Optimization of the AERA data at the Pierre Auger Observatory - Developing a new interface for data handling and improving the radio reconstruction based on data measured by fluorescence telescopes

Dissertation

zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)

der Fakultät Mathematik und Naturwissenschaften
Fachgruppe Physik
der Bergischen Universität Wuppertal

vorgelegt von
Dipl. Phys. Sebastian Mathys
(geboren in Velbert)

Dezember 2018

Die Dissertation kann wie folgt zitiert werden:

urn:nbn:de:hbz:468-20190402-104804-2
[http://nbn-resolving.de/urn/resolver.pl?urn=urn%3Anbn%3Ade%3Ahbz%3A468-20190402-104804-2]

Von der Fachgruppe Physik der Bergischen Universität Wuppertal als Dissertation angenommen. Tag der mündlichen Prüfung: 21. Dezember 2018

Referent:
Prof. Dr. Karl-Heinz Kampert
(Bergische Universität Wuppertal)

Korreferent:
Dr. Tim Huege
(Karlsruher Institut für Technologie)

Contents

List of Figures

List of Tables

List of Abbreviations and Terms

Abbreviation/Term	**Description**
AMIGA	AMIGA
APE	Auger Package and Environment
AERA	Auger Engineering Radio Array
AERAlet	The part of the surface array with denser station spacing (433 m)
AEVB	AERA event builder
BUW	Bergische Universität Wuppertal
CDAS	Central Data Acquisition System
CLF	Central Laser Facility
CRS	Central Radio Station
DAQ	Data Acquisition
EAS	Extensive Air Showers
FADC	Fast Analog to Digital Converter
FD	Fluorescence Detector
FPGA	Field Programmable Gate Array
HEAT	High Elevation Auger Telescopes
Infill	The part of the surface array with denser station spacing (750 m)
KIT	Karlsruher Institute of Technology
LNA	Low-Noise Amplifier
LNB	Low-Noise Block down converter
LPDA	Log-periodic Dipole Antenna
MD	Muon Detector
NIKHEF	National Institute for Nuclear Physics and High Energy Physics in Amsterdam
Offline	Auger analysis framework
PMT	Photomultiplier Tube

RDS	Radio Detector Station
SD	Surface Detector
UHECR	Ultra-High-Energy Cosmic Rays
XLF	eXtreme Laser Facility

1 Introduction

Mankind has been fascinated by the universe for a long time. Already thousand years ago observations of celestial objects were made. This field of research, represented nowadays by astronomy, astrophysics, astroparticle physics and cosmology made a lot of of progress through the centuries but still today some fundamental questions about the universe remain unanswered. Astroparticle physics is a link between all the research fields mentioned above. It's a prime example how research, combining different analysing methods over large orders of magnitude including large structures in the universe and the smallest particles known by using the newest technical equipment, could yield new insights for phenomena in the universe.

To get a better understanding of the universe, cosmic rays are one of the important topics to study. Those are particles reaching the Earth after traveling from outer space with nearly the speed of light. They are mainly protons but also nuclei of heavier elements up to lead are detected, also neutrinos, electrons and photons. After the discovery of cosmic rays in 1912, the sources and acceleration mechanisms of cosmic rays are still under investigation. Exploding stars, called supernova, black holes or active galactic nuclei are currently favoured. If such a high energetic particle arrives at the Earth's atmosphere, they interact with the local nuclei and a so called extensive air shower (EAS) is produced. This is a cascade of particles traveling almost with the speed of light towards ground in the direction of the incoming particle.

In the context of this topic, large experiments like the Pierre Auger Observatory were built. It uses a hybrid detection technique consisting of particle detectors measuring the particle component of EAS on the ground and fluorescence telescopes measuring the energy deposition in the atmosphere. During the recent years, the radio detection could be established as another field of research for cosmic rays emitting radio waves in the MHz and GHz range. For this purpose, the Auger Engineering Radio Array (AERA) was built. It consists of autonomous detector stations measuring the electric field strength produced by cosmic rays.

The development of a new input and especially output library named *AERAROOTIOLib* for radio data produced by the AERA experiment is the central subject of this work. The flexible, fully ROOT [1] based input/output library improves the efficiency of data handling and analysis significantly. Besides reducing the data volume by 20% and speeding up the read-in part of the analysis by a factor of 5, the *AERAROOTIOLib* enables the production of important subsets of data for different purposes, e.g. monitoring. The output/write functionality is especially necessary to extract and analyse radio events in more details and for merging data with other detector components.

In chapter 2 a short theoretical introduction in the topic of cosmic rays and their interactions with the atmosphere is given. Chapter 3 describes the different detector types at the Pierre Auger Observatory including the data acquisition system and the framework which is used for analysing data produced in the experiment. The Auger Engineering Radio Array is described in chapter 4. It has been built up in different phases which are all described in detail. In addition, the AERA acquisition system with different trigger modes is described and some technical insights about the structure of radio data used in AERA are given. Chapter 5 presents the new input and output library, *AERAROOTIO*. Its general structure, the additional *AERAConverter* tool, data handling and data streams used in AERA, adaptions for the Auger analysis framework $\overline{\text{Off}}\underline{\text{line}}$ and important applications like SD-radio and radio-radio merging in the Central Data Acquisition System (CDAS), which is now possible for the first time, are described. Chapter 6 is focused on the procedure to generate and evaluate a reference dataset consisting of events measured by the fluorescence telescopes and the radio detectors. This dataset can be used as a benchmark for comparisons of different radio analyses. In chapter 7 adaptations for the *RdObserver* which is used in $\overline{\text{Off}}\underline{\text{line}}$ for analysing radio data to use input values based on the reconstruction of the fluorescence telescopes and first results are shown. Chapter 8 contains a short summary about this work.

2

Cosmic Rays and Extensive Air Showers

More than 100 years after their discovery in 1912, cosmic rays are still a topic of active research. This is because, to gain more knowledge and a deeper understanding about the universe, they are a key topic to study. Even after this long period, fundamental questions are unanswered, for example: where and what is the origin of those particles, and which are the acceleration mechanisms which lead to energies exceeding the maximum range of human made accelerators by several orders of magnitudes. Cosmic rays with energies up to 10^{20} eV [1] have been measured [2], and, because of these high energies, cosmic rays also form the basis of searches for physics beyond the standard model. Experiment and theory have to go hand in hand to solve abovementioned problems. This chapter provides an insight into measurable quantities like the energy spectrum and the composition of cosmic rays followed by a short theoretical discourse about possible sources and acceleration procedures. In addition, extensive air showers (EAS) are described. Since this work mainly focus on the radio component of EAS, a short introduction into the radio emission theory of cosmic rays is given.

2.1 Cosmic Rays

Particles which are moving with nearly the speed of light and hitting the Earth from outer space are called cosmic rays. Most of them are protons, but also nuclei of heavier elements up to lead, electrons, photons and neutrinos are also detected. Supernovae, black holes, and active galactic nuclei are currently favored sources of cosmic rays. Besides their scientific value, cosmic rays have direct influence in our daily life, such as solar flares disturbing the communication with satellites or the famous aurora borealis for example.

The idea of an extraterrestrial source for cosmic rays was first discovered by Viktor Franz Hess with his balloon flight campaigns in 1912. This was preceded by measurements of the ionization rate of air with an electroscope due to the natural radiation of earthbound nuclides by Theodor Wulf on the Eiffel tower in 1910 [3]. The ionization rate did not decrease as strong

[1] eV, the electron-volt, is the energy one electron gets after crossing an electric potential difference of one volt.

as expected with growing distance to the ground. This unexpected result and a speculated self-radiation of the Eiffel tower demanded further investigation and the need to reach higher altitudes. However, the Eiffel tower was the highest man made building at this time. To reach higher distances from the ground, Hess did the same measurements with an electroscope in a hydrogen balloon without safety equipment at heights up to 5.3 km. He verified the first decreasing and then increasing dependency between ionization rate and altitude. This led him to the conclusion of extra terrestrial radiation from outer space penetrating the atmosphere as the source [4]. For his efforts in the discovery of cosmic rays, he has been awarded with the Nobel Prize in 1936 [5].

2.2 Energy Spectrum

Concerning the questions about cosmic rays raised in the introduction, more information can be derived by investigating the cosmic ray energy spectrum, as the flux of cosmic rays is a strong indicator for the emission processes. In addition, the knowledge about the composition of cosmic rays and the existence of specific structures in the energy spectrum is of fundamental importance for understanding topics like the transition from galactic to extragalactic sources, or cosmic ray acceleration and propagation in general. Spanning more than 32 orders of magnitudes in the flux and more than 11 orders in the energy up to 10^{20} eV, the cosmic ray energy spectrum is one of the best measured quantities in physics. In Fig. 2.1 the flux is shown. For a better visualization of small deviations, the flux is scaled with an additional factor of $E^{2.5}$ in this log-log plot. The lower part of the spectrum is dominated by the influence of the magnetic field of the Earth and the flux of cosmic rays is modulated by solar activities. Particles with less than 10^9 eV are not energetic enough to overcome the Earth's magnetic field (geomagnetic cut-off). For higher energies, the spectrum is described by a broken power law function with negative exponent of the form

$$\frac{dN}{dE} \propto E^{-\gamma}, \tag{2.1}$$

indicating a non-thermal processes accelerating those particles. γ is the *spectral index* which is not constant over the energy range but changes at some characteristic points of the spectrum: up to $\sim 4 \times 10^{15}$ eV, the so called *knee*, the spectral index γ is 2.7. Up to $\sim 4 \times 10^{17}$ eV, the spectrum becomes steeper, resulting in $\gamma = 3.1$. This transition region is called the *heavy knee* or also the *second knee* considering that γ increases to 3.3 between $\sim 4 \times 10^{17}$ eV and $\sim 4.1 \times 10^{18}$ eV. Cosmic rays with energies of more than $\sim 10^{18}$ eV are called *ultra-high-energy cosmic rays* (UHECR).

In order to explain the behavior of the energy spectrum, models for the sources of cosmic rays and their acceleration and propagation mechanisms have to be established. For example, the knee is where galactic sources are no longer powerful enough to further accelerate light elements. A likely source for cosmic rays in this range are shock regions from supernova explosions where cosmic rays undergo *Fermi acceleration* [17].

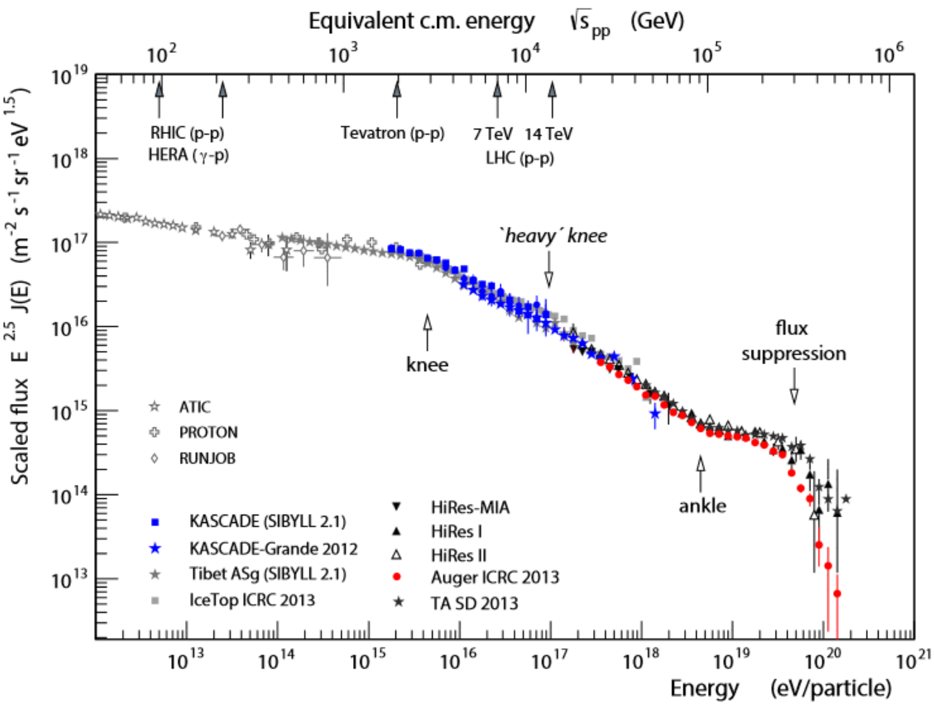


Figure 2.1: Energy spectrum of cosmic rays [6]. Shown is the particle flux depending on the primary energy measured by balloon or satellite borne experiments (ATIC [7], PROTON [8], RUNJOB [9]) and experiments located onto the surface of the Earth (Tibet [10], HiRes [11], IceTop [12], TA [13], KASCADE [14], KASCADE-Grande [15] and the Pierre Auger Observatory [16]).

This theory can be proven by looking into the energy spectrum for each element individually showing a composition dependent energy shift in the position of the knee as explained in [18]. Connected to this argument is the assumption that the magnetic field of our milky way is not strong enough to prevent particles from leaking out of the galaxy, which is more likely for lighter particles. The *ankle* hints the transition of cosmic rays from galactic to extragalactic sources [19]. Another theory, the *dip-model* refers the change in the spectrum due to the process of $e^{\pm}$ pair production during propagation through the cosmic microwave background (CMB) which modifies the source spectrum [20]. Suitable sources for UHECR are jets of active galactic nuclei, gamma-ray bursts and radio galaxies which can be checked by the Hillas criterion, see Fig. 2.2.

The end of the spectrum at $\sim 10^{20}\,\text{eV}$ features a cut-off in the flux of cosmic rays. This is predicted by Greisen [21], Zatsepin and Kuzmin [22] in the *GZK-effect*. Protons moving through the CMB with more than $6 \times 10^{19}\,\text{eV}$ can interact with photons of the cosmic microwave background, resulting in an excited state decaying after $10^{-23}\,\text{s}$:

$$\begin{aligned} p + \gamma_{\text{cmb}} &\rightarrow \Delta \rightarrow p + \pi^0 \\ p + \gamma_{\text{cmb}} &\rightarrow \Delta \rightarrow n + \pi^+ \,. \end{aligned} \tag{2.2}$$

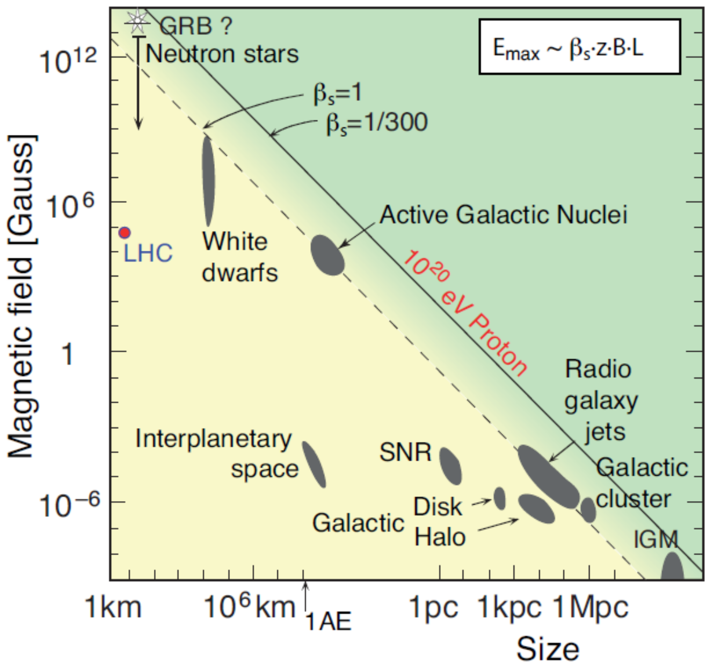


Figure 2.2: The Hillas plot. The size of an object of possible accelerators for cosmic rays and the magnetic field strength at its location are shown. Objects below the solid line are not able to accelerate protons up to 10^{20} eV and objects below the dashed line are not able to accelerate iron up to 10^{20} eV even under perfect conditions. [23]

Considering the photon density in the CMB of 412 photons/m^3, the mean free path for the p + γ_{cmb} interaction can be calculated thus deriving upper limits for the distance between the source of cosmic rays and the Earth. A commonly used simulation tool for the propagation of cosmic rays which accounts for all the effects is CRPropa [24].

2.3 Mass Composition

As mentioned in section 2.2, the detailed knowledge of the chemical composition of cosmic rays offers insights about their origin. However, concerning the steep decrease in the flux as energy increases, different detection techniques are required. In the lower energy range, cosmic rays are measured directly by satellite or balloon experiments at high altitudes. Most of the cosmic rays arriving at Earth are hydrogen nuclei (85 %), followed by helium nuclei (12 %), and heavier elements (3 %). At energies above 10^{15} eV, particles can only be measured indirectly with large detector arrays located on the surface of the Earth, like the Pierre Auger Observatory. This is because of the rarity of particles, for example at energies above 10^{19} eV, the rate is just one in a century per km^2. Based only on the indirect measurements, the composition can just be estimated with higher uncertainties. Current results favor a trend towards a lighter composition above the *ankle* and a more and more heavy composition, climbing up to iron, for the highest energies. This is illustrated in Fig. 2.3, showing the shower maximum X_{max} and the RMS $\sigma(X_{\text{max}})$ over the energy, which are two typical quantities related to composition, that

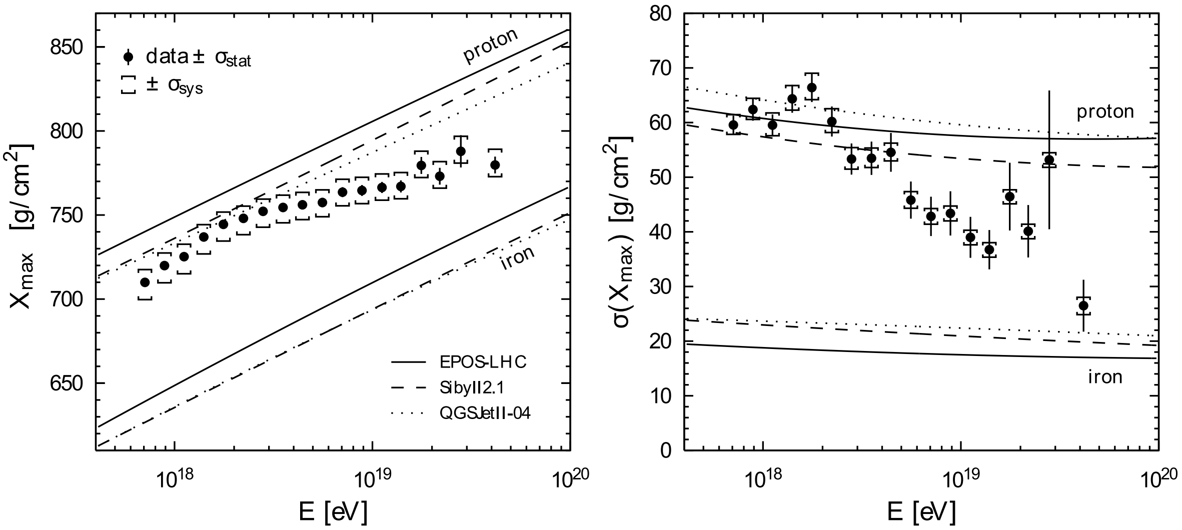


Figure 2.3: Left: X_{max} distribution for higher energies measured by the Pierre Auger Observatory showing a trend towards a heavier composition [25]. Also drawn are lines for a pure-proton or pure-iron composition calculated for different interaction models (QGSJetII-04 [26], Epos-LHC [27] and Sibyll2.1 [28]). Right: The distribution of $\sigma(X_{\mathrm{max}})$ representing the width of the X_{max} distribution shows a similar behavior favoring showers induced by heavy nuclei.

are estimated in indirect measurements. The point at which the shower reaches its maximum number of particles is called X_{max}, see section 2.4.

The composition of cosmic rays has a lot of similarities with the known distribution of matter in our solar system. A large discrepancy is visible in the lighter elements Lithium, Beryllium and Boron. This is because of while traveling through the interstellar medium, cosmic rays traverse regions with dust particles and gas atoms and heavier elements break up into lighter elements in inelastic interactions called spallation. The density of the interstellar medium can be approximated with 1 proton per cm^3. With the number of spallation products, one can calculate that cosmic rays have to cross about $10\,\mathrm{g/cm^3}$ on their way to the Earth [29]. One can conclude that cosmic rays reside mostly in thin media before escaping the galaxy.

Neutrinos are also present in cosmic rays. They are not deflected by magnetic fields in the universe like charged particles and can pass through accumulations of matter almost unhindered. Those properties make them good candidates to study the sources of cosmic rays. The extremely low cross section with matter is simultaneously a crucial point for air shower experiments. Only almost horizontal air showers or air showers crossing a huge amount of matter like mountains are suitable for neutrino experiments [30]. Additionally, neutrinos from air showers crossing the Earth from the opposite side can be detected [31].

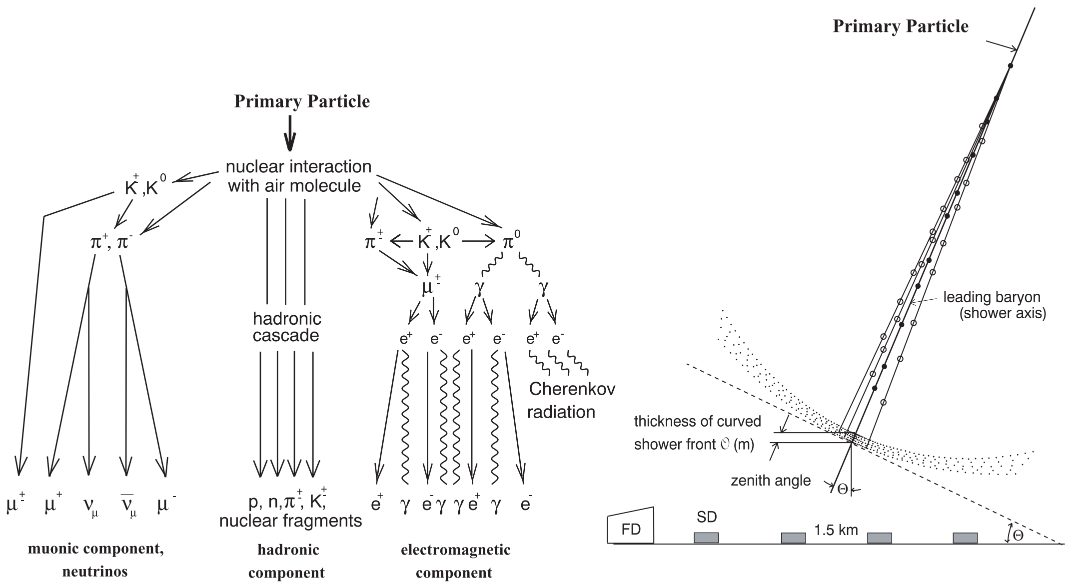


Figure 2.4: Schematic illustration of an extensive air shower with its different components in the left and the shower axis with the shower front in the right [34].

2.4 Extensive Air Showers

Regarding the energy spectrum of cosmic rays, the disadvantage of the small detection area of satellite and balloon borne experiments is clearly visible. Since time can't be manipulated, compensating for low statistics can be accomplished by indirect measurements with large experiments on the ground detecting so called extensive air showers (EAS). EAS were discovered by experiments done by Bothe, Kohlhörster in Berlin [32] and by Pierre Auger in the Swiss Alps in 1938 [33]. Auger found events occurring in detectors in a distance of $\approx 300\,\mathrm{m}$ at the same time. He concludes, that the measured event originates to the same source producing a cascade of particles.

2.4.1 Air shower properties

EAS are produced when the highly energetic cosmic ray particle, referred to as the primary particle, arrives at the Earth's atmosphere and interacts with local nuclei. In these interactions, secondary particles are produced which are energetic enough to interact themselves with further nuclei producing yet more particles. A cascade of particles, the extensive air shower, therefore develops along the direction of the incoming primary particle almost with the speed of light towards ground and is measured by detectors on the surface, as shown in Fig. 2.4. The central region of the EAS is called the *shower core*. Due to the particles' speed, the cascade arrives at the detector as a shower front with a thickness of a few meters. The primary particle itself does not reach the ground due to energy loss and interaction processes. Thus all conclusions about the primary particle have to be drawn out of the measured secondary particles.

Since different type of particles undergo different interaction mechanisms, the EAS is described as having three different components:

- *the electromagnetic component*: photons or electrons, i.e. resulting from the decay of neutral mesons:

$$\begin{aligned} &\pi^0 \to \gamma + \gamma \\ &\pi^0 \to \gamma + e^+ + e^- \,. \end{aligned} \tag{2.3}$$

  Electrons produce additional photons by the processes of bremsstrahlung

$$e^\pm + N \to N + e^\pm + \gamma \tag{2.4}$$

  with nucleus N and photons create additional electrons via pair production

$$\gamma + N \to N + e^+ + e^- . \tag{2.5}$$

  The EAS cascade evolves until the secondary particles are produced with less than the critical energy of $E_\mathrm{c} \approx 81\,\mathrm{MeV}$ in each interaction. Below this energy, energy loss due to ionization dominates.

- *the hadronic component* develops near to the shower axis and consists of protons, neutrons, pions and K mesons. As shown in (2.3), the neutral pion decays into two photons contributing to the electromagnetic cascade. The kaons and charged pions live long enough to produce further mesons in strong interactions and therefor contribute to the hadronic component. Kaons or charge pions that fail to interact further decay into muons.

- *The muonic component* extends to the ground and is made up of the muons resulting from the pion and kaon decays

$$\begin{aligned} &\pi^\pm \to \mu^\pm + \nu_\mu + (\overline{\nu}_\mu) \\ &K^\pm \to \mu^\pm + \nu_\mu + (\overline{\nu}_\mu) \,. \end{aligned} \tag{2.6}$$

To estimate how many particles will reach the ground, the *atmospheric depth*, in $\mathrm{g/cm^2}$, is used. It is an estimation of the amount of matter that particles have to cross as they pass through the atmosphere. For a vertical trajectory from the top of the atmosphere to sea level, the atmospheric depth is around $1030\,\mathrm{g/cm^2}$ [35]. In addition, one can calculate the mean free path, λ, for protons, which is dependent of the cross section σ and the density of the absorbing media, n, as:

$$\lambda_I = \frac{1}{n\,\sigma} \cong 90\,\mathrm{g/cm^2} \,. \tag{2.7}$$

From this follows that the atmosphere has a thickness of about 11 hadronic interaction lengths. For electrons and photons, the radiation length describing an energy loss, is used. This value for air is $\sim 37\,\mathrm{g/cm^2}$ resulting in 27 radiation lengths for the atmosphere. The first hadronic interaction of the primary particle with the atmosphere takes place at an altitude of about 20 km above sea level. When the EAS starts to develop, the hadronic component dominates, while on the ground, most of the particles are photons, followed by electrons and muons.

Heavier elements like iron have a larger nucleus and therefore a larger cross section resulting in a smaller interaction length. Thus, an extensive air shower induced by iron starts to develop higher in the atmosphere than a proton induced one. Since the density of the atmosphere in higher regions is lower, decays producing muons are preferred. If the number of muons and electrons, as well as the incidence angle, is known, one can draw conclusions on the mass and energy of the primary particle.

A quantitative approach to model extensive air showers was provided by Heitler [36]. In this model, a primary particle produces two new similar particles, each with half of the primary energy, after one interaction length. This is repeated for each particle and each interaction length until the critical energy, E_c, for each particle is reached. Because this is the point where the shower is the largest, it is called the shower maximum. After this point, the cascade starts to thin out. At the shower maximum, the number of particles can be calculated as

$$N_\mathrm{max} = E_0/E_\mathrm{c} \tag{2.8}$$

and the corresponding atmospheric depth of shower maximum, X_max, for the EAS is

$$X_\mathrm{max} = \lambda \frac{\ln(E_0/E_c)}{\ln 2} \tag{2.9}$$

For heavier elements, one can interpret the nucleus with mass A as a superposition of nuclei each with an energy of E_0/A. This results in

$$X_\mathrm{max} \propto \ln\left(\frac{E_0}{A \cdot E_c}\right) \tag{2.10}$$

which again shows a shower maximum located in higher regions of the atmosphere. Due to the cross section, iron induced showers have a higher first interaction point compared to ones induced by protons. A typical shower development reproduced with the EAS simulation code CORSIKA [37] can be seen in Fig. 2.5.

Several techniques exist to detect extensive air showers, e.g. fluorescence telescopes, water Cherenkov detectors, scintillators and radio antennas, see Fig. 2.6. The Pierre Auger Observatory combines all of the above named techniques and is described in chapter 3.

2.5 Radio detection of extensive air showers

In the early 1960s, in addition to the particle component of an EAS, detectable signals in the radio frequency regime were predicted. After the first successful observation of radio emission of EAS, taking place immediately after the establishment of the theoretical concepts [40],

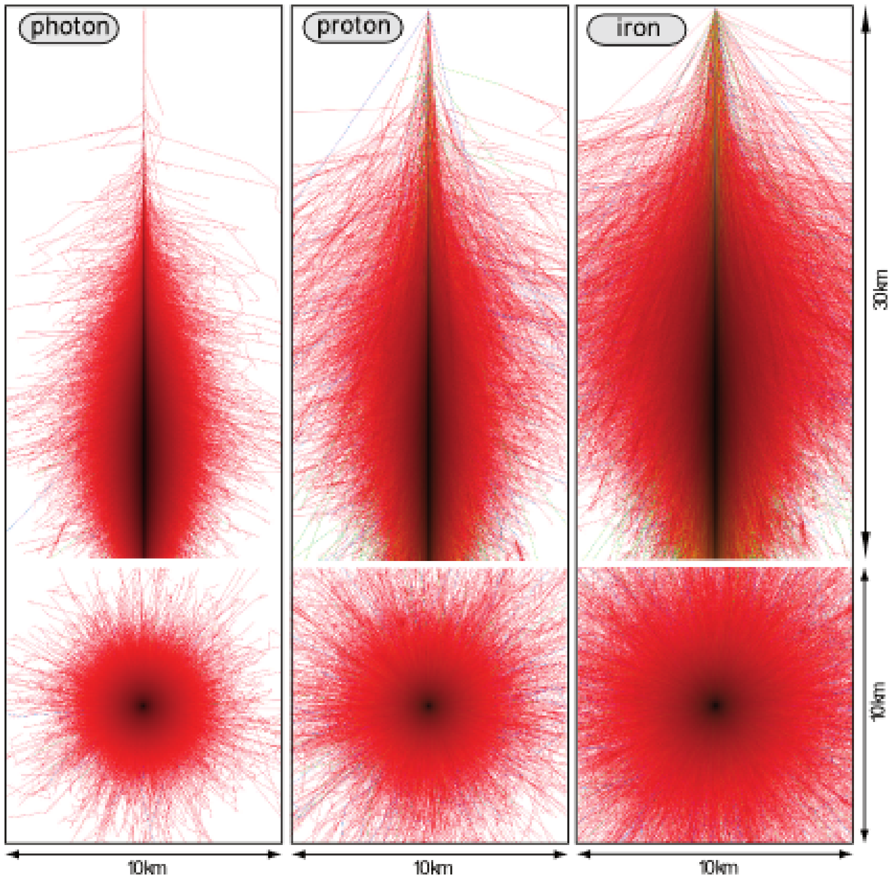


Figure 2.5: Extensive air showers induced by CORSIKA simulated photon, proton and iron primaries with an energy of 10^{14} eV [38] (modified from [39]). The top pictures show the side view and the bottom pictures show a xy-projection. The particle type is encoded in color red (electrons, positrons and photons), green (muons) and blue (hadrons). Darker colors corresponds to higher densities. For showers induced by heavier particles, the shower development starts higher in the atmosphere and the footprint is larger.

the radio detection technique more or less disappeared due to technical limitations. In the last 15 years, the LOPES [41] and CODALEMA [42] experiments and especially the AERA experiment (see chapter 4) overcame this obstacle by applying digital detection techniques and thus contributed to establishing the radio detection of air showers and the understanding of the radio emission mechanism. On the theoretical side, radio simulation codes based on different description of the mechanism models exist. It can be shown that a convergence exits between them and that they predict the measured data consistently [43]. The emission in the MHz range is dominated by two effects, geomagnetic and charge excess [44]. The first was brought up by Kahn and Lerche [45] and describes charged particles being deflected in the geomagnetic field. This leads to a time-varying transverse current and coherent dipole radiation in the radio frequency band which is linearly polarized in the direction of the Lorentz force ($\vec{v}$ x $\vec{B}$). Thus, the emission does not depend on the location of the observer and due to

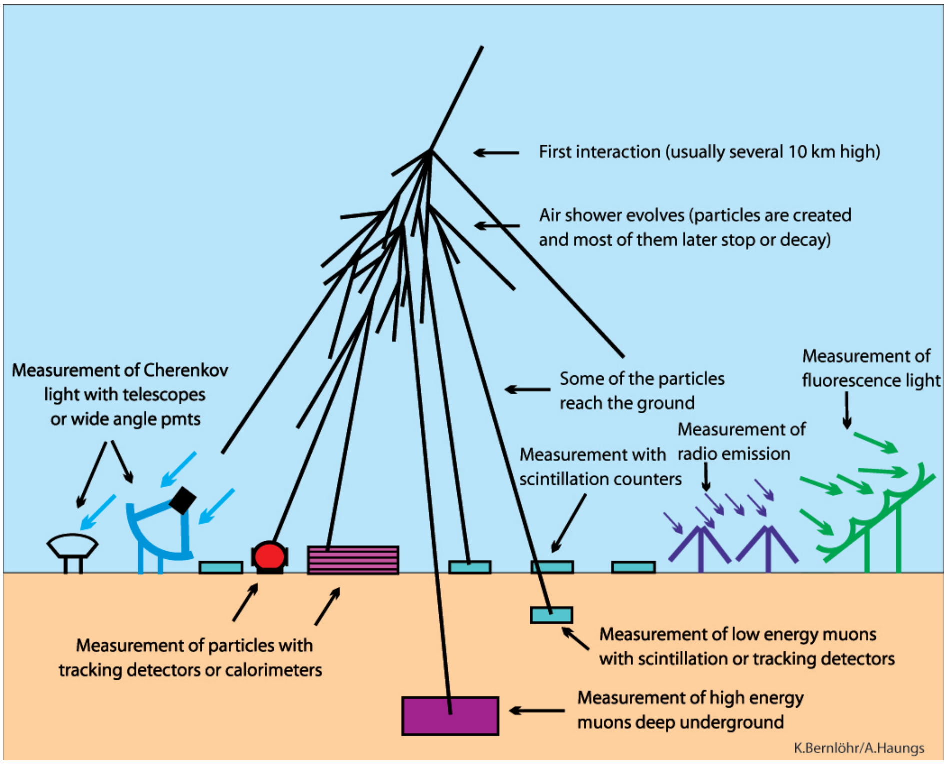


Figure 2.6: Sketch of an extensive air shower observed with different detection techniques, i.e. Cherenkov and fluorescence telescopes, scintillators, calorimeters, radio telescopes and muon trackers [6].

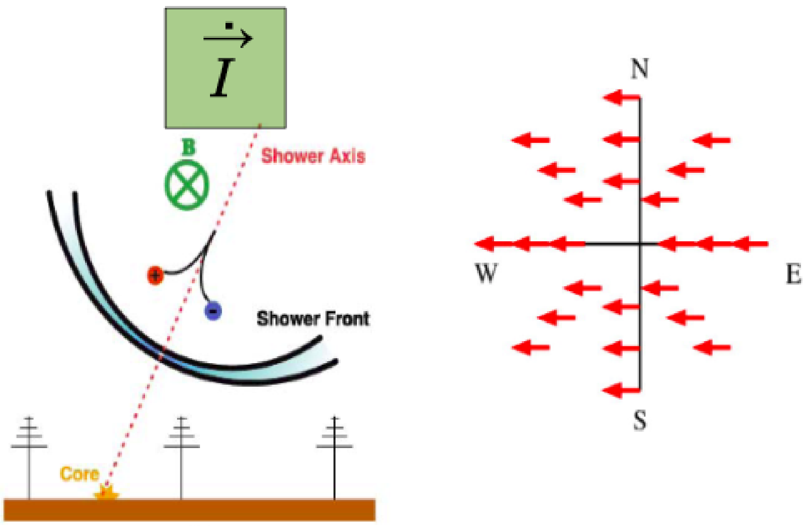


Figure 2.7: Illustration of the geomagnetic induced radio emission mechanism on the left and the polarization pattern measured on the ground on the right [46].

the high velocity of the moving dipole, the emission is strongly beamed in forward direction, see Fig. 2.7.

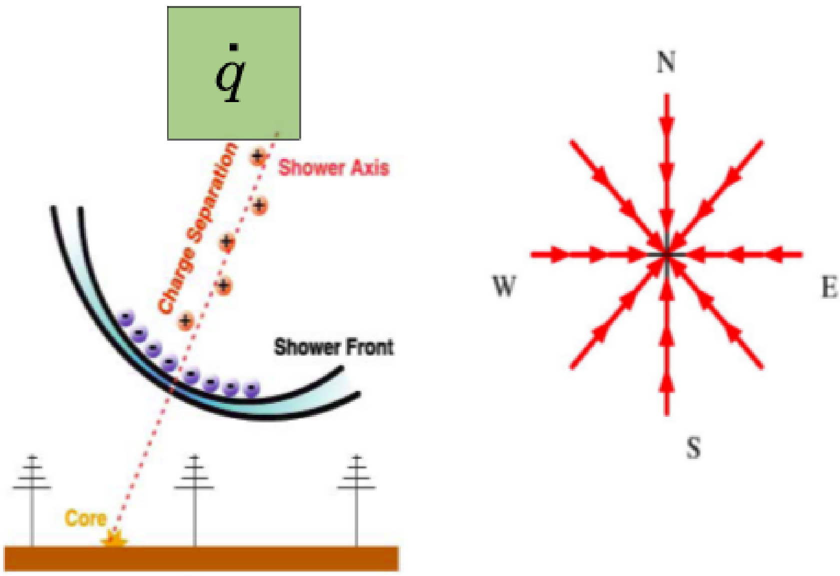


Figure 2.8: Illustration of the charge excess induced radio emission mechanism on the left and the polarization pattern measured on the ground on the right [46].

The second emission mechanism, established by Askaryan in 1962, describes the effect of the time-varying net charge excess in an EAS. This net charge is the consequence of the charge separation caused by electrons knocked out of the air molecules, which then move along with the air shower disk. This results in a linearly polarized electric field vector radially aligned with respect to the direction of the air shower axis, see Fig. 2.8. The charge excess is also influenced by the depletion of positrons due to annihilation with electrons. The time variation however is a result of the changing number of charged particles during the air shower development. The emission is based on the Cherenkov effect triggered by particles moving faster than the speed of light in air.

Both mechanisms, each with different polarization patterns, interfere resulting in an east-west asymmetry in the radio footprint with a typical bean-like shape. The emission strength due to the geomagnetic effect is dependent on the angle between the shower axis and the magnetic field of the Earth. Thus, by studying the polarization pattern measured on the ground, the ratio, a, between the two mechanism can be derived and expressed by

$$a \equiv \sin\alpha \frac{|E_C|}{|E_G|}, \tag{2.11}$$

where E_C is the charge excess and E_G the electromagnetic contribution and α the geomagnetic angle. In Fig. 2.9, the polarization angle measured by AERA and the one predicted by CoREAS / REAS3 [47], a simulation tool which is based on CORSIKA which includes the theory of the radio emission is shown, for several events. With the result of $a = 0.14 \pm 0.02$ one can conclude, that the charge excess only contributes little and that the geomagnetic effect is the dominant contribution to the radio emission.

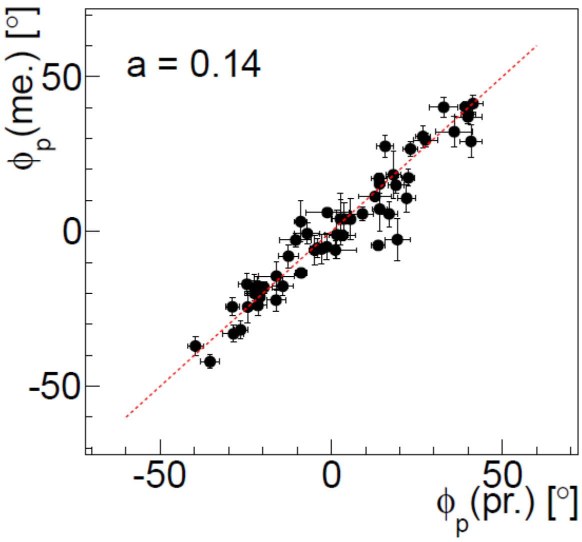


Figure 2.9: The polarization angle measured by AERA and the one predicted for several events with a fraction $a = 0.14 \pm 0.02$ is shown [48].

Another approach for detecting radio signals emitted by EAS in the GHz range has been shown by accelerator based measurements with an electron beam [49]. The results have been interpreted as Molecular Bremsstrahlung radiation (MBR) where electrons from molecules ionized in the EAS are scattered in the field of neutral atmospheric molecules. This leads to an unpolarized, isotropic emission. Benefits of this detection technique are the low atmospheric absorption below 0.5 dB/km and the low-cost of available commercial satellite hardware in the C-band 3.4 − 4.2 GHz and Ku-band 10.7 − 12.7 GHz. The main advantage is that the radiation is isotropic like fluorescence light, which allows the observation of the longitudinal profile from the side of the shower.

The CROME experiment (Cosmic-Ray Observation via Microwave Emission) proved i.e. by the polarization that the radio emission in the GHz regime is referred back to the same mechanisms applied in the MHz case as an effect of the time compression and the refractive index of air and thus put strong limits on the emission from MBR [50]. In addition to CROME, further experiments using accelerator beams, like Air Microwave Yield (AMY) [51] and Microwave Air Yield Beam Experiment (MAYBE) [52], contribute to improving the understanding of the emission mechanism in the GHz range.

3 The Pierre Auger Observatory

The rarity of UHECR means that any experiment built to measure them must confront the low statistics in their design. To this end the Pierre Auger Observatory was built as the largest EAS detector in the world [16]. Constructed by the Pierre Auger Collaboration, a group of scientists from more than 15 different countries, the observatory was completed in 2008 but has been taking data since 2004. Since then it has delivered results fundamental to the understanding of cosmic rays. It is located in the Pampa Amarilla near Malargüe in the Mendoza province of Argentina, not far from the Andes, at an altitude of 1400 m a.s.l. (equal to a mean atmospheric depth of 865 g/cm^2) and covers an area of about 3000 km^2, see Fig. 3.1. It is designed to combine different complementary detector techniques in hybrid measurements like the surface detector, the fluorescence detector and the radio detector, see Fig. 2.6. The Pierre Auger Observatory measures the flux, the mass composition and the arrival directions of cosmic rays [53]. In addition, the collaboration also contributed to measurements of the proton-air [54] and proton-proton cross-sections which are compared with LHC results that have been extrapolated from LHC energies to UHECR energies. Furthermore, limits to the photon [55] and neutrino flux [30] have been set and a transition of the composition from light to heavy particles for the highest energies has been measured. In this chapter, a short description of the different detector types used at the Pierre Auger Observatory including the antennas measuring in the GHz frequency regime is given.

3.1 The surface detector

The surface detector array (SD) consists of 1660 water-Cherenkov detectors each with a diameter of 3.6 m and a height of 1.2 m placed on a triangular grid at 1.5 km separation. Inside this grid is a sub array with a station distance of 750 m, called the *infill array* [56]. This smaller spacing increases the sensitivity to lower energies between 10^{17} eV and 10^{18} eV and thus the section in the energy spectrum between the second knee and the ankle. The surface detector working with a duty cycle of 100 % is fully efficient for vertical showers at energies above $3 \times 10^{18.0}$ eV.

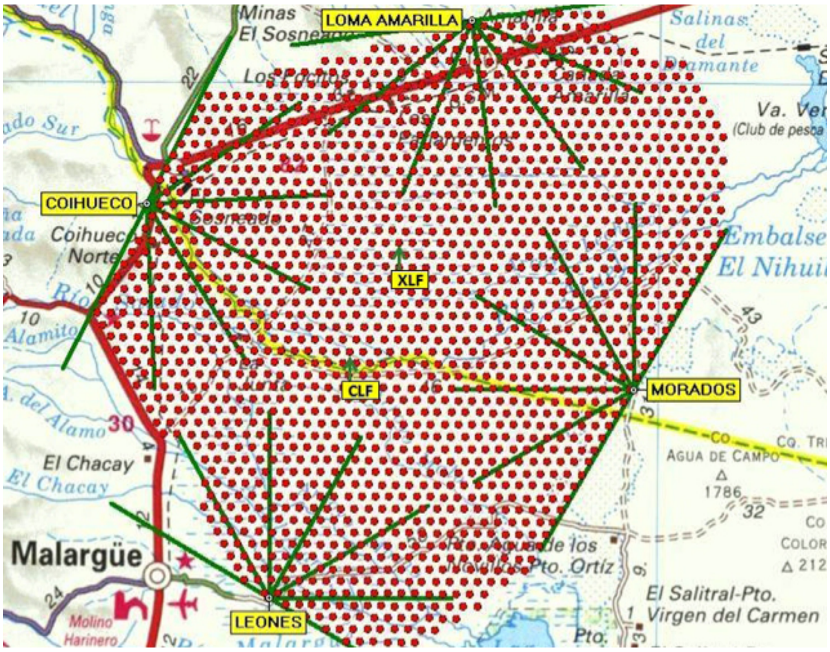


Figure 3.1: Map of the Pierre Auger Observatory near Malargüe, Argentina. Each red dot represents a surface detector station. Also shown are the 4 sites of the optical fluorescence telescopes Los Leones, Los Morados, Loma Amarilla and Coihueco. The green lines indicates the field of view of each telescope [16].

Each water-Cherenkov detector is equipped with solar panels and communication antennas to measure the particles from air shower cascades autonomously, see Fig. 3.2. The SD detects both the electromagnetic and muonic component of an EAS. Due to the refractive index of water, highly energetic particles which travel faster through the water than the light emit Cherenkov light. This is detected by three photomultiplier tubes (PMT) placed inside the 12 tons of ultra-pure water in each SD. The inner surface of each SD is covered with a reflective Tyvek layer to maximize light collection [57]. Pure water was chosen to prevent the growth of organisms which would lower water clarity. The orientation of the Earth's magnetic field was taken into account during the positioning of the PMTs to minimize interference with the station electronics. The PMT signal is digitized by 40 MHz semi-flash ADCs. The signal is converted into units called vertical equivalent muons (VEM). A VEM is the amount of light a muon produces while travelling vertically through the SD station. If an event passes the local trigger, each station will send its trigger information to the Central Data Acquisition System (CDAS). The local trigger is divided into two levels, T1 and T2 [58]. On the first level, T1, mainly two algorithms which work independently select events: a threshold trigger searching for a coincident signal in all three PMTs or a time-over-threshold trigger (ToT). In addition, a time over threshold deconvolved (ToTd) trigger and a multiplicity of positive steps (MoPS) trigger have been implemented more recently. T2 is designed to combine the first level triggers to a rate of roughly 23 Hz. For the upcoming upgrade of the surface detector, additional triggers are tested and will be installed on the station electronics for reducing the energy threshold of the whole array.

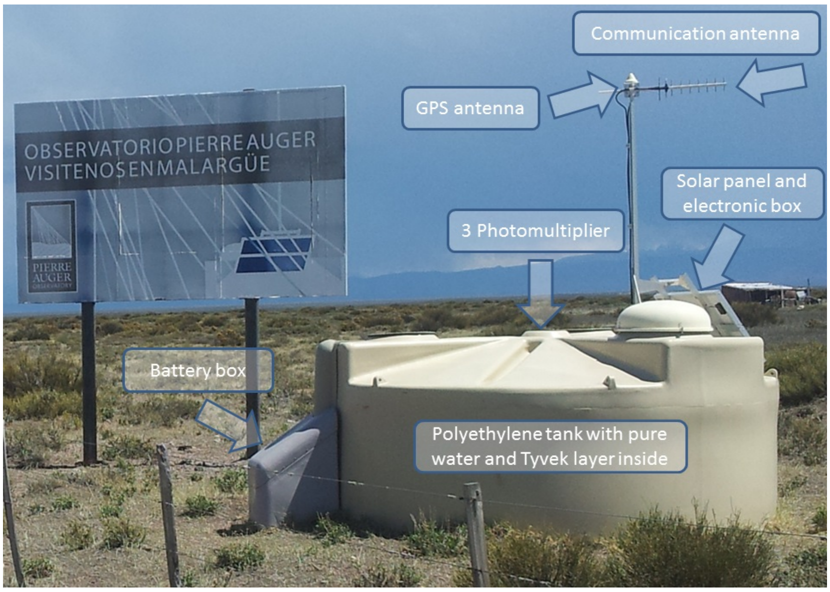


Figure 3.2: Image of a water-Cherenkov detector in the field with some of its components labeled.

3.2 The fluorescence detector

Located at four sites of the SD array, Los Leones, Los Morados, Loma Amarilla and Coihueco, the optical fluorescence telescopes (FD) detect the fluorescence light produced during an EAS, see Fig. 3.3a. This flourescene light is emitted when the Nitrogen molecules in the atmosphere which have been excited by the passing EAS de-excite and emit light in the UV spectrum isotropically. By recording the flourescence light of an EAS the FD measures the longitudinal profile, which when integrated, will yield the shower energy and also provides $X_{\max}$ and therefore information about the mass of the cosmic ray. Each site is equipped with 6 telescopes which have a field of view of $30° \times 30°$. In addition, the HEAT (High Elevation Auger Telescopes) extension is installed at the Coihueco site. HEAT consists of three additional fluorescence telescopes which can be operated in an upward or downward oriented mode and thus the elevation angle can be adjusted. By looking into the higher regions of the atmosphere, HEAT can measure the earlier development of lower energetic EAS. With HEAT, 27 FD telescopes are operated in total. If an EAS is measured at just one FD site, the event is called an FD-mono event, if it is measured in more than one site, the event is called "‘stereo"’.

After entering the Schmidt optics, the fluorescence light gets reflected by a large segmented mirror with an area of $13\,\mathrm{m}^2$ and is detected by a camera at the focal plane. The camera is equipped with 22×20 PMTs each acting as a separate pixel as is shown in Fig. 3.3b. The PMT signal is processed by a three-level trigger. The first level is a threshold trigger for each pixel. In the second level, patterns between connected pixels containing a signal have to be found. The last level is for rejecting false triggers e.g. caused by lightning and thus cleaning the data sample. After passing all levels, the event for one FD site is built locally. In total, the overall T3 trigger rate of a single telescope including the SD-hybrid trigger is 0.012 Hz.

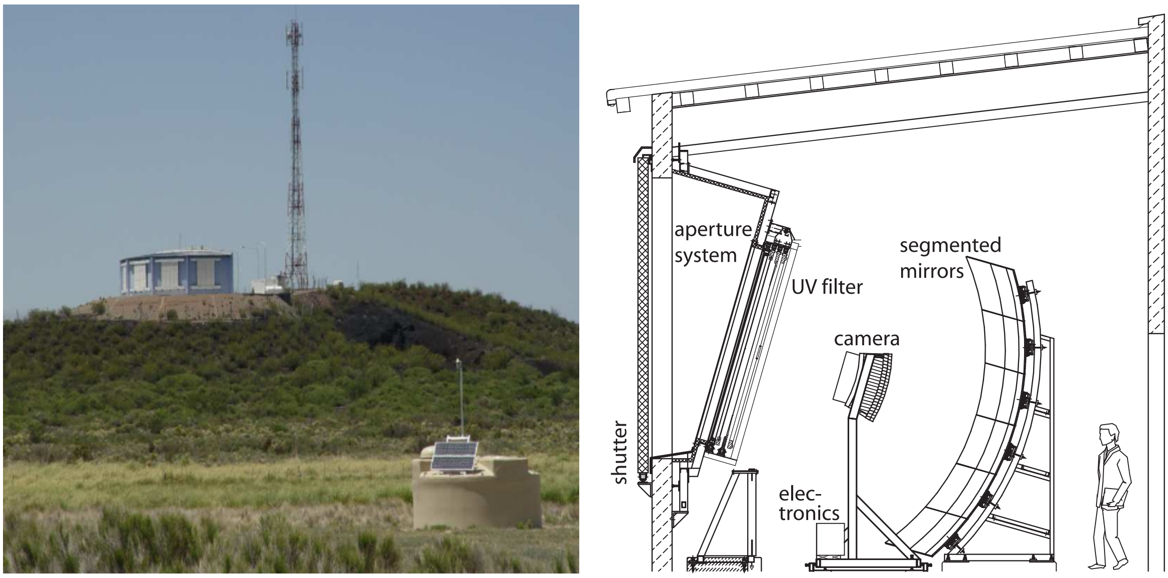


Figure 3.3: Diagram of a fluorescence telescope building in the Argentinean pampa and the co-located communication tower with one of the SD station in the foreground (a) [59] and illustration of a fluorescence telescope with its main components (b) [60].

Due to the high sensitivity of the PMT, the FD telescopes can be operated only in moonless nights with a duty cycle of about 15 %. At the moment a procedure to extend the duty cycle with reduced sensitivity is implemented. To protect the FD electronics, a shutter in front of each telescope closes automatically in case of high light intensity or bad weather conditions like storm or rain. In addition, a fail-safe curtain closes automatically if the shutter does not work properly, for example in case of power failures, acting as a two-level protection mechanism. For calibration purposes, among other methods, laser shots from the Central Laser Facility (CLF) [61] and the eXtreme Laser Facility (XLF) are used. They can be seen as tracks with known direction inside the FD telescopes and are also used to monitor atmospheric conditions like aerosol distributions [62].

The operating of the FD telescopes supervised by a FD detector shift, which is done locally in Malargüe and also via remote sites set up at major collaborating institutions [63]. SD and FD are operating stably since 2008 and provide high-quality hybrid measurements.

3.3 Central Data Acquisition System - CDAS

CDAS has been developed to combine several tasks into one system. Besides organizing the local station trigger information, it also keeps control of the data storage and includes control mechanisms and methods to monitor the system performance. It is located in an air conditioned computer center at the Observatory campus in Malargüe. To prevent losses of critical data, the data which has been acquired by CDAS is copied to the primary mirror located at the Lyon HEP Computer center.

Concerning the SD, CDAS collects the T2 trigger information, consisting of a timestamp and the trigger type, for each station every second to check if all conditions for a global third level trigger T3 are fulfilled. For those checks, a hexagon of SD stations with one SD station in the center is analyzed and stations with signals which fall within certain patterns in a well-defined coincidence window of ± 25 µs are searched for. Two different modes for the trigger criteria exist, one more permissive than the other. After passing the criteria for a T3 trigger, the request for sending the FADC data is transferred back to the water-Cherenkov station within 8 s and an event is built out of the corresponding data locally and finally sent again to CDAS with related calibration information. With this configuration, the total T3 trigger rate of the SD array is 0.1 Hz.

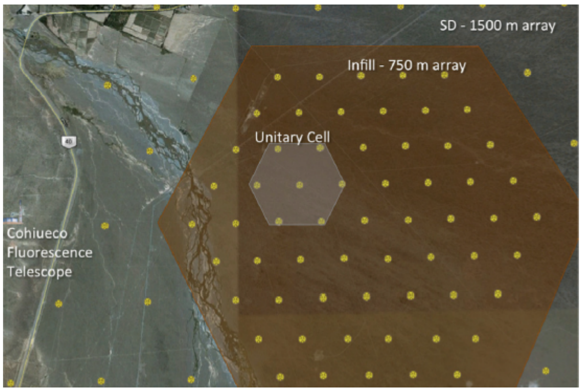
The FD data acquisition system runs independently from CDAS, however, FD T3 trigger created at the FD site are sent to CDAS. CDAS then reconstructs the air shower information to find SD stations which may have seen the signal and requests the corresponding SD FADC traces to look for coincident SD events. If found, the FD data is then merged with the SD data by CDAS on the following day to create a hybrid event. Hybrid observations require at least one triggered SD station which acts as a lever arm in the FD timing fit during analysis which reduces reconstruction uncertainties. If the event also meets the SD T3 trigger criteria, it is called a *golden hybrid event*.

3.4 Auger Muon and Infilled Ground Array - AMIGA

AMIGA is an enhancement to the Pierre Auger Observatory designed to measure the muonic component of an EAS [64]. It is integrated in the infill array and consists of segmented scintillators buried near a SD detector at different depths to provide varying shielding for the electromagnetic component of the shower, see Fig. 3.4. The Unitary Cell, which was a first prototype hexagon of AMIGA, has been fully operational since March 2015. The AMIGA scintillators are triggered by the T1 trigger of its corresponding SD station. They are connected to PMTs whos digitized signals are sent to CDAS over a dedicated communication system only when a T3 trigger occures.

3.5 GHz - AMBER, MIDAS, EASIER

At the Pierre Auger Observatory three different radio detector prototypes, with two different approaches for measuring the radio emission from air showers in the GHz range, have been installed. AMBER (Air shower Microwave Bremsstrahlung Experimental Radiometer) [65] and MIDAS (MIcrowave Detection of Air Showers) [66] use parabolic dish reflectors acting as imaging telescopes for measuring the signal similar to FD. Easier (Extensive Air Shower



(a) (b)

Figure 3.4: Layout of the AMIGA array and the Unitary Cell (a) and illustration of the Muon Counter located under the SD station together with the MD electronics (b) [64].

Identification with Electron Radiometer) consists of antennas installed on top of SD detectors to see a short, time compressed pulse with large peak power.

All of them use a similar electronics chain: First a horn antenna acts as a receiver, then a Low Noise Block down converter (LNB) amplifies the signal and converts it down to 0.95−1.75 GHz. This is followed by a band-pass filter and finally by a logarithmic power detector. This detector provides an output DC voltage proportional to the logarithm of the input power with a time response of 10 ns to digitize the envelope of the radio signal. This DC voltage is sent to FADC (Fast Analog To Digital Converter) cards.

3.5.1 AMBER

AMBER has been running since May 2011 using 12 single polarization horns in the C-band and 4 dual polarized dual band horns in the C- and Ku-band, see Fig. 3.5a. The dish has a size of 2.4 m, is elevated by 30° and has a field of view (FoV) of $14° \times 14°$. For event measurements, it is externally triggered by CDAS. A fast reconstruction algorithm provides the EAS geometry to calculate the crossing time inside the FOV to read out the corresponding data trace. To compensate for reconstruction uncertainties, the whole trace has a time length of 150 µs. The calibration of the system has been done in several steps: first the power detector was calibrated by a network analyzer, then the dish and the LNA are calibrated with the Y-factor method using RF absorber foam and a calibrated LNB. To augment this, the transit of the sun and the galactic plane through the FOV can be used. The system temperature in the C band is $45 - 65$ K and around 100 K in the KU band.

(a)

(b)

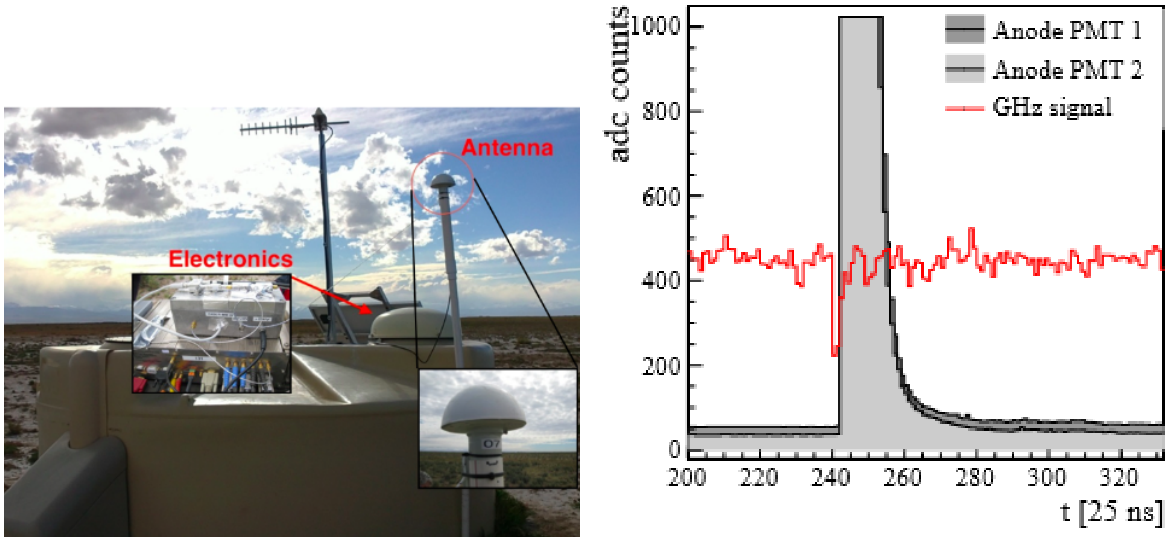
Figure 3.5: (a) The AMBER antenna with its receiver electronics placed in front [65] and (b), the antenna of the MIDAS experiment [67].

3.5.2 MIDAS

MIDAS has been taking data since the beginning of 2013 using 53 feeds in the C-band, each acting as one pixel, on a 4.5 m dish with a steerable mount, see Fig. 3.5b. The total FoV is $20^\circ \times 10^\circ$. MIDAS is self-triggered using a two level trigger. The first one is passed if the running sum in one pixel is over a given threshold. The second level requires the signal to be over a given threshold in patterns of four connected pixels. Like AMBER, for MIDAS the transit of the sun through the FoV is used as calibration source, resulting in $T_{sys} = (65 \pm 3)$ K. Since no GHz events have been found so far, MIDAS has provided limits on the power flux of the MBR. In addition, it has been ruled out that the power flux scales quadratically with the EAS energy at the power flux value measured in previous beam experiments.

3.5.3 EASIER

EASIER consists of an array of 61 feed horns with a large, 60°, FoV. Each horn is installed on an SD detector and sits 3 m above ground pointing to the zenith, see Fig. 3.6a. The polarization of 33 antennas are oriented in North-South direction and 28 antennas in East-West direction. Data readout is done coincidentally with SD. The signal is integrated by a power-log detector and sampled by the SD front-end with 40 MHz. The system operation temperature is roughly 100 K. The installation started with 7 antennas in April 2011 followed by 54 more antennas in April 2012. The first event was detected in June 2011 with an energy of 13.2 EeV, see Fig. 3.6b. To date, 3 events have been recorded, each with a pulse length smaller than 75 ns and all with East-West polarization. The results contribute to the understanding of the GHz radio emission process since the polarization favors a geomagnetic emission mechanism instead of the unpolarized MBR.



(a) (b)

Figure 3.6: (a), an example of an SD station equipped with an EASIER antenna. (b), the trace of the first detected event in the microwave region of an EAS by EASIER with two of the corresponding SD signal traces [67].

3.6 Data processing - Offline and APE

The Offline framework has been developed as a tool for simplifying data analysis for a segmented detector system with high complexity like the Pierre Auger Observatory [68]. Originally designed only for the SD and FD, it has now also been adapted for Radio and AMIGA. Offline provides an infrastructure, in which all detector components, the measured event data, monitoring values, configuration instructions and analysis algorithms are combined for reconstruction of an EAS or for simulations. Shower reconstructions are done by processing the event data using a sequence of modules containing the physics and analysis algorithms whose operation can be customized through separately specified configuration parameters. Both the sequence and the configuration can be easily changed in XML files. The infrastructure includes two main components, the *event data model* and the *detector description*, as shown in Fig. 3.7. The event data structure includes the raw event data as well as the reconstructed or simulated information of the air shower. Since modules are not allowed to communicate with each other directly, the communication is done via the event data structure.

The detector description is a readonly interface for requesting information about the detector itself, e.g. its geometry, the calibration and the atmospheric conditions for the time of the data acquisition as stored in databases. The information transfer is done via so-called *managers*. To ensure future analysis reproducibility, all reconstruction configurations, used external libraries and module version information are logged into an XML log file. The final reconstruction results can be written out in different file formats.

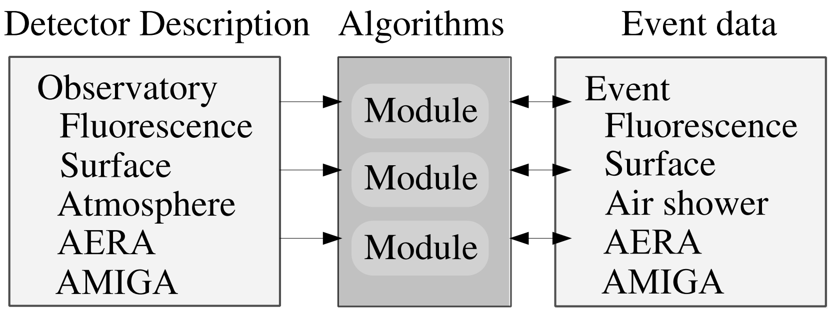


Figure 3.7: Structure of the Offline framework divided in three components. The detector description contains all information about the observatory. The event data contains all measured values by the different detectors types and in the modules, algorithms for air shower reconstruction are applied, modified from [68].

Changes to the Offline framework are committed to a corresponding SVN repository. An underlying BuildBot system automatically compiles and checks the code for errors and runs unit and acceptance tests. In order to provide a complete installation environment for Offline with all its dependent external libraries easily, APE (Auger Package and Environment) has been created [69]. This tool based on python automatically downloads and builds all external dependencies and generates the environment variables. Offline can also be installed directly via APE or installed independently from the svn repository afterwards.

4 The Auger Engineering Radio Array

The Pierre Auger Observatory offers excellent opportunities to study the radio emission of extensive air showers by comparing radio observations to different detector techniques through hybrid measurements. The radio stations of the AERA experiment at the Pierre Auger Observatory measure the electric field strength produced by cosmic rays within the energy range of $10^{16.5}$ eV to 10^{19} eV and the frequency range of 30 to 80 MHz (VHF band) with a 100 % duty cycle. Thus, AERA is well positioned to measure relevant parameters like the mass composition and the energy spectrum of cosmic rays. It also provides major contributions to the understanding of the emission mechanisms. More than 7500 events have been measured in coincidence with the surface detector so far. This chapter provides an overview of the AERA detector, including its layout, instrumentation, calibration, triggers and the structure of the data it records.

4.1 Setup

AERA is located inside the SD-Infill and within the field of view of FD telescopes at the Coihueco site with the HEAT extension, see Fig. 4.1. The deployment of AERA has been realized in three phases.

4.1.1 AERA24

The first phase, *AERA24*, consists of 24 autonomous radio detector stations (RDS). AERA24 was deployed in September 2010, reached its operational phase in March of 2011 and has been stably taking data ever since. The radio detector consists of 24 dual-polarized log-periodic dipole antennas (LPDA) placed in a 144 m triangular grid [70]. A picture of the LPDA is shown in Fig. 4.2. By using several half wave dipoles with different length, the sensitivity over a wide frequency range is achieved. Two different antenna planes arranged perpendicular to each other and adjusted in the magnetic north direction with a compass mounted directly on

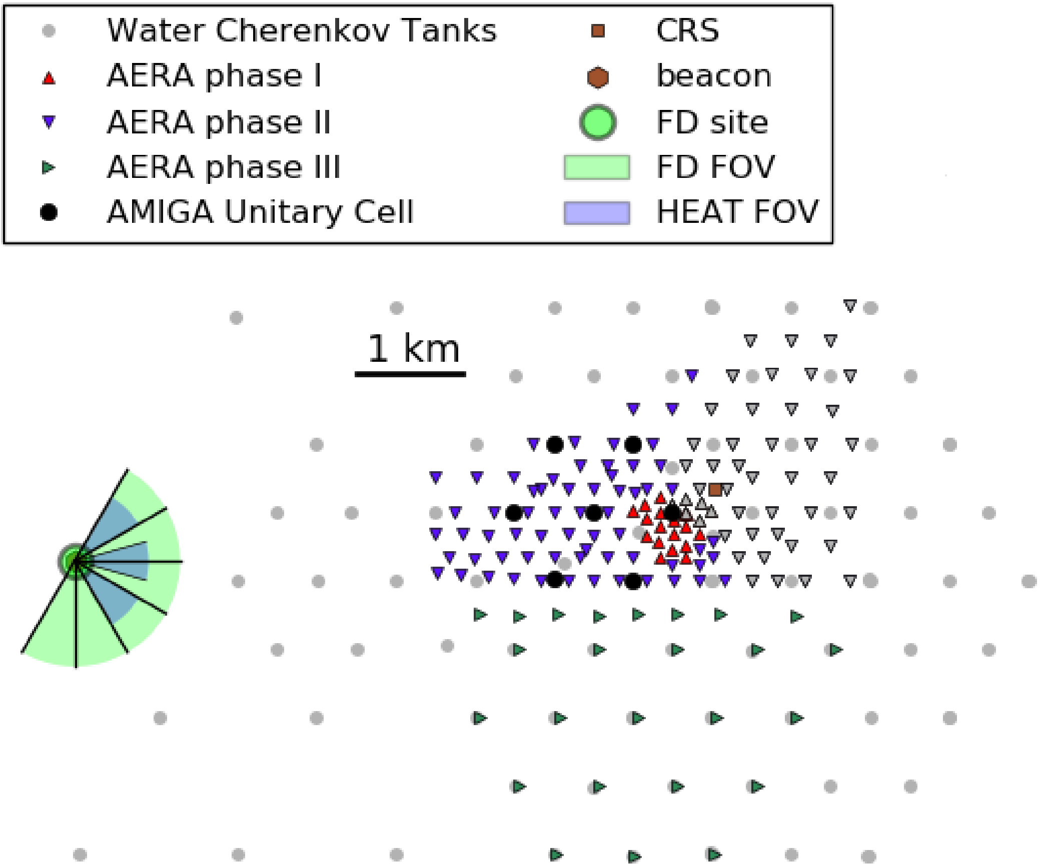


Figure 4.1: Schematic map of the AERA array. The different phases of the AERA deployments are indicated along with the co-located FD site Coihueco, the position of the beacon (see section 4.2), the Central Radio Station (CRS) and the AMIGA Unitary Cell. In addition, the positions of the surface detector stations are shown.

the antenna [71] offer the dual polarization measurements necessary to distinguish the different radio emission mechanisms. The base is located at the top of the antenna at a height of $\sim$ 4.5 m. Measured signals are transferred to a low-noise amplifier (LNA) located at the bottom of the antenna via an impedance matched coaxial cable. Integrated into the LNA are filter elements for limiting and cleaning of the frequency range. Next, the signal is transfered to a weather-proof, noise shielded box, containing the digitization electronics where it is further amplified and filtered. Solar panels are mounted in front of the box and connected to batteries via a charge controller located inside the box which allocates the correct amount of power, allowing for the autonomous operation of the radio stations throughout the year.

There are two different types of digitizing systems, the KIT/BUW (Karlsruhe Institute of Technology / Bergische Universität Wuppertal) digitizers and the NIKHEF (National Institute for Nuclear Physics and High Energy Physics in Amsterdam) digitizers. Radio stations, which contain the first type will be referred as *German stations* and stations which contain the latter will be referred as *Dutch stations* in this thesis. The different electronics have been developed to realize different trigger strategies; the 18 German stations focus on external triggering by

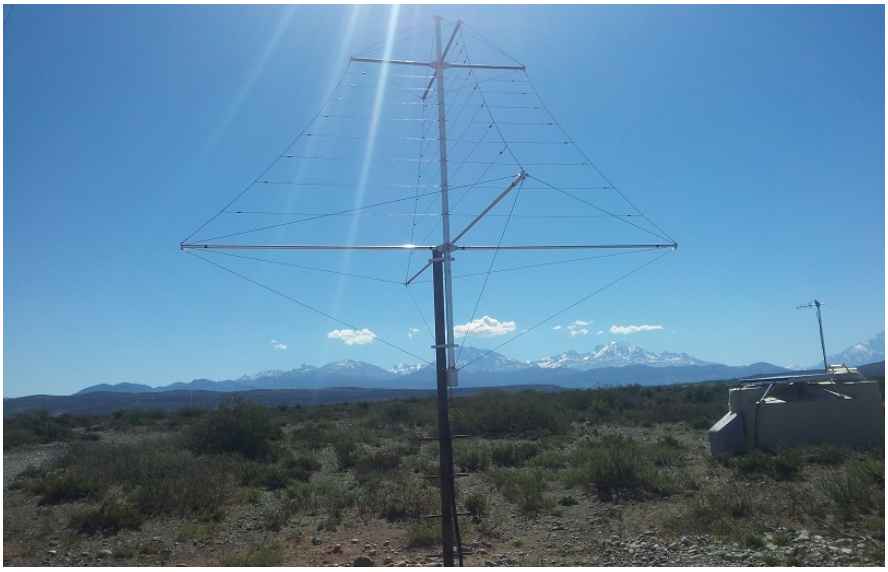


Figure 4.2: Picture of the AERA log-periodic dipole antenna next to a surface detector station. The solar panels and the casing for the electronics, located in front of the antenna and below the frame, are not visible.

the other detector components of the Pierre Auger Observatory, while the 6 Dutch stations concentrate on self-triggering. The filtered and amplified signal is digitized by a 12 bit flash analog-to-digital converter with a sampling rate of 180 MHz in the KIT/BUW digitizers and 200 MHz in the NIKHEF digitizers. In both setups the signal is then analyzed by a field programmable gate array (FPGA) able to perform a first trigger decision on ADC level. Dutch stations have 4 ADC channels, 2 for low and 2 for high gain while in German stations only the 2 high gain channels are read out. German digitizers contain a ring buffer which is able to store data for up to 7 s. The stations are connected via fibers with the Central Radio Station (CRS) located next to the AERA24 array. The data acquisition system (DAQ) was located inside the CRS. The CRS is connected via a wireless link to the Coihueco FD-site.

4.1.2 AERA124

Completed in May 2013, the second phase, *AERA124*, upgraded the Radio array from 24 to 124 stations and is spread over 6 km^2 with a station distance of 250 m and 375 m to increase statistics. AERA124 features a new dual polarized butterfly antenna and uses a more compact station design, see Fig. 4.3. In total, 60 German stations and 40 Dutch stations were added. The butterfly antenna is more sensitive over the zenith range, especially towards the horizon, and uses reflections from the ground to enhance the signal [70]. As the reflection and thus the antenna response depends on the ground permittivity and the conductivity of the soil, the ground conditions are measured continuously [73]. The measurement is done with a metal plate connected with a metal pin dug in the surface and takes place at one of the two AERA weather stations located in the AERA24 array [74], shown in Fig. 4.4. The butterfly antenna is equipped with an underlying support structure build out of fiberglass to protect the physical

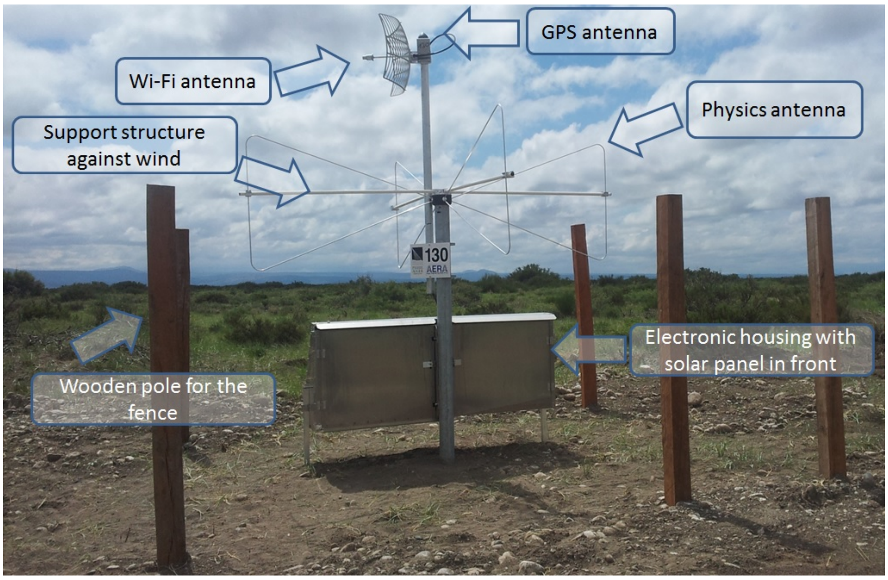


Figure 4.3: Picture of the AERA butterfly antenna used for AERA24 and AERA124 after mounting. The main components; the GPS antenna and the Wi-Fi antenna, the physical antenna with its underlying support structure protecting against strong winds and the housing for the electronics are visible. The solar panel is mounted in front of the housing. To protect the RDS from animals, a fence surrounds the whole construction (not yet installed in the picture).

antenna against the environmental conditions in the field, e.g. wind speeds up to $160\,\mathrm{km\,h^{-1}}$. The fiberglass is sprayed with a substance protect against degradation due to solar radiation. The antennas have also been aligned to the Earth's magnetic field so that one arm is pointing in the magnetic North-South direction and the other parallel to the East-West direction.

The general layout for the electronical components of the RDS, with substantial improvements, is the same as for the AERA24 array. A reduction in the power consumption of the RDS and the development of a new charge controller translated to smaller solar panels and batteries which are now mounted directly to the antenna pole. Also, the 12 bit FADC of the Dutch stations has been replaced by a 14 bit version. Additionally, the Dutch stations have been equipped with plastic scintillators located directly at the RDS replacing both low-gain channels to allow external triggering [72]. The data communication to the CRS via fibers has been replaced by Wi-Fi antennas directly mounted on top of each RDS. Stations located on the West of the CRS send their data directly to Coihueco. Stations on the East part of the CRS, which includes all Dutch stations, send their data to the CRS [75]. In addition to these enhancements, due to continuing R&D, a low frequency antenna station sensitive in the low MHz range and three 3D-dipole antenna stations, able to measure the vertical component and thus the electric field vector in all three dimensions, have been developed and deployed. Lastly, five of the Dutch stations have been equipped with 3D whisk antennas. With the deployment of *AERA124* the AERA-DAQ was moved to servers located in a computer rack at the FD-building at Coihueco, where most of the RDS communicate to.

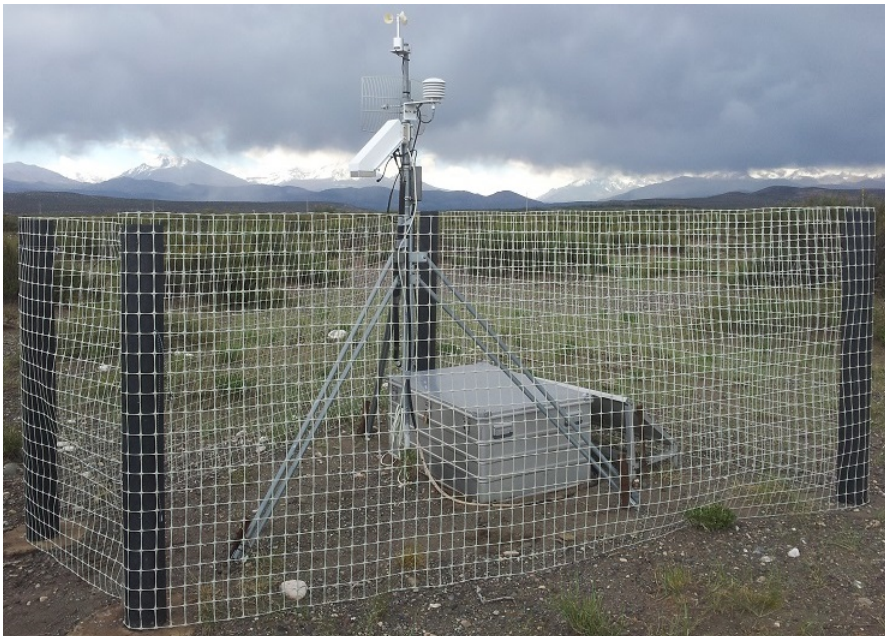


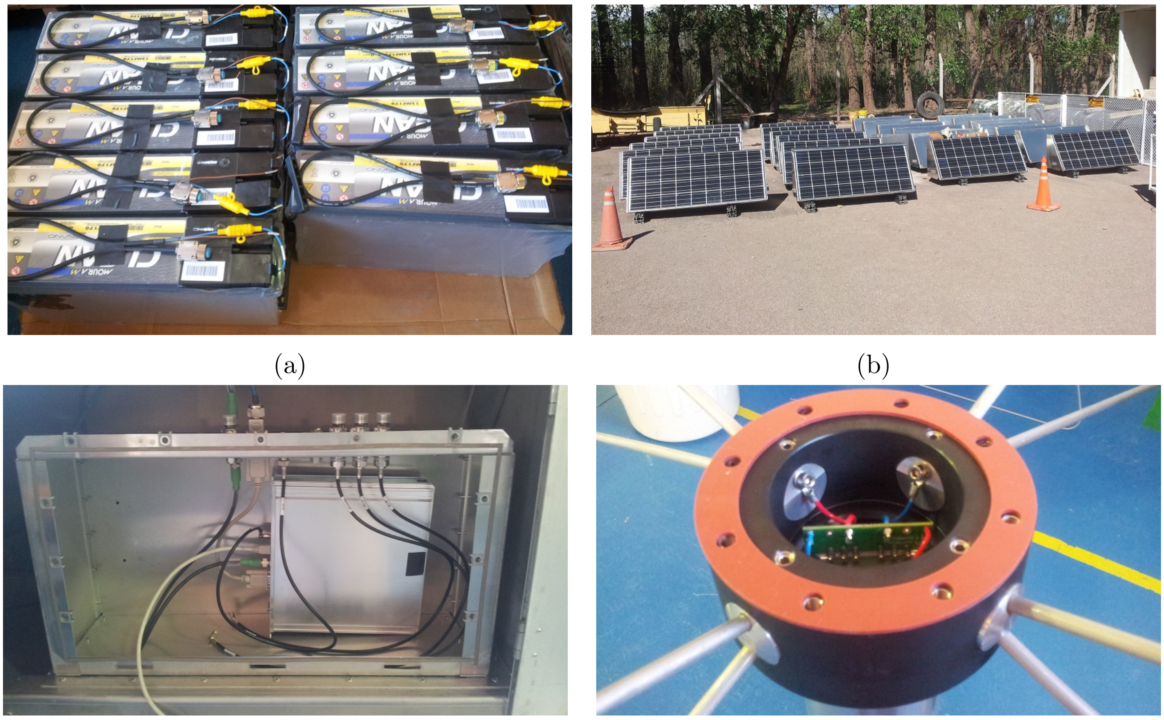
Figure 4.4: Photograph of the AERA weather station with a wind speed sensor on top, a humidity sensor and a sensor for measuring the electric field below. On the ground, a mechanical box is connected with a solar panel containing the battery and the electronics. The ground conditions are measured with a metal plane next to the box. Data is transferred via a Wi-Fi antenna directly to Coihueco.

4.1.3 AERA153

The third stage, *AERA153* was completed in March 2015 with butterfly antennas placed at distances of 375 m and 750 m to extend the measurements to higher energies. With a larger spacing, detecting the radio emission induced by very inclined EAS, which can have a very large footprint, is improved [76]. The additional 25 stations with KIT/BUW digitizers have been deployed South of the AERA124 array, sending their data directly to Coihueco. The station design, besides some modifications in the digitizing electronics, has not changed from AERA124. In Fig. 4.5, the main components of a RDS for AERA153 during the deployment are shown.

4.2 Calibration

In order to perform studies, e.g. the measurement of the energy spectrum of cosmic rays, stringent requirements for the RDS timing accuracy of $\sim 1\,\mathrm{ns}$ are required. Furthermore, the precise antenna signal response is needed. To fulfill these demands, different calibration techniques for AERA are used. Although each station is equipped with GPS clocks, the synchronization cannot be more precise than several ns and is not constant over time. This is improved by using a reference beacon antenna installed at the FD building in Coihueco

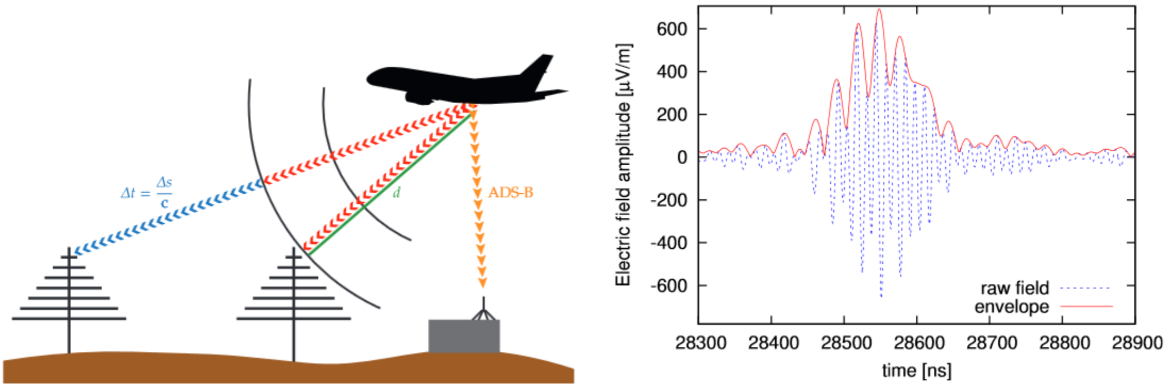


(a) (b) (c) (d)

Figure 4.5: Pictures of the main components of a RDS for AERA153 taken during the deployment. Due to improvements in the electronics, smaller batteries (a) and smaller solar panels (b) can be used. The solar panels are mounted on a shielded box containing the digitizing electronics (c). The LNA with two input channels for each polarization measurement is located in the middle of the butterfly antenna (d).

[77]. It is a dipole antenna which permanently emits four sine waves at constant frequencies detectable by the RDS. The timing offset between the stations is visible as a phase difference. Later in the data analysis, the beacon frequencies can be filtered in the frequency domain easily.

Especially in the context of time synchronization, the exact position of each RDS must be known with good accuracy. The positions are measured in campaigns using a benchmark, a geographical elevation reference point a few km from Coihueco, and a D-GPRS system. Additionally, the time synchronization can be improved by detecting the radio-signals of commercial airplanes with the RDS [78]. Combining both methods yields a timing accuracy of 2 ns. In Fig. 4.6, the principle of the timing offset determination with the aircraft detection method is shown. As a further calibration technique, in-flight campaigns with an octocopter are performed [80]. While flying over the detecting RDS, the octocopter is equipped with a GPS system and a calibrated reference antenna. With knowledge of the exact position of the octocopter as well as the properties of the reference signal, the timing offset between stations or the signal response of the antenna to be calibrated can be measured. By covering the complete zenith and azimuth range around the antenna with different frequencies, the fully



(a) (b)

Figure 4.6: Principle of the AERA timing calibration with airplanes. The position of the airplane is transmitted via the ADS-B service and can be detected with commercial electronics. In addition signals in the AERA frequency range are detected by the RDS and the offset can be calculated out of the known source position (a). Measured airplane radio pulse in units of the reconstructed electric field with a Hilbert envelope (b). Figure taken from [78].

3D direction and frequency dependent antenna sensitivity needed for later analysis can be obtained. Those campaigns have also been performed with balloon borne flights. The results of the calibration campaigns are compared to and validated with simulations using the Numerical Electromagnetics Code (NEC2++) which include the complete gain pattern of the antenna [81]. Finally, the transition of the galactic plane, with its center acting as a strong radio source, can be detected with the AERA stations and used for calibration. The radio emission from the galactic plane is indeed strong enough to act as the dominant contribution to the noise background for AERA, excluding anthropogenic sources like power lines [82].

4.3 AERA data acquisition and trigger modes

The DAQ of AERA is realized with different trigger modes which will be explained in the subsections below. As already mentioned, the whole AERA array is composed of 2 sub-arrays of stations with different digitizers and central DAQ. They also differ in the available trigger modes.

4.3.1 Self-trigger

Similar to the SD data acquisition, several trigger levels exist for AERA [83]. A T1 trigger is formed by the FPGA at each RDS station if a bandwidth-limited pulse in the voltage trace over a certain threshold is detected. Both digitizer versions differ in how they select a pulse in this level. Since broad- and narrow-band transmitters at the AERA site contribute to the noise background, additional trigger algorithms removing those undesired events are applied inside each station. Triggers which survive this additional step are called T2 triggers. They are marked with a GPS timestamp and are sent to the central DAQ. Inside the DAQ, time and spatial coincidence criteria between the RDS as well as additional algorithms to remove background events, are used to build the T3 trigger which consists of at least 3 stations. If the T3 is formed, a data request is sent to the stations and the data is then stored. Inside the DAQ, the process which computes the T3 is called *T3Maker*. Included is a first computation of the arrival direction of the radio wavefront. Afterwards, the *cone algorithm* is applied. The cone algorithm rejects events based on the zenith angle and their occurrence in time at a certain position on a sphere [84], thus removing stationary events (especially events near the horizon). If a trigger passes all the criteria, the requested data taken by the RDS is sent to the AERA event builder *AEVB* to create the event.

4.3.2 External trigger

AERA can also be externally triggered by the other Auger detectors **SD**, **FD**, and **HEAT**. Thus, the probability of detecting a real EAS event with radio antennas is higher than just relying on a self-trigger. Nevertheless, externally triggered events do not represent a pure sample of cosmic ray events, since the radio antennas could still detect a noise event occurring in the coincidence window by chance. In this context, higher level criteria for the event selection have to be applied in the Offline data analysis afterwards. The external trigger to AERA is built in CDAS and then processed in the T3maker. It can take up to 7 s until the information arrives at the AERA stations. Therefore, the external trigger can just be used with the German stations due to the ring buffer in which taken data can be stored so long. To allow sub-threshold signals to be used in the reconstruction, i.e., using the interference technique, all stations are read out and the data is stored.

A special type of external trigger is the **AERAlet** trigger which is sent if an event is measured by the AERAlet array. This is an even denser array of SD stations with 433 m spacing located inside the Infill array co-located with the AERA24 stations [79]. It is designed to lower the energy threshold of SD and to improve the estimation of the core position. Its operation began in January 2013 and consists of one hexagon with the six SD stations *Lety Jr.*, *Pipi Jr.*, *Chichino Jr.*, *INF6A*, *INF4*, *INF5* and the seventh station *Kathy Turner* in the middle, which is acting as the central station of the hexagon.

At first, just the SD trigger was implemented in the DAQ as external trigger. Since December 2013, also FD, HEAT and AERAlet are used.

4.3.3 Periodic and pass-through trigger

In order to monitor the noise background, all RDS stations are read-out **periodically**. To reduce the amount of data by a factor of 10, the read-out increased to every 100 s from every 10 s before since 2014-06-26. Noise information is extremely useful for simulation and

Table 4.1: General structure of radio binary data.

File header information
First Event header
First Station header First Station additional information ADC values of Ch. 1, 2 ...
... All other stations ...
... All other events ...

calibration purposes (e.g., the galactic plane calibration). In addition, the **pass through** trigger is used to evaluate how efficient the triggering works by passing the data directly to the DAQ without any filtering. For the German stations, this is done for every 999th self-triggered event and for the Dutch stations for every 40000th event.

4.3.4 Airplane and scintillator trigger

A special type of trigger in the German DAQ is the **airplane trigger**. As already mentioned in section 4.2, airplanes can be used for the timing calibration of the RDS. Since they are also identified as a noise source in the trigger algorithms, special treatment is needed. If airplanes should be detected by the stations, the cone algorithm is stopped for a small region in the sky around the position of the airplane estimated with help of the ADS-B service. Since some of the Dutch stations are equipped with scintillators, a special **scintillator** trigger is applied. If one of the scintillator channels has seen a signal, the read-out of the radio voltage trace of this station is triggered. In addition, the trigger information is sent to the DAQ to search for coincidences with other stations.

4.3.5 AEVB and others trigger modes

There are few more trigger modes in AERA which will be only briefly described. For calibration purposes, a **calibration** trigger can be sent to the stations. The **GUI** trigger can be sent out of the graphical user interface of the DAQ monitoring. The **AEVB** trigger is used if a station has not seen a signal, but a T3 is requested.

4.4 Radio data in AERA

The data taken by the AERA stations are written into a run based binary raw file by the DAQ without being interpreted. The RDS raw data is stored into this binary file in a general, hierarchical structure which is ordered as follows: file header level, event level, station level and the ADC values, see table 4.1. The file header contains a summary of information for the whole file, like run number and event trigger time of the first triggered local station. It is followed by a sequentially written list of events. Each event consists of an event header followed by an event body which is filled by a list of all station structures which contributed to the event. The station structure is built up by a station header in analogy to the event header, followed by additional information dependent on station hardware, the *station header additional information*, which contains data like the surrounding temperature or specific noise threshold values for each channel, and finally the ADC traces. The number of ADC values depends on the station digitizer type used and the event trigger. The structure for the binary data is the same for both German and Dutch stations, except for the station header additional information. The layout of the additional information for stations with NIKHEF digitizers changed during an update in December 2012. The old design is referred as DutchV1 and the new design as DutchV2 in this thesis. A detailed overview of all values stored in the binary file independent of the digitizer version is shown in table 4.2. A detailed overview of all values stored in the station header additional information for each digitizer version is shown in appendix A.1.

The third value of the station header, bytes 89 - 90, contains three different variables. The first 8 bits contain the ID of the local station itself. Bits 9 - 11 contain the LS hardware type and bits 12 - 16 contain the LS hardware version. The last two variables are achieved by using bit operators. First right shifting by 8 bits and comparing with the logical and 0x7 yields the hardware type and right shifting by 11 bits and comparing with 0x1f yields the hardware version.

The transition of the DAQ to Coihueco with the deployment of *AERA124* was delayed for the Dutch stations. Until December 2015 these continued to use a parallel DAQ in the CRS.

Table 4.2: Detailed overview over all digitizer independent values stored in the binary data structure.

Level	Byte	Description
File header	1 - 4	Header length in bytes
	5 - 8	Run number
	9 - 12	Run mode (e.g. calibration)
	13 - 16	File ID
	17 - 20	Event number of first event in file
	21 - 24	Event second of first event in file
	25 - 28	Event number of last event in file
	29 - 32	Event second of last event in file
	33 - 36	Additional header information 1
	37 - 40	Additional header information 2
Event header	41 - 44	Event length in bytes, including all stations and ADC values
	45 - 48	Run number
	49 - 52	Event number
	53 - 56	Event T3 number
	57 - 60	ID of first (timing) local station in event
	61 - 64	Event second
	65 - 68	Event nanosecond
	69 - 70	Event type
	71 - 72	Event version
	73 - 76	Event additional information 1
	77 - 80	Event additional information 2
	81 - 84	Number of stations in event
Station header	85 - 86	Station length in bytes
	87 - 88	Event number
	89 - 90	Station ID, hardware type and hardware version
	91 - 92	Station header length in bytes
	93 - 96	Station second
	97 - 100	Station nanosecond
	101 - 102	Station trigger flag
	103 - 104	Station trigger position
	105 - 106	Sampling frequency
	107 - 108	Channel mask
	109 - 110	ADC resolution
	111 - 112	Trace length
	113 - 114	Event version

5 Developing a new input/output library for AERA

A lot of effort has been put into improving the analysis and reconstruction chain of the measured radio data. Via the new *RdObserver*, a specific Offline module sequence, a standard reconstruction chain for radio events has been established [85]. This requires a combined data analysis with Auger detector components. For the integration of the AERA data into the Auger data, a new, flexible and ROOT based data format had to be developed. In this chapter, the new input and output library, AERAROOTIO, is presented. Using AERAROOTIO, more than 7500 radio events can be reconstructed for the time period between mid of 2012 and March 2016.

In section 5.2 a detailed description about the library itself and its advantages is given. Section 5.3 covers AERAConverter, a tool included in the library which convert the binary data to ROOT files. Its usage and ability to create different data streams based on time and trigger source are also included. The focus of section 5.6 is on the low level merging of radio into the Auger data using CDAS to create a combined dataset, as well as on the radio-radio data merging. Section 5.7 includes a description of how radio data converted by the AERAROOTIO library is handled in Offline, followed by useful applications of AERAROOTIO. The AERA-ROOTIOLib is the basis of many important recent analyses and has contributed much to the integration of radio as an officially used part of Offline. Unless otherwise stated, explanations about the data and the DAQ refer to the German DAQ before the implementation of all stations into the combined dataset in December 2015. [1]

5.1 Introduction

The AERA-DAQ combines the data collected from the radio stations in run-based binary files. Each binary file, with naming adXXXXXX.fNNNN, contains a number of recorded events. The Xs represented the run number and the Ns the run specific file number. Accessing the data was quite cumbersome, although possible with the *aevread*-library, an interface in Offline by

[1]Parts of this chapter have been internally documented in [86].

the module *EventFileReader* with the Offline file type *RadioAERA*, however, the access was limited to only input functionality. The radio data could then be read-in and processed with Offline module sequences.

When a radio event was reconstructed successfully by Offline, two output formats could then be chosen: The *ADST* format and, optionally, the Offline file format. The latter contains the whole event class structure of Offline and is large compared to the raw binary data. In the ADST format, just the reconstructed event quantities are saved, which can then be used for high-level analysis, but not for re-reconstruction based on a different module sequence. Therefore, if different analysis parameters needed to be tested, the whole process of reading in and analyzing the complete dataset had to be repeated. It was not possible to read in just events which had been reconstructed, apply a different module sequence and compare the output of the reconstruction without using the huge original data sample. Thus, in terms of processing time, data volume, easy data access and to maximize the potential of the radio data reconstruction, it was clear that a new way of handling data was urgently needed. Besides using a more *raw-like* data format, the AERAROOTIOLib has been developed to act as a new multi-functional interface for a complete IO to ROOT [1] format translation.

5.2 AERAROOTIO library

The new package combining data handling and functionality is called AERAROOTIOLib. The AERAROOTIOLib is a fully ROOT based input/output library written in C++ [87]. Both its advantages and layout are explained in the subsections below.

5.2.1 Advantages

The AERAROOTIOlib is the first common interface to the radio binary data with output functionality. It enables the direct read-in of the raw binary files of all the different RDS types independent of *aevread*. There are several advantages of the ROOT based file format, such as enabling sequential or also direct access to one specific event (n) without iterating through all (n-1) events as in the binary file before. Furthermore, the data is automatically compressed by ROOT by a factor of roughly 20 % in comparison to the binary format. The reduction of the file size is important regarding the limited transport bandwidth from Argentina to Europe which should not exceed 2 GB per day.

The ROOT based file format implementation also provides an increased processing speed of input files in Offline by a factor of 5 as compared to the binary format. This is due to aevread reading through the whole binary file each time a file is loaded. Using the AERAROOTIOLib and the AERAConverter, see section 5.3, the file has to be converted once from binary to ROOT, which takes roughly 1 minute per binary file dependent on the disc write speed. Table 5.1 shows the comparison between the read in of the binary file ad100601.f0003 and the converted ROOT file in Offline using the *EventFileReaderOG* module on the dedicated AERA server in Coihueco. Furthermore, the ROOT based file format provides a fast overview

Table 5.1: Comparison of size and processing time between a binary and a ROOT file. The ROOT format is 20 % smaller and a factor of 5 faster.

	Binary file	**ROOT file**
Size [MB]	465	373
Processing time in Offline [s]	207	41

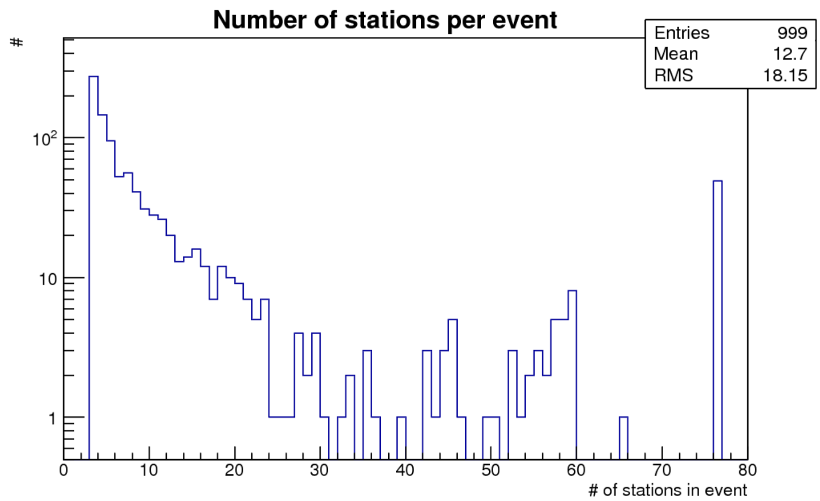


Figure 5.1: Histogram of the number of stations participating in all events in one file. The big peak at the end is caused by all externally triggered events since in this case all stations are read out by the DAQ. This plot can immediately be generated by opening the ROOT file in a TBrowser.

of information. Events are stored in a ROOT-*TTree* structure. By simply opening the ROOT file in the ROOT-*TBrowser* it is possible to view a summary of variables for all events in this file by just clicking on the corresponding branch.

Figure 5.1 shows a histogram of the number of RDS participating in all events of one binary file. As already mentioned, the AERA array is built up of stations based on different hardware types. All different types of stations are supported by the AERAROOTIOLib and the AERAConverter for all different software versions (e.g. including the interpretation of the new additional information in the station header of NIKHEF stations with software version 7 or the additional trigger flags in the German DAQ). The AERAROOTIOLib and the AERAConverter are the first and only possibility to analyze the data of the new NIKHEF stations in Offline since aevread is not able to handle this data type. The library is also the only way to handle the new combined DAQ dataset.

5.2.2 Layout

The AERAROOTIOLib uses basic fundamental data types and STL containers without dependencies on *Boost* [88] and is compatible with older C++ standards and compilers. The

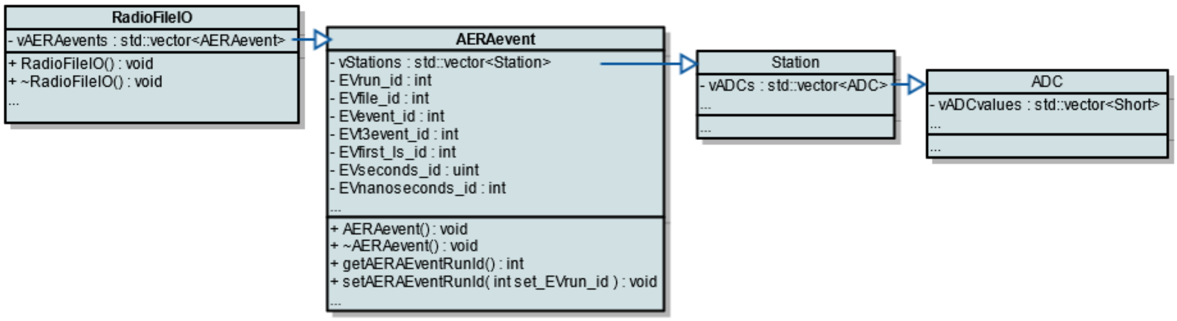


Figure 5.2: Sketch of the AERAROOTIOLib class structure. Shown are the four classes RadioFileIO, AERAevent, aerarootiostation::Station and aerarootioadc::ADC with some of the member variables.

library follows the rules of encapsulation (i.e. the access to member variables is realized with *setter* and *getter* functions). The usage of STL containers enables a flexible data structure, dynamical memory allocation and memory management. The system memory capacity and the access speed of the data storage is the only limiting factor in processing the data. The AERAROOTIOLib was able to handle the data of AERA153 out of the box without any changes directly after deployment, since the knowledge about the maximum number of stations participating per event is not needed during compilation time. Thus, also for further upgrades, the library is future-proof if the data structure of new stations does not change. But even in this case, changes are easily applicable.

The AERAROOTIOLib is under version control, see [89] for the SVN repository address, to enable backtracking and an easier way to debug of existing converted ROOT files. The SVN revision number of the trunk is directly inserted into the ROOT file. The AERAROOTIOLib is also distributed via Ape. The reference documentation using doxygen is implemented and can be activated in cmake with the "‘-DDOCUMENTATION_ENABLED = on"’ flag. The current version of the documentation is publicly available, see [90]. An example representation for the information of the class structures in the doxygen documentation is shown in appendix A.2. A description how to install and use the library in Offline, CDAS and especially the functionalities which are available in the AERAConverter, which is described in section 5.3, can be found in a Readme file located in the library's doc folder. Both the library and the integrated AERAConverter tool support Linux and Mac OS operating systems.

5.2.3 Class and data infrastructure

Data stored in the AERAROOTIO format is based on ROOT-*TTrees*. Each ROOT file consists of two ROOT-TTrees, the *AERAIOTree* and the *AERAIOTreeEventsInfo*. The first TTree, the *AERAIOTree*, represents the hierarchical structure of the AERAROOTIOLib, see Fig. 5.2. The highest level is based on an instance of the highest class in the AERAROOTIOLib called *RadioFileIO*. This class includes a vector of instances of the underlying class *AERAevent*, which for its part includes a vector of instances of *aerarootiostation::Station* and so on. The lowest stage is the *aerarootioadc::ADC* class which includes all the ADC traces of each station.

With this concept, adding a new event, station or ADC channel can simply be done by a single push back call to the particular std::vector with a new instance of the corresponding class as argument. As an example, adding a new event can be simply done by the syntax: `RadioFileIO1.setAeraRadioFileEvent()`. This is just executing the line in the RadioFileIO class: `vAERAevents.pushback(AERAevent())`.

Since the flexibility of the data structure has been one of the design goals of the AERAROOTIOLib, new fields can be added to the existing library class structure without losing backward compatibility with ROOT files containing older class structures. This can be achieved by increasing the ROOT internal *ClassDef* value in each class which have been changed. Variables which don't exist in older versions are initialized to 0 and returned as this dummy value if getter functions try to access them. When new trigger flags have been applied in the DAQ in December 2013, this backward functionality became very useful. Those new flags could be implemented in the existing AERAROOTIO data structures without the need of reconverting all old already converted data.

The second TTree, the *AERAIOTreeEventsInfo* TTree, stores the run ID and the event number of each event. Before filling a new event inside a TTree, the second TTree is used to check for the unique event identifiers to prevent placing dublicate events into the same ROOT file. Although the run ID and the event number are already contained in the *AERAIOTree*, less resources are used if just the second TTree is checked and without mapping the complete event into the memory. Information about custom classes need to be transported to ROOT. This is realized with the integrated AERAROOTIOLibLinkDef.h inside the AERAROOTIOLib. Thereby, a ROOT class dictionary is created during the library installation with CMAKE which includes all the important information about the class members for the internal ROOT streamer.

5.3 AERAConverter

Integrated into the AERAROOTIO package is the stand-alone AERAConverter tool. It is a small application which takes over the fast conversion from binary files to ROOT files via the AERAROOTIOLib. With the AERAConverter, the conversion of huge amounts of data can be done automatically. A filename of a single binary file or alternatively an ASCII document which contains a list of binary files can be given as an input argument. Great effort has been spent in the design of the AERAConverter to guarantee that several data conversion tasks can be run in parallel. In the following subchapters, its functionality is explained in detail.

5.3.1 Time and file based conversion

One of the main features of the AERAConverter is the ability to produce output files with different processing-modes as a criterion for event selection. In the *file*-mode, binary files are one-to-one converted into AERAROOTIOLib based ROOT files without any modification (besides implemented workarounds on corrupt data as discussed in section 5.5). Thus the

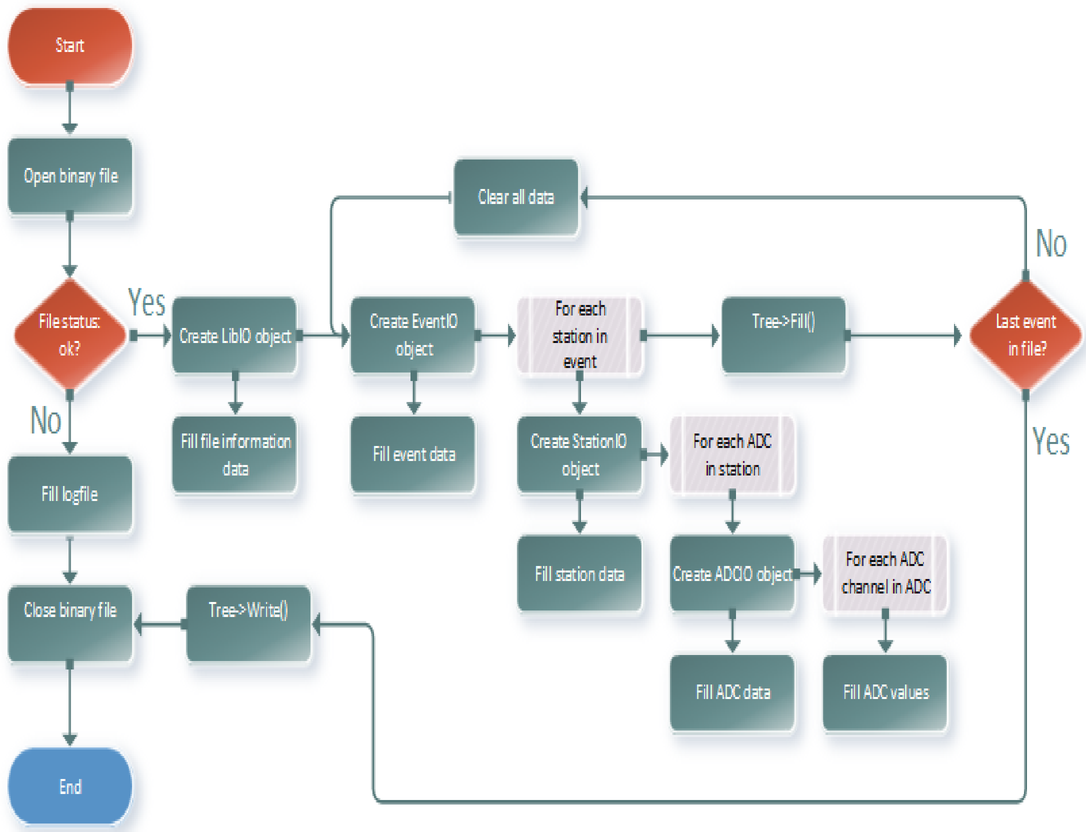


Figure 5.3: AERAConverter flow chart scheme showing the conversion process. First, the binary file is opened and its status is checked. Then the different class objects are created and the data is filled into them. The process is repeated until the last event inside the binary file is processed.

output file exactly matches the structure in the corresponding binary file but translated into the library structure. Furthermore, events can also be selected according to their properties and placed in specific output files. This can be either the event-time, resulting in a time-selection, or the trigger condition, allowing the separation into self- or externally triggered events. A general flowchart scheme for the conversion process based on the file mode is shown in Fig. 5.3. For a more detailed explanation, see section 5.3.5. The conversion on time basis can be done per hour, where all events recorded in the same hour are put together into the same file with the naming structure *YYYYMMDD_HH.root*, or per day (to make radio compatible to the Auger file naming convention), with the naming structure *YYYYMMDD.root*. Since events in the Auger files start at 12:00 p.m. UTC and not at noon (i.e. all events recorded between 2016-04-21 12:00 UTC and 2016-04-22 11:59:59 UTC are stored to the file 20160421.root), the same is done for the radio events in the day-wise conversion mode. Further selection criteria, like the minimum number of triggered stations per event have been tested during development and can be easily implemented.

Table 5.2: Amount of data with respect to different trigger flags for a complete day.

Trigger type	Sum [MB]	Mean [MB/h]	#	Mean [#/h]
self	2477	103	99233	4135
SD	9417	392	5079	212
FD	579	72	316	40
HEAT	1974	247	1081	135
AERAlet	593	25	313	13
physics	12563	523	6789	283
periodic	15947	665	8602	358
pass-through	2938	122	3581	149
monitoring	18885	787	12183	508
Aevb	1	< 1	99	11
overall	33926		118304	

5.3.2 Conversion based on the event trigger flag

In addition to the time or file based conversion mode, a subset of different events can be created based on the event trigger flag to split up the full data sample and produce different output streams in parallel to reduce the data volume. All events with the same corresponding event trigger can be written into the same file with a distinguishable filename suffix. Furthermore, all air shower triggered events which are used for physics analysis can be written into the same file with suffix _*Physics* with the additional input parameter "--physics", called the 'physics' stream. Those are the relevant events for merging radio with SD in CDAS, described in section 5.6. In addition, a 'monitoring' stream is produced. It is used in $\overline{\text{Off}}\underline{\text{line}}$ in the *RdMonitoring* module to review the status of the station hardware components and the data quality [92]. By using the stream, the monitoring information is updated with a delay of just 3 hours compared to real time. See section 5.4.1 for further details on data management in AERA and which events are contained in which stream.

New trigger flags named as *FD*, *HEAT*, *periodic*, *AERAlet*, *airplane* and *pass-through* have been implemented since the beginning of December 2013 (i.e. run 100760) for all radio stations and also into the AERAROOTIOlib. To decrease the data volume by a factor of 1/3, the rate of the *periodic* trigger has been changed in the DAQ from every 10s to every 100s in July 2014. The sub-streams based on the event trigger flag are easier to handle, more compatible to transport and the separation is useful for several purposes like the above mentioned continuous monitoring or using only airplane triggered data for timing calibration. In those cases, there is no need to analyse the full data sample. Especially for the analysis of airplane data, to calibrate the timing differences between the AERA stations, a significant decrease in the processing time was achieved [91] as airplanes occur infrequently and are limited to only a few hours a month. The amount of data produced by AERA on a typical day is shown in table 5.2.

The 'monitoring' stream is also used as the basis for the development of a noise library for AERA. For each externally triggered event in the 'physics' stream later reconstructed in $\overline{\text{Off}}\underline{\text{line}}$, the library searches the next periodically triggered event, which will contain information about the radio background, inside the monitoring stream in a time window of about

10 minutes around the detected EAS event. This event, taken for calibration purposes, is then used in simulations to get a quantitative measure of the noise. To thin out the data volume with the AERAConverter, an individual rate for the conversion with the integrated 'converttimeinvervall' option can be used. As an example, if this is set to 1000, only events which have a time difference of at least 1000 seconds are written to the ROOT file.

5.3.3 Input parameter list for the AERAConverter

A detailed explanation about the current state of available output streams, based on the trigger type, the conversion modes, and different functionalities in the AERAConverter, is given in the input parameter list below:

--s activates the file conversion mode. This is a one-by-one conversion of the binary file to a ROOT file. The old file name is used with ending .root attached

--h activates the hourly based conversion mode. All events, which have been triggered in the same hour are put into the same file with the structure YYYYMMDD_HH.root

--d activates the daily based conversion mode. All events triggered between 12:00 p.m. of the previous day and 11:59 a.m. of the following day are put into the same file with the structure YYYYMMDD.root

--debug activates the ASCII dump mode. All binary values are written out in a human readable format into an ASCII file. The input file has to be an already converted ROOT file or an ASCII file with converted ROOT files listed.

--streams if this flag is activated all output streams will be generated. Individually selectable output streams are:

- **--AERALET** events triggered by AERAlet stations - SD stations closed to AERA on a 433 m grid - are written out in a file with ending _AERAlet.root

- **--AEVB** events, where the event type is set to empty EventBuilder events - (_Aevb.root)

- **--AIRPLANE** events triggered by airplanes - (_Airplane.root)

- **--CALIBRATION** calibration triggered events - (_Calibration.root)

- **--FD** events triggered by FD - (_FD.root)

- **--GUI** DAQ GUI triggered events - (_GUI.root)

- **--HEAT** events triggered by HEAT - (_HEAT.root)

--LSNOEVENT events with stations which are sending the LS_No_Event type. These stations have no data inside and thus trace length = 0 (like stations with AEVB trigger flag but a hardware type != 7, i.e. aevb events). Internally in the IOLib, the AEVB trigger flag is set for these stations by the AERAConverter because it is not important to distinguish both triggers on the station trigger level. To differentiate, the station status variable is used. - (_LSNoEvent.root)

--pass-through T3 random triggered events. Those are events, which are passed randomly without filters, e.g. cone algorithm - (_pass-through.root)

--PERIODIC events, which are triggered regularly every 100s (before July 2014: every 10s) - (_Periodic.root)

--SCINTILLATOR local scintillator triggered events for stations with NIKHEF hardware - (_Scintillator.root)

--SD events triggered externally by SD stations - (_SD.root)

--SELF just self triggered events are written out - (_Self.root)

--physics this flag combines *scintillator*, *SD*, *FD*, *HEAT* and *AERAlet* triggered events into one file - (_Physics.root)

--monitoring for monitoring purposes, this flag stores all *periodic*, *calibration* and *pass-through* triggered events into one file - (_Monitoring.root)

--noADC if this flag is activated, no ADC values are dumped into the ASCII-file during debug mode, which reduces the size of the output file by more than a factor of ten

--logfile YYYYMMDD_HH a logfile is produced which stores the event number per trigger source, the mean number of stations and the station variance for each trigger source on a 10 minutes base.

--binarylistsdaily this option enables the production of sorted binary file input lists based on the event time stamp.

--binarylistshourly see binarylistsdaily. Produces binary file input lists on an hourly base into BinaryFileYYYY-MM-DD_HH.txt

--convertday YYYYMMDD if this flag is activated, just events with a time stamp on the given day are converted. No ROOT-file is created for other than the exact time period to avoid creating half-filled files at the border of each day at noon. Especially important for multi-process conversion to avoid writing into the same file from different jobs at the same time. For events with a time stamp beyond the selected time period, the message "Event is skipped due to input parameters." is shown during conversion.

--converthour YYYYMMDD_HH see convertday. Just convert events for the given hour.

--converttimeinterval ti activates selectable time interval functionality for conversion, with the parameter *ti* as the time in seconds between two events. This mode is useful for many tasks in analysis, e.g., the periodic trigger to thin out the data sample (use '... –PERIODIC –converttimeinterval 100' for the old 10s periodic trigger to just get a data sample which consists of periodic events with a time distance of 100s and which is equal to the new 100s trigger condition). This option starts always with the first event in a file. If this mode is used with a list of input files, the difference between the last event in the previous file and the first event in the next file can differ from the time interval. This is not a problem, since no exact rate between events is needed but just an additional layer of data reduction

--extract –outputstream enables the extraction of events based on the event trigger type from an already converted binary file into the same AERAROOTIOLib format.

Below are some examples on how to use the converter for different applications:

1. To convert the binary file ad012345.f6789 in daily mode and just write out the physics stream, use: ./AERAConverter ad012345.f6789 ./ –d –physics

2. To convert the binary file list Filelist.txt in hourly mode and just write out self triggered events, use: ./AERAConverter Filelist.txt ./ –h –SELF

3. To write out the values of the converted file ad012345.f6789.root, located in the next higher directory, in debug mode into the subdirectory "logs", use: ./AERAConverter ../ad012345.f6789.root ./logs –debug

5.3.4 Debug, binarylists and extraction mode

To review the data on a raw base, a so called *debug* mode can be used via the input parameter "--debug" to write out all the converted data out of a ROOT file as an ASCII dump into a text based file. Since the ADC values represent the biggest part of this dump, the additional parameter "--noADC" can be used to just write out the individual header information. A small example binary file for AERAConverter testing purposes is also included in the AERA-ROOTIOLib package. In table 5.3 important variables used internally in the AERAConverter which can be read out and interpreted in the debug mode are shown.

Since it is not possible to identify the event timestamps inside the binary file by looking at the binary file name, the 'binarylists' option has been designed which can be used in a daily or hourly mode. If this option is used, the AERAConverter loops over all events inside all binary files which are read in and produces sorted binary file input lists based on the event timestamp. For each event, the name of the binary file which contains the event is written into a text file with naming BinaryFileYYYY-MM-DD.txt or BinaryFileYYYY-MM-DD_hh.txt which corresponds to the date when the event has been measured. The content of the text file is checked and cleaned for duplicate entries. The Auger convention to store events from noon to noon is followed. As an example, if the file "'BinaryFile-2014-05-17.txt"' contains

Table 5.3: Detailed overview over all important values used internally in the AERAConverter.

Variable	**Value**	**Description**
AeraEventStatus:	0	event corrupt
	1	event ok
	2	no ADC channel with data in all stations
	4	trigger flag: AEVB
	8	trigger flag: LS_No_Event
AeraStationStatus:	0	no ADC channel with data
	1	station ok
	2	trigger flag: LS_No_Event
AeraStationHWType:	1	NIKHEF
	3	KIT/BUW
	7	AEVB

the entries ad100900.f0001 - .f0007, all events measured between 2014-05-17T12:00:00 and 2014-05-18T11:59:59 can be found in those seven binary ad-files.

The *extraction* mode, activated via the input parameter "--extract –outputstream", can be used to extract events based on the event trigger type which have already been converted into the AERAROOTIOLib format. The available output streams are the same as for the –streams parameter. Additionally, several output streams can be combined together. For example, the options '–extract –SELF –SD' extracts all self-triggered and SD-triggered events found in the ROOT based input file and combines them into a single ROOT based output file. The naming of the new file is built up on the input file and the selected output stream parameters. Extracting events is useful to thin out huge data volumes for which reconversion would take too much time. This has already been done for noise background analysis in which the pass-through trigger has been removed from the monitoring dataset and a pure periodic triggered dataset has been produced. Lastly, it's important to note that this mode can't be used in parallel to the debug or normal read-in mode.

5.3.5 Conversion logic

The logic to convert a binary file to a ROOT file inside the AERAConverter is explained below. First, it is checked that the binary file is available and can be opened. Then the file header length is checked. If one of the checks fails, the conversion is aborted and information about the error is written to a logfile. Next, the ROOT-TTree is generated inside memory, the file header information is read and the file content is iterated over for the first time to read all event types. The events are sorted based on their trigger type to increase the performance during write-out, see section 5.3.7.

After sorting, the loop over all events is performed. First, the event info is read, several checks based on the event timestamp are performed (e.g., GPS/Unix time or corrupt time periods), and the event trigger is set. A loop over all stations then follows. The station information is read and checks for the station timestamps and the trigger flag are done. The station hardware type is evaluated which is the basis for getting the correct additional station header information and then all ADC values of each ADC channel. After the station loop is performed, the required ROOT file is generated and the ROOT-TTree is filled into the file (if not already available inside by an earlier conversion). This process is then repeated for each event inside the binary file. If all events have the same trigger flag, the ROOT-TTree is written to the file and the ROOT file is closed. Otherwise, this is done after the last event of one trigger type and before reading the first event with a different trigger flag.

Generating the correct output ROOT file is not an easy procedure due to the abundance of input modes. To keep it as simple as possible, it is done as follows. First, a text string for the filename is built using the event trigger flag. Then, the event timestamp is written into a text string after being converted from GPS time to Unix time with the C++ *asctime* and *gmtime* functions. Since in the daily conversion mode, events are divided at 12:00 UTC and not at 00:00 UTC, special consideration has to be given to the exact timestamp. The algorithm first checks if events are detected before noon or after and then subtracts a day in the first case by using the C++ functionality *struct tm*. In the case of an event being before noon on the first day of a month, this would lead to an error, since day 0 does not exist. This problem is solved by simply subtracting an integer between 43200 and 86400 representing the amount of seconds of half a day from the event timestamp before applying the gmtime function. Thus, the timestamp is automatically handled as being from the day before, see listing 5.1 for the application in C++.

Listing 5.1: Algorithm to generate the string out of the event timestamp

```
if (mode & cModeDaily) {
  if (ptm->tm_hour <12) {
    if (ptm->tm_mday != 1) sprintf (root_filename, "%02i%02i%02i%
       ↪ s.root", ptm->tm_year + 1900, ptm->tm_mon + 1, ptm->
       ↪ tm_mday - 1, sTriggerType.c_str());
    else {
      iEvtSecondUnix -= 70000;
      ptm = gmtime (&iEvtSecondUnix);
      sprintf (root_filename, "%02i%02i%02i%s.root", ptm->tm_year
         ↪  + 1900, ptm->tm_mon + 1, ptm->tm_mday, sTriggerType.
         ↪ c_str());
    }
  }
  else sprintf (root_filename,"%02i%02i%02i%s.root", ptm->tm_year
     ↪  + 1900, ptm->tm_mon + 1, ptm->tm_mday, sTriggerType.
     ↪ c_str());
}
```

In addition, the exact time period in which events should be converted, can be specified as an input parameter for the AERAConverter. The event timestamp is then compared to this time period and if the event is not inside this range, the memory for the event is cleared and the event is skipped. The final ROOT output file name is the combination of the specified output directory, the name of the binary input file, the timestamp and the trigger flag. Some additional checks are performed. First, the status of the output file is checked. If the file exists, it is checked for corruption, e.g. due to power cuts during conversion, by ROOT internal functions. Then, it is checked to see if a TTree already exists inside the ROOT file. If this is the case, the TTree is just loaded into the memory and the event is linked by using the *SetBranchAddress* function of ROOT. If the event is already present in this file, it is skipped. If the TTree doesn't exist, a new TTree and its branches are created inside the file and the event is linked by using the *Branch* function of ROOT. Finally, after all information for one event is read in from the binary file and stored inside the ROOT-TTree structure inside memory, the 'Tree→fill()' method is called to fill the TTree. At the end of the conversion process for all events, the TTree is written to file by the 'Tree→Write()' method and the ROOT file is closed. See the *fSetTreeAndFile* function in the AERAConverter for further details.

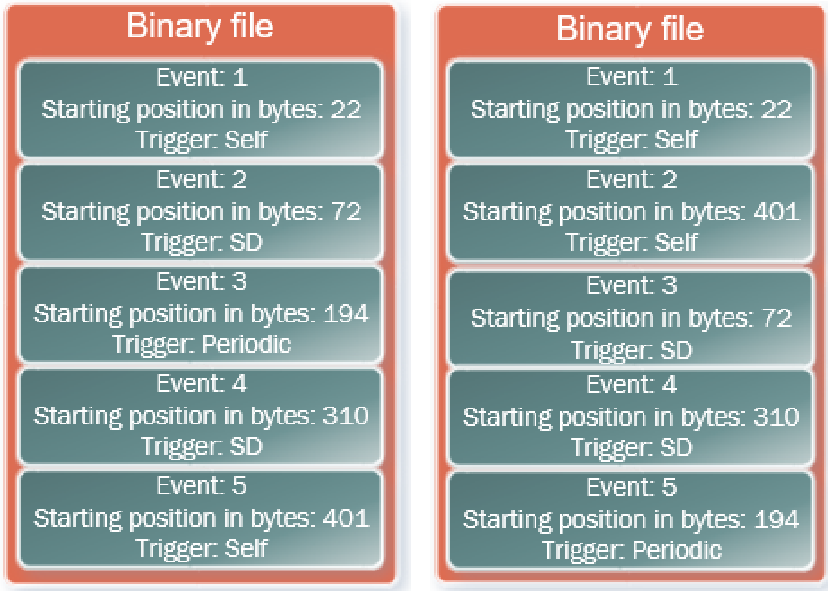
5.3.6 Modification of data stored in ROOT-TTrees

Although it is not directly possible to modify existing values inside a TTree already written to a ROOT file afterwards, attaching new events to the others inside the same TTree can still be done whenever needed, and the TTree gets updated automatically. Therefore, the conversion process does not need to take place completely at once, but can be arranged flexibly. This is an important feature for all conversion modes except the file-by-file mode, since not all binary files are necessarily available when the conversion is first started. However, if values which have already been stored need to be modified, a workaround based on copying data from one TTree to another can be used. In this case, a new TTree is created and its memory structure is linked to the old TTree inside ROOT internally. Before writing the new TTree into a file with the 'Tree→Write()' method, events can be loaded into the memory and their values can be changed. This is mainly used for pre-filtering the periodically triggered data for the noise library.

In Offline, Radio RFI are filtered by the *SineWaveSupressor* module, which is time consuming as it is a factor of 5 slower than real-time and must be repeated each time the analysis is done again. Alternatively, the output can be saved in the huge Offline file format. As a solution, pre-processed ADC traces, which have been filtered, can be stored again in new files with the AERAROOTIO-format with the ending '*_rficleaned.root'. This doubles the data-size but the gain in the CPU-time dominates. The 'WriteSelectedEventsOfflineWithTrace' functionality in the AERAROOTIOLib has been developed especially for this purpose of storing modified traces, but can be easily extended for other cases.

5.3.7 Processing speed optimization by trigger flag pre-sorting

The most time consuming part during the conversion is the call of the 'Tree→Write()' function used before each ROOT file is closed. In order to keep the memory consumption low, the logic of the AERAConverter keeps just one TTree in the memory at a time. This is also true for the 'streamer mode', in which the TTree only contains events of the same trigger type. Thus, for each event, the correct stream has to be chosen to be filled. As long as the following events have the same trigger type, the TTree can be kept inside the memory. If the next event has



(a) (b)

Figure 5.4: Scheme of the presorting algorithm based on the trigger type inside the binary file. An invented example file before sorting is shown in (a) and after applying the sorting algorithm in (b)

a different trigger flag compared to the previous one, the 'Write' function is called, the file is closed and a different file is opened. To boost the conversion procedure, events are sorted by their trigger flag in the streamer mode beforehand, as shown in Fig. 5.4. This is done by iterating over the whole file twice. In the first run, the trigger flag of each event is read out and the byte position of the event inside the binary file is stored inside a vector for each trigger flag separately. After looping over all events, all the vectors are merged to one combined vector and thereby all events are now sorted by their trigger flag. During the conversion, this vector is used to evaluate the starting position of the next event. This minimizes the number of 'Tree→Write()' calls for any set of events inside a binary file. In the worst case, the number of 'Tree→Write()' calls would now be limited to the number of possible trigger flags which is 13 at maximum. Pre-processing is needed anyway regardless of optimization since the event trigger flag has to be built based on the individual station trigger flags participating in the event. This is due to problems with the event trigger flag variable in the DAQ, see section 5.5.

5.3.8 Data interpretation and logfiles

During the conversion process for all conversion modes, the event-content is interpreted. This allows for consistency and data quality checks (e.g., wrong timestamps can be immediately corrected during conversion by algorithms directly implemented in the AERAConverter). This central place for corrections in the early stage guarantees that they are done once and consistently. Inconsistencies which are found during the conversion process are logged for later analysis. The values written to this log file are the input filename, the error message, the event run number, the event number, the event second, the event trigger type, the local station ID and the local station trigger flag. If the error message appears on event level and not on station level, the last 2 variables are set to 0. Present monitored inconsistencies and their consequences for the conversion process are:

- The binary file can't be opened - conversion aborted
- The file header length does not match the expected value or could not be read at all - conversion aborted
- The event length is incorrect - conversion of subsequent events is skipped since their position inside the binary file can't be calculated
- The ROOT output file is corrupt - conversion aborted
- The event type is of value Aevb or LSNoEvent - warning
- The event timestamp is out of range - skip the event
- Event binary data could not be read correctly - skip the event
- The station trigger flag is LSNoEvent - warning

5.3.9 Data analysis with the AERAConverter

The AERAConverter can be even used to do some simple but powerful statistics. For some data in 2013, the timestamps of two consecutive events showed a strange behavior. This was found by analysing these timestamps inside the AERAConverter and directly calculating their difference Δ T = T_2 - T_1. Although written later into the binary file, the timestamp of the second event is earlier than the first one. The largest ΔT between two events has been observed with -40000 seconds. The problem was reported to the Radio DAQ task who found it was due to a bug in the T3Maker. In the case of T2 triggers arriving late, a T3 trigger is built in the T3Maker for data in the past although this data is no longer buffered. This leads to events containing no data which can be minutes or hours late. Another reason was found to be due to network problems in the DAQ. In Fig. 5.5, an example for the event delay analysis is shown for the time period from April 2013 to August 2013. Here, the average delay was found to be 0.8 $\pm$ 1.0 s, well below the buffer size of 7 s.

5.4 Data management in AERA

With a data rate of roughly 15 GB - 20 GB per day, the data processing in AERA has to run automatically and regularly. Due to physical demands, a scheme of using different

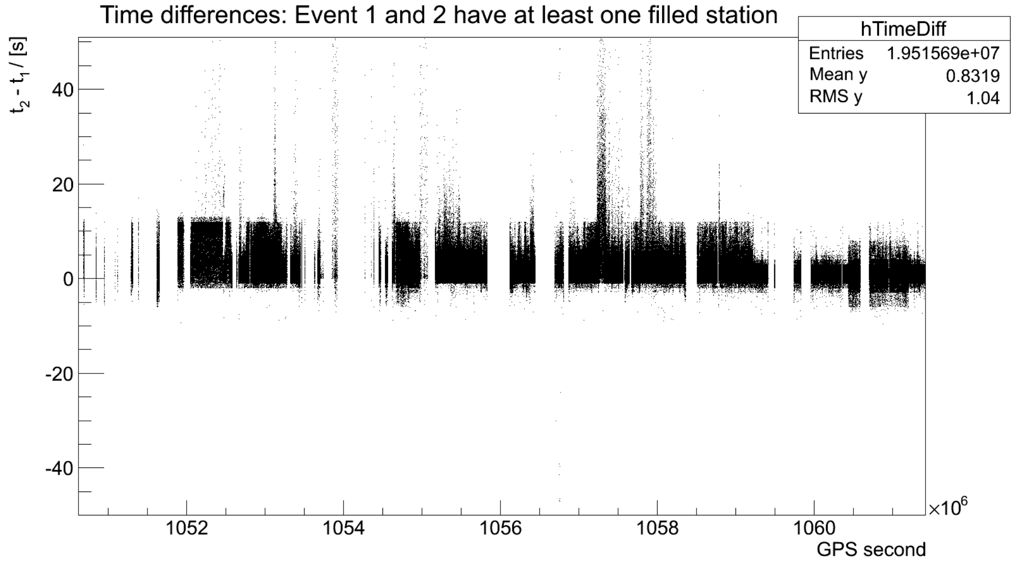


Figure 5.5: Event delay analysis done by the AERAConverter. The time difference Δ T = T_2 - T_1 between two consecutive events for the time period from April 2013 to August 2013 is shown. If everything works fine, this difference will be no larger than the maximum buffer length of the RDS, which is 7 s.

data streams has been invented in order to handle the measured radio data. This scheme and details about the mechanisms developed to autonomously run data production despite disturbing interferences, like power outages, are explained in this section.

5.4.1 Data streams

As shown in Fig. 5.6, the basic idea of handling the raw radio data is based on the event trigger flags. Events are split and combined into different files according to their trigger source. As already mentioned, the AERA DAQ stores all Radio events in run based binary files combining all triggers on the AERA server in Coihueco, the raw binary data. They are brought to Europe via HDDs several times a year and stored on FTP servers at KIT, RWTH Aachen and in Lyon. To backup the data, they are stored redundantly. In Coihueco, the AERAConverter splits the binary data into 4 different streams based on the AERAROOTIO format. The first, the ‘physics’ stream, contains all events triggered by SD, FD, HEAT and AERAlet which might be ‘real’ events caused by cosmic rays for a given day. The second, the ‘monitoring’ stream, is collected hourly and includes the pass-through, periodic and calibration events used to evaluate the detector behavior and noise background. As the name implies, the ‘self’ and the ‘airplane’ streams, both produced daily, contain events triggered by the RDS themselves or by airplanes flying over the AERA array. The last stream is copied directly to Europe via the internet due to its low trigger rates, while the monitoring and self stream are also brought to Europe via HDDs. The physics stream is merged daily via the CDAS EventMerger to the Auger xad or ad data in the IoAuger format to combine all detector types

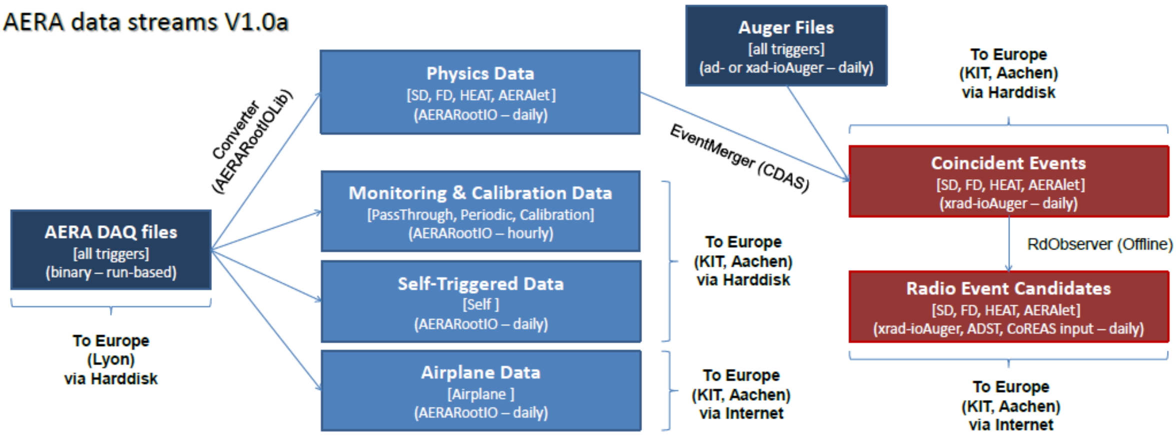


Figure 5.6: Data streams in AERA produced on the AERA server in Coihueco[93]. Different output files based on the event trigger flags are created by the AERAConverter. The physics data stream, containing all externally triggered events, is further processed in CDAS with the Auger data of other detector components and analyzed in Offline to create a subsample of radio event candidates. All files are transfered to Europe via the internet or hard discs.

in one xrad file which is easier to handle, see section 5.6.1. After merging, the stream can be deleted since all relevant events are included in the xrad file. This file is also transferred to Europe via HDDs since its size can exceed 10 GB. In the last step, the xrad file is processed via the RdObserver in Offline together with CoREAS simulation input files to select real radio event candidates. The output of the 'selected xrads or sel_xrad' files is processed in both the ADST format and in the IoAuger format. The latter is smaller in size and can be copied to Europe via the internet.

5.4.2 Data production in Argentina

The conversion of binary data in different streams and the merging of the German data with the Auger data is done automatically by cronjobs on a dedicated server for AERA in Coihueco. Currently, as seen in Fig. 5.6, the following data streams are produced:

- The *physics*, the *self-triggered*, the *airplane* and the *pass-through* stream on a daily basis
- The *monitoring* stream on an hourly basis

One cronjob is responsible for the daily production, a second for the hourly production and the third for merging. Parallel to the production of the monitoring stream, a monitoring logfile is updated on 10 minutes intervals with the input parameter '--logfile'. The timestamp, the number of events and the standard deviation for each of the event trigger types for the 10 minutes period is written into this logfile. Based on this data information like T2 rates can be calculated. In appendix A.3 the number of events per trigger flag for the time period from May 2015 until March 2016 using the information in the monitoring logfile is shown.

The automatic data processing has to meet two main requirements. First, it must prevent corruption during the data conversion and merging in the case of power cuts in Coihueco. The AERA server is protected against power disruption by an uninterruptible power supply (UPS) and is able to run on batteries for two hours to overcome short outages and to shut down the system properly. However, since power cuts can last for several days, the process of data production can be interrupted. As soon as the power is available, the server is switched on automatically and the data processing has to be checked for failures and catched up to be up-to-date again. The second requirement is parallisation. If several days have to be processed at once (e.g., after a power cut, if the Auger data couldn't be transferred over the network or in case of reprocessing lots of binary data), processing time is the main bottleneck. This is solved by running several instances of the process just with different time periods as input parameters in parallel. The daily and hourly conversion mode inside the AERAConverter and the merging algorithm are designed with this requirement in mind and thus the output of one job is not dependent on that of other jobs. With this in place, the bottleneck is limited to the data access speed. With conversion rates of 0.75 month/day, large speed-up through parallelization was achieved. To compare, if just one job is processed at a time, converting a single day of radio data into the different streams takes roughly 40 minutes, one hour of monitoring data 5 - 10 minutes and merging roughly 2 hours for a single day with 8000 events.

The automatic conversion done with Bash shell scripts will be explained based on the daily stream production and works as follows:

1. The typical time interval of Radio events inside one binary file is 60 minutes, depending on the amount of triggers. If a new binary file is started at 11:59 and closed one hour later, events detected before noon might first be available around 1 p.m. As a small buffer for lower trigger rates but to be still on time with the daily production and to match the Auger convention to store events in files from 12 a.m. to 11:59 a.m., the daily stream production starts at 3 p.m. local time. Each day of processing is one single call of the shell script.

2. First, it is determined by a variable how many days back from the current date should be checked in a decreasing for-loop. For the daily stream production, this is set to 20 days as default. If the used data volume exceeds 98%, the process is interrupted as a further protection against corrupted ROOT files. Then, the uptime is calculated to estimate if files have been created before or after a system reboot. In the next step, a file with naming 'YYYYMMDDrunning.txt' is checked for existence and for its date of creation. It is created directly after this check and immediately deleted after the conversion process at the end of the shell script when a file with naming 'YYYYMMDDdone.txt' is written. As long as the file 'YYYYMMDDrunning.txt' exists and if it has been created after the system uptime, other processes will skip the related day and continue on the next one. If this file is older than the system uptime, this might be due to a power cut appearing during the conversion process. In this case, the file is deleted. Just by using this additional file, both requirements parallelisation and catching power cuts can be fulfilled.

3. The last check is made on the file 'YYYYMMDDdone.txt'. If it exists, the data processing for the specific date has already been done successfully and the script continues

with the same procedure on the next day. In the other case, the actual procedure starts. First, all already processed and existing files are deleted for this day, because they might be corrupted. Then the relevant events inside the binary files detected on the day to be converted have to be found. Since it is not clear, which event is written in which binary file, this is done in two steps.

- As a first rough estimation, all available binary files are checked for being created in a time period of 3 days around the actual date with the additional option $-mtime+(i-2)$ and $-mtime-(i+1)$ for the 'find' command with i representing the day which is processed. If no binary data is available, an empty logfile called 'NoBinYYYYMMDD' is created. Files which meet the date criterion are written into a list of binary files with the structure 'BinaryFileListYYYYMMDD'.
- This binary list is used as input parameter for the AERAConverter with mode '–binarylistsdaily'. Using this mode, the AERAConverter generates a file in which only binary files are written which contain events detected on the relevant day. If no events have been detected on this day by the AERAConverter, an empty logfile called 'NoRadioYYYYMMDD' is created.

4. Finally, the AERAConverter is started again using the list of binary files and the input parameters 'OutputDir –d –streams –physics –AIRPLANE –SELF –convertday YYYYMMDD'. In the last step, all produced files are moved to their appropriate folder on the AERA server for common use.

The algorithm for the hourly monitoring stream conversion works similarly, with an additional loop for each hour. The script runs with a delay of 2 hours and thus has a small buffer to process the monitoring data. The first binary list is created using the '-mmin' argument of the 'find' command to search for binary files created in the time interval spanning the prior 4 hours and 30 minutes. This file is then used for the AERAConverter to find relevant events with the option '–binarylistshourly'. Finally, the AERAConverter is started again with the binary list and the input parameters 'OutputDir –h –monitoring –logfile –converthour YYYYMMDD_HH'.

The shell script for the CDAS merging also works similarly, and runs every night at 11 p.m. local time, but does so without creating the binary lists and doing the conversion. Instead, the script checks whether the physics stream exists and checks for the corresponding AugerXAD file and if not found for the AugerAD file. Finally, the merging is started with the input parameters './BuildXRAugerFile –force -o OutputDir -v2 -r0 -i PhysicsStreamInputFile AugerFile'.

5.4.3 Data production at KIT/Lyon

The mass data production can be set up easily on any other server if the binary data is available and their md5sums are compared with officially available lists. If not already available, the AERAConverter is first used to produce the binary file lists in both the daily and hourly base. With those lists, which include the information of which binary file contains which event timestamps, time periods can be parallel processed on a cluster. The same shell script logic explained in section 5.4.2 is used to produce the streams and directly merge Radio and Auger data. This has been done already on the cluster at the KIT, to mass produce the Auger xrad files in Europe. In addition, the noise library has already been processed on the cluster at the RWTH Aachen.

5.5 Known problems and their solutions in the AERAROOTIOLib

In almost 5 years of handling the data of a complicated multi-component detector like AERA, technical problems are almost inevitable. The main objective when finding solutions is to recover as much of the data as possible. These so called 'workarounds' are implemented inside the AERAConverter and the library itself to catch all known problems. They are explained below:

- Timing conversion bug: For some events before 2013, the Unix time, instead of the GPS time, for the event was been written to the binary files on event or station level. Therefore, incorrect Unix timestamps are converted into the GPS standard taking into account leap seconds. If a new leap second is announced by the International Earth Rotation and Reference Systems Service (IERS) [94], it only needs to be implemented once in the AERAConverter. Unix timestamps are found by determining if the event has been detected after the year 2021 due to the timing difference between Unix and GPS and taking into account that the data taking started in 2011. This workaround needs to be adapted after 2021 when converting old files since dates after this time will then be valid. At this point, an event from 2011 with an incorrect Unix time will be indistinguishable from a properly recorded event from 2021 without also checking the run number of the event. For the AERAROOTIOLIB, existing functionalities to convert timing formats in Offline can't be used as the library needs to work as a stand-alone tool.

- Timing bug: For some events, the event timestamp is completely off (e.g., it has a time in the year 2070). As a temporary solution, to prevent these incorrect timestamps, events are checked to be in the time range between 2010-01-01 and 2029-31-12. If it is not in this range, the event is skipped during the conversion process, since an event without a correct timestamp is of very little use.

- Corrupt data: Many events during the runs 100746 - 100752 can have corrupt data (e.g., contain timestamps after the year 2030). These runs took place during Nov. 22, 2013 and Nov. 28, 2013. All events from those runs are skipped.

- ADC resolution bug: ADC values from Dutch stations need to be corrected for the ADC resolution for events from December 2012 until the station hardware was switched to version 7 in April 2013. ADC counts are written inside the binary file based on a station resolution equal to 14 bits although the real resolution of the stations is 12 bits as is correctly written in the station header information.

- German additional station information: Until run 100220, the German additional station information can contain incorrect values. As a solution, all of the additional information is set to 0 for these events. In addition, with run 100221, the data structure of the additional information changed from 13 data fields to 21 fields. This is taken into account by correcting the byte pointer operations while reading in the binary data.

- Mac OS operating systems: For some compiler versions on Mac OS, the conversion led to a ROOT segfault. The problem appeared if events inside the binary file had more stations than the first event, since the maximum of the allocated memory per event has been calculated internally in ROOT based on the number of stations in the first event. Thus, the memory was not correctly re-allocated for the other events. This could be solved by implementing the 'SetAutoDelete(kTRUE)' option on the corresponding TBranch. However, on some newer setups with Linux operating systems, this option caused rare seg-faults. The SetAutoDelete option is now disabled as default as the problems on Mac OS setups disappeared. Another point which has to be distinguished for Mac OS setups affects the linker flags. To compile the AERAROOTIOLib properly, the option

```
SET_TARGET_PROPERTIES(aerarootio PROPERTIES LINK_FLAGS "-
  ↪ undefined␣suppress␣-flat_namespace")
```

  is set in CMAKE by default.

- Periodic trigger: Before run 100760, no unique trigger flag for periodically triggered events, sometimes referred to as *10s or 100s trigger*, existed in the data. Instead, the SD trigger flag had been assigned to them. As a workaround, those events were identified by the nanosecond of the event-time to be close to the 0 GPSns. To be consistent with the trigger flag definition, the AERAConverter sets the old periodically triggered events with SD event trigger to *periodic*.

- Trigger flags: Inside the binary files, the event trigger flag is set as an integer value to represent only one specific event type. In many cases, stations in one event can have different trigger flags (e.g., if the event is triggered by SD, one or more stations can have no data). In this case, the common event trigger flag is set only to SD, but the individual station trigger information is hidden inside the station data. In addition, for some data periods in the Dutch DAQ, the event trigger flag was not set correctly at all but the stations trigger flags showed no problem. All of this is solved by the AERAConverter. It loops over all stations in one event and reads out every station's trigger information before the actual conversion of the data starts. Then, the event trigger level is set as a bit pattern based on all different station's triggers. In addition, a getter and setter function exists for each individual trigger type on event level so that each trigger type can be checked separately. By doing this, the event trigger has to be interpreted only once during the conversion and not in later applications. This is especially important for the combined dataset of German and Dutch RDS.

- Run based dataset: The different DAQ systems for German and Dutch stations started data taking with different run numbers. While the German DAQ started at 100000, the Dutch DAQ started at 200000. Changes in the binary data structure are included starting at a specific run number and because a unique revision number is not stored in the binary data itself, workarounds are based mostly on the run number. This is problematic especially for the combined dataset of both DAQs and also in cases where changes are applied and undone in the same run. This is solved by using the station hardware version variable in addition with the run number. In some rare cases for stations with Dutch hardware, the station hardware variable is not set correctly and contains the value 0 which leads to a misinterpretation of some binary values. A more robust and less error-prone method will be used in the future. One possible solution is to write a DAQ svn revision number directly into the binary file. With this variable, a unique identification of changes is possible and the workaround can be simplified.

5.6 CDAS - Low level merging

The different detector components of the Pierre Auger Observatory write separate ROOT based raw files. CDAS merges SD-FD into a combined Auger *ad*-file. The AMIGA group then merges the MD data into this *ad*-file to make a combined *xad*-file. CDAS is now extended for low-level file merging with radio. This is easily implementable with ROOT based radio files and with an extended version of the class *IoAuger* using the CDAS-*AugerEvent*-structure. These modifications of the IoAuger file-type have been implemented in $\overline{\text{Off}}\underline{\text{line}}$. Therefore, the merged events are fully supported. In this section, the new *BuildXRAugerfile*-application which takes over the FD-SD-RD merging process as well as the RD-RD merging of German and Dutch data to one combined dataset is explained.

5.6.1 BuildXRAugerfile application - Overview

The new application *BuildXRAugerfile* has been introduced and already included into the CDAS repository to implement RD-SD-FD-MD merging interfacing AERAROOTIOLib with CDAS. The radio files taken for the merging process are those produced in daily mode with the flag '--physics' created with the AERAConverter. To set the correct output file name, the radio input file has to be named according to the xad-file as *rd_yyyy_mm_dd_12h00.root*.

The merging procedure works in analogy to MD, but has been enhanced to adapt to radio as follows: Since the radio events are in some cases not written to the binary file in a chronologically order as well as to save computing time, the radio events are first sorted by their timestamp. Then the whole Auger xad-file, or if not available, the ad-file, is processed event by event and the whole radio event list is scanned for a match in a distinct time window, until the RD timestamp is greater than the current SD timestamp. This match is based on the time difference between the timestamp of a radio event (earliest triggered RDS inside the event) and the IoAuger SD GPS timestamp. The distribution of those time differences peaks around 0.01 ms for *AERAlet* triggered radio events and around 0.137 ms for the others, as is shown

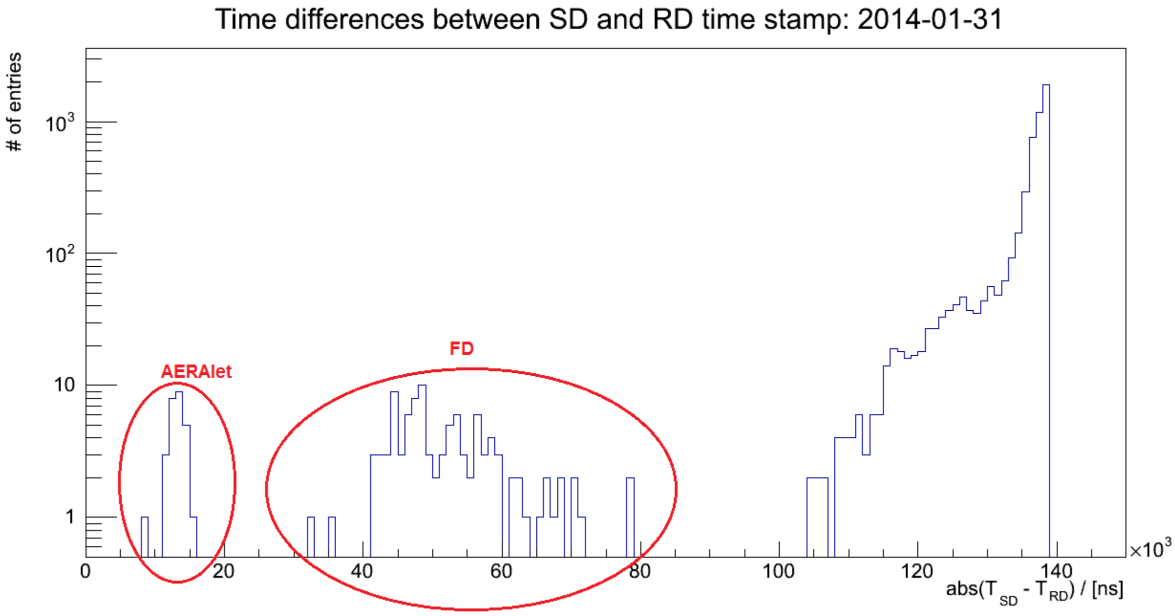


Figure 5.7: Timing differences between SD/FD and RD timestamp. Different distributions are visible. The first one, with a peak at around 0.01 ms is due to events triggered by AERAlet and the second in the middle contains FD triggered events. The distribution at the end which peaks at around 0.137 ms is generated by SD triggered events. Its form can be explained by the different geometries of an EAS and the abrupt end is related to the technical implementation of the DAQ.

in Fig. 5.7. Then the next SD event is taken and the process is repeated. If a match is found, the radio event is added via the new PushREvent function of the AugerEvent class and the current position of the radio event inside the list of events is stored in order to save computing time by not scanning through the whole radio list again. In case of an external trigger by FD, the Radio DAQ writes out two, or in case of HEAT three, events caused by the same EAS and thus containing the same data just with a minimal time offset and a different event header inside the binary file. To prevent merging the same event several times to the same Auger event, the BuildXRAugerfile builds just one event of the multiple Radio events. This is done by taking the event with the earliest timestamp and modifying its event trigger flag with the bit pattern of the other events. This event can then be merged to the corresponding Auger event.

In cases when a Radio event can be matched to the Auger event, several new AugerKeys have been introduced:

- XRdSd: merged SD and RD data
- XRdFd: FD + RD
- XRHybrid: SD + FD + RD

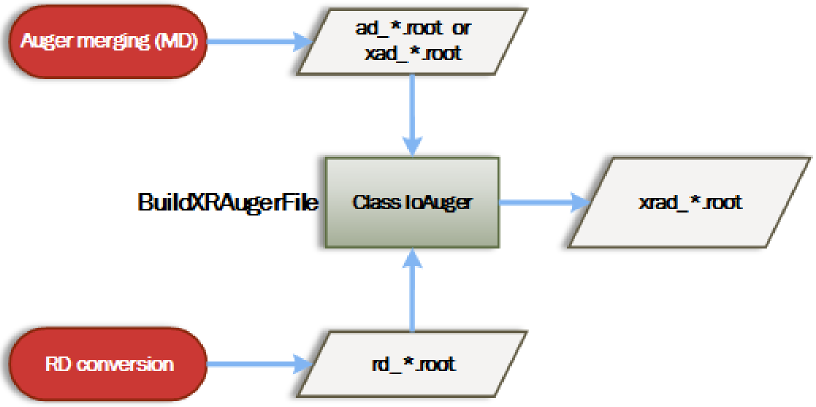


Figure 5.8: Radio merging scheme of Auger ad- or AMIGA xad- and Radio rd-files into the final *xrad*-file in CDAS with the BuildXRAugerfile application developed for Radio.

- XRM: SD + MD + RD
- XRMHybrid: SD + FD + MD + RD

Radio events are written into the new output file named *xrad*, if a corresponding SD and/or FD partner has been found. Additionally, the timestamps of the radio events and the auger-*xad* events can be dumped into ASCII files for later analysis in a debugging mode. This mode also writes out a list of event numbers which have not been merged with a short explanation telling why no matching candidate has been found. A short sketch of the procedure is shown in Fig. 5.8. A small debug application to read out all Radio related values and some SD values stored in the xrad format, in an analogy to the debug mode of the AERAConverter, is also integrated. As a result of this new merging procedure, the data from different detector setups are combined in one file in the IoAuger-format. There are several advantages of the new combined data format (e.g., just one input file is needed for hybrid reconstruction in Offline).

The merging algorithm works reliably, for example, in July 2014 less than 1% of all externally *SD* triggered radio events were not merged with the SD. For those missing events, no corresponding SD event in the Auger file is found in the specific time window, indicating a problem in CDAS. In Fig. 5.9, the distribution of externally triggered radio events in the ROOT file and the merged events in the xrad file for the data period between July 2014 and October 2015 is shown. Figure 5.10 shows the difference between both distributions. This helps also to discover problems with CDAS data-taking.

5.6.2 BuildXRAugerfile application - Reducing the data volume

As a nice bi-product, the low level CDAS radio merging enables more detailed analysis for different purposes. As seen in table 5.2, the amount of data of externally triggered events is roughly 10 GB per day which is at the limit of what can be transferred from Argentina

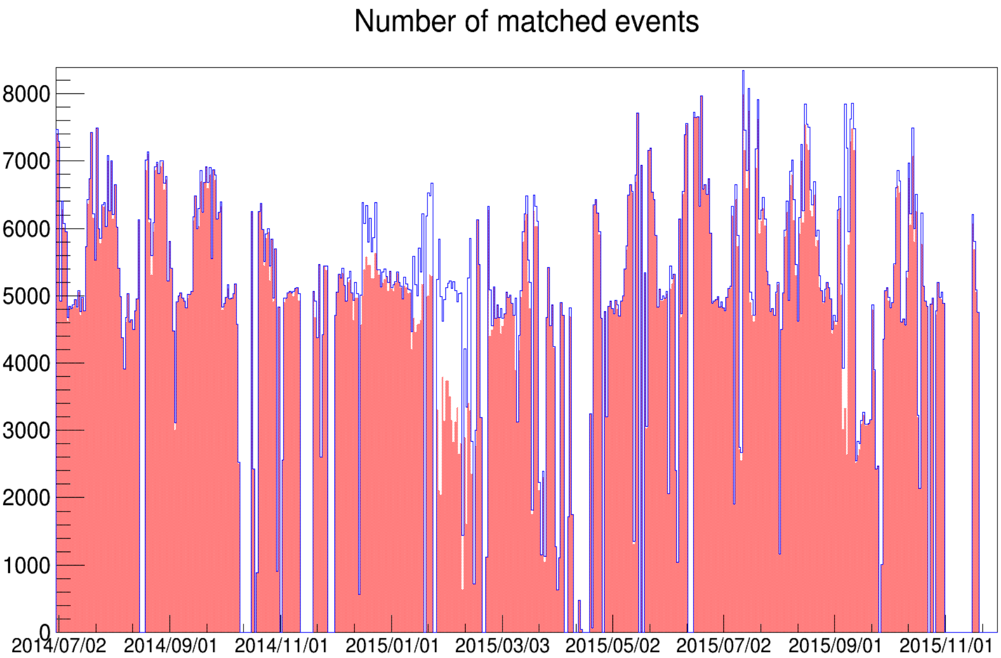


Figure 5.9: Distribution of externally triggered radio events, the *event candidates* in blue, and matched events with Auger SD/FD in red. In overall, both distributions agree well, discrepancies can be explained by missing SD events due to T2 data losses in CDAS, or other factors.

to Europe on a regular basis. To reduce the data volume, it would be sufficient to exclude events that don't contain physically relevant data for AERA from the merging process in the early stage (e.g., if the distance between the RD and SD station is larger than 10 km). Figure 5.11 shows the distribution of minimum distances between the closest SD station to an AERA station for each event in one day. This first rough estimation excludes just a few events ($\mathcal{O}10$). A deeper look into this scenario is necessary, but for higher efficiency, a reconstructed shower axis or core position would be needed, neither of which are available in the raw SD data used for low-level merging. Simply reducing the maximum distance between the core position of the EAS and the RDS in the DAQ trigger would not be possible as it increases the risk of losing horizontal EAS with a large Radio footprint.

5.6.3 BuildXRAugerfile application - Input parameters

The following set of input parameters has been defined for the *BuildXRAugerfile* application:

--v activates verbose or debugging mode which provides more information about the merging procedure in the output file.

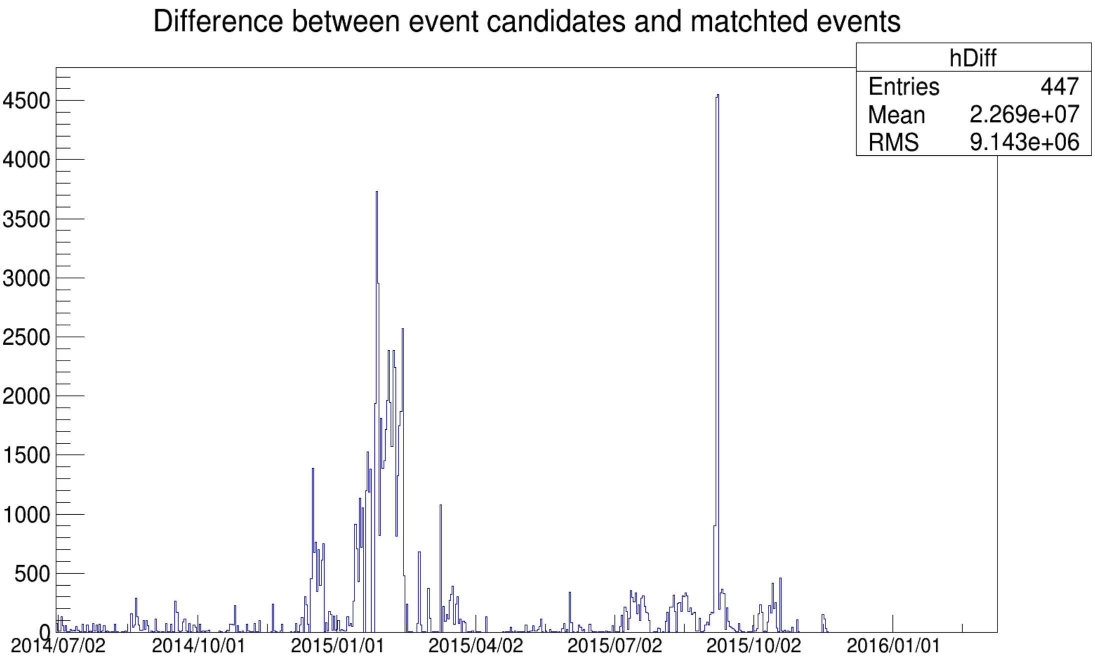


Figure 5.10: Distribution of differences between event candidates and matched events seen in Fig. 5.9.

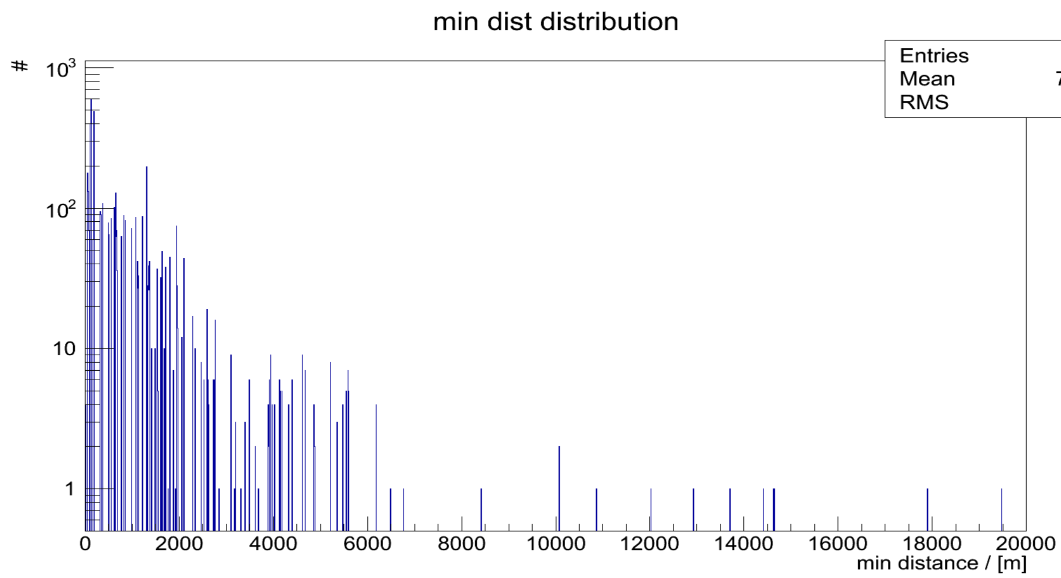


Figure 5.11: Distribution of minimum distances between any SD and RD station for all detected events on a single day. The data volume measured by the RDS cannot be lowered by simply excluding all events with a minimum distance of more than 10 km since they make up only a small fraction of the complete dataset.

--r activates *RdObserver* mode. This mode uses files with radio events already reconstructed by *RdObserver* in the AERAROOTIOLib format as input files. In this mode, in contrast to the standard merging procedure where radio is merged to a SD/FD/MD event if a match is found, the SD/FD/MD partner is merged to the radio event set to get an output file where all events have radio data. This mode is primarily useful for time periods where only *ad* files are available. To avoid stopping the radio analysis chain with *RdObserver*, the *ad* files can be processed together with the radio files with the *RdObserver* to get an interesting subset of radio events written out by the *EventFileExporter* which can then be merged together with the *xad* MD data afterwards.

--m activates the merging of RD-RD. The command to start the merging procedure is

```
./BuildXRAugerFile -o OUTPUTDIR -v2 -m1 -i GermanRD-file
   ↪ Auger-file DutchRD-file
```

The RD-files in the AERAROOTIO-format must have the naming structure *rd_ yyyy_ mm _ dd_ 12h00.root* while the Auger file must be structured as *ad_ yyyy_ mm_ dd_ 12h00.root*.

5.6.4 CDAS - RD-RD merging

For a time period from the *AERA124* deployment (see section 4.1.2) until the beginning of 2016 the NIKHEF stations were taking data in a parallel DAQ. As a side product of the other Radio merging processes in CDAS it is also possible to merge RD-RD together into one file with AERAROOTIOLib which is necessary for events measured before 2016. The process is analogous to normal merging: For the data measured by KIT/BUW stations, the produced *physics* stream is loaded into the *BuildXRAugerFile* application. For those measured by NIKHEF stations, the files which contain scintillator and self triggered events in coincidence with SD are taken manually and converted using daily mode with the AERAConverter and loaded into the application[1]. At first the coincidence between the SD and the KIT/BUW stations is checked. The same is then done with the NIKHEF stations. If one of of those matching possibilities is found, the corresponding radio event is merged with the SD. If both possibilities are matched, the KIT/BUW stations' event header is updated with the new number of stations, a new trigger flag pattern is built and a new event status is set[2] by the AERAConverter to a specific pattern for RD-RD merging. Then, the vector of NIKHEF stations is attached to the KIT/BUW vector of stations via one simple push-back now possible with the new ROOT based file format structure and the extended *AugerEvent*-class.

Additionally, a visual interpretation of the merging process has been added to CDAS. For each event which has been merged, the triggered KIT/BUW and NIKHEF radio stations in this specific event are drawn. Also, the position of the SD stations which have been triggered and the position of the CRS are added to the map. Figure 5.12 shows an example plot for a

[1] As soon as these files are regularly available on the dedicated AERA server, the above production of the NIKHEF daily files for RD-RD merging will be automated.
[2] value: 0x10

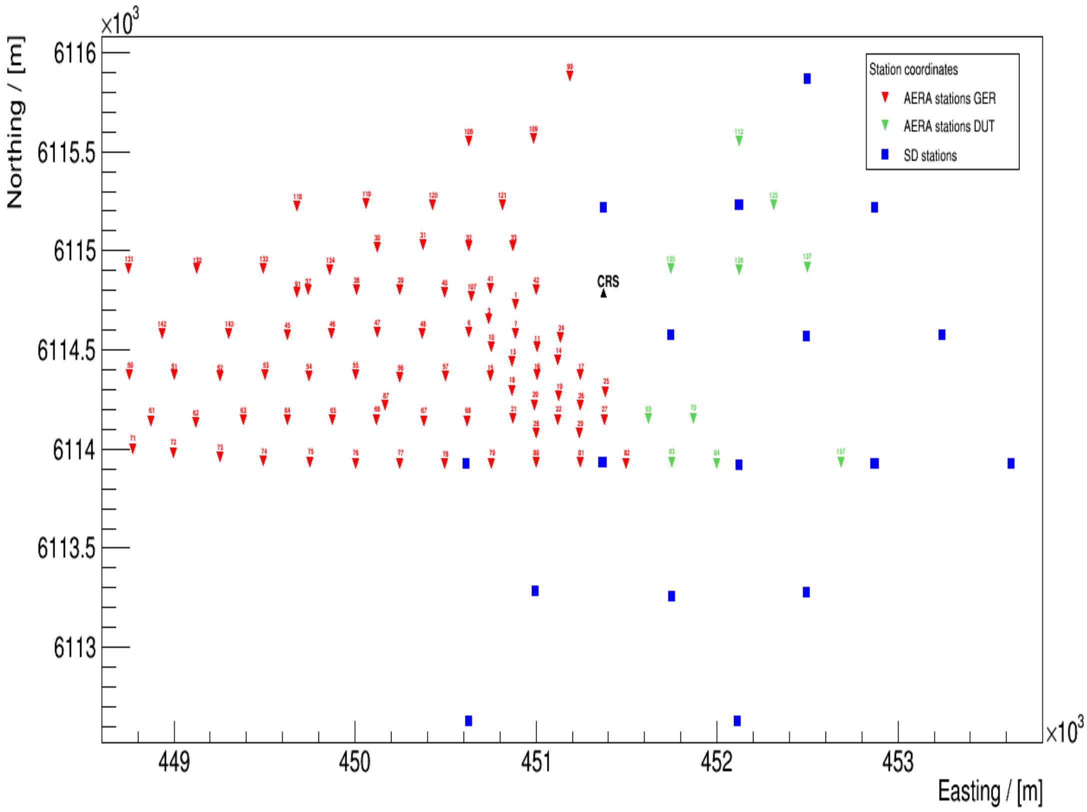


Figure 5.12: Positions of SD and RD stations participating in an event for a KIT/BUW and NIKHEF RD-RD merged file in CDAS. The RDS ID is attached to the marker position.

randomly chosen event. The RD-RD merging option is already in use for the data analysis of horizontal EAS.

5.7 Offline

Besides CDAS, also Offline has been extended to handle the new AERAROOTIOLib data format and is now used for almost all radio analysis. A new version of the EventFileReader, using the AERAROOTIO library for the new *RadioAERAroot* input file type, and an updated version of the interface for the new merged *xrad* files in the *IoAuger* input file type have been implemented in Offline. In addition, new enum values for the new trigger flags have been added.

The export of specific, though not necessarily reconstructed, radio events in Offline is done via the *EventFileExporter* using the AERAROOTIOLib. This module allows low-level selection based on different criteria (e.g., min. number of stations in one event), similar to the

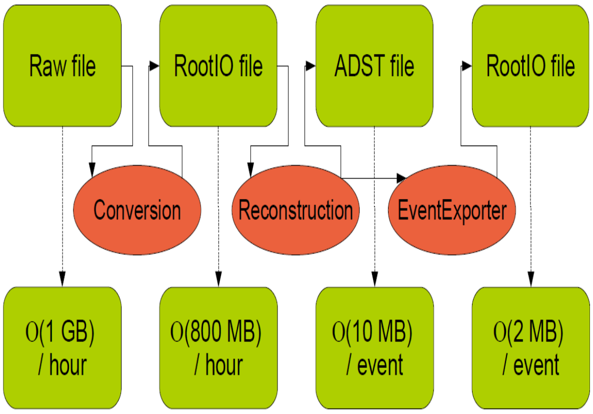


Figure 5.13: Data processing procedure in Offline with the amount of data per step [95]. Data reduction in the order of 500 is achieved.

procedure integrated in the AERAConverter. In this case no reconstructed quantities are needed. Alternatively, the selection of events is based on reconstructed quantities (e.g., SNR or SD-coincidence) can also be based on low-level variables like the min number of stations for reconstructed events. All selected events are then written into a common ROOT file by the *EventFileExporter* in either the AERAROOTIO format or, if the input file has been a merged xrad file in addition to the ADST output file, into the AugerIO format. This allows storing the full raw information of an event and not only the reconstructed information as in the ADST format, or including the whole huge internal class structures as is the case for the Offline format which can be a factor of 1000 larger than the raw format. Because of this, fast re-reconstruction of those selected events with different input parameters is possible. Thus, interesting subsets can be created for each analysis which goes hand in hand with reducing the data volume and improving the processing speed. Figure 5.13 shows the general procedure and the average reduction of file size per cycle. Besides physical analysis, the subsets can also be used to improve the whole Radio reconstruction chain in Offline [96].

Offline has also been extended to handle the new Radio trigger flags in the German DAQ. It is now also able to handle the temperature values of the station electronics which are written into the binary file to evaluate whether the hardware temperature has any influence on the measured data. Example files, containing reconstructable events, to test the new AERAROOTIOLib based format are located in the SampleShowers directory within the Offline documentation. Table 5.4 shows all current trigger flags available, the hex value in the DAQ, the corresponding ROOT file name suffix (if the streamer mode in the AERAConverter is activated) and the translation of these trigger flags to the Offline enum values.

Table 5.4: HexValues and translations of trigger flags in Offline

Trigger type	**DAQ hex value**	**Streamer mode file ending _***	Offline **enum value**
self	0x1	Self	eSelf
calibration	0x4	Calibration	eCalib
scintillator	0x2	Scintillator	eScint
SD	0x8	SD	eExternal & eSD
GUI	0x20	GUI	eGUI
FD	0x40	FD	eExternal & eFD
HEAT	0x80	HEAT	eExternal & eHEAT
periodic	0x100	Periodic	ePeriodic
AERAlet	0x200	AERAlet	eExternal & eAERAlet
airplane	0x400	Airplane	eAirplane
pass-through	0x800	pass-through	epass-through
Aevb	0x0	Aevb	eSkipEvent

6

Preparations for an FD based dataset for Radio X_{max} analyses

In order to analyze the data taken by the RDS, many different steps are involved. It starts with data acquisition and preparation as the raw data provides the base of all further activities, as shown in chapter 5. Calibration is also very important. The antenna sensitivity response of each station, the timing differences between all stations and the environmental conditions have to be known. All of this information is used as input for the sophisticated analysis algorithms in $\overline{\text{Off}}\underline{\text{line}}$. The hybrid design of the Pierre Auger Observatory adds another level of quality to the analysis. As most of the radio events are also measured by the other detector techniques, each detector's advantages can be combined when a common event dataset is available. In this chapter, the procedure to generate and evaluate this reference dataset, with a focus on events measured by the FD telescopes and the radio detectors, is explained and first results are shown.

6.1 Standard ADST cuts

By measuring the atmospheric depth of an EAS, predictions about the composition of the cosmic rays can be made. FD-telescopes are designed to measure the deposited energy as a function of the slant depth and therefore measure the $X_{\max}$ very precisely. Simulations have shown that radio detectors can contribute to the estimation with a similar precision. However, in spite of the FD telescopes, no common standardized way exists for how to evaluate $X_{\max}$ using radio data. Currently different methods to reconstruct the atmospheric depth of the shower maximum are under development. One method is reconstructing $X_{\max}$ by comparing the amplitude of the signal to the signal strength in the LDF. The distance to $X_{\max}$ can be estimated through the different widths of the radio footprint. The other method uses the signal timing information to find correlations between the radio wave front and $X_{\max}$.

In order to test the different methods for the $X_{\max}$ analyses, a reference dataset which can be used as a benchmark for comparison of the different reconstruction methods is necessary. This dataset should contain radio events which are reconstructable in $\overline{\text{Off}}\underline{\text{line}}$ and fulfill some basic quality requirements like a minimum number of triggered stations. Since the FD $X_{\max}$ recon-

Table 6.1: Official ADST cuts available in Offline

General description	Cut	Value	Detailed description
fiducial volume	badFDPeriodRejection		reject periods with bad calConsts, GPS-glitches
geometry related	maxCoreTankDist	1500	max. shower plane distance core and hybrid-tank
	nAxisPixels	5	min number of pixels used in axis fit
profile related	xMaxInFOV	0.0	max distance of X_{max} from borders
	xMaxError	40.0	max error on xMax [g/cm^2]
	energyError	.2	min / max error on energy (relative)
	profile χ^2	2.5	max reduced GH χ^2
	minViewAngle	20.	minimum viewing angle

struction is well established, also having a 'good' FD reconstruction for all events inside the reference dataset to directly compare to in order to evaluate the radio methods is preferred.

As a first approach for this set, a list of 22 events is used as the baseline for comparisons [97]. These events have been chosen without applying the quality criteria, but instead by the application of loose quality cuts. These events must be both reconstructable with the RdObserver in RD, and also have an FD event and a comparable simulation event. To evaluate the quality of the FD reconstruction, two different quality checks can be used. One check contains the standard official ADST quality cuts available in Offline. The other check uses an enhanced ADST cut set and is described in the next section.

The ADST cuts, listed in table 6.1, are applied to the Auger dataset. For each of the 22 radio events, the corresponding FD event is found in the ADST files. Two different ADST datasets have been used, version v9r5 - hybrid and version v10r0 - HECO. HECO is the combination of HEAT and Coihueco to one virtual eye. Since a complete re-reconstruction of the Auger data would be necessary, the new FD calibration database was not used in this analysis. The v9r5 dataset does not contain a HECO reconstruction.

The results of applying the ADST cuts to all events are listed in table 6.2. The first column counts the number of events. The second column lists the Radio run number and event number. The last two columns describe the results of both Auger data versions. If the cell is marked green, the event passes all cuts in at least one of the two datasets. If it is marked red, the event failed at least one of the cuts. If it is marked yellow, no Auger data for either v9r5 or v10r0 is available. Using the v9r5 dataset, only 9 of the 22 events pass the cuts while the rest of them fail because no HEAT profile was reconstructed. For the v10r0 dataset, 18 events pass the FD cuts and no Auger data is available for three. The differences between both datasets for one example event can be seen in Fig. 6.1. Although the light curve looks similar, the HEAT X_{max} profile is reconstructed for the v10r0 event, but not for the v9r5 event.

Table 6.2: Results of the analysis using the official ADST cuts for the different Auger reconstruction datasets v9r5 and v10r0. If the cell is marked red, the event fails at least one cut. If it is marked yellow, no Auger data is available and if it is marked green, the event has a good basic FD reconstruction. To be taken as a high quality event and therefore marked green, it has to pass all cuts in at least one of both datasets.

#	Event	v9r5 - Hybrid	v10r0 - HECO
1	100759-175530		
2	100759-303383		
3	100787-456636		
4	100795-171250		
5	100823-46784		
6	100823-59774		
7	100823-68115		
8	100838-57641		
9	100865-302837		
10	100869-455911		
11	100870-357969		
12	100870-85032		
13	100871-2702		
14	100885-83147		
15	100887-154817		
16	100896-575539		
17	100898-104598		
18	100909-67806		
19	100951-44027		
20	101005-57928		
21	101019-35126		
22	101024-113090		

6.2 Enhanced ADST cuts

Since the standard ADST cuts only cover basic quality criteria, a more sophisticated set of quality cuts is applied to only the v10r0 dataset [98]. These FD cuts have been designed to measure the elongation rate with the FD telescopes and are very strict to avoid a bias in X_{max} through the event-selection. But the low number of events remaining makes it difficult to evaluate the radio reconstructions. To increase the number of events, some of the quality cuts are not applied, like the anti-bias cuts are not necessary for RD-FD comparison. The modified list of quality cuts used for this analysis is shown in table A.5. Cuts which are not used have the # symbol in front of their name. Table 6.3 contains the results for Coihueco, HEAT and HECO. In addition, the first cut in the list of quality cuts that the event fails is given. Most of the reconstructions (10) fail at the "X_{max} observed in expected field of view" cut. It describes a region in the energy deposition curve which can be smaller than the geometrical field of view taking into account virtual showers [98]. This region is associated with high quality measurements. The "depth track length", a cut concerning the reconstruction quality follows with the removal of 8 events. For events 3 - 18 and 20 no MieDatabase, which is an estimation for the atmospheric conditions, is available. For events 5 - 8, 10 - 12, 15 - 18 and 20, no cloud data is available. Because these two cuts remove a large portion of the events,

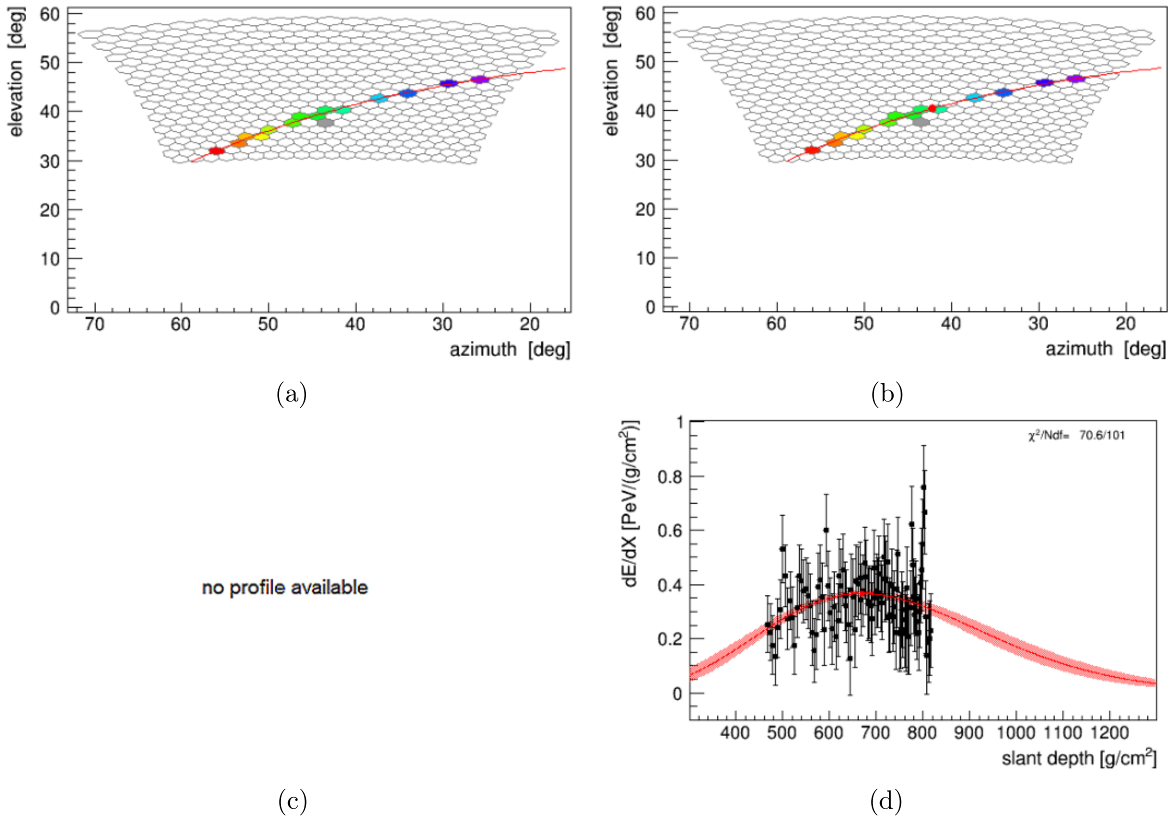


Figure 6.1: Comparison of the FD reconstruction for the v9r5 and the v10r0 datasets for the corresponding Radio event 175530 of run 100759. Although the light curves look identical for the v9r5 dataset (a) and the v10r0 dataset (b), the X_{max} profile for HEAT is not generated in the v9r5 dataset (c) but for the v10r0 dataset (d).

the MieDatabase and cloud cuts are left out for all 22 events. It is important to have a good FD X_{max} reconstruction but not perfect atmospherical conditions for comparing the radio reconstruction methods. In this early stage of development, a rough estimation using a small data sample is more important than going too much into detail and therefore not being able to validate the methods at all. In later stages with a larger data sample, the remaining cuts should be applied as well. To conclude, using the analysis based on the standard ADST and the enhanced FD quality cuts, 18 of the 22 events have been selected as high quality FD events which can be used to compare the output of the radio X_{max} analyses with FD.

6.3 Enlarging the FD dataset

For evaluating the different reconstruction methods with a higher statistical significance, it would be desirable to have more than the 18 FD events which pass the quality criteria. Another approach to enlarge this sample is to use a different dataset as input. For this purpose, the updated RdObserver v1.3 dataset, now from January 29 in 2012 until March 03 in 2015, is

Table 6.3: Results of the analysis using the advanced ADST cuts for the Auger v10r0 dataset and for different FD sides. The color coding is the same as in table 6.2. The text inside the cell represents the first quality cut the analysis fails.

#	Event	CO	HEAT	HECO
1	100759-175530			
2	100759-303383	depthTrackLength	depthTrackLength	depthTrackLength
3	100787-456636	xMaxObsInExpFOV	xMaxObsInExpFOV	
4	100795-171250		depthTrackLength	
5	100823-46784	xMaxObsInExpFOV		
6	100823-59774	xMaxObsInExpFOV		
7	100823-68115		xMaxObsInExpFOV	
8	100838-57641			xMaxObsInExpFOV
9	100865-302837		xMaxObsInExpFOV	
10	100869-455911			
11	100870-357969		maxCoreTankDist	xMaxObsInExpFOV
12	100870-85032	depthTrackLength	maxDepthHole	
13	100871-2702	xMaxObsInExpFOV		
14	100885-83147	xMaxObsInExpFOV		
15	100887-154817	depthTrackLength		
16	100896-575539			
17	100898-104598		maxDepthHole	
18	100909-67806	depthTrackLength	depthTrackLength	
19	100951-44027			
20	101005-57928			HeCoHasUpTime
21	101019-35126			
22	101024-113090			

used. The input files for the RdObserver are xrad files with the updated IoAuger input format described in section 5.6 and the analysis of the quality criteria is done afterwards with the produced ADST files. The results are listed in table 6.4.

Table 6.4: Results of the analysis with the complete RDObserver v1.3 dataset with additional quality cuts.

Analysis stage	# of events
Reconstructable RD Events	5782
Apply FD reconstruction	547
Enhanced FD quality cuts	137
Basic RD quality cuts	29

In total, 5782 radio events have been reconstructed in coincidence with the surface detectors. By requesting a successful FD reconstruction, 547 SD-FD-RD events are left over. Using the extended FD quality cuts introduced in section 6.2, 137 events survive. To have comparable results with the previous analysis, the 'hasMidatabase' cut and also the 'cloud' cut are not used. Another important variable for the RD-RD comparisons is the quality of the RD event. Therefore, basic RD quality criteria cutting on the multiplicity of the RD event and the geometry of the EAS are applied in addition. The radio event needs at least 5 stations with signals and a reconstructed zenith angle of less than 55°. In total, only 29 events suitable for

Table 6.5: Official list of radio events which have a min. multiplicity of at least 5 stations and a zenith angle less than 55°. All of these events pass the enhanced FD quality cuts.

Run.EventNr.	**Good FD eye**	**RD Multiplicity**	**RD Zenith [deg]**
100178-89268	HECO	5	29.78
100200-35175	CO, HECO	5	14.16
100231-2341083	CO, HECO	5	32.50
100650-3515	CO	5	39.513
100699-842012	CO, HECO	5	26.13
100759-175530	HE, HECO	6	41.22
100787-456636	HECO	5	33.5
100795-171250	CO, HECO	5	54.82
100796-239412	CO	5	47.63
100822-210278	HECO	7	43.39
100823-46784	HE, HECO	6	37.1
100823-59774	HE, HECO	5	42.80
100838-57641	CO	5	40.37
100840-53509	CO, HECO	9	42.39
100869-455911	CO, HE, HECO	6	37.43
100873-74624	HECO	5	50.06
100874-190861	CO, HECO	8	20.17
100879-95847	HE, HECO	5	45.68
100885-83147	HE, HECO	6	42.4
100886-48074	HECO	7	23.61
100887-184965	HE, HECO	6	35.90
100890-273541	HECO	6	32.97
100895-839145	HECO	11	32.36
100907-96472	HE, HECO	6	41.77
100911-55079	HE, HECO	7	47.19
100919-131281	CO	6	46.59
100969-81860	HE	5	52.74
101005-57928	CO	6	39.62
101006-76110	HE	5	45.1

the $X_{\max}$ discussion were found. The results showing the RD run and event number, the FD eyes which have passed the FD cuts, the RD multiplicity and the reconstructed zenith angle are listed in table 6.5.

6.4 RD-RD comparison list

The results of the previous analyses have shown that in the complete RDObserver v1.3 dataset only 29 events fulfil the requirements to be included in the high quality FD-RD data sample. This is still not enough to select radio events for an $X_{\max}$-comparison among the different radio methods. Based on this result, as a first approach the different RD reconstruction methods should be compared only using a data sample of reconstructable RD events which pass the basic RD quality cuts. For this purpose, the full RDObserver v1.3 dataset from 2012-01-29 until 2015-03-02 is again used. By demanding zenith angles below 55°, 4535 of the 5782 in

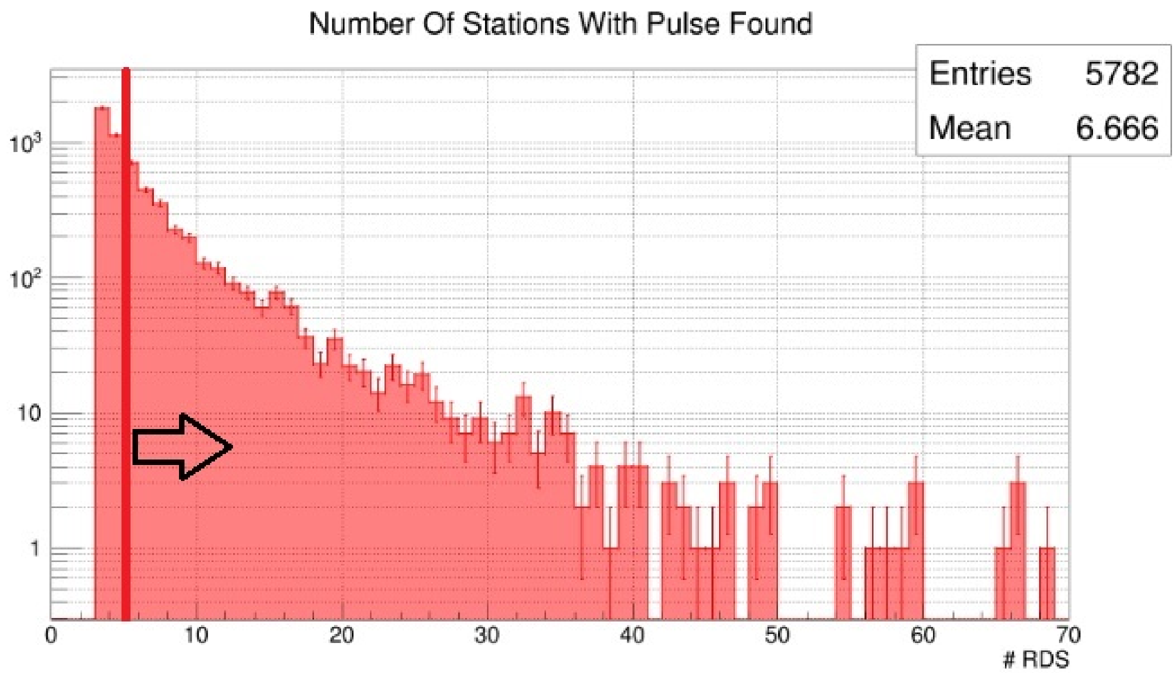


Figure 6.2: Multiplicity distribution of the RDObserver v1.3 dataset. The selection criterion of at least 5 signal stations is indicated.

total are left over. After the requirement of having at least 5 radio stations with a measured signal, 2313 events survive. If both cuts are combined, 1846 events remain. The multiplicity and zenith distributions of the v1.3 dataset plotted with the selection criteria are shown in Fig. 6.2 and Fig. 6.3.

To have a handy but still statistically relevant dataset, a list consisting of 150 of these RD events has been specified acting as a benchmark for comparisons. The first 25 events of each month from July - December 2014 which fulfill these criteria have been chosen. During this time period, no big changes were applied to the software or hardware of the AERA stations. Different months have been used to prevent being biased by seasonal effects. The list of events is attached in table A.6. The first column consists of the run and event number, followed by the zenith angle, the theta angle and finally the multiplicity (i.e., the number of signal stations which are not rejected in Offline). The list defined by this work has been used in the AERA analysis group as a standard to compare the different methods Once this validation is done, the RD results can be compared with the FD results based on the list in section 6.3. The size of this high quality FD-RD dataset will constantly increase while the whole AERA dataset gets larger due to the continuous data taking. Having a larger high quality FD-RD sample is important to increase the significance of the analysis.

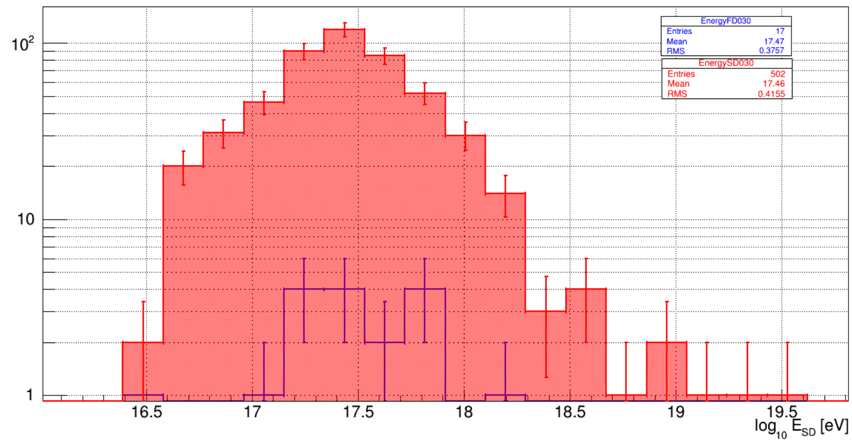
Figure 6.3: Zenith distribution of the RDObserver v1.3 dataset. The selection criteria of less than 55° is indicated.

7 Improvements of the Radio analyses based on FD data

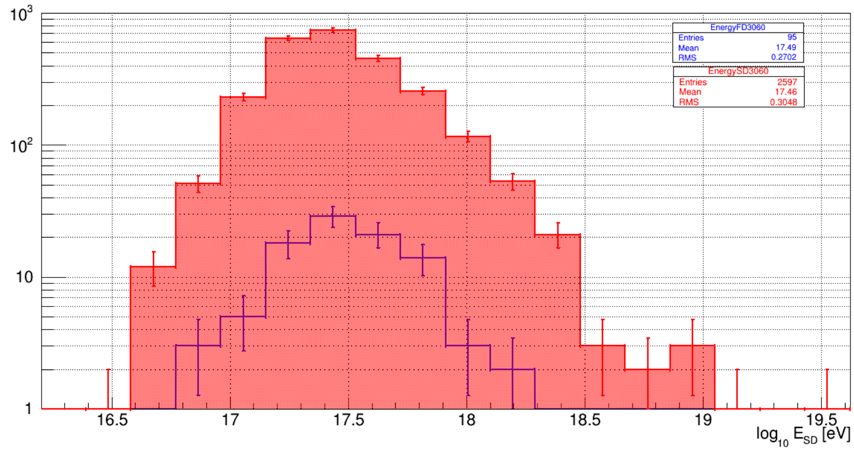
Optimizing the data handling of AERA, shown in chapter 5, was a necessary step to allow for reconstructing the radio data more effectively. To get a larger data sample, consisting of coincident RD and FD events, as described in chapter 6, one could simply wait for more events to be detected. However, waiting for more events isn't the only way to get better results, there is also room for improvements in the radio analyses themselves. In the current state, the *RdObserver* takes values reconstructed by SD as input for the Radio reconstruction. For example, since the SD reconstruction is already well established, the shower direction for Radio is set initially to the shower direction evaluated by SD. The same applies for the SD core position. Based on these starting conditions, the Radio modules reconstruct the radio event. After the SD reconstruction and after all radio modules are applied, FD is only used optionally in the end. Thus, for all coincident RD/FD events, a successful SD reconstruction is also necessary, corresponding to the use of only golden hybrids. More events could be recovered by also directly taking the FD reconstruction instead of SD for Radio into account. In this chapter, adaptations for the *RdObserver* in Offline to use input values based on the FD reconstruction alone, along with first results are shown.

7.1 Radio reconstruction based on SD and FD

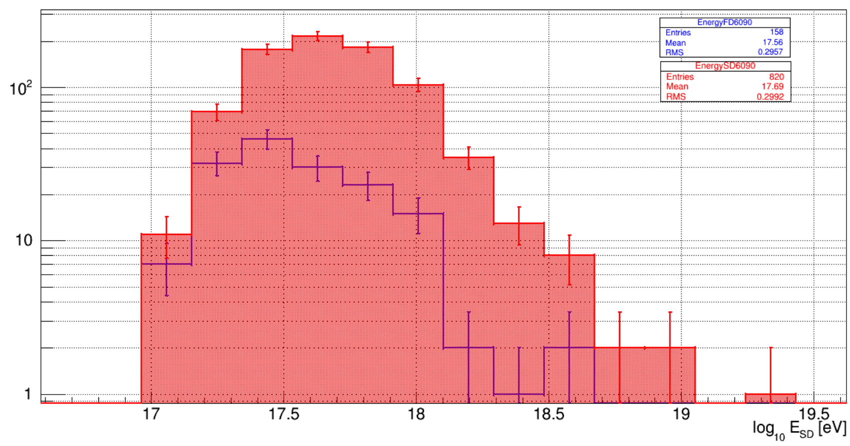
Because of the SD energy threshold and the efficiency of the infill, especially towards higher shower axis zenith angles, the addition of the FD reconstruction as RD input could lead to an increase of reconstructed RD events. In order to test this hypothesis, the RD-Observer v1.3 production for AERA phase II (2013/06 - 2014/11), containing 3919 reconstructed SD-RD events, was used. In this dataset, 390 FD events are measured in coincidence with Radio and SD. By looking at the energy distribution of SD and FD events, the efficiency of the reconstructions is evaluated. The analysis is done for 3 different zenith angle bins $0 - 30°$, $30 - 60°$ and $60 - 90°$ and the distributions are fit with a Gaussian to get a rough estimate of the maximum, shown in Fig. 7.1a, Fig. 7.1b and Fig. 7.1c.



(a)



(b)



(c)

Figure 7.1: Energy distribution of SD and FD events with zenith angles ranging between 0° and 30° (a), between 30° and 60° (b) and between 60° and 90° (c).

In table 7.1, the results of the analysis are shown. The number of FD and SD events, the standard deviations of the fit and the RMS representing the uncertainties are shown for each zenith angle.

Table 7.1: Energy distribution of the SD and FD reconstructions for different zenith angle bins.

Zenith angle	Events SD	Events FD	**SD energy [$\log_{10}$ eV]**	**FD energy [$\log_{10}$ eV]**
0° - 30°	502	17	17.46±0.42	17.47±0.38
30° - 60°	2597	95	17.46±0.30	17.49±0.27
60° - 90°	820	158	17.69±0.30	17.56±0.30

For higher zenith angles, the maximum of the FD energy distribution is lower than the maximum of the SD distribution. Hence, taking FD values as input for the RD reconstruction we should gain some events, especially at lower energies and higher zenith angles, which might normally fail the SD reconstruction since all reconstructed RD events also have SD.

7.2 FD based technical adjustments in the RD-reconstruction

To develop a standard reconstruction sequence where the FD measurement is used instead of the SD as input for the RD reconstruction, several changes to Offline had to be made. The technical changes concerning the source code of the radio modules used in the *RdObserver* and necessary timing adjustments are explained in the following subsections. SD will not be completely removed from the module sequence, since as a first step, the FD hybrid reconstruction with a hottest SD station serving as lever arm for the FD timing fit will be used for radio. In this case, no full SD reconstruction is required. Using FD mono as input for radio will be discussed shortly in section 7.3.2.

7.2.1 Integrating FD into the RD reconstruction chain

As the first step, the new option to choose FD as input had to be included into the RD reconstruction modules. Some of these modules already contained structures enabling the user to choose between SD and Monte Carlo as input. In this case, the FD integration was unproblematic, as the adaption only involved correctly assigning the corresponding FD reconstructed values in a new control statement. The more problematic modules were those containing only hard-coded SD structures. All these modules were restructured by adding a new and flexible coding design in which FD or SD can be set as input parameter which requires no recompiling of the code.

As a second step, FD had to be integrated as a new option into the XML-cards. Existing XML-flags like 'used geometry', 'used core position' or 'used axis', which only offered the option 'SD' or 'Monte Carlo', were extended with the new option 'FD'. Since every FD telescope has its own reconstruction sequence inside Offline, a new input option, called 'used eye' has been integrated. As default, eye number 6 is used, which correspond to the hybrid HECO reconstruction. All of these changes were also propagated into the RD-Observer bootstrap file. With the bootstrap file, the standard input parameters of the Offline modules can be overwritten in a central place.

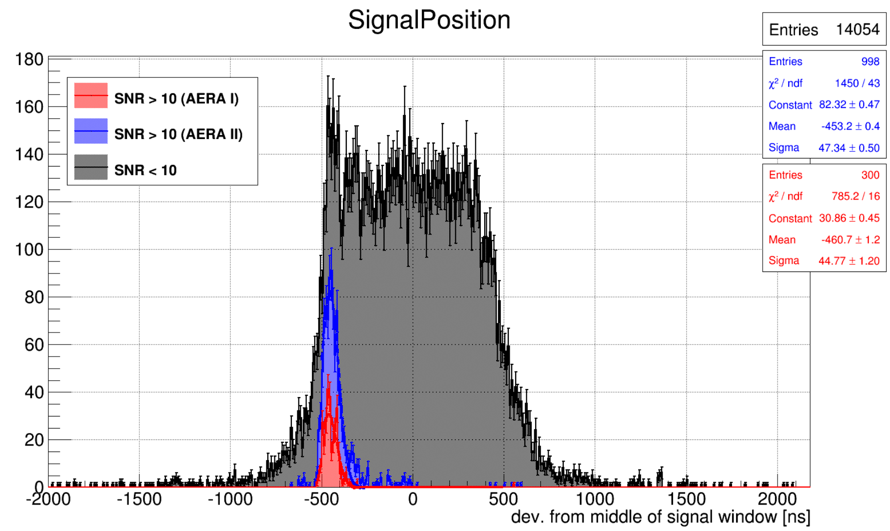
The last step concerns the module sequence of the Radio reconstruction. In the current state of the RD-Observer v1.3, the reconstruction is done in the following order: SD, followed by RD and then, as a last resort, optionally the FD reconstruction. This has been changed in a special version of the RD-Observer and the new module sequence is given in appendix A.3.3. To activate this FD based version, both files "ModuleSequence.xml" and "bootstrap.xml" located inside the RD-Observer directory have to be replaced by their counterparts "ModuleSequence_FD.xml" and "bootstrap_FD.xml" found in the same directory.

To speed up the reconstruction for the FD case, the 'RdEventPreSelector' module, used at the beginning of the sequence, has been modified. If FD has not triggered, then there is also no FD-data available. Consequently, in case FD is selected as input, the event is skipped when no FD-trigger is given. The trigger state is checked by looking at the event trigger flag inside the xrad-file at the start of the event reconstruction.

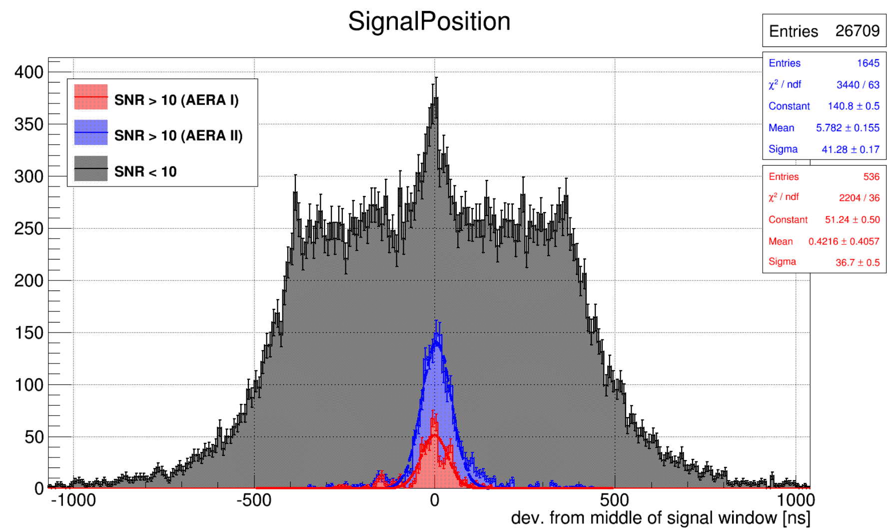
7.2.2 Adjusting the timing window

It is essential for the correct reconstructing of the RD data to know the exact timing information of the event, i.e., the arrival time of the signal at each RDS, to find the signal inside the trace. In case of being externally triggered by SD or FD, the Radio DAQ receives the trigger information from SD or FD, requests the data from all RDS and builds the event containing the trigger timing information. Shower to shower fluctuations, signal propagation times between stations and the DAQ, internal delays inside the electronics as well as reconstruction uncertainties all have to be taken into account later in the analysis. This is done by defining a *search window* around the expected signal position during the RdObserver analysis. This search window includes the actual signal window due to the length of the radio signal itself and is enlarged to account for the uncertainties mentioned above. If the search window is too large, the probability to select a high noise pulse instead of the EAS signal increases. If the search window is too strict, some events, for which the pulse is outside this window, are not reconstructed at all. As a good compromise, the RdObserver uses the reconstructed SD information for each event and estimates the expected radio signal time inside the trace of each RDS based on the SD input. The *signal window* is set to -100 ns and +150 ns relative to this position. This is large compared to the length of the EAS signal. In addition, the *signal window* is enlarged to the *search window* by including some other relevant factors. These extra factors consist of a phenomenological formula resulting from Monte Carlo simulations which describe a systematic offset between the reconstructed SD core position and the one simulated, a shift due to an offset between the SD and RD GPS timestamps of approximately $\Delta t_{\mathrm{SD,RD}} \approx 610\,\mathrm{ns}$, as well as the error propagation of the SD reconstruction uncertainties.

The same kind of correction has to be determined for the FD as input. As a first analysis, the RD search window is set to $\pm 2000\,\mathrm{ns}$ around the expected signal arrival time of each RDS and no offset is taken into account to evaluate the general signal position. In Fig. 7.2a, the signal distribution for the RdObserver v1.3 dataset where FD has been set as the input for RD is shown. The reconstructed radio signal position is compared to the signal position expected from the FD. The x-axis corresponds to the deviation from the middle of the signal window which is the estimated signal position. For a successful reconstruction, an SNR value larger



(a)



(b)

Figure 7.2: Distribution of the deviation of the reconstructed signal positions relative to the signal position expected from the FD without (a) and with (b) a corrected offset and their fit results. A successful reconstruction requires an SNR value $>$ 10. The grey distribution contains all stations which have not been reconstructed, the red distribution all AERA phase I stations and the blue distribution all AERA phase II stations which have been reconstructed. Without correction, a shift in the distributions is clearly visible. With the correction, the distributions are centered around the expected signal position.

than 10 is required. The grey distribution contains all stations with an SNR smaller than 10, the red distribution is for AERA phase I stations and the blue distribution is for AERA phase II stations which have been reconstructed. In addition, the distributions are fit with a Gaussian.

By looking at the results of the fit, a shift in the distributions of the reconstructed signal time relative to the expected signal time is visible. Both AERA phase I and II distributions are shifted consistantly by approximately 460 ns. Within the uncertainty, the rounded value has been chosen. Figure 7.2b shows the results of the same analysis with the offset taken into account in the RdObserver XML card.

In the analysis with corrected offset, the distributions are centered around the signal position predicted by the FD. An excess in the distribution consisting of stations with SNR less than 10 near the center is visible. This suggests that more signal stations could be gained by lowering the SNR threshold however this would increase the probability of reconstructing noise pulses. Thus, this topic has to be studied in more detail.

To determine the optimal width of the RD search window based on FD data, the same principles as for SD have to be adopted. The formula which is used in the *RdEventInitializer* to estimate the start time of the search window from the SD is

$$t_{\text{start}} = t_{\text{exp}} - \sigma + t_{\text{signal,start}} - t_{\text{inc}} \tag{7.1}$$

and the stop time is

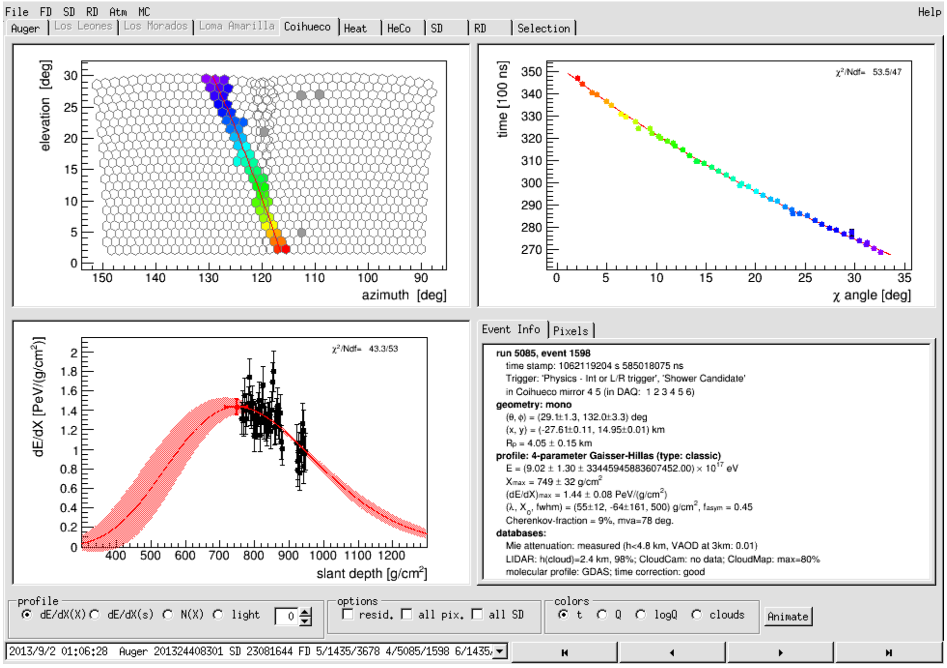
$$t_{\text{stop}} = t_{\text{exp}} + \sigma + t_{\text{signal,stop}} + t_{\text{inc}} \tag{7.2}$$

where t_{exp} is the expected signal time based on the SD reconstruction, σ is its propagated error, $t_{\text{signal,start}}$ is the lower limit of the signal search window, $t_{\text{signal,stop}}$ is the upper limit of the signal search window, and t_{inc} is the outcome of the MC based formula which depends on the reconstructed zenith angle and describes SD core position uncertainties. For the FD case, some deductions have to be made. Since no compatible formula is known for the FD core position, t_{inc} is set to the maximum value obtained by the SD formula as a conservative approach. σ describes the error propagation for the projected vector from the core position to the RDS onto the shower axis. For this calculation, the fit uncertainties are required. In the FD case, the values *GetAxisCorrelation* and *CoreTimeError* are not calculated during the FD plane fit and are therefore not available in the FD reconstruction chain. The best estimator for the uncertainty of the FD shower geometry reconstruction is the one on the angle of the shower axis in the shower plane, χ_0.

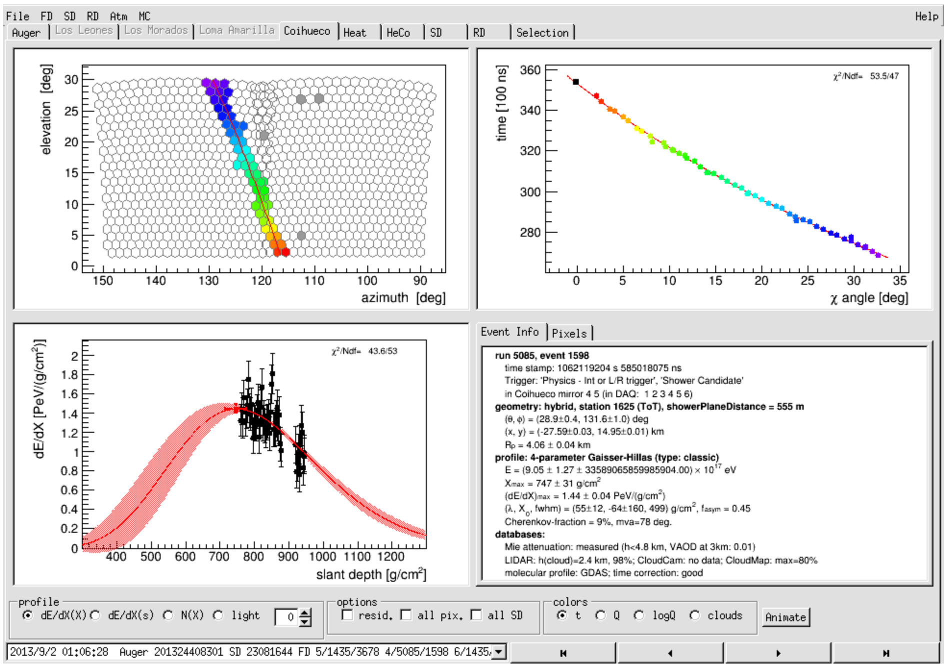
With these assumptions, the FD based signal search window is in general overestimated. This approach is chosen since the objective of this analysis is to study how much the number of FD-RD events can be increased, so the increase in the efficiency. Optimizing the purity of the dataset which can be achieved, (among other things), by reducing the number of noise pulses incorrectly reconstructed as EAS signals could be a future topic.

7.2.3 Verifying the implementation

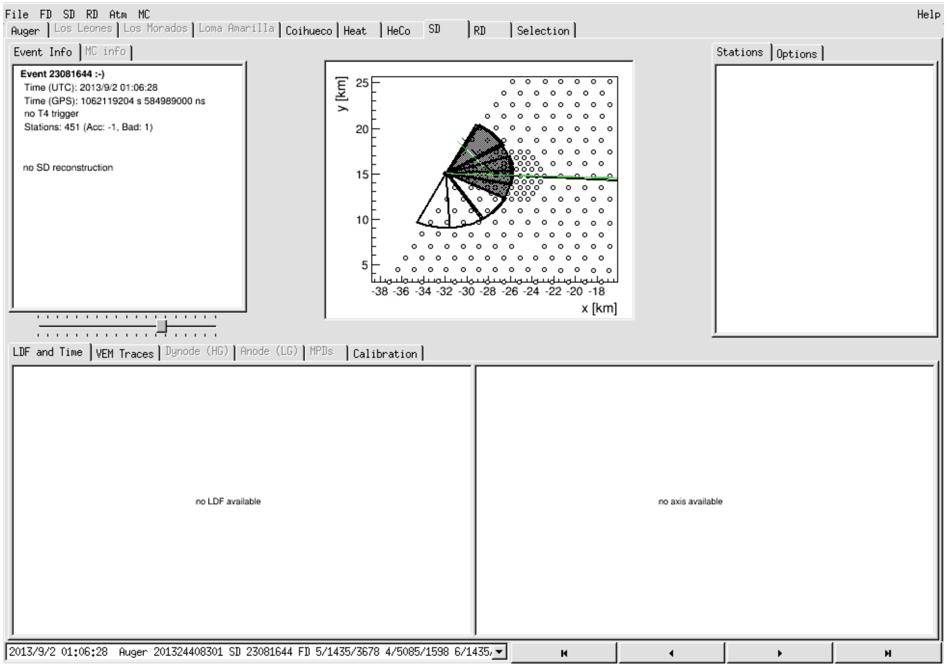
In order to check if the implementations described above work as expected, the different reconstruction chains are compared for a randomly chosen event. This method reduces the available event information to compare the reconstruction with the full information that has an SD reconstruction. Figure 7.3 shows the output of the RD-Observer displayed with the Offline event browser for the different reconstructions.



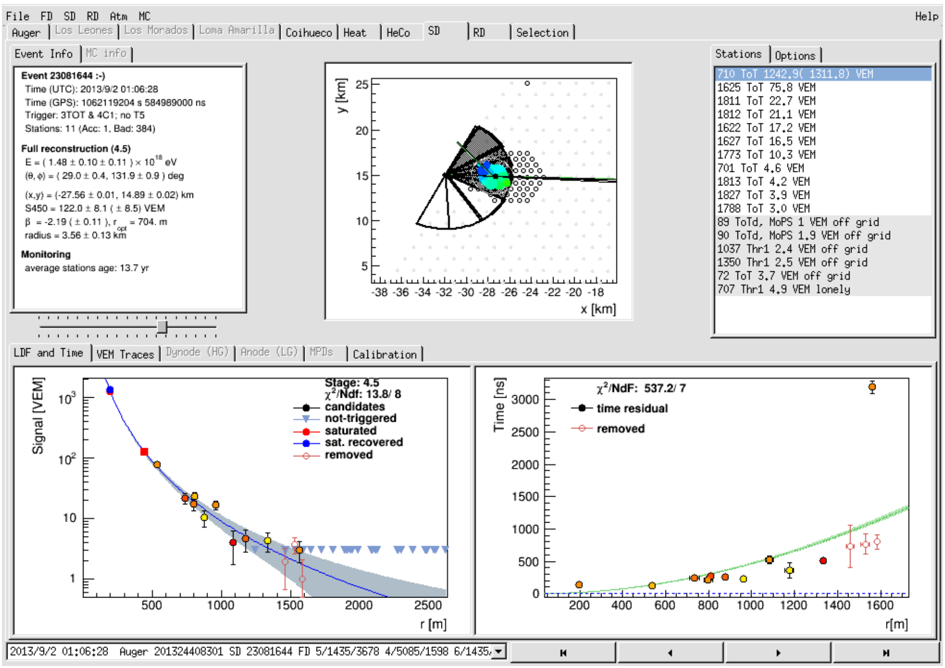
(a)



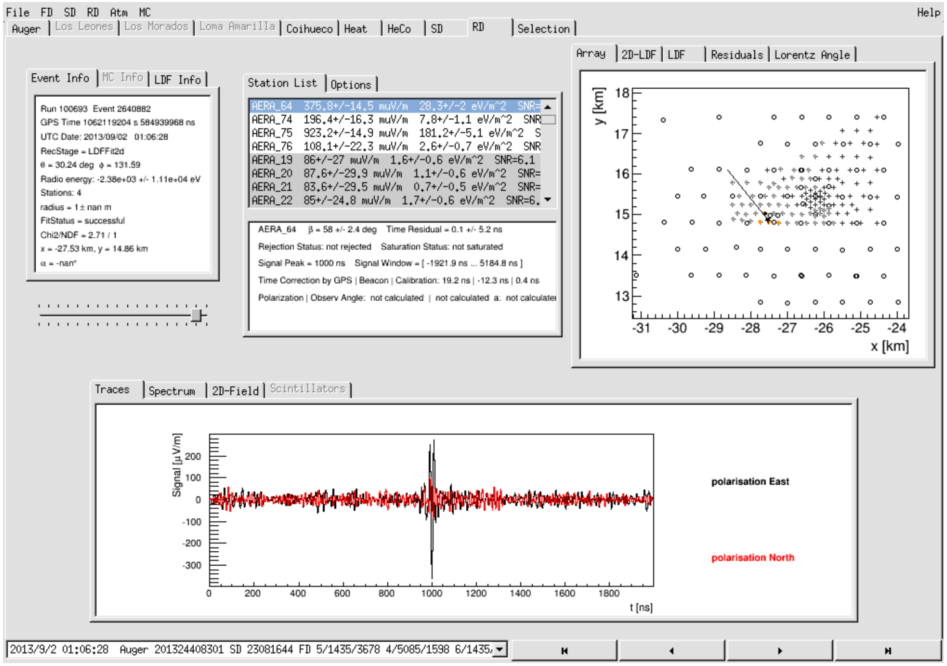
(b)



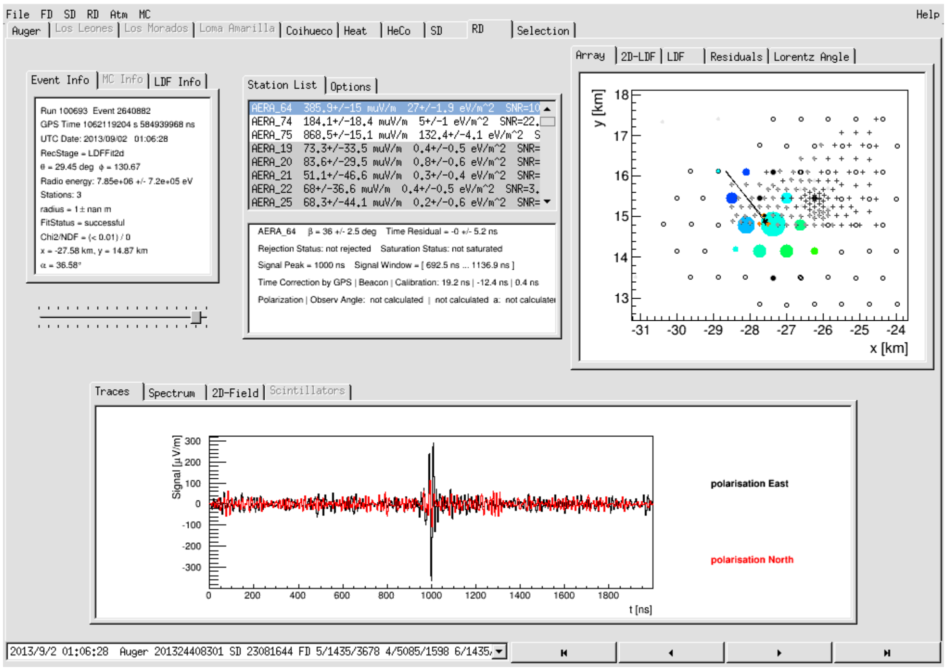
(c)



(d)



(e)



(f)

Figure 7.3: Comparing event ID = 100693-2640882 (randomly chosen) reconstructed with the different reconstruction chains of the RD-Observer. Shown is the output of the event browser with FD used as input in (a), (c), (f) and SD used as input (b), (d) and (e) for RD. In (a) and (b) the FD part of the reconstruction is shown, in (c) and (d) the SD part and in (e) and (f) the FD part. It is clearly visible, that the SD reconstruction is missing in (c). Both reconstructions show consistent results which are summarized in table 7.2.

The comparison of both SD-FD-RD and SD-RD-FD reconstruction chains is shown in table 7.2. It contains the reconstructed azimuth and zenith angles and the core positions with their uncertainties for different cases: SD describes the results for the SD reconstruction, FD_{SD} lists the FD results obtained by using the complete SD reconstruction including the SD plane fit and $FD_{noSDrec}$ contains the FD results if SD is only used to get the coordinates of the hottest SD station but omitting the plane fit reconstruction. RD_{SD} lists the results of the reconstructed shower geometry for the standard SD-RD-FD reconstruction chain and RD_{FD} the newly implemented order as defined in $FD_{noSDrec}$. In the RD reconstruction in general, the uncertainties are not calculated. They could be gained via detailed Monte Carlo simulations which are not available and thus beyond the scope of the analysis described in this chapter.

Since the results of the RD_{SD} and RD_{FD} case are well within the given uncertainties, FD can be used as input for RD for further analysis.

Table 7.2: Summary of the reconstructed azimuth and zenith angles and the core positions with their uncertainties for the different reconstruction methods: The SD results (SD), SD used as input for FD (FD_{SD}), only the hottest SD station is used for FD ($FD_{noSDrec}$), SD used as input for RD (RD_{SD}) and FD as defined in ($FD_{noSDrec}$) used as input for RD (RD_{FD}).

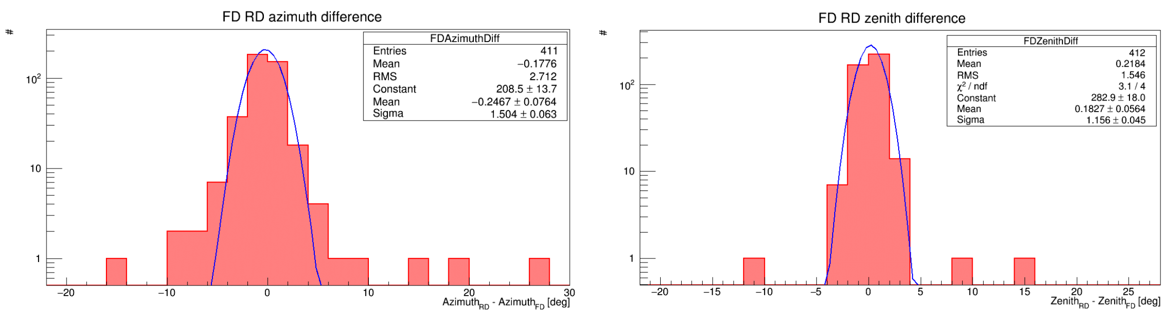
	axis azimuth [deg]	**axis zenith [deg]**	**core x [km]**	**core y [km]**
SD	131.9±0.9	29.0±0.4	-27.56±0.01	14.89±0.02
FD_{SD}	131.6±1.0	28.9±0.4	-27.59±0.03	14.95±0.01
$FD_{noSDrec}$	132.0±3.3	29.1±1.3	-27.61±0.11	14.95±0.01
RD_{FD}	131.59	30.24	-27.53	14.86
RD_{SD}	130.67	29.45	-27.58	14.87

7.3 Comparison of datasets based on SD and FD reconstruction

With the adaptations described in the previous sections, the RD reconstructions based on the FD and SD inputs can be compared. The analysis is done on the PLEIADES cluster at the university of Wuppertal [100] (a Tier-2 center inside the world wide LHC computing grid). The complete dataset from AERA phase II (2013/06 - 2014/11) is available as daily xrad files on the cluster. For a faster reconstruction, days without FD data are removed from the dataset resulting in 289 remaining days. To analyze the dataset in parallel, each day is submitted as one independent job to the cluster, resulting in around 5 hours of real time calculations in total. If the analysis would run on a single core machine, it would take several years to process all the data.

7.3.1 Results

Using the standard RdObserver SD-RD-FD module sequence, 390 events with full SD-RD-FD information are reconstructed. By applying the alternative SD-FD-RD module chain explained



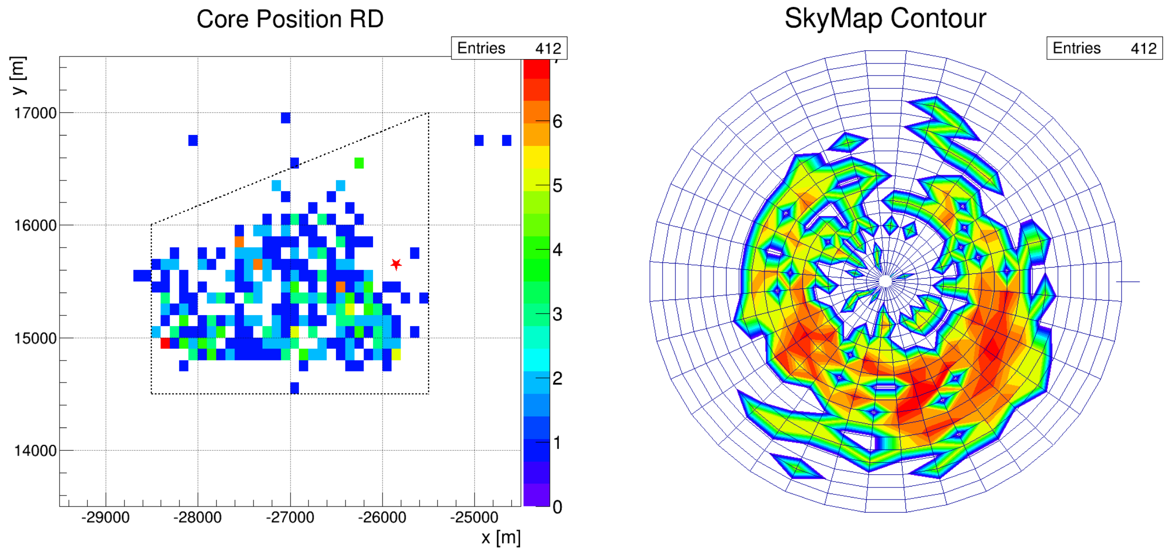
(a) (b)

Figure 7.4: Distribution of differences between the azimuth (a) and the zenith angle (b) of the air shower reconstructed by RD and by FD fitted with a Gaussian. The azimuth angles reconstructed by RD are systematically lower by $(0.25 \pm 0.08)^\circ$ than the values by FD, whereas the reconstructed zenith angles are systematically higher by $(0.18 \pm 0.06)^\circ$.

in section 7.2, 412 events could be reconstructed. In Fig. 7.4 - Fig. 7.6 general properties of those 412 events are shown. In Fig. 7.4, the distribution of differences between the zenith and the azimuth angle of the air shower reconstructed by RD and by FD for each event fitted with a Gaussian is shown. The azimuth angle reconstructed with RD is systematically lower by $(0.25 \pm 0.08)^\circ$ than by FD, while the zenith angle reconstructed with RD is systematically higher by $(0.18 \pm 0.06)^\circ$. Figure 7.5 contains the reconstructed core positions by RD and a contour plot of the sky map. An excess in the south-east direction (0 starting at east and going counter-clockwise) is visible which is compatible with the theory of radio emission described in section 2.5. Figure 7.6 shows the $X_{\max}$ distributions reconstructed by FD for the telescopes Coihueco, HEAT and HECO and the combined distribution for all telescopes. There is a statistically insignificant peak inside the combined distribution at $630\,\mathrm{g/cm^2}$ which is also visible in the HECO distribution. It might be a feature which should be analysed in more detail when more data is available.

The difference in the results of both reconstructions is comprised as follows: 30 events of the SD-RD-FD result data set are not present in the new reconstruction. 6 of them don't pass the signal to noise ratio cut, 4 events don't pass the angle cut and one event the core distance cut. The FD hybrid reconstruction is not passed by 9 events. On the other side, 52 new events have been reconstructed using the new FD based module sequence. A list including the run number and the event number for the 52 additional events and the 30 missing events is shown in table 7.3. In total the event sample is increased by 22 events.

In order to understand why some of the events don't pass the radio reconstruction by using the FD based RdObserver, the following example is considered: As seen in table 7.2, the reconstructed shower axis differs slightly for the SD and FD reconstruction. The gain pattern of the radio antennas and thus the amplification is direction sensitive. Already small differences in the shower axis can therefore be responsible for a different unfolding of the measured radio-signal trace. This can lead to an event not passing the "minimum number of present candidate stations" RD reconstruction criterion if there are not many stations with a high SNR involved in the event. In Fig. 7.7, such an example is shown when the radio trace differs just because



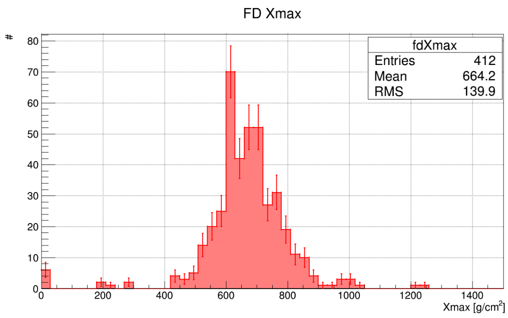
(a) (b)

Figure 7.5: Reconstructed core positions by RD (a) and a contour plot of the sky map (b). In (a), the red star is the position of the CRS and the dotted line is the expected area of the core position due to the geometry in which the RDS are located inside of the AERA array. In (b), an excess in the south-east direction (0 starting at east and going counter-clockwise) is visible which is compatible with the theory of radio emission described in section 2.5.

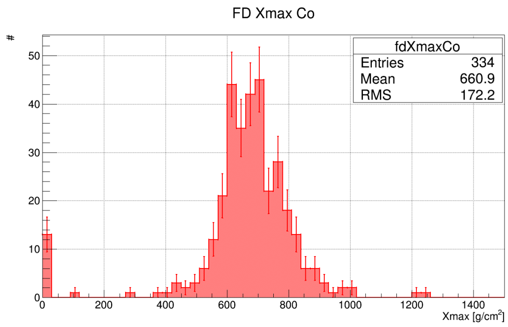
.

of the different shower-axis and detector unfolding. If SD is taken as input, the event is successfully reconstructed. This is not the case for using FD as input because not enough RD signal stations are present. For the SD case, the event contains just 3 stations with a reconstructable signal with an SNR larger than 10. One of the stations has an SNR of 10.9. The SNR is calculated by dividing the signal peak value by the mean of the noise level. The mean of the noise level is calculated in a noise search window set at the beginning of the signal trace. For the case of the slightly different reconstructed shower axis by FD, the SNR of this station is just 9.4. Inside the signal window, two clear peaks are visible. The reconstruction algorithm searches for the highest peak in this window. For FD, the peak at 20 750 ns, for SD, the peak at 20 475 ns is taken as the signal peak. Although the peak for FD is higher, the overall noise level in the noise search window is higher than for the SD case resulting in a smaller SNR which removes the RDS from the list of candidate stations.

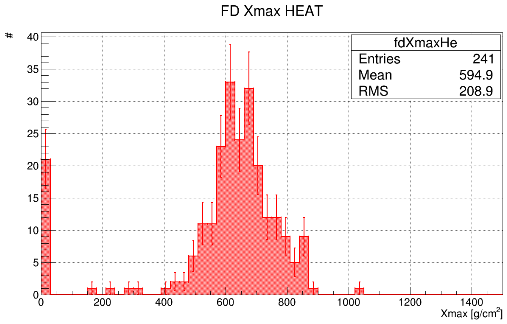
Since 52 events are gained in the FD-RD set and 30 events are lost the SD-RD set in the first iteration of the FD based RdObserver, the following analyses are only slightly significant but should give a first impression about the quality of the additional RdObserver improvements. In Fig. 7.8 - Fig. 7.16, both sets, the FD-RD set (always the left plot) and the SD-RD set (always the right plot) are shown. Figure 7.8 shows the reconstructed core positions. For the FD set, the events are more equally distributed over the expected area defined by the dotted line whereas more events are lost which are located in the north-west region. Figure 7.9 shows the number of events reconstructed per date for the year 2014. The interesting point is the



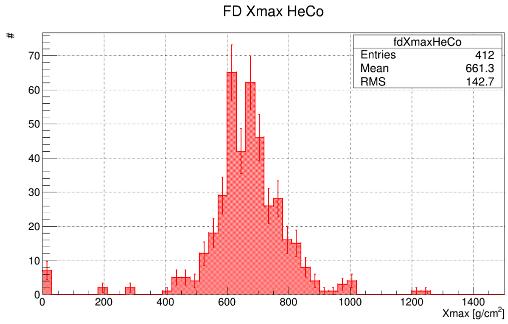
(a)



(b)



(c)



(d)

Figure 7.6: Distribution of the reconstructed FD X_{max} for the combination (a) of Coihueco (b), HEAT (c) and HECO (d). The peak at $630\,\mathrm{g/cm^2}$ visible in (a) and (d) is not yet statistically significant.

hotspot which is contained in the FD set at the end of September 2014 which origin is not clear yet. In Fig. 7.10, the energies of the primary particle reconstructed with FD fitted with a Gaussian are shown. The mean of the distribution in the FD set is a little lower than the mean for the SD set but this is not significant due to the uncertainties. Figure 7.11 shows the number of events reconstructed per hour of the day. Since FD can't be operated during the day, all events have been recorded during the night with a small excess between 4 and 6 a.m. UTC. In Fig. 7.12 the reconstructed azimuth angle is plotted. For the FD set, the excess in the south-east direction is again visible as already seen in the data set consisting of all 412 reconstructed events, while the SD set is more equally distributed. This might also indicate that more events are reconstructed which are oriented towards the Coihueco / HEAT telescopes with the new RdObserver implementation. Figure 7.13 shows the reconstructed zenith angle. An excess to smaller values is visible for the FD set. In Fig. 7.14 the number of signal stations is shown which is compatible for both data sets. Figure 7.15 shows the dependency of the reconstructed X_{max} value and the radio energy of the shower. For the FD set, events with X_{max} smaller than $600\,\mathrm{g/cm^2}$ can be found. In Fig. 7.16 a contour plot of the sky map is shown which is kind of a summary plot for the azimuth and zenith plot. The excess to the south-east is clearly visible for the FD set.

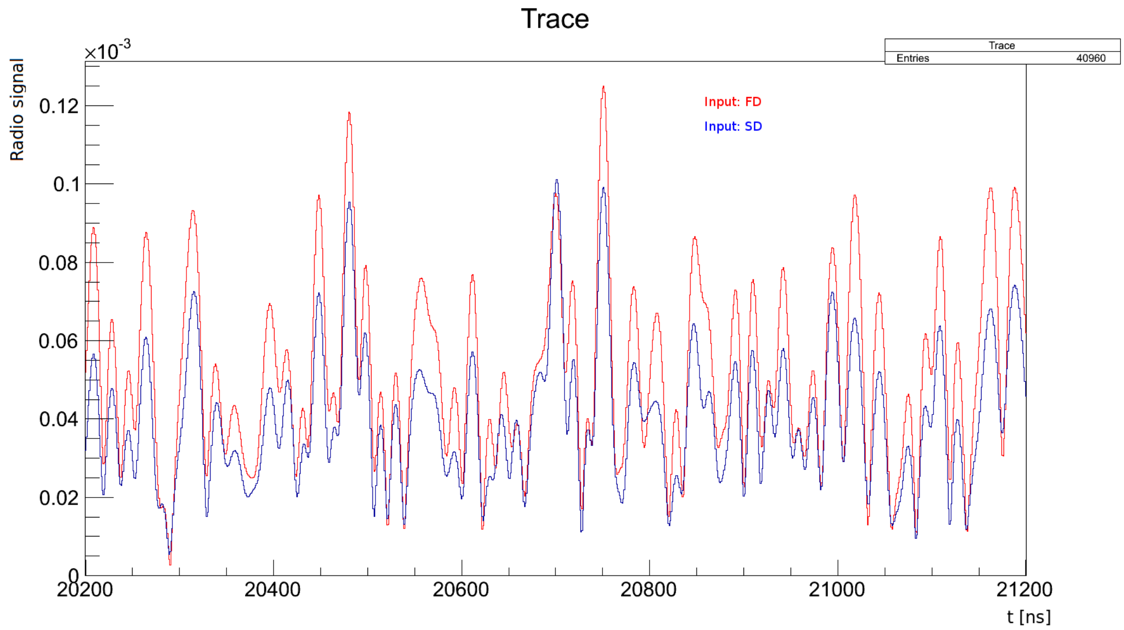
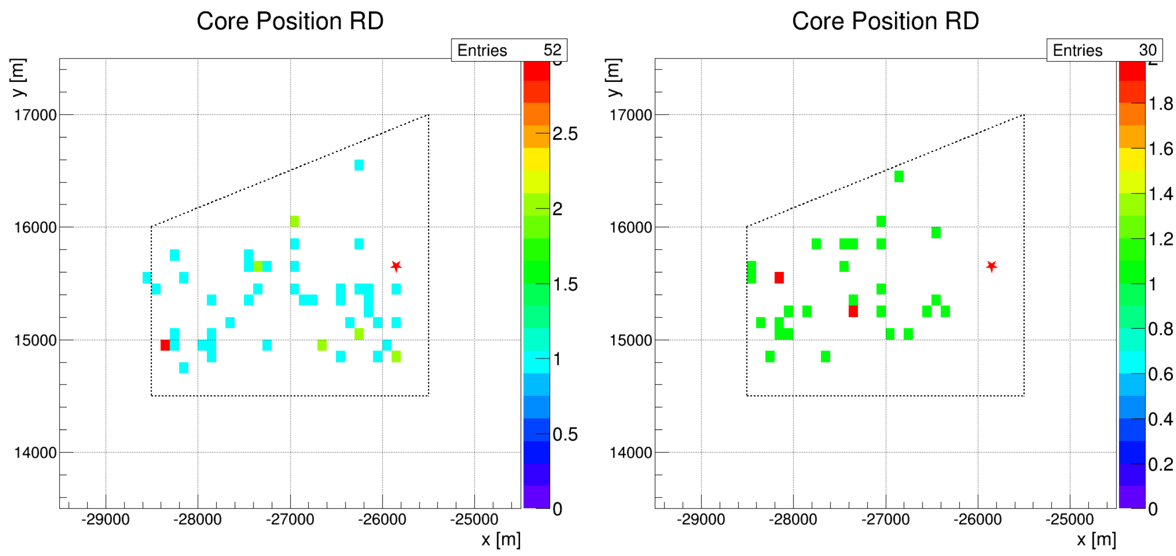


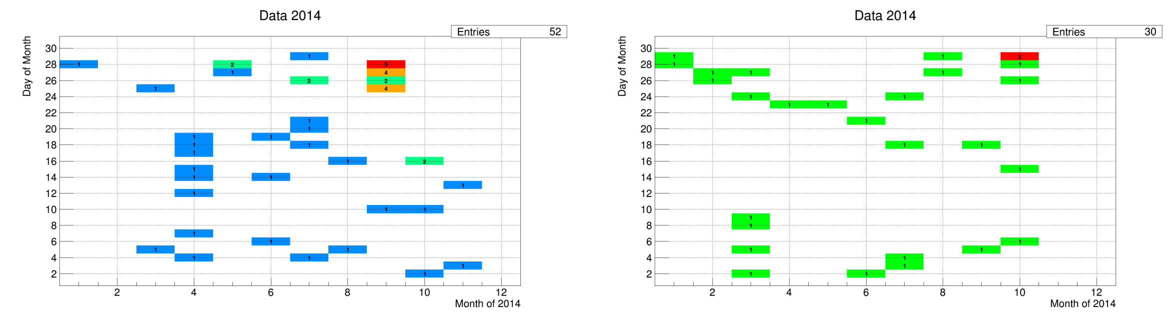
Figure 7.7: Comparison of the signal traces with FD (red) or SD (blue) used as input for RD signal reconstruction for an example event. On the x axis the general trend is the same for both reconstructions, but the absolute values of the measured radio signal shown as the Hilbert envelope on the y axis differ due to the direction sensitive amplification of the RDS. For FD, the peak at 20 750 ns is taken as the signal peak and for SD the peak at 20 475 ns. Even if the other peak is higher, the event is not reconstructed if FD is used as input since also the mean of the noise level calculated in the noise search window is higher.

As already mentioned, the number of events needs to be increased to draw a final conclusion about the characteristics of those events gained by the new FD based RdObserver implementation. This can be achieved by e.g. taking a larger input data set into account, by improving the calculation of the signal search window especially adapted for FD and by tuning the parameters used by the modules of the RdObserver e.g., for the allowed shower geometry. A different solution would be a dynamical module sequence which uses FD as input for RD and SD would only be used as input if the RD reconstruction fails in the first iteration with FD. Additionally, the *UsedEye* variable, which is currently fixed to HECO for each module could be set individually based on the parameters of the FD reconstruction. Already at this point in the analysis it could be shown that more FD-RD events can be gained with a specific geometry by using the adaptations in the FD based RdObserver and the limited data set implemented so far.



(a) (b)

Figure 7.8: Reconstructed core positions of the FD-RD set (left) and the SD-RD set (right). The red star is the position of the CRS and the dotted line is the expected area of the core position due to the geometry in which the RDS are located inside of the AERA array. The core positions in the SD set are more located to the north-east while the events are more equally distributed in the FD set.

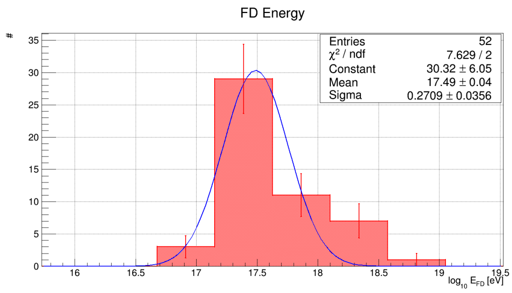


(a) (b)

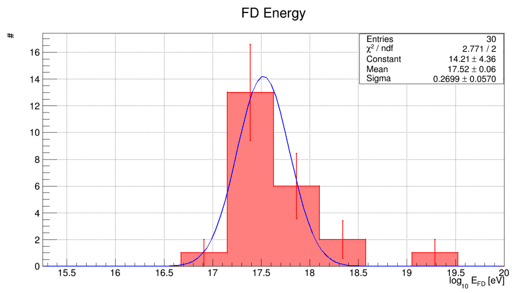
Figure 7.9: Events per time reconstructed of the FD-RD set (left) and the SD-RD set (right). For the FD set, a hotspot at the end of September 2014 is visible which origin is not clear yet.

7.3.2 FD mono reconstruction

As a different approach but only as a short outlook, the FD mono reconstruction instead of FD hybrid is used as input for the RdObserver again with the complete dataset from AERA phase II. With this configuration, RD-FD events might be reconstructed where no SD station has triggered, which can normally be used as hot station for the FD timing fit. To be sure that FD mono is used, SD is removed from the RdObserver reconstruction chain completely.

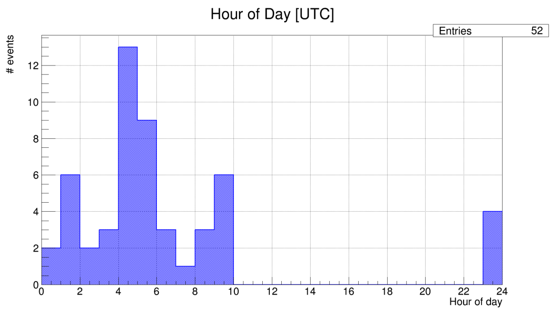


(a)

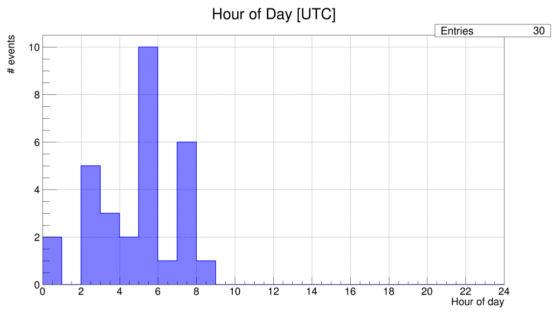


(b)

Figure 7.10: Energy of the primary particle reconstructed by FD and fitted with a Gaussian of the FD-RD set (left) and the SD-RD set (right). The differences between the distributions is inside the range of the reconstruction uncertainties.

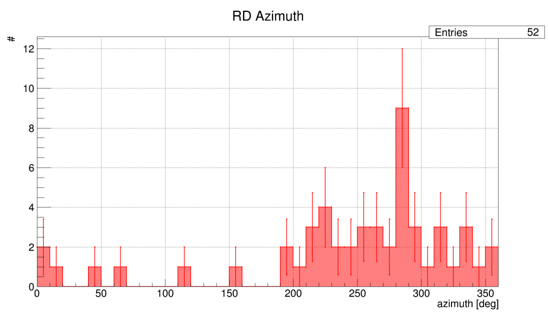


(a)

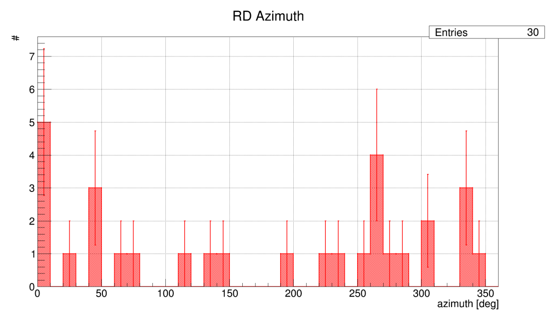


(b)

Figure 7.11: Events per hours reconstructed of the FD-RD set (left) and the SD-RD set (right). The lack of events during the day is due to the operation times of FD. A small excess between 4 and 6 a.m. UTC is visible in both distributions.

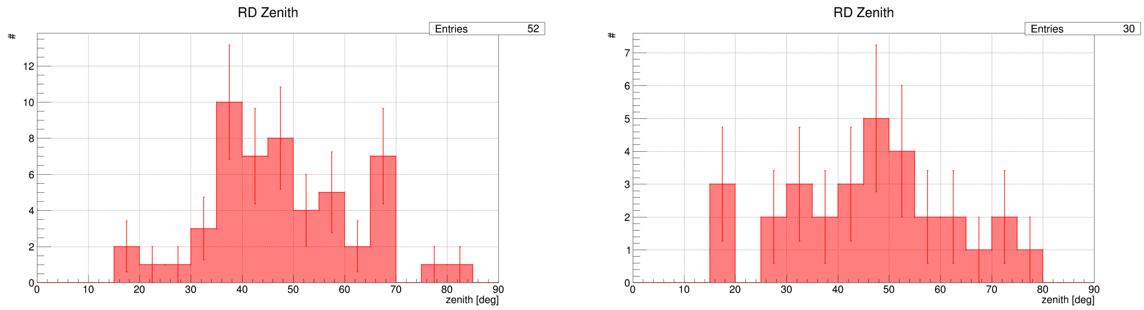


(a)



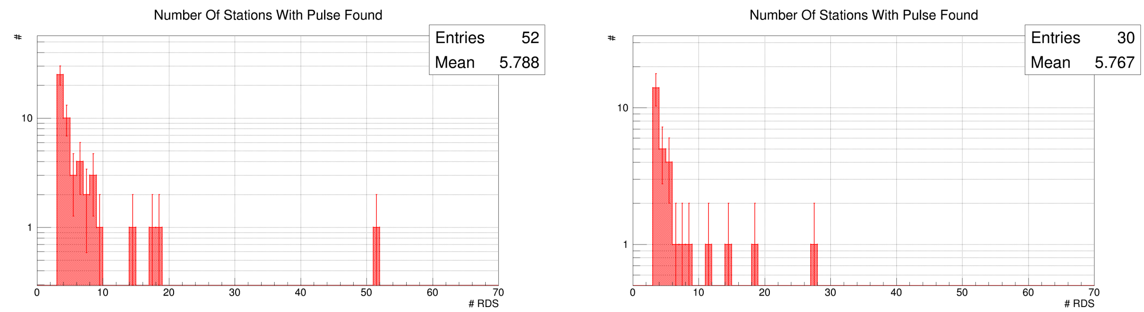
(b)

Figure 7.12: Reconstructed azimuth angles of the FD-RD set (left) and the SD-RD set (right). An excess in the south-east direction is visible for the FD set, indicating that more events are reconstructed which are oriented towards the Coihueco / HEAT telescopes. The excess is also compatible with the theory of radio emission due to the magnetic field of the Earth.



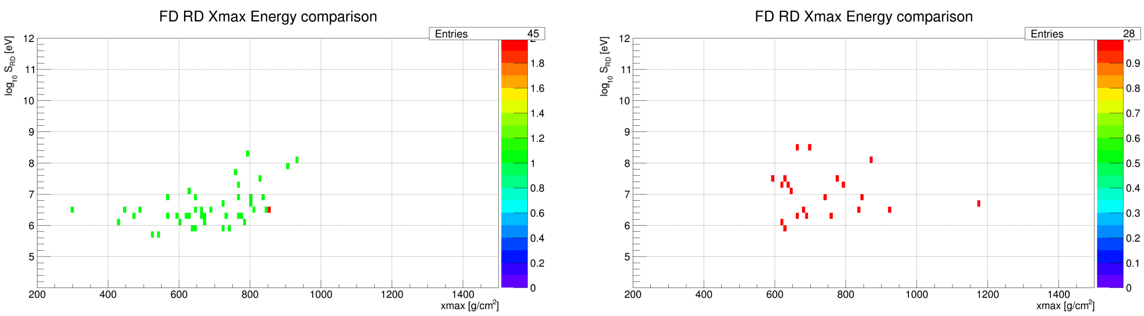
(a) (b)

Figure 7.13: Reconstructed zenith angles of the FD-RD set (left) and the SD-RD set (right).



(a) (b)

Figure 7.14: Number of reconstructed signal stations per event which is compatible for the FD-RD set (left) and the SD-RD set (right).

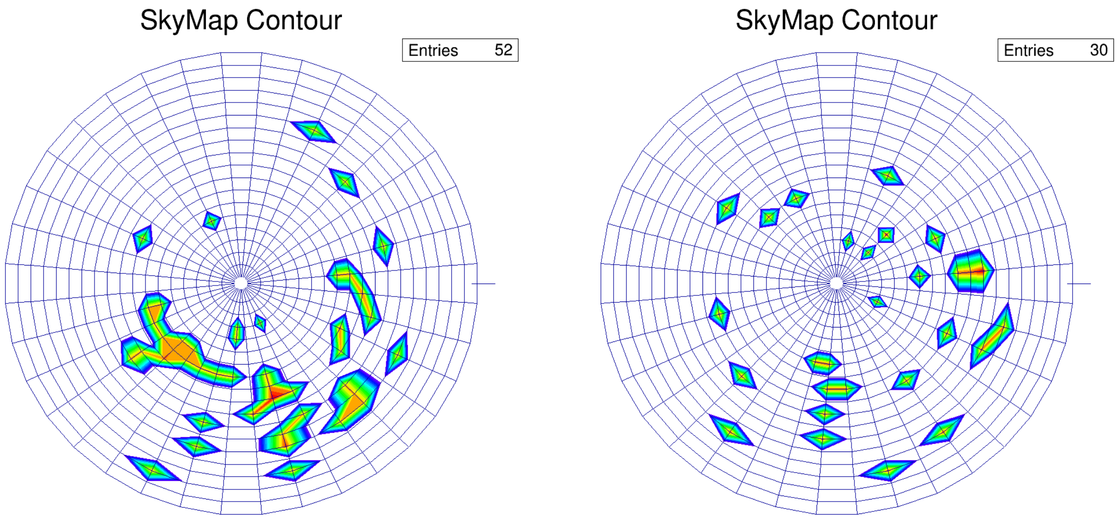


(a) (b)

Figure 7.15: Reconstructed X_{max} and reconstructed radio energy emitted by the air shower of the FD-RD set (left) and the SD-RD set (right). For the FD set, events with X_{max} smaller than $600\,\mathrm{g/cm^2}$ can be found.

In Fig. 7.17, an example for such a reconstructed event is shown and that using FD mono for the RdObserver is successfully reconstructed.

For the complete dataset, just 206 events are reconstructed which is only half the number of events if FD hybrid is used. This is not a surprise since no adaptations to the signal



(a) (b)

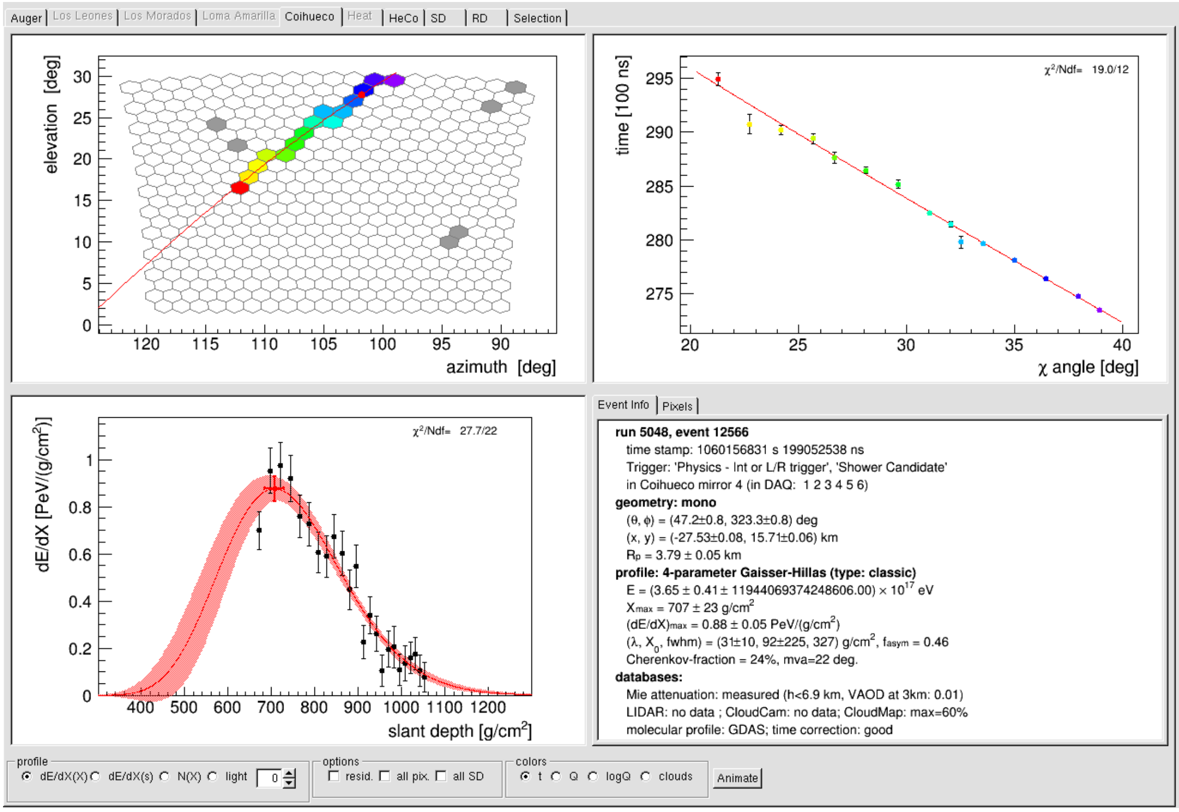
Figure 7.16: Contour plot of the sky map of the FD-RD set (left) and the SD-RD set (right). It visualizes the incoming direction of the primary particle. The excess to the south-east is clearly visible for the FD set.

search window which is optimized for using the SD geometry have been made. The signal search window of RD is based on the expected arrival time of the radio signal which is in turn dependent on the reconstructed shower geometry. Using FD mono, the geometry can only be calculated with larger uncertainties and thus it gets more difficult to determine the exact position of the radio pulse inside the radio trace. The other problem is the already mentioned variable *UsedEye*. Also for using FD mono, setting the eye individually based on the reconstruction parameters would be an option for improvements. Besides increasing the number of FD-RD events, investigating this topic in more detail could also have advantages for the FD reconstruction. One use case could be replacing the hottest station used as lever arm for the FD timing fit by the *hottest antenna*. Since the sampling rate of the RDS is higher than the rate of the SD electronics, this could reduce uncertainties of the fit. The implementation needs to be done by solving the following kind of problem: To estimate the hottest antenna for the FD reconstruction, the position of the radio signal inside the traces of the RDS need to be evaluated. As already mentioned, finding the radio signal inside the trace is based on the signal search window which is defined relatively to the expected arrival time which in turn depends on the already available FD or SD reconstruction. Two possibilities exist. On the one hand, one could try to reconstruct the shower axis only with radio itself by doing triangulation over several antennas to find the radio signal inside the traces. Just looking at the highest peak would increase the chance of selecting a noise pulse. Then the FD reconstruction would be processed with the estimated signal times of all possible RDS to find the best parameters. On the other hand, first the FD mono reconstruction could be used to have a rough estimation of the shower geometry. With a broader radio signal search window, the hottest antenna could be evaluated and then the FD reconstruction would be processed again as "FD hybrid" with the RDS as lever arm. This calculated shower axis could then again be used to search for the hottest antenna. This would be done iteratively until the uncertainties of the FD fit have reached a minimum.

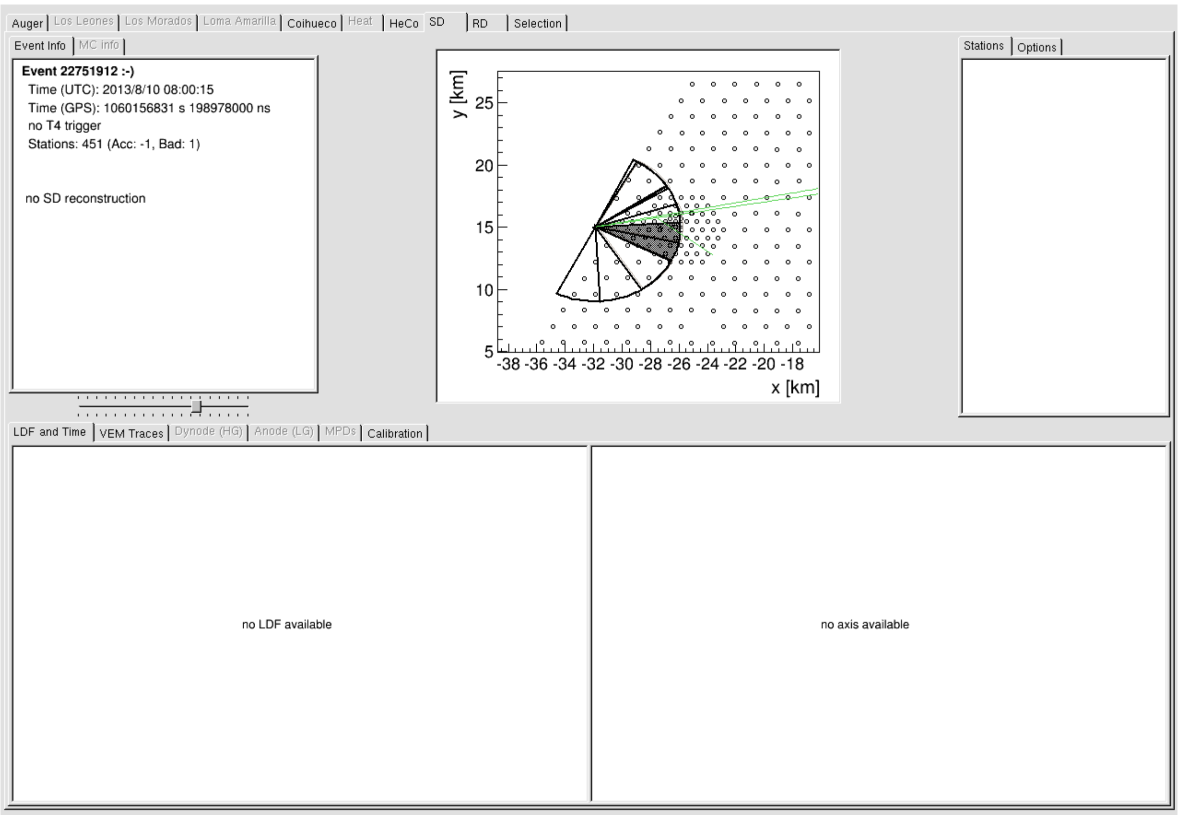
As a conclusion, it could be shown that RD and FD can benefit from the adaptations implemented in the RdObserver described in this chapter but there is still much room for improvements and also a larger data set is needed for further evaluation.

Table 7.3: Results of the analysis for the complete dataset from AERA phase II using the standard RdObserver reconstruction chain and the RdObserver with FD used as input for RD. The run and the event number for 52 events which are not reconstructed in the standard chain are listed in the first two columns. In the last column, the run and the event number are listed for the 30 events which are not reconstructed using the new FD driven RdObserver.

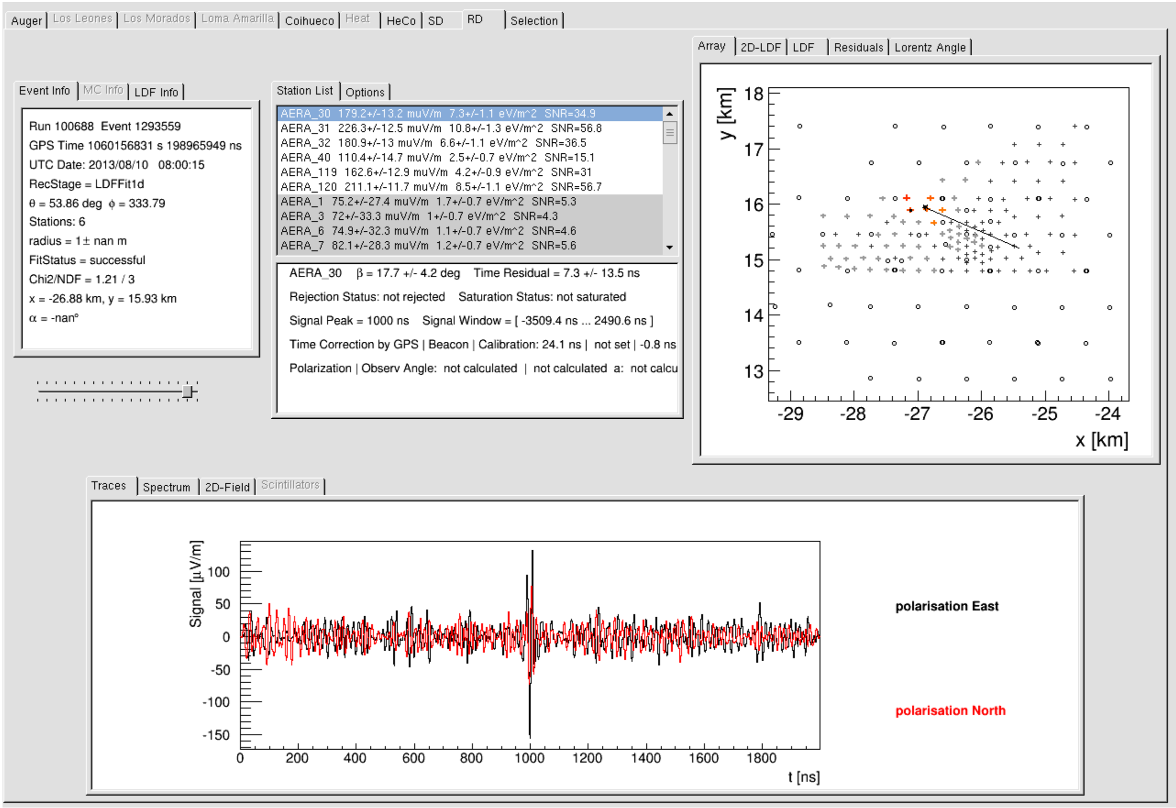
Event RdObs$_{\mathbf{FD}}$	**Event RdObs$_{\mathbf{FD,continued}}$**		**RdObs$_{\mathbf{Standard}}$**
100237-735077	100874-96375		100650-319125
100237-1592716	100874-518104		100653-120433
100237-1786919	100874-554782		100696-831520
100237-1878073	100884-290568		100726-555899
100237-2117809	100900-58764		100731-707346
100237-2218653	100900-63587		100732-102048
100237-2300177	100900-66036		100795-185940
100637-110070	100900-73676		100795-291214
100653-142311	100900-117701		100806-54970
100682-503892	100900-137768		100806-91338
100687-418169	100900-197554		100808-182364
100699-804366	100900-200975		100808-396582
100718-6	100900-203092		100808-604678
100723-10211	100900-209241		100808-699221
100727-431484	100900-262536		100812-47898
100733-594132	100900-268808		100812-206375
100759-175917	100900-271196		100822-35737
100759-178678	100900-272510		100829-26681
100795-180048	100900-272594		100838-67319
100808-386832	100907-64925		100843-3564
100812-85453	100907-83620		100865-298598
100818-26803	100919-40304		100873-72034
100824-65316			100874-295490
100834-45769			100890-76536
100834-89042			100890-184220
100834-101227			100896-390792
100850-88184			100901-140056
100865-39080			100906-73611
100873-67288			100911-60549
100873-318299			100911-68135
...			


Auger
Los Leones
Los Morados
Loma Amarilla
Coihueco
Heat
HeCo
SD
RD
Selection
elevation [deg]
azimuth [deg]
time [100 ns]
χ angle [deg]
χ²/Ndf= 19.0/12
dE/dX [PeV/(g/cm²)]
slant depth [g/cm²]
χ²/Ndf= 27.7/22
Event Info
Pixels
run 5048, event 12566
time stamp: 1060156831 s 199052538 ns
Trigger: 'Physics - Int or L/R trigger', 'Shower Candidate'
in Coihueco mirror 4 (in DAQ: 1 2 3 4 5 6)
geometry: mono
(θ, φ) = (47.2±0.8, 323.3±0.8) deg
(x, y) = (-27.53±0.08, 15.71±0.06) km
Rp = 3.79 ± 0.05 km
profile: 4-parameter Gaisser-Hillas (type: classic)
E = (3.65 ± 0.41 ± 11944069374248606.00) × 10¹⁷ eV
Xmax = 707 ± 23 g/cm²
(dE/dX)max = 0.88 ± 0.05 PeV/(g/cm²)
(λ, X0, fwhm) = (31±10, 92±225, 327) g/cm², fasym = 0.46
Cherenkov-fraction = 24%, mva=22 deg.
databases:
Mie attenuation: measured (h<6.9 km, VAOD at 3km: 0.01)
LIDAR: no data ; CloudCam: no data; CloudMap: max=60%
molecular profile: GDAS; time correction: good
profile
dE/dX(X) dE/dX(s) N(X) light
options
resid. all pix. all SD
colors
t Q logQ clouds
Animate

(a)


Auger
Los Leones
Los Morados
Loma Amarilla
Coihueco
Heat
HeCo
SD
RD
Selection
Event Info
MC info
Event 22751912 :-)
Time (UTC): 2013/8/10 08:00:15
Time (GPS): 1060156831 s 198978000 ns
no T4 trigger
Stations: 451 (Acc: -1, Bad: 1)
no SD reconstruction
y [km]
x [km]
Stations
Options
LDF and Time
VEM Traces
Dynode (HG)
Anode (LG)
MPDs
Calibration
no LDF available
no axis available

(b)



(c)

Figure 7.17: Comparing event ID = 100688-1293559 (randomly chosen) reconstructed with the RD-Observer by using FD mono as input. Shown is the output of the Event-Browser with the FD part in (a), the SD part in (b) and the RD part in (c) as a verification for the FD mono implementation. It is clearly visible, that no SD information is available in (b) and the RD event is however reconstructed.

8 Summary

The AERA experiment at the Pierre Auger Observatory provides ideal conditions to measure the radio component of extensive air showers. It has been been built up in three phases and contains 124 autonomous detector stations in phase III. In order to handle the measured data more efficiently, the development of a new input and especially output library named *AERAROOTIOLib* for radio data produced by the AERA experiment was the central subject of this work. Integrating the output functionality could yield to some major improvements. Besides easier data handling, compared to the raw binary file, a compression factor of roughly 1.2 and a significantly increased processing speed in $\overline{\textrm{Off}}\underline{\textrm{line}}$ by a factor of 5 could be achieved. The decreased data size is important for the limited transport bandwidth from Argentina to Europe.

With the *AERAConverter* it is now possible to produce different data streams and to store the events based on their trigger flags and/or based on the time into different output files. They can be used as subsets for different analyses and a re-reconstruction of the data based on different module sequences in $\overline{\textrm{Off}}\underline{\textrm{line}}$ is possible without the need to read-in the complete data set each time. The *AERAConverter* has been designed to run many jobs in parallel and is therefore able to process the dataset for several years in a short time. The additional debug mode prints the radio data in human readable values and thus enables an immediate look into the measured values. Problems inside the input radio data can be solved by integrating workarounds directly into the *AERAConverter* at one central place.

The *AERAROOTIOLib* could be successfully integrated for low level merging into CDAS. The new *BuildXRAugerfile* application has been developed to merge all different detector systems used at the Pierre Auger Observatory, SD, FD, MD and RD-data, successfully together into one single xrad file. In addition, the library is the first and only possibility to use the data of the new NIKHEF stations in $\overline{\textrm{Off}}\underline{\textrm{line}}$. Besides this, the *AERAROOTIOLib* enables radio-radio merging. Before the combined DAQ has started data taking in December 2015, two different DAQ systems existed, one for stations with KIT/BUW digitizers and one for NIKHEF digitizers. By using the new merger tool integrated in CDAS, both data sets can be searched for coincidences and can be combined together into one dataset.

The data processing is fully automated and running on the dedicated AERA server in Coihueco and afterwards the data is transferred to Europe. The data conversion into different streams and also the merging of RD data with Auger data is running by daily and hourly cron jobs which have been designed to be robust against power failures. The *AERAROOTIOLib* provides all functionality to shift the conversion process onto different machineries.

The library is already in use for creating a noise library to study the environmental conditions at the Pierre Auger Observatory and for timing calibration of the stations by using airplanes flying over the AERA array. In addition, the library is used to monitor the radio data automatically with only a small delay of 3 hours compared to real time. In Ape, the newest version of the *AERAROOTIOLib* is available. In Offline, all changes which are necessary for the *AERAROOTIOLib*, the correct interpretation of the radio extended *IOAuger*-class, the adaptions for the *EventFileExporter* to select and write out events in the new *RadioAERAroot* format and the new trigger flags were implemented. The release of the modifications in CDAS is ongoing. Using the *AERAROOTIOLib*, more than 7500 radio events could be reconstructed for the time period between the middle of 2012 and March 2016.

Different methods are currently developed to reconstruct the $X_{\max}$ based on RD data. To evaluate these methods, a benchmark dataset was urgently needed. Two approaches using different quality criteria have to be taken into account. There was a need to evaluate how efficiently the different approaches work among each other. In a prefect environment and without errors, the different methods should lead to the same results. In addition, these methods have to be compared to the well established $X_{\max}$ reconstruction in FD. Hence, the benchmark set should consist of high quality RD and FD events which are reconstructable. This benchmark dataset has been established by applying different FD and RD quality criteria and by looking at different input datasets. Unfortunately, depending on the method used, only 18 respectively 29 events could be found which pass the FD as well as the RD quality cuts. As a conclusion, a different dataset, consisting of 150 high quality RD events without FD reconstruction to evaluate the RD methods, has been established.

In order to find more combined RD-FD events, the standard Offline module sequence for reconstructing radio data, the *RDObserver*, has been extended to be able to use also FD values as input for the radio reconstruction. The coding structure in some of the radio reconstruction modules had to be changed from hard coded input values to a more flexible approach. The module sequence itself was also changed and new values like the FD eye were integrated into the xml cards. The timing window has been corrected and shows an offset of -575 ns between the timestamp of the RD and the FD event. By using the FD reconstruction instead of the SD one as input, the reconstructed data set could be enlarged by 52 SD-RD-FD events from 412 to 464 while losing 30 events compared to the original reconstruction based on SD. A first check, if FD mono can be used as input for RD, has been made but further analyses with larger datasets are needed for a final conclusion.

A

Appendix

A.1 Additional station information in the AERA data

In this section, the structure of the additional station header information in the AREA data for the different digitizer versions is shown.

A.1.1 Additional station header data structure for Dutch digitizers V1

Table A.1: AERA additional station header data structure for Dutch digitizers V1.

Byte	Description
1	Message ID
2 - 3	Message Length
4	Channel mask/select pattern
5	Trigger mask/source pattern
6 - 7	Pre trigger window
8 - 9	Coincidence window
10 - 11	Post trigger window
12 - 18	GPS time
19 - 22	CTD
23 - 24	Signal threshold Channel 1
25 - 26	Signal threshold Channel 2
27 - 28	Signal threshold Channel 3
29 - 30	Signal threshold Channel 4
31 - 32	Noise threshold Channel 1
33 - 34	Noise threshold Channel 2
35 - 36	Noise threshold Channel 3
37 - 38	Noise threshold Channel 4

A.1.2 Additional station header data structure for Dutch digitizers V2

Table A.2: AERA additional station header data structure for Dutch digitizers V2.

Byte	General description	Specific description
1 - 2	Trigger Mask	
3 - 9	GPS time	
10	V2 status	
11 - 14	CTD	
15 - 16	V2 Length channel 1	
17 - 18	V2 Length channel 2	
19 - 20	V2 Length channel 3	
21 - 22	V2 Length channel 4	
23 - 24	V2 signal treshold ch1	
25 - 26	V2 noise treshold ch1	
27 - 28	V2 signal treshold ch2	
29 - 30	V2 noise treshold ch2	
31 - 32	V2 signal treshold ch3	
33 - 34	V2 noise treshold ch3	
35 - 36	V2 signal treshold ch4	
37 - 38	V2 noise treshold ch4	
39 - 42	Quantile 1	
43 - 46	Quantile 2	
47 - 50	CTP	(Count Ticks between PPS) PPS = pulse per second signal from GPS receiver
51 - 52	Sync	
53 - 92	PPSGPS	Byte 1 - 4: Board serial number Byte 5 - 11: GPS Time in DDMMYYYY HH:MM:SS Byte 12: Status Byte 13 - 20: Longitude Byte 21 - 28: Latitude Byte 29 - 36: Altitude Byte 37 - 40: Temperature
93 - 104	PPSControl	Byte 1 - 2: Control Byte 3 - 4: Trigger Byte 5: ChEn Byte 6: TrDiv Byte 7 - 8: CoinT

Continued on next page

Table A.2 – continued from previous page

Byte	General description	Specific description
105 - 120	PPSWindows	Byte 1 - 2: Channel 1 pre Byte 3 - 4: Channel 1 post Byte 5 - 6: Channel 2 pre Byte 7 - 8: Channel 2 post Byte 9 - 10: Channel 3 pre Byte 11 - 12: Channel 3 post Byte 13 - 14: Channel 4 pre Byte 15 - 16: Channel 4 post
121 - 132	PPS Channel 1	Byte 1 - 2: Channel 1 gain Byte 3: Channel 1 offset Byte 4: Channel 1 inttime Byte 5 - 6: Channel 1 basemax Byte 7 - 8: Channel 1 basemin Byte 9: Channel 1 pmv Byte 10: Channel 1 filt
133 - 144	PPS Channel 2	Byte 1 - 2: Channel 2 gain Byte 3: Channel 2 offset Byte 4: Channel 2 inttime Byte 5 - 6: Channel 2 basemax Byte 7 - 8: Channel 2 basemin Byte 9: Channel 2 pmv Byte 10: Channel 2 filt
145 - 156	PPS Channel 3	Byte 1 - 2: Channel 3 gain Byte 3: Channel 3 offset Byte 4: Channel 3 inttime Byte 5 - 6: Channel 3 basemax Byte 7 - 8: Channel 3 basemin Byte 9: Channel 3 pmv Byte 10: Channel 3 filt
157 - 168	PPS Channel 4	Byte 1 - 2: Channel 4 gain Byte 3: Channel 4 offset Byte 4: Channel 4 inttime Byte 5 - 6: Channel 4 basemax Byte 7 - 8: Channel 4 basemin Byte 9: Channel 4 pmv Byte 10: Channel 4 filt

Continued on next page

Table A.2 – continued from previous page

Byte	General description	Specific description
169 - 180	PPS Trigger 1	Byte 1 - 2: Channel 1 threshold 1 Byte 3 - 4: Channel 1 threshold 2 Byte 5: Channel 1 Tprev Byte 6: Channel 1 Tper Byte 7: Channel 1 Tcmax Byte 8: Channel 1 Ncmax Byte 9: Channel 1 Ncmin Byte 10: Channel 1 Qmax Byte 11: Channel 1 Qmin
181 - 192	PPS Trigger 2	Byte 1 - 2: Channel 2 threshold 1 Byte 3 - 4: Channel 2 threshold 2 Byte 5: Channel 2 Tprev Byte 6: Channel 2 Tper Byte 7: Channel 2 Tcmax Byte 8: Channel 2 Ncmax Byte 9: Channel 2 Ncmin Byte 10: Channel 2 Qmax Byte 11: Channel 2 Qmin
193 - 204	PPS Trigger 3	Byte 1 - 2: Channel 3 threshold 1 Byte 3 - 4: Channel 3 threshold 2 Byte 5: Channel 3 Tprev Byte 6: Channel 3 Tper Byte 7: Channel 3 Tcmax Byte 8: Channel 3 Ncmax Byte 9: Channel 3 Ncmin Byte 10: Channel 3 Qmax Byte 11: Channel 3 Qmin
205 - 216	PPS Trigger 4	Byte 1 - 2: Channel 4 threshold 1 Byte 3 - 4: Channel 4 threshold 2 Byte 5: Channel 4 Tprev Byte 6: Channel 4 Tper Byte 7: Channel 4 Tcmax Byte 8: Channel 4 Ncmax Byte 9: Channel 4 Ncmin Byte 10: Channel 4 Qmax Byte 11: Channel 4 Qmin
217 - 232	PPS Filter11	8 values each with 2 bytes
233 - 248	PPS Filter12	8 values each with 2 bytes
249 - 264	PPS Filter21	8 values each with 2 bytes
265 - 280	PPS Filter22	8 values each with 2 bytes
281 - 296	PPS Filter31	8 values each with 2 bytes
297 - 312	PPS Filter32	8 values each with 2 bytes
313 - 328	PPS Filter41	8 values each with 2 bytes

Continued on next page

Table A.2 – continued from previous page

Byte	**General description**	**Specific description**
329 - 344	PPS Filter42	8 values each with 2 bytes

A.1.3 Additional station header data structure for German digitizers V1

Table A.3: AERA additional station header data structure for German digitizers V1.

Byte	Description
1 - 2	Start Low Gain 1
3 - 4	End Low Gain 1
5 - 6	Start Low Gain 2
7 - 8	End Low Gain 2
9 - 10	Trigger Rate 1
11 - 12	Trigger Rate 2
13 - 14	Trigger Threshold 1
15 - 16	Trigger Threshold 2
17 - 18	Ger status 1
19 - 20	Ger status 2
21 - 22	GPS status
23 - 26	Trigger Status and T2 subsec
27 - 28	T3 delay
29 - 30	Temp int (Bit 1 - 8) and ext (Bit 9 - 16)
31 - 32	DAQ Rev
33 - 34	MSP430rev
35 - 36	Firm Rev.

A.1.4 Additional station header data structure for German digitizers V2

Table A.4: AERA additional station header data structure for German digitizers V2 [99].

Short	Description
0	Additional header version
1	SVN number of the local DAQ program
2	Rev. of software on the micro-contr.
3	Rev. of the FGPA firmware
4	Rev. of the Linux distribution
5 - 8	Internal digital board serial-number from IC
9	Version of the digital board [0 = unknown, 1-3 = AERA24 - AERA153]
10 - 29	10 ASCII characters describing the 4 channels
30	GPS receiver status
31 - 32	GPS latitude in milliarcseconds
33 - 34	GPS longitude in milliarcseconds
35 - 36	GPS height
37	MSP430 status register bits 16-29 of power supply errors
38	MSP430 status register bits 0-15 of power supply errors
39	Delay of T3 GetEventMessage: 8 Bit delay in sec + 8 Bit delay in ms
40	External temperature (FPGA)
41	Internal temperature (micro-contr.)
42	Current trigger rate channel 1 (old low-gain ch. 1), -1: no triggering
43	Current trigger rate channel 2 (old high-gain ch. 1)
44	Current trigger rate channel 3 (old low-gain ch. 2)
45	Current trigger rate channel 4 (old high-gain ch. 2)
46	Current trigger threshold channel 1 (old low-gain ch. 1), -1: no triggering
47	Current trigger threshold channel 2 (old high-gain ch. 1)
48	Current trigger threshold channel 3 (old low-gain ch. 2)
49	Current trigger threshold channel 4 (old high-gain ch. 2)
50	Trigger status byte, encoding which channel has triggered
51	If externally triggered event also triggered internally: High byte of T2 subsecond. Else: 0
52	If externally triggered event also triggered internally: Low word of T2 subsecond. Else: 0.

A.2 AERAROOTIOLib structure information in the documentation

Figure A.1 and Fig. A.2 show an example for the AERAROOTIOLib information from the doxygen documentation. In general, a menu with four entries can be found on top of the page. A special class can be selected via *Classes* → *Class List* to get a list of all public member functions and private attributes, a detailed description of the class and the constructor, destructor and full member function documentation with explanations.

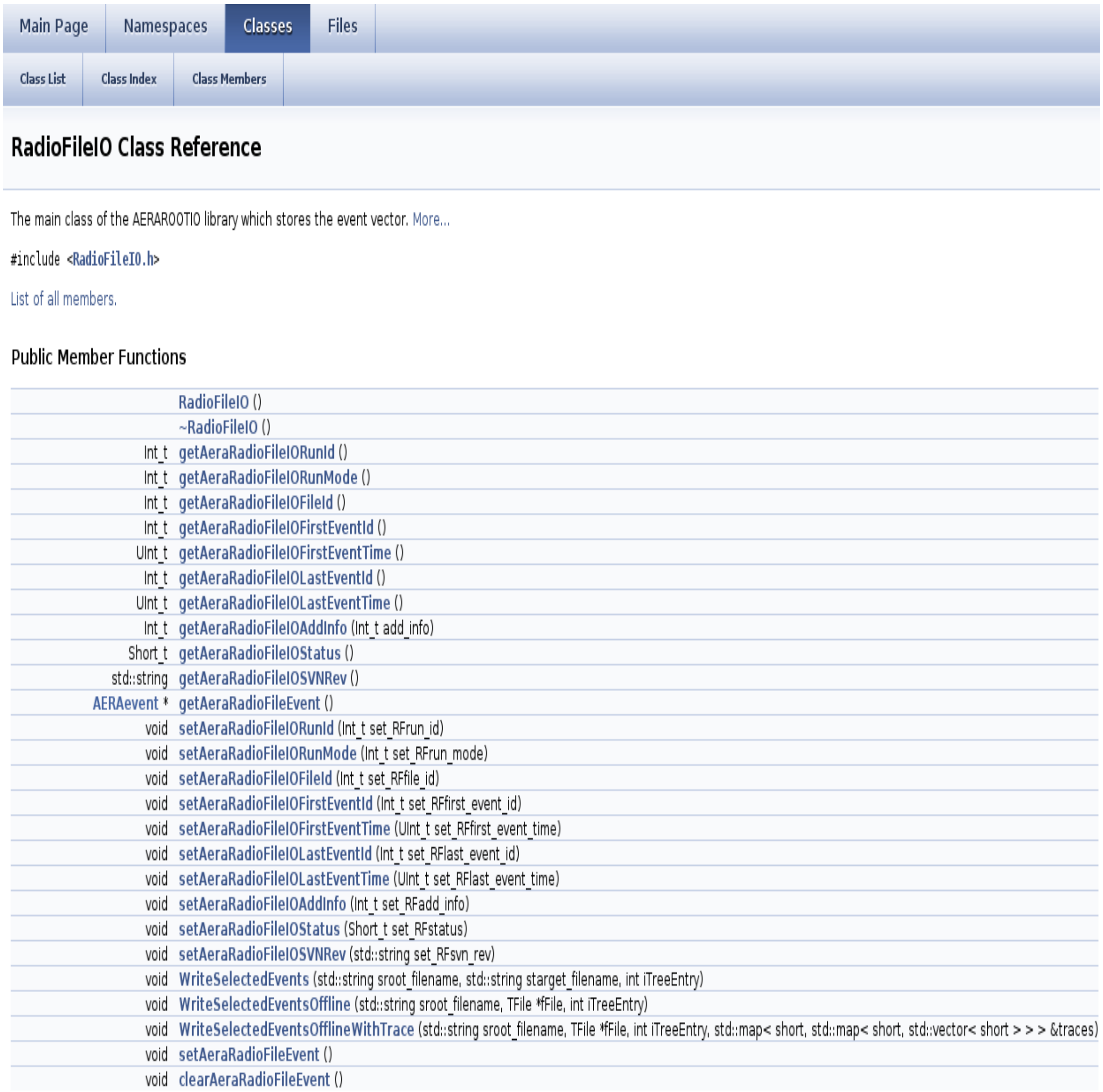


Figure A.1: An example for the class structure information in the doxygen documentation showing the main menu and a list of the public member functions.

Private Attributes

Int_t	RFrun_id
Int_t	RFrun_mode
Int_t	RFfile_id
Int_t	RFfirst_event_id
UInt_t	RFfirst_event_time
Int_t	RFlast_event_id
UInt_t	RFlast_event_time
Short_t	RFstatus
std::string	RFsvn_rev
std::vector< Int_t >	vRFadd_info
std::vector< AERAevent >	vAERAevents

Detailed Description

The main class of the AERAROOTIO library which stores the event vector.

Store also the svn revision number and the binary file header information. Include the WriteSelectedEvents functions for the converter and for Offline.

Author:
S. Mathys, University of Wuppertal

Constructor & Destructor Documentation

RadioFileIO::RadioFileIO ( )

Standard constructor

RadioFileIO::~RadioFileIO ( )

Standard deconstructor

Member Function Documentation

void RadioFileIO::clearAeraRadioFileEvent ( )

Clear the vAERAevents vector

AERAevent * RadioFileIO::getAeraRadioFileEvent ( )

Get the pointer to an event inside the vector vAERAevents. Since just one event is written into the vector at the same time, and the vector is cleared after the event is written into the tree, all the e
with the other classes.

Figure A.2: An example for the class structure information in the doxygen documentation showing private attributes and the member function description.

A.3 Monitoring with the AERAROOTIOLib

During the hourly conversion of the monitoring stream, the AERAConverter also writes information about the number of events and the number of participating stations into a logfile. In Fig. A.3 - Fig. A.11 the number of events for each trigger flag from May 2015 to March 2016, based on the the information of this logfile for the German DAQ, is shown, see section 4.3. Since no GUI-trigger, scintillator-trigger, LS_NO_Event trigger and calibration-trigger have been seen, there is no plot for those trigger types.

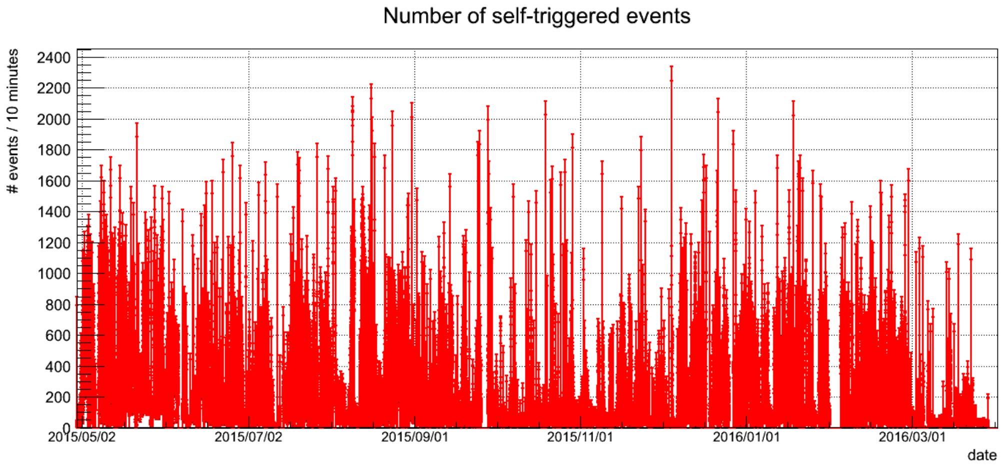


Figure A.3: Number of self-triggered events in 2015 / 2016.

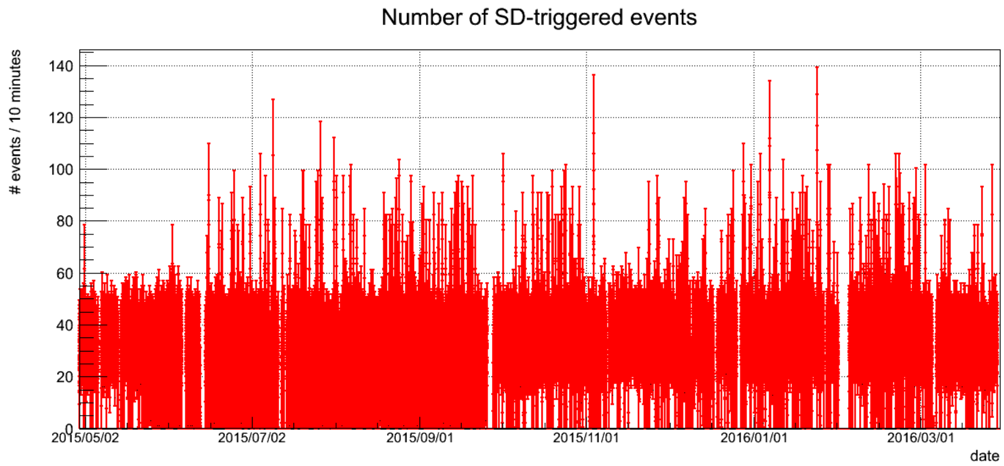


Figure A.4: Number of SD-triggered events in 2015 / 2016.

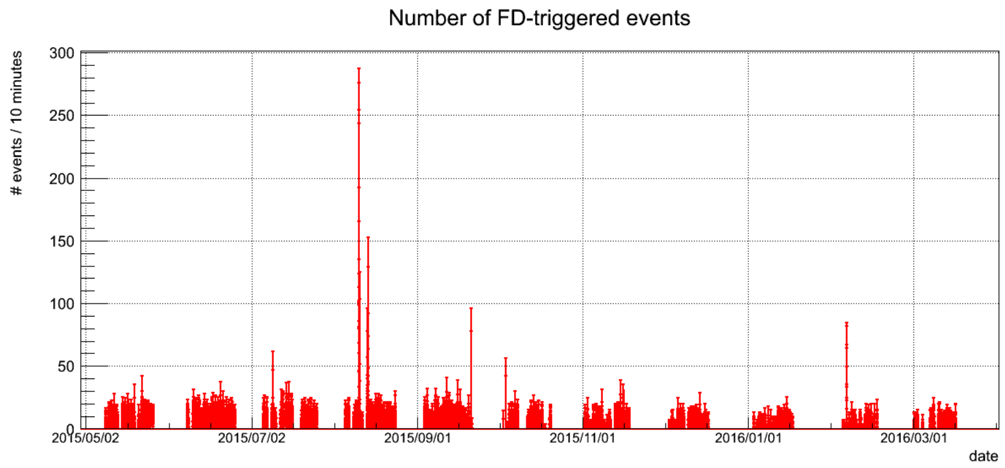


Figure A.5: Number of FD-triggered events in 2015 / 2016.

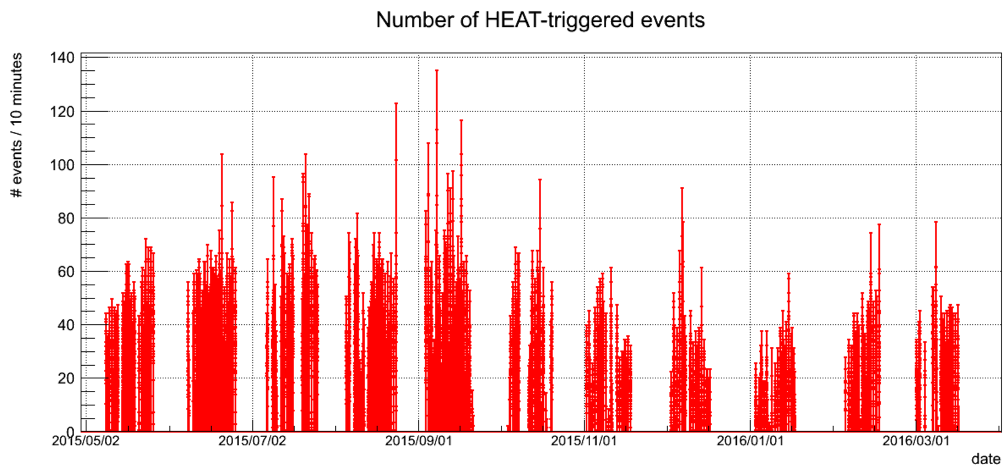


Figure A.6: Number of HEAT-triggered events in 2015 / 2016.

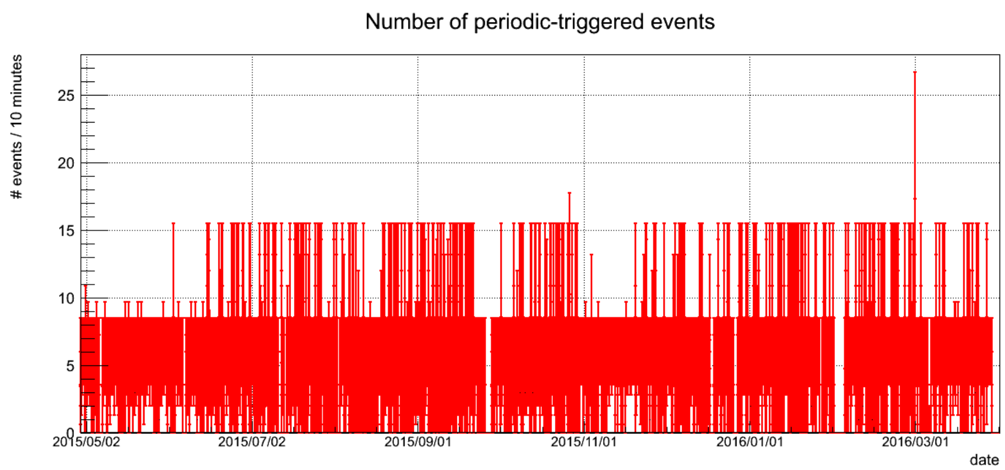


Figure A.7: Number of periodic-triggered events in 2015 / 2016.

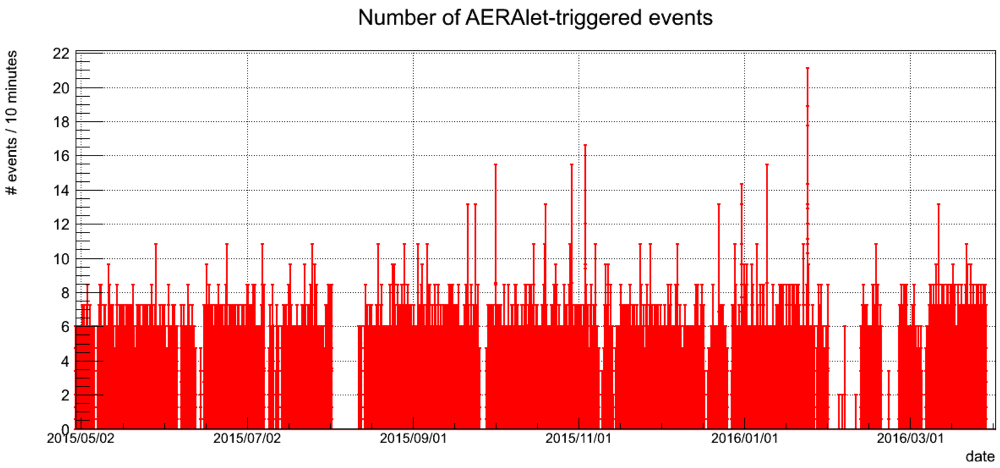


Figure A.8: Number of AERAlet-triggered events in 2015 / 2016.

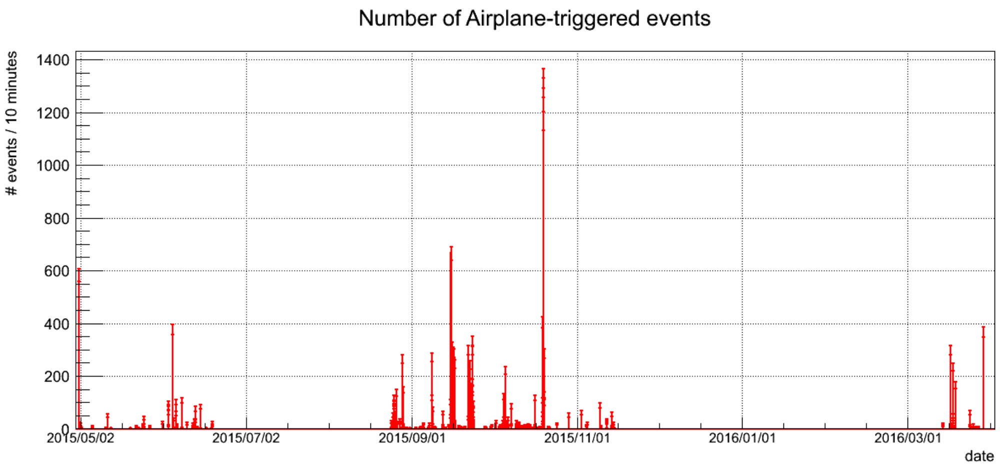


Figure A.9: Number of airplane-triggered events in 2015 / 2016.

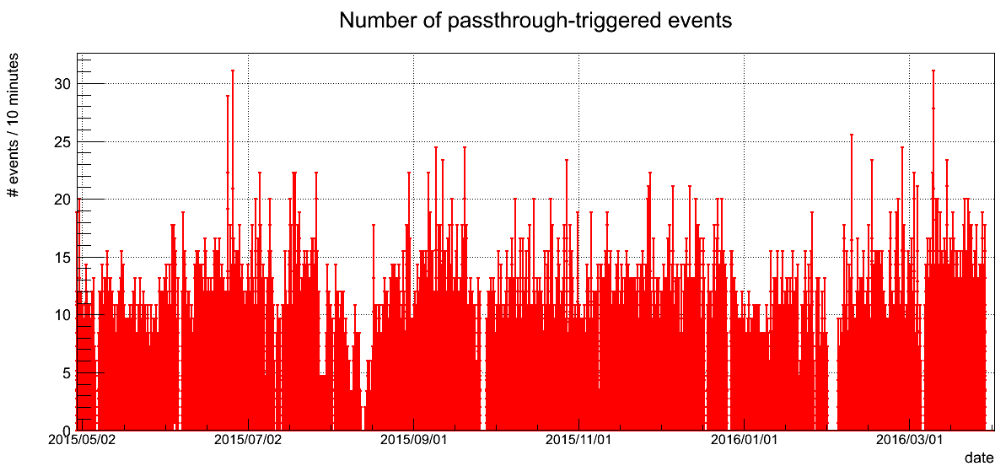


Figure A.10: Number of pass-through-triggered events in 2015 / 2016.

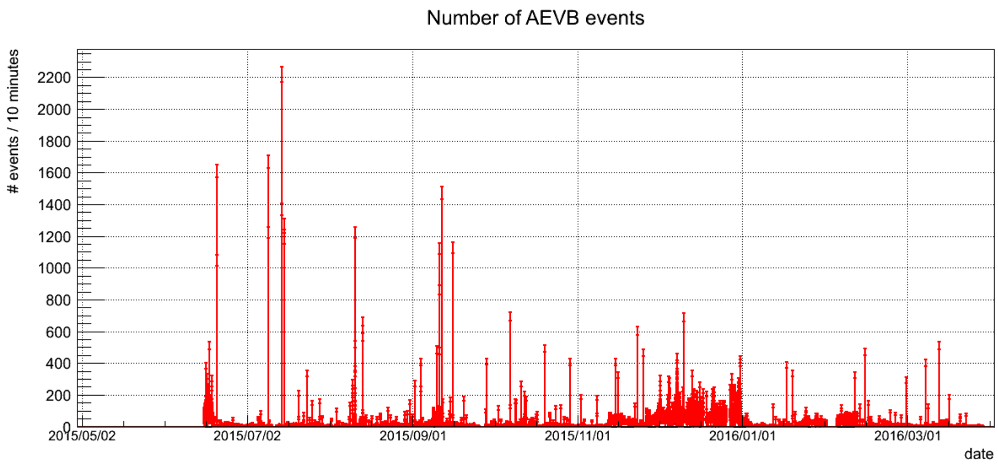


Figure A.11: Number of AEVB-triggered events in 2015 / 2016.

A.3.1 Modified FD cuts

Table A.5: Cuts used for benchmarking the FD dataset.

Cut	Cut value	Cut description
DataAcquisition.cuts		DAQ selection
badFDPeriodRejection		
skipSaturated		
noBadPixelsInPulse		
#minBkgr.RMSMergedEyes	17 6 110000	
#minBkgr.RMSSimpleEyes	17 11111	
good10MHzCorrection		
Hybrid.cuts		Hybrid selection
maxCoreTankDist	1.e20	
#hybridTankTrigger	2	
maxZenithFD	85	
minLgEnergyFD	1.e-20	
maxCoreTankDist	1500	
ambiguousHybridRejection		
Atmosphere.cuts		Atmosphere selection
hasMieDatabase		
maxVAOD	params: .1	
	nMinusOne: 100 0. 1.	
cloudCut		
HeCo.cuts		HeCo selection
HeCoHasUpTime		
minPBrass	0.9	
maxPBrassProtonIronDiff	0.05	
RejectCDASVetoPeriods	100000 4	
RejectFDASVetoPeriods	100000 4	
RejectT3VetoPeriods	100000 4	
Reconstruction.cuts		Reconst. selection
depthTrackLength	200	
XmaxErrorLessThenXmax		
maxDepthHole	30	
profileChi2Sigma	4. -0.742	
FOV.cuts		FoV selection
xMaxObsInExpectedFOV	40 20	
#fidFoV		
Time and other cuts		
T3TimesFileCo	t3TimeStampsCO.txt	RejectT3VetoPeriods
T3TimesFileHEAT	t3TimeStampsHEAT.txt	Rej.T3VetoPeriods
CDASVetoFileCo	cdasVetoTimesCO.txt	Rej.CDASVetoPeriods
CDASVetoFileHEAT	cdasVetoTimesHEAT.txt	Rej.CDASVetoPeriods
FDASVetoFileCo	fdasVetoTimesCO.txt	Rej.FDASVetoPeriods
FDASVetoFileHEAT	fdasVetoTimesHEAT.txt	Rej.FDASVetoPeriods

Continued on next page

Table A.5 – continued from previous page

Cut	Cut value	Cut description
T3VetoFileHEAT	t3ReductionTimesHEAT.txt	Rej.T3VetoPeriods
FDAmbiguousHybridList	ambiguousHybrids.txt	ambiguousHybridRej.
#FDCloudData	cloudCamHECOv9r5.txt.gz	cloudCut

A.3.2 Rd Selection

Table A.6: Event information for the RdSelection. Shown is the number of the event, the zenith and the theta angle and the multiplicity of the event.

Event	Zenith	Theta	Multiplicity
100870-30051	46.3564	221.719	5
100870-31792	45.4763	284.269	6
100870-68355	42.3773	327.342	7
100870-204852	38.6524	358.67	9
100870-280884	45.111	261.598	5
100870-393351	24.4853	322.131	7
100871-127445	32.2993	99.0122	5
100871-203400	53.6871	132.033	8
100871-213271	51.0643	316.659	5
100871-213872	24.0754	284.785	7
100871-222928	50.3014	191.411	10
100871-223540	35.3753	4.87414	7
100871-239910	50.7262	150.573	5
100871-257271	8.98158	125.982	6
100871-266249	52.4552	53.9323	5
100871-357466	54.6059	6.71887	5
100871-365508	53.8204	219.541	17
100871-382453	47.094	304.994	6
100871-390592	54.278	244.081	5
100871-501710	42.7696	259.27	6
100871-527920	54.5495	218.866	5
100871-551152	43.5821	331.421	9
100872-55058	52.9396	9.32838	9
100872-114775	49.2344	272.933	7
100872-117260	46.1718	260.517	6
100879-263931	48.4358	234.659	11
100879-306126	52.8366	354.842	6
100879-347873	47.3579	265.627	9
100879-379530	45.0634	228.121	7
100879-437412	51.4917	263.821	17
100879-489921	27.3402	347.864	6
100880-21066	49.877	246.942	9
100880-70837	41.8105	316.867	5
100880-123304	48.4733	168.407	7
100880-187928	30.394	145.708	9
100880-187938	27.6012	222.91	5
100880-204904	45.1798	298.295	5
100880-206188	51.7321	302.149	5
100882-17493	37.1075	19.5186	7
100882-32853	22.958	339.143	8

Continued on next page

Table A.6 – continued from previous page

Event	Zenith	Theta	Multiplicity
100882-83089	45.0569	238.698	6
100882-86598	22.5819	128.641	5
100882-173726	40.6395	244.853	7
100882-250638	50.5412	332.504	5
100883-20964	54.0234	250.027	8
100883-112214	30.3018	337.065	5
100883-230632	46.0574	323.047	9
100883-292668	45.9993	283.213	6
100883-314887	49.493	159.473	12
100884-33118	22.2086	138.028	5
100891-44711	48.975	232.54	5
100891-70754	44.4755	258.831	5
100891-99391	45.1883	244.169	6
100891-101930	54.1952	340.016	5
100894-2634	40.4889	244.494	5
100895-46902	46.7887	246.609	5
100895-49359	45.97	218.981	5
100895-65566	51.5006	175.855	5
100895-90417	44.5703	209.169	7
100895-143727	48.6783	269.509	7
100895-174818	42.0631	291.726	5
100895-196883	51.1585	279.245	14
100895-201171	47.1687	299.714	6
100895-210442	52.3047	359.963	5
100895-226986	20.7602	203.491	6
100895-261437	48.6612	216.293	6
100895-267178	52.8739	216.284	9
100895-287859	50.5802	253.915	6
100895-326910	43.9718	176.182	7
100895-388832	31.9301	184.17	6
100895-445570	54.6056	30.9984	8
100895-497978	54.9764	231.322	7
100895-527457	29.2961	222.56	6
100895-552490	42.1537	289.436	9
100895-648022	18.0398	311.528	5
100901-179114	47.4538	213.641	5
100901-288760	42.973	214.174	6
100901-299691	44.9629	220.832	7
100901-336872	35.406	315.781	8
100901-340578	40.3629	3.48183	10
100901-345758	44.8168	295.146	9
100901-450819	47.2927	288.711	8
100902-38996	49.7025	179.58	9

Continued on next page

Table A.6 – continued from previous page

Event	Zenith	Theta	Multiplicity
100902-51943	34.0147	141.133	5
100902-53383	46.6494	313.449	9
100902-80149	34.6989	324.811	5
100903-16123	36.161	211.565	5
100903-100468	42.8399	294.592	6
100904-27536	51.985	13.737	6
100905-15000	49.6844	332.4	5
100905-35401	47.3309	21.1198	5
100907-25618	38.6366	167.098	5
100907-64671	44.8257	213.882	6
100907-96472	41.7685	224.225	6
100907-127771	48.1707	300.033	6
100907-134206	42.6039	12.732	8
100909-56846	39.4553	295.191	5
100909-72862	25.6549	177.901	7
100909-81403	40.6886	336.843	8
100909-144309	45.8915	41.7867	5
100912-72019	50.9539	303.298	11
100912-78920	49.2094	331.335	5
100912-83159	45.0887	157.492	6
100912-102095	53.948	44.4845	5
100912-127471	54.0317	240.053	7
100912-131188	32.0155	288.811	5
100912-131405	45.2389	335.315	7
100912-161995	54.2449	293.437	7
100912-209640	50.8898	150.516	5
100912-238371	34.7804	44.3527	9
100912-264450	48.4524	306.204	5
100912-323645	54.873	322.292	5
100914-69279	52.273	283.173	7
100914-79954	51.5581	331.187	5
100914-83832	52.7467	205.292	12
100914-107570	52.7282	250.108	5
100914-130761	43.1261	293.457	5
100914-151925	37.5972	176.428	5
100914-171579	48.7785	350.209	5
100914-235225	10.2379	359.919	6
100914-284132	47.8144	268.087	9
100914-321620	53.0875	293.689	10
100915-47993	32.4025	353.287	7
100915-51795	50.2447	319.019	6
100915-58152	39.8144	203.393	5
100927-14772	41.5414	216.449	6

Continued on next page

Table A.6 – continued from previous page

Event	Zenith	Theta	Multiplicity
100927-69365	46.7119	285.452	8
100927-83586	41.8421	294.873	5
100928-4521	47.2371	230.177	11
100928-89122	49.2688	311.072	7
100928-98516	47.2544	37.8504	8
100928-147283	36.3055	67.774	5
100928-166264	44.9516	310.036	5
100928-190901	48.5954	312.889	12
100928-207707	45.4318	171.808	22
100928-265051	42.1624	259.943	5
100928-298072	48.9316	300.429	7
100928-359513	53.0866	283.439	5
100928-367188	35.3766	277.731	7
100928-388887	47.1113	244.581	7
100928-397204	42.9993	33.3447	5
100928-440083	43.4652	271.095	5
100928-474519	49.5294	15.9464	5
100928-518955	52.3877	9.26005	9
100928-522713	21.21	226.754	7
100929-20507	41.8609	213.147	5
100929-55245	36.0281	291.522	7
100929-153308	49.4214	247.91	5
100929-313264	51.7687	323.72	6
100929-313548	19.7866	255.181	5

A.3.3 Rd Observer module sequence with FD based input

```
<moduleControl>

  <loop numTimes="unbounded">

    <module> EventFileReaderOG                  </module>
    <module> RdEventPreSelector                 </module>

    <module> EventCheckerOG                     </module>
    <module> SdPMTQualityCheckerKG              </module>
    <module> TriggerTimeCorrection              </module>
    <module> SdCalibratorOG                     </module>
    <module> SdBadStationRejectorKG             </module>
    <module> SdSignalRecoveryKLT                </module>
    <module> SdEventSelectorOG                  </module>
    <module> SdPlaneFitOG                       </module>
    <module> LDFFinderKG                        </module>
    <try>
      <module> SdHorizontalReconstruction       </module>
    </try>

    <module> FdCalibratorOG                   </module>
    <module> FdEyeMergerKG                    </module>
    <module> FdPulseFinderOG                  </module>
    <module> FdSDPFinderOG                    </module>
    <module> FdAxisFinderOG                   </module>
    <module> HybridGeometryFinderOG           </module>
    <module> HybridGeometryFinderWG           </module>
    <module> FdApertureLightKG                </module>
    <module> FdEnergyDepositFinderKG          </module>
    <module> FdProfileReconstructorKG         </module>

    <module> RdEventInitializer                 </module>
    <module> RdStationPositionCorrection        </module>
    <module> RdStationRejector                  </module>
    <module> RdChannelADCToVoltageConverter     </module>
    <module> RdChannelSelector                  </module>
    <module> RdChannelPedestalRemover           </module>
    <module> RdChannelResponseIncorporator      </module>
    <module> RdChannelBeaconTimingCalibrator    </module>
    <module> RdChannelBeaconSuppressor          </module>
    <module> RdStationTimingCalibrator          </module>
    <module> RdStationTimeWindowConsolidator    </module>
    <module> RdChannelTimeSeriesTaperer         </module>
    <module> RdChannelBandstopFilter            </module>
```

```
    <module> RdChannelUpsampler                    </module>
    <module> RdChannelRiseTimeCalculator           </module>
    <module> RdAntennaChannelToStationConverter    </module>
    <module> RdStationSignalReconstructor          </module>
    <module> RdStationEFieldVectorCalculator       </module>

    <loop numTimes="unbounded">
      <module> RdTopDownStationSelector </module>
      <module> RdPlaneFit               </module>
    </loop>

    <module> RdClusterFinder                       </module>
    <module> RdPlaneFit                            </module>
    <module> RdStationRiseTimeCalculator           </module>
    <module> RdEventPostSelector                   </module>
    <module> RdLDFMultiFitter                      </module>
    <module> Rd2dLDFFitter                         </module>
    <module> RdStationTimeSeriesWindowCutter       </module>
    <module> RdStationTimeSeriesTaperer            </module>
    <module> RdREASSimPreparator                   </module>
    <module> EventFileExporterOG                   </module>
    <module> RecDataWriterNG                       </module>

  </loop>

</moduleControl>
```

Bibliography

[1] R. Brun and F. Rademakers. "ROOT: An object oriented data analysis framework". In: *Nucl. Instrum. Meth.* A389 (1997), pp. 81–86.

[2] D. J. Bird et al. "Detection of a cosmic ray with measured energy well beyond the expected spectral cutoff due to cosmic microwave radiation". In: *Astrophysical Journal* 441.1 (Mar. 1995), pp. 144–150. ISSN: 0004-637X.

[3] T. Wulf. "Beobachtungen über die Strahlung hoher Durchdringungsfähigkeit auf dem Eiffelturm". In: *Physikalische Zeitschrift* 11 (1910), pp. 811–813.

[4] V. F. Hess. "Über Beobachtungen der durchdringenden Strahlung bei sieben Freiballonfahrten". In: *Physikalische Zeitschrift* 13 (1912), pp. 1084–1091.

[5] Nobel Lectures. *Physics 1922-1941.* Amsterdam: Elsevier Publishing Company, 1965.

[6] A. Haungs. "Ultra-High Energy Cosmic Rays: A Recap of the Discussions at the European Cosmic Ray Symposium 2014". In: *Journal of Physics: Conference Series* 632.1 (2015), p. 012093.

[7] T. G. Guzik et al. "The ATIC long duration balloon project". In: *Advances in Space Research* 33.10 (2004), pp. 1763–1770.

[8] N. L. Grigorov et al. "Investigation of energy spectrum of primary cosmic particles with high and superhigh energies of space stations *proton*". In: *Yad. Fiz.* 11 (1970), pp. 1058–1069.

[9] V. A. Derbina et al. "Cosmic-ray spectra and composition in the energy range of 10-TeV - 1000-TeV per particle obtained by the RUNJOB experiment". In: *The Astrophysical Journal Letters* 628 (2005), pp. L41–L44.

[10] M. Amenomori et al. "Cosmic-ray energy spectrum around the knee observed with the Tibet air-shower experiment". In: *Astrophysics and Space Sciences Transactions* 7.1 (2011), pp. 15–20.

[11] J. Belz. "Overview of Recent HiRes Results". In: *Nuclear Physics B - Proceedings Supplements* 190 (2009), pp. 5–11.

[12] R. Abbasi et al. “IceTop: The surface component of IceCube”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 700 (2013), pp. 188–220. ISSN: 0168-9002.

[13] P. Tinyakov et al. “Latest results from the telescope array”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 742 (2014). 4th Roma International Conference on Astroparticle Physics, pp. 29–34. ISSN: 0168-9002.

[14] T. Antoni et al. “The Cosmic ray experiment KASCADE”. In: *Nuclear Instruments and Methods in Physics Research A* 513.3 (2003), pp. 490–510.

[15] W.D. Apel et al. “The KASCADE-Grande experiment”. In: *Nuclear Instruments and Methods in Physics Research A* 620.2-3 (2010), pp. 202–216. ISSN: 0168-9002.

[16] Alexander Aab et al. “The Pierre Auger Cosmic Ray Observatory”. In: *Nucl. Instrum. Meth.* A798 (2015), pp. 172–213. arXiv: `1502.01323 [astro-ph.IM]`.

[17] E. Fermi. “On the Origin of the Cosmic Radiation”. In: *Phys. Rev.* 75.8 (Apr. 1949), pp. 1169–1174.

[18] W.D. Apel et al. “KASCADE-Grande measurements of energy spectra for elemental groups of cosmic rays”. In: *Astroparticle Physics* 47 (2013), pp. 54–66. ISSN: 0927-6505.

[19] K.-H. Kampert. “Cosmic Rays at the Highest Energies – First Data from the Pierre Auger Observatory”. In: *Int.J.Mod.Phys.E* 16.04 (2007), pp. 1093–1105.

[20] V. Berezinsky, A. Gazizov, and S. Grigorieva. “Propagation and Signatures of Ultra High Energy Cosmic Rays”. In: *Nuclear Physics B - Proceedings Supplements* 136 (2004), pp. 147–158. ISSN: 0920-5632.

[21] K. Greisen. “End to the Cosmic-Ray Spectrum?” In: *Phys. Rev. Lett.* 16.17 (Apr. 1966), pp. 748–750.

[22] G. T. Zatsepin and V. A. Kuzmin. “Upper limit of the spectrum of cosmic rays”. In: *JETP Lett.* 4 (1966), pp. 78–80.

[23] H. Blümer and K.-H. Kampert. “Die Suche nach den Quellen der kosmischen Strahlung: Antworten versprechen die Luftschauerexperimente KASCADE und AUGER”. In: *Physik Journal* 56.3 (2000), pp. 39–45. ISSN: 1521-3722.

[24] R. Alves Batista et al. “Cosmic ray propagation with CRPropa 3”. In: *J. Phys. Conf. Ser.* 608.1 (2015), p. 012076. arXiv: `1410.5323`.

[25] A. Aab et al. “Depth of maximum of air-shower profiles at the Pierre Auger Observatory. I. Measurements at energies above $10^{17.8}$ eV”. In: *Phys. Rev. D* 90.12 (Dec. 2014), p. 122005.

[26] S. Ostapchenko. “QGSJET-II: physics, recent improvements, and results for air showers”. In: *European Physical Journal Web of Conferences*. Vol. 52. European Physical Journal Web of Conferences. June 2013, p. 02001.

[27] T. Pierog et al. “EPOS LHC: Test of collective hadronization with data measured at the CERN Large Hadron Collider”. In: *Phys. Rev. C* 92.3 (Sept. 2015), p. 034906.

[28] E.-J. Ahn et al. “Cosmic ray interaction event generator SIBYLL 2.1”. In: *Phys. Rev. D* 80.9 (Nov. 2009), p. 094003.

[29] M. Risse. “Test und Analyse hadronischer Wechselwirkungsmodelle mit KASCADE-Ereignisraten”. PhD thesis. Universität Karlsruhe, 2000.

[30] Pedro Abreu et al. “Ultrahigh Energy Neutrinos at the Pierre Auger Observatory”. In: *Adv. High Energy Phys.* 2013 (2013), p. 708680. arXiv: `1304.1630 [astro-ph.HE]`.

[31] M. G. Aartsen et al. “Evidence for High-Energy Extraterrestrial Neutrinos at the Ice-Cube Detector”. In: *Science* 342 (2013), p. 1242856. arXiv: `1311.5238 [astro-ph.HE]`.

[32] K.-H. Kampert and A.A. Watson. “Extensive air showers and ultra high-energy cosmic rays: a historical review”. In: *EPJ H* 37.3 (Aug. 2012), pp. 359–412.

[33] Pierre Auger et al. “Extensive Cosmic-Ray Showers”. In: *Rev. Mod. Phys.* 11.3-4 (July 1939), pp. 288–291.

[34] Bianca G. Keilhauer. “Investigation of atmospheric effects on the development of extensive air showers and their detection with the Pierre Auger Observatory”. PhD thesis. Karlsruhe, Forschungszentrum, 2004.

[35] T. Stanev. *High Energy Cosmic Rays*. Berlin New York Chichester, UK: Springer Praxis, 2004. ISBN: 3-540-40653-0.

[36] W. Heitler. *The Quantum Theory of Radiation*. London: Oxford University Press, 1944.

[37] D. Heck et al. “CORSIKA: A Monte Carlo code to simulate extensive air showers”. In: *Forschungszentrum Karlsruhe, Wissenschaftliche Berichte FZKA 6019* (Jan. 1998).

[38] D. Kuempel. “Multivariate Search for a Directional Excess of EeV Photons with the Pierre Auger Observatory”. Ph.D. Thesis. Bergische Universität Wuppertal, 2011.

[39] F. Schmidt. *CORSIKA Shower Imagese*. `http://www.ast.leeds.ac.uk/~fs/showerimages.html`. [Online; accessed 2016]. 2016.

[40] J. V. Jelley et al. “Radio pulses from extensive cosmic-ray air showers”. In: *Nature* 205 (1965), pp. 327–328.

[41] H. Falcke et al. “Detection and imaging of atmospheric radio flashes from cosmic ray air showers”. In: *Nature* 435 (2005), pp. 313–316.

[42] O. Ravel. “The CODALEMA experiment”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 662, Supplement 1 (2012). 4th International workshop on Acoustic and Radio EeV Neutrino detection Activities, S89–S94. ISSN: 0168-9002.

[43] T. Huege et al. “The Convergence of EAS Radio Emission Models and a Detailed Comparison of REAS3 and MGMR Simulations”. In: *Nuclear Instruments and Methods in Physics Research A* 662 (2012), pp. 179–186.

[44] T. Huege. “Theory and simulations of air shower radio emission”. In: *AIP Conference Proceedings* 1535.1 (2013), pp. 121–127.

[45] F. D. Kahn and I. Lerche. “Radiation from Cosmic Ray Air Showers”. In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 289.1417 (1966), pp. 206–213.

[46] H. Schoorlemmer. “Tuning in on Cosmic Rays”. Ph.D. Thesis. Radboud University Nijmegen, 2012.

[47] T. Huege, M. Ludwig, and C. W. James. "Simulating Radio Emission from Air Showers with CoREAS". In: *American Institute of Physics Conference Series* 1535 (2013), pp. 128–132.

[48] A. Aab et al. "Probing the radio emission from air showers with polarization measurements". In: *Phys. Rev. D* 89.5 (Mar. 2014), p. 052002.

[49] P. W. Gorham et al. "Observations of microwave continuum emission from air shower plasmas". In: *Phys. Rev. D* 78.3 (Aug. 2008), p. 032007.

[50] R. Smida et al. "First Experimental Characterization of Microwave Emission from Cosmic Ray Air Showers". In: *Phys. Rev. Lett.* 113.22 (Nov. 2014), p. 221101.

[51] Alvarez-Muñiz, J. et al. "The Air Microwave Yield (AMY) experiment to measure the GHz emission from air shower plasmas". In: *EPJ Web of Conferences* 53 (2013), p. 08011.

[52] C. Williams et al. "First results from the microwave air yield beam experiment (MAYBE): Measurement of GHz radiation for ultra-high energy cosmic ray detection". In: *EPJ Web Conf.* 53 (2013), p. 08008.

[53] J. Abraham et al. "Correlation of the highest energy cosmic rays with nearby extragalactic objects". In: *Science* 318 (2007), pp. 938–943. arXiv: `0711.2256 [astro-ph]`.

[54] P. Abreu et al. "Measurement of the Proton-Air Cross Section at $\sqrt{s}$=57 TeV with the Pierre Auger Observatory". In: *Phys. Rev. Lett.* 109.6 (Aug. 2012), p. 062002.

[55] A. Aab et al. "A Search for Point Sources of EeV Photons". In: *The Astrophysical Journal* 789.2 (2014), p. 160.

[56] E. Varela and the Pierre Auger Collaboration. "The low-energy extensions of the Pierre Auger Observatory". In: *Journal of Physics: Conference Series* 468.1 (2013), p. 012013.

[57] A. Kääpä. "Study of the reflective properties of Tyvek used in water Cherenkov detectors at the Pierre Auger Observatory". Bachelor Thesis. Bergische Universität Wuppertal, 2013.

[58] J. Abraham et al. "Trigger and aperture of the surface detector array of the Pierre Auger Observatory". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 613.1 (2010), pp. 29–39. ISSN: 0168-9002.

[59] P. Abreu et al. "Identifying clouds over the Pierre Auger Observatory using infrared satellite data". In: *Astroparticle Physics* 50 - 52 (2013), pp. 92–101. ISSN: 0927-6505.

[60] J. Abraham et al. "The fluorescence detector of the Pierre Auger Observatory". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 620.2 - 3 (2010), pp. 227–251. ISSN: 0168-9002.

[61] B. Fick et al. "The Central Laser Facility at the Pierre Auger Observatory". In: *Journal of Instrumentation* 1.11 (2006), P11003.

[62] The Pierre Auger Collaboration. "Techniques for measuring aerosol attenuation using the Central Laser Facility at the Pierre Auger Observatory". In: *Journal of Instrumentation* 8.04 (2013), P04009.

[63] Julian Rautenberg. "Remote operation of the Pierre Auger Observatory". In: *Proceedings, 32nd International Cosmic Ray Conference (ICRC 2011)*. Vol. 3. 2011, p. 282.

[64] A. Aab et al. "Prototype muon detectors for the AMIGA component of the Pierre Auger Observatory". In: *Journal of Instrumentation* 11.02 (2016), P02012.

[65] P. Abreu et al. "The Pierre Auger Observatory V: Enhancements". In: *Proceedings, 32nd International Cosmic Ray Conference (ICRC 2011)*. 2011. arXiv: `1107.4807 [astro-ph.IM]`.

[66] J. Alvarez-Muñiz et al. "Search for microwave emission from ultrahigh energy cosmic rays". In: *Phys. Rev. D* 86.5 (Sept. 2012), p. 051104.

[67] R. Gaior. "Detection of cosmic rays using microwave radiation at the Pierre Auger Observatory". In: *Proceedings, 33rd International Cosmic Ray Conference (ICRC2013): Rio de Janeiro, Brazil, July 2-9, 2013*, p. 0883.

[68] S. Argirò et al. "The offline software framework of the Pierre Auger Observatory". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 580.3 (2007), pp. 1485–1496. ISSN: 0168-9002.

[69] *Auger Package Environment*. `https://svn.auger.unam.mx/trac/projects/ape/`. Accessed: 2018-11-21.

[70] P. Abreu et al. "Antennas for the detection of radio emission pulses from cosmic-ray induced air showers at the Pierre Auger Observatory". In: *Journal of Instrumentation* 7.10 (2012), P10011.

[71] M. Erdmann et al. "Antenna Alignment for the first 24 Stations of AERA". In: *Internal Auger Publication* GAP-2010-083 (2010).

[72] C. Timmermans et al. "Description of the scintillator triggered AERA-II stations". In: *Internal Auger Publication* GAP-2013-074 (2013).

[73] M. Kasten. "Aufbau einer Mess-Station zur kontinuierlichen Bestimmung der Bodeneigenschaften beim Auger Engineering Radio Array". Diploma Thesis. Bergische Universität Wuppertal, 2015.

[74] Q. Dorosti and T. Huege. "The weather stations at the Auger Engineering Radio Array". In: *Internal Auger Publication* GAP-2016-030 (2016).

[75] J. Schulz. "Reconstruction of Cosmic-Ray Properties from Radio Emission of Extensive Air Showers". Ph.D. Thesis. Radboud Universiteit Nijmegen, 2016.

[76] O. Kambeitz. "Radio Detection of Horizontal Extensive Air Showers with AERA". In: *Proceedings of ARENA 2014, Anapolis, USA* (2014).

[77] F.G. Schröder et al. "New method for the time calibration of an interferometric radio antenna array". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 615.3 (2010), pp. 277–284. ISSN: 0168-9002.

[78] A. Aab et al. "Nanosecond-level time synchronization of autonomous radio detector stations for extensive air showers". In: *Journal of Instrumentation* 11.01 (2016), P01018.

[79] S. Messina and Ad M. van den Berg. “Reconstruction of ‘AERA’-triggered air showers”. In: *Internal Auger Publication* GAP-2013-115 (2013).

[80] F. Briechle. “Octocopter Position Reconstruction for Calibrating the Auger Engineering Radio Array”. Master Thesis. RWTH Aachen, 2015.

[81] K. F. Weidenhaupt. “Antenna calibration and energy measurement of ultra high energy cosmic rays with the Auger Engineering Radio Array”. PhD thesis. Aachen, 2014.

[82] A. Nelles. “Radio emission of air showers. The perspective of LOFAR and AERA”. Ph.D. Thesis. Radboud Universiteit Nijmegen, 2014.

[83] J.L. Kelley. “Data acquisition, triggering, and filtering at the Auger Engineering Radio Array”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 725 (2013). Proceedings, 5th Very Large Volume Neutrino Telescope Workshop (VLVnT 11): Erlangen, Germany, October 12-14, 2011, pp. 133–136. ISSN: 0168-9002.

[84] B. Revenu. “AERA Central trigger”. In: *Internal Auger Publication* GAP-2012-115 (2012).

[85] J. Neuser. “Cosmic Rays and the Atmospheric Electric Field: Reconstruction and Data Analysis of Radio Emission from Air Showers at the Auger Engineering Radio Array”. Ph.D. Thesis. Bergische Universität Wuppertal, 2015.

[86] S. Mathys, J. Rautenberg, and J. Neuser. “Development and applications of a ROOT based IO library for AERA”. In: *Internal Auger Publication* GAP-2015-011 (2015).

[87] B. Stroustrup. *The Design and Evolution of C++*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1994. ISBN: 0-201-54330-3.

[88] *Boost*. `http://www.boost.org/`. Accessed: 2018-11-21.

[89] *AERAROOTIOLib repository*. `http://mysql.physik.uni-wuppertal.de/svn/AERARootIo/`. Accessed: 2018-11-21.

[90] *AERAROOTIOLib doxygen documentation*. `http://auger.uni-wuppertal.de/AERAROOTIOLib/`. Accessed: 2018-11-21.

[91] A. Lang and T. Huege. “Time Calibration of the Auger Engineering Radio Array (AERA) Using Airplanes”. In: *Internal Auger Publication* GAP-2015-013 (2015).

[92] A. Nelles S. Grebe and H. Schoorlemmer. “RdMonitoring - An Offline Module to monitor the raw data from AERA”. In: *Internal Auger Publication* GAP-2011-122 (2011).

[93] T. Huege. private communication. 2015.

[94] *International Earth Rotation and Reference Systems Service (IERS)*. `https://www.iers.org`. Accessed: 2018-21-11.

[95] J. Neuser. private communication. 2014.

[96] T. Winchen M. Gottowik J. Rautenberg. “Optimizing the Event Reconstruction of the Auger Engineering Radio Array for GPGPU”. In: *Internal Auger Publication* GAP-2014-101 (2014).

[97] J. Schulz. private communication. 2015.

[98] A. Porcelli. "Measurement of the Depth of Shower Maximum in the Transition Region between Galactic and Extragalactic Cosmic Rays with the Pierre Auger Observatory". PhD thesis. KIT, Karlsruhe, 2014.

[99] B. Zimmermann. private communication. 2016.

[100] *Pleiades cluster*. `http://www.pleiades.uni-wuppertal.de/`. Accessed: 2018-11-21.

B Acknowledgment

At this place I would like to take the chance to thank some people. This work would not have been possible without their support.

- I am very thankful to Prof. Dr. Karl-Heinz Kampert who offered me the possibility to be a part of the astroparticle physics group in Wuppertal, to work on the topic described in this thesis and for his mentoring.

- Special thanks to Dr. Tim Huege for being the co-examiner of this work and who was always willing to help with input concerning questions related to Offline and the AERAROOTIOLib.

- Many thanks to Dr. Julian Rautenberg for his huge support, mentoring and who always had an open ear for me. Thanks for proofreading and all the helpful comments.

- Furthermore I would like to thank the whole astroparticle physics group at the BUW for the pleasant and motivating working atmosphere, specifically Eric Mayotte for proof-reading this work and all the useful comments, Ms. Schaarwächter and my former office colleagues Jens Neuser, Lukas Niemietz, Sven Querchfeld and Lu Lu. Thanks to Prof. Dr. Klaus Helbing and Uwe Naumann for participating at all the sports events as a team. I would also like to thank Philipp Papenbreer and Dennis Pfeifer for the really nice time we had during our stays in Argentina.

- The AERA and Radio Offline Group for all the help during our meetings and conferences.

- My fellow students Andreas Behrendt, Tobias Neumann, Sascha Reinecke und Simon Schlesinger. We had some really nice years of study.

- I am incredibly grateful especially to my parents and to my friends who always support me all the way. Finally I would like to thank Monika for her patience and all the support escpecially during the last months of this thesis.

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Wuppertal, den ____________ ______________

Unterschrift