg
```

Listing 1: Compiling and running the LbARMS offline project

```
1 #/data/11.csv
2 #DD-MM-YYYY HH:MM:SS, BPW, CMRP, LAAS, NTC(TEMPERATURE), REM
3 09-09-2010 00:01:02.537,5.94E-1,1.21E+0,4.79E+0,2.08E+1,3.29403
4 09-09-2010 00:23:55.584,5.94E-1,1.21E+0,4.79E+0,2.08E+1,3.29403
5 09-09-2010 00:46:48.600,5.94E-1,1.21E+0,4.79E+0,2.08E+1,3.29403
6 09-09-2010 01:09:41.600,5.94E-1,1.21E+0,4.79E+0,2.08E+1,3.29403
7 09-09-2010 01:32:34.708,5.94E-1,1.21E+0,4.79E+0,2.08E+1,3.29403
```

Listing 2: Exemplary structure of the voltage CSV file



Figure 34: Collaboration diagram of the correction algorithm in the LbARMS offline project.

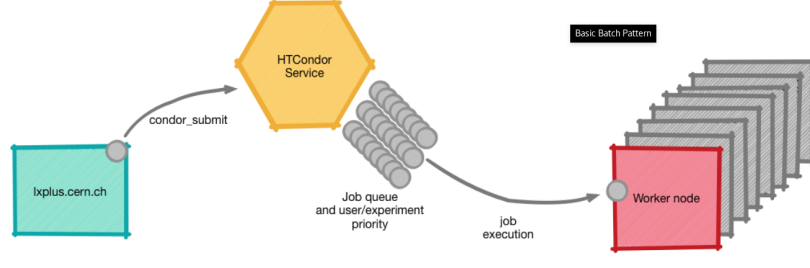


Figure 35: Job cycle in the HTCondor batch service on lxplus.cern.ch.

```

1 #/data/lum.csv
2 #DAY  MONTH  YEAR  HOUR  MINUTE  SECOND  INTEGRATED LUMINOSITY
3 13      3    2011   18     6     16     0
4 13      3    2011   18     8     22    59.346882
5 13      3    2011   18    10     23   116.085113
6 13      3    2011   18    12     29   175.066469
7 13      3    2011   18    14     35   233.753363

```

Listing 3: Exemplary structure of the optional CSV file containing luminosity

Before running the program, the files containing raw voltage and, optionally, the integrated luminosity have to be provided in the `data/` directory. The structure of the voltage CSV file (Listing 2) is usually dictated by the script that exports the data from the Oracle database and can be accessed on the Online Group machine. The user has to define how the data is represented in the CSV files by specifying variables in the configuration file like the delimiter type, time format etc. The only condition is that the way data is presented has to be consistent.

The CSV file containing the integrated luminosity should include a date followed by the integrated luminosity value in cm^{-2} (Listing 3). For LHCb, it can be obtained by integrating the instantaneous luminosity. These values are recorded every 3 seconds and stored in the Online Group directory in CSV files.

5.1.2 Running

The LbARMS project consists of four major steps. Firstly, it ensures that all CSV and configuration files are loaded correctly. Once it is done, the values are passed to the signal processing part where they are smoothed and the negative effect of temperature is

removed, if the option is selected. By default, the temperature correction is turned off. Smoothed values are then passed to the drift correction part, which is responsible for removing all drops in the signal caused by annealing. Finally, converted values of dose and fluence are saved in the CSV files in the `output/` directory. Subsequently, gnuplot scripts are generated.

In order to run the program for all the sensors available at the LHCb Experiment efficiently, it is helpful to run them in parallel using the HTCondor [25] batch system provided by the `lxplus.cern.ch`. Each job is then run on a different machine (Figure 35), allowing to obtain the results in around two hours for the size of the dataset as of 2019.

```

1  /**
2  * @fn det
3  * @brief computes a determinant of the matrix
4  * @return determinant
5  */
6
7  T det(){
8      T det = 0;
9      if(rows==1) det=data[0][0];
10     else {
11         for (int c = 0; c < cols; ++c) {
12             det += ((c % 2 == 0) ? 1 : -1) * data[0][c] * ((*this).trun(0,c)).
13                 det();
14         }
15         return det;
16     }

```

Listing 4: An example of a matrix operation (computing a determinant) in the LbARMS offline program

```

1  1. Setting up the algorithm:
2
3  - - - - -
4  | | | | | / _ _ _ | | | | | \ / | | | \ / | / _ _ _ |
5  | | | | | _ _ | | | | | _ _ / \ | | | | | \ / | ( _ _
6  | | | | | _ _ | | | | | ' _ \ / \ | | | | | _ / | \ / | \ _ _ \
7  | | _ _ _ | | | | | _ _ _ | | | | | / _ _ _ \ | | | | | \ | | | | | _ _ _ |
8  Offline project for LHCb Active Radiation Monitoring System
9  -> Reading config file
10
11 -----
12 |          OPTION          |          VALUE          |
13 -----
14 | DATA_DIR                | ../data/                |
15 | OUTPUT_DIR               | ../output/              |
16 | CONFIG_DIR               | ../config/              |
17 | SNRS_FILE_NAME           | example_sensors_list.csv |
18 | SNRS_DATE_FORMAT         | %d-%m-%Y %H:%M:%S      |
19 | SN                        | 0                        |
20 | VOLT_FILE_NAME           | example_data.csv        |
21 | VOLT_COLUMN_NO           | 1                        |

```

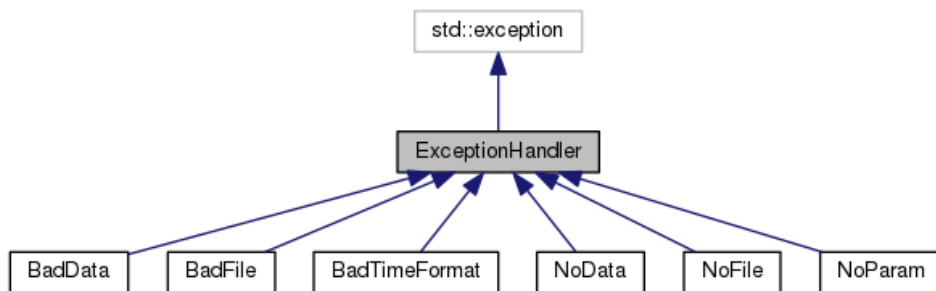
```

21 | VOLT_CSV_DELIMITER | , |
22 | VOLT_DATE_COLUMN_NO | 0 |
23 | VOLT_DATE_FORMAT | %d-%m-%Y %H:%M:%S |
24 | INCLUDE_TEMP_CORR | NO |
25 | VOLT_TEMP_COLUMN_NO | 4 |
26 | INCLUDE_SMOOTHING | YES |
27 | KALMAN_WINDOW | 50 |
28 | KALMAN_MUTE | 0.002 |
29 | KALMAN_DEG_OF_FRDM | 2 |
30 | INCLUDE_DRIFT_CORR | YES |
31 | LUM_FILE_NAME | example_lum.csv |
32 | LUM_CSV_DELIMITER | , |
33 | LUM_DATE_COLUMN_NO | 0 |
34 | LUM_DATE_FORMAT | %d %m %Y %H %M %S |
35 | LUM_COLUMN_NO | 1 |
36 -----
37 -> Reading sensors list
38 -----
39 | Parameter | Value |
40 -----
41 | ---- | ---- |
42 | ID | 1 |
43 | Sensor | BPW |
44 | Offset [V] | 0.600000 |
45 | Start | Fri Jan 1 00:00:00 2010 |
46 | End | Fri Apr 11 01:00:00 2014 |
47 | Title | SN Example |
48 | Unit | Fluence [cm$^{-2}$] |
49 -----
50 1. Setting up the algorithm:
51 -> Loading voltage
52 -> Preparing luminosity
53 -> Reading luminosity file
54 -> Converting luminosity to beam on flag
55 #1 period in progress
56 1. Signal processing:
57 -> Removing the extrema
58 -> Smoothing the signal
59 [=====] 100 % SNR: 75444034.424096 MEAN: 0.000000 STDDEV:
    0.007861786202
60 2. Conversion:
61 -> Converting units
62 3. Drift correction:
63 -> Correcting drift
64 -> Writing results to ../output/SNExample.csv
65 -> Preparing gnuplot script
66 Done.

```

Listing 5: An example of a correct output from the LbARMS offline program

The LbARMS project informs the user whenever the execution of the program is terminated due to an inappropriate file structure (Listing 7). If the user does not remember the input arguments for the binary file, they can be looked up with the `--help` flag (Listing 6). All project-specific exceptions are depicted in the class inheritance diagram (Figure 36).



```
- - - - -
```

```
| | | / _ _ _ | |          /\   | _ _ \ | \ / | / _ _ _ |
```

```
| | _ _ | | | | _ _      / \  | | _ ) | \ / | ( _ _
```

```
| _ _ | | | | ' _ \    / /\ \ | _ / | \| | \| _ _ _ \
```

```
_ _ | | | | | _ _ _ | _ ) | / _ _ _ \ | | \ \ | | | | _ _ _ ) |
```

```
_ _ _ | | | | \ _ _ _ _ | _ _ _ / _ /    \ \ _ | \ \ _ | | | _ _ _ _ /
```

```
line project for LHCb Active Radiation Monitoring System
```

```
ge: ./arms [FLAG] [VALUE]
```

```
- - - - -
```

FLAGS	DESCRIPTION
--help	lists all possible commands
	general config file path (mandatory) ex. ../config/config.cfg
	overrides the name of the data file
	overrides the serial number
col	overrides the voltage column number

```
- - - - -
```

```
1 Error: Cannot convert sdsdsadasd16-11-2012 03:43:33.50200000 to timestamp!
```

5. DESCRIPTION OF SOFTWARE IMPLEMENTATION
59

5.1.4 MATLAB implementation of Kalman filters

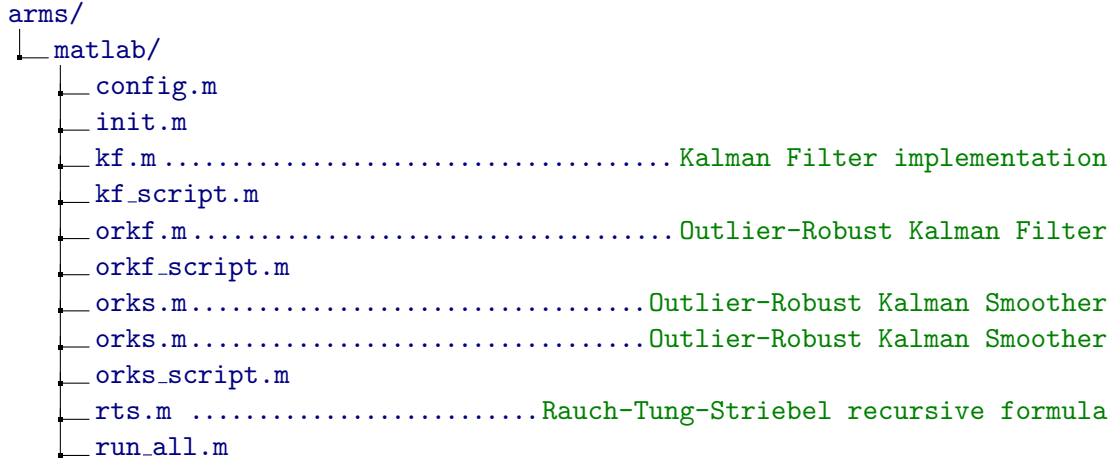


Figure 37: Directory tree of the MATLAB implementation of the Kalman filtering part

The algorithms from the theoretical part 1 (Algorithm 1,3,4 and 5) were also implemented using MATLAB software, because it is easier to test the performance of the correction filters with already included advanced mathematical libraries and there is a possibility to view the output on a zoomable plot. All .m files were placed in the `matlab/` directory of the LbARMS offline project (Figure 37). An example listing with the Rauch-Tung-Striebel formula is shown in Listing 8.

```

1 function [x,P] = rts(x_prev,xf,P_prev,Pf,F,Q)
2 % RTS Recursive formula for Rauch-Tung-Striebel smoother.
3 % The algorithm follows BACKWARD recursion scheme. "Previous" means the
4 % same variable as in the case of "following" in forward recursion.
5 %% Input variables:
6 % x_prev - previous smoothed estimate from RTS backward recursion,
7 % xf      - a posteriori estimate from forward Kalman recursion,
8 % P_prev - previous smoothed error covariance from RTS backward recursion,
9 % Pf      - a posteriori error covariance from forward Kalman recursion,
10 % F       - state-transition model matrix,
11 % Q       - the covariance matrix of the process noise.
12 %% Output variables:
13 % x       - smoothed state estimate vector,
14 % P       - smoothed error covariance matrix,
15 %% Recursive RTS formula
16 J=Pf*F'*inv(F*Pf*F'+Q);
17 x=xf+J*(x_prev-F*xf);
18 P=Pf+J*(P_prev-F*Pf*F'-Q)*J';
19 end

```

Listing 8: An example of the offline MATLAB function (Rauch-Tung-Striebel smoother) in the LbARMS project

5.2 Online project in WinCC OA

The online project is an extension of the already existing INFARMS project. For the sake of clarity and compactness of the results, a completely new panel in WinCC OA was created. The general purpose of this panel is to provide a user with an ability to:

- check the corrected dose and fluence values of each sensor quickly with a possibility to obtain insights into each step,
- monitor the relevance of the correction algorithm and the accuracy of the conversion,
- change the sensor with a full support in resetting the parameters,
- compare the output with another dosimetry system called Radiation and Environment Monitoring Unified Supervision (REMUS).

The main panel of the INFARMS project is presented in Figure 38. A current date is displayed in the upper left side of the panel. A plot of integrated luminosity was placed next to it. The large circle with a ‘Beam’ sign above it turns green whenever the beam is present or grey otherwise. A table with a complete list of all the sensors of the system is displayed beneath it. Each cell displays a value after the final step of the correction algorithm. The cell colors give information about the current status of the sensor according to the following rules:

- **FwStateOKPhysics**, if the correction was successful;
- **FwStateEquipmentDisabled**, if the sensor has not been updated since the last replacement of the sensors;
- **FwStateStateAttention2**, if the correction algorithm has been stopped due to maintenance;
- **FwStateStateAttention1**, when the sensor is on the verge of its dynamic range and a replacement is suggested,
- **FwStateStateAttention3**, when an error in the connection appeared.

On the right side of the main panel, a layout of the detector is depicted. The green box in the middle displays the layout of sensors in front of the M1 station. Each cell of the table is clickable and, as a result, a new panel with additional information is displayed (Figure 39). It gives information about the position of the sensor in the detector (blue description) and the computed initial threshold voltage, which is subtracted from the subsequent measurements that has been referred to as offset. On the right side, the results from the correction algorithm are displayed. The upper frame corresponds to a measurement, which is **window** steps before the current one, and is the most precise output. It displays the smoothed values before and after drift correction on the indicated date. Additionally, an estimation

of the most recent value based on the ORKS is displayed in the frame below. All the results are presented in the plot at the bottom of the panel.

A much simpler panel has been designed for the temperature sensor, providing the user with a small plot of the current readout. The correction algorithm in the WinCC control system does not remove the negative effects caused by the temperature.

Whenever the user decides to change the sensor, it can be performed by clicking the **Maintenance** button, which results in stopping the correction algorithm. Only then is it possible to open another panel (Figure 40) by clicking a **Change Sensor** button, where an administrator can reset all the parameters used in the correction algorithm. Not only does that prevent any adverse operation of the smoothing algorithm, but also separates the statistical pattern from the previous sensor as the values will be added up but the behaviour of the new sensor will be treated on its own by the algorithm.

```
1 float det(dyn_dyn_float m){
2     double d = 0.0;
3     int rows = dynlen(m);
4     int cols = dynlen(m[1]);
5     if(rows==1) d = m[1][1];
6     else{
7         for(int col=1; col<=cols;col++){
8             dyn_dyn_float truncated = m;
9             truncated = trun(truncated,1,col);
10            d += ((col % 2 == 1) ? 1.0 : -1.0) * m[1][col] * det(truncated);
11        }
12    } return d;
13 }
```

Listing 9: An example of a matrix operation (computing a determinant) in the WinCC OA project

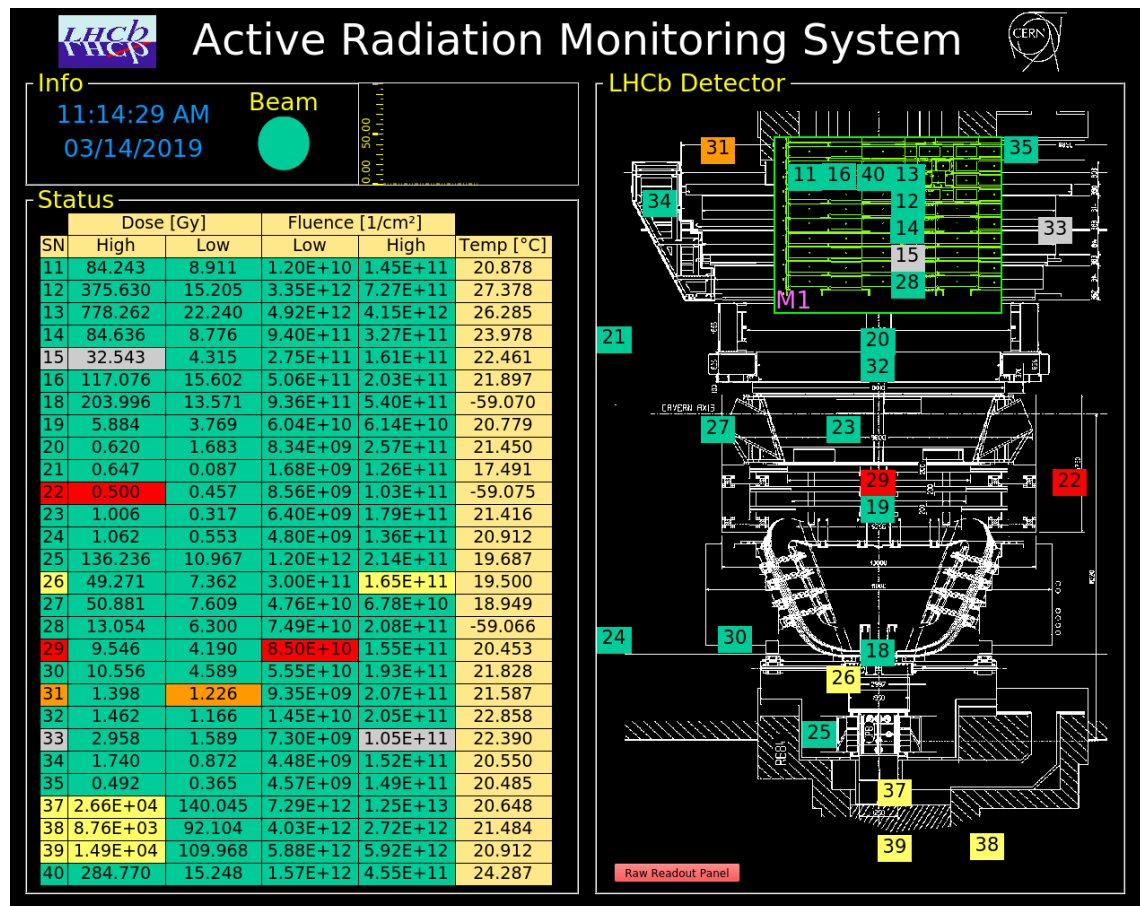


Figure 38: New main panel of the LHCb Active Radiation Monitoring System

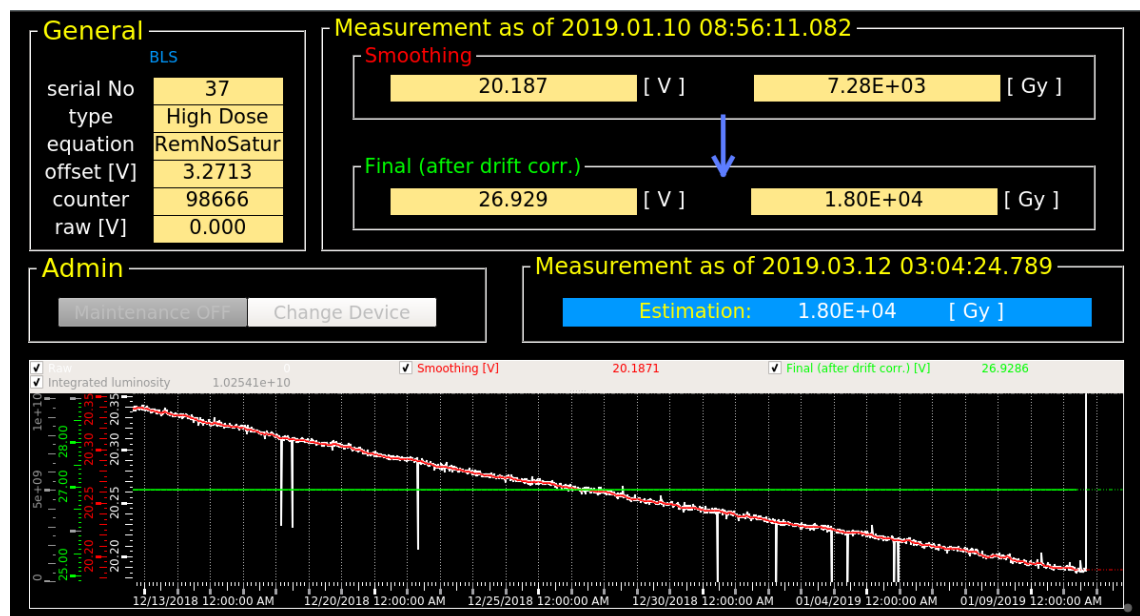


Figure 39: Panel with more detailed information about the correction algorithm.

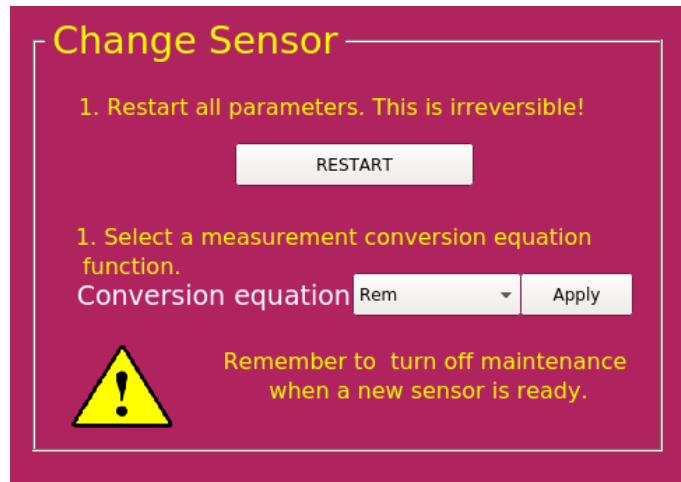


Figure 40: Panel allowing to restart the sensor's parameters.

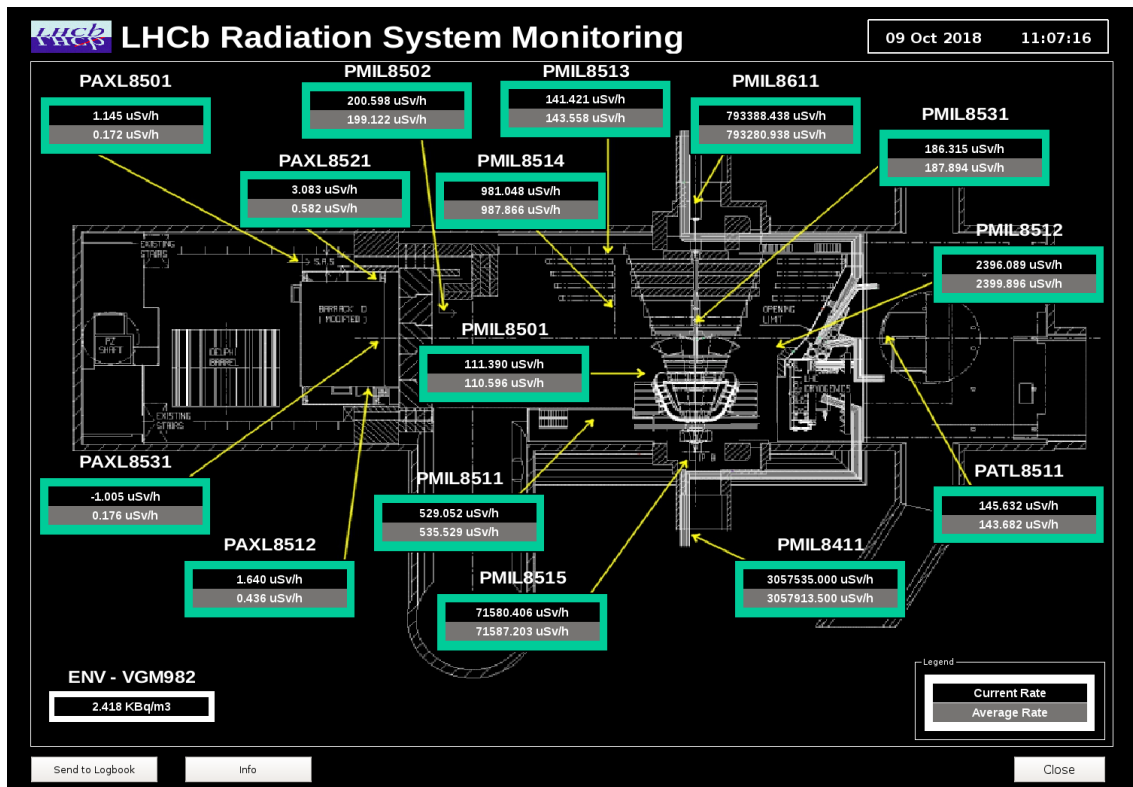


Figure 41: LHCb panel that displays the values coming from the Radiation and Environment Monitoring Unified Supervision (REMUS) system.

5.3 Django Web APIs

When analyzing the results produced by the correction algorithm it is important to be able to study the device's past performance. This can help to decide as to whether the device is, for instance, still operable. WinCC OA plots do not provide options to sample the archived data, which is crucial in order to display the whole history consisting of around 200000 records as of 2019.di3lock

A more flexible approach to access data stored in relational databases is through RESTful APIs – the web services that use HTTP (Hypertext Transfer Protocol) methods: GET, HEAD, PUT, DELETE, and POST to operate on data [45]. The REST (Representational State Transfer) is perceived as a way of ‘judging’ architectures. The Resource-Oriented Architecture (ROA) is an architecture that allows to redefine the problem as a RESTful resource. A resource is usually designed with an object-oriented approach. Django [12] with a ‘Django REST framework’ toolkit [14] can be used as a framework implementing object-relational mapping (ORM) tool that maps the data model to the resources in relational databases.

5.3.1 Generic application

In 2018 L. Pira [38] created a RESTful API for the Experiment Control System (ECS) in LHCb that reads the archived data directly from the Oracle database and returns a JSON object. This idea was reused to create two web applications, sharing the same core application `lbapi` available at the main branch – `master` – of the repository:

`https://gitlab.cern.ch/lhcb-rad/lbapi`.

`lbapi` implements all back-end methods needed to retrieve data from the database, which is then used by the system-oriented applications that display the data on the client side. The application is connected to two databases: Oracle database used only for reading the ECS data and a PostgreSQL [50] database for all the resources created and used by this website. Oracle database uses two tables to store data from the ECS system. Analogously, two models were implemented in the Django application and are presented in Listing 10. All the datapoint names are available at the `Elements` model and the historical values at the `EventHistory` model, which is a child of the `Elements` model.

The datapoint names can be looked up with a `search` view. It takes an `element_name` as an input, splits it into words and looks for all datapoint names containing any of them. Once the name of the datapoint name is known, its historical values can be obtained using `present_in_json` view. It takes four parameters as an input: `element_name`, time boundaries `start_time`, `end_time` and `sample_size`. The last parameter defines a number of values taken from a given timespan. Moreover, the values are evenly sampled, including the first and the last one from the input query. The default maximum size of the sample is 5000 values. It is worth mentioning though that this tool is only useful when looking

for a general trend in time and singular defects of the signal might not appear. Data obtained from the JSON object are visualized with ‘dygraphs’ [54] charting library written in JavaScript (Figure 42).

```

1 class EventHistory(models.Model):
2     element_id = models.ForeignKey(Elements, on_delete=models.PROTECT,
                                     db_column='element_id',
                                     related_name='element_no')
3     ts = models.DateTimeField(primary_key=True)
4     value_number = models.FloatField()
5
6     class Meta:
7         managed = False
8         app_label = 'winccoa'
9         db_table = 'S362_INF_EVENTHISTORY'

```

Listing 10: An example of the model used to map ECS data stored in the Oracle database.

```

1 SELECT v FROM (SELECT ROWNUM, v FROM table) WHERE MOD(ROWNUM, sample)=0

```

Listing 11: An example of a raw SQL that can be used to sample the database.

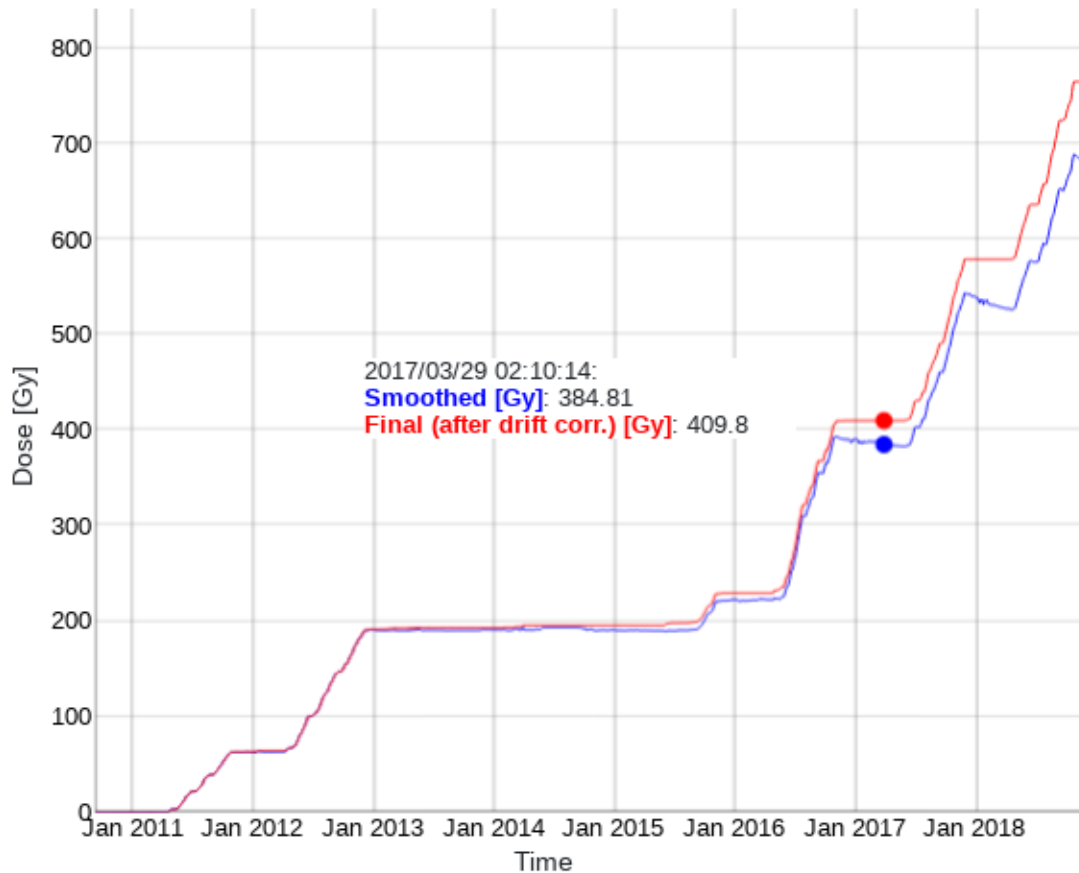


Figure 42: An example of the plot generated by the ‘dygraphs’ library.

5.3.2 LbECSData

lbECSData is a web application whose purpose is, analogously to the one created by L. Pira, to provide an access to the archived data of the Experiment Control System directly from the database and deliver it to the user as a JSON object, which can be then easily manipulated. The API uses the ORM provided by Django that decouples the way the data is retrieved, from the database structure. This API can be still used to retrieve the data, even if the database back end changes in the future. The API is available at the address:

<https://lbecsdata.cern.ch>

and the repository can be accessed at: <https://gitlab.cern.ch/lhcb-rad/lbecsapi>. The home page depicts the ECS's datapoint structure with a D3.js JavaScript library [5].

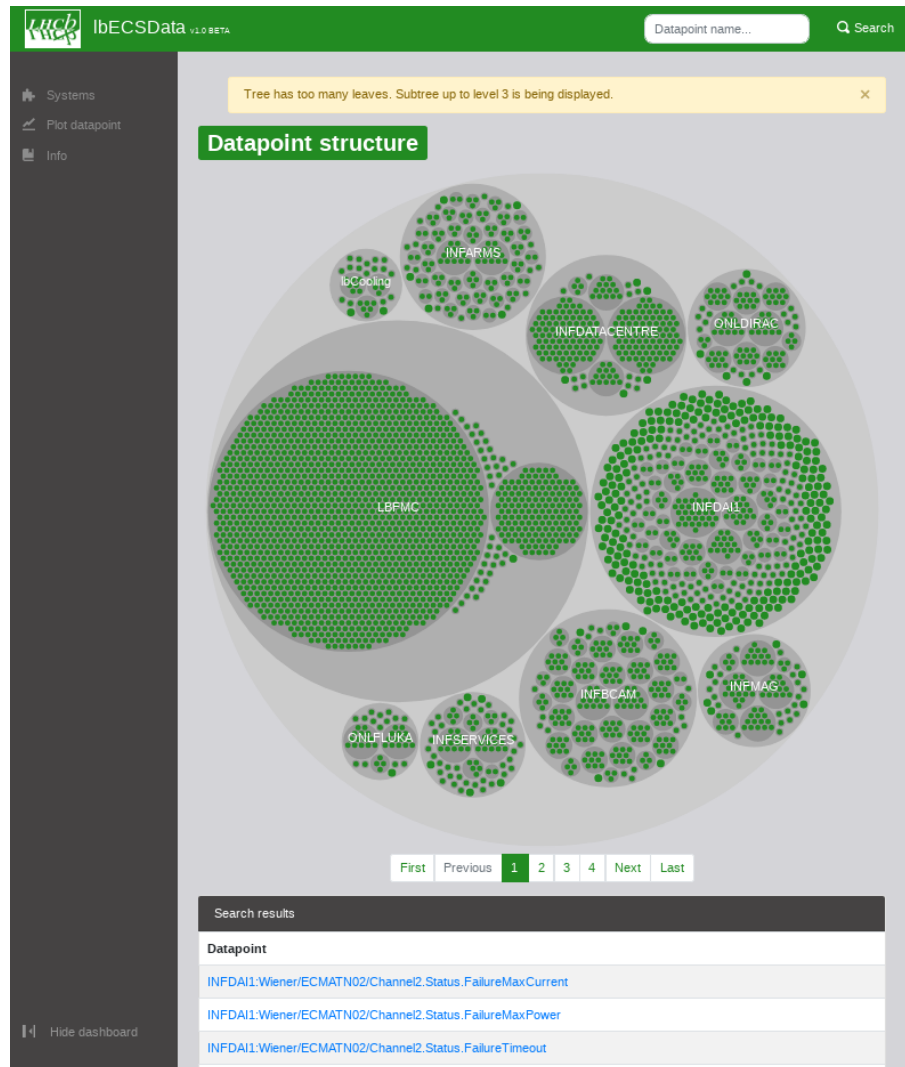


Figure 43: LbECSData's home page containing datapoint structure of the ECS system.

5.3.3 LbARMS

The LbARMS website is a little more complex in structure than the LbECSDat and is focused solely on the radiation systems: LbARMS and REMUS. The website is available at the address:

<https://lbarms.cern.ch/>

and its repository at: <https://gitlab.cern.ch/lhcb-rad/lbarms>. A responsive layout of the radiation sensors is available for both radiation systems via a link on the sidebar (Figure 44). A small tooltip with general information pops up when a label is hovered over. It is possible to generate plots for each device by clicking on the label. Even more information is shown, with a possibility to modify the plot's parameters, when a More button is clicked.

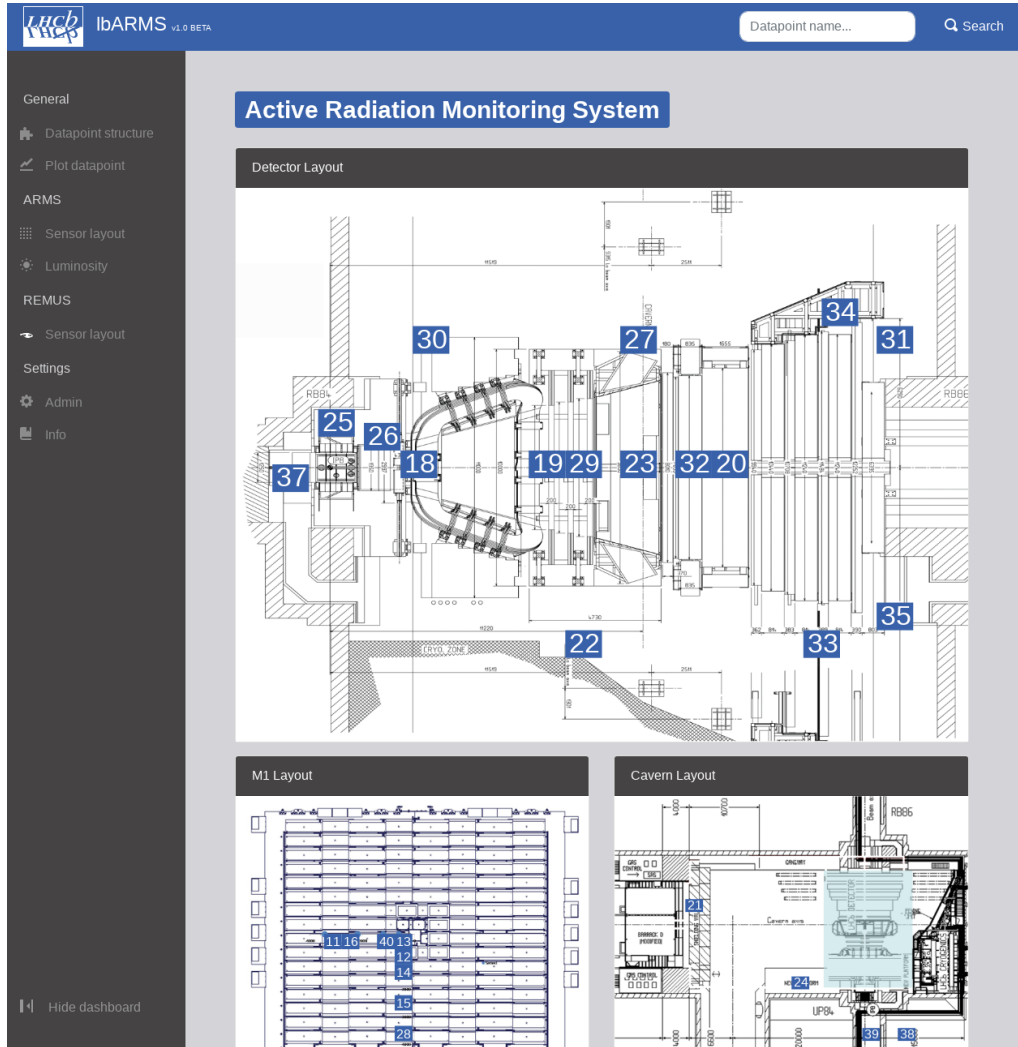


Figure 44: LbARMS's home page containing responsive layouts of the radiation sensors.

6 Outcome of the correction algorithm

The most recent values produced by the correction algorithm are presented in Table 6. Additional box plots were created (Figure 45, 46, 47, 48, 49 and 50) in order to facilitate comparisons, particularly for particle fluence values. The results are presented in groups based on layouts. In addition, the M1 station devices are ordered starting from the end of the x -axis and ending on the last device placed along the y -axis. 37, 38 and 39 devices were plotted separately in Figure 49 and 50, because of their comparably high exposure and therefore unique behaviour. The results from Low Dose sensors (i.e. LAAS) are considered generally unreliable. In addition to issues with initial threshold voltages for some sensors which were irradiated before the readout started, these sensors were found to be generally inaccurate due to calibration issues.

Layout	SN	HD [Gy]	LD [Gy]	HF [$\cdot 10^{10} \text{cm}^{-2}$]	LF [$\cdot 10^{10} \text{cm}^{-2}$]
M1 Station	11	84.243	^a 8.911	^d 14.487	^c 1.195
	16	117.076	^a 15.602	^d 20.277	50.642
	40	284.770	^a 15.248	^d 45.498	157.052
	13	778.262	^a 22.240	414.918	^b 492.271
	12	375.630	^a 15.205	^d 72.666	^b 335.407
	14	84.636	^a 8.776	^d 32.745	94.018
	15	32.543	^a 4.315	^d 16.062	27.522
	28	13.054	^a 6.300	^d 20.839	7.491
	18	203.996	^a 13.571	^d 54.011	93.552
	19	^d 5.884	^a 3.769	^d 6.136	6.036
Detector	20	^d 0.620	^a 1.683	^d 25.679	0.834
	22	^d 0.500	^a 0.457	^d 10.334	0.856
	23	^d 1.006	^a 0.317	^d 17.879	0.640
	25	136.236	^a 10.967	^d 21.377	120.471
	26	49.271	^a 7.362	^d 16.511	30.019
	27	50.881	^a 7.609	^d 6.776	4.757
	29	^d 9.546	^a 4.190	^d 15.513	8.496
	30	10.556	^a 4.589	^d 19.345	5.546
	31	^d 1.398	^a 1.226	^d 20.676	0.935
	32	^d 1.462	^a 1.166	^d 20.489	1.448
Cavern	33	^d 2.958	^a 1.589	^d 10.467	0.730
	34	^d 1.740	^a 0.872	^d 15.177	0.448
	35	^d 0.492	^a 0.365	^d 14.861	0.457
	37	^b 26626.752	^a 140.045	1254.396	^{b, a} 729.381
	21	^d 0.647	^a 0.087	^d 12.576	0.168
	24	^d 1.062	^a 0.553	^d 13.555	0.480
	38	^b 8760.391	^a 92.104	272.484	^{b, a} 402.813
	39	^b 14885.784	^a 109.968	591.680	^{b, a} 587.874

Table 6: Final correction algorithm results.

^awrong initial threshold voltage and/or calibration

^bsaturating or already saturated

^cbroken

^dbelow the lower limit of the dynamic range

The M1 station results clearly show, as expected, that the closer the sensor is to the beam pipe, the bigger the particle fluence and the absorbed dose are. A vast majority of the sensors were not exposed to significant radiation and therefore some of the results are on the verge of the lower threshold of the dynamic range. It is of great importance to always take a look at the measurement history of the device (Figure 51, 52, 53 and 54). For example, information on sensors requiring replacement or on connection problems can be gained.

As a representative example for the performance of the correction algorithm, eight plots of all the sensors: High Dose (Figure 55, 56), Low Dose (Figure 57, 58), Low Fluence (Figure 59, 60) and High Fluence (Figure 61, 62) with a serial number 37 are presented. 37 is a set of sensors that is placed directly next to the beam line and hence it sees the highest dose and fluence among all LHCb active sensors. This necessitates an operator of the system to change the sensor frequently. In this case, a change of all the sensors was performed during the Long Shutdown 1 at the beginning of 2015 and it is visible as a characteristic drop in voltage to the initial threshold voltage.

The High Dose sensors (2 REM RadFETs) with a serial number 37 (Figure 55) received the highest dose and therefore show the strongest annealing effects, which already started at the beginning of 2012. Because of the high dose and high signal-to-noise ratio, electronic noise and spikes are almost not noticeable at all. A closer look at part of the plot (Figure 56) reveals some small spikes, easily detectable by the algorithm. All the drops were correctly added to the drift variable. In general, the smoother has a tendency to mollify rapid changes in the signal. The results coming from the 37 Low Dose sensors (2 LAAS RadFETs) in Figure 59 are meaningless, because they were already saturated before their operation and the upper limit of the dynamic range of the LAAS RadFET corresponds to only 10 Gy. A group of 37 Low Fluence sensors consists of one CMRP and one Si-2 diode. The CMRP diode worked fine, however it saturated extremely fast. The Si-2 diode was installed with a readout current of 1 mA. Unfortunately, the Si-2 diodes were calibrated with a readout current of 25 mA and hence a new conversion formula has to be devised in order to convert the voltage coming from the Si-2 diodes properly. The High Fluence sensors (2 BPW diodes) are of particular interest regarding the correction algorithm, because their signal is distorted with various kinds of noise: starting from the electronic noise and ending on huge outliers or wide spikes. An example of a wide spike and a tall outlier was depicted in Figure 62. In both cases, the ORKS algorithm gets rid of them successfully, leaving a small bump as in the section describing the calibration of the smoother in Figure 30.

The computation time of the algorithm was measured including the time needed to store necessary values in the WinCC OA datapoint system. The algorithm with the default settings, which are listed in the exemplary configuration files of the offline project, takes from 150 ms up to 250 ms. Execution time depends also on the load of processes run in parallel.

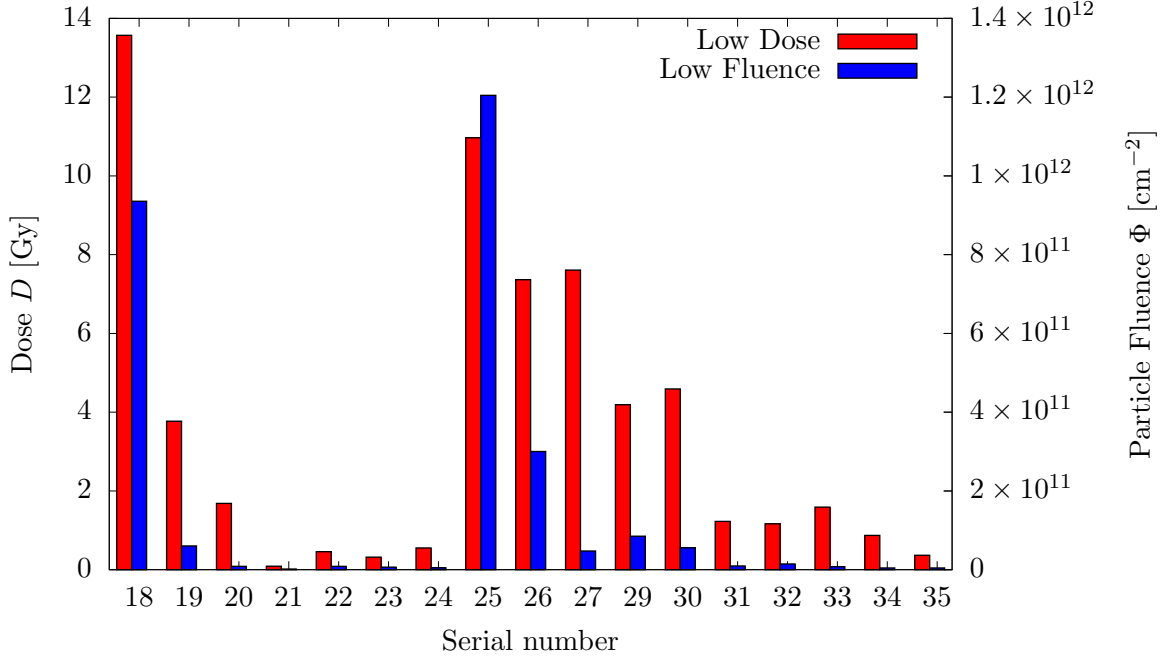


Figure 45: Final correction algorithm results for the devices with a lower dynamic range placed in the cavern and the LHCb detector (without 37, 38, 39). The Low Dose sensors are in general considered inaccurate due to calibration issues.

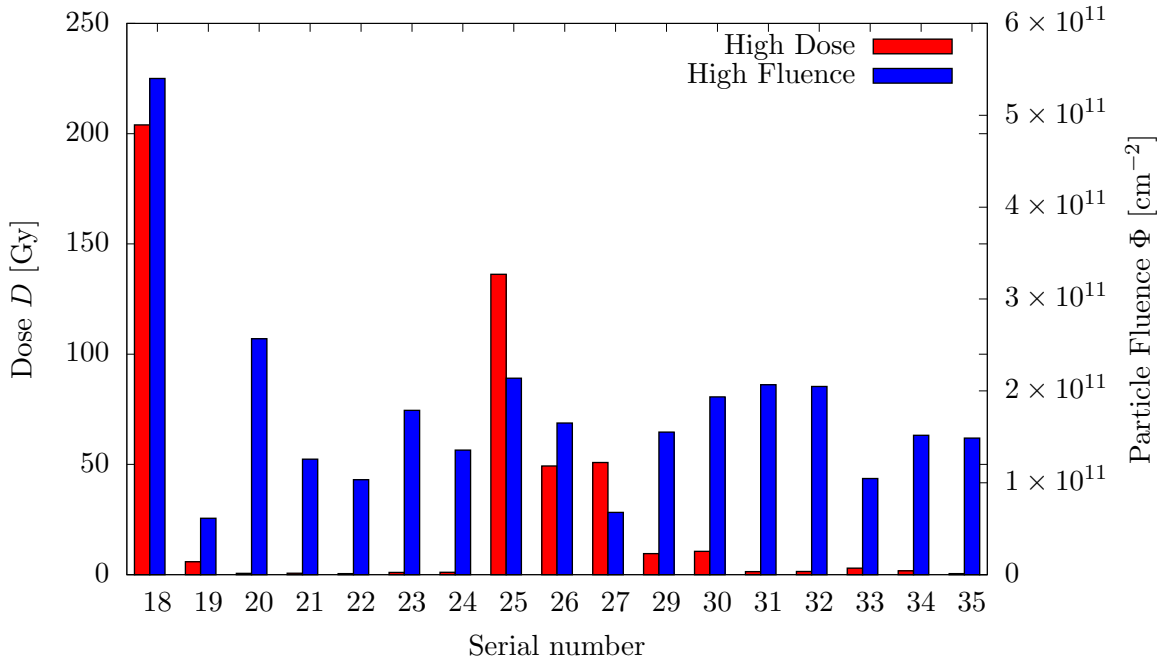


Figure 46: Final correction algorithm results for the devices with a higher dynamic range placed in the cavern and the LHCb detector (without 37, 38, 39). All the High Fluence results are irrelevant, because no sensor in this layout reached the lower limit of the dynamic range of the BPW diode.

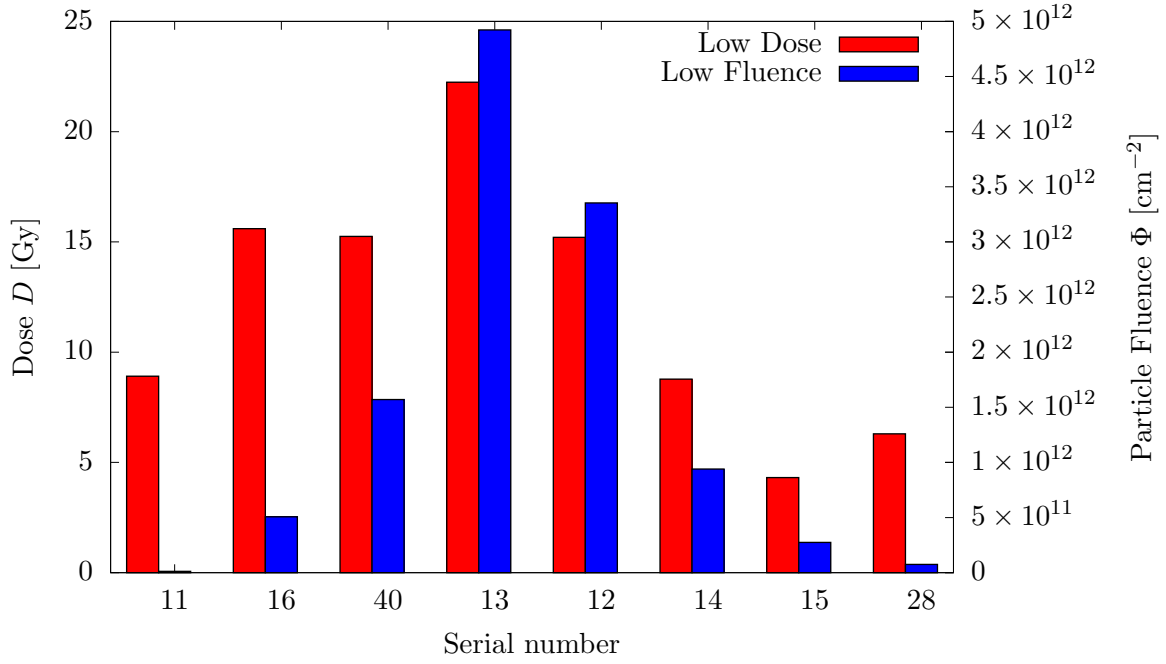


Figure 47: Final correction algorithm results for the devices with a lower dynamic range placed on the surface of the M1 station. All the Low Dose results are irrelevant, because the LAAS dosimeters were already saturated at the beginning of 2011. Sensors 12 and 13 reached the upper threshold of the dynamic range of the CMRP diode. High Fluence results should be considered instead in this case.

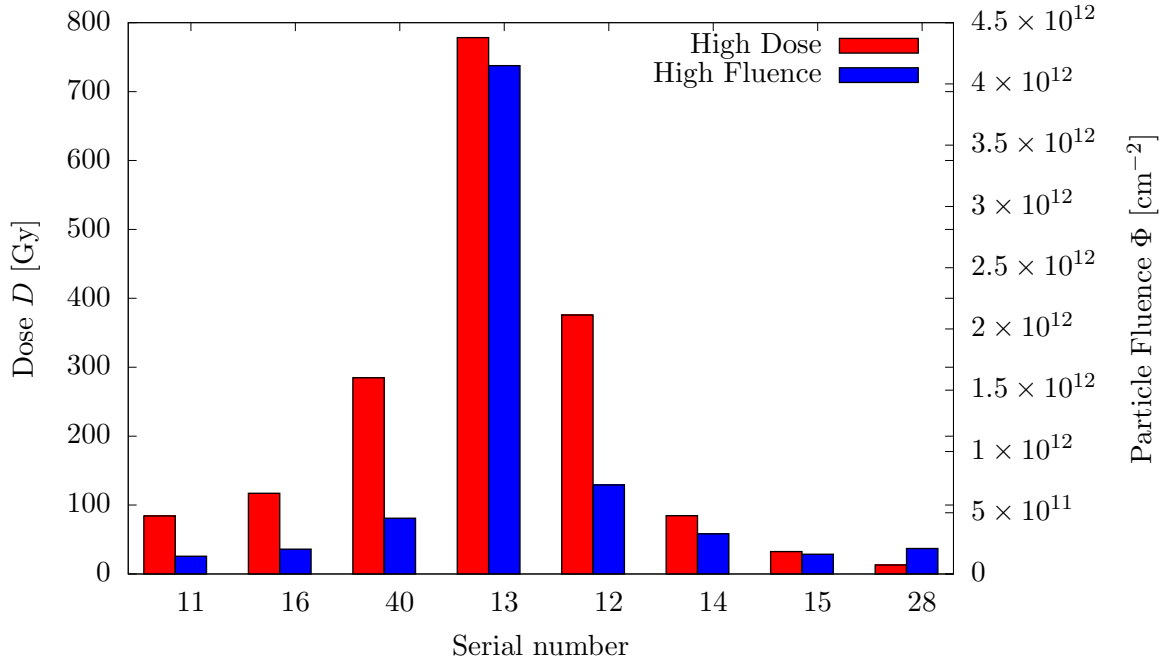


Figure 48: Final correction algorithm results for the devices with a higher dynamic range placed on the surface of the M1 station. The majority of the High Dose sensors are at the beginning of the lower threshold of the dynamic range of the REM RadFET. In the case of High Fluence sensors, only the sensor with a serial number 13 managed to enter the dynamic range.

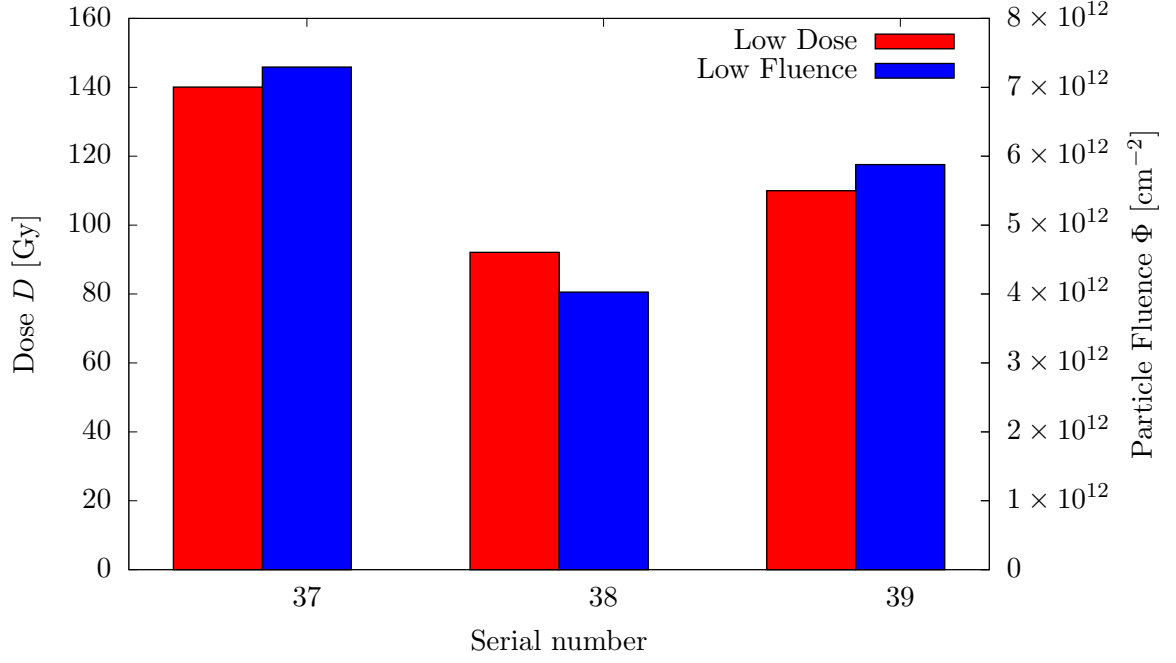


Figure 49: Final correction algorithm results for the devices with a lower dynamic range that were highly irradiated (37, 38, 39). Both the Low Dose and the Low Fluence were quickly saturated and none of these results are relevant anymore.

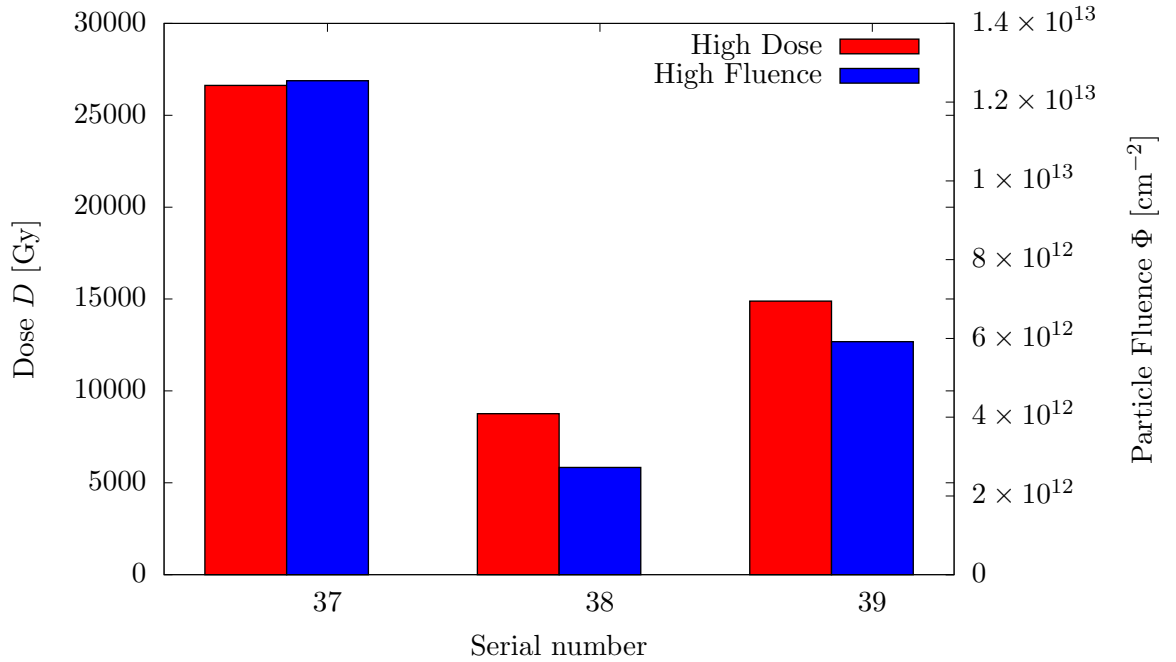


Figure 50: Final correction algorithm results for the devices with a higher dynamic range that were highly irradiated (37, 38, 39). The High Dose sensors are already saturating, whereas the High Fluence sensors are still in the dynamic range of the BPW diode.

7 Summary

The LHCb Active Radiation Monitoring System (LbARMS) at the LHCb Experiment had to be improved in order to provide more precise values of absorbed dose and particle fluence in a quasi-real time environment.

In order to do so, some changes in the calibration models were proposed. A new ‘smooth-piecewise’ calibration model was proposed for REM dosimeters based on PAD measurements collected until 2018. It is still consistent with the power law that models the dynamic range of the dosimeter and extends its operation by anticipating the behaviour in the saturating range. As a result, REM dosimeters are capable of obtaining correct doses up to 30 kGy. On the other hand, the calibration of fluence-measuring diodes could only be tested by comparing it with the simulation results.

The Outlier-Robust Kalman Smoother (ORKS) [2] was implemented in the signal post-processing part of the algorithm. Its main purpose is to smooth out the electronic noise and remove spikes corrupting the signal. The ORKS state space was modeled using a nondecreasing piecewise linear function. The noise model was calibrated by monitoring the standard deviation of the differenced signal. In general, the ORKS successfully detects and removes the majority of single spikes, smoothing the signal in a satisfactory manner. The ORKS is equipped with the Rauch-Tung-Striebel smoothing formula [41] that increases the result precision by incorporating the subsequent measurements. It is impossible, however, to implement the ORKS for a current measurement. For this reason, the current measurement is estimated with the value produced in the last step of the forward recursion (starting from N values before the current one).

The whole correction algorithm was moved to the online project in WinCC OA. A completely new GUI was devised for this purpose. The user can now track the correction process and compare it with another system used for radiation monitoring – Radiation and Environment Monitoring Unified Supervision (REMUS). If there is a need to replace the sensor, the admin panel helps to reset necessary parameters. It also displays a status of each sensor, indicating as to whether the measurement is out of range or a sensor’s change is suggested, because it has already entered the saturating range. The time needed for the correction algorithm to be completed is up to 250 ms compared to 15 minutes for the former algorithm.

The online project’s history cannot be modified once written to the database. For this reason, the offline program ‘arms’ was developed. It consists of two configuration files that list all fully adjustable parameters of the correction algorithm. For instance, the user may decide to toggle temperature correction for individual sensors, which cannot be done within the WinCC project. It is also possible to turn off the smoothing and the drift correction that can help to quickly convert data that underwent the smoothing process beforehand into appropriate units. The offline script is fully documented with an automatically generated documentation using Doxygen and with a PDF presenting the

results for all the sensors of the LbARMS system.

Finally, two Web APIs were created: LbARMS and LbECSDData, in order to read and visualize data coming from the Experiment Control System and stored in the Oracle database. LbECSDData is a general web application that helps to access data as a JSON object and visualize them with a ‘dygraphs’ charting library written in JavaScript. It is also possible to display the datapoint structure generated with the aid of the D3.js library. LbARMS focuses on giving full insights into the historical performance of the radiation sensors of the LbARMS system. It is also possible to compare them with the values generated by the REMUS system. The interface with sensor positions is fully responsive to user interaction. LbARMS is also equipped with an admin panel that allows to store additional information about the radiation sensors in the PostgreSQL database.

In the future, it will be necessary to recalibrate the fluence sensors by comparing them with a reference value different than the one from the literature. Moreover, it is advisable to check the new REM calibration model’s correctness with higher doses. Software implemented for the LHCb Active Radiation Monitoring System and presented in this thesis will enable these tasks to be performed in a reasonably simple way and greatly facilitate the re-integration of exchanged devices.

Appendix A Comparison of the results

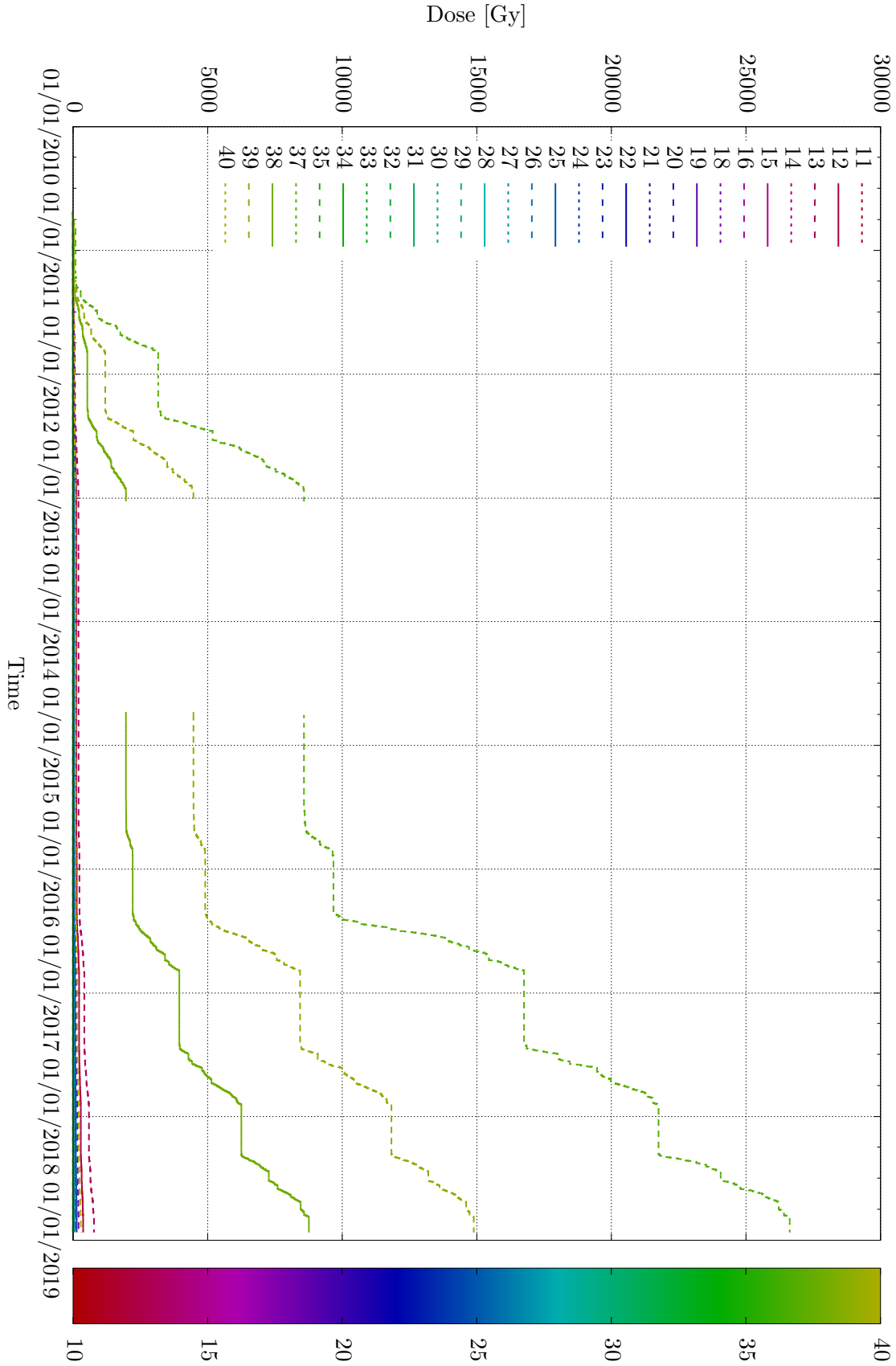


Figure 51: Summary of the final results for each 'High Dose' sensor

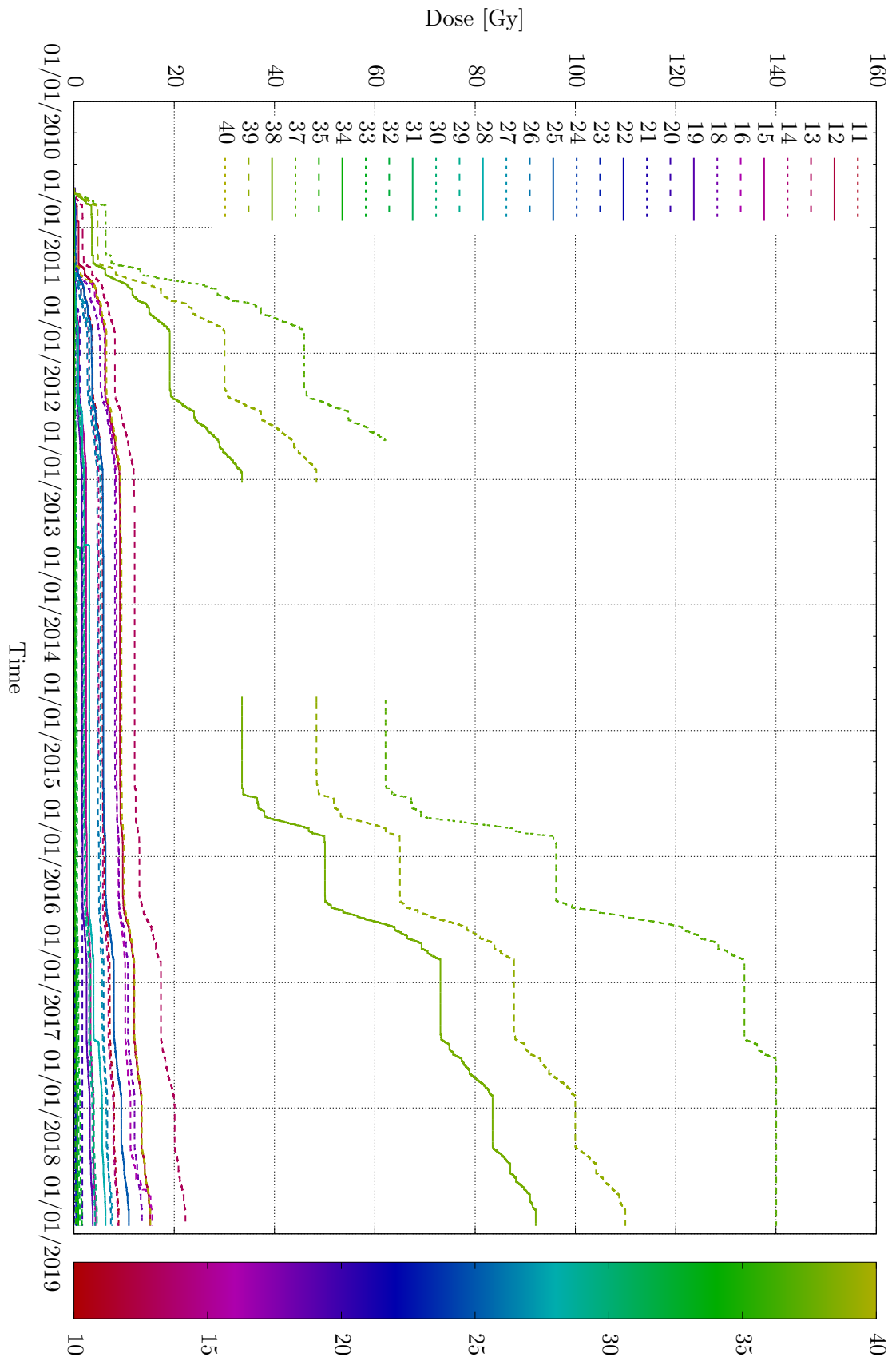


Figure 52: Summary of the final results for each 'Low Dose' sensor

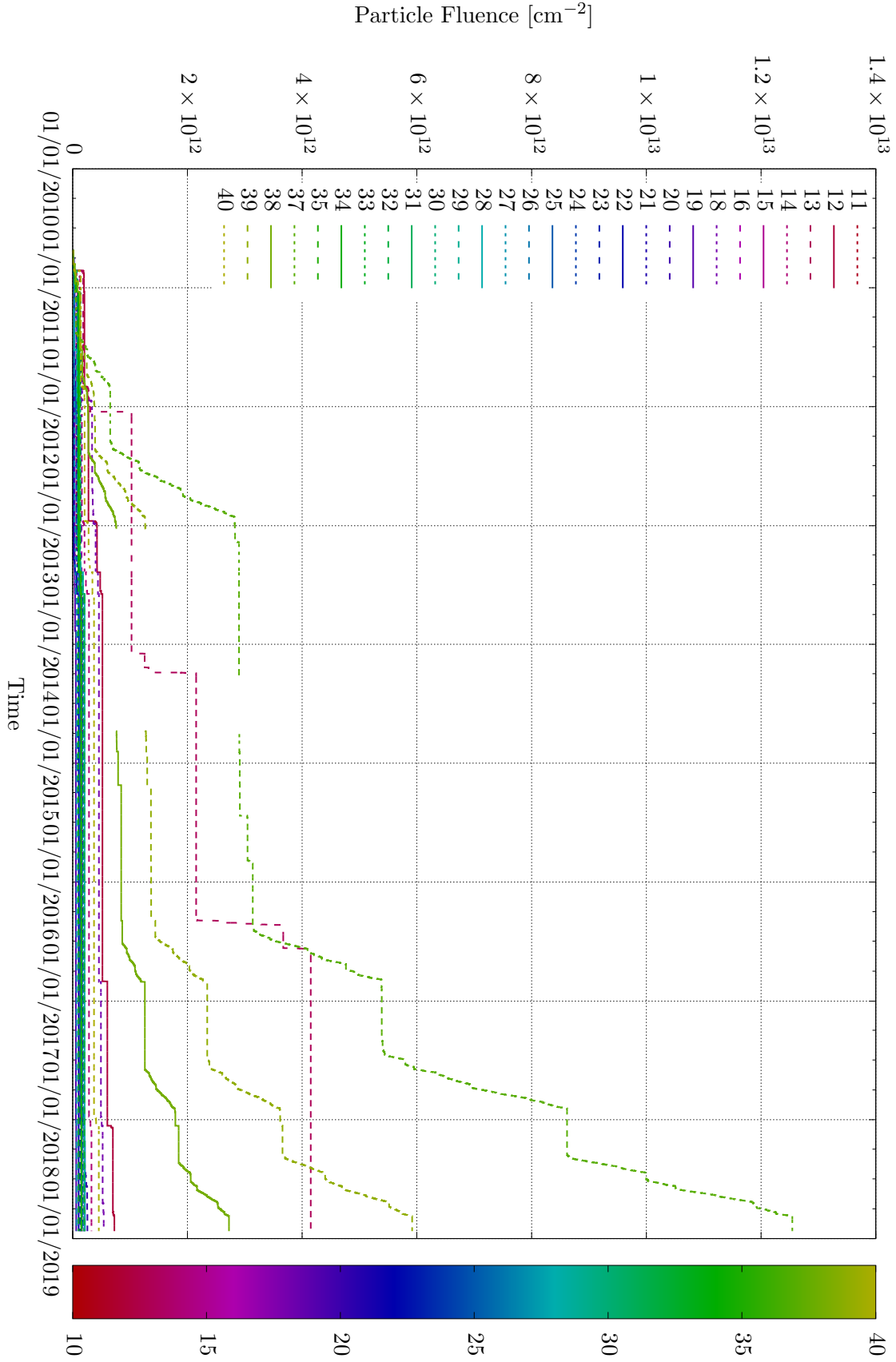


Figure 53: Summary of the final results for each 'High Fluence' sensor

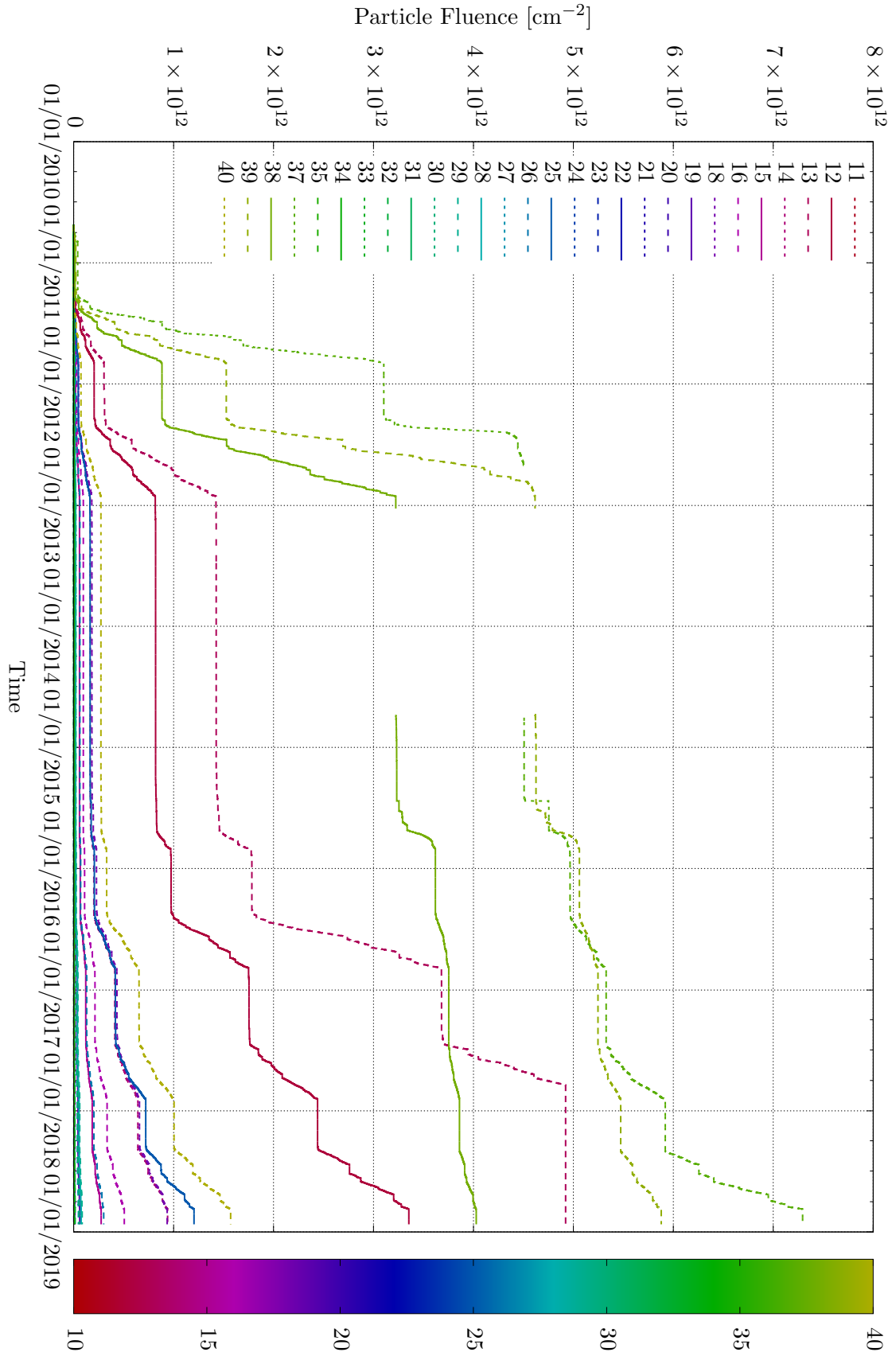


Figure 54: Summary of the final results for each 'Low Fluence' sensor

Appendix B Results for the devices with a serial no. 37

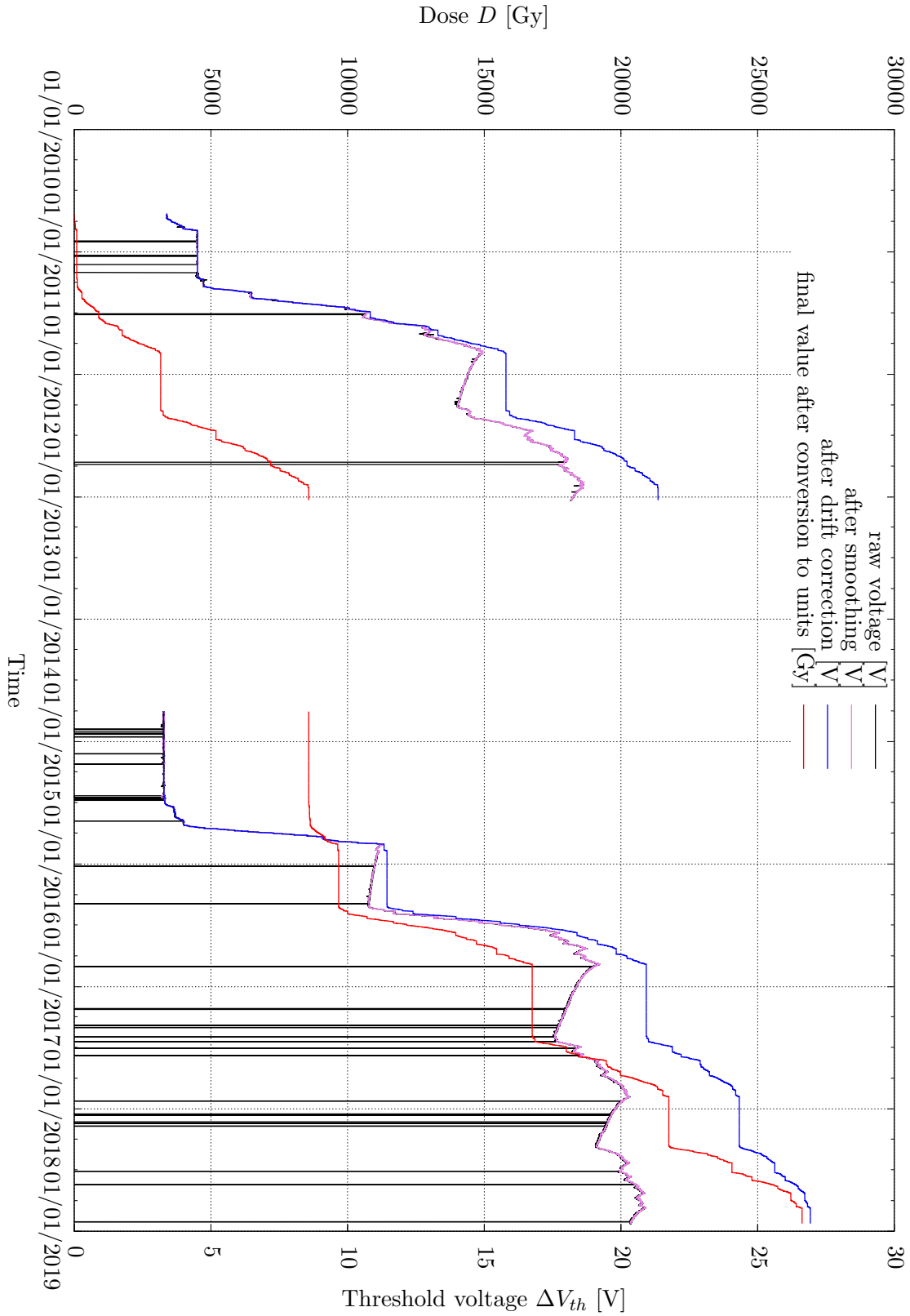


Figure 55: Final results for the 37 'High Dose' sensor with two REMs during 2010-2018

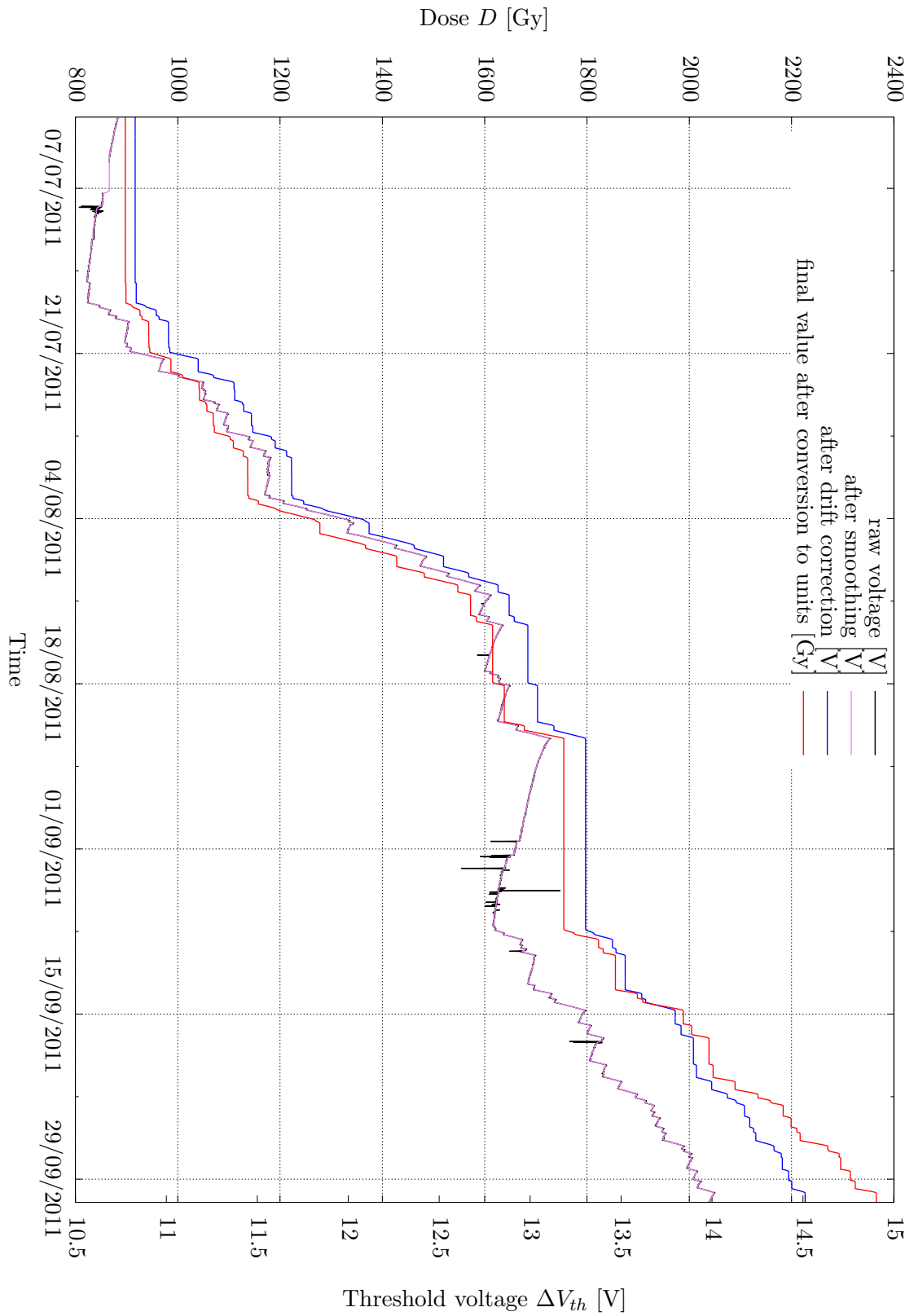


Figure 56: Signal processing performance for the 37 'High Dose' sensor with two REMs in a 3 month timespan

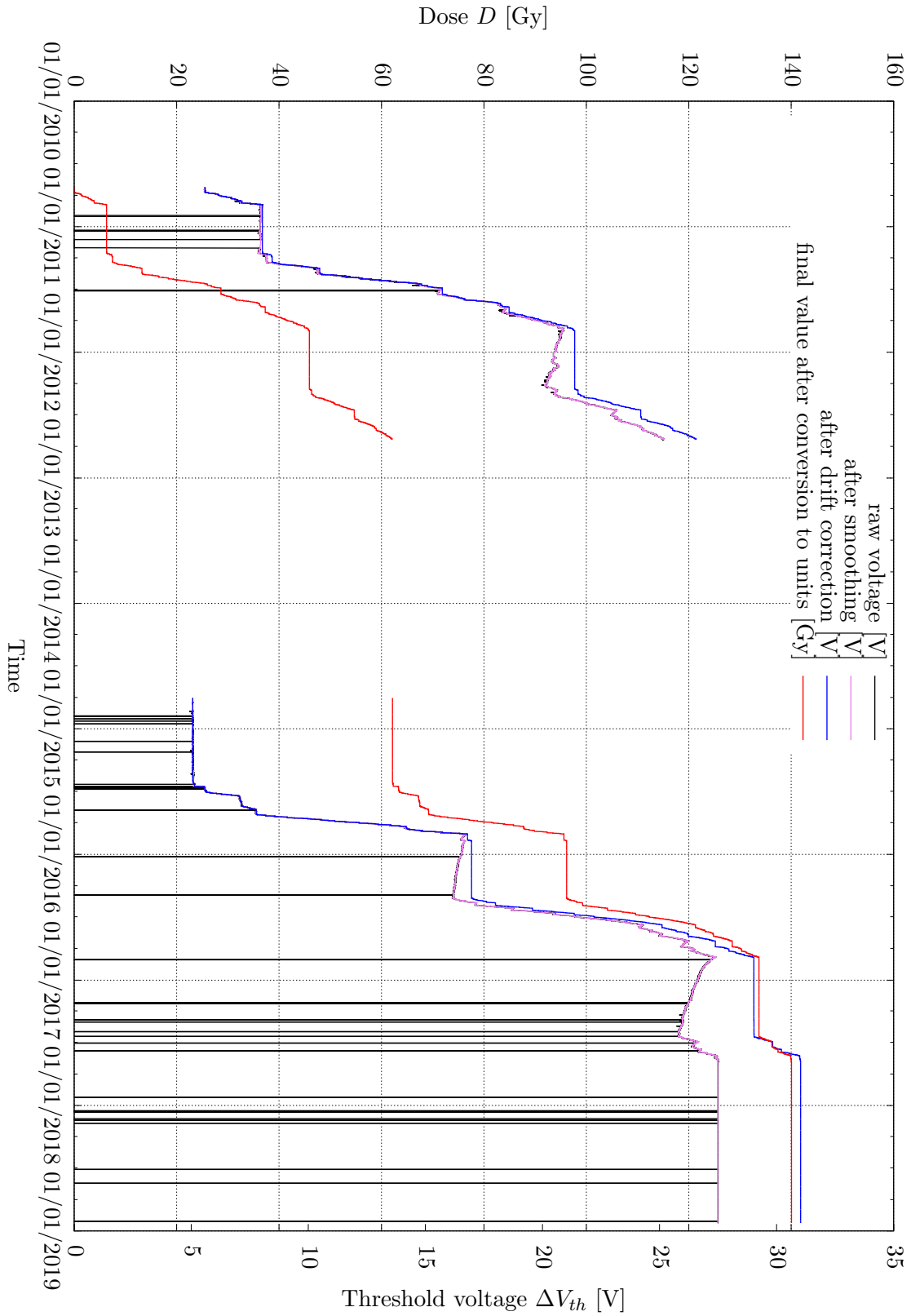


Figure 57: Final results for the 37 ‘Low Dose’ sensor with two LAAS during 2010-2018

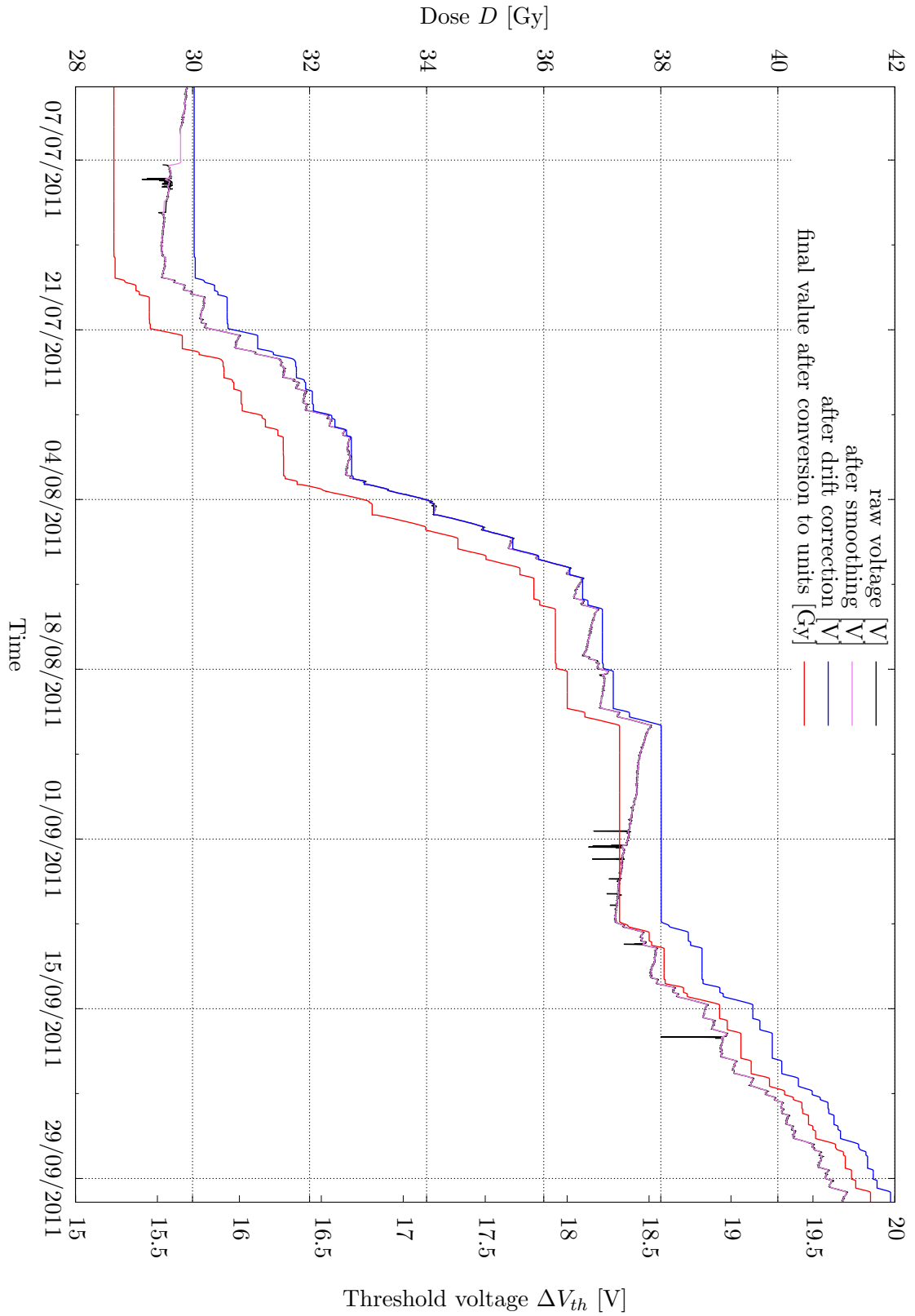


Figure 58: Signal processing performance for the 37 ‘Low Dose’ sensor with two LAAS in a 3 month timespan

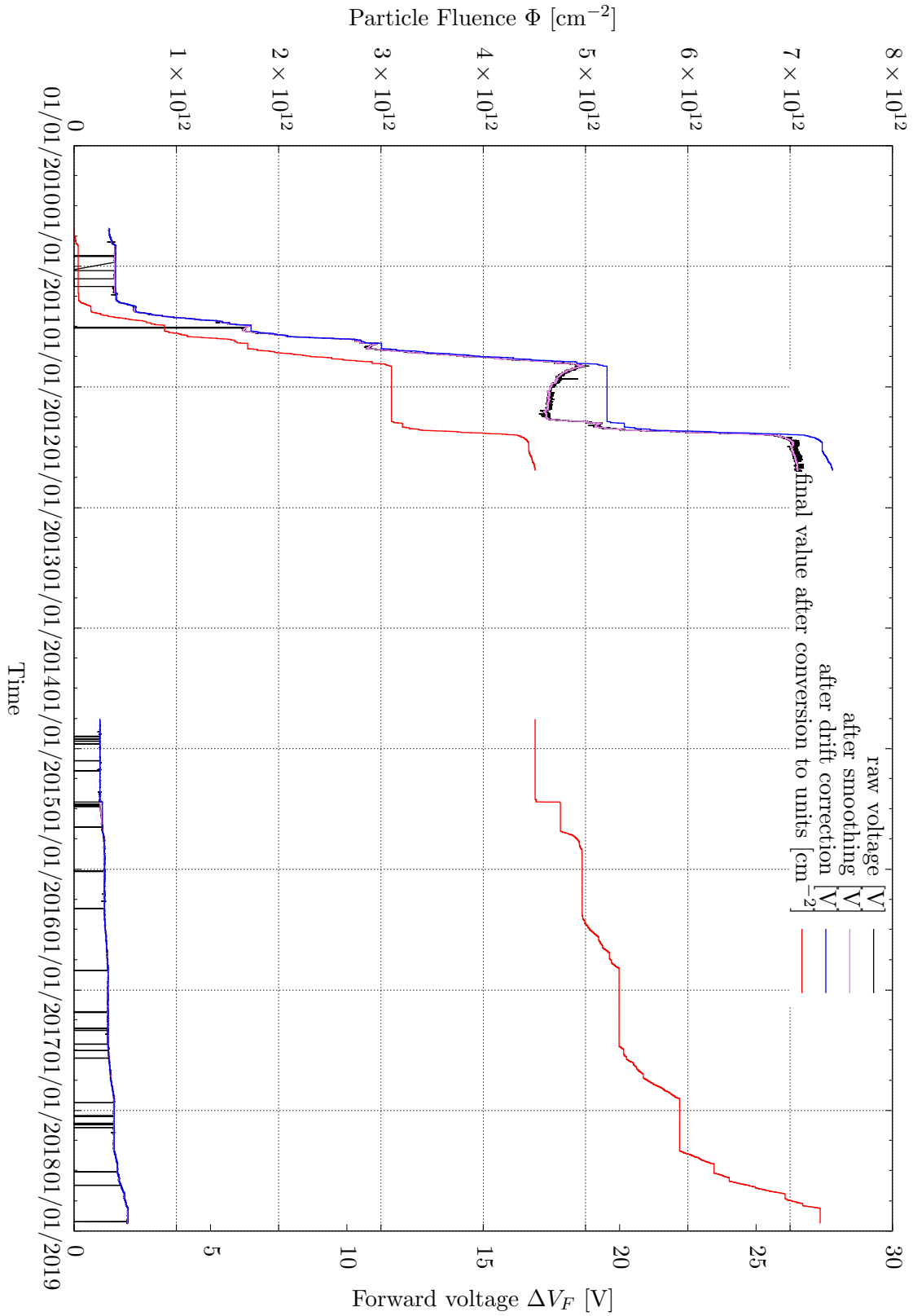


Figure 59: Final results for the 37 'Low Fluence' sensor with with one CMRP and one Si-2 during 2010-2018.

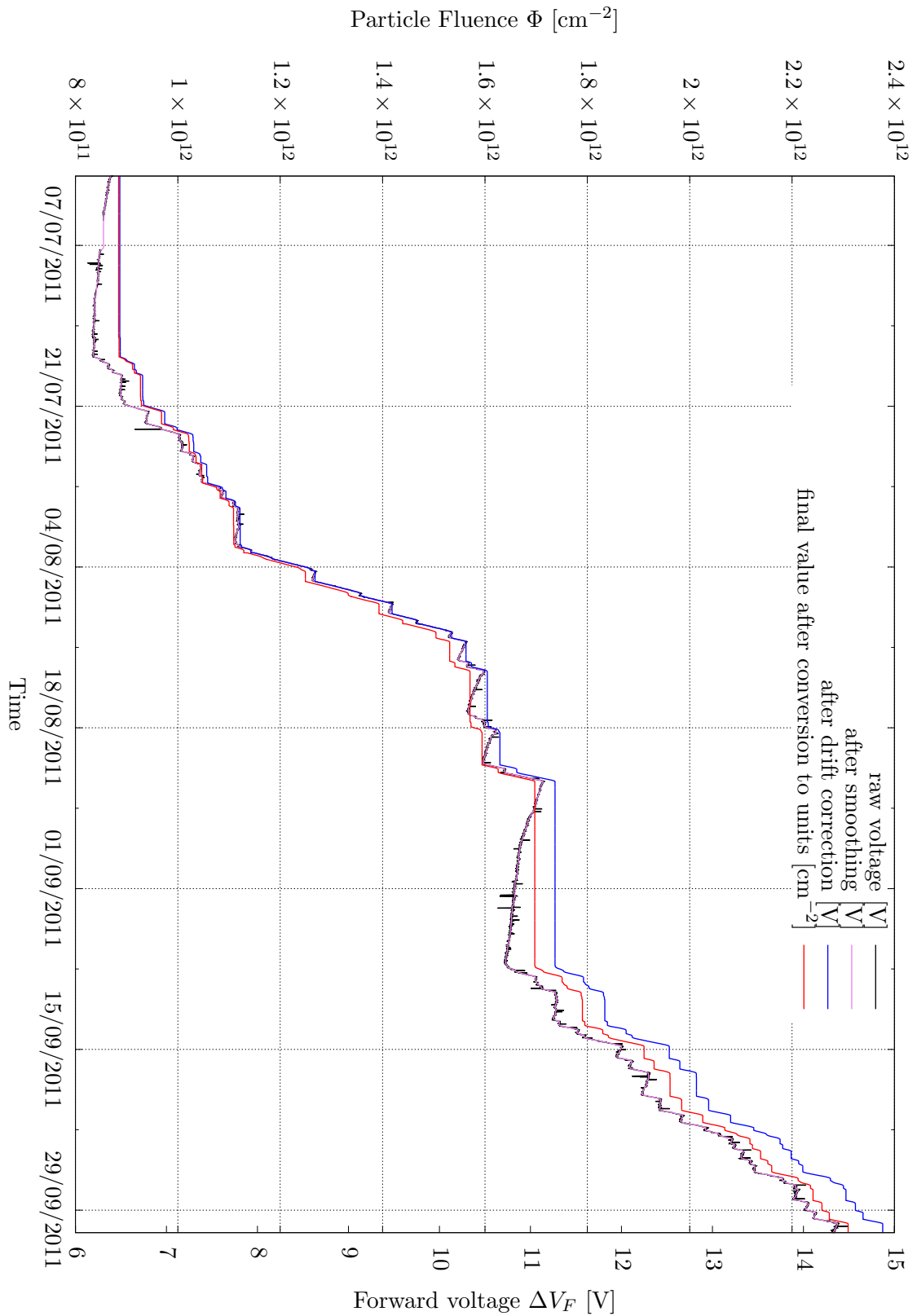


Figure 60: Signal processing performance for the 37 ‘Low Fluence’ sensor with one CMRP and one Si-2 in a 3 month timespan.

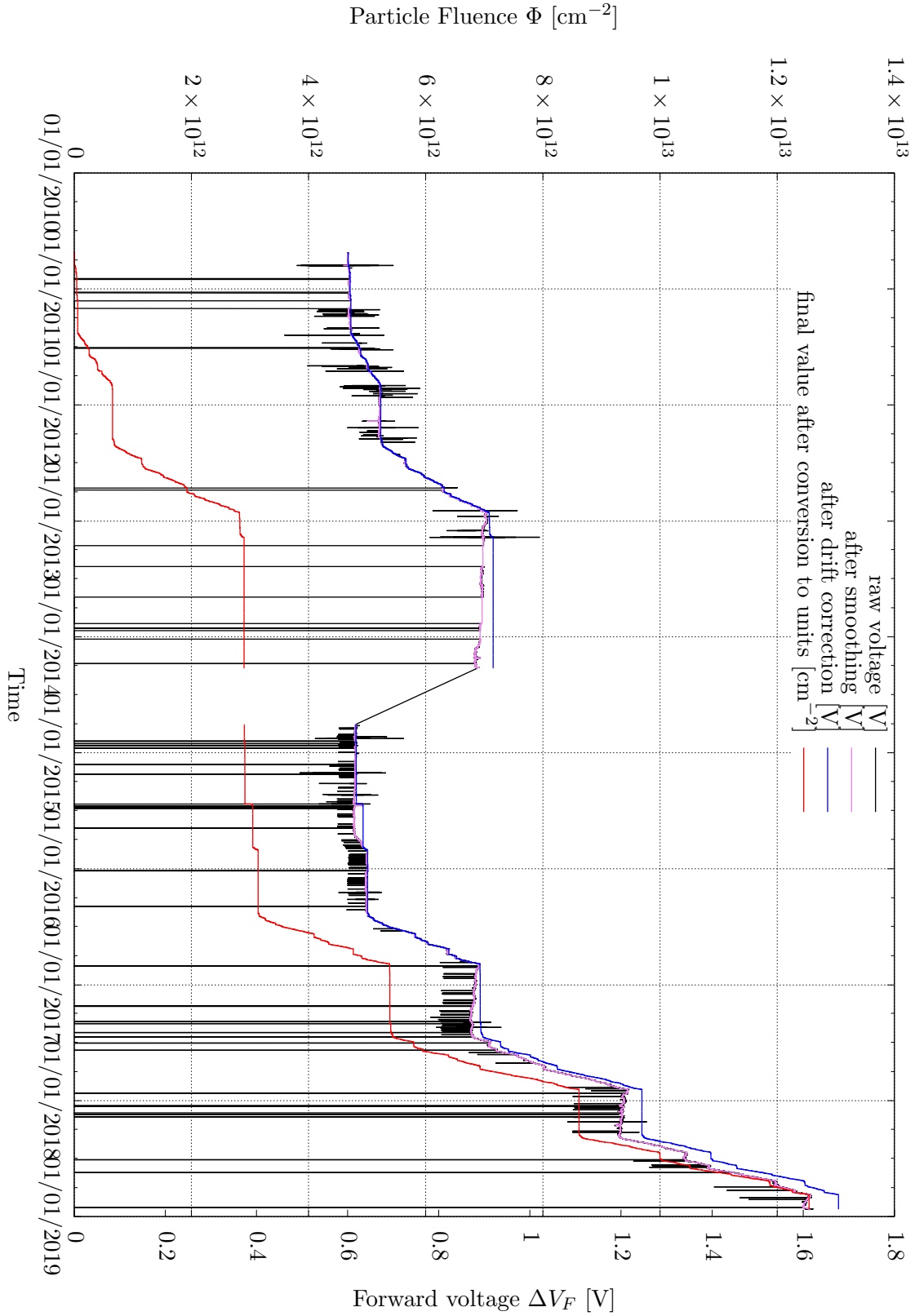


Figure 61: Final results for the 37 'High Fluence' sensor with two BPWs during 2010-2018.

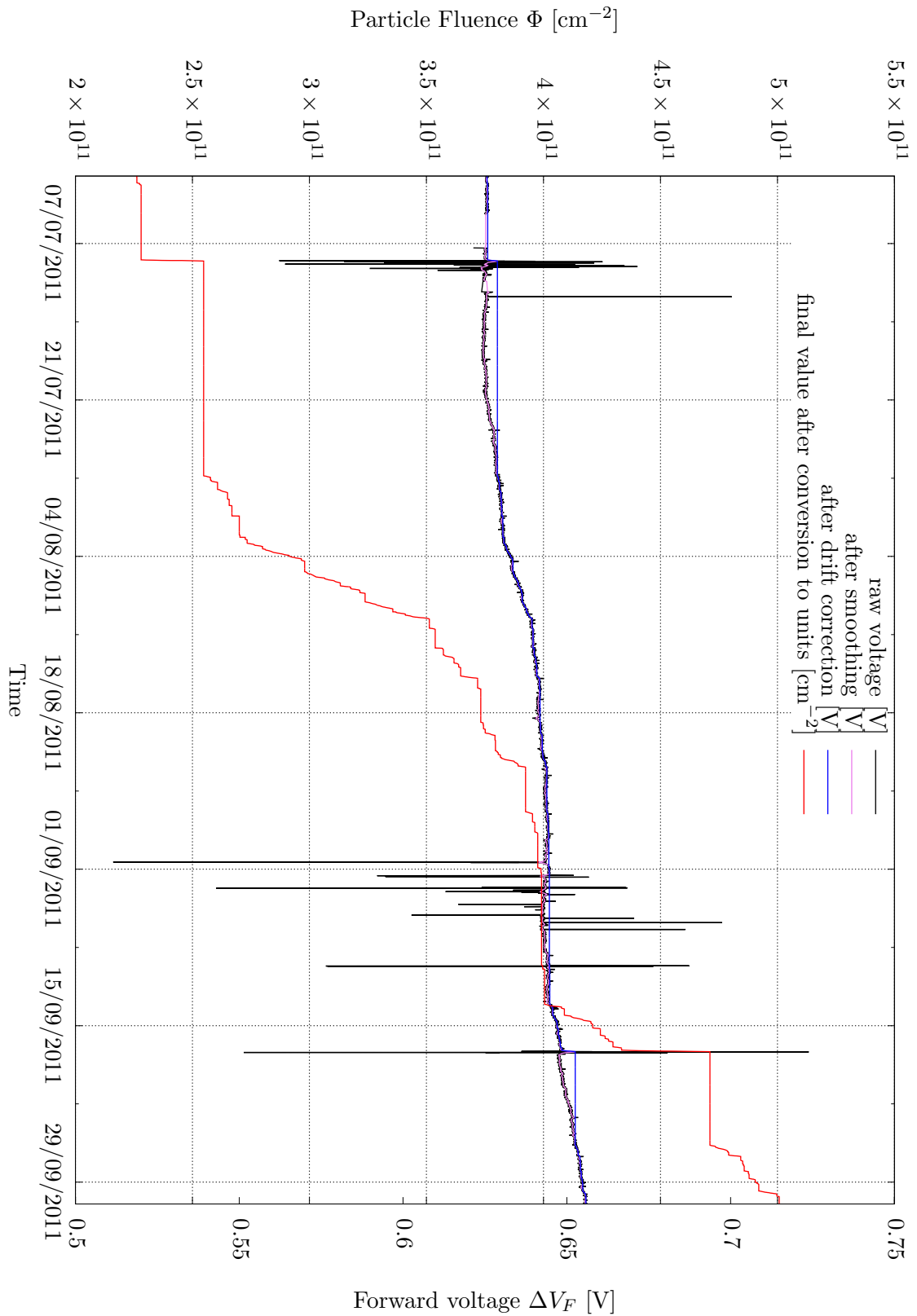


Figure 62: Signal processing performance for the 37 ‘High Fluence’ sensor with two BPWs in a 3 month timespan.

List of Algorithms

1	Kalman filter	37
2	Rauch-Tung-Striebel backward recursion [41]	38
3	Rauch-Tung-Striebel smoother [41]	39
4	Outlier-Robust Kalman Filter [2]	40
5	Outlier-Robust Kalman Smoother [2]	41

Listings

1	Compiling and running the LbARMS offline project	55
2	Exemplary structure of the voltage CSV file	55
3	Exemplary structure of the optional CSV file containing luminosity	56
4	An example of a matrix operation (computing a determinant) in the LbARMS offline program	57
5	An example of a correct output from the LbARMS offline program	57
6	A list of input arguments of the LbARMS offline program	59
7	An example of an error when the data files have wrong structure in the LbARMS project	59
8	An example of the offline MATLAB function (Rauch-Tung-Striebel smoother) in the LbARMS project	60
9	An example of a matrix operation (computing a determinant) in the WinCC OA project	62
10	An example of the model used to map ECS data stored in the Oracle database.	66
11	An example of a raw SQL that can be used to sample the database.	66

List of Figures

1	CERN Accelerator Complex	11
2	Side view of the LHCb detector.	12
3	Integrated luminosity at the LHCb Experiment recorded in 2010-2018. . . .	13
4	Creation of a stable alanine radical from L- α -alanine exposed to radiation. .	15
5	Space charge accumulation in an insulator. The electrodes are metal and silicon.	16
6	Scheme of a RADiation-sensitive Field-Effect Transistor (RadFET) and a forward biased p-i-n diode.	17
7	Reconstruction of the plot obtained by F. Ravotti [44] depicting calibration results ^g for the REM RadFET.	18
8	Reconstruction of the plot obtained by F. Ravotti [44] depicting calibration results ^h for the LAAS RadFET.	18
9	Reconstruction of the plot obtained by Ravotti [44] depicting calibration results ⁱ for the CMRP diode with a readout current of 1 mA.	21
10	Reconstruction of the plot obtained by Ravotti [44] depicting calibration results ^j for the BPW diode with a readout current of 1 mA.	22
11	Reconstruction of the plot obtained by Ravotti [44] depicting calibration results ^k for the LSB D Si-2 diode with a readout current of 25 mA.	22
12	Scope of the Experiment Control System (ECS). ECS controls and monitors many systems: DAQ, DCS, External Systems etc.	23
13	WinCC OA function-specific managers design. All managers are linked to the Event Manager and displayed in the console panel (Figure 14.)	25
14	WinCC OA console panel. It lists all managers of the project.	26
15	Part of the PARA module of the LHCb Active Radiation Monitoring System with visible tree-like structure of the datapoint elements.	26
16	Context diagram showing the environmental requirements of the LHCb On-line System.	27
17	The Active Radiation Monitoring set of sensors with a serial number 11. The box of Polymer-Alanine Dosimeter is on the left and the box containing CMRP and BPW p-i-n diodes with REM and LAAS RADFETs is on the right.	28
18	The detector layout of the radiation sensors. Each set is depicted as a blue rectangle with a serial number on it.	29
19	The M1 station layout of the radiation sensors. Each set is depicted as a blue rectangle with a serial number on it. The station is perpendicular to the z -axis that goes along the beamline in the detector layout in Figure 18. .	29

20	The cavern layout of the radiation sensors. Each set is depicted as a blue rectangle with a serial number on it. The detector layout (Figure 18) is highlighted with a light blue rectangle.	29
21	INFARMS panel used to display the integrated luminosity, voltage and temperature history for each device.	31
22	INFARMS main panel that lists the most recent voltage measurements. . .	32
23	INFARMS panel with a detector layout used to indicate the location of some devices.	32
24	Correction algorithm flowchart.	42
25	An example of a piecewise continuous function.	43
26	Testing variations of the muting parameter μ responsible for signal-to-noise ratio using Outlier-Robust Kalman Filter. The delay in the response to the signal is bigger with lower values of the muting parameter.	45
27	Testing muting parameter μ responsible for signal-to-noise ratio using Outlier-Robust Kalman Smoother. The ORKS removes the lag from the signal completely.	46
28	Testing the value of degrees of freedom s used for spike detection in the Outlier-Robust Kalman Filter.	46
29	Testing the value of degrees of freedom s used for spike detection in the Outlier-Robust Kalman Smoother.	47
30	Testing different values of window's length N with a wide spike.	48
31	Testing different values of window's length N with a step function.	48
32	REM dosimeters calibrated with PAD measurements ¹ obtained at the end of 2011, 2012, 2016 and 2017.	52
33	Directory tree of the offline project	54
34	Collaboration diagram of the correction algorithm in the LbARMS offline project.	56
35	Job cycle in the HTCondor batch service on lxplus.cern.ch.	56
36	Inheritance diagram for exception handling in the LbARMS project. . . .	59
37	Directory tree of the MATLAB implementation of the Kalman filtering part	60
38	New main panel of the LHCb Active Radiation Monitoring System	63
39	Panel with more detailed information about the correction algorithm. . . .	63
40	Panel allowing to restart the sensor's parameters.	64
41	LHCb panel that displays the values coming from the Radiation and Environment Monitoring Unified Supervision (REMUS) system.	64
42	An example of the plot generated by the 'dygraphs' library.	66
43	LbECSDData's home page containing datapoint structure of the ECS system.	67
44	LbARMS's home page containing responsive layouts of the radiation sensors.	68

45	Final correction algorithm results for the devices with a lower dynamic range placed in the cavern and the LHCb detector (without 37, 38, 39). The Low Dose sensors are in general considered inaccurate due to calibration issues.	71
46	Final correction algorithm results for the devices with a higher dynamic range placed in the cavern and the LHCb detector (without 37, 38, 39). All the High Fluence results are irrelevant, because no sensor in this layout reached the lower limit of the dynamic range of the BPW diode.	71
47	Final correction algorithm results for the devices with a lower dynamic range placed on the surface of the M1 station. All the Low Dose results are irrelevant, because the LAAS dosimeters were already saturated at the beginning of 2011. Sensors 12 and 13 reached the upper threshold of the dynamic range of the CMRP diode. High Fluence results should be considered instead in this case.	72
48	Final correction algorithm results for the devices with a higher dynamic range placed on the surface of the M1 station. The majority of the High Dose sensors are at the beginning of the lower threshold of the dynamic range of the REM RadFET. In the case of High Fluence sensors, only the sensor with a serial number 13 managed to enter the dynamic range.	72
49	Final correction algorithm results for the devices with a lower dynamic range that were highly irradiated (37, 38, 39). Both the Low Dose and the Low Fluence were quickly saturated and none of these results are relevant anymore.	73
50	Final correction algorithm results for the devices with a higher dynamic range that were highly irradiated (37, 38, 39). The High Dose sensors are already saturating, whereas the High Fluence sensors are still in the dynamic range of the BPW diode.	73
51	Summary of the final results for each ‘High Dose’ sensor	76
52	Summary of the final results for each ‘Low Dose’ sensor	77
53	Summary of the final results for each ‘High Fluence’ sensor	78
54	Summary of the final results for each ‘Low Fluence’ sensor	79
55	Final results for the 37 ‘High Dose’ sensor with two REMs during 2010-2018	80
56	Signal processing performance for the 37 ‘High Dose’ sensor with two REMs in a 3 month timespan	81
57	Final results for the 37 ‘Low Dose’ sensor with two LAAS during 2010-2018	82
58	Signal processing performance for the 37 ‘Low Dose’ sensor with two LAAS in a 3 month timespan	83
59	Final results for the 37 ‘Low Fluence’ sensor with with one CMRP and one Si-2 during 2010-2018.	84
60	Signal processing performance for the 37 ‘Low Fluence’ sensor with one CMRP and one Si-2 in a 3 month timespan.	85

61	Final results for the 37 ‘High Fluence’ sensor with two BPWs during 2010-2018.	86
62	Signal processing performance for the 37 ‘High Fluence’ sensor with two BPWs in a 3 month timespan.	87

List of Tables

1	Parameters of RADFETs used at the LHCb Experiment [42, 44].	19
2	Parameters of the p-i-n diodes of the LHCb Experiment [44, 42].	20
3	Polymer-Alanine Dosimeters (PAD) dose measurements obtained by M. Karacson [29] at the end of 2011, 2012, 2016 and 2017.	51
4	Calibration parameters for the REM dosimeter with their asymptotic standard errors provided by gnuplot software.	52
5	Software components of the LHCb Active Radiation Monitoring System. . .	53
6	Final correction algorithm results.	69

References

- [1] LHCb Trigger and Online Upgrade Technical Design Report. Technical Report CERN-LHCC-2014-016. LHCB-TDR-016, May 2014. URL <https://cds.cern.ch/record/1701361>.
- [2] G. Agamennoni, J. I. Nieto, and E. M. Nebot. An outlier-robust Kalman filter. In *2011 IEEE International Conference on Robotics and Automation*. IEEE, may 2011. doi:10.1109/icra.2011.5979605. URL <https://doi.org/10.1109/icra.2011.5979605>.
- [3] ATLAS Collaboration. Embedded Local Monitoring Box (ELMB). URL <http://atlasinfo.cern.ch/ATLAS/GROUPS/DAQTRIG/DCS/ELMB/elmb.html>.
- [4] V. Battista, M. Karacson, and G. Corti. Active Radiation Monitors in LHCb (CERN Summer Student Report). 2012.
- [5] M. Bostock. D3.js. URL <https://github.com/d3/d3>.
- [6] M. Buehler. Design curves for predicting fast-neutron-induced resistivity changes in silicon. *Proceedings of the IEEE*, 56(10):1741–1743, 1968. doi:10.1109/proc.1968.6728. URL <https://doi.org/10.1109/proc.1968.6728>.
- [7] CERN. Joint COntrols Project, 2019. URL <http://jcop.web.cern.ch/jcop-framework>.
- [8] L. Chua and S. M. Kang. Section-wise piecewise-linear functions: Canonical representation, properties, and applications. *Proceedings of the IEEE*, 65(6):915–929, June 1977. ISSN 0018-9219. doi:10.1109/PROC.1977.10589.
- [9] C. K. Chui and G. Chen. *Kalman Filtering with Real-Time Applications*. Springer Berlin Heidelberg, 1987. doi:10.1007/978-3-662-02508-6. URL <https://doi.org/10.1007/978-3-662-02508-6>.
- [10] T. A. S. Corporation. *Applied Optimal Estimation (The MIT Press)*. The MIT Press, 1974. ISBN 0262570483. URL <https://www.amazon.com/Applied-Optimal-Estimation-MIT-Press/dp/0262570483?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0262570483>.
- [11] G. Daquino, G. Corti, and G. Folger. Background radiation studies at LHCb using geant4. *IEEE Transactions on Nuclear Science*, 53(5):2907–2915, oct 2006. doi:10.1109/tns.2006.881544. URL <https://doi.org/10.1109/tns.2006.881544>.
- [12] Django Software Foundation . Django. URL <https://djangoproject.com>.

- [13] J. Duchi. Properties of the Trace and Matrix Derivatives, 2019. URL https://web.stanford.edu/~jduchi/projects/matrix_prop.pdf.
- [14] Encode OSS Ltd. Django REST framework. URL <https://www.django-rest-framework.org/>.
- [15] ETM. PVSS II Instruction, 2004. URL https://readthedocs.web.cern.ch/download/attachments/21178115/0407PVSS_II_GettingStarted_Basics_en.pdf?version=1&modificationDate=1299054760000&api=v2.
- [16] R. Experiments and Monitors. REM LOW-FADE SILICON MOSFET DOSIMETER. URL http://www.ee.nmt.edu/~anders/courses/ee521s12/radfet_datasheet.pdf.
- [17] J. Farrugia. Online Data Conversion for the LHCb Active Radiation Monitor System (ARMS). Aug 2015. URL <https://cds.cern.ch/record/2046150>.
- [18] Z. Ghahramani and G. E. Hinton. Parameter estimation for linear dynamical systems. 09 1996.
- [19] M. Glaser, F. Ravotti, and M. Moll. Dosimetry Assessments in the Irradiation Facilities at the CERN-PS Accelerator. In *2005 8th European Conference on Radiation and Its Effects on Components and Systems*, pages PI5–1–PI5–7, Sept 2005. doi:10.1109/RADECS.2005.4365632.
- [20] M. S. Grewal and A. P. Andrews. *Kalman Filtering*. John Wiley & Sons, Inc., nov 2014. doi:10.1002/9781118984987. URL <https://doi.org/10.1002/9781118984987>.
- [21] W. Herr, B. J. Holzer, and B. Muratori. 6.4 concept of luminosity. In *Accelerators and Colliders*, pages 140–146. Springer Berlin Heidelberg, 2013. doi:10.1007/978-3-642-23053-0_9. URL https://doi.org/10.1007/978-3-642-23053-0_9.
- [22] A. Holmes-Siedle. The space-charge dosimeter. *Nuclear Instruments and Methods*, 121(1):169–179, oct 1974. doi:10.1016/0029-554x(74)90153-0. URL [https://doi.org/10.1016/0029-554x\(74\)90153-0](https://doi.org/10.1016/0029-554x(74)90153-0).
- [23] A. Holmes-Siedle and L. Adams. RADFET: A review of the use of metal-oxide-silicon devices as integrating dosimeters. *International Journal of Radiation Applications and Instrumentation. Part C. Radiation Physics and Chemistry*, 28(2):235 – 244, 1986. ISSN 1359-0197. doi:[https://doi.org/10.1016/1359-0197\(86\)90134-7](https://doi.org/10.1016/1359-0197(86)90134-7). URL <http://www.sciencedirect.com/science/article/pii/1359019786901347>.
- [24] A. Holmes-Siedle, F. Ravotti, and M. Glaser. The dosimetric performance of RADFETs in radiation test beams. In *2007 IEEE Radiation Effects Data Workshop*.

- IEEE, jul 2007. doi:10.1109/redw.2007.4342539. URL <https://doi.org/10.1109/redw.2007.4342539>.
- [25] HTCondor Team. HTCondor. URL <https://research.cs.wisc.edu/htcondor/index.html>.
- [26] V. M. Jimenez-Fernandez, M. Jimenez-Fernandez, H. Vazquez-Leal, E. Muñoz-Aguirre, H. H. Cerecedo-Núñez, U. A. Filobello-Niño, and F. J. Castro-Gonzalez. Transforming the canonical piecewise-linear model into a smooth-piecewise representation. *SpringerPlus*, 5(1):1612, Sep 2016. ISSN 2193-1801. doi:10.1186/s40064-016-3278-y. URL <https://doi.org/10.1186/s40064-016-3278-y>.
- [27] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35, 1960. doi:10.1115/1.3662552. URL <https://doi.org/10.1115/1.3662552>.
- [28] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83(1):95, 1961. doi:10.1115/1.3658902. URL <https://doi.org/10.1115/1.3658902>.
- [29] M. Karacson, H. Aiginger, and G. Corti. Evaluation of the Radiation Environment of the LHCb Experiment, 2016. URL <https://cds.cern.ch/record/2243499>. Presented 12 Dec 2016.
- [30] Laboratory of Analysis and Architecture of Systems. LAAS RADFET supplier. URL <https://www.laas.fr/>.
- [31] LHCb Collaboration. *LHCb online system, data acquisition and experiment control: Technical Design Report*. Technical Design Report LHCb. CERN, Geneva, 2001. URL <https://cds.cern.ch/record/545306>.
- [32] LHCb Collaboration. The LHCb Detector at the LHC. *JINST*, 3(LHCb-DP-2008-001. CERN-LHCb-DP-2008-001):S08005, 2008. URL <https://cds.cern.ch/record/1129809>. Also published by CERN Geneva in 2010.
- [33] LHCb Collaboration, 2019. URL <http://lhcb.web.cern.ch/lhcb/>.
- [34] M. Moll. *Radiation damage in silicon particle detectors: Microscopic defects and macroscopic properties*. PhD thesis, Hamburg U., 1999. URL <http://www-library.desy.de/cgi-bin/showprep.pl?desy-thesis99-040>.
- [35] N. Neufeld. The LHCb online system. *Nuclear Physics B - Proceedings Supplements*, 120:105 – 108, 2003. ISSN 0920-5632. doi:[https://doi.org/10.1016/S0920-5632\(03\)01888-7](https://doi.org/10.1016/S0920-5632(03)01888-7). URL <http://www.sciencedirect.com/science/article/pii/S0920563203018887>. Proceedings of the 8th International Conference on B-Physics at Hadron Machines.

- [36] Oracle. Oracle database, 2019. URL <https://www.oracle.com/database/>.
- [37] E. Pauwels, H. De Cooman, M. Waroquier, E. O. Hole, and E. Sagstuen. On the identity of the radiation-induced stable alanine radical. *Phys. Chem. Chem. Phys.*, 12:8733–8736, 2010. doi:10.1039/C004380J. URL <http://dx.doi.org/10.1039/C004380J>.
- [38] L. Pira. Restful API for ECS data. Aug 2018. URL <http://cds.cern.ch/record/2636334>.
- [39] Z. Prouza, O. Obraz, B. Sopko, F. Spurný, A. Skubal, J. Kits, and F. Látal. Dosimetric Parameters of a New Czechoslovak Neutron Si Diode. *Radiation Protection Dosimetry*, 28(4):277–281, 09 1989. ISSN 0144-8420. doi:10.1093/oxfordjournals.rpd.a080517. URL <https://dx.doi.org/10.1093/oxfordjournals.rpd.a080517>.
- [40] C. E. Rasmussen. Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning*, pages 63–71. Springer Berlin Heidelberg, 2004. doi:10.1007/978-3-540-28650-9_4. URL https://doi.org/10.1007/978-3-540-28650-9_4.
- [41] H. E. Rauch, C. T. Striebel, and F. Tung. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, aug 1965. doi:10.2514/3.3166. URL <https://doi.org/10.2514/3.3166>.
- [42] F. Ravotti, M. Glaser, and M. Moll. Sensor catalogue data compilation of solid-state sensors for radiation monitoring. Technical Report CERN-TS-Note-2005-002. TS-Note-2005-002, CERN, Geneva, May 2005. URL <https://cds.cern.ch/record/835408>.
- [43] F. Ravotti, M. Glaser, M. Moll, C. Ilgner, B. Camanzi, and A. G. Holmes-Siedle. Response of RadFET dosimeters to high fluences of fast neutrons. *IEEE Transactions on Nuclear Science*, 52(4):959–965, Aug 2005. ISSN 0018-9499. doi:10.1109/TNS.2005.852709.
- [44] F. Ravotti, L. Dusseau, M. Moll, and E. Tsismelis. Development and Characterisation of Radiation Monitoring Sensors for the High Energy Physics Experiments of the CERN LHC Accelerator, 2006. URL <http://cds.cern.ch/record/1014776>. Presented on 17 Nov 2006.
- [45] L. Richardson and S. Ruby. *RESTful web services - web services for the real world*. 01 2007. ISBN 978-0-596-52926-0.
- [46] S. Sarkka and A. Nummenmaa. Recursive noise adaptive kalman filtering by variational bayesian approximations. *IEEE Transactions on Automatic Control*, 54(3): 596–600, mar 2009. doi:10.1109/tac.2008.2008348. URL <https://doi.org/10.1109/tac.2008.2008348>.

- [47] G. Sarabayrouse and F. Gessinn. Thick oxide MOS transistors for ionizing radiation dose measurement. *Radioprotection*, 29(4):557–572, oct 1994. doi:10.1051/radiopro/1994006. URL <https://doi.org/10.1051/radiopro/1994006>.
- [48] M. Schmidt, G. Fung, and R. Rosales. Fast optimization methods for l1 regularization: A comparative study and two new approaches. In *Machine Learning: ECML 2007*, pages 286–297. Springer Berlin Heidelberg. doi:10.1007/978-3-540-74958-5_28. URL https://doi.org/10.1007/978-3-540-74958-5_28.
- [49] J. M. Swartz and M. O. Thurston. Analysis of the effect of fast-neutron bombardment on the current-voltage characteristic of a conductivity-modulated p-i-n diode. *Journal of Applied Physics*, 37(2):745–755, feb 1966. doi:10.1063/1.1708249. URL <https://doi.org/10.1063/1.1708249>.
- [50] The PostgreSQL Global Development Group. PostgreSQL. URL <https://www.postgresql.org/>.
- [51] W. Thomas and K. Colin. gnuplot. URL <http://www.gnuplot.info/>.
- [52] J.-A. Ting, E. Theodorou, and S. Schaal. Learning an Outlier-Robust Kalman Filter. In *Machine Learning: ECML 2007*, pages 748–756. Springer Berlin Heidelberg. doi:10.1007/978-3-540-74958-5_76. URL https://doi.org/10.1007/978-3-540-74958-5_76.
- [53] J.-A. Ting, E. Theodorou, and S. Schaal. A kalman filter for robust outlier detection. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, oct 2007. doi:10.1109/iros.2007.4399158. URL <https://doi.org/10.1109/iros.2007.4399158>.
- [54] D. Vanderkam. dygraphs. URL <https://github.com/danvk/dygraphs>.
- [55] M. Waligórski, G. Danialy, K. S. Loh, and R. Katz. The response of the alanine detector after charged-particle and neutron irradiations. *International Journal of Radiation Applications and Instrumentation. Part A. Applied Radiation and Isotopes*, 40(10-12):923–933, jan 1989. doi:10.1016/0883-2889(89)90018-x. URL [https://doi.org/10.1016/0883-2889\(89\)90018-x](https://doi.org/10.1016/0883-2889(89)90018-x).