Efficiencies of the drift chambers in the EMMA experiment

Tiia Monto

University of Jyväskylä
Department of Physics

Master's thesis
Supervisor: Timo Enqvist
13.9.2013

Abstract

Cosmic rays are high-energy subatomic particles which travel almost at the speed of light all over the space. The shape of the cosmic ray energy spectrum is measured experimentally, but it is not perfectly understood. The slope of the spectrum at high energies is constant up to the knee energy (about 10^{15} eV) where the slope steepens. The knee has been tried to be explained by several models which aim to describe the origin and the acceleration mechanisms of the cosmic ray. The stars and the shockwaves from supernova explosions are believed to be at least a part of cosmic ray evolution.

The cosmic rays were found in the early 20th century and they have been studied with several methods. When a primary cosmic particle collides with Earth atmosphere, different reactions create a cascade of secondary particles (air shower) which may be detected on Earth. The EMMA experiment studies the cosmic ray with the knee energy by detecting the muons of air shower. EMMA is operating underground at the depth of 75 meters in Pyhäsalmi mine. The measurement stations are reached only by the muons with 50 GeV thresold energy. The stations are consisted of drift chambers, scintillation detectors and limited streamer tubes. The gas-filled drift chambers form the basis of the experiment. Their operation is based on the gas ionization which causes signals on the electrical wires. One plank is formed of seven drift chambers attached together.

In this work I study the efficiencies of the drift chambers of EMMA. My C++ program evaluates the efficiencies as a function of time and as a function of position by using data measured in calibration runs in the surface laboratory. According to my results the average ”mean top efficiency” (excluding the largest efficiency peaks) of the drift chambers is 76.5 %. The efficiency of the worst plank is 65.4 % and the best 88.1 %. The results' systematical inaccuracy may be from the method of determining the efficiency and possible fault in data. The random inaccuracy may be from the problems of the measurement system and the method of calculating the mean top efficiency by excluding the efficiency peaks. My results seem to be slightly lower than the others' results, thus my method of determining the efficiency may be stricter. Actually some efficiency peaks may be explained by the external factors like pressure changes and problems in electronics. The differencies of left and right part of a chamber may be due to the possible problems in grading lines.

Tiivistelmä

Kosminen säteily koostuu korkeaenergisistä subatomisista hiukkasista, jotka liikkuvat avaruudessa lähes valonnopeudella. Kosmisen säteilyn energiaspektrin muoto on mitattu kokeellisesti, mutta sitä ei täysin ymmärretä. Energiaspektrin kulmakerroin on vakio polvienergiaan (n. 10^{15} eV) saakka, tässä kohdassa spektrin derivaatta pienenee. Spektrin polvea on yritetty selittää useilla malleilla, jotka pyrkivät kuvaamaan kosmisen säteilyn alkuperää ja kiihdytysmekanismeja. Tähtien ja supernovien shokkiaaltojen uskotaan liittyvän ainakin osittain kosmisen säteily evoluutioon.

Kosminen säteily löydettiin 1900-luvun alussa ja sitä on tutkittu useilla menetelmillä. Primaarisen kosmisen hiukkasen törmätessä ilmakehään syntyy erilaisten reaktioiden kautta sekundaarihiukkasten kaskadi eli ilmakuuro, josta voidaan tehdä havaintoja Maapallolla. EMMA-koe tutkii polvienergian kosmista säteilyä havaitsemalla ilmakuuron myoneita. EMMA:n koeasema sijaitsee Pyhäsalmen kaivoksella noin 75 metrin syvyydessä. Kalliokerroksen läpi koeasemalle pääsevät vain sellaiset myonit, joiden energia on vähintään noin 50 GeV. Koeasema koostuu ajautumiskammioista, tuikeilmaisimista ja limited streamer tube -tyyppisistä ilmaisimista. Kokeen rungon muodostavat ajautumiskammiot, jotka ovat kaasutäytteisiä lankailmaisimia. Niiden toiminta perustuu kaasussa tapahtuvaan ionisaatioon, joka aiheuttaa signaalin sähkölankoihin. Yksi plankki koostuu seitsemästä yhteen liitetystä kammiosta.

Työssäni tutkin ajautumiskammioiden tehokkuuksia. C++-ohjelmani laskee tehokkuudet sekä ajan että paikan funktiona käyttäen mittausdataa, joka on saatu maanpäällisistä kalibraatiomittauksista. Tulosteni mukaan kammioiden keskiteho on 76.5 % (keskiarvon laskemisessa suurimmat tehopiikit on jätetty huomioimatta). Huonoimman plankin tehokkuus on 65.4 % ja parhaan 88.1 %. Tulosteni systemaattinen virhe voi aiheutua tavasta määritellä tehokkuus ja mahdollisesta mittausdatan virheestä. Satunnaisvirheet voivat johtua hetkellisistä ongelmista mittausjärjestelmässä sekä tavasta laskea keskiteho jättämällä huomiotta suurimmat tehopiikit. Tulosteni tehokkuudet ovat hieman pienemmät verrattuna muiden tuloksiin, joten tehokkuuden määrittämistapani lienee vaativampi. Osa hetkellisistä tehokkuuden alentumista voidaan selittää mittausjärjestelmään vaikuttaneilla ulkoisilla tekijöillä, kuten paineen vaihteluilla ja sähkölaitteiden ongelmilla. Kammion vasemman ja oikean puolisen tehokkuuden eroavaisuudet saattavat johtua ongelmista gradinglangoissa.

Contents

1 Cosmic rays

Cosmic rays are high energetic subatomic particles arriving from the outer space and traveling from all the directions in the universe almost at the speed of light. As being the high energetic particles, the energy of a cosmic ray can be measured as mega electron volts (MeV) or giga electron volts (GeV)[43] instead of joules. When converting electron volts (eV) into SI-units, one eV is equivalent to $1.6 \cdot 10^{-19}$ joules. The energy range of a cosmic particle is about from 10 eV to 10^{20} eV or above.

Cosmic rays include charged particles, mainly protons, but also electrons and heavier particles from helium nuclei to iron. The cosmic particle must be stable in order that it does not decay into other particles during the travel throught the space. Also photon background radiation is bombarding the earth isotropichally, but it may not be properly classified as cosmic radiation.

On the basis of the origin, the cosmic rays can be divided into two groups: the ones emitted from the Sun and the ones originated from the outside of our Solar System [43]. The solar cosmic rays have usually lower energy and they have emerged along the solar flares [35]. In the present work the term ”cosmic ray” means mainly the ones coming from the outside of our Solar System and they have energy of above 10^9 eV [52]. Additionally cosmic rays originated outside of our Solar System can be categorized into ”galactic cosmic radiation” produced inside of our galaxy and ”extragalactic cosmic radiation” produced outside of our galaxy [42]. The particles with an energy below the ankle energy 10^{18} eV (mentioned in section 1.3) are assumed to be producted inside the galaxy [39]. There is no consensus about the origin of the particles with an energy above 10^{15} eV. Cosmic rays with very high energies are supposed to be extragalactic and they have energy much more than 10^{15} eV. The ultra-high energy particles may reach even 10^{20} eV. According to Greisen-Zatsepin-Kuzmin (GZK) Cut-Off Theory the cosmic proton cannot have energy higher than $6 \cdot 10^{19}$ eV at the Earth due to the interaction with the microwave background radiation [14]. The very high energy particles are not believed to be originated near us, so according the GZK theory they should have been interacted with the microwave background.

Especially the cosmic rays with higher energies are not perfectly understood. The origin and acceleration mechanism of a cosmic ray are still

unsolved problems, but they are tried to be described by several models and theories. It is strongly believed that the cosmic ray origin is connected with the stars and the shockwaves from supernova explosions are commonly assumed to be at least a part of the acceleration process.

1.1 Particle's journey through the space

Contrary to the microwave and cosmic photons, the route of a charged cosmic particle is not absolute straight. There are forces, which interact with the particle changing its path and make it hard to indentify, in which direction the particle was originally coming from.

When a particle with a charge q enters a magnetic field $\vec{B}$ with a velocity $\vec{v}$, the Lorentz force $\vec{F}$ acts on a particle and pushes it to a curved track so that the force $\vec{F} = (\vec{E} + q\vec{v} \times \vec{B})$, where $\vec{E}$ is an electric field. In Fig. 1 a charged particle is moving in a constant magnetic field and zero electric field. In this case of homogenous magnetic field the particle trajectory is helical, because the initial velocity vector has a component parallel to the magnetic field. The Lorentz force is oppositely directed for negative and positive particles.

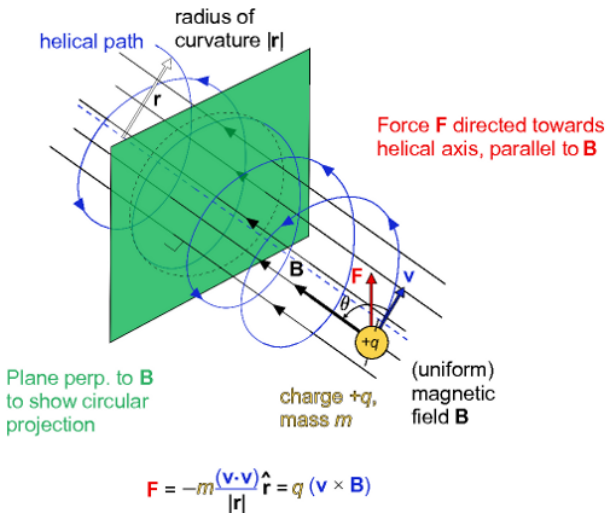


Figure 1: A charged particle travels along an helical path affected by Lorentz force (F) in an homogenous magnetic field (B) (by user Maschen in Wikimedia Commons, Public domain)

The situation is more involute, when a cosmic particle travels through the inhomogeneous electromagnetic fields of the planets and stars. The Lorentz force turns and pushes the cosmic particle so that the path is considerable complicated. When detecting the particles with not very

high energy in our solar system it is impossible to define the initial direction of the particle.

Additionally cosmic rays interact with the background microwave radiation, photons from stars and interastellar matter [1]. Actually these interactions may decrease the energy of the cosmic ray affecting the shape of the energy spectrum.

1.2 The air shower

Cosmic rays may be divided into primary and secondary particles. The primary cosmic-ray particles are striking the Earth's atmosphere from every angle incessantly with a flux of 1000 per second per square meter. The primary particle collides with the upper part of the atmosphere at the altitude of about 25 km from the sea level. In the collision various reactions occur producing a cascade of secondary cosmic particles. The cascade widens into a large area when falling towards the Earth surface. This cascade forms so called air shower, whose shape is drawn in Fig. 2. Cosmic rays detected on the Earth surface are then secondary particles.

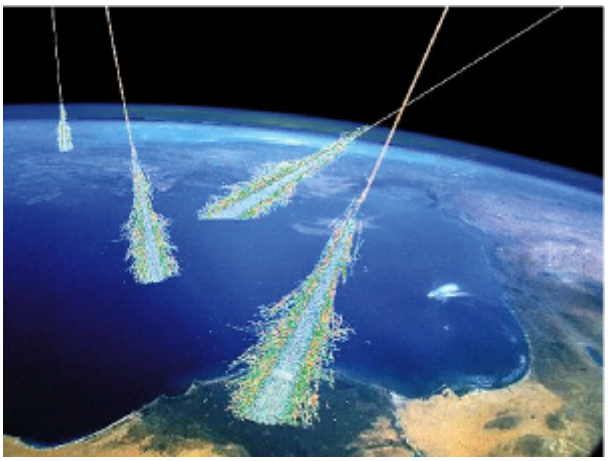


Figure 2: Cosmic ray showers (by Simon Swordy, Public domain)

The air shower may be divided into three components: electromagnetic, muonic and hadronic. The primary cosmic particle interacts usually with the atmospheric oxygen or nitrogen nuclei and the collision produces mostly pions (π), but also other particles like kaons (K). Decay processes after the primary collision are illustrated in Fig. 3. The neutral pion (π^0) has a very short lifetime and it rapidly decays into photons (γ), which further generate electron (e^-) positron (e^+) pairs. The electrons,

positrons and photons form the electromagnetic component of the air shower. The charged pions (π^+, π^-) may decay into muons (μ) and neutrinos (ν), and also charged kaons may decay into muons. Pions and kaons are the source of the muonic component of the shower. The shower core developes the hadronic component including non-decayed pions, kaons and baryons.

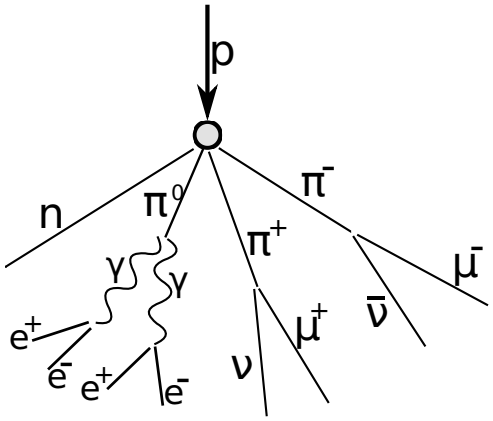


Figure 3: A primary cosmic particle (proton) collides with an atmosphere particle (by Tiia Monto, CC BY-NC)

The remarkable wide air showers are called extensive air showers (EAS). In the case of a primary particle with an energy of 10^{19} eV, the shower may include billions of particles and the shower area may spread even over several square kilometers [51]. Although much of the energy of the primary cosmic particle is absorbed in the atmosphere, the ground-level is reached by many particles: muons, electrons, positrons, photons and hadrons. A square meter of the Earth surface is hitted by about 180 particles per second and about three quarters of them are muons [57].

The muons and neutrinos are the only part of the cosmic rays, which are able to penetrate deep into the Earth surface. From the point of the EMMA experiment (discussed in section 4) the high-energy muons are the most interesting part of the air shower. The muons are producted in the upper part of the air shower and they move along straight paths. During the flight in the air the muons interact with nuclei very rarely. Thus the number of muons is not likely reduced, when they reach the sea level. The neutrinos can penetrate the Earth even a Earth-size distance. The high-energy muons are able to penetrate even 2 kilometers into the rock and as compared with the neutrinos muons are easy to be detected.

1.3 Energy spectrum

The important non-solved problem of the cosmic ray research is associated with the shape of the energy spectrum of the primary cosmic particles. The experimentally measured energy of the cosmic ray particle is represented in Fig. 4, which shows the energy spectrum from the energy 10^{13} eV to 10^{20} eV. The horizontal axis is the particle energy and vertical axis shows the particle flux. The flux axis is scaled by $E^{2.7}$ so that the remarkable properties of the graph is showed distinctly. The shape of the spectra can be approximated by a power law E^{γ} [30]. Up to an energy of 10^{15} eV the power law index is $\gamma \approx -2.7$ and after 10^{15} eV the index is $\gamma \approx -3.1$ up to the ankle.

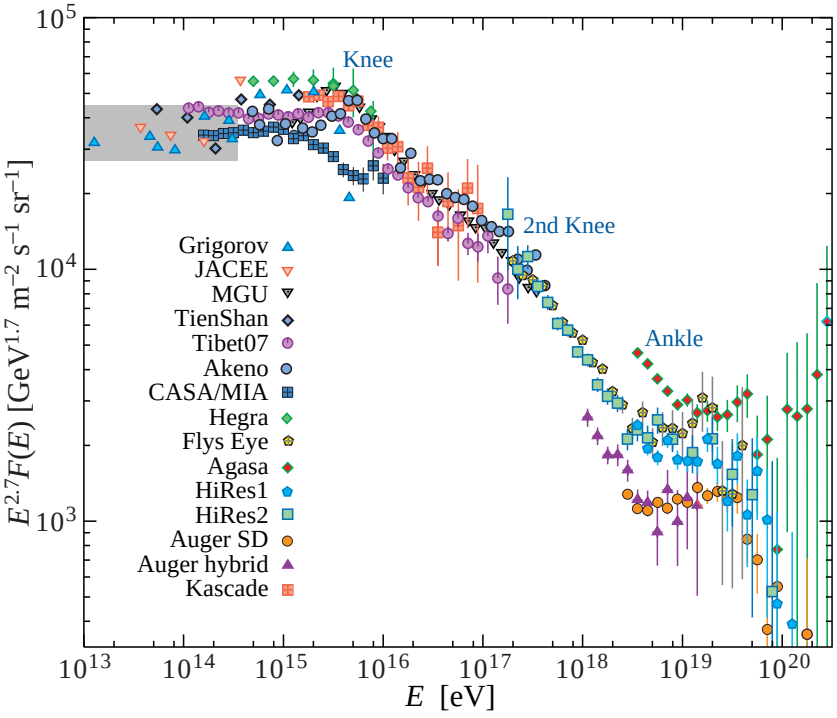


Figure 4: Energy spectrum of the primary cosmic particles is a main mystery in cosmic ray research (by Particle Data Group [6])

It is easy to see from Fig. 4 that the flux decreases as the energy of a particle increases. As mentioned in the section 1.2 the cosmic primary particle flux at the Earth is about 1000 particles per square meter per second, which is the flux of the particles with an energy from 10^{8} eV to 10^{10} eV. But the flux of the particles with an energy of 10^{19} eV is only a few particles per square kilometer per century [15]. When considering Fig. 4 more carefully, the features of the spectrum shape can be perceived. The slope of the spectrum changes approximately at the $3 \cdot 10^{15}$ eV. This

change of the spectrum is called knee. On this energy range the flux starts to diminish more steeply, when particle energy increases. The other change of the slope occurs on the energy 10^{18} eV, which is called the ankle. Anyway, the whole spectrum seems to be quite smooth with no sharp peaks.

The origin of the spectrum shape is not clear and especially the knee is a mystery. The softness of the spectrum could indicate, that only one particle production mechanism exists [45], which actually does not seem to be the truth. The most likely acceleration mechanism inside a galaxy is believed to be the supernova chock waves [45]. But the knee is commonly thought to represent the maximum acceleration, which a galactic supernova is able to give to a particle [30]. If the supernova shock wave is not the origin of the higher energy particles, also other acceleration mechanisms should exist. Thus there is no consensus which mechanism affects the knee in the spectrum. Nevertheless the particles with energies higher than the ankle are considered to likely be extragalactic particles.

1.4 Origin of the knee

In 1958 Kulikov and Khristiansen were detecting an electron number spectrum in the air shower [30] and they discovered the knee. As mentioned above, the supernova is the most likely providing the acceleration mechanism of the cosmic particles and many models suppose supernova to be the origin of the cosmic ray. But, it does not seem to be independently able to explain the knee. During the decades the knee has been tried to be explained by several models. Some models assume that the knee is a real feature of the energy spectrum, whereas other theories even speculate that the knee is just an effect when observing the air shower [29], not a real feature of the primary cosmic particles.

Many models predict that the knee originates from one or several acceleration prosesses and these models are usually associated with the supernovas. According to some models the cosmic particles are first accelerated to the knee energy by some mechanism, which is the most usually assumed to be the shock waves of the supernova explosion. Then other mechanism accelerate the particles more. One model supposes that galactic wind may re-accelerate the cosmic particles to the higher energy, which explains why the energy spectrum is smooth on

the knee region [32]. The galactic wind is believed to accelerate the higher energy particles up to the ankle energy [29].

The oblique shock model differs from the models, which suppose the supernova shock front to be parallel with the magnetic field. The shock front meets the randomly directed magnetic field lines [29], which makes the particles possible to be reflected or transmitted in shock fronts. Thus the oblique shock front makes the particle motion more complicated [21], but it allows that the particles are accelerated to the larger energies due to the reflections.

According to the cannonball model during the bipolar supernova explosion two baryonic plasma balls are shot to opposite directions. The theory assumes the balls to be accelerated to knee energies in the ultra-relativistic shocks in interstellar medium and the second order Fermi acceleration mechanism gives more energy inside the cannon ball [29]. The second order Fermi acceleration is based on the phenomena of the magnetic mirrors, which move randomly.

There are also theories, which presumes an acceleration mechanism to be able to give to a particle an energy higher than the knee. Those kind of theories explain that the spectrum shape above the knee is caused by the loss of the higher energy particles during the escape from the galaxy or via interactions inside a galaxy [32]. For example it has been hypothesized that the knee is due to interactions between magnetic fields and the cosmic particles. The knee has been also speculated to be originated from the interaction between cosmic particles and background radiation [29].

The very different way of considering the knee is represented by the theories, which suppose that new kind of interactions in an air shower transfer energy to the particles, which can not be observed by the air shower experiments [29]. Due to this ”particle loss” of the higher energy particles the measured energy spectrum includes the steepening on the knee region. This phenomena has been guessed to be associated with the supersymmetry or technicolor theories [29]. According to the measurements performed in LHC experiments there is no reason to pay attention to the ”new physics”, but the knee is just a real feature of the flux of the primary particles [13].

2 Early research of cosmic rays

The cosmic ray research started in the 20th century, when radioactivity was discovered (1869 by Becquerel) and some radiation detection techniques were developed. After the cosmic ray discovery the phenomena has been studied with several methods by using many kind of detectors making clear its nature. But still researchers don't know everything about cosmic rays.

2.1 Discovery of cosmic rays

Before the existence of the cosmic radiation was discovered, high penetrating radiation was observed on the Earth at the beginning of the 1900s and it seemed to appear everywhere. The radiation was detected with an electroscope in many places: on the ground, air, mountains and buildings. Although the detector was shielded, the radiation still seemed to be detected. Some believed that this radioactivity originated from the Earth surface or from the air [38].

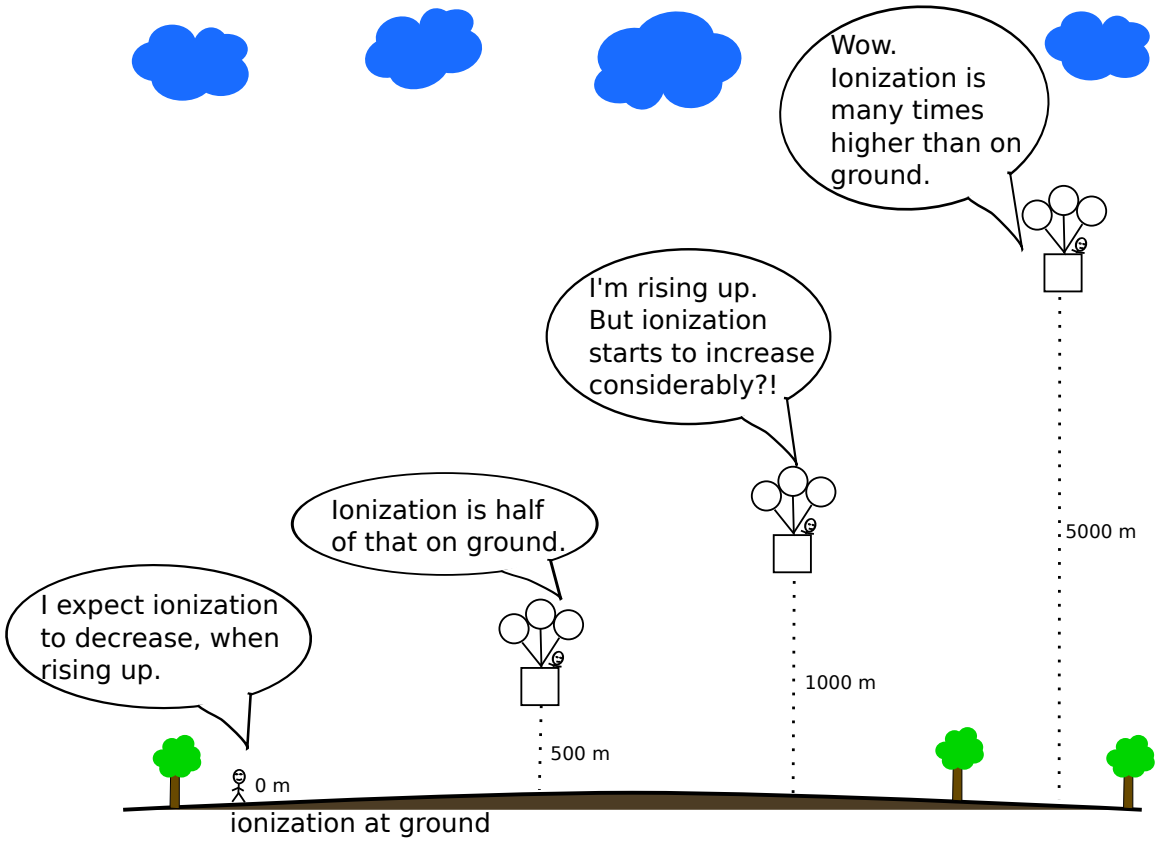


Figure 5: Victor Hess discovered cosmic rays and realized that it is originated from above of our atmosphere (by Tiia Monto, CC BY-NC)

Properly the cosmic radiation was discovered by Victor Hess in 1912. Initally Hess believed that at high altitudes the radiation should be very little, because he assumed that it was originated from the ground [40]. Hess performed radiation measurements with an electroscope in a balloon even at the altitude more than 5 km from the ground. As presented in Fig. 5 Hess measured the ionization on several levels and he found that ionization decreased up to 1000 m, just as he assumed. At the altitude of 500 m the ionization was only half of that on the ground level, but at the altitude of 1000 meters and above the ionization increases noticiably. Contrary to his expectation, at 5 km from the Earth surface the ionization was many times higher than on the ground level [38]. Based on this experiment Victor Hess concluded that there was a source of high-penetrating radiation above the atmosphere. Later Werner Kohlhörster performed same kind of experiments by going up to the altitude of 9300 m, and confirmed Hess's conclusion [34]. The measurements of Hess were performed both on daytime and at night, which confirmed that the radiation didn't come from the Sun.

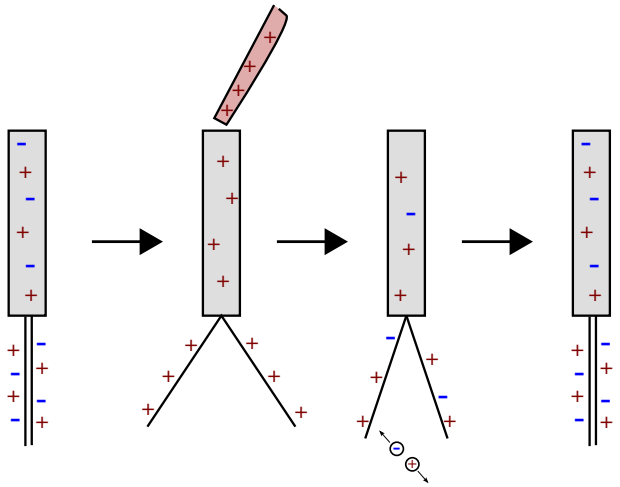


Figure 6: Hess used Wulf's electroscope in his experiments. First the system is charged purposely and the leaves repel each other. When a cosmic particle penetrates the electroscope, it may collide with a molecule breaking it into electron and ion. Then the electron or ion moves to the leave discharging the system, which makes the leaves to relax. (by Tiia Monto, CC BY-NC)

Hess used the electroscope developed by Wulf in his measurements to detect the radiation [56]. Basic operation of the electroscope is presented in Fig. 6. It includes two conducting leaves, whose position indicates the charge of the system. If there is no charge, the leaves are relaxed as presented on the left in Fig. 6. If the system is charged, the leaves aim to repel each other pointing at different directions. The radiation can be easily detected, because it discharges the system. When a particle penetrates to the gas of the detector volume, it may ionize a

gas atom. Consequently, the resulting ion and electron are attracted by the charged leaves. The electron and ion reach the leaves lowering their charge. When the radiation has caused collisions enough, the charge is lost and leaves are back at the relaxed position.

2.2 The nature of cosmic particles

The disvovery of cosmic rays by Hess was a denotative step in cosmic ray research, but it was not yet known, what the cosmic ray really is. The electroscope detects only the existence of cosmic ray, but it can't measure direction, energy or type of a particle. The term "cosmic ray" was invented by Robert Millikan in 1925. But Millikan believed mistakenly the cosmic radiation to be formed of very high energy photons, which were generated when lighter nuclei merge with each other creating heavier nuclei.

According to the cosmic ray measurements in 1932 carried out in different parts of the Earth by Arthur Compton, the radiation rate seemed to be higher near Earth poles than near the equador [40]. The phenomena implies that cosmic rays include charged particles, because Earth's magnetic field affects them. If cosmic rays consisted only of photons, their path wouldn't be deflected by the electromagnetic fields. Now the cosmic rays were determined to include charged particles, but it was not known if they carry positive or negative particles.

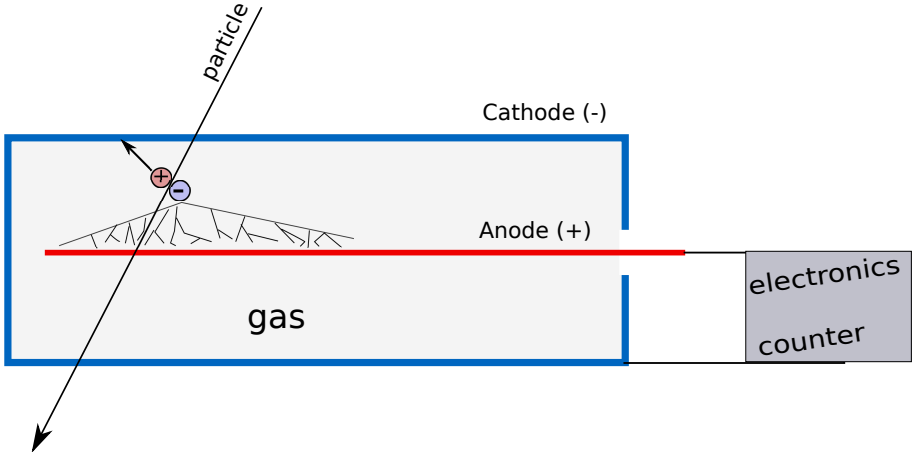


Figure 7: The wall of the Geiger counter is the cathode and inside is a anode wire. A particle penetrates through the tube and ionizes a gas atom. The electron (blue circle) aims to reach the anode wire and the ion (red circle) goes to the cathode. The electric field accelerates the electron, which knocks other molecules generating widened electron avalanches. This causes a signal. (by Tiia Monto, CC BY-NC)

Geiger counter was invented in 1928 by Hans Geiger and Walther Müller

[12] and it was an important innovation in cosmic ray research. Actually the principle of this device was developed already in 1908. Geiger counter made it possible to measure the direction of cosmic particles. The basic operation of Geiger counter is presented in Fig. 7. It is composed of a cylindrical negatively charged cathode and a positively charged anode wire in the center, in parallel with the tube. The electrodes create a strong electric field inside the tube, which is filled with a low-pressure gas. The Geiger counter operates such that the particle penetrates through the tube and ionizes a gas atom, which creates an electron-ion pair. The strong electric field pushes the electron towards the anode wire and the ion towards the cathode wall. The electron is accelereted and hits other atoms in the gas generating a widened avalanche of electrons. The large avalanche produces an electric pulse, which activates the counter. This detector is sensitive, because already a small amount of ionization may generate a signal [12].

Werner Kolhörster and Walter Müller used Geiger counters wisely by setting two counters parallel. When a particle penetrates through both of the counters, two signals appear at the same time [10]. This is a way to define the direction of a cosmic particle.

In 1933 Bruno Rossi measured cosmic rays with Geiger counters connected in a triangle plugged into the signal registering electronics. In that kind of detector composition one single particle can't penetrate all the detectors. Nevertheless many simultaneus signals were observed, which means the particle showers exist [12]. This is because the primary particle produces a great number of the secondary particles when colliding with the atmosphere as described in section 1.2.

2.3 The type of cosmic particles

Cloud chambers made it possible to examine the track of the ionizating particle and even to take photographs of them. The cloud chamber was invented by Charles Wilson, albeit his initial intention was to simulate cloud formations [4]. The cloud condensation is caused by expanding rapidly the saturated air volume, which lowers the temperature and makes the air supersaturated [28]. As presented in Fig. 8 radiation ionizes the supersaturated gas and the water vapor is condensed into droplets along the path of the ionizating particle. The droplets can be observed visually by an eye. Wilson invented the principle in the 1890s,

but built the first actual cloud chamber just in 1911 for particle detection [25].

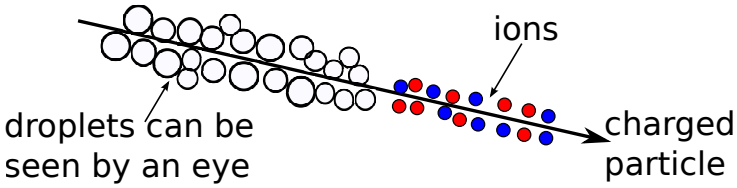


Figure 8: A charged particle (long arrow) penetrates through a cloud chamber and collides with the gas molecules ionizing them (blue and red balls). The vapor near ions condense into droplets (white balls), which form a visible path of the charged particle. (by Tiia Monto, CC BY-NC)

Carl Anderson built an improved cloud chamber over that of Wilson to detect cosmic particles in a strong magnetic field. Instead of water Anderson used water-alcohol mixture, which causes the clearer images of the particle paths. The measurements showed that there occurred both negatively and positively charged particles [48]. As mentioned in section 1.1 the Lorentz force for a negatively charged particle is oppositely directed than for a positively particle, which means that negatively and positively charged particles turn to different directions in a magnetic field. Anyway, during that time protons were the only positive particles known, but the positive particles in Anderson's experiment seemed to have mass similar to an electron. Thus in 1932, Anderson discovered a positron, which is actually an anti-electron. The experiments by Patrick Blacket and Giuseppe Occhialini showed that these positrons were originated from cosmic rays.

Now it was known that cosmic rays on the ground surface consist at least of electrons and positrons. In 1937 Anderson and Neddermeyer discovered with a cloud chamber a new cosmic ray particle, whose mass was less than that of a proton but more than that of an electron. Now the particle is called muon.

The primary cosmic ray particles are able to be detected directly only at high altitude. In 1948 a balloon experiment by Phyllis Freier et al. showed that cosmic rays contain also heavier particles than proton [20], including elements from helium to iron. In those experiments the Ilford nuclear emulsion and cloud chambers were used such that the nuclear emulsions piles were placed vertically below and above the cloud chamber.

The nuclear emulsion is a kind of photographic plate composed of silver

bromide and it is built to detect cosmic radiation. The emulsion sheets are attached together to form a solid block. It works such that ionizing radiation penetrates the emulsion and electrons are liberated. A free silver ion in silver halide crystal can move through the lattice. And the silver ion encounters a trapped electron creating a metallic silver. When silver atoms accumulate enough, a stable nucleus is created and a latent image centre formed. In development the latent image is made visible and it shows the track of the particle.

3 Cosmic ray research

Some ”primitive” methods to study cosmic rays are described in chapter 2. In this context the term ”primitive” means the techniques, which just provide the information about existence of radiation, but not the detailed features of the particles. For example, the electroscope used by Victor Hess gives a clue about the ionizing particles penetrating through the device. To reach more detailed information (particle type, energy and distribution) about cosmic rays one needs to use more sophisticated detector systems, for example air shower arrays.

Generally speaking cosmic ray detecting methods can be divided into direct and indirect measurements. In the direct measurements the primary cosmic particles with energy below the knee (up to 10^{14} eV) can be detected directly above the atmosphere with balloons or satellites. In the indirect measurements secondary particles can be observed with the ground based devices. Weak points of the direct measurements are the limited detection area and the measurement time [45]. Actually the higher-energy particles are almost impossible to be detected directly above the atmosphere due to their low flux. Whereas the indirect measurements on the ground makes it possible to install the detectors on a large area and make long-time measurements.

Additionally the indirect measurements can be divided into three different methods: Cherenkov detectors (observing Cherenkov light emitted by charged particles), fluoresence detectors (observing fluoresence light emitted by atmospheric nitrogen excited by charged air shower particle) and shower arrays (observing shower size and lateral distribution) [7]. The air shower arrays consist of several detectors spread into a large area.

3.1 Detector types

One way to categorize the particle detectors is to divide them into three main groups as shown in Tab. 1: gaseous detectors, solid-state detectors and visual detectors [53]. This list is not perfect and it does not include all detector variants.

The visual techniques mentioned in Tab. 1 consist of detectors, which provide visual information about the radiation and this kind of methods

Table 1: Types of the particle detectors (by [53])

Gaseous detectors	Solid-state detectors	Visual Techniques
-ionization chamber -proportional counter -Geiger-Müller counter -Spark chamber	-scintillation counter -solid-state semicond. counter	-cloud chamber -bubble chamber -diffusion cloud chamber -nuclear emulsion -Cherenkov detector -liquid semicond. detector

don't need much of the electronics. For example the cloud and bubble chambers are filled with a fluid, in which a visually detectable track along the path of the radiation is created. Most of the visual detectors are not generally used today and they are replaced by gaseous detectors [41]. Detectors equipped with electronics are more proper to transfer measurement data, which may be collected and managed by computers. This section discusses in more details the scintillation detectors and gaseous detectors, because they are part of the EMMA experiment. These two detector types also allow to construct economically large detector areas important in cosmic-ray detection.

3.1.1 Scintillation counter

Scintillation counter generally consists of a scintillator, photomultiplier tube (PMT) and electronics as presented in Fig. 9. When radiation

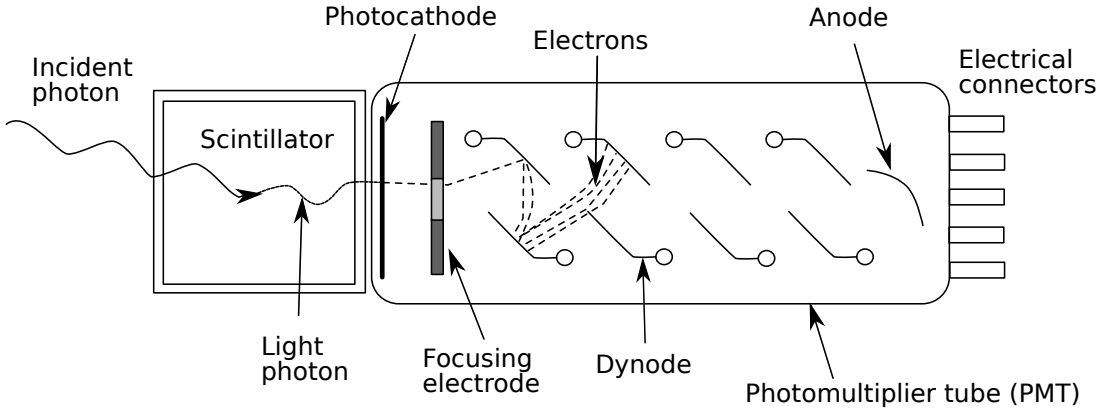


Figure 9: Scintillation counter operation (by Arpad Horvath, public domain)

penetrates to the scintillation medium, light or UV photons are created. These photons go to the photocathode, where an electron is produced. The electron is directed into a dynode, where more electrons are released. The group of electrons reflects from dynode to dynode and the electron number is multiplied until the electron group collides with the an-

ode at the end of the PMT. The pulse height detected by the electronics is proportional to the energy of the initial radiation particle. This detector type can provide goos time resolution (order of ns). The position resolution depends on the size of the scintillation medium and it varies from mm to a pixel size.

3.1.2 Gaseous detectors

The purpose of the gaseous detectors is to measure the particle's position. Four types of gaseous detectors are listed in Tab. 1 and they have in common that the detection is based on the ionization inside the gas medium. The spark chamber can be defined as oldfashioned detection technique and this section treats only of the three other types of gaseous detectors: ionization chamber, proportional counter and geiger counter (mentioned in section 2.2).

The gaseous detectors discussed here have a similar basic structure: a chamber with a cathode wall is filled with gas and in the center is an anode wire. The radiation penetrates to the detector and hits a gas molecule. Molecule is separated into electron and ion forming an initial ionization. The strong electric field forces the positive ions to the cathode and electrons to the anode creating a detectable signal current to the electronic circuit.

These detectors have different voltage between the electrodes. The operation regions (the relation between voltage difference and signal strength) of the gaseous detectors are presented in Fig. 10. The logarithmically scaled vertical axis is the collected charge and linearly scaled horizontal axis is the potential difference between anode and cathode. The more voltage is applied, the more ions are collected. If the voltage is zero, the signal does not occur, because the ions recombine [53]. The operation on the molecular level of an ionization chamber, a proportional counter and a geiger counter are described in Fig. 11. Actually the voltage affects the occurence of the electron avalanches. The more electron avalanches occur, the stronger signal is detected by the electronics. The creation avalanche requires that the voltage is high enough, so that the electrons from the initial ionization collide with the gas mocelules releasing more electrons. The secondary electrons ionize further more the gas molecules forming the electron avalanche widening towards the anode.

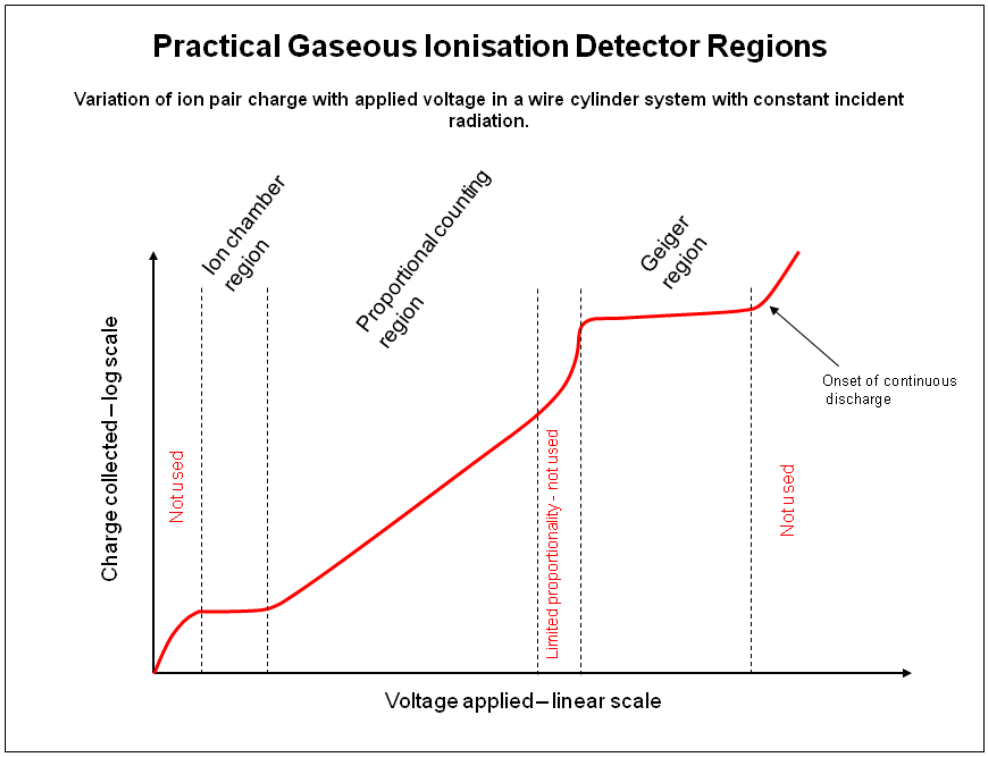


Figure 10: The diagram presents the relation between ions collected by electrodes and the applied voltage for different gaseous detector types (by Doub Sim, CC BY-SA 3.0)

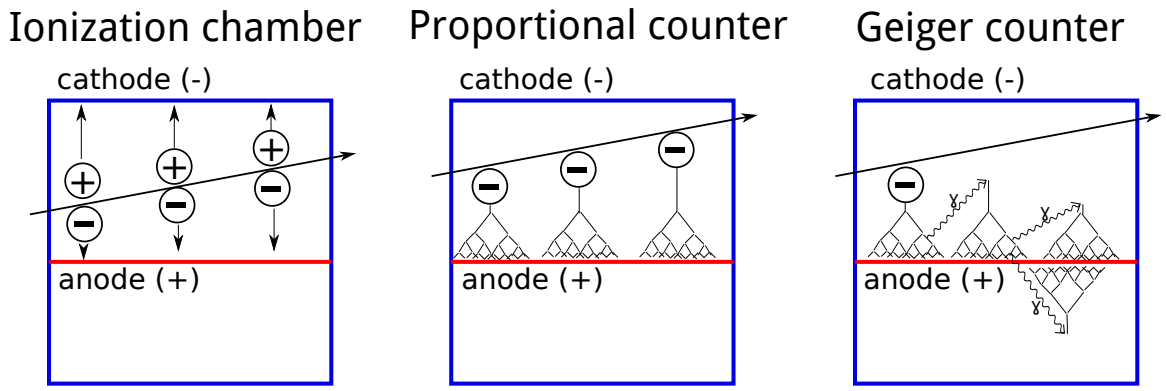


Figure 11: In **ionization chamber** (left) electrons and ions go directly to the elctrodes. In the **proportional counter** (center) electrons product electron avalanches. In **geiger counter** (righ) electrons product electron avalanches, which are multiplied by the photons emitted by the avalanches. (by Tiia Monto, CC BY-NC)

The case of the ionization chamber is presented as the horizontal part on the left of the curve in Fig. 10. The voltage difference between the electrodes is set to a certain value (more than 10 volts) such that almost all the ions reach the electrode without recombining [53]. The electron ion pairs from initial ionization fly directly to the electrodes without creating an electron avalanche as can be seen on the left in Fig. 11. The initial particle's energy is proportional to the signal current, which makes it possible to observe whether the particle had low or high energy. This

detector can't detect the particles separately, if they are coming in an intervall less than a millisecond [53].

The proportional counter region in Fig. 10 is the long, linearly rising part of the curve. Now the voltage difference is more than in ionization chamber but less than in geiger counter. As is shown in the figure, the stronger the voltage the more particles are collected. Near the anode the electrons from the initial ionization gain energy enough and collide with the gas molecules producing electron avalanches as presented on the center in Fig. 11. The avalanche happens until all the electrons near the avalanche volume are detached from the molecules and forced to the anode [47]. Due to this amplification effect one ion pair may be multiplied into 1–1000000 "new" ion pairs [47]. Because the multiplication is directly proportional to the primary ionization, which actually depends on the primary particle energy, it is possible to observe the particles with different energies.

The geiger counter region is showed as a very slowly rising (almost horizontal) part on the right of the curve in Fig. 10. In geiger counter the electron avalanche is self-multiplied. The avalanches produce photons, which collide with the gas creating more avalanches spreading along the length of the anode wire. The positive ions are produced so much that the voltage near the anode is lowered. This effect stops the ionization [47]. The ionization stops always, when a certain number of avalanches is produced. That's why the signal is constant and it does not depend on the primary ionization energy [53], thus the geiger counter may observe the single signals, but not the energy of the particles.

3.2 Research with muons

The muon is one of the leptons and an elementary particle, which means it does not have an internal structure. Actually it is partly similar to the electron. Muon (μ^-) has an electric charge e^- and antimuon (μ^+) has an electric charge e^+ being equivalent to that of an electron absolute value. The spin of muon is $\frac{1}{2}$, which equals the spin of the electron. But muon's mass is 105.658 MeV, which is more than 200 times the electron's mass. The muon lifetime is approximately 2 microseconds, which usually means that they have to have relativistic energy in order to be studied on the Earth.

Although an air shower includes many kind of particles, muons are a good choise to be studied. They are created mainly in pion and kaon decays near the primary particle collision position, and actually there is a relation between the muon energy spectrum at the sea level and the pion source spectrum. At the sea level about 80 % of the charged component of the air shower are muons [24]. Muons lose usually 2 GeV of their initial energy while penetrating through the atmosphere to the Earth ground [7], but still they are the most energetic part of the air shower at the sea level.

The air showers are not identical even in the case of the primary particles of the same mass and energy [7]. The lateral distribution and number of the secondary particles at the ground implies the primary particle's energy and mass [33]. Because muons arrive as bundles, it is possible to measure them simultaneously. The large number of muons is needed to be detected to define the muon lateral distribution. The advantage to detect muons is that they may penetrate the medium with minor energy loss and there are no background muons, because the cosmic rays are the only natural source of muons on the Earth. Muons are detected mainly with scintillation counters and gaseous detectors (for example drift detectors).

Muons are highly penetrating particles and they are easily able to be detected both on the Earth surface and underground. Also neutrinos penetrate deep underground, but they are hard to be detected. The rock filters all other particle types and mainly only the high energy muons reach the underground detectors. Actually by setting the depth one can define the cut-off energy of the observed muons. Fig. 12 shows the muon intensity in different depths in unit of kilometers of water equivalent (km w.e.; i.e. 1400 m of rock is equivalent of 4 km.w.e.). The disadvantage in muon measurement is that the flux decreases quickly as a function of the depth. On the other hand one must know the rock structure and density above the detectors to estimate the energy loss of the muons. In the case of the underwater measurements the medium density may be easier to estimate [11].

When muons move through material, they lose energy via ionization, bremsstrahlung, electron-positron pair production and electromagnetic interaction with a nuclei [11]. When a muon penetrates a medium with thickness X, the total energy loss of muon E_μ is expressed in the equa-

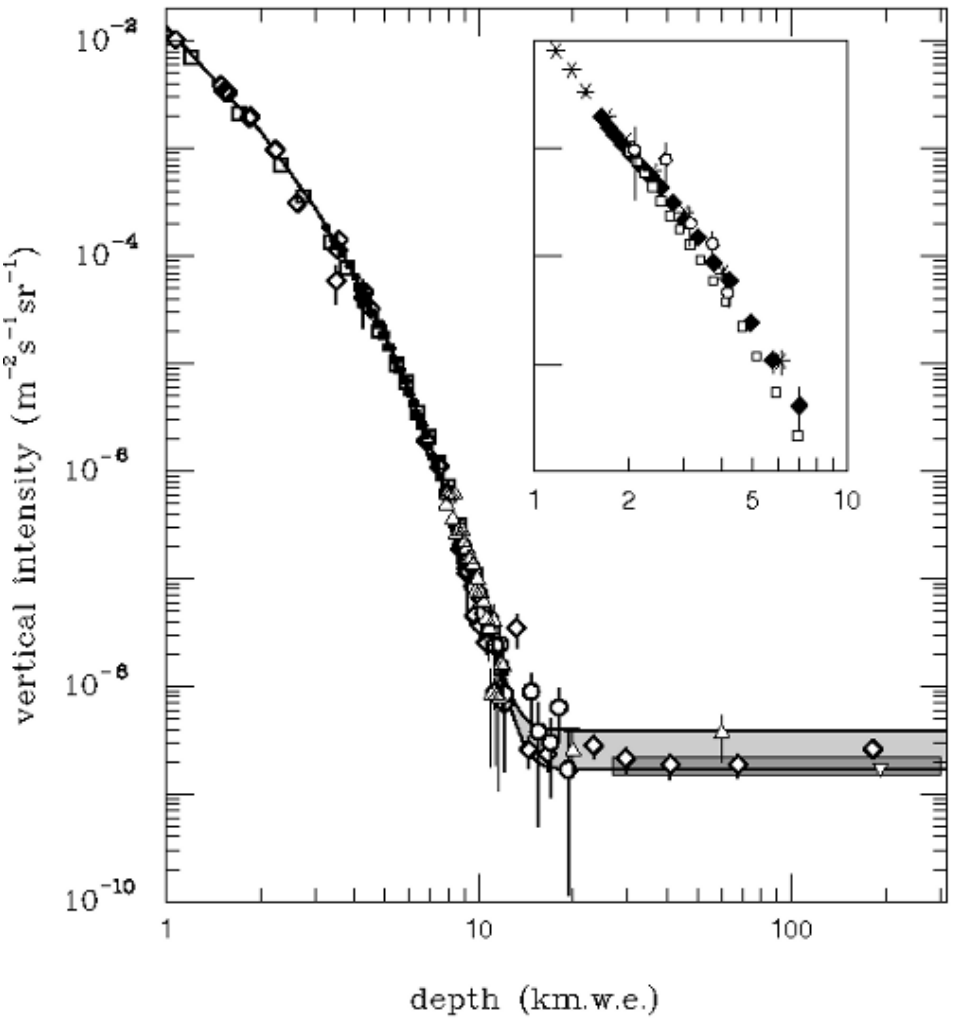


Figure 12: Muon intensity in different depths plotted with experimental data (by Particle Data Group [8])

tion

$$\frac{dE_\mu}{dX} = -A - (B_{brem} + B_{pair} + B_{elec})E_\mu, \tag{1}$$

where A is energy loss via ionization, B_{brem} expresses the ratio of energy loss in bremsstrahlung, B_{pair} in electron-positron pair production and B_{elec} in electromagnetic interactions with nuclei. The thickness is different for different mediums. The chemical composition of medium affects a bit on the factories A and B, but often they can be expressed as $A \approx 2$ MeV g^{-1}cm^2 and $B = B_{brem} + B_{pair} + B_{elec} \approx 4 \cdot 10^{-6}$ g^{-1}cm^2 [11]. So the energy loss of the muon reaching a certain depth may be approximated as $\frac{dE_\mu}{dX} \approx -A - B \cdot E_\mu$. This equation is valid for the minimum ionizing particles (MIP), which means that their energy loss rate is nearly minimum and constant. The particles with small energy E_μ are not MIP and then the parameter A depends heavily on the energy E_μ. Anyway the cosmic ray muons are considered to be MIP.

3.3 Cosmic ray experiments

This section focuses on a few indirect cosmic ray experiments on the Earth surface, underground and under ice. The underground experiments with different research motivation have been in operation since the late 1970's and early 1980's [23]. Some of the collider experiments, for example, are also able to detect the cosmic muons and on the other hand there are experiments just specialized in measuring cosmic rays.

Pierre Auger Observatory is a cosmic ray experiment situated in Mendoza Province, Argentina. It studies ultra-high-energy neutrinos and cosmic rays by detecting air showers with a large array spread to an area about 3000 km^2. The detectors are located on the ground at the altitude of 1400 meters above the sea level. The detector system consists of about 1700 Cherenkov detectors, which are surrounded by 4 fluorescence detectors [2]. The Cherenkov detectors are cylinder-shaped with 3.6 m diameter and filled with pure water. When particles enter the detector, they emit Cherenkov light, which is detected by the photomultiplier tubes. The signal is then transmitted wirelessly for further processing.

KASKADE-Grande experiment in Forschungszentrum Karlsruhe, Germany stopped data collection in 2011, but all data is not yet analysed. It studies cosmic rays with the energy range from 10^{16} eV to 10^{18} eV by detecting air shower particles. KASKADE-Grande was the most sophisticated detector for studying cosmic rays between the knee and ankle regions, and consisted of hadron calorimeters, scintillation detectors and limited streamer tubes. The experiment is able to observe muonic, hadronic and elecromagnetic components of the air shower and measure the shower size and muon number. The results from the experiment give detailed information about the structure of the energy spectrum. Actually part of the limited streamer tube detectors are now in use in the EMMA experiment.

MACRO (Monopole, Astrophysics and Cosmic Ray Observatory) was an underground cosmic ray experiment at Gran Sasso Laboratory in Italy and it was operating between the years 1988 and 2000. In addition to searching for the magnetic monopoles the MACRO studied also features of cosmic rays in the knee region. The MACRO array in a cavern at the depth of approximately 1.2 km consisted of scintillation counters, limited streamer tubes and nuclear track etch detectors. The rock filters the low-energy part of cosmic-ray induced shower and only the muons with

energy more than 1.3 TeV are able to reach the detectors [9]. The experiment collected data about the muon bundles by detecting the rate and distribution of bundles.

DELPHI (DEtector with Lepton, Photon and Hadron Identification) was one of the detector stations of the former LEP (Large Electron–Positron Collider) experiment situated in the underground tunnel at CERN. DELPHI was primarily a particle physics experiment but it also measured the air shower muon bundles at the depth of 100 meters. In order to reach this level a vertical muon needs an energy of 52 GeV. Due to the limited measuring time the experiment was able to study the primary particles with energy between 10^{14} eV and 10^{18} eV [55]. DELPHI observed muon multiplicity distributions by using cathode redout of the calorimeter, time projection chambers and muon barrel chambers. Now the muon barrel chambers are in use in the EMMA experiment.

IceCube is a neutrino observatory at the South Pole. The detector system inside the ice is one cubic kilometer. It is composed of 5160 digital optical modules (DOM), which are installed at the depth from 1450 m to 2450 m forming 86 vertical strings [22]. The basic idea is that neutrinos interact with the ice and produce charged particles, which create Cherenkov radiation. The photomultiplier tubes of the DOM detect the Cherenkov light and send the signal up to the surface via cables. The detectors inside the ice can detect both muons produced during the interactions of the neutrinos and the muon bundles created in the air-shower in the atmosphere. The vertical air-shower muons should have an energy of 400 GeV or more to be observed by the detectors in the ice.

4 EMMA experiment

EMMA (Experiment with MultiMuon Array) is an underground experiment which studies cosmic rays by measuring the high-energy muon component of an airshower [36]. The intention is to study the composition of the primary particles with an energy between 10^{15} eV and $10 \cdot 10^{15}$ eV [37]. The experiment is operated by CUPP (Center for Underground physics in Pyhäsalmi) and the measurement station is located in Pyhäsalmi mine, Finland. The EMMA experiment is being developed and built by the cooperation of University of Oulu, University of Jyväskylä and Russian Academy of Sciencies. The idea of such an experiment was presented in 1999 and the building process of the EMMA experiment was started in 2005 [50].

EMMA observes muons which are created in the upper part of the atmosphere in pion and kaon decays as explained in section 1.2. The muons of the air shower form a bundle. By detecting a muon bundle one may discover the features of the primary cosmic particle [54]. Since the main detector system is being built at the depth of 75 meters (210 m.w.e) [36], the rock filters the air shower particles such that only muons with energy above 50 GeV are detected. At this depth the muon flux is about one muon per square meter per second.

EMMA is expected to measure the muons separately and its purpose is to observe the multiplicity, lateral distribution and the arrival direction of them. The position resolution of the EMMA experiment is good (about 1 cm^2). Because geometry of the EMMA array is able to localize the muon shower core. The accuracy of evaluating the core position is affected by the detector size, detector number and distance between them. The ability to detect the lateral distribution of the muons individually makes the EMMA experiment different from the previous underground experiments [36].

4.1 The Pyhäsalmi mine and the detector array

The Pyhäsalmi mine is situated in Pyhäjärvi, Finland and it is owned by Inmet Mining Corporation (Canada). It is the deepest metal mine in Europe (1.4 km deep). The intention of building the EMMA experiment in the Pyhäsalmi mine is to filter the low-energy air-shower particles and

detect just the high energy (>50 GeV) muons. But building a detector array underground sets some limitations compared with building on surface. For example, it would have been more expensive to build a tunnel at spesific depth just for the EMMA experiment. The Pyhäsalmi mine allowed the EMMA experiment to be operated in an already existing tunnel (in Fig. 13), which is not anymore in use of the miners. Additionally, the environment in the mine (dropping acid water and high air humidity [50]) need to be taken into account when designing the structure and the materials of the EMMA experiment. The undergrond conditions also affected in choosing the gas content of the drift chambers such that methane was not considered for the safety reasons [50]. On the other hand, the cavern keeps the outside temperature of the experiment approximately constant (10 °C) over the whole year.

Figure 13: Measurement station C at the depht of 75 m in Pyhäsalmi Mine on May 16, 2013. (by Tiia Monto, CC-BY-NC)

The EMMA array consists of three detector types: drift chambers, limited streamer tubes (LST) and scintillation detectors. Next sections discuss especially the drift chambers, but also limited streamer tubes and scintillation detectors are presented shortly. The drift chambers are the most important part of the EMMA array and they are used more versatile than other detector types, because there are plenty of them. Due to theirs excellent position resolution the drift chambers are used to measure the direction of muons. Additionally, the drift chambers are able to measure the muon multiplicity, but they may be saturated at high multiplicities. Scintillation detectors can measure the muon multiplicities also at high multiplicities due to their small size and they provide an initial value of the arrival direction making it easier to track muons. The LST detectors are used in the stations at the depth of 45 m and at the outermost stations at the depth of 75 m measuring the widening of the shower (there the position resolution does not need to be as good as in the central stations).

There is also data acquisition system, which makes it possible to collect the measured data electrically from the detectors. The gas needed by the drift chambers and LST detectors is delivered using a 100 m long pipe line. The gas container at the surface is currently a 600 l LAr Dewar. The pressure at the pipe line is between 3 to 4 bar and it is lowered to approximately 1 bar when it enters to the detectors.

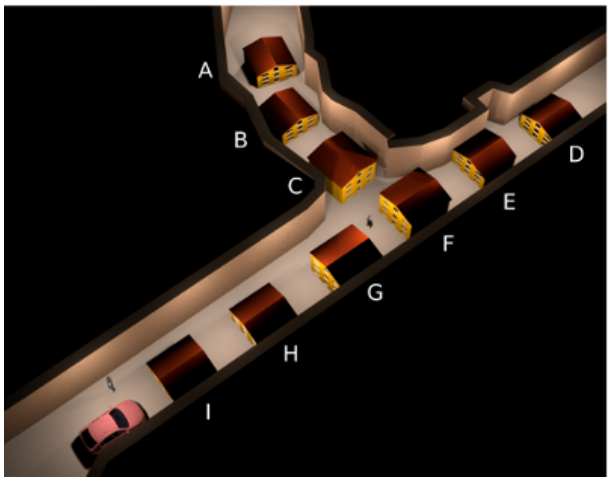


Figure 14: 3D modelling of placing nine EMMA detector stations (”cottages”) in the tunnel at the depth of 75 meters (modified version of the original figure by Jari Joutsenvaara)

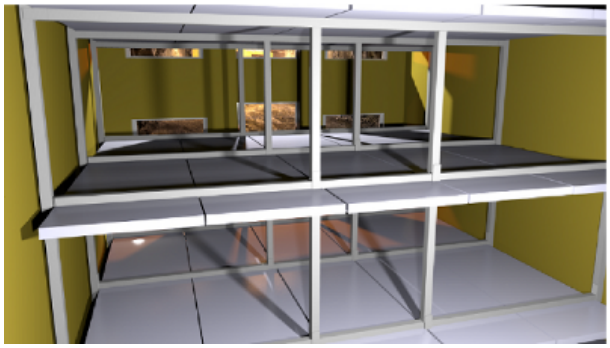


Figure 15: 3D modelling of the placing the planks inside a three layer station. Each layer is consisted of five planks set side by side. (by Jari Joutsenvaara)

The detectors are being installed at the detector stations, which remind small ”cottages” with plywood walls and each of those has a detecting area of about 15 m^2. The picture of the station C is shown in Fig. 13. The detectors in the stations form layers and the distance between the overlapping layers is 1.1 meters. The 3D modelling of the layout of nine stations (A, B, C, D, E, F, G, H and I) in the tunnel at the depth of 75 m is shown in Fig. 14. They are installed in the shape of letter ”T” forming three-arm detector system. The outermost station is about 30 meters from the midmost station and the distance between the stations is about 10 meters. Additionally, a few more stations (not shown in the figure) are

being built at the depth of 45 m forming the fourth hand of the detector array (stations X, Y and probably Z). The building of those additional stations have been started in 2013.

At end of June 2013 there are total of 9 detector stations fully constructed: 7 at the depth of 75 m (A, B, C, D, E, F and G) and 2 at the depth of 45 m (X and Y). At the stations C, D, E, F and G the planks are already collecting data. Additionally the scintillation counters at the station F are performing test measurements. The stations H and I will be constructed probably during this year. The construction of the station Z depends on whether it will be needed or not.

The stations are cathecorized according to the number of the layers of the gaseous detectors (planks and limited streamer tubes) [50]. Three of the stations (C, F & G) are composed of three plank layers and scintillation detectors. All the other stations are planned to include two layers, which are formed of the planks and / or limited streamer tubes. The 3D modelling of a three layer station interior is shown in Fig. 15.

4.2 Drift chambers

The drift chambers of the EMMA experiment were previously part of the LEP experiment at CERN. They were called Muon Barrel chambers (MUBs) and they situated around the collision point in the DELPHI detector system. Between years 2002 and 2005 the drift chambers were transferred to Finland and EMMA obtained in total 588 chambers (84 planks, each of which consisting of 7 chambers). A picture of two planks is shown in Fig. 16.

Figure 16: Two planks are being transferred at the surface laboratory of EMMA in 2011 (modified version of the original photo by Tomi Räihä)

The drift chambers are the most important part of the detector array. The role of them (and limited streamer tubes) in this experiment is to measure the position of muons, in which they penetrate the detector array. The position accuracy of the drift chambers is 1 cm^2, albeit they are quite slow detectors.

4.2.1 Structure of drift chambers

The inner dimensions (gas volume) of a standard drift chamber are $365 \times 20 \times 1.6$ cm^3. The standard chamber is 365 cm long, but the EMMA experiment has obtained also shorter chambers, which are 295 cm long [46]. The chambers are covered by aluminium walls, which are about a couple of millimeters thick so the outer diemensions of a chamber's cross-section are 20.8×2.6 cm^2, as presented in Fig. 17. The figure shows also how seven drift chambers are attached to each other to form so a called plank in which four chambers (X1, X2, X3 and X4) form the bottom layer and three chambers (Y1, Y2 and Y3) form the top layer. The dimensions of a plank are $85 \times 365 \times 6$ cm^3 with an area of 2.9 m^2 and an individual plank weigh of about 120 kg.

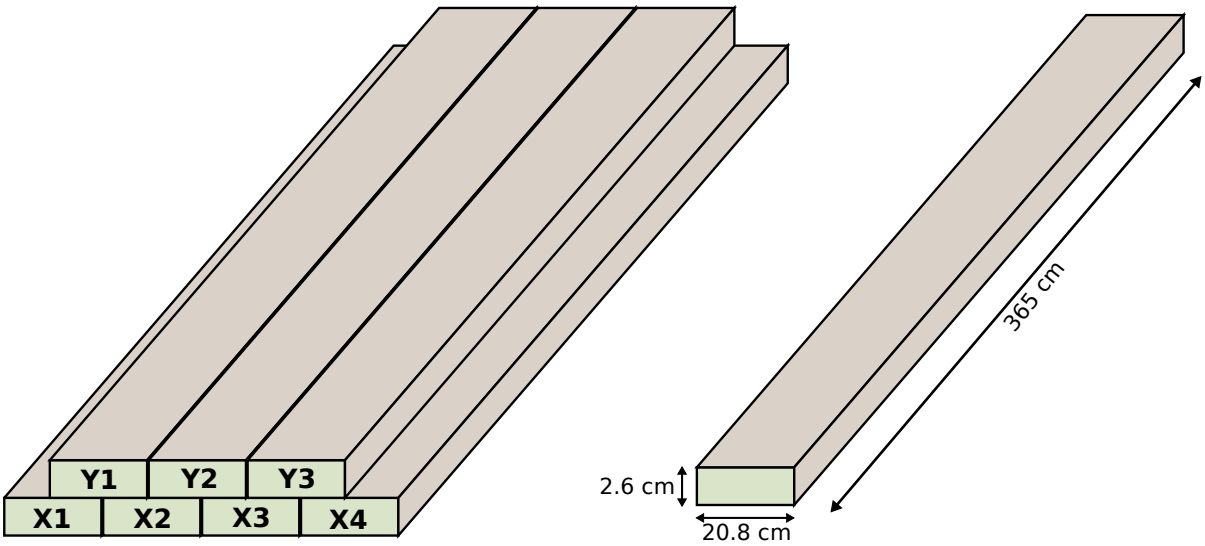


Figure 17: On the left side is a plank consiting of seven drift chambers and on the right side is a single drift chamber with outer dimensions (by Tiia Monto, CC BY-NC)

The inner structure of a drift chamber is shown in Fig. 18. Longitudinal cross-section is drafted on the bottom and transverse cross-section is drafted on the top. Both images show the anode wire marked as red, which is made of tungsten. The anode is installed on the center alingned with the longitudinal axis [46] and it is held up by plastic brackets ("spi-

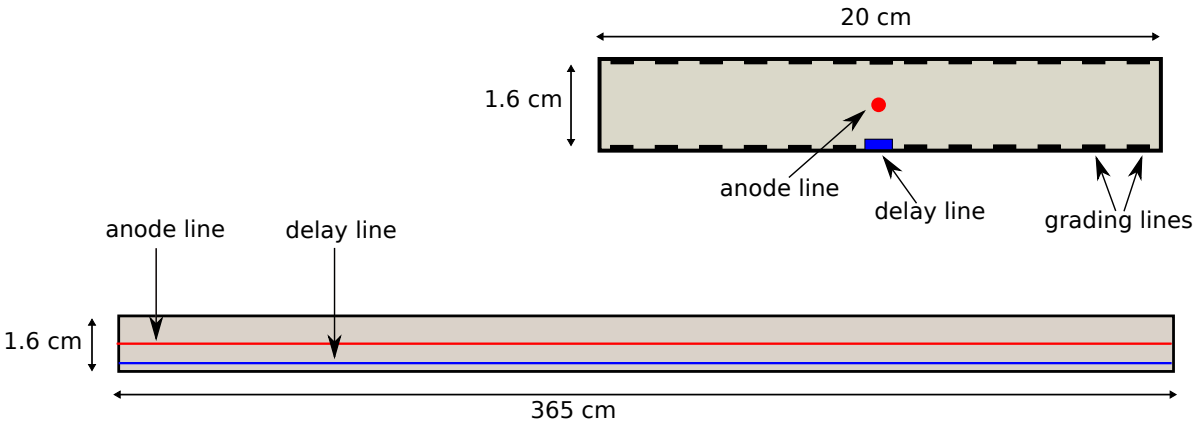


Figure 18: On top right is a transverse cross-section and on bottom is a longitudinal cross-section of a drift chamber's interior with inner dimensions. In both images anode wire is marked as red and delay line is marded as blue. (byt Tiia Monto, CC BY-NC)

ders"). The delay line (cathode) marked in blue is set on the center of the bottom wall parallel with the anode wire. It is enmeshed in a spiral so that the length is much more longer than the anode's length, which means the signal takes more time to transfer along the delay line. The upper image shows also the grading lines (cathode), which are placed both on bottom and top inner wall of the chamber.

Inside the chamber the grading lines provide a homogenous electric field, which is maintained by high voltage sources. The anode wire is connected to about 6 kV voltage source whereas the delay line is connected to about 4 kV voltage source. The voltage difference between the adjacent grading lines is about 572 V. The voltage of central grading line is about 3974 V and the outermost 542 V [46].

Through the chamber flows gas mixture, which consists of argon (92 %) and carbon dioxide (8 %) [36]. Currently the flow rate is such that one gas volume of a chamber is changed approximately twice a day.

4.2.2 Operation of drift chambers

The drift chambers operate in proportional mode [3]. The operation principle of the proportional counter is described in section 3.1. Contrary to a conventional proportional counter the drift chambers are not expected to be able to measure properly the energy left by muon, because the chambers are so thin and the gas inside them is not dense enough [18].

The signal producing process in this case is introduced in Fig. 19. All particle events (muon, electron or gamma) produce in total three signals

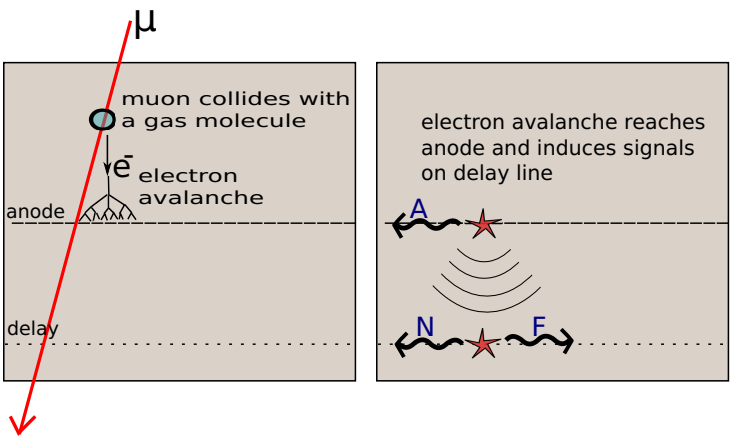


Figure 19: Muon generates three signals in a drift chamber: one from anode (A) initiated by electron avalanche and two from the delay line. The delay line signals N (near) and F (far) are induced when the electron avalanche reaches the anode wire. (by Tiia Monto, CC BY-NC)

(one anode and two delay line signals), from which the particle interaction position may be solved with an accuracy of 1 cm^2. When a muon penetrates a chamber, it ionizes a gas molecule producing a free electron (or it ionizes several gas molecules producing several electrons), which creates an electron avalanche near the anode wire. The electric field forces an avalanche to fly towards the anode wire producing an anode signal (A). The avalanche at the anode induces [3] signals ”near” (N) and ”far” (F) on the delay line. In the delay line N signal moves to near end and F signal moves to far end.

Since the signals (A, N and F) are created they go via the cables to the electronics to be processed before the signal information is stored in a hard drive. These signals are transferred through a pre-amplifier and a discriminator and then they reach a Time To Digital Converter (TDC).

In the EMMA experiment the prompt time of muon penetration (T_0) is measured with a scintillation detector. The anode signal (T_A) expresses the time at which the electrons reach the anode wire. Using these time values the distance between muon interaction and anode wire (x) may be calculated with the equation

$$x = \int_{T_0}^{T_A} v \mathrm{d}t, \tag{2}$$

where v is the drift velocity of the electrons inside the chamber [46]. Actually this equation does not spesify if the muon collision occurred on left or right side of the anode wire, but it may be found out when making the tracking [50]. The muon position in the chamber's longitudinal direction

can be measured by calculating the difference between near (N) and far (F) signals in the delay line [27]. When muon penetrates multiple overlapping planks, the direction of the muon may be solved by examining the signals as introduced above.

4.2.3 Plank calibration

Before using the planks underground they have to be calibrated. The calibration process is performed in a surface laboratory and the intention is to examine how accuratelly the planks are able to measure the position of the penetrating muon. A few planks are initially calibrated with a radioactive source (^{22}Na) and these are called as reference planks [36]. The calibration with the radioactive source had to be performed by measuring with the source located in 30 different position on every chamber. If all the planks were calibrated with this method, the calibration process would take too long time. That's why other planks are calibrated with atmospheric muons by using these pre-calibrated ones as reference planks. The calibration measurements with one plank stack consisting of six non-calibrated planks took about six weeks [36]. As presented in Fig. 20, the calibration stack is formed in total of ten planks so that every third plank (pink) is a reference plank and the others (grey) are non-calibrated planks.

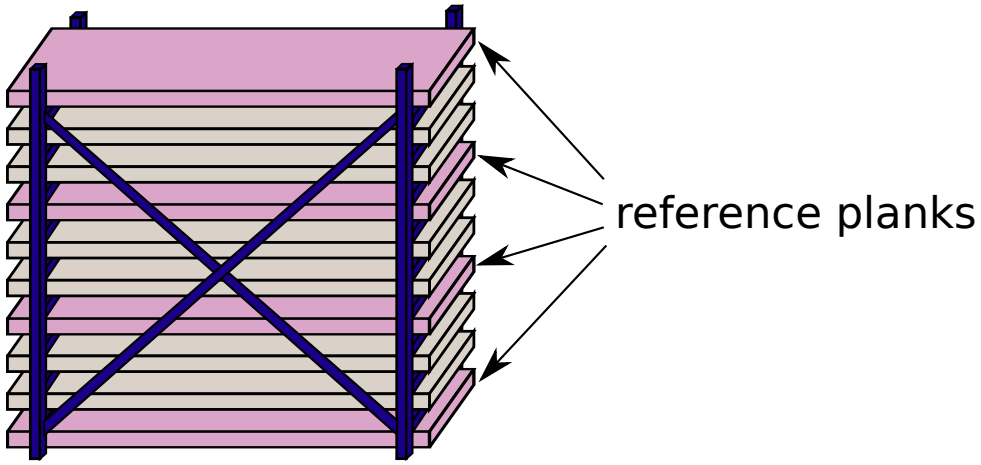


Figure 20: Calibration stack, the reference planks are marked as pink and non-calibrated planks are marked as grey (by Tiia Monto, CC BY-NC)

When a particle penetrates the stack, its track is reconstructed by using the signals from the reference planks (see section 6.1.1). The particle's hit positions in the non-calibrated planks may then be defined with a reconstructed track. Then the time signal from a non-calibrated plank

is compared with the actual hit position and this information is collected into position vs. time table [50].

The calibration measurements produce position vs. time tables for every plank. Those tables present time channel signals (time difference between near and anode or between far and anode) and the corresponding position of a particle in the direction of the delay. The table is consisted of 15 columns and 1200 rows. The left column is the time channel (one channel is 0.8 ns) and so there are total of 1200 time channels. Other 14 columns express the particle's distance between near and far end of the delay line of all seven chambers. The table is used such a way that first one checks the time signal from the chamber and then finds the row correspoding to the measured time signal in the left column of the table. In the same row the column of the corresponding chamber must be found and then the hit position value is read.

The calibration measurements of 84 planks took in total nearly four years and 20 calibration stacks were needed. In March 2013 the calibration coefficients (tables) for approximately 60 planks have been determined.

4.3 Scintillation detectors

The plastic scintillation detectors of the EMMA experiment are built by Russian Academy of Sciences, Moscow [37] and they were designed in cooperation with Russian Academy of Sciences, University of Oulu and University of Jyväskylä. Contrary to the drift chambers and limited streamer tubes these detectors are designed particularly for the EMMA experiment. The scintillation detectors are part of the central stations in the array as described in section 4.1 and their intention is to measure the prompt time of the arriving particle and their multiplicity. Time accuracy of these detectors is 2 nanoseconds [37].

An individual detector pixel has a volume of $12 \times 12 \times 3$ cm^3. One detector module SC16 (see Fig. 21) contains 16 pixels (4 on each side) and it is covered by a metallic box with an area of 0.5×0.5 m^2 and a mass of 20 kg. The EMMA array will contain in total 96 SC16 modules (1536 detector pixels). The signal is produced such that light is produced in the ionisation process in the scintillation crystal, optical fibre collects the light and then it is directed to avalanche photodiode [50]. Thus in this case photodiode replaces the photomultiplier tube, which is mentioned

in section 3.1.1.

Figure 21: Scintillation detector SC16 (Cropped version of the original photo by Tomi Räihä)

4.4 Limited streamer tubes

The limited streamer tubes (LST) were initially used to track muons in the KASKADE Grande experiment [26], which is introduced in section 3.3. In total 60 LST modules were bought by the EMMA experiment. As the drift chambers, also LSTs are specialized in measuring the position of the particle and thus they improve the experiment's ability to reconstruct the shower core.

An individual LST module is of the size 1×2.9 m^2 [26] and its outward reminds of plank. A module includes 6 LST chambers, each of them made of 16 LST tubes. A tube consists of an anode wire in the center and a cathode wall. Inside the tube flows CO_2 gas [50]. The operation is based on ionization of gas by the penetrating particle. The avalanche effect is limited due to the limited space inside the tube.

5 The measurements

The calibration of the drift chambers has been performed on the surface laboratory of CUPP and the calibration method is described in section 4.2.3. I studied only the calibration runs T5, T6, T7, T8, T9 and T10, where the measurements were performed during the years 2009 and 2010. Each of those stack has the same reference planks (P15, P39, P18 and P17). In some of the stacks, a part of the planks has been disqualified due to the broken chambers. The stacks with successfully calibrated planks are listed in Tab. 2 which also shows the date of the measurements. As can be seen in the table, the stack T9 contains most of the successfully calibrated planks and that's why T9 was chosen for my special project (in Finnish: erikoistyö). Whereas the stacks T7 and T8 seem to be the poorest ones with only two successfully calibrated planks. Actually T9 was the first stack which consisted of six planks to be calibrated.

Table 2: Calibration stacks with successfully calibrated planks. The stacks before T9 included four researchable planks, whereas the stacks T9 and T10 included six researchable planks.

stack	**planks**	**duration**
T5	P34 P35 P38 P78	Sep 4, 2009 – Sep 23, 2009
T6	P13 P21 P33	Sep 25, 2009 – Oct 26, 2009
T7	P20 P80	Nov 16, 2009 – Dec 16, 2009
T8	P2 P7	Dec 22, 2009 – Feb 16, 2010
T9	P1 P37 P47 P49 P51 P53	Feb 25, 2010 – Mar 25, 2010
T10	P41 P42 P45 P58 P84	Mar 31, 2010 – May 10, 2010

5.1 Layout of the planks

In the calibration stack the planks have been set horizontally in-line and vertically with respect to each other. This has been done by measuring the vertical distance between the topmost plank (P15) and the other planks. The distance has been measured with an accuracy of a millimeter in total for 12 points of each plank: three points (near end, far end and middle) on both sides (X1 and X2) on both walls (top and bottom). These position parameters are collected in a table.

For example the planks' positions in stack T9 are shown in Fig. 22. On the left are the points measured on the X1 side and on the right are

the points measured on the X4 side. The upmost row with turquoise background shows the horizontal distance from the near end and all the other number parameters give the vertical distance from the plank P15. If all the parameters in a same row are the same, the plank has been installed straight. Some planks are a bit aslant so that the height of the near end and far end may be different (at most a few millimeters). For my program I have calculated the average height of each plank by calculating the mean value of the six points measured on the bottom wall of a plank. The information about the planks' position is used for particle tracking.

X1 side

PLANK #	3550		1755		100
P15C	0 59		0 59		0 60
P1	179 239		179 239		179 239
P37	360 420		360 419		360 420
P39C	540 599		539 599		539 599
P49	717 777		717 777		718 777
P53	901 960		901 961		902 962
P18C	1084 1143		1081 1141		1082 1141
P47	1259 1319		1260 1320		1261 1321
P51	1440 1499		1440 1499		1439 1498
P17C	1621 1680		1620 1680		1621 1681

Top
Bottom

X4 side

PLANK #	100		1755		3550
P15C	0 60		0 59		0 59
P1	179 239		179 239		180 239
P37	359 419		359 419		360 420
P39C	539 599		539 599		539 599
P49	718 777		717 777		718 778
P53	901 960		900 960		901 960
P18C	1082 1141		1082 1142		1083 1142
P47	1260 1319		1259 1319		1260 1320
P51	1440 1500		1440 1499		1439 1498
P17C	1621 1681		1620 1680		1621 1681

← distance from the near end

Figure 22: Planks' position measured in millimeters from the upper edge of the plank P15 in stack T9

5.2 Measurement data files

The counting rate of a drift chamber on the surface is about 200 Hz and on the EMMA level at the depth of 75 m it is 100 Hz. On the surface about half of the counting rate is caused by muons, but on the EMMA level only 1 Hz is caused by muons. The rest of the signals is caused mainly by the gammas and probably the noise of the electronics. Anyway both on the surface and underground about 100 Hz is from noise [50]. Due to the high counting rate in the calibration the passage of a particle is saved in hard drive only if anode signal is produced in both the topmost plank (P15) and the under-most plank (P17).

The measurement data I was given was actually pre-handled. The initial files were one hour long, but Timo Enqvist merged them so that most of the files are about eight hours long. Most of the random signals were removed and my data contains only the particle events, which cause signal in each pre-calibrated plank (P15, P17, P39 and P18). Although the original data contains all the measured hits, my data contains only the first hit (first near, first far and first anode signal) per chamber in a particle shower. The signals are usually caused by the first particle of the shower but sometimes part of these signals may be originated from the earlier event. The ”wrong” signals are removed in tracking. Additionally all the events were removed except the ones in which the particle has penetrated all the planks vertically through the same chamber pair. Actually there are in total six overlapping chamber pairs (X1Y1, X2Y1, X2Y2, X3Y2, X3Y3 and X4Y3) as seen in Fig. 17. When choosing only the particles which went through the same chamber pair, the path is easier to reconstruct with 2-dimensional fitting equation.

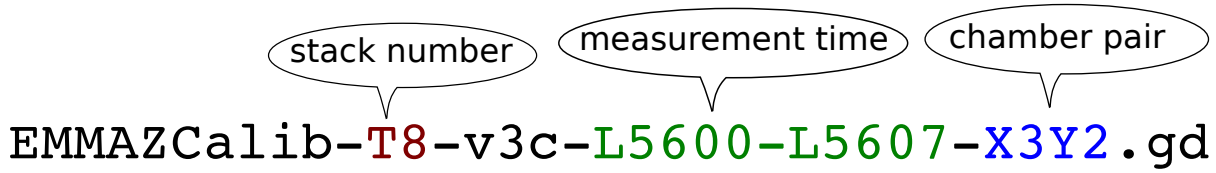


Figure 23: One measurement data file name of stack 8. It consists of the stack number, measurement time and chamber pair.

I obtained one data file per chamber pair and per a certain time period. For example Fig. 23 shows a single data file name, which expresses the stack number (T8) written in red, time period (L5600-L5607) written in green and chamber pair (X3Y2) written in blue. Actually the time period does not correspond to the exact time, but it rather expresses the chronological order of the file. There are several hundred files per stack. The main program can identify the stack number, the chronological order and the chamber pair by reading the file name.

A structure of the data file is shown in Fig. 24. In this figure the signal parameters are too small to be legible, but it rather shows the structure of the file content. In this case the file contains the signals from the chambers X3 and Y2 in each plank of the stack T8. One line consists the signals of one particle event. By using this information the particle may be tracked. The parameters marked with red text are the signals from the reference planks and the parameters marked with blue text are the signals from the researchable planks. The last parameter in a line is the particle event time in seconds (unix time) and all the other numbers

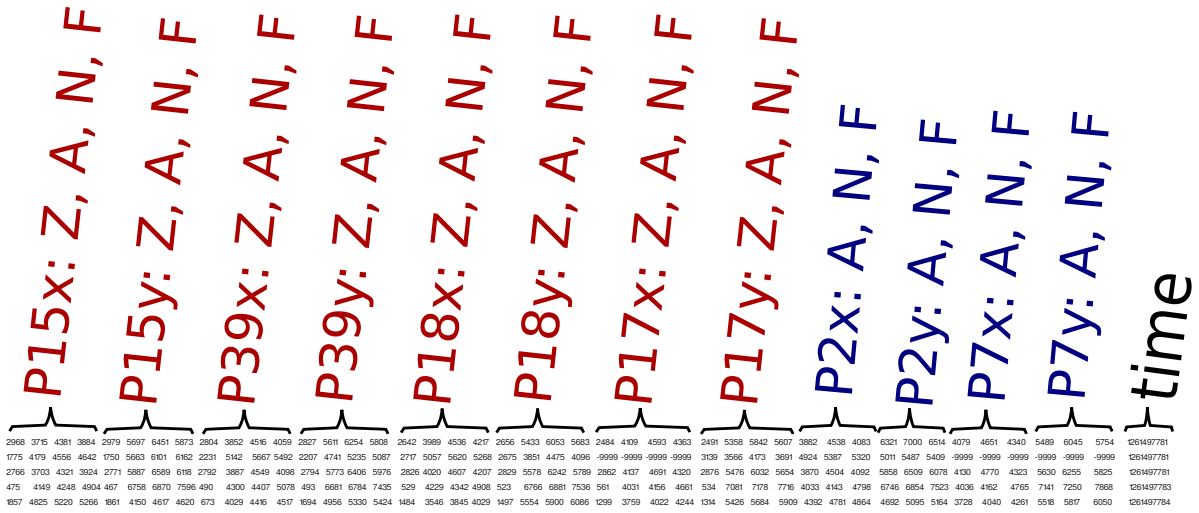


Figure 24: Four first lines (particle events) of the data file EMMAZCalib-T8-v3c-L5600-L5607-X3Y2.gd (stack 8, chamber pair X3Y2). The reference planks' signals (red) are saved as millimeters (Z) and as three time channels: anode (A), near (N) and far (F). The reseachable planks's signals (blue) are only time channels A, N and F.

are the measurement parameters of the planks. In Fig. 24 the first 32 parameters are the signals of the reference planks (P15, P39, P18 and P17). In every line each reference plank provides 8 parameters, both x and y chamber gives the hit position in millimeters (Z) and the time channels: anode (A), near (N) and far (F). Whereas each researchable plank has only 6 parameters: the time channels (A, N and F) from x and y chamber.

5.3 Calibration tables

As mentioned before the researchable planks' signals don't give the hit position but just the time channels. The intention of the calibration process is to produce the calibration tables. They determine how the time signals correspond to the hit position in each chamber. There is one calibration file for each plank and they are named after the plank name (P1-auto.cal, P37-auto.cal etc.). My program uses the calibration tables to determine the hit positions.

A calibration file consists of 15 columns and 1200 rows as mentioned in section 4.2.3. The first parameter in a row is a channel (time difference) and other 14 are the position parameters. The channel increases by one in each row, at the first row it is -100 and at the last row it is 1100. Each channel gives the time difference between near and anode signal (N-A) or between far and anode signal (F-A). The channels correspond to

the position values thus that each seven chamber are given two position parameters (N-A and F-A) which form the 14 parameters in a row of calibration table.

6 Efficiency programs

My task is to determine the efficiencies of the planks programmatically. This section describes the method of determining the efficiencies and the results are shown in section 7.

The efficiency of a drift chamber defines how many of the penetrated muons the chamber detects. It is a ratio between detected muons and all the muon hits, thus it can be expressed as percents. If the efficiency is poor, too many muons go through the detector without been detected. This makes the measurements unreliable if the efficiencies are not known.

The intention in studing the efficiencies is to find out the chambers, whose efficiency is good enough and does not depend on the environmental factors. I studied the efficiencies both as a function of time (Appendix B) and as a function of position (Appendix C) of the chamber. The efficiencies have been evaluated only for the first hit (A, N, F triplet) in a chamber. Actually in the surface laboratory have been measurements which aim to study efficiencies for multiple hits per chamber, but the analysis is still not completed.

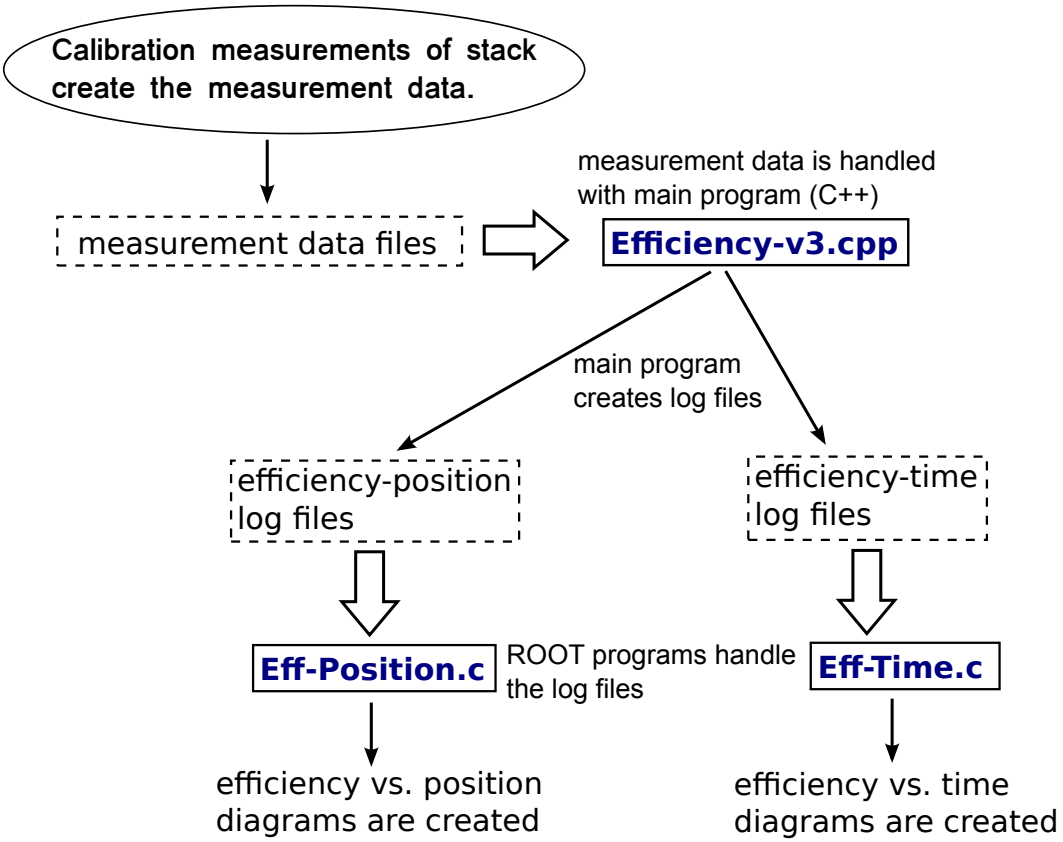


Figure 25: Steps of the efficiency calculating for one stack

In the present work I have studied the efficiencies of the drift chambers of six calibration stacks (T5, T6, T7, T8, T9 and T10). The diagram in Fig. 25 introduces the steps for evaluating the efficiencies for one calibration stack and thus these steps must have been done separately for all 6 stacks. The measurement data from the calibrations contain the signals registered by the drift chambers. I made C++ program Efficiency-v3.cpp (Appendix A.1), which analyzes the measurement data and creates the log files. Actually the main program was the most time consuming part in my work. Those log files include the information of the measurement data in different forms (more legible). Then I handle the log files with my ROOT programs Eff-Position.c (Appendix A.3) and Eff-Time.c (Appendix A.2), which create the efficiency plot diagrams presenting the efficiency as a function of time and function of position.

In my special project [44] I made the first version of the main program (Timo Enqvist took part in the contributing process) and I studied the efficiencies only for the stack T9. In this work I improved the main program (and the ROOT programs) to be able to handle measurement data also of other stacks. Actually the code of the main program shortened remarkably and its operation became more effective. Additionally, I found and repaired many mistakes of the code and I changed the poorly defined method of calculating the efficiency vs. position in a better way.

6.1 Description of the main program

In a nut shell my main program Efficiency-v3.cpp reads the measurement data, evaluates the tracks of the muons (see section 6.1.1) and studies the acceptability of the signals from the chambers. Finally it calculates the efficiencies of the chambers by evaluating how many of the penetrated muons the chambers have detected.

For each stack the main program needs the data files and a calibration table as mentioned in section 5.2. In addition to that the program needs also a file list and a plank list of a stack. A file list is a text file which includes the data file names of a stack in a choronoligal order and a plank list gives the planks' position in a stack.

The main program handles the calibration stacks separately so that the user gives the stack number (5, 6, 7, 8, 9 or 10), therefore the program has to run six times. The program processes one data file at a time and

the file lists enable the program to go through the data files in a chronological order. The plank list shows in which order the planks are in the data file. The data file is read line by line and each line consists of information about one particle event as mentioned in section 5.2. The program reads the parameters of a line and stores in memory the moment of time, the position parameters in millimeters of the reference planks and the time channels of the researchable planks. By using these measurement parameters the program is able to calculate the efficiencies.

6.1.1 Reconstructing the particle's path

This section discusses the reconstructing the track of the particle which penetrates the calibration stack. The reconstruction is performed with the help of the signals from the reference planks.

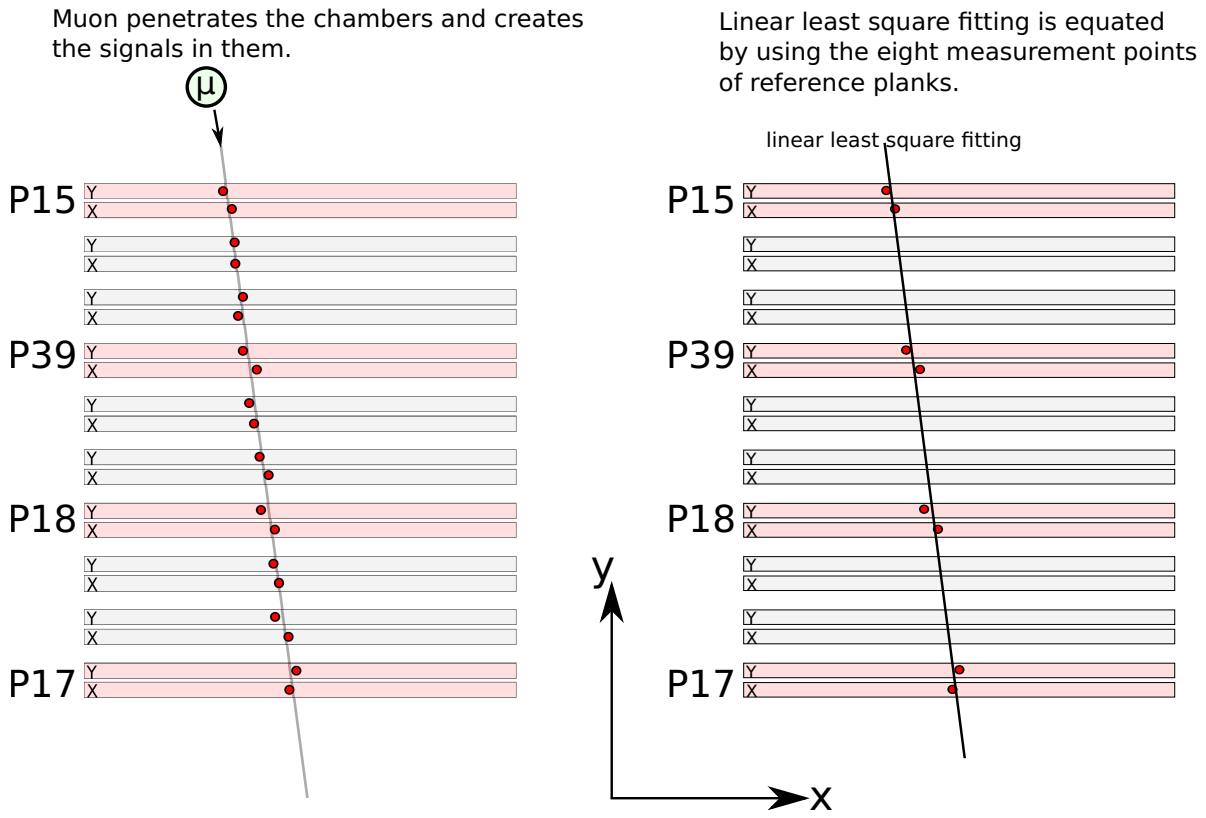


Figure 26: View from the side of the stack. On the left muon penetrates the stack and causes signals in chambers. On the right is the linear least square fitting for the eight data points of the reference planks. (by Tiia Monto, CC BY-NC)

Fig. 26 presents a schematic view of a muon penetrating a stack. The chambers of the reference planks are drawn as pink and researchable planks' chambers are drawn as grey. As seen in the figure, each plank has two overlapping chambers (Y and X) and in a successful case both

of them give a signal. The signals from the chambers are marked as red points and they may not be exactly at the same line. The position of those data points are determined with coordinates y and x whose directions are shown in Fig. 26. The variable y expresses an height and x expresses a position in the direction of chamber. Each individual measured hit point may be referred by the subindex i (1, 2, 3, 4, 5, 6, 7 or 8). For example the coordinates of the upmost hit point in a stack are expressed as (x_1, y_1).

When studying a particle event the program first tracks the particle's path by fitting a straight line for the hit points of the reference planks as presented on the right of Fig. 26, when the fitting equation is of the form

$$y = x \cdot A + B, \tag{3}$$

where the constant parameters A and B must be solved to evaluate the fitting equation. The fitting is made by using the linear least square fitting method. In this method the fit line is defined so that the sum of the squares of the differencies between the measured value y_i and the fit value y is minimum, in other words the equation $\sum_{i=1}^{N}(y_i - (x_i \cdot A + B))^2$ is as small as possible [5]. In the case of linear least square fitting the parameters A and B of equation 3 may be expressed as

$$A = \frac{N \cdot \sum_{i=1}^{n}(x_i y_i) - \sum_{i=1}^{n}(x_i) \cdot \sum_{i=1}^{n}(y_i)}{N \cdot \sum_{i=1}^{n}(x_i x_i) - \sum_{i=1}^{n}(x_i) \cdot \sum_{i=1}^{n}(x_i)} \tag{4}$$

and

$$B = \frac{\sum_{i=1}^{n}(x_i x_i) \cdot \sum_{i=1}^{n}(y_i) - \sum_{i=1}^{n}(x_i) \cdot \sum_{i=1}^{n}(x_i y_i)}{N \cdot \sum_{i=1}^{n}(x_i x_i) - \sum_{i=1}^{n}(x_i) \cdot \sum_{i=1}^{n}(x_i)}, \tag{5}$$

where $n = 8$ is the total number of the hit points of the reference chambers. The height of a chamber y_i is actually known and the value of hit position x_i is expressed in millimeters in the data file. Since the coordinates x_i and y_i of each reference hit point are known the parameters A and B can be solved by using the equations 4 and 5. Then the main program is able to solve the fitting equation 3, which represents the reconstructed track of the particle.

Since the fitting equation is determined, the program needs to know the total number of the accepted particle events (good fits), each of which is taken into account when calculating the efficiencies. An indivual fitting is accepted by the program only if the conditions below are fullfilled:

- each reference data point is between 0 mm and 3650 mm

- horizontal distance between the reference data points in X and Y chambers is at most 60 mm
- horizontal distance between each reference data point and fit line is at most 60 mm
- the fit line goes through all the chambers

6.1.2 Processing the measurement signals

The program uses the method described in section 6.1.1 to define if the particle event (fit line) is accepted or not. If the particle event is accepted, then the program studies also the signals from the researchable planks caused by the same muon. The intention is to find whether the particle hit has caused successful signals in researchable chambers or not. The more successful signals appeared in a chamber the better is the efficiency. The schematic view of the signals and linear least square fitting in a single chamber is presented in Fig. 27.

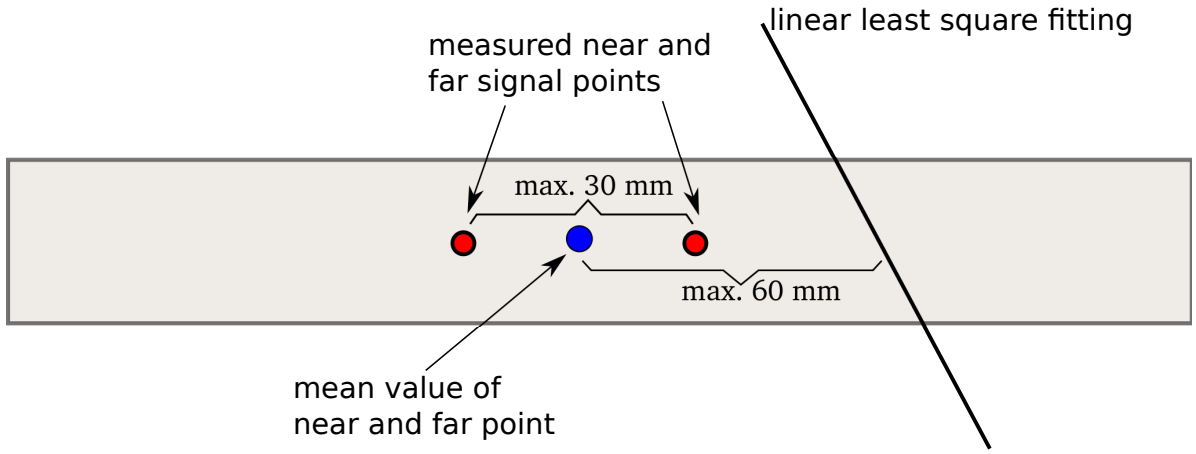


Figure 27: Closeup view of a chamber. The line is linear square fitting, red points are the measured position values of near and far signals and the blue point is the mean value of those. (by Tiia Monto, CC BY-NC)

The program reads the time channels from the measurement data and then it finds out which position value (in millimeters) the time channels correspond to. To do this the program uses the calibration tables and the position value is obtained both from near and far signals. In Fig. 27 the near and far signals are marked as red points and the mean value of those is marked as blue point. Since the program has checked the signal position (mean value) of each plank, it evaluates if these signals are acceptable or not. Fig. 27 shows the accepted maximum distance

between the near and far points, and the maximum distance between signal position and fitting line. A single signal position point of a chamber is accepted only if the conditions below are fullfilled:

- point is between 0 mm and 3650 mm
- in a chamber near and far signal positions are at most 30 mm from each other
- the horizontal distance between point and fit line is at most 60 mm

6.1.3 Method of calculating efficiency vs. time

The main program determines the efficiency as a function of time for each X and Y chamber of the planks. Each data file corresponds to a certain time period and certain X and Y chambers. The time period of data file is defined as the time interval between the first and the last line of the file.

Actually the program is able to define the efficiency separately for each left and righ half of a chamber. Fig. 28 shows the left and right halfs of each seven chamber. For example the halfs of the chamber X2 can be written as X2L (left) X2R (right). When sorting out which chamber pair the particle penetrated, the chamber halfs may be deduced. For example if a particle went through the chamber pair Y1 and X2, the signals must have been produced in overlapping halfs Y1R and X2L.

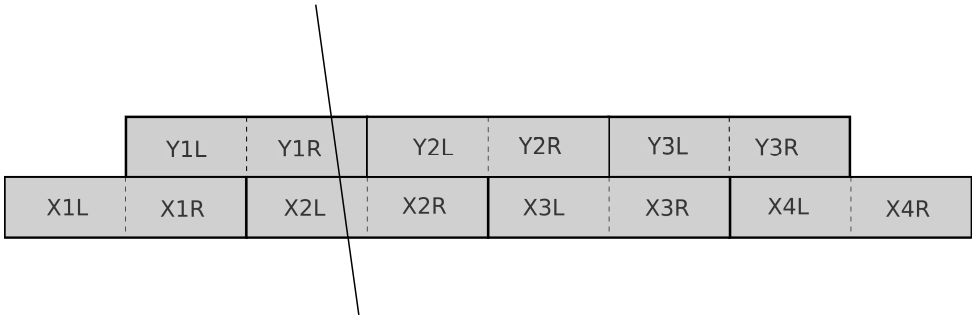


Figure 28: Cross section of a plank. Each chamber are being divided in to left and right half. Particle causes signals in the halfs and the program may track in which half they are produced. (by Tiia Monto, CC BY-NC)

When studying one data file each acceptable particle event described in section 6.1.1 rises the value of ”good fits”. Each acceptable signal described in section 6.1.2 from a chamber rises the value of ”good points”

for that chamber. The efficiency of a chamber (or chamber half) per a certain time period t is calculated by dividing the "good points" $gPoints[t]$ with the "good fits" $gFits[t]$ and it can be expressed as

$$eff[t] = \frac{gPoints[t]}{gFits[t]}. \quad (6)$$

Each data file provides one plot point $eff[t]$ in an efficiency vs. time diagram of certain X and Y chambers in each plank. The number of plot points in a diagram is the number of the data files (time periods). The main program saves the efficiencies in log table thus that each line contains chamber half, efficiency and time.

6.1.4 Method of calculating efficiency vs. position

The main program determines the efficiency vs. position of each seven chamber in each plank. Contrary to the case of efficiency vs. time the efficiency as a function of position is determined only for whole chambers, not for chamber halfs.

In this case the length of a drift chamber (3650 mm) is divided into parts of the size of 4 mm so that the first part corresponds to the region of 0–4 mm, next part 4–8 mm etc. My program calculates the efficiency of each 4 mm sized part and there is a total of 912 parts which results in 912 plot points in the efficiency vs. position diagram.

When an accepted particle (linear fitting) goes through a certain part (p) of a chamber, the number of good fits $gFits[p]$ is increased by one (see section 6.1.1). If the particle has caused an acceptable signal in the chamber, the program rises the number of good points $gPoints[p]$ (see section 6.1.2). The accepted maximum distance between linear fitting and signal is 60 mm as described in section 6.1.2. Even if the measured signal seems to appear in an other part than fitting line, the number of good points $gPoints[p]$ is increased just for the part which was penetrated by the fitting line. Anyway the efficiency of a part $eff[p]$ is expressed as relation between the accepted fitting lines and accepted measurement points with the equation

$$eff[p] = \frac{gPoints[p]}{gFits[p]}. \quad (7)$$

Each value of $eff[p]$ is one data point in the efficiency vs. position diagram. This value is determined for each 912 chamber parts in each

chamber of the researchable planks. The main program saves the efficiency vs. position in log table so that each line contains the chamber part (position) and the efficiencies of each seven chamber .

6.2 ROOT programs

The ROOT programs Eff-Time.c and Eff-Position.c are more simple and shorter than the main program. They go through the log files and draw the plot diagrams efficiency vs. time and efficiency vs. position. As the main program also these ROOT programs are able to handle one stack in a time and they have to run separately for each six stack.

There is two log files per each plank: efficiency vs. position and efficiency vs. time. The log files are recognised by their file name. The ROOT programs read the log file line by line and save the parameters.

Eff-Time.c (appendix A.2) reads the efficiency vs. time log files and saves the parameters of a chamber, an efficiency and time. The program handles one file at a time. Since a whole file is processed it is able to make the diagrams for one plank. The program creates the graphical output by using TCanvas method and produces the diagram plottings by using TMultiGraph and TGraph method.

Eff-Position.c (appendix A.3) saves the parameters of position and chamber by reading the efficiency vs. position log files. The diagrams are produced in the same way as in the program Eff-Time.c: by using the methods TCanvas, TMultiGraph and TGraph.

7 Results and notices

This section describes the efficiency results which are obtained by using the method discussed in section 6. The analysis of the results is shown in section 8.

The efficiencies have been determined as a function of time and as a function of position. These results are shown in diagrams which imply how well the chambers have been working. By examining the diagrams more closely one may deduce how the efficiency have been influenced by some external factors.

7.1 Efficiency vs. time

Examining the efficiency as a function of time makes it possible to evaluate how the external time dependent factors have affected the efficiency. The next subsections discuss especially about the efficiency decreases and simultaneously appeared pressure changes or modifications in electronics of experiment setup.

The efficiency vs. time diagrams are shown in appendix B. The diagrams are arranged such that one page contains the diagrams of one plank. The diagram pair on the top provides the efficiencies of the whole chambers and each chamber's mean efficiency of the plot points is represented in parenthesis. On the bottom are diagrams of the chamber halfs. Vertical axis is the efficiency as percents and the horizontal axis is the time in hours from the beginning of the measurements.

As can be seen in the diagrams in appendix B, the efficiencies are not stable in every chamber. The lines are rugged and in some cases one can clearly observe the peaks. Some of the features of the efficiencies may be caused by the external factors. EMMA group members have written manually in the logbooks (Logbook I and Logbook II) how the measurements have being progressed. The most important factor in the calibration measurements was to observe that the counting rate was high (the efficiency was not paid attention). Usually the anode high voltage (HV) have been tried to keep at 5950 V and grading HV at 3950 V but sometimes they may have been changed to improve the counting rate. The changes of HV have been usually written down in a logbook. Also the air pressure and temperature measured outside of the cham-

bers have been written down in a logbook almost daily, the temperature and pressure plotting are shown in appendix D. By reading the logbooks some of the efficiency decreases in efficiency diagrams may be explained.

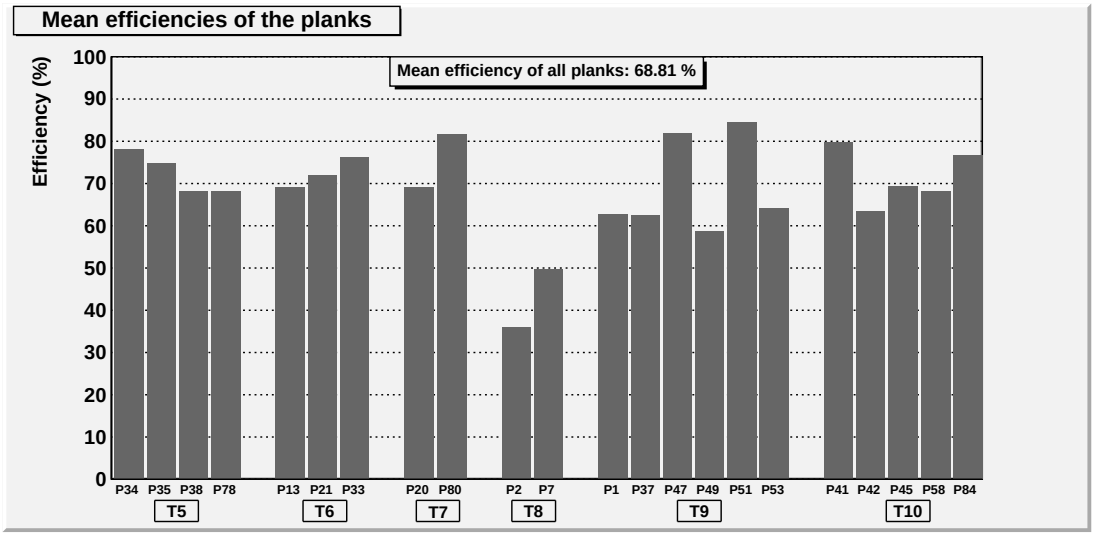


Figure 29: Mean efficiencies of the planks (average value of all plotting points in efficiency vs. time diagrams per plank)

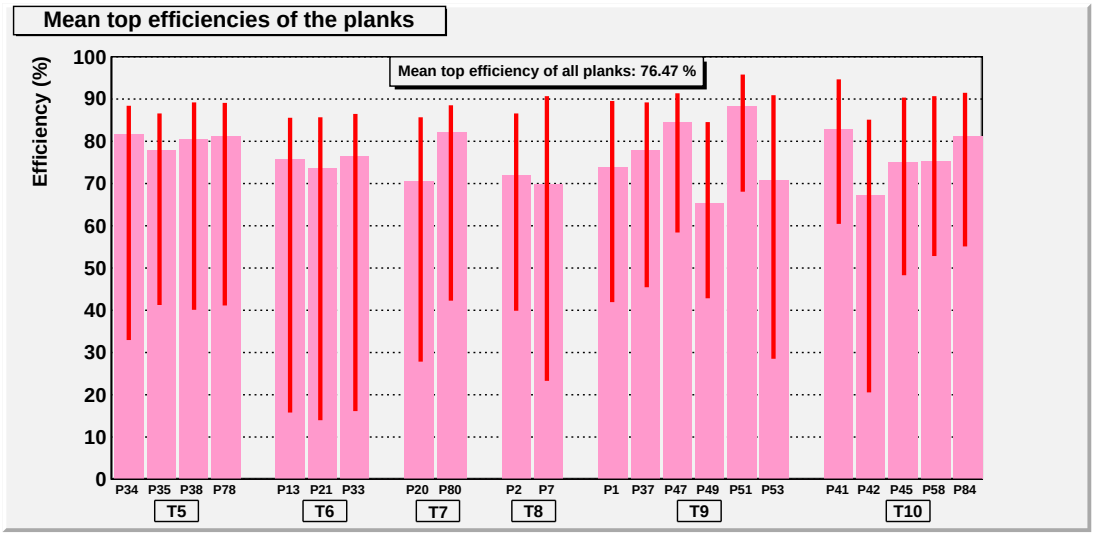


Figure 30: Mean top efficiencies of the planks (average value of plotting points in efficiency vs. time diagrams excluding the larger peaks). The red vertical lines show the minimum and maximum efficiency of the accepted plotting points.

By using the efficiency vs. time diagrams I have determined the efficiency of a plank in two different way. The more simple method is just to calculate the mean efficiency of all the plotting points of each seven chamber (see grey diagram in Fig. 29). The more involute method is to calculate the mean top efficiency of the plotting points thus that the larger downwards efficiency peaks are excluded (see pink diagram in Fig. 30). This was not simple, because in some of the chambers the efficiency

was not stable at all and the peaks appeared all the time. By determining the mean efficiency of all the planks by the first method the mean efficiency is 68.8 % and by using the second method the mean efficiency is 76.5 %. The last one (76.5 %) can be interpreted to be the planks' efficiency in the case of external factors don't disturb the chambers remarkably. And maybe the pink diagrams give a clue of the efficiencies of the planks working underground.

7.1.1 Efficiency vs. time of the stack T5

The stack T5 consists of four successfully calibrated planks (P34, P35, P38 and P78) whose efficiencies are shown in appendix B.1. The efficiency of the majority of the chambers seem to be about 80–90 % most of the time, but one can see small unstability appearing in those efficiencies. For example the efficiency diagrams of the whole chambers of the plank P34 are presented in Fig. 31 and the chamber halfs are shown in Fig. 32.

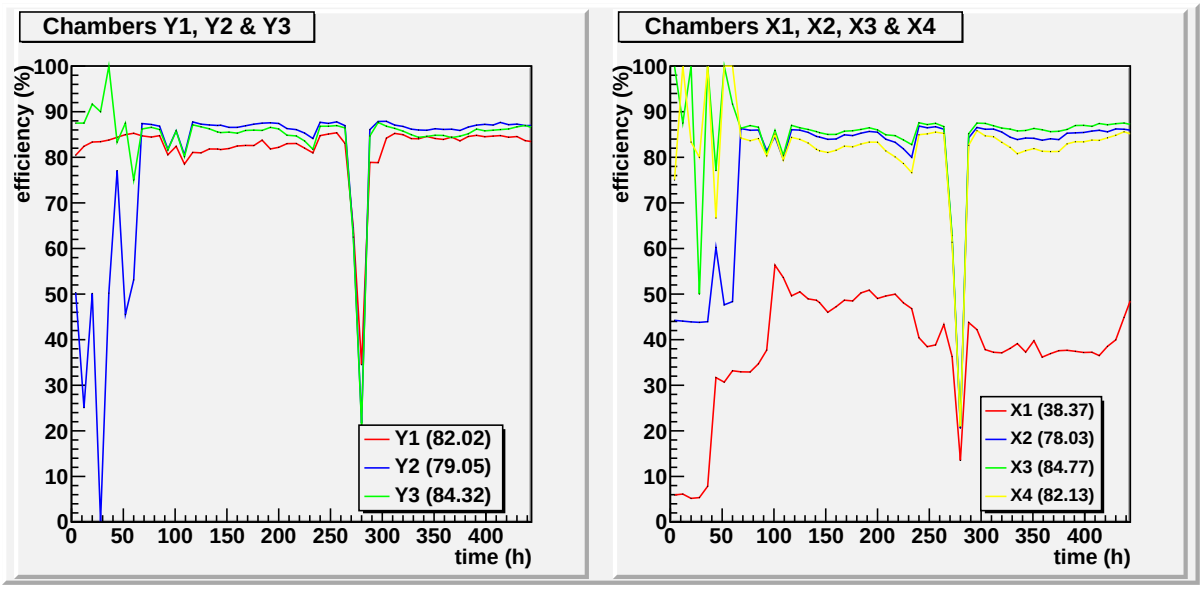


Figure 31: Efficiencies of the chambers in the plank P34 in the stack T5

Generally speaking the diagrams in Fig. 32 show that the efficiencies in both halfs of each chamber in the plank P34 seem to be similar (the plotting lines are superimposing). Thus it seems that the left half of chamber works as well as the right one. Actually most of the chambers in the other planks have the right and left part been working similary as seen in appendix B.1.

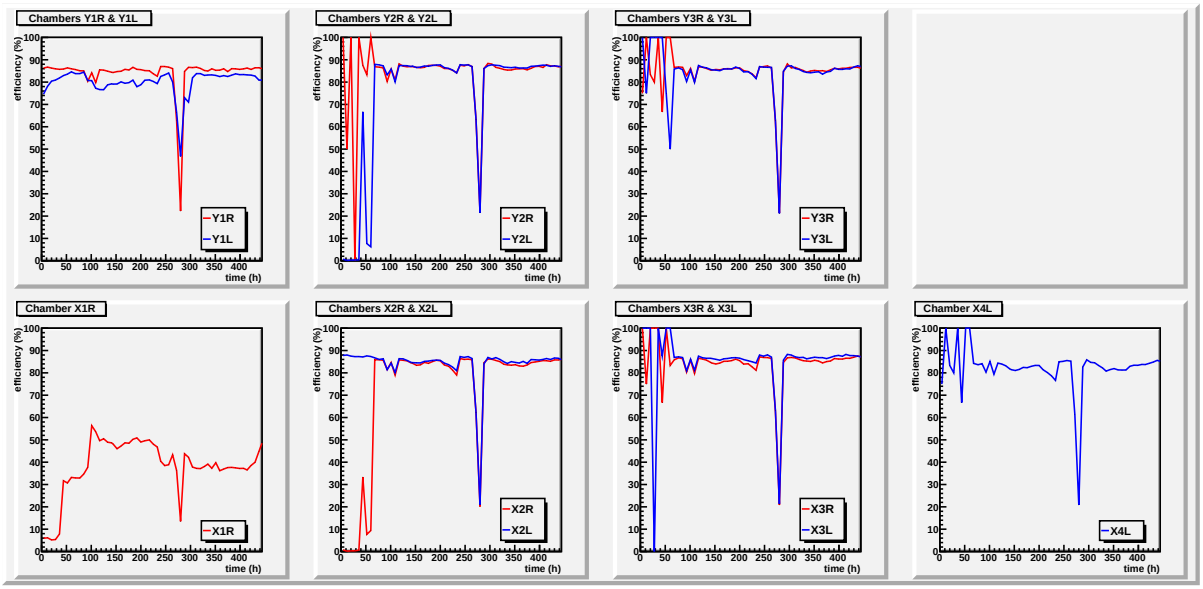


Figure 32: Efficiencies of the chamber halfs in the plank P34 in the stack T5

For some reason during a few first days of the T5 measurement the efficiency varied drasticly in each chamber. Most clearly this appeared in chamber Y2. Actually the chamber X1 of P78 has an efficiency of zero during many days at the beginging of the measurements. Maybe there have been some problems, because the data wires of X1 in P78 have been changed according to the logbook LI [16]. Actually many times there were problems in the chambers in the beginning of the measurements. Those problems were tried to be fixed.

The diagrams reveal also that the chamber X1R of each plank has an efficiency much lower than in the other chambers during the whole calibration measurement. The same problem of X1R is appearing in each plank of the stacks T5, T6 and T7.

Each of the planks in the stack T5 have common that the efficiency is reduced to between 10 % and 40 %. Therefore there is a peak at the time 280 h (15 – 16 Septemper 2009). According to the logbook, on 16 Septemper the anode HV has been changed from 5975 V to 5950 V and the counting rate has recorded to be dropped [16]. Actually the pressure vs. time diagram of T5 in appendix D reveals that pressure has decreased tens of millibars during the 16 and 17 Septemper while the air temperature is increased a couple of degrees.

Additional to the peak at 280 h there is an even deeper peak in the planks P38 and P78 appeared at 100 h which corresponds to the date 8 Septemper 2009. According to the logbook the anode HV of all planks was then 5975 V and the air pressure 991 mbar on. Maybe P38 and

P78 are more sensitive for external factors for some reason, because at 280 h their efficiency peak is deeper than in the case of the planks P34 and P35. In addition, during the last 200 hours of measurements the efficiency of P38 and P78 varies more suddenly than the efficiencies of P34 and P35.

7.1.2 Efficiency vs. time of the stack T6

The appendix B.2 shows the efficiency vs. time of the planks in the stack T6 which contains three successfully calibrated planks (P13, P21 and P33). Fig. 34 shows the pressure in a function of time during T6 measurements. As can be seen, the plank P33 has been working the best, because its efficiency was the most stable. But, in P13 and P21 the efficiency has been varied clearly most of the measuring time. In P13 and P33 the efficiency seems to be quite stable during the first 200 hours of the measurements.

As in the case of the stack T5, also in T6 the X1R chambers seem to have systematically lower efficiency than in other chambers. This has been appearing in each plank.

In the efficiency curves of the planks P13, P21 and P33 some parallelism may be detected. Around 220 h (4 October 2009) the chambers of each plank seem to have a different sized and shaped downwards peak on efficiency curve. The deepest and widenest peak appeared in P21 shown in Fig 33. The peak of P21 seem to be a double peak and the efficiency drops down to 30 %. According to the logbook counting of P21 was poor on 5 October 2009 and the low pressure (958 mbar) has been doubted to be cause of it. As can be seen in the pressure plot in Fig. 34, the pressure becomes very low around 5 October. Because there are the peaks also in the other planks at 220 h, the low pressure may have affected to all of the planks but in P21 the influence has been the strongest. There is also other remarkable peak in P21 at 80 h which may also be explained with the lowering pressure as seen in the left of Fig. 34.

Each chamber has a non-sharp and relatively wide peak on efficiency curve at 380 h (10–11 October 2009). In P13 this peak is the most remarkable going down to about 50 %. The pressure plotting does not reveal radical changes in pressure during the time period of Fig. 34.

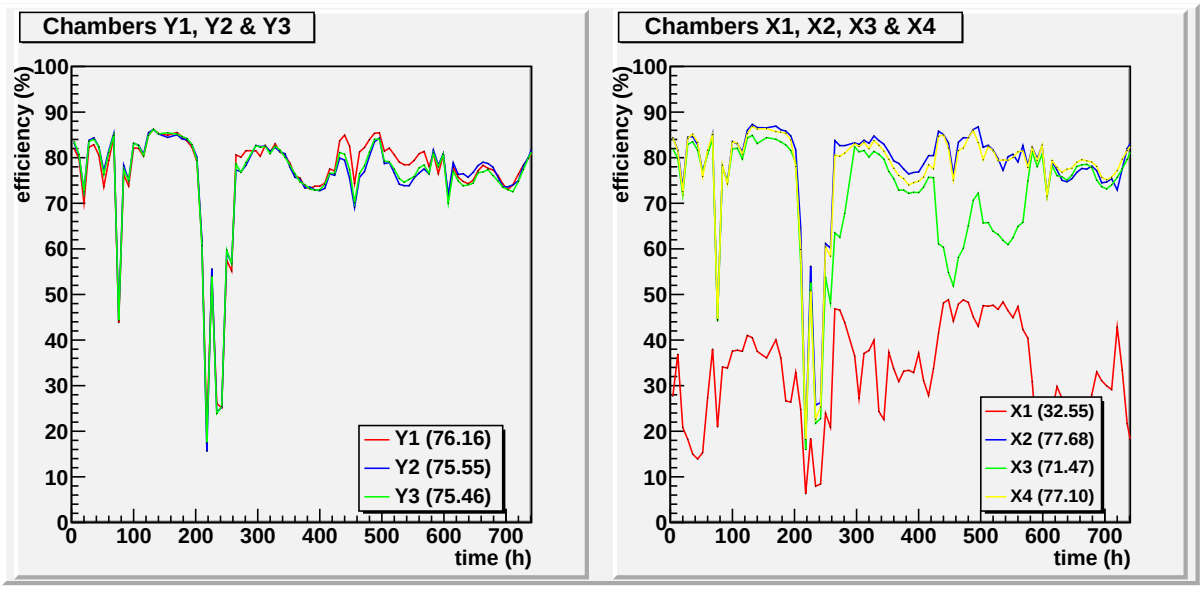


Figure 33: Efficiencis of the whole chambers of the plank P21 in the stack T6

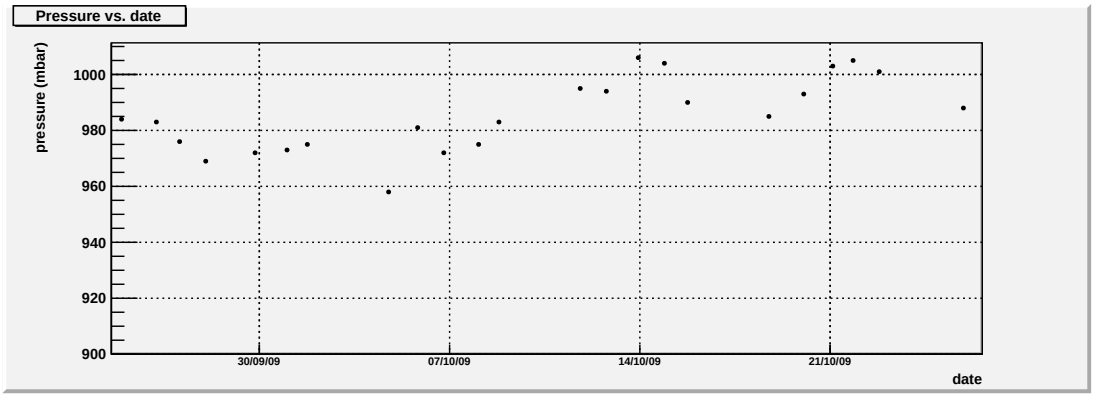


Figure 34: Pressure during the calibration measurements of the the stack T6

When considering the last 300 hours of the measurements, one can see four upwards peaks at the 440 h, 500 h, 580 h and 660 h in efficiency diagrams of each plank. This phenomenom appears most clearly in P13 and a quite clearly in P33, but in P21 there are several peaks which makes it harder to distinguish these four peaks. The temperature is just increasing composedly and the pressure has only two upward peaks during these days.

7.1.3 Efficiency vs. time of the stack T7

The measurements with the stack T7 started already on 30 October 2009, but due to the problems during the measurement my data starts on 16 November 2009.

In the stack T7 there is only two successfully calibrated planks P20 and

P80 whose efficiencies are shown in appendix B.3. P80 was better with the mean efficiency of more than 81 % (third best of this study) whereas the efficiency of P20 was about 69 %. These planks worked quite well, because there were not massive peaks in efficiency diagrams.

Some coincidental variation and a few short peaks occurred in the efficiency curve of each chamber. During the first 50 hours the efficiency was increasing smoothly in both planks. There are small downwards peaks at 80 h (19 November 2009), 140 h (21.-22. November), 330 h (30 November), 400 h (2.-3.December) and 530 h (8. December). In P20 the peaks in the efficiency curves are clearer than in the case of P80. The most clear peaks appeared at 330 h and 400 h which may be due to the low pressure. Actually in the pressure curve there is a wide downwards peak between about 17. November and 3. December. During the last clear efficiency peak at 530 h there were also a downwards peak in the temperature vs. time curve.

7.1.4 Efficiency vs. time of the stack T8

Like T7 also in the stack T8 there were only two successfully calibrated planks P2 and P7. The mean efficiency of each chamber in these stacks is low due to the efficiency dropping at the middle of the measurements. During these measurements the temperature changed a lot, between 13 °C and 22 ° while the pressure changed between 970 mbar and 1025 mbar as seen in the section ”Stack 8” in appendix D.

Maybe the efficiency variance of each chamber appearing during all the measurements is due to the sustainable change in the pressure and temperature. As seen in appendix B.4 the most remarkable change in efficiency is the sudden decrease at 700 h (around 20 January 2010) without rising up again. The pressure was high (1017 mbar). In the plank P2 this occurs in each chamber half and in P7 (see Fig. 35) this occurs in all chamber halfs except X3R, Y3R, Y3L, X4R and X4L. After 700 h the efficiency of the worse chambers had reduced radically, in some chambers the efficiency was even less than 1 %. On 18 January 2010 (about two days before the sudden efficiency reduction), some changes were performed for the power supplies so that HV will be shut down if the leakage current increases too much [17]. On 22 January the pressure were 1025 mbar (high) and the anode HV was increased to 5975 V due to the low counting rate. On 25 December HV has recorded to be

run down. Maybe the leakage current had increased. Additionally the measurements were stopped and started many times for testing. One of these activities may have caused the efficiency drop at 700 h.

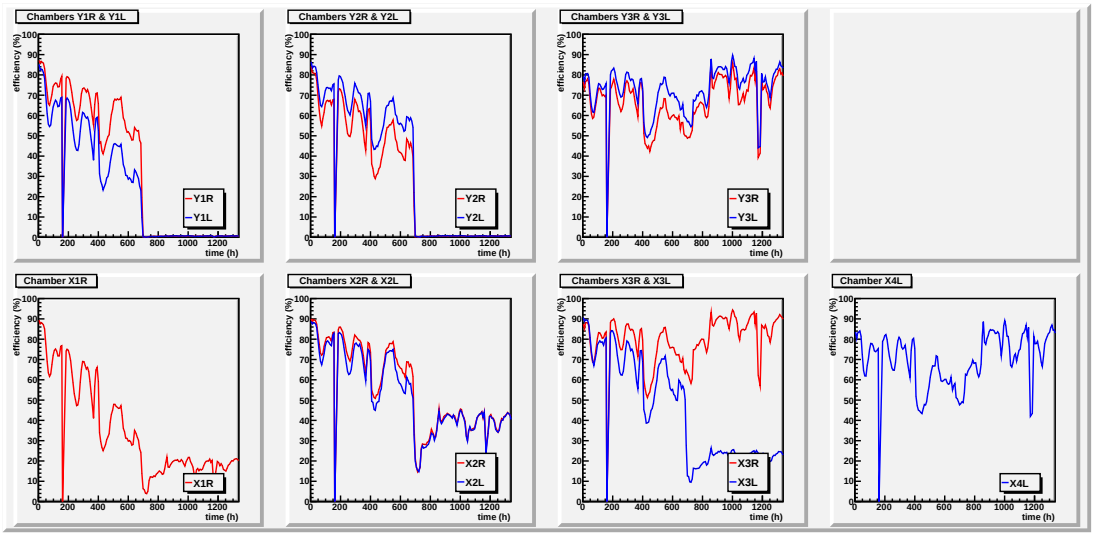


Figure 35: Efficiency in the chamber halfs in the plank P7 of the stack T8. The efficiencies of X3R and X3L are not similar after 700 h.

Second diagram from the left on the bottom in Fig. 35 shows one interesting feature of the chamber X3 of the plank P7. The efficiency of the chamber halfs X3R and X3L are not similar after 700 h. The efficiency of X3L is radically decreased although the efficiency of X3R is as high as before 700 h. There may have been some problems with the grading lines on the left part of the chamber. For example, they may have lost the connection with the HV supply. This kind of problem could make the electric field inside the chamber non-symmetric, which may impede the forming of the signal.

After 700 h part of the chambers of P7 have an measurable efficiency. The instantaneous efficiency drop at 850 h (26-27 January) may be due to the rapid decrease of the pressure (from 1020 mbar to 980 mbar) during those days. Actually more remarkable efficiency reduction occurred at 1170 h, but not as dramatic changes in pressure or temperature happened then.

7.1.5 Efficiency vs. time of the stack T9

The stack T9 contained six successfully calibrated planks: P1, P37, P47, P49, P51 and P53 whose efficiencies are shown in appendix B.5. The planks P51 and P47 were the best ones of all when comparing all the

planks of the stacks T5 – T10. P51 has a mean efficiency of more than 84 % and P47 a mean efficiency of almost 82 %.

The efficiency curves of the planks P1 and P37 are similar with several deep and wide downwards peaks occurring at the same time. They have common that the planks are clearly unstabile. Their efficiency changed many dozens percentage units several times during these measurements. The planks P51 and P47 are the most stable ones in this stack (in addition to the highest efficiency). The planks P49 and P53 are more stable than P1 and P37 but theirs efficiency curves are not as uniform as the curves of the planks P51 and P47.

In most of the planks the efficiency has reduced at around 110 h (2 March, 2010). Only in the plank P51 and P47 this has not occurred such clearly than in other planks, because in these planks the efficiency dropped only in the chambers Y3 and X3. During that time the pressure has dropped to less than 970 mbar [17].

Almost all the chambers in each plank have an efficiency reduction at 230 h (6–7 March 2010), but there is no markings concerning this finding in Logbook II, because it was a weekend.

For some reason there is two deep and wide spikes in the efficiency curves of the planks P1 and P37 between 300 h and 400 h. The efficiency of those planks is reduced to even less than 10 % at 380 h (12–13 March). Also in this case there is no markings on Logbook II.

In the case of the planks P1, P37, P49 and P53 small simultaneous downwards peak at 550 h (around 20 March) in the efficiency curves is detectable. On 19 March the measurement was stopped due to test runs and the calibration measurements was started again during the same day.

7.1.6 Efficiency vs. time of the stack T10

The five successfully calibrated planks of the stack are P41, P42, P45, P58 and P84. P41 was the best of these planks with a mean efficiency of almost 80 %.

Efficiency of each plank in the stack T10 has decreased sharply to zero simultaneously three times: at 310 h (13 April 2010), 640 h (26–27 April) and 840 h (5 May). Considering the efficiency at 310 h one can see that

actually there is no efficiency plots during several days (13–19 April) after that. Maybe that's due to some events happened during these days. On 12 April a new code was installed, on 15 April the HV was marked to be off and on 20 April the measurement was started again [17]. After the two other points of efficiency decreasing at 640 h and 840 h there is no efficiency points during a few dozen hours. So these three points don't seem to be due to weather but maybe rather due to the electronics.

In addition to those radical peaks one can see a drop of the efficiency curves around 740 h (30 April – 1 May) which is the most clearly visible in the case of P41 and P58. In other planks this spike is smaller or not clearly notable. No markings in Logbook II has been written during that time, because it was May Day.

7.2 Efficiency vs. position

The intention to evaluate the efficiency as a function of position is to clarify whether the efficiency is placid along the whole chamber length or not. The downwards spikes in an efficiency vs. position curve may be due to a problem which depends on position. If the efficiency vs. position curve is systematically low without spikes, the chamber may be poor, but then the problem does not seem to depent on the position.

Generally speaking the efficiency curves of the chambers are more flat in the stacks T5, T6, T7 and T8. In these stacks the unstability was usually appeared clearly as local spikes. On the other hand the efficiency curve of some planks in the stacks T9 and T10 are less smooth and it vary more strongly.

7.2.1 General features of efficiency vs. position diagrams

Appendix C represents the efficiency vs. position diagrams of each plank. Each figure includes a pair of diagrams which show the efficiency of each chamber in one plank. For example, the efficiency vs. position diagram of the plank P21 is represented in Fig. 36 (this was one of the most placid planks). The left diagram contains the chambers Y1, Y2 and Y3 and and the right contains X1, X2, X3 and X4. The vertical axis is the efficiency in percents and the horizontal axis represents the position in millimeters. The bin-size of the horizontal axis is 4 mm.

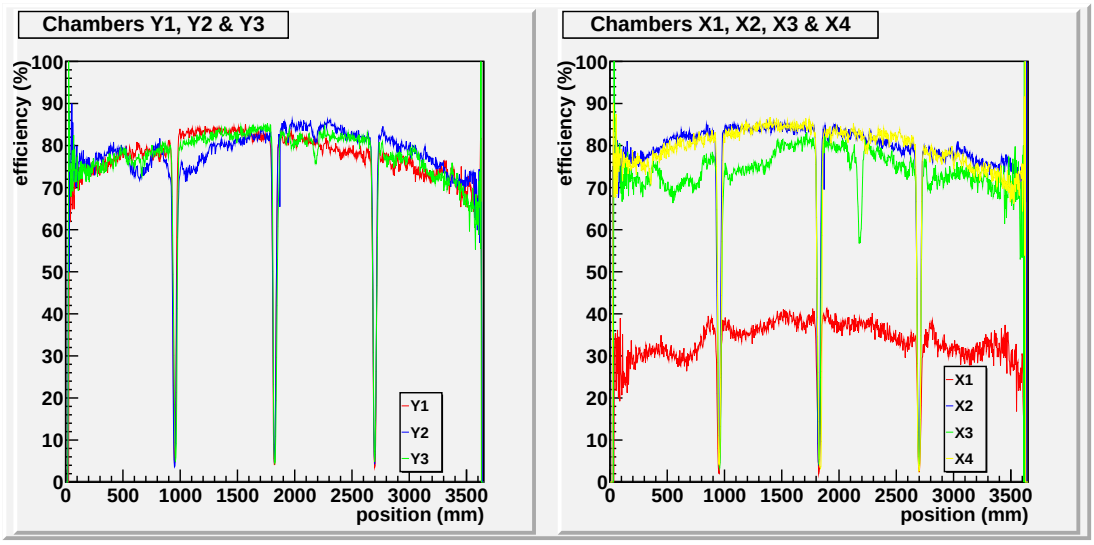


Figure 36: Efficiency vs. position diagram of the plank P21 in T6. It was a quite placid.

In each diagram the plotted efficiency curve oscillates strongly which makes the curve to look like bold and rugged, but not sleek like the curves in the efficiency vs. time diagrams. The amplitude of this noise effect seems to be a few percentage units.

Inside each drift chamber there are three plastic ”spiders” which hold the anode wire. The efficiency is decreased notably at those three points. In diagrams this is seen as three long downwards spikes at 950 mm, 1820 mm and 2700 mm for P21 as seen in Fig. 36.

7.2.2 Variation of efficiency

In some efficiency vs. position diagrams there are local downwards spikes, which don’t seem to be an effect of the spiders. For example in the plank P35 the efficiency of the chamber Y2 (blue plotting in left diagram) was decreased between 100 mm and 1200 mm as seen in Fig. 37. The wide spike is almost one meter wide and about 20 percentage units deep. Maybe the most remarkable local big spike has appeared in the plank P20 of the stack T7 (see Fig. 38). The efficiency of the chambers Y1, X1 and X2 has lowered at the same position between 0 mm and 1000 mm and in the case of Y1 the efficiency is dropped even 40 percentage units.

There are no clear explanations for the spikes in the efficiency vs. position curves. Maybe there is some uncleaniless inside the chambers (ions) or perhaps there is momentarily and locally occurring problems

with the gas mixture. Of course the time dependent factors are hard to be associated with the features of the efficiency vs. position diagrams, but in a couple of cases there is a mention in a logbook related to a position dependent efficiency spike.

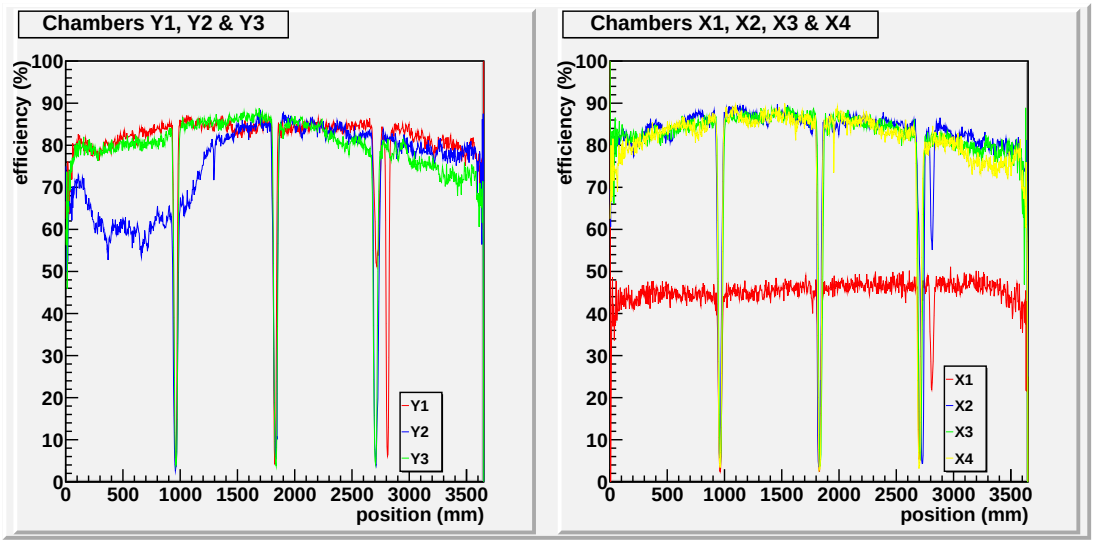


Figure 37: Efficiency vs. position diagram of the plank P35 in T5. The efficiency of Y2 (blue plotting in the left diagram) was decreased between 100 mm and 1300 mm which is seen as a spike on left.

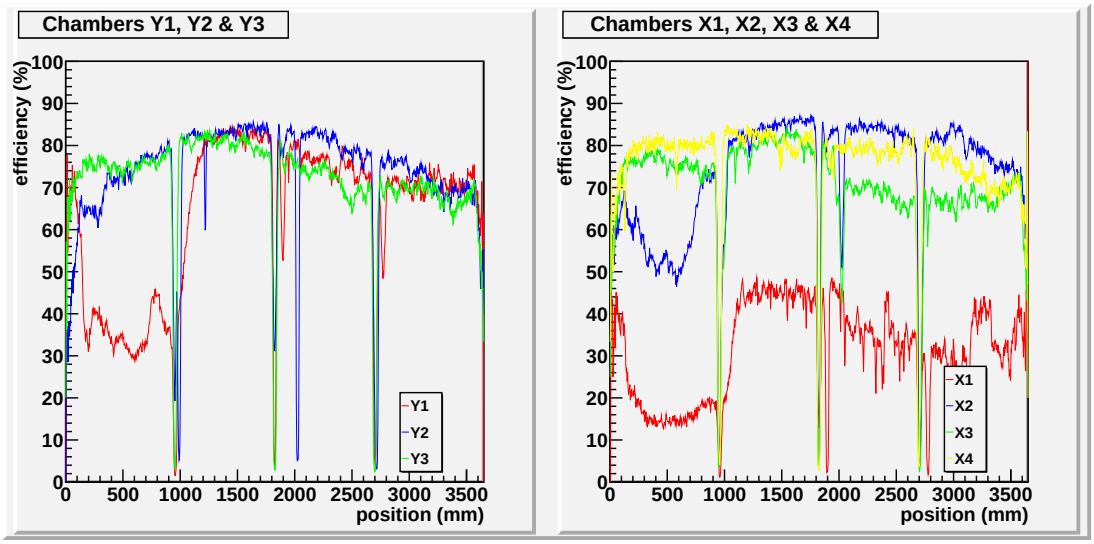


Figure 38: Efficiency vs. position diagram of the plank P20 in T7. The efficiency of Y1 (red plotting in left diagram) is locally lowered at the near end. This same chamber has been reported to have possible problems with near end.

Actually during the measurements of the stack T7 the chamber Y1 of the plank P20 has been reported to have a ”possible” problem with near end on 17 November 2009 [16], but no stricter comments are given. In the efficiency vs. position curve of chamber Y1 there is a clear pike at the near end as shown in Fig. 38 which would confirm the existence of that problem. According to the efficiency vs. time diagram of Y1

the efficiency has been systematically lower than that of the other Y chambers during the whole measurements. Thus it seems that there may have been a lack of signals at the near end of Y1 all the time during the calibration run.

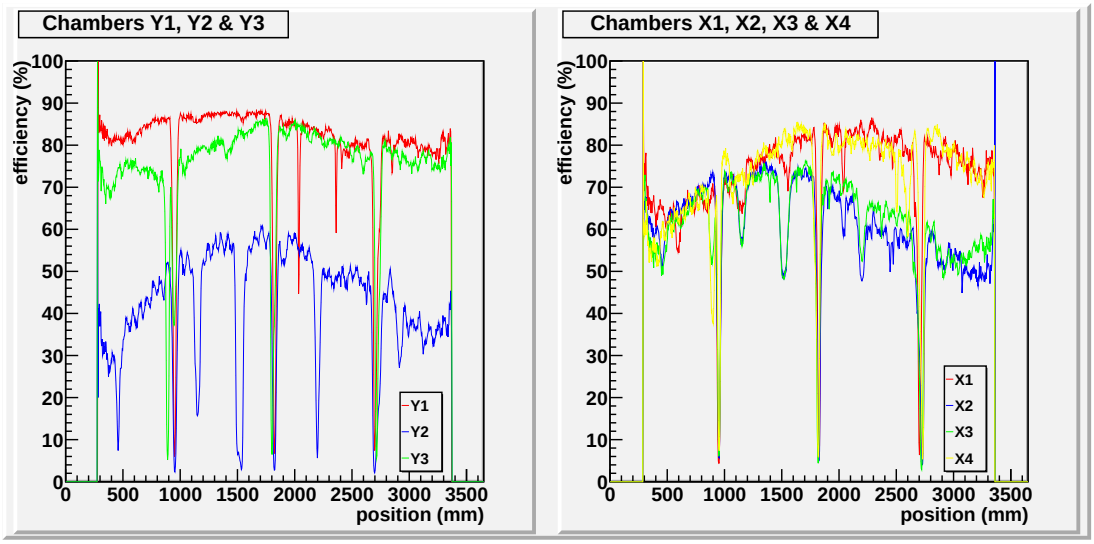


Figure 39: Efficiency vs. position diagram of the plank P42 in T10. The efficiency of X4 (yellow plotting on right) is lower at near end than at far end and according to a logbook X4 has been problematic.

On 27 April 2010 during the calibration of T10 the near end of the chamber X4 in P42 has been reported to be ”problematic” [17]. The efficiency vs. position curve of X4 (yellow plotting on the right diagram) in Fig. 39 is lower around the near end than around the far end. I don’t know how long time this ”problem” has been occurring, but perhaps it has caused the decrease of efficiency around near end.

By the way, in the case of P42 the efficiency between 0 mm and 280 mm from both the near and the far end is recorded to be zero, although the real efficiency may be higher. This is due to a shorter plank (P84) which is set just below the plank P42. The reconstructed muon tracks are not taken into account if they didn’t penetrate also the shorter plank. If a muon went through the stack very close to the chambers’ end, the muon signals were excluded from this study, because it penetrated only the ”normal” planks and not the short one.

7.2.3 The worst planks

As mentioned in the beginning of the section 7.2 the worst cases of this study are the stacks T9 and T10, because they include a few planks whose efficiency variation has been remarkably radical.

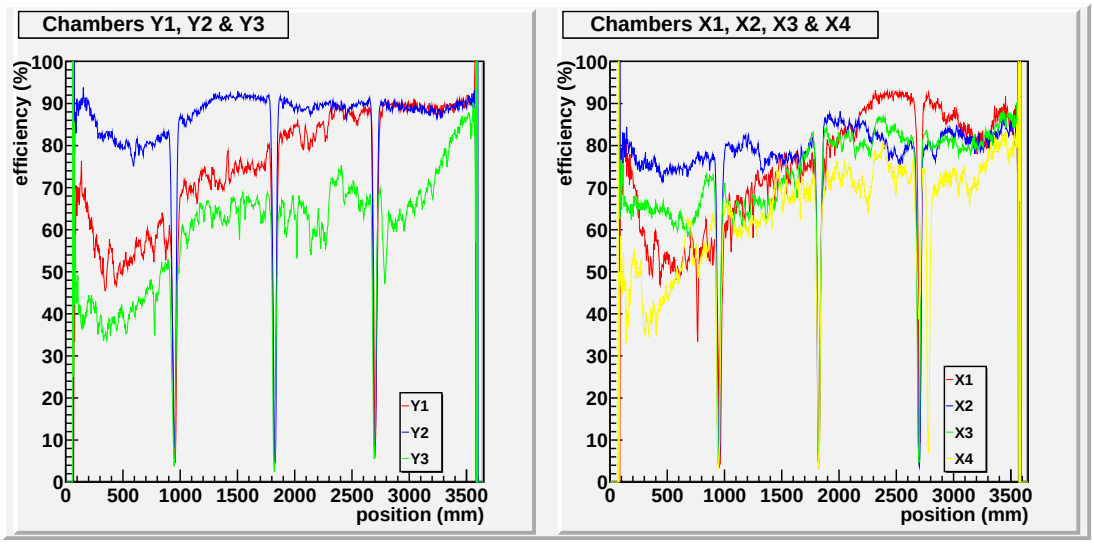


Figure 40: Efficiency vs. position diagram of the plank P45 in T10. In most of the chambers the efficiency is lowering towards the left end.

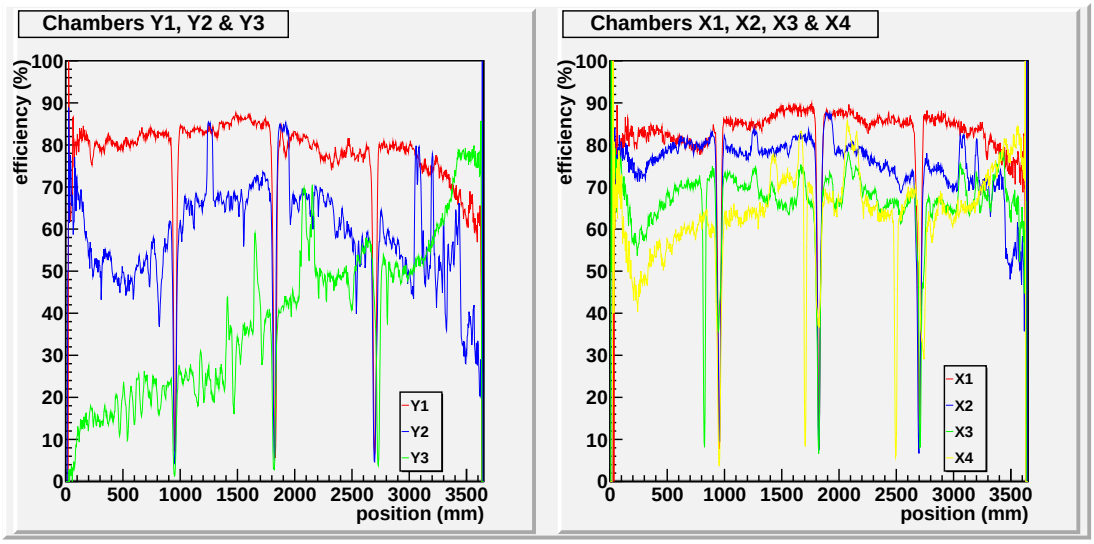


Figure 41: Efficiency vs. position diagram of the plank P53 in T9. The efficiency of Y3 is much lower at the near end than at the far end.

In a few planks of the stacks T9 and T10 the efficiency seems to vary dramatically along the whole chamber. For example, the efficiency vs. position diagrams of P45 in T10 are shown in Fig. 40 and diagrams of P53 in T9 are shown in Fig.41. In the case of P45 the efficiency is clearly lowering towards the left end in the chambers Y1, Y3, X1 and X4. On the left the efficiency curves have a spike upwards at the position near 100 mm. These changes have appeared at about the same positions at several chamber. Maybe the most reamarkable lowering has occurred in the plank P53 of the stack T9 shown in Fig. 41. In the chamber Y3 of P53 the efficiency rises from about 10 % at 100 mm to about 80 % at 3500 mm.

It is interesting to notice that in some planks the unstable efficiency curves of some chambers have almost similar shape. For example, in the stack T9 some chambers in the planks P1, P47, P49 and P51 have an efficiency which descends from the mid point to 3550 mm. It looks like there is the same problem at the same longitudinal position in several chambers. I don't know if this is a feature of the chambers or does it originate from some external factor.

8 Analysis of the efficiency results

This section analyses the efficiency results by considering possible sources of uncertainties and comparing the results with other results.

8.1 Mean efficiency and mean top efficiency

The quality of a drift chamber may be determined as the efficiency which defines how many of the penetrated charged particles (muons) the chamber detected. Since the chambers are not used independently but they are physically attached together to form planks, it is useful to define the efficiency of each plank.

The planks are listed in the order of the mean efficiency in Tab. 3. The mean values are simply calculated as an average of all the plotting points of the efficiency vs. time diagram. Thus these values are affected by all the disturbances and they may be smaller than the ”real” efficiencies.

Table 3: The planks listed in the order of mean efficiency (including all the plotting points in efficiency vs. time diagram of a plank)

more than 80 %		**70 % – 80 %**		**60 % – 70 %**		**less than 60 %**	
P51	84.51	P41	79.86	P45	69.42	P49	58.61
P47	81.93	P34	78.07	P13	69.06	P7	49.80
P80	81.61	P84	76.54	P20	69.05	P2	35.86
		P33	76.11	P58	68.09		
		P35	74.69	P38	68.03		
		P21	71.86	P78	68.01		
				P53	64.21		
				P42	63.49		
				P1	62.55		
				P37	62.36		

In Tab. 4 the planks are shown in the order of mean top efficiency. The mean top efficiency of a plank is an average of the chamber halfs’ plotting points so that the larger peaks of the efficiency vs. time diagram are excluded. The mean top efficiency is higher than the mean efficiency, because the worst drops of the efficiency are excluded. The subscript expresses the difference between the mean top efficiency and the smallest efficiency value (minimum). Corresponding the superscript expresses the difference between the mean top efficiency and the largest efficiency value (maximum). The minimum and the maximum values are

determined from the non-excluded parts of the chambers' efficiency diagrams. According to Tab. 4 the best planks with a mean top efficiency more than 80 % are P51, P47, P41, P80, P24, P78, P84 and P38.

Table 4: The planks listed in the order of mean top efficiency (the larger peaks in efficiency vs. time diagram are excluded). Subscript and superscript do not express an actual accuracy. The subscript is the difference between the mean top efficiency and the efficiency minimum value. The superscript is the difference between the mean top efficiency and the efficiency maximum value. The minimum and maximum values are determined from non-excluded part of the efficiency diagram.

more than 80 %		**70 % – 80 %**		**60 % – 70 %**	
P51	$88.10^{+7.60}_{-20.02}$	P37	$77.81^{+11.37}_{-32.38}$	P7	$69.72^{+20.95}_{-46.46}$
P47	$84.44^{+6.91}_{-26.05}$	P35	$77.76^{+8.74}_{-36.57}$	P42	$67.28^{+17.75}_{-46.76}$
P41	$82.85^{+11.72}_{-22.48}$	P33	$76.37^{+10.08}_{-60.19}$	P49	$65.38^{+19.14}_{-22.60}$
P80	$81.97^{+6.54}_{-39.72}$	P13	$75.75^{+9.79}_{-60.01}$		
P34	$81.49^{+6.87}_{-48.59}$	P58	$75.25^{+15.42}_{-22.48}$		
P78	$81.24^{+7.83}_{-40.14}$	P45	$74.94^{+15.34}_{-26.72}$		
P84	$81.10^{+10.34}_{-26.02}$	P1	$73.88^{+15.06}_{-31.96}$		
P38	$80.40^{+8.80}_{-40.32}$	P21	$73.57^{+12.05}_{-59.65}$		
		P2	$71.94^{+14.67}_{-32.06}$		
		P53	$70.64^{+20.19}_{-42.10}$		
		P20	$70.37^{+15.30}_{-42.59}$		

Although I calculated the mean top efficiencies with ROOT macro, I selected the excluded parts (time frames) by checking the diagrams by an eye. In some cases the plotting is not placid at all and it was hard to separate the "worst" parts from the "accepted" parts of the plottings.

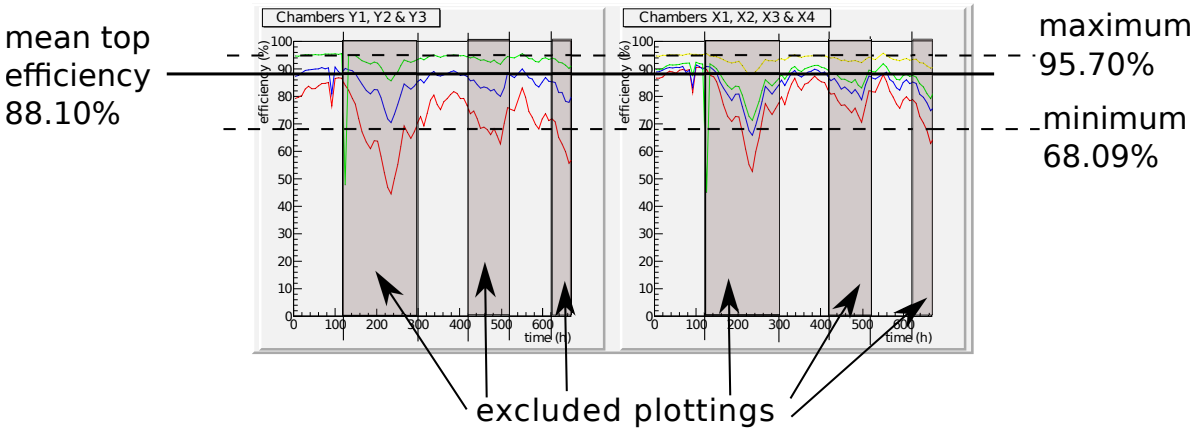


Figure 42: Mean top efficiency defined for the plank P51. The larger peaks (marked with grey background) are excluded.

For example, the efficiency vs. time diagram of P51 in Fig. 42 shows the

excluded parts of the plots with grey background. As seen in the figure the efficiency varies continuously and it is not obvious which parts of the plottings should be excluded.

8.2 Possible sources of uncertainties

My efficiency results in Tab. 3 and Tab. 4 don't necessarily show the absolute efficiency or the absolute quality order of the planks. The efficiency results may be inaccurate due to the factors listed below:

- my method of determining the efficiency
- problems in the measurement system
- method of defining the mean top efficiency
- possible fault in data

The instantaneous problems of the measurement system are the source of random inaccuracy. For example they may be due to broken electronics or weather changes. At least some of the measurement system problems may be partly explained by using the logbooks. These factors didn't affect similarly in different stacks, because the weather (pressure, temperature) changed and problems with electronics occurred differently during the separate calibration runs. By defining the mean top efficiency the planks may be more comparable because the most remarkable efficiency decreases (and random inaccuracy) are excluded.

Another possible source of random inaccuracy is the method of defining the mean top efficiency (selecting the excluded parts of the diagrams) as described in section 8.1.

The systematical uncertainty may originate from my method of determining the efficiency (see section 8.2.1). The possible fault in data may be an origin of the doubtfully (and systematically) low efficiency of X1 chambers in the stacks T5, T6 and T7.

8.2.1 Inaccuracy of my method

The systematical uncertainty is due to the method of calculating the efficiency. My method of calculating the efficiency is decribed in section

6.1. The efficiency of a chamber is calculated by dividing the good signals with all the acceptable fitting lines. The problem was to define rules which determine if a position signal of a researchable plank is good or not. By adjusting these rules one may affect the magnitude of the efficiencies obtained by the calculations.

The chamber efficiency is proportional to the number of the accepted signals. The signal position of a researchable plank is defined as a center point of the near and far signals. The maximum distance between near and far signal is 30 mm and the maximum distance between signal position and fit line is 60 mm. If these conditions are fulfilled, the signal is accepted.

I can't say which is the optimal maximum distance between near and far signals caused by one muon. If the acceptable distance was too long, it would be more likely that the signal is from a previous muon event and the calculated efficiency value becomes larger than it should be. On the other hand if the acceptable maximum distance is too short, a part of the real signals may be excluded. These same problems appear in the case of defining the acceptable maximum distance between a signal position and a fit line.

8.2.2 Pressure

According to the results shown in section 7 the clearest possible correlations between pressure and efficiency was revealed when the downwards peaks of efficiency curves appeared when the pressure was low. I detected in each stack at least one downwards peak during low pressure except in T10. In these cases the peaks were different sized and they appeared usually in several chambers of several planks at the same time.

The only case in which I detected the correlation between high pressure and efficiency decreasing was T8 in which the efficiency of most of the chambers decreased remarkably during exceptionally high pressure. Maybe the high pressure caused some problems in electronics as described in section 7.1.4. On the other hand, during most of the time of the calibration runs the measured air pressure was less than 1000 mbar which may be the reason why I couldn't find other features of efficiency related to high air pressure.

The pressure of the gas inside the chambers is actually slightly higher than the air pressure. The high air pressure increases the gas pressure and low air pressure decreases the gas pressure. Increasing of the pressure is believed to reduce the afterpulses and the number of the signals (trigger rate) observed by the drift chambers. This is shown in Fig. 43 which represents the trigger rate (coloured plottings) and the air pressure (black plotting) as a function of time measured underground. According to this figure the number of the detected muons is decreased if the air pressure is increased but this effect should not affect remarkably on the chamber efficiency [50] as the pressure variation is expected to change the efficiency a few percents at most [19].

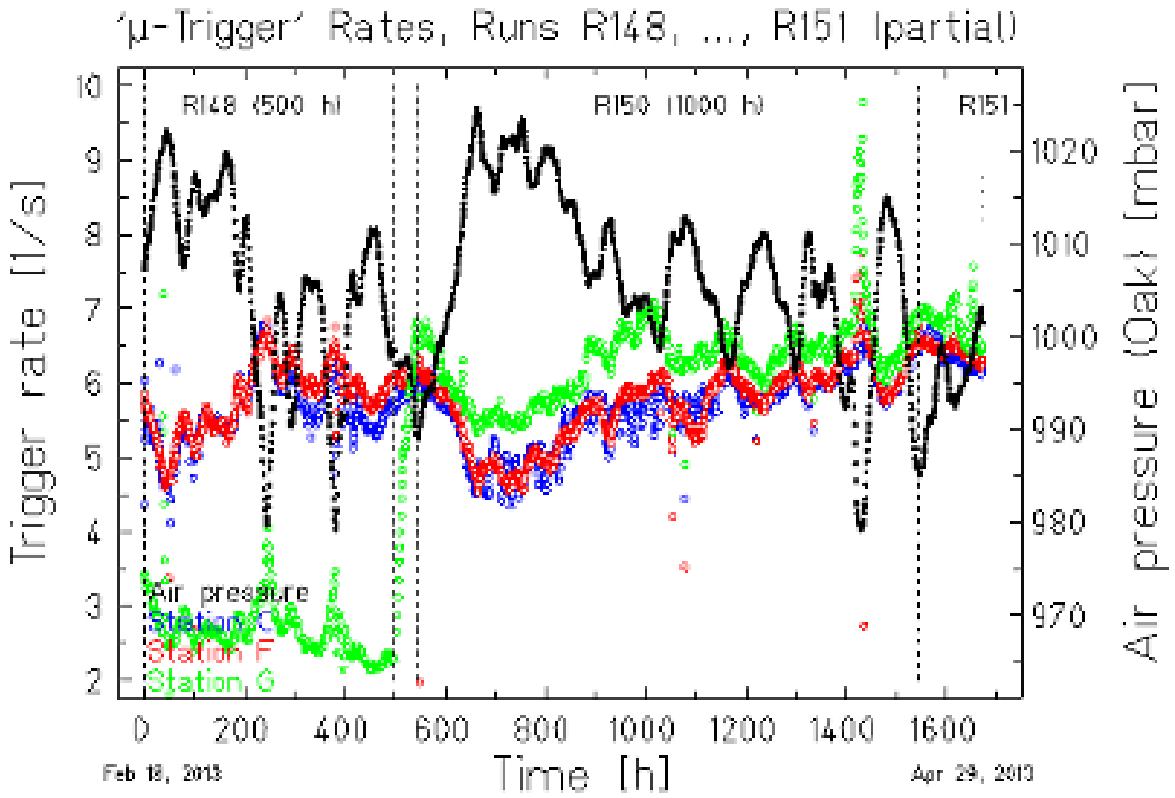


Figure 43: Air pressure (black plotting) and the muon rate (colourful plottings) measured underground as a function of time (horizontal axis). The blue curve is the muon rate observed by the station C, the red curve is the muon rate observed by the station F and the green curve the muon rate observed by the station G.

When the air pressure is increased, the pressure difference between the chamber and the air is smaller. The gas is coming out of the chambers through the device ”pulputin” sketced in Fig. 44 which indicates the flowing gas as bubbles. It is operating so that the gas pipe goes to a small transparent jar filled with oil in which the gas is formed into bubbles. High air pressure makes the pressure difference smaller and decreases the bubble rate [31]. It is not known how the air pressure affects the

oil viscosity or operation of the measurement system in detail. If the high pressure changes the features of the oil so that it forms bubbles at a lower rate, the gas may flow inside the chambers in pulses which probably weakens the chamber's ability to detect the muons.

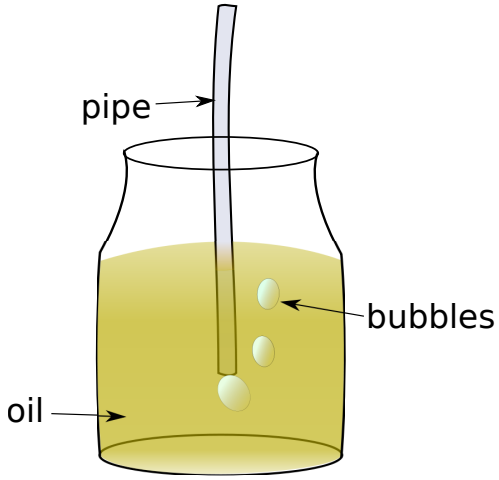


Figure 44: Gas arrives via the pipe to the glas jar filled with oil (by Tiia Monto, CC BY-NC)

Thus a high pressure possibly reduces a bit the efficiency by making the gas to flow less uniformly due to the change of the oil's features inside "pulputin". On the other hand the high pressure does not seem to affect the efficiency remarkably as mentioned earlier. According to my results, in some cases a low pressure may cause decreasing of efficiency. However, not all the peaks appear during low pressure. Since the low pressure increases the afterpulse rate, the chamber may have registered a signal from an afterpulse of a previous event instead of the signal of a muon which was just penetrating through the chamber. Maybe low pressure have sometimes caused a failure in electronics leading to efficiency drop.

8.2.3 Temperature

An increase of temperature may increase the number of afterpulses. An afterpulse is a signal which is created when an initial ionization in a chamber produces a photon. The photon may produce more electron avalanches, which are detected by the drift chamber. According to my results in a couple of cases the temperature changes may be related to the decreasing efficiency. If the temperature is higher than $27\ ^{\circ}$C, the afterpulse rate becomes remarkably high. On the other hand during the calibration measurements the temperature was not such high.

8.2.4 Electronics and HV supply

Behaviour of the electronics and the HV supply may affect the chambers' efficiency. Some failures of the electronics can even reduce the trigger rate and efficiency to zero.

The sketch of the connection between HV supply and the signal wires of one chamber is shown in Fig. 45. All the chambers were connected to a single HV supply. The signals are transferred via coaxial cables to be further handled. In the figure the "leakage current" (measured with HV supply) may appear due to a poor contact of the anode wire or maybe the impurity in the gas (there is not an ideal electric insulator). The restarts of the measurements have caused efficiency decreasing as seemed to happen in the stack T9 at 550 h, indicating that this system needs time to perform uniformly after restarting.

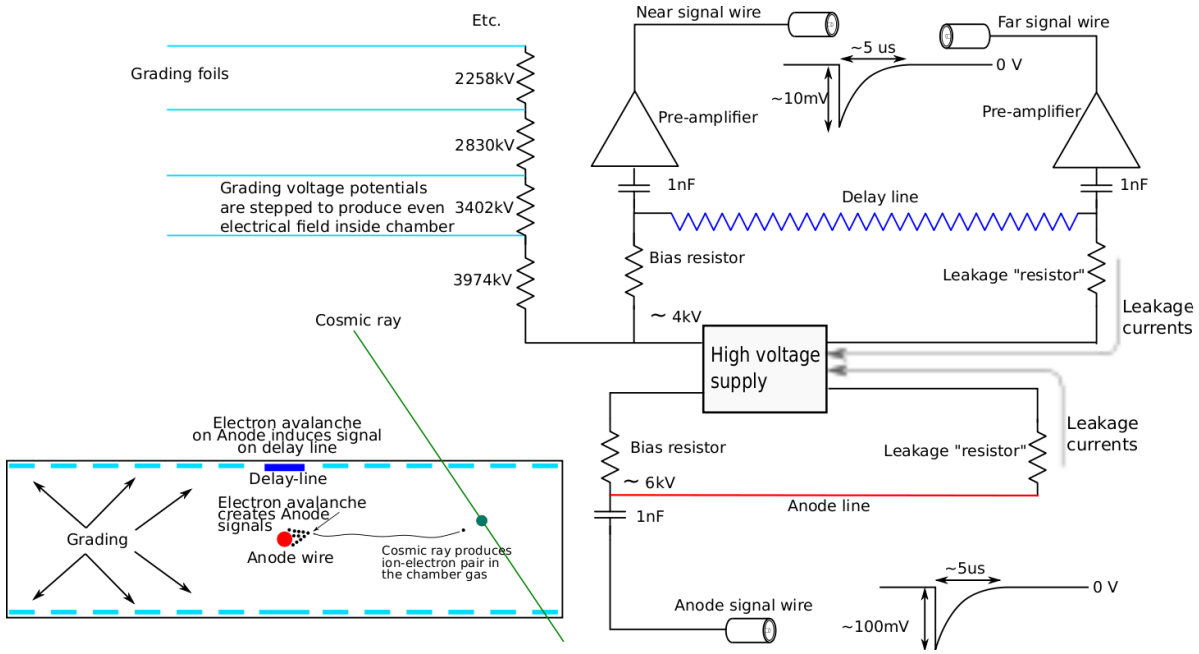


Figure 45: A HV supply connected to the anode line (red), delay line (blue) and grading lines (light blue) of one chamber (modified version of the original figure by Antto Virkajärvi)

Because the chambers are connected to the same HV supply, the broken ones may disturb other chambers. According to the logbooks, problems have been raported in the near end of the chamber Y1 in P20 of stack the T7 and X4 in P42 of the stack T10. In both cases there is clear lowering at the near end of the efficiency vs. position curve and also other chambers' efficiency were decreased around the near end. On the other hand sometimes in the beginning of the measurements the broken

chambers may have reduced other chambers' efficiency.

In the case of low counting rate due to low pressure, the HV may have been changed intentionally aiming to improve the counting rate. However sometimes it have worsen the counting rate (and the efficiency) against the intention. Thus some momentary changes of the efficiency curve may be due to adjusting of HV. According to my results there are few cases in which the HV was adjusted near the peak of the efficiency curve.

In general the efficiencies of the left and the right part of a chamber were similar. The problems in the grading lines (light blue lines in Fig. 45) may cause the different efficiency diagrams of the left and right part. Unfortunately, it is not possible to measure the electric field inside the chambers to prove this. Some individual grading lines may have lost the connection to the HV supply so that the "broken" electric field transfers only a part of the electron avalanches to the anode. In the stack T5 the chambers Y2 and X2 in each plank have efficiencies of the left and righ part varying strongly and not following each other during the first 70 hours. The worst case was the chamber X3 in plank P7 of T8. During the last half of the measurements the efficiency difference between X3R and X3L is about 70 percentage units. There are several chambers in which the left and right part have similarly shaped but systematically different efficiency. In those cases the difference is usually less remarkable. This is shown most clearly maybe in the chamber Y1 of P20 in T7, for which the efficiency of the right part is approximately 10–15 percentage units more than that of the left part.

8.2.5 Other factors

One can't measure the electric field, pressure, temperature or actually see what happens in molecular level inside the chambers.

Especially it is hard to deduce why some of the efficiency vs. position curves are non-symmetric. Maybe the gas is unclean thus that it contains ions. Then the electrons of the gas molecules detouched by muon may interact with the ions and not an electron avalanche nor a signal is created. If this was the reason for the peaks in the efficiency vs. position curve, it would not be clear why the impurities appear just in certain parts of the chamber's gas volume.

Some observations on comparing the humidity with both the high voltage and leakage current of the anode and grading lines are represented in an appendix of my special project [44]. According to that the increased humidity clearly reduces the voltage and increases the leakage current and vice verca. The dense peaks in humidity curve appeared at the same time than the peaks of voltage curve, thus I think that the humidity may affect the system (and efficiency). Maybe it increases the leakage current by affecting the electronics or the gas mixture inside the chambers.

Overall all the factors affecting the measurement system or efficiency are hard to find out. Additionally many factors may affect each other in unkown different ways.

8.3 Comparing the efficiencies with others' results

In these sections my efficiency results are compared with the results of Timo Enqvist and Tomi Räihä. Timo Enqvist made his own program which calculates the efficiencies of the chambers in the calibration stacks T5 and T6. Tomi Räihä has calculated the efficiencies from the underground data. I don't know their efficiency determination method, but I just compare the results with my results.

8.3.1 Efficiency results by Timo Enqvist

Timo Enqvist made a program which uses the data of the calibration stacks T5 and T6. Those stacks are also part of my study, thus I can compare them directly with my results. Enqvist gave me his results as text files which I handled with my ROOT program to draw the efficiency vs. time diagrams of his results.

When comparing my efficiency vs. time diagrams with Enqvist's diagrams, it is easy to notice that they have similar shape and the most remarkable peaks appear at the same time. However, the efficiencies calculated by Enqvist are systematically a few percents higher than mine. Additionally, some of the peaks of the efficiency curves in my results are deeper than in Enqvist's results.

For example, the efficiencies of X chambers in P34 by me and by Timo

Enqvist are shown in Fig. 46. These diagrams have the similarities mentioned above (same shaped and peaks at the same time). When conserning the figures closely one may see that Enqvist's efficiencies are a bit higher than mine. During the first 70 hours both my and Enqvist's curves oscillate strongly and they are partly different shaped.

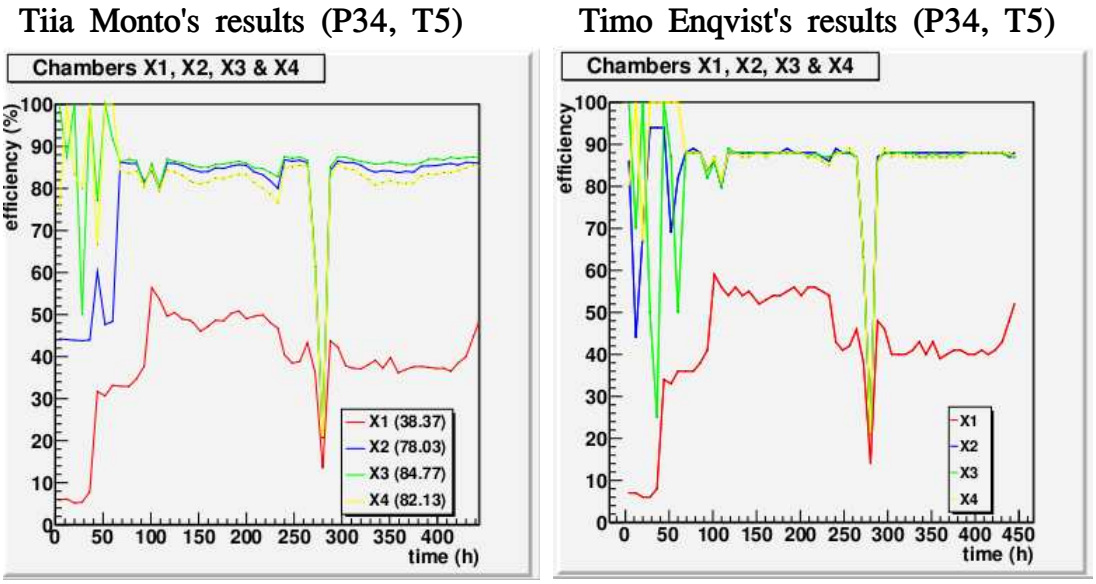


Figure 46: Efficiency vs. time diagram of X chambers in P34 of T5. On left is results by me and on the right is results by Timo Enqvist

Briefly, my efficiencies are a bit lower than the efficiencies calculated by Enqvist. Probably the conditions of acceptable signals defined by me are stricter than the determination method of Enqvist.

According to both mine and Enqvist's results the X1 chambers of each plank in the stacks T5 and T6 have an efficiency which is systematically less than the efficiency of the other chambers. Thus probably the problem is related the data, not my programs or my method of determining the efficiency.

8.3.2 Efficiency results by Tomi Räihä

Tomi Räihä has calculated the chambers' efficiencies by using a few data sets gathered in summer 2012 from underground measurements in the station C [49]. Räihä's study included only two same planks as my study (P13 and P33). According to his results in paper [49] the average efficiency of the chambers in the top layer is about 93.1 %, middle layer 92.2 % and bottom layer 88.7 %. (Räihä excluded the chambers with an efficiency less than 70 %.)

The average mean top efficiency of the planks determined by me is 76.5 %. Thus Räihä's results are remarkable higher than mine. The difference between my and Räihä's chamber efficiency may be due to different methods in determining the efficiency and different external factors of the measurements.

I don't know how Räihä determined the efficiency in his study. He might have set less strict conditions to accept signals. At least he excluded the worst chambers with an efficiency less than 70 % from his study, which set the minimum value (70 %) for the average efficiency.

Räihä calculated the efficiencies from the underground measurements at the EMMA level, whereas my results are based on the calibration measurements in the surface laboratory. The external conditions between the EMMA level in underground and the surface laboratory are different. The pressure on underground is about 10 mbar higher than in the surface laboratory [50], but the pressure changes are pretty similar. Maybe the higher pressure has decreased the afterpulses reducing the wrong signals whereas on surface the afterpulses disturb the results more. Inside the measurement stations at the EMMA level the air is probably more dry than in the surface laboratory which also may affect the results. The air temperature inside the measurement stations at EMMA level is about 20 °C and it is pretty steady. Maybe the steady temperature keeps the measurement devices more stable which perhaps enables better chamber efficiency. At least the particle and afterpulse rates are lower on underground.

Additionally Räihä didn't note strong correlation between the air pressure and chamber efficiency. According to his study one rapid pressure increase perhaps reduced the efficiency of the bottom plank layer. In my study I noticed several situations in which the pressure decrease seemed to reduce the efficiency and only one case in which the efficiency has decreased clearly during high pressure.

9 Conclusions

In this work I studied the efficiency of the drift chambers in a total of 22 planks of the EMMA experiment by using the data from the calibration runs T5, T6, T7, T8, T9 and T10, which were performed on the surface laboratory in years 2009 and 2010.

I created C++ program which calculates the efficiencies for the drift chambers as a function of time and as a function of position (chamber length) and creates log files for each drift chamber. The log files were handled with my ROOT programs which created the efficiency vs. time and efficiency vs. position diagrams.

When analysing the results I calculated the mean top efficiency of each drift chamber by excluding the most remarkable downwards peaks of the efficiency vs. time diagrams. By doing this I aimed to remove the most prominent external factors from the results. Because seven drift chambers are attached together to form a plank it was convenient to calculate the efficiency separately for each plank. According to my results the mean top efficiency of the worst plank (P49) is 65.4 % and the best one (P51) is 88.1 %. The average mean top efficiency of all the planks is 76.5 %.

Timo Enqvist has determined the efficiencies of the drift chambers in the stacks T5 and T6 and his efficiency vs. time curves were quite similar to mine, except that my efficiency curves were a few percentage units smaller. Tomi Räihä has calculated the efficiencies by using underground data, which included only two same planks as my study. He excluded the chambers with an efficiency less than 70 % and he obtained the mean efficiency of more than 90 %. His result is remarkably more than mine (76.5 %). The difference might be due to different external factors between underground and surface. Additionally my method of determining the efficiency may be stricter than the method of Enqvist and Räihä, because both of them obtained higher efficiency than I.

One challence of this work was to deduce sources of uncertainties of the efficiency results. Possible source of systematical uncertainties are my method of determining the efficiency and possible fault in data. A random inaccuracy may be due to the problems in the measurement system and method of determining the mean top efficiency.

My method of determining the efficiency in the main program may have

caused systematical uncertainty of the results. There is no one right way to determine the efficiency, but more strictly determined conditions give smaller efficiency and looser conditions may give falsely too high efficiency. The systematically low efficiency of the chamber X1 in each plank of the stacks T5, T6 and T7 may be due to a fault in data, because according also to Timo Enqvist's results for T5 and T6, the efficiency of X1 is systematically lower than other chambers' efficiency. It would not be very likely that in many planks just the chamber X1 is poor.

The problems in the measurement system may have caused momentary unstable efficiency which appear as peaks in the efficiency vs. time curves of my results. Part of these peaks could be explained by the external factors (pressure and temperature) and the problems with electronics. The different efficiency of a chamber's left and right part may be due to some problems in the grading lines. Additionally, the broken chambers are assumed to affect other chambers' efficiency, because the chambers are connected to the same HV supply. This appears as simultaneous peaks of the efficiency curves of several chambers. While determining the mean top efficiency, selecting the parts of the efficiency vs. time curves to be excluded may cause inaccuracy, because some peaks were not clear and sometimes they didn't appear exactly at the same time in different chambers.

The peaks of efficiency vs. position curves are mostly hard to explain, because it is not at all clear which factors may cause them. In some cases the peaks were local and in other cases the efficiency seemed to vary forcefully along the whole chamber length. Probably the asymmetry of the efficiency vs. position diagrams may be due to random inaccuracy (momentary problems in measurement system).

References

[1] *Astronomy and Astrophysics for the 1980's, (Vol 2)*. National Academies Press, 1983. ISBN 9780309554367. p. 57.

[2] P. Abreu et al. Search for ultrahigh energy neutrinos in highly inclined events at the pierre auger observatory. *PHYSICAL REVIEW D*, 84:122005, 2011. doi: 10.1103/PhysRevD.84.122005.

[3] T. J. Adye. *A Study of J Psi Production at the LEP e+e- Collider; and the Implementation of the DELPHI Slow Controls System*. PhD thesis, Lincoln College, Oxford, 1998.

[4] J. Angelo. *Nuclear Technology*. Greenwood Publishing Group, 2004. ISBN 9781573563369. p. 106-107.

[5] U. A. Bakshi et al. *Measurements And Instrumentation*. Technical Publications, 2009. ISBN 9788184316476. p. 1-60.

[6] J. Beringer et al. *Phys. Rev.*, D86(010001), 2012. (figure 26.8).

[7] J. Beringer et al. *Phys. Rev.*, D86(010001), 2012. (section 26).

[8] J. Beringer et al. *Phys. Rev.*, D86(010001), 2012. (figure 26.6).

[9] P. Bernardini. The macro detector at gran sasso. *Nuclear Instruments and Methods in Physics Research A*, 486:663–707, 2002.

[10] J. A. M. Bleeker et al. *The century of space science*. Springer, 2002. ISBN 9780792371960. p. 120-121.

[11] S. Cecchini et al. Atmospheric muons: experimental aspects. 2012. arXiv:1208.1171 [astro-ph.EP].

[12] F. E. Close et al. *The particle odyssey*. Oxford University Press, 2004. ISBN 9780198609438. p. 51-53.

[13] D. D'Enterria et al. Constraints from the first lhc data on hadronic event generators for ultra-high energy cosmic-ray physics. *Astroparticle Physics*, 35:98–113, 2011.

[14] J. Drexler. *Comprehending and decoding the cosmos*. Universal-Publishers, 2006. ISBN 9781581129298. s. 238.

[15] J. Drexler. *Comprehending and decoding the cosmos*. Universal-Publishers, 2006. ISBN 9781581129298. s. 242.

[16] EMMAgroup. *Logbook LI*. 2009. (Manually written logs of the calibration measurements with stacks T1-T7).

[17] EMMAgroup. *Logbook LII*. 2009-2010. (Manually written logs of the calibration measurements with stacks T8-T12).

[18] T. Enqvist. Email discussion "Elektronivyöryt" on 11 June 2013.

[19] T. Enqvist. Email discussion "Tehokkuuksista" on 29 April 2013.

[20] P. Freier et al. Evidence of heavy nuclei in the primary cosmic radiation. *Physical Review*, 74(2):213–217, July 1948.

[21] D. L. Gallagher. *Particle acceleration in astrophysical plasmas: geospace and beyond*. American Geophysical Union, 2005. ISBN 9780875904214. p. 72 (Article 'Studies of Relativistic Shock Acceleration' by A. Meli).

[22] L. Gerhardt et al. Lateral distribution of muons in icecube cosmic ray events. 2012. arXiv:1208.2979 [astro-ph.HE].

[23] P. Grieder. *Cosmic Rays at Earth*. Gulf Professional Publishing, 2001. ISBN 978044450710. p. 495.

[24] C. Grupen. *Astroparticle physics*. Springer, 2005. ISBN 9783540253129. s. 147.

[25] J. Hamblin. *Science in the Early Twentieth Century: An Encyclopedia*. ABC-CLIO, 2005. ISBN 9781851096657. p. 53.

[26] A. Haungs et al. A large area limited streamer tube detector for the air shower experiment kascade-grande. *Nuclear Instruments and Methods in Physics Research*, A 533:387–403, 2004.

[27] E. Heikkilä. About detection of multimuon bundles at the emma array. Master's thesis, Jyväskylän yliopisto, 2009.

[28] J. Heilbron. *The Oxford Companion to the History of Modern Science*. Oxford University Press, 2003. ISBN 9780195112290. p. 162.

[29] J. R. Hörandel. Models of the knee in the energy spectrum of cosmic rays. *Astroparticle Physics*, 21:241–265, 2004.

[30] J. R. Hörandel et al. Cosmic rays from the knee to the highest energies. *Progress in Particle and Nuclear Physics*, 63:293–338, Apr. 2009. arXiv:0904.0725v1.

[31] J. Joutsenvaara. Email discussion "paine" on 13 August 2010.

[32] T. Kalliokoski et al. Can emma solve the puzzle of the knee? *Progress in Particle and Nuclear Physics*, 66(2):468–472, 2011.

[33] K.-H. Kampert et al. Electron, muon, and hadron lateral distributions measured in air showers by the kascade experiment. *Astroparticle Physics*, 14:245–260, 2001.

[34] H. Kragh. *Quantum Generations - A History of Physics in The Twentieth Century*. Princenton University Press, 1999. ISBN 0691012067. p. 37.

[35] T. Kusky et al. *Encyclopedia of Earth and Space Science*. Infobase Publishing, 2010. ISBN 9781438128597. p. 173.

[36] P. Kuusiniemi et al. Underground multi-muon experiment emma. *Astrophysics and Space Sciences Transactions*, 7:93–96, 2011.

[37] P. Kuusiniemi et al. Underground cosmic-ray experiment emma. *Journal of Physics: Conference Series*, 409:012067, 2013. doi:10.1088/1742-6596/409/1/012067.

[38] M. F. L'Annunziata. *Radioactivity: Introduction and History*. Elsevier, 2007. ISBN 9780444527158. p. 349.

[39] M. F. L'Annunziata. *Handbook of Radioactivity Analysis*. Academic Press, 2012. ISBN 9780123848741. p. 123.

[40] C. Liu. *The Handy Astronomy Answer Book*. Visible Ink Press, 2008. ISBN 9781578591930. p. 172.

[41] B. Martin. *Nuclear and Particle Physics: An Introduction*. John Wiley & Sons, 2006. ISBN 9780470025321. p. 131-132.

[42] V. Mittal. *Introduction To Nuclear and Particle Physics*. PHI Learning Pvt. Ltd., 2009. ISBN 9788120343115. p. 373-374.

[43] M. Moldwin. *An Introduction to Space Weather*. Cambridge University Press. ISBN 9780521711128. p. 43.

[44] T. Monto. Emma-kokeen ajautumiskammioiden tehokkuuksien määrittämisestä, 2012. Special project, University of Jyväskylä.

[45] D. Müller. Cosmic rays beyond the knee. *Space Science Reviews*, 99(1/4):105–116, 2001.

[46] J. Narkilahti. Efficiency and resolution tests of delphi muon barrel chambers with cosmic-ray muons. Master's thesis, Oulun yliopisto, 2007. s. 43-48.

[47] M. Noz et al. *Radiation Protection - In the Health Sciences*. World Scientific, 1995. ISBN 9789810224066. p. 32-33.

[48] E. H. Oakes. *Encyclopedia of World Scientists*. Infobase Publishing, 2007. ISBN 9781438118826. p. 19.

[49] T. Räihä. *Chamber efficiency vs air pressure in stations C*. (The efficiency evaluated by using data from underground).

[50] T. Räihä. *Analysis tools for the EMMA experiment*. PhD thesis, University of Oulu, 2012.

[51] M. V. S. Rao. *Extensive Air Showers*. World Scientific, 1998. ISBN 9789810228880. p. 5.

[52] F. Schröder. *Instruments and Methods for the Radio Detection of High Energy Cosmic Rays*. Springer, 2012. ISBN 9783642336607. p. 5.

[53] Sharma. *Atomic And Nuclear Physics*. Pearson Education India, 2008. ISBN 9788131719244. p. 255-260.

[54] M. Spurio. Final results of the macro experiment. *Beyond the Desert 2003, Springer Proceedings in Physics*, 92:925–939, 2004. doi: 10.1007/978-3-642-18534-2_61.

[55] J. Timmermans. Study of multi-muon bundles in cosmic ray showers detected with the delphi detector at lep. *Astroparticle Physics*, 28:273–286, 2007.

[56] M. Veltman. *Facts and Mysteries in Elementary Particle Physics*. World Scientific, 2003. ISBN 9789812381491. p. 164.

[57] A. Watson. *The quantum quark*. Cambridge University Press, 2004. ISBN 9780521829076. p. 124.

A Programs

A.1 Efficiency-v3.cpp

```
//  CREATE FILE LIST (text file):
//  ls -1 *.gd > T9-Files.txt
//  (delete an empty row in the end of the file)
//
//  RUN THE CODE:
//  make clean, make, ./Efficiency
//  when asking, give station number (5,6,7,8,9 or 10) and number of files

#include <cstring>
#include <cstdlib>
#include <iostream>
#include <fstream>
#include <cstdio>
#include <cmath>
#include <iomanip>
#include <map>
#include <vector>
#include <ctime>
#include <time.h>

/***
#include <TApplication.h>
#include <TCanvas.h>
#include <TH1F.h>
#include <TF1.h>
#include <TGraph.h>
***/

using namespace std;

#define CHAMBERS     7
#define CODEFILEPATH  "/home/kulmalukko/PlankEff/Code/"
#define CALIBFILEPATH "/home/kulmalukko/PlankEff/Calibration/T"

const char *XY[CHAMBERS] = { "X1", "Y1", "X2", "Y2", "X3", "Y3", "X4" };
const char *XXYYLR[2*(CHAMBERS-1)] = { "X1R", "Y1L", "X2L", "Y1R", "X2R", "Y2L", "X3L",
                                       "Y2R", "X3R", "Y3L", "X4L", "Y3R" };

//the bigger this is, the bigger calculated efficiency and the bigger number of approved events
const int maxDistanceFromFit = 60, calibNearFarMaxDifference = 30;


// CREATE map types: http://en.wikipedia.org/wiki/Associative_containers_(C%2B%2B)#Iterators

typedef map <string, vector<int> > MapTypeVV;
typedef map <string, MapTypeVV > MapTypeCalTable;

typedef map <string, double > MapTypePoint;

typedef map <string, double > MapTypeHeight;

typedef map <string, double > MapTypeFitPoint;

typedef map <string, int > MapTypeMeasure;

typedef map <string, vector<int> > MapTypeInt;
typedef map <string, vector<double> > MapTypeDouble;

typedef map <string, MapTypeInt> MapTypeCounter;


int kx = 16, ky = 43; // x & y chambers distance from the bottom


//::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

//The method of least squares makes fitting using the data points of reference planks (4 planks, 8 points).
//This function makes the fitting by using the measured points xcal ja ycal.

bool FitTrack(double xcal[], double ycal[], double& aParam, double& bParam)
{
    //double& aParam means, that aParam can be modified without return

    bool trackOk;

    // EVALUATE parameters to build least squares fit

    double sumxi=0.0, sumyi=0.0, sumxiyi=0.0, sumxxi=0.0, sumyyi=0.0;

    // EVALUATE npoints = number of approved points
    //          ipoints = index of points

    double npoints=0.0;

    for( int ipoints = 0; ipoints < 8; ipoints++ )
    {
        if( xcal[ipoints] >= 0 && xcal[ipoints] <= 3650 )  //from data file: xcal[0] == refPoint_P15x etc.
```

```
        {
            npoints++;
            sumxi += xcal[ipoints];
            sumyi += ycal[ipoints];
            sumxxi+= ( xcal[ipoints] * xcal[ipoints] );
            sumyyi += ( ycal[ipoints] * ycal[ipoints] );
            sumxiyi += ( xcal[ipoints] * ycal[ipoints] );
        }
    }

    double Divider = npoints*sumxxi - sumxi*sumxi;

    if( Divider == 0 )
    {
        aParam=0.0;
        bParam=0.0;
    }
    // fitting parameters: aParam & bParam: y=aParam*x + bParam
    else
    {
        aParam=(( npoints*sumxiyi - sumxi*sumyi ) / Divider);
        bParam=(( sumxxi*sumyi - sumxi*sumxiyi ) / Divider);
    }

    if(npoints == 8){ trackOk = true; }
    else{ trackOk = false; }

    return trackOk;
}
//:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

//This function calculate a (reference plank's) measured point distance from the fitting.

bool ArePointsNearEnoughtFit(double x[], double y[], double& aParam, double& bParam)
{
/*  point's coordinates ycal(i), xcal(i)
    ycal(i)=p0+p1*x  -->   x = ( ycal(i) - p0 ) / p1
    point's horizontal distance from fitting: L = x-xcal(i)
*/
    //cout<<"aParam "<<aParam<<endl;
    //cout<<"bParam "<<bParam<<endl;

    bool areAllPointsOk = true;

    double fitX, distance;

    // calculate every reference planks' signals' distance from fitting
    for( int k = 0; k != 8; k++ )
    {
        fitX = ( y[k] - bParam )/aParam;
        distance = abs( x[k] - fitX );

        if( distance > maxDistanceFromFit ){ areAllPointsOk = false; break;}
    }

    return areAllPointsOk;
}
//:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

//This function jumps over the vain lines of file.

void JumpOverFileLines(std::ifstream &inFile, int uselessLines)
{
    //uselessLines = number of lines, which must be jumped over
    //uselessLine = index of a useless line
    char temppi[500];
    int uselessLine;

    for( uselessLine = 0; uselessLine < uselessLines; uselessLine++ )
    {
        inFile.getline(temppi,500);
    }
}
//:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

//This function calculate the measured point's time channel index.

int CalculateTimeChanIndex(int PA, int PB)
{
    int PAB = PA - PB; //aikakanavien erotus (N/F - A)
    int index = PAB + 100;
    return index;
}
//:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

//This function modify CalTables by copy pasting values from files P**-auto.cal into the tables

void FillCalTables( int numberOfPlanks, vector<string> CPLANKS, const char *XY[], char station[],
              char calibFile[], MapTypeCalTable& CalTable_N, MapTypeCalTable& CalTable_F )
{
    int channel, itemp, i_plank, i_cham, i_channel;
    string PL, KA;

    for( i_plank = 0; i_plank < numberOfPlanks; i_plank++ )
    {
        PL = CPLANKS[i_plank];
```

```
        //adjust to zero
        for( i_cham = 0; i_cham < 7; i_cham++ )
        {
            KA = XY[i_cham];

            CalTable_N[PL][KA].resize(1201);
            CalTable_F[PL][KA].resize(1201);
        }

        //set the calibration file name
        strncpy( calibFile, CALIBFILEPATH, 40 );
        strncat( calibFile, station, 3 );
        strncat( calibFile, "/", 2 );
        strncat( calibFile, PL.c_str(), 4 );
        strncat( calibFile, "-auto.cal", 10 );

        //OPEN calibration file
        ifstream inCalibFile;
        inCalibFile.open(calibFile,ios::in);
        if( !inCalibFile )
        {
            cout<<"   ("<<calibFile<<")"<<endl;
            cerr << " <E> Can't open File inCalibFile of plank "<<PL<< endl;
            exit( -2 );
        }

        //Jump over first 3 lines
        JumpOverFileLines(inCalibFile, 3);

        //FILL calibration table
        for( i_channel = -100; i_channel != 1101; i_channel++ )
        {
            channel=i_channel+100;

            inCalibFile>>itemp;  //position coordinates of chamber

            for( i_cham = 0; i_cham < 7; i_cham++ )
            {
                KA = XY[i_cham];
                inCalibFile >> CalTable_N[PL][KA].at(channel);
                inCalibFile >> CalTable_F[PL][KA].at(channel);
            }
        }

        inCalibFile.close();
    }

}
//:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

//This function defines the chamber variables

void DefineChamberVariables(char chamPair[], char xCham[], char yCham[], char xChamHalf[], char yChamHalf[])
{
    // overlapping chambers: x1R y1L,  x2L y1R,  x2R y2L,  x3L y2R,  x3R y3L,  x4L y3R

    if( strncmp( chamPair, "X1Y1", 4 ) == 0 )
    {
        strncpy(xChamHalf,"X1R", 3);
        strncpy(yChamHalf,"Y1L", 3);
        strncpy(xCham,"X1", 2);
        strncpy(yCham,"Y1", 2);
    }

    if( strncmp( chamPair, "X2Y1", 4 ) == 0 )
    {
        strncpy(xChamHalf,"X2L", 3);
        strncpy(yChamHalf,"Y1R", 3);
        strncpy(xCham,"X2", 2);
        strncpy(yCham,"Y1", 2);
    }

    if( strncmp( chamPair, "X2Y2", 4 ) == 0 )
    {
        strncpy(xChamHalf,"X2R", 3);
        strncpy(yChamHalf,"Y2L", 3);
        strncpy(xCham,"X2", 2);
        strncpy(yCham,"Y2", 2);
    }

    if( strncmp( chamPair, "X3Y2", 4 ) == 0 )
    {
        strncpy(xChamHalf,"X3L", 3);
        strncpy(yChamHalf,"Y2R", 3);
        strncpy(xCham,"X3", 2);
        strncpy(yCham,"Y2", 2);
    }

    if( strncmp( chamPair, "X3Y3", 4 ) == 0 )
    {
        strncpy(xChamHalf,"X3R", 3);
        strncpy(yChamHalf,"Y3L", 3);
        strncpy(xCham,"X3", 2);
        strncpy(yCham,"Y3", 2);
    }
```

```
    if( strncmp( chamPair, "X4Y3", 4 ) == 0 )
    {
        strncpy(xChamHalf,"X4L", 3);
        strncpy(yChamHalf,"Y3R", 3);
        strncpy(xCham,"X4", 2);
        strncpy(yCham,"Y3", 2);
    }

    xCham[2] = '\0';
    yCham[2] = '\0';
    //cout << "chambers:  " << chamPair << " | ";
    //cout << "xCham, yCham: " << xCham << ", " << yCham;
    //cout << " | xChamHalf, yChamHalf: "<<xChamHalf<<", "<<yChamHalf<<endl;
}
//::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

//This gives booleans, which check if fit points are between 0-3650 and are also short planks penetrated

void CheckFitPoints(int numberOfPlanks, double aParam, double bParam, vector<string> CPLANKS,
                MapTypeHeight map_heightTable, int kx, int ky,
                MapTypeFitPoint& fitPoint_X, MapTypeFitPoint& fitPoint_Y,
                bool& areFitPointsBetw0_3650, bool& fitPenetratesShortChamber)
{
    int i_plank;
    double height_X, height_Y;
    string PL;

    for( i_plank = 0; i_plank < numberOfPlanks; i_plank++ )
    {
        PL = CPLANKS[i_plank];

        // Save the fit track points into the variables fitPoint
        height_X = map_heightTable[PL] + kx;
        height_Y = map_heightTable[PL] + ky;

        fitPoint_X[PL] = ( height_X - bParam ) / aParam;
        fitPoint_Y[PL] = ( height_Y - bParam ) / aParam;

        // Check are the fit points between 0-3650
        if(fitPoint_X[PL] < 0 || fitPoint_X[PL] > 3650 || fitPoint_Y[PL] < 0 || fitPoint_Y[PL] > 3650 )
        {
            areFitPointsBetw0_3650 = false;
        }


        // OBS! Planks P81-84 are especially short.
        // If fitting doesn't penetrate every plank, the event isn't taken into account
        if( PL == "P81" || PL == "P82" || PL == "P83" || PL == "P84" )
        {
            if
            (
                fitPoint_X[PL] < 350 || fitPoint_X[PL] > 3300
                || fitPoint_Y[PL] < 350 || fitPoint_Y[PL] > 3300
            )
            {
                fitPenetratesShortChamber = false;
            }
        }
    }
}
//::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

// This function adds goodPoints and map_PosCounterUp, which are later used to calculate efficiency
// I put '&' symbols for all map parameters, because without '&' the code is running very slowly.
// Only the two last maps are meant to be modified in this function.

void AddGoodPoints(string PL, char iCham[],
                MapTypeMeasure& measuredTime_iN,
                MapTypeMeasure& measuredTime_iA, MapTypeMeasure& measuredTime_iF,
                MapTypeCalTable& CalTable_N, MapTypeCalTable& CalTable_F, MapTypeFitPoint& fitPoint_i,
                MapTypeMeasure& goodPoints_i, MapTypeCounter& map_PosCounterUp)
{
    int index1, index2, calibPoint_iN, calibPoint_iF, nearFarDifference_i;
    double point_i, point_distance_i;

    // Check if the point is ok
    if
    (
        measuredTime_iN[PL] != -9999 &&
        measuredTime_iA[PL] != -9999 &&
        measuredTime_iF[PL] != -9999
    )
    {
        // index = near-anodi + 100  or  far-anodi + 100
        index1 = CalculateTimeChanIndex( measuredTime_iN[PL], measuredTime_iA[PL] );
        index2 = CalculateTimeChanIndex( measuredTime_iF[PL], measuredTime_iA[PL] );

        if( 0 <= index1 && index1 <= 1200 && 0 <= index2 && index2 <= 1200 )
        {
            calibPoint_iN = CalTable_N[PL][iCham].at(index1);// x1 y1 x2 y2...
            calibPoint_iF = CalTable_F[PL][iCham].at(index2);

            nearFarDifference_i = abs( calibPoint_iN - calibPoint_iF );

            point_i = ( (double)( calibPoint_iN + calibPoint_iF ) / 2 );
```

```
            point_distance_i = abs( point_i - fitPoint_i[PL] );

            // If point is between 0-3650, near & far are near each other
            // and distance from fit is small, approve it.
            if
            (
                point_i >= 0 && point_i <= 3650  &&
                point_distance_i < maxDistanceFromFit &&
                nearFarDifference_i < calibNearFarMaxDifference
            )
            {
                goodPoints_i[PL]++;
                //if( (int) (point_X/4) == (int) (fitPoint_X[PL]/4) ){
                //map_PosCounterUp[PL][xCham].at( (int) ( point_X / 4 ) )++; }
                map_PosCounterUp[PL][iCham].at( (int) ( fitPoint_i[PL] / 4 ) )++;
            }
        }
    }
}


// = = = = = = = = = = = = = = = = = = = = M A I N = = = = = = = = = = = = = = = = = = = =

int main(int argc, char* argv[])
{

    //ooooooooooooo CREATE VARIABLES ooooooooooooo


    //+++++ STATION, PLANK & FILE VARIABLES +++++

    // station number, plank name
    char station[3], plankName[4];

    // number of chars in station, number of choosed files & total files
    int stationChars, filesChoosed, filesMax = 0;

    // lines in plank list & number of planks
    int plankListLines = 0, numberOfPlanks;

    // station height & plank distance from top
    double stationHeight=1800.0, distanceFromTop;

    vector<string> CPLANKS; //plank names
    string PL, KA, KAP;

    //chamber (later these are copied from file name)
    char chamPair[5], xChamHalf[4], yChamHalf[4], xCham[3], yCham[3];

    // file names
    char filelist[20] = "";      // ..Files.txt
    char planklist[75] = "";         // kansio/..levels.dat
    char dataFileName[40];       // EMMA...gd
    char dataFileNameFull[70];   // kansio/EMMA...gd
    char calibFile[55];          // ../Calibration/../..auto.cal
    char EffPosLog[20];          // ..EffPos..gd
    char EffTimeLog[20];         // ..EffTime..gd

    string fileLine;


    // +++++ MISC VARIABLES +++++

    //these are indexies used in for loops
    int i_datafile, i_datafileLine;
    int i_plank, i_cham/*, i_channel*/, i_time, i_pos;

    //"roskis"
    double itemp;

    //general counter
    int counter;

    // Plank height
    MapTypeHeight map_heightTable;

    int kx = 16, ky = 43; // x & y chambers distance from the bottom

    // durationMax = max duration of measurements
    // durationSelectedMax = duration of selected files
    int durationMax, durationSelectedMax;


    //++++++ CALCULATORS AND EFFICIENCY VARIABLES ++++++

    MapTypeCounter map_PosCounterUp, map_PosCounterDown;

    double effPosUp, effPosDown, eff_value, eff_sum;

    typedef map <string, MapTypeDouble> MapTypeEffTime;
    MapTypeEffTime map_EffTime;


    // +++++ CALIBRATION VARIABLES +++++


    // calibration tables (plankki, kammio, kanava)
```

```
MapTypeCalTable CalTable_N, CalTable_F;

//+++++ MEASUREMENT VARIABLES +++++

unsigned long momentOfTime;

long int timeFileFirst, timeStationFirst, timeMid, timeLast;

// signals of the reference planks
int refPoint_P15x, refPoint_P15y,  refPoint_P39x, refPoint_P39y;
int refPoint_P18x, refPoint_P18y,  refPoint_P17x, refPoint_P17y;

// measured data (time) and good points in chamber
MapTypeMeasure measuredTime_XA, measuredTime_XN, measuredTime_XF;
MapTypeMeasure measuredTime_YA, measuredTime_YN, measuredTime_YF;
MapTypeMeasure goodPoints_X, goodPoints_Y;

// boolean variables
bool fitPenetratesShortChamber, areRefPointsNearFit, areRefPointsBetw0_3650, isUpDownOk;
bool areFitPointsBetw0_3650;

// +++++ FIT VARIABLES +++++

// EVALUATE the coordinates of reference planks (these are used when making fit)
double referencePoint[8] = {0.0}, referenceHeight[8] = {0.0};

// EVALUATE parameters of least squares fitting
double aParam, bParam;

// points in fit track
MapTypeFitPoint fitPoint_X, fitPoint_Y;

int long goodFits, allGoodFits, allEvents=0;

//ooooooooooooooo USER CHOOSES THE STATION ooooooooooooooo

cout<<"Choose the station number (5, 6, 7, 8, 9 or 10):  ";
cin >> station;

//number of the characters in station number (1 or 2)
stationChars =  strlen(station);

//ooooooooooooooo APPROXIMATE THE MAX. DURATION OF MEASUREMENT ooooooooooooooo

//text file "filelist" includes the names of the measurement data files
strncat( filelist, "T", 1 );
strncat( filelist, station, 3 );
strncat( filelist, "-Files.txt", 10 );

ifstream inFilelist;
inFilelist.open( filelist );

//Calculate the number of the files = number of the lines in filelist
while( getline(inFilelist, fileLine) )
{
    filesMax++;
}

//with clear the file can be read again
if(inFilelist.tellg() == -1){ inFilelist.clear(); }

inFilelist.seekg (0, ios::beg);

//Station measurement approximated maximum duration (number of files * 8h * 2)
durationMax = filesMax*8*2;

//ooooooooooooooo USER CHOOSES THE NUMBER OF FILES ooooooooooooooo

cout<<"In station T"<<station<<" is "<<filesMax<<" datafiles."<<endl;

cout<<"  ( Estimation: measurement duration is not higher than "<<durationMax<<" hours. )"<<endl;

cout<<"How many data files do you want to go through? (max "<<filesMax<<"): ";
cin >> filesChoosed;
cout<<endl;

//durationSelectedMax is approximated max. duration of selected files
durationSelectedMax = filesChoosed*8*2;

//ooooooooooooooo CHECK UP THE PLANKS AND THEIR HEIGHTS IN THE STATION ooooooooooooooo

//Plank names and heights are listed in planklist
strncat( planklist, CODEFILEPATH, 35 );
strncat( planklist, "T", 2 );
strncat( planklist, station, 3 );
strncat( planklist, "-levels.dat", 15 );

cout<<"planklist:   "<<planklist<<endl;

ifstream inPlanklist;
```

```
inPlanklist.open(planklist,ios::in);

if( !inPlanklist )
{
    cerr << " <E> Can't open File inPlanklist" << endl;
    exit( -2 );
}

JumpOverFileLines(inPlanklist, 1);


//GO THROUGHT planklist -> put plank names into vector & fill the map
while( !inPlanklist.eof() )
{
    inPlanklist >> plankName >> distanceFromTop ;

    plankListLines++;
    if( plankListLines > 4 ) //first 4 are the reference planks
    {
        CPLANKS.push_back(plankName);
    }

    map_heightTable[plankName] = stationHeight - distanceFromTop;
    }

numberOfPlanks = plankListLines-4;

// "Iterate over the map and print out all key/value pairs.
// Using a const_iterator since we are not going to change the values."
MapTypeHeight::const_iterator end = map_heightTable.end();
for ( MapTypeHeight::const_iterator it = map_heightTable.begin(); it != end; ++it )
{
    cout << it->first << " " << it->second << '\n';
}

inPlanklist.close();


//ooooooooooooooo ADJUST THE EFFICIENCY CALCULATORS ooooooooooooooo


for( i_plank = 0; i_plank < numberOfPlanks; i_plank++ )
{
    PL = CPLANKS[i_plank];

    for( i_cham = 0; i_cham != 2*(CHAMBERS-1); i_cham++ )
    {
        map_EffTime[PL][XXYYLR[i_cham]].resize(durationMax, -6666);
    }

    for( i_cham = 0; i_cham < 7; i_cham++ )
    {
        map_PosCounterUp[PL][XY[i_cham]].resize(914);
        map_PosCounterDown[PL][XY[i_cham]].resize(914);
    }
}


//ooooooooooooooo FILL CALIBRATION TABLE ooooooooooooooo


//This function fills the calibration table by using files P**-auto.cal
FillCalTables(numberOfPlanks, CPLANKS, XY, station, calibFile, CalTable_N, CalTable_F );


//ooooooooooooooo GO THROUGH THE DATA FILES ooooooooooooooo


for( i_datafile = 0; i_datafile != filesChoosed; i_datafile++)
{
    cout<<endl;
    cout<<"=======  DATAFILE BEGINS  ======="<<endl;

    // adjust these variables to zero

    i_datafileLine = 0;
    goodFits = 0;

    for( i_plank = 0; i_plank < numberOfPlanks; i_plank++ )
    {
        PL = CPLANKS[i_plank];
        goodPoints_X[PL]=0;
        goodPoints_Y[PL]=0;
    }


    //**************** MEASUREMENT FILE *****************


    // SET file name into dataFileName
    inFilelist >> dataFileName;

    strncpy( dataFileNameFull, "/media/HD-PCU2/EMMA/Teline-", 30 );
    strncat( dataFileNameFull, station, 3 );
    strncat( dataFileNameFull, "/", 1 );
    strncat( dataFileNameFull, dataFileName, 40 );

    // READ chamPair from file name
    strncpy( chamPair, &dataFileName[29+stationChars], 4);
```

```
cout << " dataname:    " << dataFileName << " (" << i_datafile << ")" << endl;
chamPair[4] = '\0';

//**************** CHAMBER PAIRS *****************

//This function modifies chamber names
DefineChamberVariables(chamPair, xCham, yCham, xChamHalf, yChamHalf);

//**************** OPEN MEASUREMENT DATAFILE *****************

ifstream inFile;
inFile.open( dataFileNameFull, ios::in );
if( !inFile )
{
    cerr << "Can't open file " << dataFileNameFull << endl;
    return( -1 );
}

//JUMP OVER first 7 lines
JumpOverFileLines( inFile, 7 );


//***************** GO THROUGH EVENTS *****************

while( !inFile.eof() )
{
    i_datafileLine++;

    fitPenetratesShortChamber = true;
    areFitPointsBetw0_3650 = true;

    // FORMAT measurement variables
    for( i_plank = 0; i_plank < numberOfPlanks; i_plank++ )
    {
        PL = CPLANKS[i_plank];
        measuredTime_XA[PL]=-888;
        measuredTime_XN[PL]=-888;
        measuredTime_XF[PL]=-888;
        measuredTime_YA[PL]=-888;
        measuredTime_YN[PL]=-888;
        measuredTime_YF[PL]=-888;
    }

    refPoint_P15x=-888;
    refPoint_P15y=-888;
    refPoint_P39x=-888;
    refPoint_P39y=-888;
    refPoint_P18x=-888;
    refPoint_P18y=-888;
    refPoint_P17x=-888;
    refPoint_P17y=-888;

    // ---------------------------- READ the signals from datafile ----------------------------

    //READ reference planks' signals (position)
    inFile >>
    refPoint_P15x >> itemp >> itemp >> itemp >> refPoint_P15y >> itemp >> itemp >> itemp >>
    refPoint_P39x >> itemp >> itemp >> itemp >> refPoint_P39y >> itemp >> itemp >> itemp >>
    refPoint_P18x >> itemp >> itemp >> itemp >> refPoint_P18y >> itemp >> itemp >> itemp >>
    refPoint_P17x >> itemp >> itemp >> itemp >> refPoint_P17y >> itemp >> itemp >> itemp ;

    //READ researchable planks' signals (time)
    for( i_plank = 0; i_plank < numberOfPlanks; i_plank++ )
    {
        PL = CPLANKS[i_plank];

        inFile >> measuredTime_XA.at(PL)  >> measuredTime_XN.at(PL) >> measuredTime_XF.at(PL) >>
                  measuredTime_YA.at(PL) >> measuredTime_YN.at(PL) >> measuredTime_YF.at(PL) ;
    }

    inFile >> momentOfTime;

    //The below presents the seconds as date
    //cout << ctime((time_t *)&momentOfTime);

    if( i_datafileLine == 1 )
    {
        timeFileFirst = momentOfTime;

        if( i_datafile == 0 ){ timeStationFirst = momentOfTime; }
    }

    timeLast = momentOfTime;

    // ---------------------------- REFERENCE POINTS ----------------------------

    referencePoint[0] = refPoint_P15x;
    referencePoint[1] = refPoint_P15y;
    referencePoint[2] = refPoint_P39x;
```

```
    referencePoint[3] = refPoint_P39y;
    referencePoint[4] = refPoint_P18x;
    referencePoint[5] = refPoint_P18y;
    referencePoint[6] = refPoint_P17x;
    referencePoint[7] = refPoint_P17y;

    referenceHeight[0] = map_heightTable["P15"]+kx;
    referenceHeight[1] = map_heightTable["P15"]+ky;
    referenceHeight[2] = map_heightTable["P39"]+kx;
    referenceHeight[3] = map_heightTable["P39"]+ky;
    referenceHeight[4] = map_heightTable["P18"]+kx;
    referenceHeight[5] = map_heightTable["P18"]+ky;
    referenceHeight[6] = map_heightTable["P17"]+kx;
    referenceHeight[7] = map_heightTable["P17"]+ky;

    // ------------------------------ CHECK IF THE FIT IS GOOD ------------------------------

    // If in any plank distance of x & y chambers is more than 60mm, event must not be approved
    if
    (
        abs(refPoint_P15x - refPoint_P15y) > 60 ||
        abs(refPoint_P39x - refPoint_P39y) > 60 ||
        abs(refPoint_P18x - refPoint_P18y) > 60 ||
        abs(refPoint_P17x - refPoint_P17y) > 60
    )
    {
        isUpDownOk = false;
    }
    else{ isUpDownOk = true; }

    // CALL function, which makes least squares fitting using referencePoints & referenceHeights
    // boolean return tells if all ref points are reasionable (0-3650)
    areRefPointsBetw0_3650 = FitTrack( referencePoint, referenceHeight, aParam, bParam );

    // CHECK if all ref points are near enought the fit
    areRefPointsNearFit = ArePointsNearEnoughtFit( referencePoint,
                                    referenceHeight, aParam, bParam );

    // CHECK if all fit points are between 0-3650
    // and in case of short planks they are penetrated by the fit
    CheckFitPoints(numberOfPlanks, aParam, bParam, CPLANKS,
                map_heightTable, kx, ky, fitPoint_X, fitPoint_Y,
                areFitPointsBetw0_3650, fitPenetratesShortChamber);

    // CHECK UP if the conditions are fullfilled, fit will be approved by adding goodFits
    if
    (
        isUpDownOk == true &&           // x & y signals are near each other
        areRefPointsBetw0_3650 == true &&   // all ref points between 0 & 3650
        areRefPointsNearFit == true &&      // all ref points max. 60mm from fit
        fitPenetratesShortChamber == true &&    // short plank is penetrated
        areFitPointsBetw0_3650 == true      // all fit points in planks between 0 & 3650
    )
    {
        goodFits++;
    }
    else{ continue; }

    // ------------------------------ RESEARCHABLE PLANKS' POINTS ------------------------------

    // Chamber length 3650mm may be divided into 4mm parts -> 3650/4=912.5,  (3650-2)/4=912
    // Parts are 0..4mm, 4..8mm etc..

    for( i_plank = 0; i_plank < numberOfPlanks; i_plank++ )
    {
        PL = CPLANKS[i_plank];

        map_PosCounterDown[PL][xCham].at( (int) ( fitPoint_X[PL] / 4 ) )++;
        map_PosCounterDown[PL][yCham].at( (int) ( fitPoint_Y[PL] / 4 ) )++;


        // If point is good, this function adds goodPoints & map_PosCounterUp (for X & Y chambers)
        AddGoodPoints(PL, xCham, measuredTime_XN, measuredTime_XA, measuredTime_XF,
                    CalTable_N, CalTable_F, fitPoint_X, goodPoints_X, map_PosCounterUp);

        AddGoodPoints(PL, yCham, measuredTime_YN, measuredTime_YA, measuredTime_YF,
                    CalTable_N, CalTable_F, fitPoint_Y, goodPoints_Y, map_PosCounterUp);
    }

} // ------------------------------ EVENT ENDS HERE


// ********************** SET FILE TIME & ADD COUNTER VARIABLES ***************************


// mean time of this file (hours from measurement start)
timeMid = ( (timeLast+timeFileFirst)/2 - timeStationFirst ) / 3600 ;

//cout<<timeMid<<" "<<( timeFileFirst - timeStationFirst )/3600;
//cout<<" "<<( timeLast - timeStationFirst )/3600<<endl;

cout<<endl;
cout <<"All events: "<<i_datafileLine<<", good fits: "<<goodFits<< endl;
```

```
    cout<<endl;

    allGoodFits += goodFits;
    allEvents += i_datafileLine;

    cout << "    Chamber, approved points & efficiency "<<endl;

    eff_sum=0;


    // ********************** CALCULATE EFFCIENCY-TIME ***************************


    for( i_plank = 0; i_plank < numberOfPlanks; i_plank++ )
    {
        PL = CPLANKS[i_plank];
        //cout<<timeMid<<endl;
        map_EffTime[PL][xChamHalf].at(timeMid) = (double)goodPoints_X[PL] / (double)goodFits;
        map_EffTime[PL][yChamHalf].at(timeMid) = (double)goodPoints_Y[PL] / (double)goodFits;

        cout << PL<<"x    " << goodPoints_X[PL]  << "    ";
        cout << map_EffTime[PL][xChamHalf][timeMid]  << endl;
        cout << PL<<"y    " << goodPoints_Y[PL]  << "    ";
        cout << map_EffTime[PL][yChamHalf][timeMid]  << endl;

        eff_sum += map_EffTime[PL][xChamHalf][timeMid] + map_EffTime[PL][yChamHalf][timeMid];
    }

    cout<<" --> mean efficiency of the above:  "<<100*eff_sum/(numberOfPlanks*2)<< "%" <<endl;

//------------------------------ DATAFILE ENDS HERE


cout<<"---------- DATATIEDOSTOT KAYTY LAPI ----------"<<endl;

cout<<endl;
cout<<"    "<<100*(double)(allGoodFits) / (double)(allEvents)<<"  % of all the events are good."<<endl;
cout<<endl;


//ooooooooooooooo FILL LOG FILES EFF-POSITION & EFF-TIME ooooooooooooooo


for( i_plank = 0; i_plank < numberOfPlanks; i_plank++ )
{
    PL = CPLANKS[i_plank];

    // - - - - - - - - - - - - - - - - - EFF - POSITION - - - - - - - - - - - - - - - - -


    strncpy( EffPosLog, "T", 2 );
    strncat( EffPosLog, station, 3 );
    strncat( EffPosLog, "-EffPos-", 10 );
    strncat( EffPosLog, PL.c_str(), 4 );
    strncat( EffPosLog, ".gd", 4 );

    ofstream ofEffPos;
    ofEffPos.open(EffPosLog, ios::out);
    ofEffPos <<"channel     position        effX1            effY1";
    ofEffPos<<"             effX2              effY2";
    ofEffPos<<"              effX3              effY3              effX4        "<<endl;

    for( i_pos = 0; i_pos < 913; i_pos++ )
    {
        ofEffPos <<i_pos<<"              "<< "["<<i_pos*4<<","<<i_pos*4+4<<"[            ";

        for( i_cham = 0; i_cham < 7; i_cham++ )
        {
            KA = XY[i_cham];

            effPosUp = (double)(map_PosCounterUp[PL][KA].at(i_pos));
            effPosDown = (double)(map_PosCounterDown[PL][KA].at(i_pos));

            eff_value = effPosUp / effPosDown;

            //cout<<PL<<" "<<KA<<" "<<i_pos<<" "<<effPosUp<<" "<<effPosDown<<"  "<<eff_value<<endl;

            // C R E A T E    L O G    F I L E

            if( eff_value > 0 && eff_value <= 1 )
            {
                //if this is the last parameter in line, no need for space
                if( i_cham==6 ){ ofEffPos << eff_value; }
                else{ ofEffPos << eff_value << "           "; }
            }
            else
            {
                if( i_cham==6 ){ ofEffPos << 0; }
                else{ ofEffPos << 0 << "           "; }
            }
        }

        if(i_pos != 912){ ofEffPos<<endl; }
    }
    ofEffPos.close();

    // - - - - - - - - - - - - - - - - - EFF - TIME - - - - - - - - - - - - - - - - -
```

```
        strncpy( EffTimeLog, "T", 2 );
        strncat( EffTimeLog, station, 3 );
        strncat( EffTimeLog, "-EffTime-", 10);
        strncat( EffTimeLog, PL.c_str(), 4 );
        strncat( EffTimeLog, ".gd", 3 );

        ofstream ofTehoAika;
        ofTehoAika.open(EffTimeLog, ios::out);
        ofTehoAika << " chamber     eff     time (h) " << endl;

        counter = 0;

        for( i_cham = 0; i_cham != 2*(CHAMBERS-1); i_cham++ )
        {
            KAP = XXYYLR[i_cham];

            for( i_time = 0; i_time < durationSelectedMax; i_time++ )
            {
                eff_value = map_EffTime[PL][KAP].at(i_time);

                if( eff_value == -6666 ){ continue; }

                // jos ei ole eka lokitiedoston rivi, tehdaan rivinvaihdos
                if( counter != 0 ){ ofTehoAika << endl; }

                counter++;

                // C R E A T E    L O G    F I L E

                if( eff_value > 0 && eff_value <= 1)
                {
                    ofTehoAika << KAP << "          "<< eff_value << "         "<< i_time ;
                }
                else
                {
                    ofTehoAika << KAP << "          "<< 0 << "         "<< i_time ;
                }

            }

        }

        ofTehoAika.close();
    }

    inFilelist.close();

    cout<<"test (just before return) "<<endl;
    return( 0 );

}//------------------------------ MAIN ENDS HERE
```

A.2 Eff-Time.c

```
// Start ROOT and run the code giving the command .x Eff-Time.c
// Programs ask the stack number (5, 6, 7, 8, 9, 10)
// Program draws the figures efficiency-time for the planks of a given station
//
// ROOT must be turned off before new running
//
{
    #include <map>
    #include <string>
    #include <iomanip> // makes setprecision(9) to operate

    gROOT->Reset();

    cout<<setprecision(9)<<endl; //all decimals output 9 digits

    //***** VARIABLES *****

    //chamber names (* pointer)
    char *XY[7] = { "X1", "Y1", "X2", "Y2", "X3", "Y3", "X4" };
    char *XXYYLR[12] = { "X1R", "Y1L", "X2L", "Y1R", "X2R", "Y2L", "X3L",
                         "Y2R", "X3R", "Y3L", "X4L", "Y3R" };

    char station[4] = "", stationFile[55] = "", fileList[50] = "";

    int stationFileLines = 0;

    char temppi[500];

    char dataFirst[50] = "";
    string dataEnd = ".gd";

    char plankName[4];
    vector<string> CPLANKS; //plank name table

    double itemp;

    int filesMax=0, timeMax=0;
    string lineTiedosto;

    // VARIABLES for calculating mean efficiency per chamber
    double effSum_X1, effSum_X2, effSum_X3, effSum_X4, effSum_Y1, effSum_Y2, effSum_Y3;
    //double effSum_X1R, effSum_X2R, effSum_X2L, effSum_X3R, effSum_X3L, effSum_X4L,
    //            effSum_Y1R, effSum_Y1L, effSum_Y2R, effSum_Y2L, effSum_Y3R, effSum_Y3L;
    double effMean_X1, effMean_X2, effMean_X3, effMean_X4, effMean_Y1, effMean_Y2, effMean_Y3;

    //***** MORE VARIABLES *****

    string PL;

    //these paremeters are used to read the values of the file
    char read_chamb[22];
    double read_eff;
    double eff_sum;
    int read_time;

    int points;

    //index used in for
    int iChamPart, iplank, iTime;

    typedef map <string, vector<double> > MapTypeEff;
    MapTypeEff effMap;

    //data file name of plank
    string dataNimi;

    //legend position
    double legx1, legx2, legy1, legy2, legy2mini;


    //***** ASK THE STATION NUMBER *****

    cout<<"Give the station number (5, 6, 7, 8, 9, 10):  ";
    cin.clear(); //user's choise
    cin.get(station, 4);

    //first part of file name
    strncat( dataFirst,"/home/kulmalukko/PlankEff/ROOT/T", 35 );
    strncat( dataFirst, station, 2 );
    strncat( dataFirst, "/T", 2 );
    strncat( dataFirst, station, 2 );
    strncat( dataFirst, "-EffTime-", 9 );


    //***** READ STATION FILE *****

    //plank names are given in station file
    strncat( stationFile, "/home/kulmalukko/PlankEff/Code/", 35 );
    strncat( stationFile, "T", 2 );
    strncat( stationFile, station, 3 );
    strncat( stationFile, "-levels.dat", 15 );

    ifstream inStationFile;
    inStationFile.open(stationFile,ios::in);

    if( !inStationFile )
```

```
{
    cerr << " <E> Can't open File inStationFile " << inStationFile << endl;
    exit( -2 );
}

//ignore the first line
inStationFile.getline(temppi,500);

//explore the filelist by line the line
while( !inStationFile.eof() )
{
    inStationFile >> plankName >> itemp ;

    stationFileLines++;
    if( stationFileLines>4 ) //first 4 are pre-calibrated planks
    {
        cout<<plankName<<endl;
        CPLANKS.push_back(plankName);
    }

    }


//***** FILE LIST *****

strncat( fileList, "/home/kulmalukko/PlankEff/Code/T", 35 );
strncat( fileList, station, 3 );
strncat( fileList, "-Files.txt", 12 );

cout<<"File list: "<<fileList<<endl;

ifstream inFileList;
inFileList.open( fileList );


//calculate number of lines in datafile (filesMax == total number of lines)
while( getline(inFileList, lineTiedosto) )
{
    filesMax++;
}

inFileList.close();

//timeMax approximated duration of measurements (can't be smaller than real duration)
timeMax = filesMax*8*2;
cout<<"Number of files: "<<filesMax<<", approximated timeMax "<<timeMax<<"h"<<endl;


//***** MAKE FIGURES FOR EVERY PLANKS IN STATION *****

for(iplank = 0; iplank < stationFileLines - 4; iplank++)
{
    PL = CPLANKS[iplank];

    eff_sum=0;

    cout<<" ---- "<<PL<<" ---- "<<endl;

    //connect the parts of the plank's datafile name
    dataNimi = dataFirst + PL + dataEnd;

    effSum_X1 = 0, effSum_X2=0, effSum_X3=0, effSum_X4=0, effSum_Y1=0, effSum_Y2=0, effSum_Y3=0;
    //effSum_X1R = 0, effSum_X2R = 0, effSum_X2L = 0, effSum_X3R = 0, effSum_X3L = 0, effSum_X4L = 0;
    //effSum_Y1R = 0, effSum_Y1L = 0, effSum_Y2R = 0, effSum_Y2L = 0, effSum_Y3R = 0, effSum_Y3L = 0;
    effMean_X1 = 0, effMean_X2=0, effMean_X3=0, effMean_X4=0, effMean_Y1=0, effMean_Y2=0, effMean_Y3=0;

    // --- FORMAT effMap with "timeMax" approximation ---

    for( iChamPart = 0; iChamPart < 12; iChamPart++)
    {
        effMap[XXYYLR[iChamPart]].resize(timeMax, -6666);
    }


    // --- OPEN PLANK'S DATA FILE ---

    ifstream effTimeFile;
    effTimeFile.open(dataNimi.c_str(),ios::in);    //nimi.c_str() convert string into char

    //ignore the first line
    effTimeFile.getline(temppi,500);

    //these tables include the time value (hours)
    vector <int> timeTable_X1R, timeTable_X2L, timeTable_X2R, timeTable_X3L, timeTable_X3R, timeTable_X4L,
                 timeTable_Y1L, timeTable_Y1R, timeTable_Y2L, timeTable_Y2R, timeTable_Y3L, timeTable_Y3R, ;

    //explore lines of the efficiency time file
    while( !effTimeFile.eof() )
    {
        effTimeFile >> read_chamb >> read_eff >> read_time;

        eff_sum += read_eff;

        if(read_time<5){points=0;}

        //points == how many lines per chamber pair
        points++;
```

```
    if(strcmp ("X1R",read_chamb)==0){ timeTable_X1R.resize(points, read_time); }
    if(strcmp ("X2L",read_chamb)==0){ timeTable_X2L.resize(points, read_time); }
    if(strcmp ("X2R",read_chamb)==0){ timeTable_X2R.resize(points, read_time); }
    if(strcmp ("X3L",read_chamb)==0){ timeTable_X3L.resize(points, read_time); }
    if(strcmp ("X3R",read_chamb)==0){ timeTable_X3R.resize(points, read_time); }
    if(strcmp ("X4L",read_chamb)==0){ timeTable_X4L.resize(points, read_time); }

    if(strcmp ("Y1L",read_chamb)==0){ timeTable_Y1L.resize(points, read_time); }
    if(strcmp ("Y1R",read_chamb)==0){ timeTable_Y1R.resize(points, read_time); }
    if(strcmp ("Y2L",read_chamb)==0){ timeTable_Y2L.resize(points, read_time); }
    if(strcmp ("Y2R",read_chamb)==0){ timeTable_Y2R.resize(points, read_time); }
    if(strcmp ("Y3L",read_chamb)==0){ timeTable_Y3L.resize(points, read_time); }
    if(strcmp ("Y3R",read_chamb)==0){ timeTable_Y3R.resize(points, read_time); }

    effMap[read_chamb].at(read_time) = read_eff;

}

cout<<"Total number of data points: "<<points<<endl;

//close the file
effTimeFile.close();

//legend position in figure
legx1=0.68, legx2=0.84, legy1=0.14, legy2=0.29, legy2mini=0.21;


//= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

TCanvas *c_cham = new TCanvas("c_cham","Mean efficiency of the chambers",
                              0,0,1800,900);
TCanvas *c_half = new TCanvas("c_half","Efficiency of the left & right part of the chambers",
                              0,0,1800,900);

c_cham->Divide(2,1);
c_half->Divide(4,2);

c_cham->SetGrid();


// - - - CREATE MULTIGRAPHS - - -

//2 multigraphs: first one includes mean efficiensies of the x chambers and other one y chambers
TMultiGraph *mg_X = new TMultiGraph();
TMultiGraph *mg_Y = new TMultiGraph();

//7 multigraphs: each one include efficiency of righ and left part of the chamber
TMultiGraph *mg_X1 = new TMultiGraph();
TMultiGraph *mg_X2 = new TMultiGraph();
TMultiGraph *mg_X3 = new TMultiGraph();
TMultiGraph *mg_X4 = new TMultiGraph();
TMultiGraph *mg_Y1 = new TMultiGraph();
TMultiGraph *mg_Y2 = new TMultiGraph();
TMultiGraph *mg_Y3 = new TMultiGraph();

//n is number of points
const int n = points;
double axTime[n], axTime_X1R[points], axTime_X2L[points], axTime_X2R[points], axTime_X3L[points],
         axTime_X3R[points], axTime_X4L[points],
         axTime_Y1L[points], axTime_Y1R[points], axTime_Y2L[points], axTime_Y2R[points],
         axTime_Y3L[points], axTime_Y3R[points];

double  eff_X1[n], eff_X2[n], eff_X3[n], eff_X4[n], eff_Y1[n], eff_Y2[n], eff_Y3[n],
        eff_X1R[n], eff_X2R[n], eff_X2L[n], eff_X3R[n], eff_X3L[n], eff_X4L[n],
        eff_Y1R[n], eff_Y1L[n], eff_Y2R[n], eff_Y2L[n], eff_Y3R[n], eff_Y3L[n];

//explore through time indexies

for(iTime = 0; iTime < points; iTime++)
{
    //cout<<iTime<<" "<<timeTable_X1R[iTime+1]<<endl;
    //axTime[iTime] = timeTable[iTime];

    axTime_X1R[iTime] = timeTable_X1R[iTime];
    axTime_X2L[iTime] = timeTable_X2L[iTime];
    axTime_X2R[iTime] = timeTable_X2R[iTime];
    axTime_X3L[iTime] = timeTable_X3L[iTime];
    axTime_X3R[iTime] = timeTable_X3R[iTime];
    axTime_X4L[iTime] = timeTable_X4L[iTime];

    axTime_Y1L[iTime] = timeTable_Y1L[iTime];
    axTime_Y1R[iTime] = timeTable_Y1R[iTime];
    axTime_Y2L[iTime] = timeTable_Y2L[iTime];
    axTime_Y2R[iTime] = timeTable_Y2R[iTime];
    axTime_Y3L[iTime] = timeTable_Y3L[iTime];
    axTime_Y3R[iTime] = timeTable_Y3R[iTime];

    //efficiensies of the chamber parts
    eff_X1R[iTime] = effMap["X1R"][axTime_X1R[iTime]]*100;
    eff_X2L[iTime] = effMap["X2L"][axTime_X2L[iTime]]*100;
    eff_X2R[iTime] = effMap["X2R"][axTime_X2R[iTime]]*100;
    eff_X3L[iTime] = effMap["X3L"][axTime_X3L[iTime]]*100;
    eff_X3R[iTime] = effMap["X3R"][axTime_X3R[iTime]]*100;
    eff_X4L[iTime] = effMap["X4L"][axTime_X4L[iTime]]*100;

    eff_Y1L[iTime] = effMap["Y1L"][axTime_Y1L[iTime]]*100;
    eff_Y1R[iTime] = effMap["Y1R"][axTime_Y1R[iTime]]*100;
```

```
    eff_Y2L[iTime] = effMap["Y2L"][axTime_Y2L[iTime]]*100;
    eff_Y2R[iTime] = effMap["Y2R"][axTime_Y2R[iTime]]*100;
    eff_Y3L[iTime] = effMap["Y3L"][axTime_Y3L[iTime]]*100;
    eff_Y3R[iTime] = effMap["Y3R"][axTime_Y3R[iTime]]*100;

    //mean efficiensies of the chambers
    eff_X1[iTime] = eff_X1R[iTime];
    eff_X2[iTime] = ( eff_X2R[iTime] + eff_X2L[iTime] ) / 2.0;
    eff_X3[iTime] = ( eff_X3R[iTime] + eff_X3L[iTime] ) / (double)2;
    eff_X4[iTime] = eff_X4L[iTime];

    eff_Y1[iTime] = ( eff_Y1R[iTime] + eff_Y1L[iTime] ) / 2.0;
    eff_Y2[iTime] = ( eff_Y2R[iTime] + eff_Y2L[iTime] ) / 2.0;
    eff_Y3[iTime] = ( eff_Y3R[iTime] + eff_Y3L[iTime] ) / 2.0;

    effSum_X1 += eff_X1[iTime];
    effSum_X2 += eff_X2[iTime];
    effSum_X3 += eff_X3[iTime];
    effSum_X4 += eff_X4[iTime];
    effSum_Y1 += eff_Y1[iTime];
    effSum_Y2 += eff_Y2[iTime];
    effSum_Y3 += eff_Y3[iTime];

    /*
    effSum_X1R += eff_X1R[iTime];
    effSum_X2R += eff_X2R[iTime];
    effSum_X2L += eff_X2L[iTime];
    effSum_X3R += eff_X3R[iTime];
    effSum_X3L += eff_X3L[iTime];
    effSum_X4L += eff_X4L[iTime];

    effSum_Y1R += eff_Y1R[iTime];
    effSum_Y1L += eff_Y1L[iTime];
    effSum_Y2R += eff_Y2R[iTime];
    effSum_Y2L += eff_Y2L[iTime];
    effSum_Y3R += eff_Y3R[iTime];
    effSum_Y3L += eff_Y3L[iTime];
    */

}

//These gives mean eff of chamber to labels
effMean_X1 = effSum_X1/ (double)points;
effMean_X2 = effSum_X2/ (double)points;
effMean_X3 = effSum_X3/ (double)points;
effMean_X4 = effSum_X4/ (double)points;
effMean_Y1 = effSum_Y1/ (double)points;
effMean_Y2 = effSum_Y2/ (double)points;
effMean_Y3 = effSum_Y3/ (double)points;

//cout<<effSum_Y3<<" "<<points<<" "<<effMean_Y3<<endl;

//Mean eff of plank is mean eff of all the effs in file
//(not the same as (effMean_X1 + effMean_X2+..)/7) <- weighting is missing
//cout<<"Sum of all eff's * 100 / (12 * points) = "<<eff_sum*100<<" / 12*"<<points<<endl;
cout<<"-> Plank's eff = "<<eff_sum*100/((double)points*12)<<endl;

//cout <<effSum_X3<<" vs. " <<effSum_X3L<<" "<<effSum_X3R<<endl;

//"wrong" way to calculate the ave. eff of whole plank
//The right way is to calculate the ave. of all chamber parts
//cout<<"Mean eff of plank: "<<(effMean_X1+effMean_X2+effMean_X3+effMean_X4+effMean_Y1+effMean_Y2+effMean_Y3)/7.0<<en


// --- CREATE GRAPHS ---

//These graphs include efficiency of chamber
TGraph *gr_X1 = new TGraph(n, axTime_X1R, eff_X1);
TGraph *gr_X2 = new TGraph(n, axTime_X2R, eff_X2);
TGraph *gr_X3 = new TGraph(n, axTime_X3R, eff_X3);
TGraph *gr_X4 = new TGraph(n, axTime_X4L, eff_X4);

TGraph *gr_Y1 = new TGraph(n, axTime_Y1R, eff_Y1);
TGraph *gr_Y2 = new TGraph(n, axTime_Y2R, eff_Y2);
TGraph *gr_Y3 = new TGraph(n, axTime_Y3R, eff_Y3);

//These graphs include efficiency of chamber part
TGraph *gr_X1R = new TGraph(n, axTime_X1R, eff_X1R);
TGraph *gr_X2R = new TGraph(n, axTime_X2R, eff_X2R);
TGraph *gr_X2L = new TGraph(n, axTime_X2L, eff_X2L);
TGraph *gr_X3R = new TGraph(n, axTime_X3R, eff_X3R);
TGraph *gr_X3L = new TGraph(n, axTime_X3L, eff_X3L);
TGraph *gr_X4L = new TGraph(n, axTime_X4L, eff_X4L);

TGraph *gr_Y1R = new TGraph(n, axTime_Y1R, eff_Y1R);
TGraph *gr_Y1L = new TGraph(n, axTime_Y1L, eff_Y1L);
TGraph *gr_Y2R = new TGraph(n, axTime_Y2R, eff_Y2R);
TGraph *gr_Y2L = new TGraph(n, axTime_Y2L, eff_Y2L);
TGraph *gr_Y3R = new TGraph(n, axTime_Y3R, eff_Y3R);
TGraph *gr_Y3L = new TGraph(n, axTime_Y3L, eff_Y3L);


// --- LINE & MARKER SETTINGS ---

//Marker settings of graphs
double markerSize1 = 0.2, markerSize2 = 0.1;
```

```
int markerStyle = 21;

//chambers
gr_X1->SetLineColor(2); //red
gr_X1->SetMarkerSize(markerSize1);
gr_X1->SetMarkerStyle(markerStyle);

gr_X2->SetLineColor(4); //blue
gr_X2->SetMarkerSize(markerSize1);
gr_X2->SetMarkerStyle(markerStyle);

gr_X3->SetLineColor(3); //green
gr_X3->SetMarkerSize(markerSize1);
gr_X3->SetMarkerStyle(markerStyle);

gr_X4->SetLineColor(5); //yellow
gr_X4->SetMarkerSize(markerSize1);
gr_X4->SetMarkerStyle(markerStyle);

gr_Y1->SetLineColor(2); //red
gr_Y1->SetMarkerSize(markerSize1);
gr_Y1->SetMarkerStyle(markerStyle);

gr_Y2->SetLineColor(4); //blue
gr_Y2->SetMarkerSize(markerSize1);
gr_Y2->SetMarkerStyle(markerStyle);

gr_Y3->SetLineColor(3); //green
gr_Y3->SetMarkerSize(markerSize1);
gr_Y3->SetMarkerStyle(markerStyle);


//x chamber parts
gr_X1R->SetLineColor(2);     //red
gr_X1R->SetMarkerSize(markerSize2);
gr_X1R->SetMarkerStyle(markerStyle);

gr_X2R->SetLineColor(2);     //red
gr_X2R->SetMarkerSize(markerSize2);
gr_X2R->SetMarkerStyle(markerStyle);
gr_X2L->SetLineColor(4);     //blue
gr_X2L->SetMarkerSize(markerSize2);
gr_X2L->SetMarkerStyle(markerStyle);

gr_X3R->SetLineColor(2);     //red
gr_X3R->SetMarkerSize(markerSize2);
gr_X3R->SetMarkerStyle(markerStyle);
gr_X3L->SetLineColor(4);     //blue
gr_X3L->SetMarkerSize(markerSize2);
gr_X3L->SetMarkerStyle(markerStyle);

gr_X4L->SetLineColor(4);     //blue
gr_X4L->SetMarkerSize(markerSize2);
gr_X4L->SetMarkerStyle(markerStyle);

//y chamber parts
gr_Y1R->SetLineColor(2);     //pun
gr_Y1R->SetMarkerSize(markerSize2);
gr_Y1R->SetMarkerStyle(markerStyle);
gr_Y1L->SetLineColor(4);     //sin
gr_Y1L->SetMarkerSize(markerSize2);
gr_Y1L->SetMarkerStyle(markerStyle);

gr_Y2R->SetLineColor(2);     //pun
gr_Y2R->SetMarkerSize(markerSize2);
gr_Y2R->SetMarkerStyle(markerStyle);
gr_Y2L->SetLineColor(4);     //sin
gr_Y2L->SetMarkerSize(markerSize2);
gr_Y2L->SetMarkerStyle(markerStyle);

gr_Y3R->SetLineColor(2);     //pun
gr_Y3R->SetMarkerSize(markerSize2);
gr_Y3R->SetMarkerStyle(markerStyle);
gr_Y3L->SetLineColor(4);     //sin
gr_Y3L->SetMarkerSize(markerSize2);
gr_Y3L->SetMarkerStyle(markerStyle);


// _____________ CANVAS c_cham _____________

double titleOff = 1.3;

// these are for evaluating the mean efficiency per chamber
char x1teksti [100], x2teksti [100], x3teksti [100], x4teksti [100], y1teksti [100], y2teksti [100], y3teksti [100];
snprintf (x1teksti, 100, "X1 (%.2f)", effMean_X1);
snprintf (x2teksti, 100, "X2 (%.2f)", effMean_X2);
snprintf (x3teksti, 100, "X3 (%.2f)", effMean_X3);
snprintf (x4teksti, 100, "X4 (%.2f)", effMean_X4);
snprintf (y1teksti, 100, "Y1 (%.2f)", effMean_Y1);
snprintf (y2teksti, 100, "Y2 (%.2f)", effMean_Y2);
snprintf (y3teksti, 100, "Y3 (%.2f)", effMean_Y3);

// - - - Draw mg_Y (Y1, Y2, Y3)

c_cham->cd(1); mg_Y->Add(gr_Y1);
c_cham->cd(1); mg_Y->Add(gr_Y2);
```

```
c_cham->cd(1); mg_Y->Add(gr_Y3);

mg_Y->SetTitle("Chambers Y1, Y2 & Y3");
mg_Y->Draw("ALP");

legY = new TLegend(0.60,0.12,0.85,0.27);
legY->AddEntry(gr_Y1, y1teksti, "l");
legY->AddEntry(gr_Y2, y2teksti, "l");
legY->AddEntry(gr_Y3, y3teksti, "l");
legY->Draw();

mg_Y->GetXaxis()->SetTitle("time (h)");
mg_Y->GetXaxis()->SetLimits(0,timeTable_X1R[n-1] );
mg_Y->GetYaxis()->SetTitle("efficiency (%)");
mg_Y->GetYaxis()->SetTitleOffset(titleOff);
mg_Y->SetMaximum(100);
mg_Y->SetMinimum(0);


// - - - Draw mg_X (X1, X2, X3, X4)

c_cham->cd(2); mg_X->Add(gr_X1);
c_cham->cd(2); mg_X->Add(gr_X2);
c_cham->cd(2); mg_X->Add(gr_X3);
c_cham->cd(2); mg_X->Add(gr_X4);

mg_X->SetTitle("Chambers X1, X2, X3 & X4");
mg_X->Draw("ALP");

legX = new TLegend(0.64,0.12,0.85,0.32);
legX->AddEntry(gr_X1, x1teksti, "l");
legX->AddEntry(gr_X2, x2teksti, "l");
legX->AddEntry(gr_X3, x3teksti, "l");
legX->AddEntry(gr_X4, x4teksti, "l");
legX->Draw();

mg_X->GetXaxis()->SetTitle("time (h)");
mg_X->GetXaxis()->SetLimits(0,timeTable_X1R[n-1] );
mg_X->GetYaxis()->SetTitle("efficiency (%)");
mg_X->GetYaxis()->SetTitleOffset(titleOff);
mg_X->SetMaximum(100);
mg_X->SetMinimum(0);


// _____________ CANVAS c_half _____________

// - - - Draw mg_Y1 (Y1R & Y1L)

c_half->cd(1); mg_Y1->Add(gr_Y1R);
c_half->cd(1); mg_Y1->Add(gr_Y1L);

mg_Y1->SetTitle("Chambers Y1R & Y1L");
mg_Y1->Draw("ALP");

legY1 = new TLegend(legx1, legy1, legx2, legy2);
legY1->AddEntry(gr_Y1R, "Y1R", "l");
legY1->AddEntry(gr_Y1L, "Y1L", "l");
legY1->Draw();

mg_Y1->GetXaxis()->SetTitle("time (h)");
mg_Y1->GetXaxis()->SetLimits(0,timeTable_X1R[n-1] );
mg_Y1->GetYaxis()->SetTitle("efficiency (%)");
mg_Y1->GetYaxis()->SetTitleOffset(titleOff);
mg_Y1->SetMaximum(100);
mg_Y1->SetMinimum(0);


// - - - Draw mg_Y2 (Y2R & Y2L)

c_half->cd(2); mg_Y2->Add(gr_Y2R);
c_half->cd(2); mg_Y2->Add(gr_Y2L);

mg_Y2->SetTitle("Chambers Y2R & Y2L");
mg_Y2->Draw("ALP");

legY2 = new TLegend(legx1, legy1, legx2, legy2);
legY2->AddEntry(gr_Y2R, "Y2R", "l");
legY2->AddEntry(gr_Y2L, "Y2L", "l");
legY2->Draw();

mg_Y2->GetXaxis()->SetTitle("time (h)");
mg_Y2->GetXaxis()->SetLimits(0,timeTable_X1R[n-1] );
mg_Y2->GetYaxis()->SetTitle("efficiency (%)");
mg_Y2->GetYaxis()->SetTitleOffset(titleOff);
mg_Y2->SetMaximum(100);
mg_Y2->SetMinimum(0);


// - - - Draw mg_Y3 (Y3R & Y3L)

c_half->cd(3); mg_Y3->Add(gr_Y3R);
c_half->cd(3); mg_Y3->Add(gr_Y3L);

mg_Y3->SetTitle("Chambers Y3R & Y3L");
mg_Y3->Draw("ALP");

legY3 = new TLegend(legx1, legy1, legx2, legy2);
legY3->AddEntry(gr_Y3R, "Y3R", "l");
```

```
legY3->AddEntry(gr_Y3L, "Y3L", "l");
legY3->Draw();

mg_Y3->GetXaxis()->SetTitle("time (h)");
mg_Y3->GetXaxis()->SetLimits(0,timeTable_X1R[n-1] );
mg_Y3->GetYaxis()->SetTitle("efficiency (%)");
mg_Y3->GetYaxis()->SetTitleOffset(titleOff);
mg_Y3->SetMaximum(100);
mg_Y3->SetMinimum(0);


// --- Draw mg_X1 (X1R)

c_half->cd(5); mg_X1->Add(gr_X1R);

mg_X1->SetTitle("Chamber X1R");
mg_X1->Draw("ALP");

legX1 = new TLegend(legx1, legy1, legx2, legy2mini);
legX1->AddEntry(gr_X1R, "X1R", "l");
legX1->Draw();

mg_X1->GetXaxis()->SetTitle("time (h)");
mg_X1->GetXaxis()->SetLimits(0,timeTable_X1R[n-1] );
mg_X1->GetYaxis()->SetTitle("efficiency (%)");
mg_X1->GetYaxis()->SetTitleOffset(titleOff);
mg_X1->SetMaximum(100);
mg_X1->SetMinimum(0);


// --- Draw mg_X2 (X2R & X2L)

c_half->cd(6); mg_X2->Add(gr_X2R);
c_half->cd(6); mg_X2->Add(gr_X2L);

mg_X2->SetTitle("Chambers X2R & X2L");
mg_X2->Draw("ALP");

legX2 = new TLegend(legx1, legy1, legx2, legy2);
legX2->AddEntry(gr_X2R, "X2R", "l");
legX2->AddEntry(gr_X2L, "X2L", "l");
legX2->Draw();

mg_X2->GetXaxis()->SetTitle("time (h)");
mg_X2->GetXaxis()->SetLimits(0,timeTable_X1R[n-1] );
mg_X2->GetYaxis()->SetTitle("efficiency (%)");
mg_X2->GetYaxis()->SetTitleOffset(titleOff);
mg_X2->SetMaximum(100);
mg_X2->SetMinimum(0);


// --- Draw mg_X3 (X3R & X3L)

c_half->cd(7); mg_X3->Add(gr_X3R);
c_half->cd(7); mg_X3->Add(gr_X3L);

mg_X3->SetTitle("Chambers X3R & X3L");
mg_X3->Draw("ALP");

legX3 = new TLegend(legx1, legy1, legx2, legy2);
legX3->AddEntry(gr_X3R, "X3R", "l");
legX3->AddEntry(gr_X3L, "X3L", "l");
legX3->Draw();

mg_X3->GetXaxis()->SetTitle("time (h)");
mg_X3->GetXaxis()->SetLimits(0,timeTable_X1R[n-1] );
mg_X3->GetYaxis()->SetTitle("efficiency (%)");
mg_X3->GetYaxis()->SetTitleOffset(titleOff);
mg_X3->SetMaximum(100);
mg_X3->SetMinimum(0);


// --- Draw mg_X4 (X4L)

c_half->cd(8); mg_X4->Add(gr_X4L);

mg_X4->SetTitle("Chamber X4L");
mg_X4->Draw("ALP");

legX4 = new TLegend(legx1, legy1, legx2, legy2mini);
legX4->AddEntry(gr_X4L, "X4L", "l");
legX4->Draw();

mg_X4->GetXaxis()->SetTitle("time (h)");
mg_X4->GetXaxis()->SetLimits(0,timeTable_X1R[n-1] );
mg_X4->GetYaxis()->SetTitle("efficiency (%)");
mg_X4->GetYaxis()->SetTitleOffset(titleOff);
mg_X4->SetMaximum(100);
mg_X4->SetMinimum(0);


// --- SAVE THE FIGURES ---

string figNameStart="ROOT-ET-T";
string figNameEnd=".pdf";

string figName=figNameStart+station+"-"+PL+figNameEnd;
string figName2=figNameStart+station+"-"+PL+"-2"+figNameEnd;
```

```
        c_cham->SaveAs(figName.c_str());
        c_half->SaveAs(figName2.c_str());

        //= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

        // - - - DELETE THINGS - - -

        //delete tuhoo muistin, clear tuhoo indexin

        delete c_cham;
        delete c_half;

        //When deleting multigraphs, also TGraphs are deleted automaticly
        delete mg_X;
        delete mg_Y;

        delete mg_X1;
        delete mg_X2;
        delete mg_X3;
        delete mg_X4;
        delete mg_Y1;
        delete mg_Y2;
        delete mg_Y3;

        delete legX;
        delete legY;
        delete legX1;
        delete legX2;
        delete legX3;
        delete legX4;
        delete legY1;
        delete legY2;
        delete legY3;

        //delete effMap["X1R"];
    }

    return 0;
}
```

A.3 Eff-Position.c

```
// Start ROOT and run the code giving the command .x Eff-Position.c
// Programs ask the station number (5, 6, 7, 8, 9, 10)
// Program draws the figures efficiency-position for the planksa of a given station
//
// ROOT must be turned off before new running
//
{
    gROOT->Reset();

    //***** VARIABLES *****

    char station[4] = "", stationFile[55] = "";

    //lines in station file
    int stationFileLines = 0;

    char temppi[500];
    double itemp;

    //parts of data file name
    char dataFirst[56] = "";
    string dataEnd = ".gd";

    char plankName[4];
    vector<string> CPLANKS; //plank name table
    string PL;

    //these are used as for inex
    int iplank, iPos;

    double itemp, kanava;
    string position;
    double eff_X1, eff_X2, eff_X3, eff_X4, eff_Y1, eff_Y2, eff_Y3;

    //size of posVec
    int n;

    //***** ASK THE STATION NUMBER *****

    cout<<"Give the station number (5, 6, 7, 8, 9, 10):  ";
    cin.clear(); //user's choise
    cin.get(station, 4);

    //first part of file name
    strncat( dataFirst,"/home/kulmalukko/PlankEff/ROOT/T", 35 );
    strncat( dataFirst, station, 4 );
    strncat( dataFirst, "/T", 3 );
    strncat( dataFirst, station, 4 );
    strncat( dataFirst, "-EffPos-", 9 );


    //***** HANDLE STATION FILE *****

    //plank names are given in station file
    strncat( stationFile, "/home/kulmalukko/PlankEff/Code/", 35 );
    strncat( stationFile, "T", 2 );
    strncat( stationFile, station, 4 );
    strncat( stationFile, "-levels.dat", 12 );

    ifstream inStationFile;
    inStationFile.open(stationFile,ios::in);

    if( !inStationFile )
    {
        cerr << " <E> Can't open File inStationFile " << inStationFile << endl;
        exit( -2 );
    }

    //ignore the first line
    inStationFile.getline(temppi,500);


    //***** READ FILE *****

    while( !inStationFile.eof() )
    {
        inStationFile >> plankName >> itemp ;

        stationFileLines++;
        if( stationFileLines > 4 ) //first 4 are pre-calibrated planks
        {
            cout<<plankName<<endl;
            CPLANKS.push_back(plankName);
        }

        }


    //***** HANDLE PLANKS *****

    for( iplank = 0; iplank < stationFileLines - 4; iplank++ )
    {
        PL = CPLANKS[iplank];

        cout<<" -- "<<PL<<" -- "<<endl;
```

```
//connect the parts of the plank's datafile name
string dataNimi = dataFirst + PL + dataEnd;

cout<<" "<<dataNimi<<endl;

//create canvas
TCanvas *can = new TCanvas("can", PL.c_str(), 0, 0, 3650, 1825);
can->Divide(2,1);
can->SetGrid();


// --- CREATE MULTIGRAPHS ---
TMultiGraph *mg_X = new TMultiGraph();
TMultiGraph *mg_Y = new TMultiGraph();

//open the file
ifstream effPosFile;
effPosFile.open(dataNimi.c_str(),ios::in);    //nimi.c_str() convert string into char

//ignore the first line
effPosFile.getline(temppi,500);

//--- CREATE VECTORS ---
vector <double> posVec, effVec_X1, effVec_X2, effVec_X3, effVec_X4, effVec_Y1, effVec_Y2, effVec_Y3;
int lines = 0;


// --- READ FILES ---

while( !effPosFile.eof() )
{

    effPosFile >> kanava >> position >> eff_X1 >> eff_Y1 >> eff_X2 >> eff_Y2
    >> eff_X3 >> eff_Y3 >> eff_X4;

    //if( effPosFile.eof() ){ cout<<"eof "; }

    lines++;

    //cout<<kanava<<"   "<<((kanava*4)+2)<<"    "<<eff_X1<<"    "
    //<<eff_X2<<"    "<<eff_X3<<"    "<<eff_X4<<"    "
    //<<eff_Y1<<"    "<<eff_Y2<<"    "<<eff_Y3<<endl;


    // --- FILL VECTORS ---

    posVec.resize( lines, ((kanava*4)+2) );

    effVec_X1.resize( lines, eff_X1 );
    effVec_X2.resize( lines, eff_X2 );
    effVec_X3.resize( lines, eff_X3 );
    effVec_X4.resize( lines, eff_X4 );
    effVec_Y1.resize( lines, eff_Y1 );
    effVec_Y2.resize( lines, eff_Y2 );
    effVec_Y3.resize( lines, eff_Y3 );

    //cout<<lines<<" "<<kanava;

    //cout<<endl;
}

effPosFile.close();

n = posVec.size();


// --- CREATE TABLES ---

double posTab[n];
double effTab_X1[n], effTab_X2[n], effTab_X3[n],
       effTab_X4[n], effTab_Y1[n], effTab_Y2[n], effTab_Y3[n];


// --- FILL THE TABLES ---

for( iPos = 0; iPos < posVec.size() ; iPos++ )
{
    //cout<<i<<" "<<posVec[i]<<" "<<effVec_X1[i]<<" "<<effVec_Y2[i]<<" "<<effVec_Y3[i]<<endl;

    posTab[iPos] = posVec[iPos];

    effTab_X1[iPos] = effVec_X1[iPos]*100;
    effTab_X2[iPos] = effVec_X2[iPos]*100;
    effTab_X3[iPos] = effVec_X3[iPos]*100;
    effTab_X4[iPos] = effVec_X4[iPos]*100;
    effTab_Y1[iPos] = effVec_Y1[iPos]*100;
    effTab_Y2[iPos] = effVec_Y2[iPos]*100;
    effTab_Y3[iPos] = effVec_Y3[iPos]*100;

}


// --- CREATE GRAPHS ---

//These graphs include efficiency of chamber
TGraph *gr_X1 = new TGraph( n, posTab, effTab_X1 );
TGraph *gr_X2 = new TGraph( n, posTab, effTab_X2 );
TGraph *gr_X3 = new TGraph( n, posTab, effTab_X3 );
```

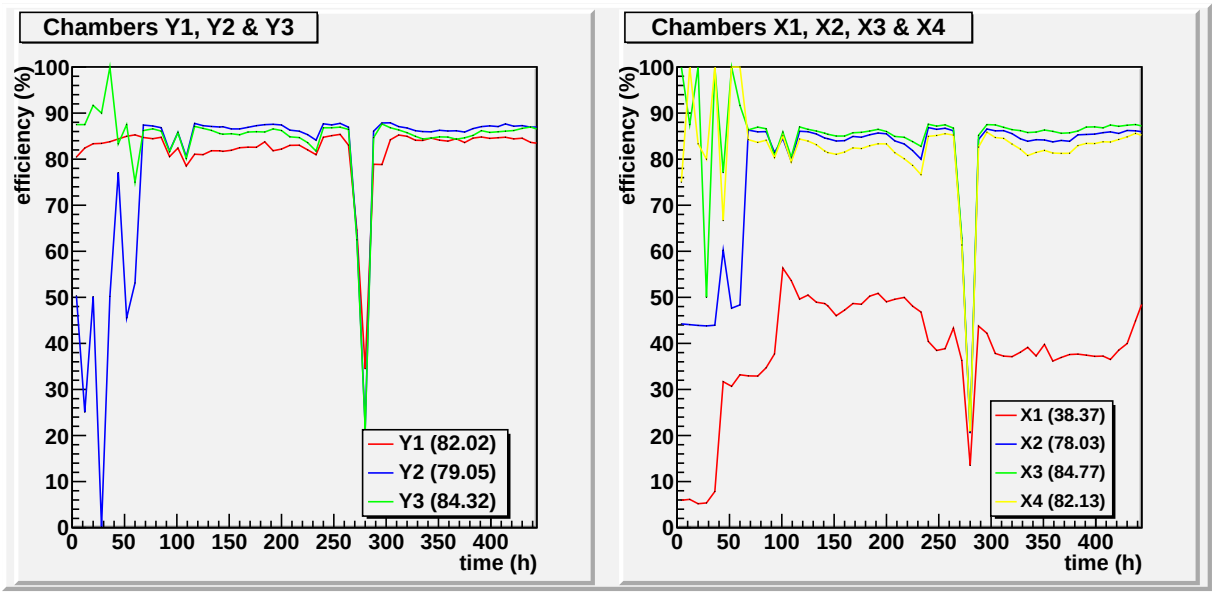
```
TGraph *gr_X4 = new TGraph( n, posTab, effTab_X4 );

TGraph *gr_Y1 = new TGraph( n, posTab, effTab_Y1 );
TGraph *gr_Y2 = new TGraph( n, posTab, effTab_Y2 );
TGraph *gr_Y3 = new TGraph( n, posTab, effTab_Y3 );


// --- LINE & MARKER SETTINGS ---

//Marker settings of graphs
double markerSize1 = 0.1, markerSize2 = 0.1, lineWidth=0.2;
int markerStyle = 21;

//chambers
gr_X1->SetLineColor(2); //red
gr_X1->SetLineWidth(lineWidth);
gr_X1->SetMarkerSize(markerSize1);
gr_X1->SetMarkerStyle(markerStyle);

gr_X2->SetLineColor(4); //blue
gr_X2->SetLineWidth(lineWidth);
gr_X2->SetMarkerSize(markerSize1);
gr_X2->SetMarkerStyle(markerStyle);

gr_X3->SetLineColor(3); //green
gr_X3->SetLineWidth(lineWidth);
gr_X3->SetMarkerSize(markerSize1);
gr_X3->SetMarkerStyle(markerStyle);

gr_X4->SetLineColor(5); //yellow
gr_X4->SetLineWidth(lineWidth);
gr_X4->SetMarkerSize(markerSize1);
gr_X4->SetMarkerStyle(markerStyle);

gr_Y1->SetLineColor(2); //red
gr_Y1->SetLineWidth(lineWidth);
gr_Y1->SetMarkerSize(markerSize1);
gr_Y1->SetMarkerStyle(markerStyle);

gr_Y2->SetLineColor(4); //blue
gr_Y2->SetLineWidth(lineWidth);
gr_Y2->SetMarkerSize(markerSize1);
gr_Y2->SetMarkerStyle(markerStyle);

gr_Y3->SetLineColor(3); //green
gr_Y3->SetLineWidth(lineWidth);
gr_Y3->SetMarkerSize(markerSize1);
gr_Y3->SetMarkerStyle(markerStyle);


double titleOff = 1.3;

//CANVAS CD(1)
can->cd(1); mg_Y->Add(gr_Y1);
can->cd(1); mg_Y->Add(gr_Y2);
can->cd(1); mg_Y->Add(gr_Y3);

mg_Y->SetTitle("Chambers Y1, Y2 & Y3");
mg_Y->Draw("ALP");

TLegend *legY = new TLegend(0.74,0.12,0.82,0.27);
legY->AddEntry(gr_Y1, "Y1", "l");
legY->AddEntry(gr_Y2, "Y2", "l");
legY->AddEntry(gr_Y3, "Y3", "l");
legY->Draw();

mg_Y->GetXaxis()->SetTitle("position (mm)");
mg_Y->GetXaxis()->SetLimits(0,posTab[n-1] );
mg_Y->GetYaxis()->SetTitle("efficiency (%)");
mg_Y->GetYaxis()->SetTitleOffset(titleOff);
mg_Y->SetMaximum(100);
mg_Y->SetMinimum(0);


//CANVAS CD(2)
can->cd(2); mg_X->Add(gr_X1);
can->cd(2); mg_X->Add(gr_X2);
can->cd(2); mg_X->Add(gr_X3);
can->cd(2); mg_X->Add(gr_X4);

mg_X->SetTitle("Chambers X1, X2, X3 & X4");
mg_X->Draw("ALP");

TLegend *legX = new TLegend(0.74,0.12,0.82,0.32);
legX->AddEntry(gr_X1, "X1", "l");
legX->AddEntry(gr_X2, "X2", "l");
legX->AddEntry(gr_X3, "X3", "l");
legX->AddEntry(gr_X4, "X4", "l");
legX->Draw();

mg_X->GetXaxis()->SetTitle("position (mm)");
mg_X->GetXaxis()->SetLimits(0,posTab[n-1] );
mg_X->GetYaxis()->SetTitle("efficiency (%)");
mg_X->GetYaxis()->SetTitleOffset(titleOff);
mg_X->SetMaximum(100);
mg_X->SetMinimum(0);
```

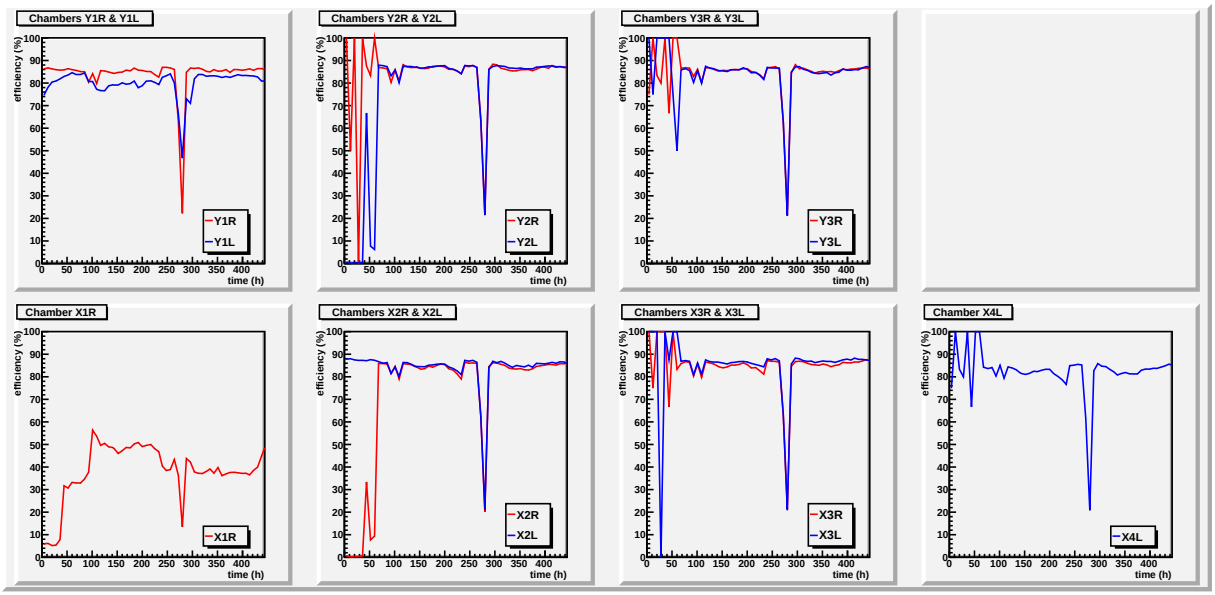
```
        // --- SAVE THE FIGURES ---

        string figNameStart="ROOT-EP-T";
        string figNameEnd=".pdf";

        string figName=figNameStart+station+"-"+PL+figNameEnd;

        can->SaveAs(figName.c_str());


        // --- REMOVE THINGS ---

        delete can;

        //when delete mg, also gr are deleted automaticly
        delete mg_X;
        delete mg_Y;

        delete legX;
        delete legY;

    }

    return 0;
}
```

B Efficiency vs. time figures

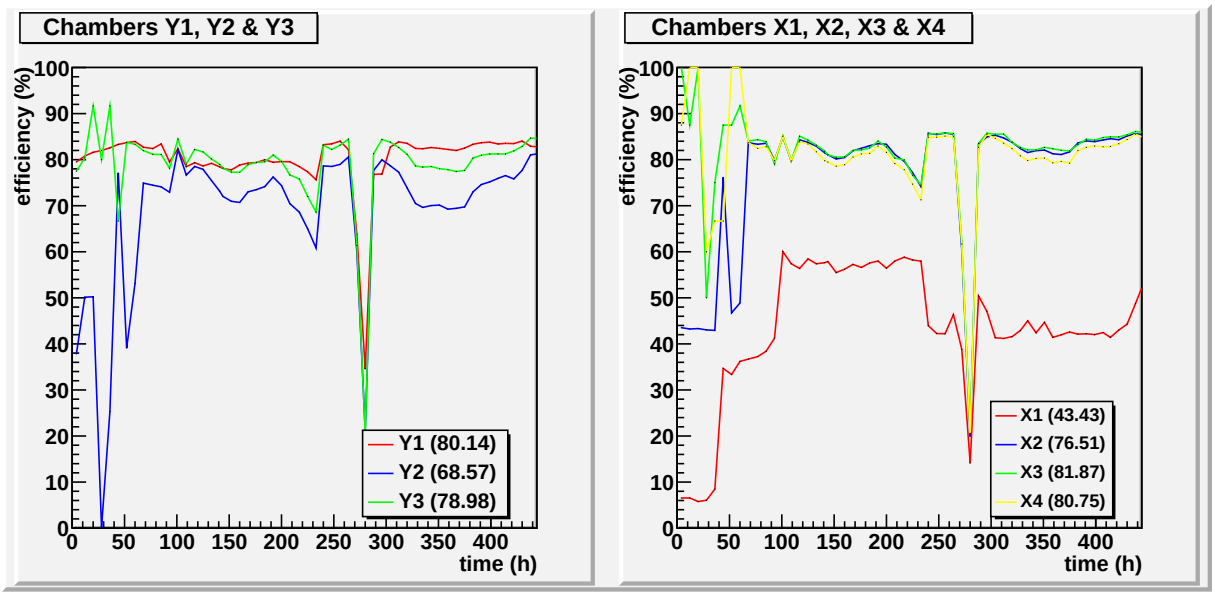
B.1 Efficiency vs. time of stack 5 (started 4 Septemper, 2009)

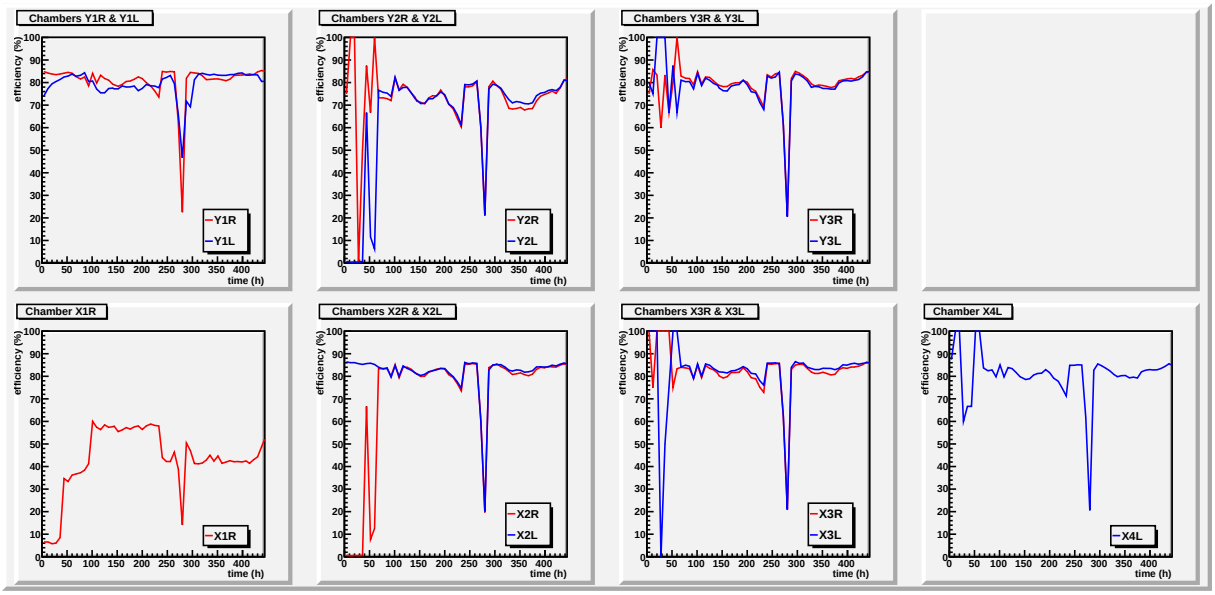
Stack 5, plank 34



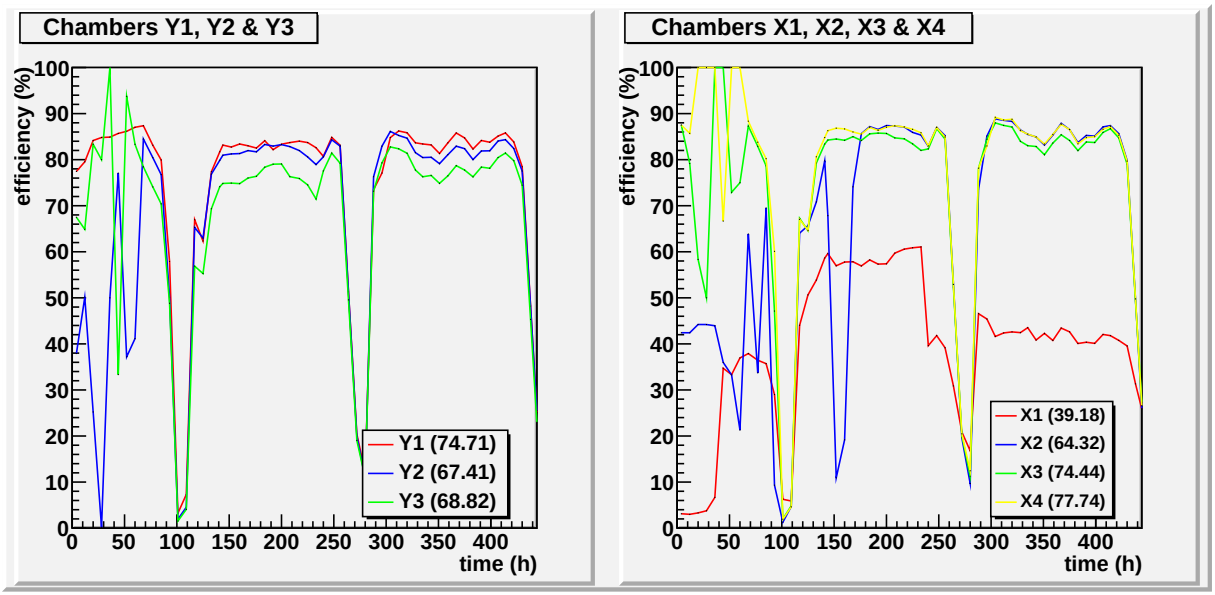


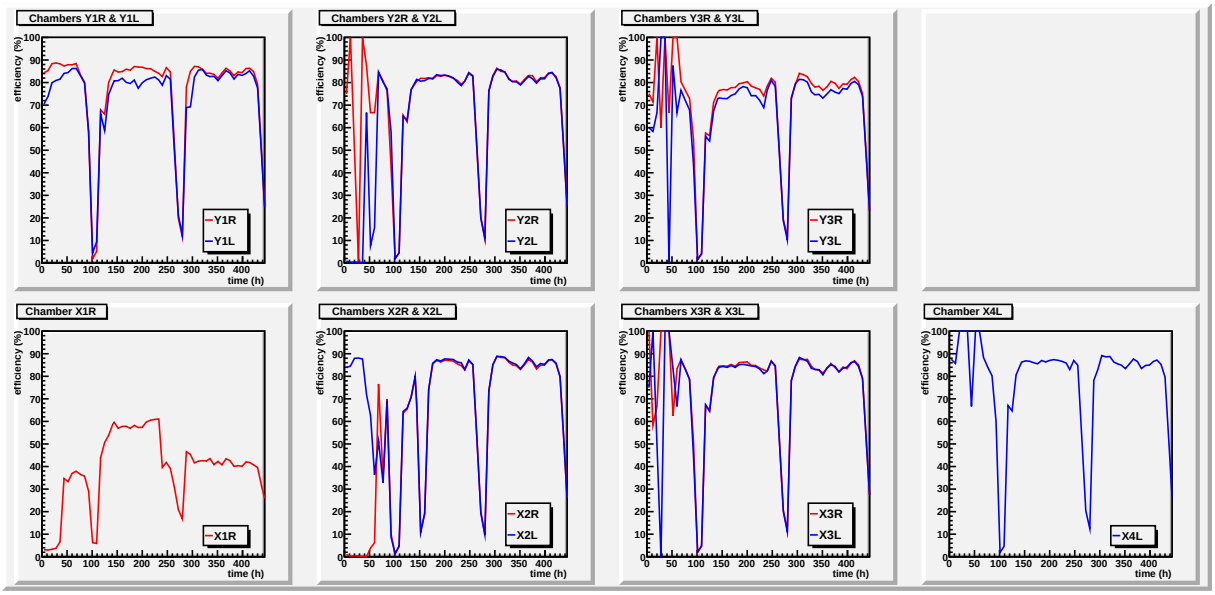
Stack 5, plank 35



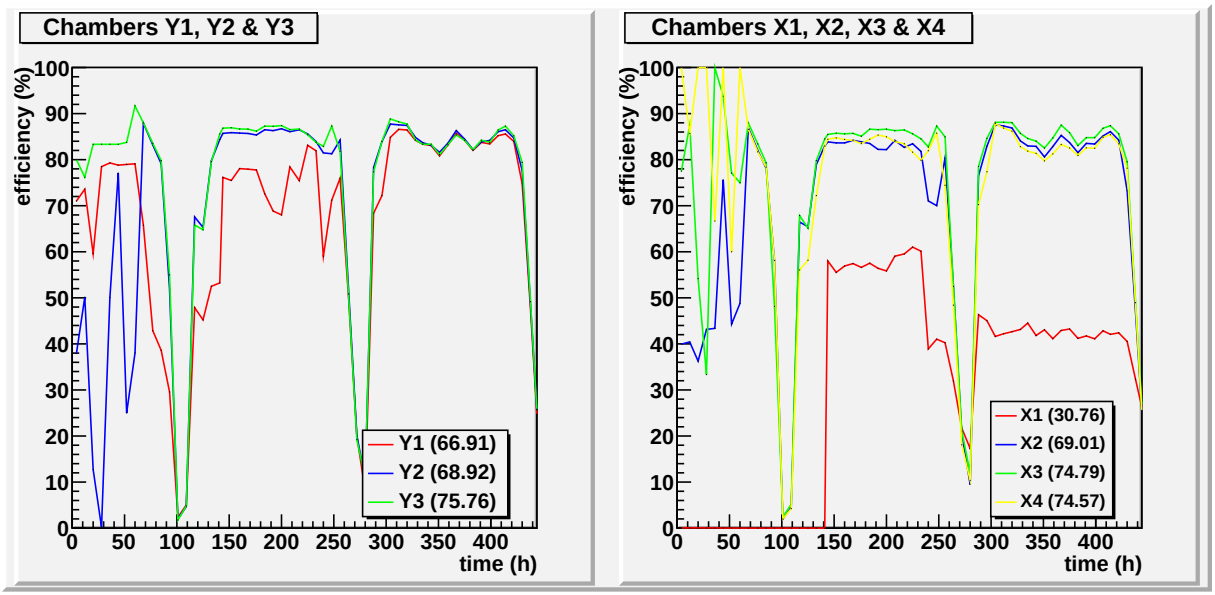


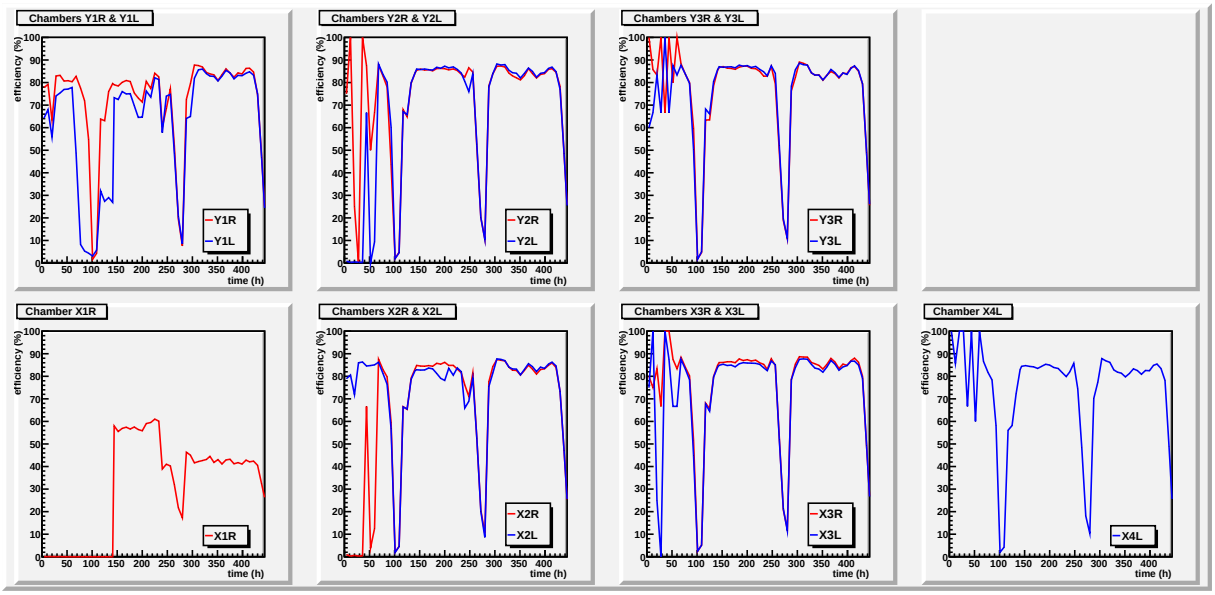
Stack 5, plank 38





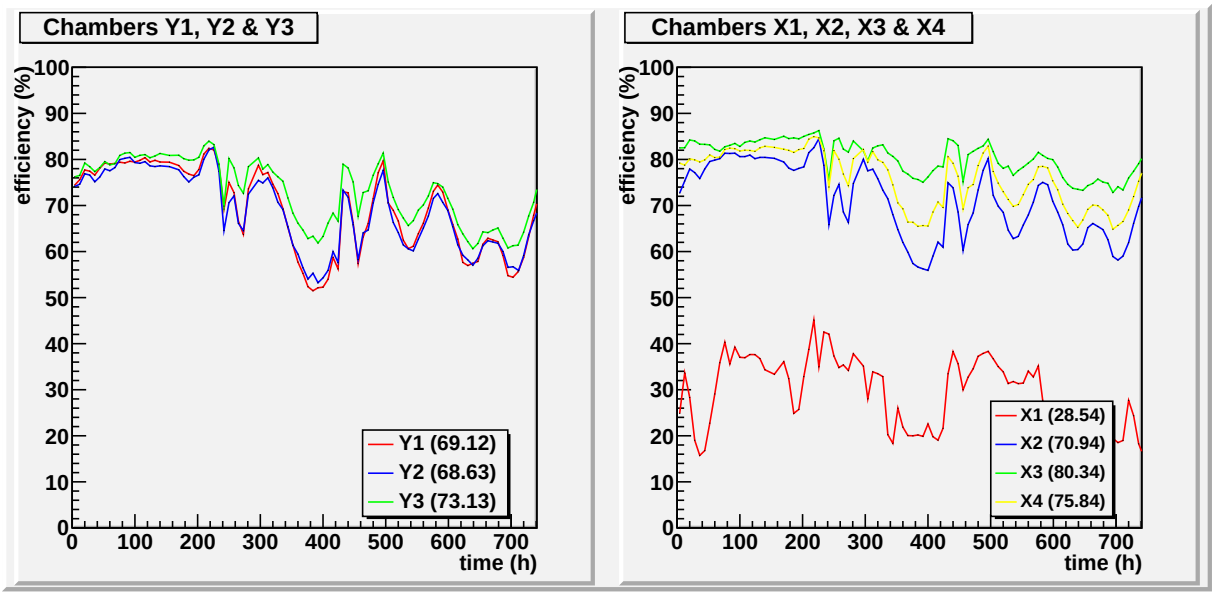
Stack 5, plank 78

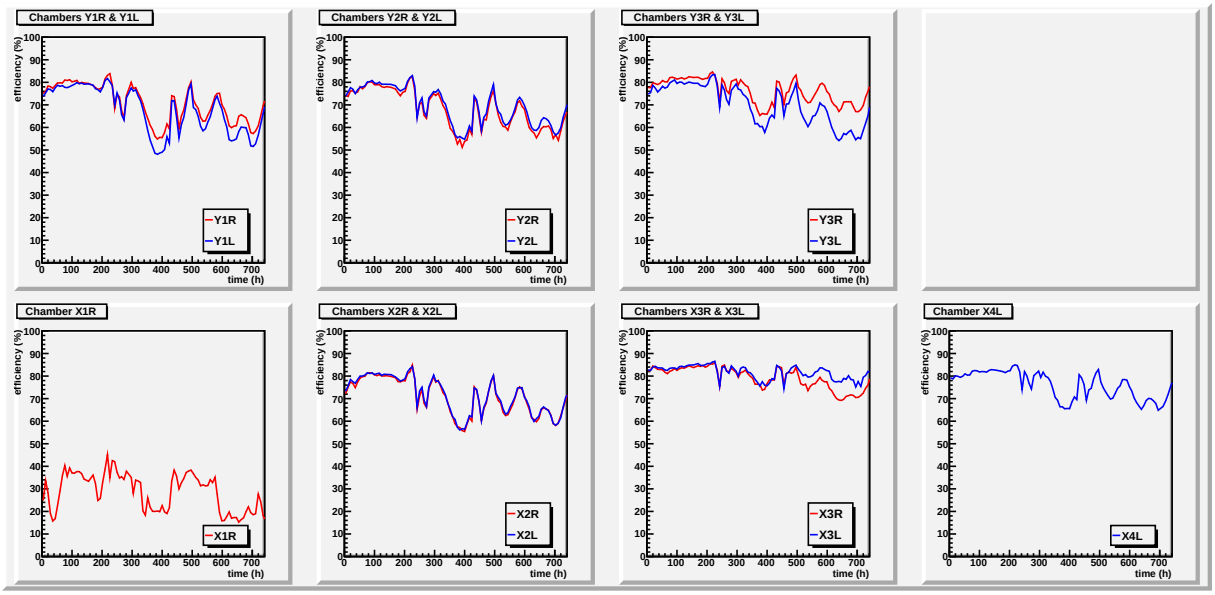




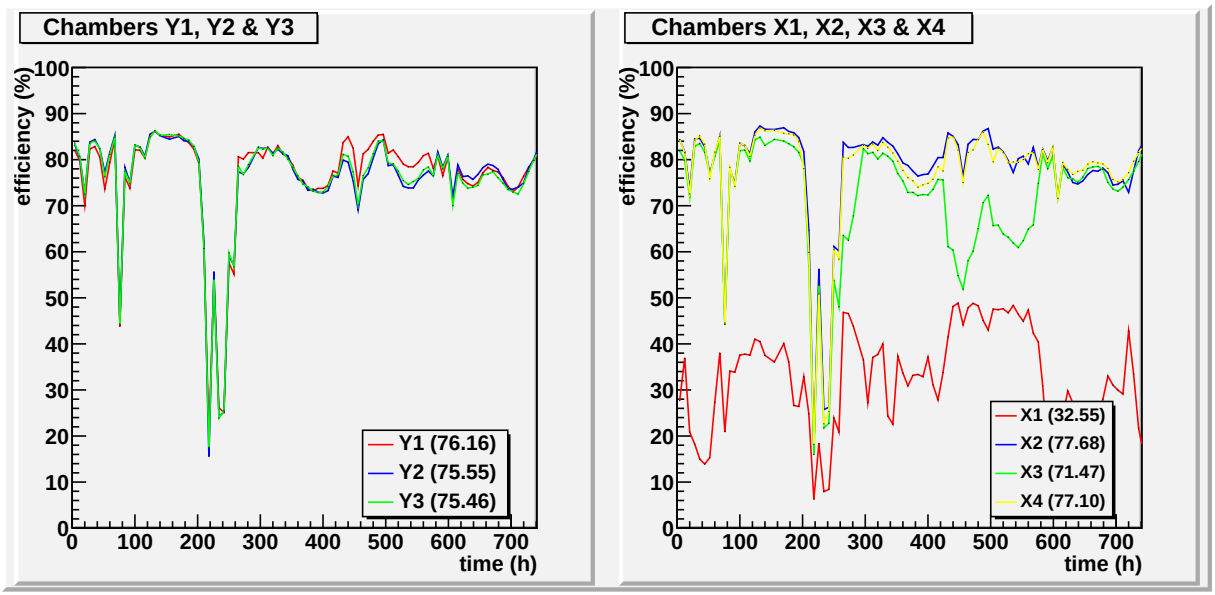
B.2 Efficiency vs. time of stack 6 (started 25 September, 2009)

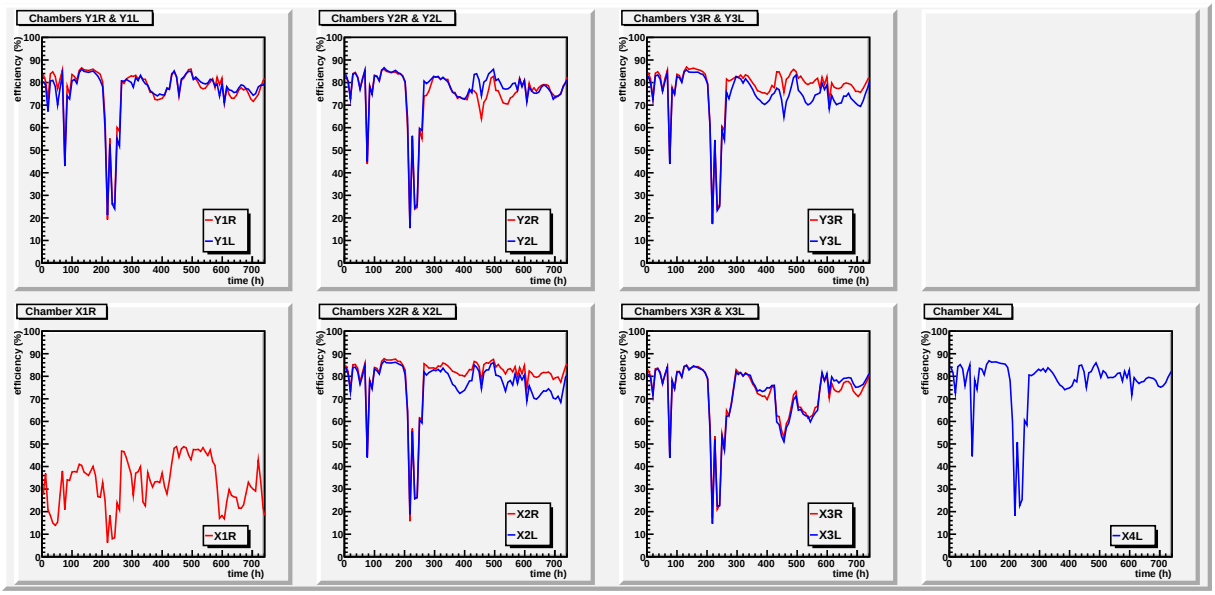
Stack 6, plank 13



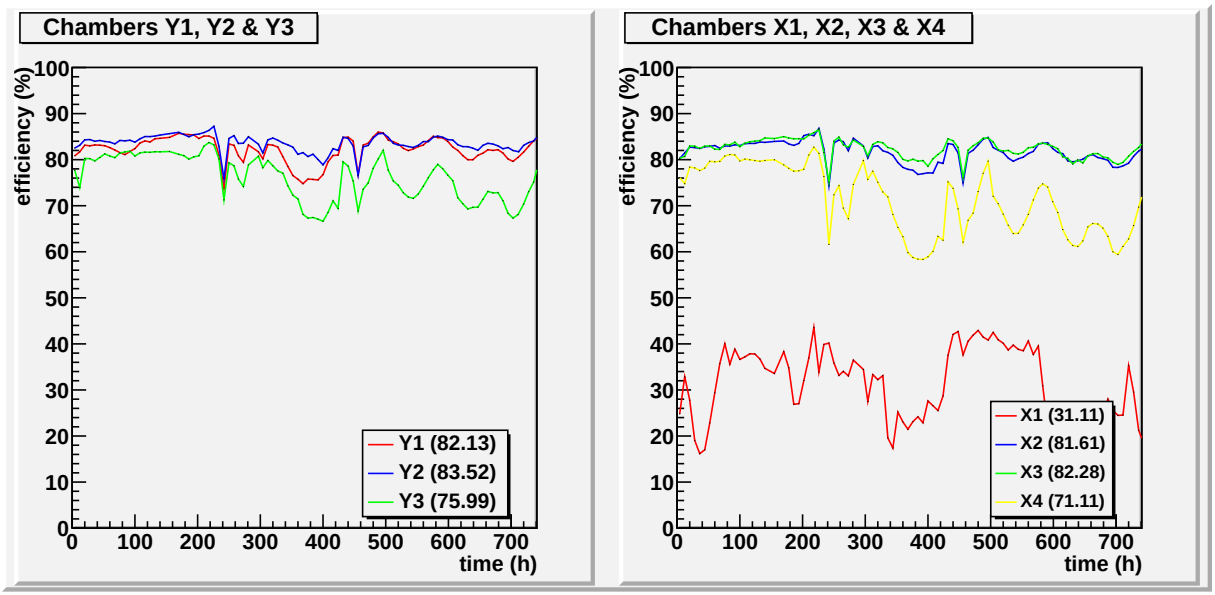


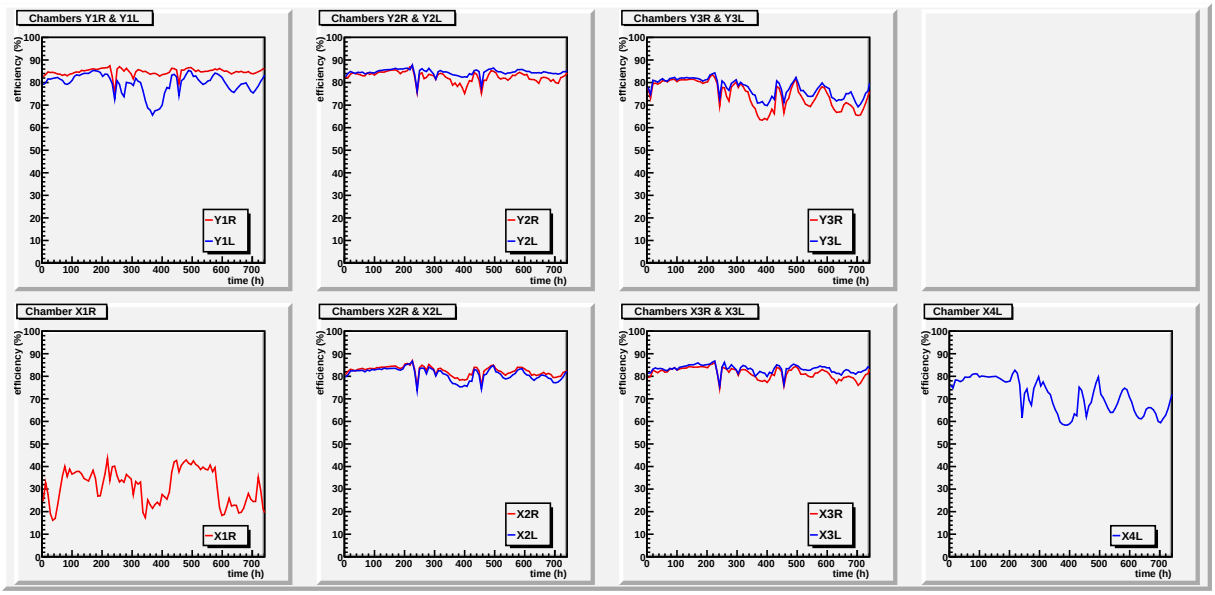
Stack 6, plank 21





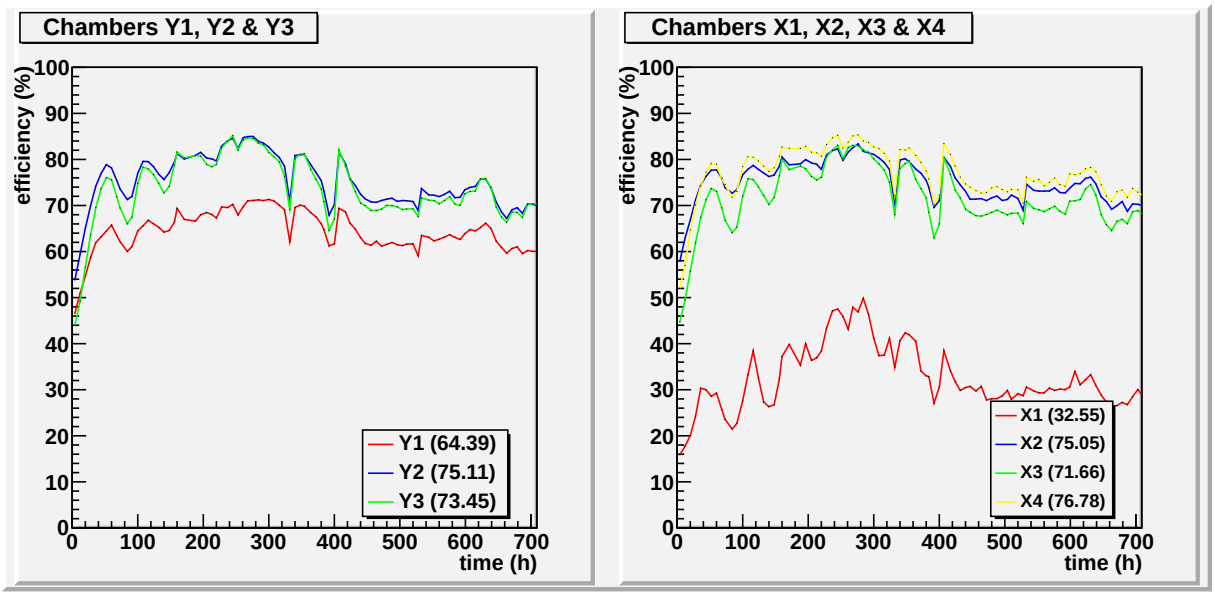
Stack 6, plank 33

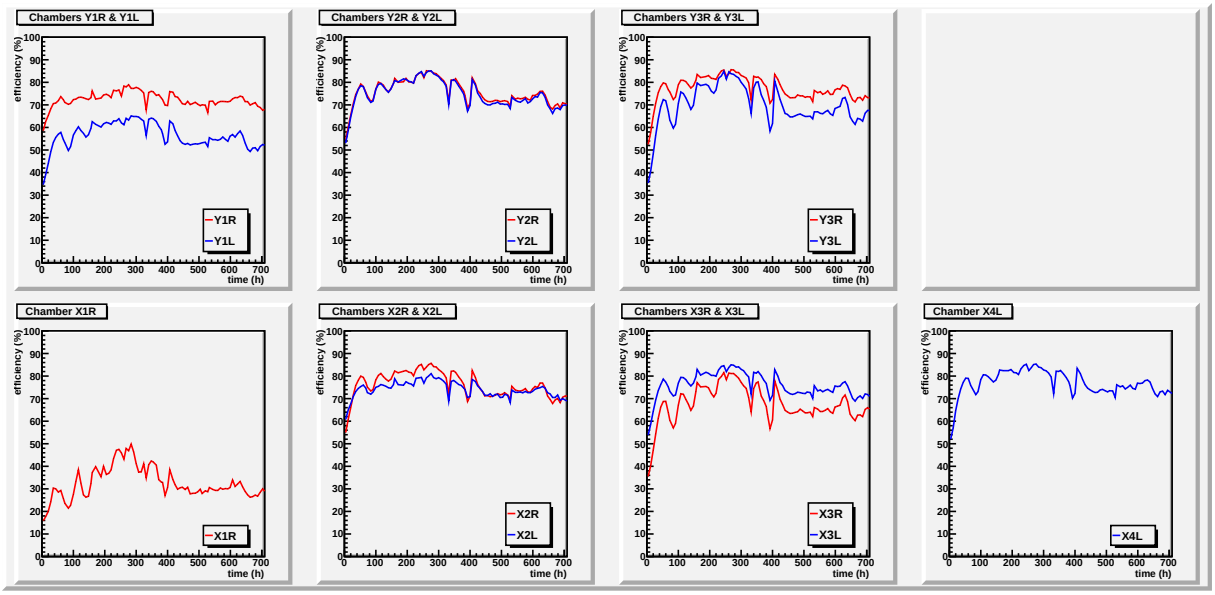




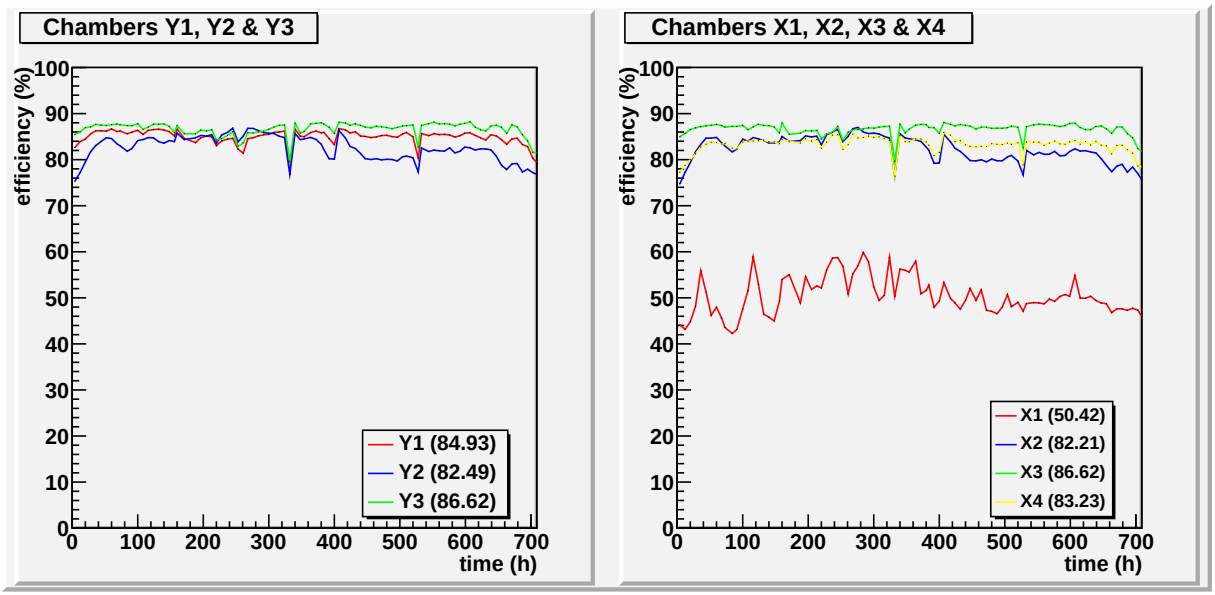
B.3 Efficiency vs. time of stack 7 (started 16 November, 2009)

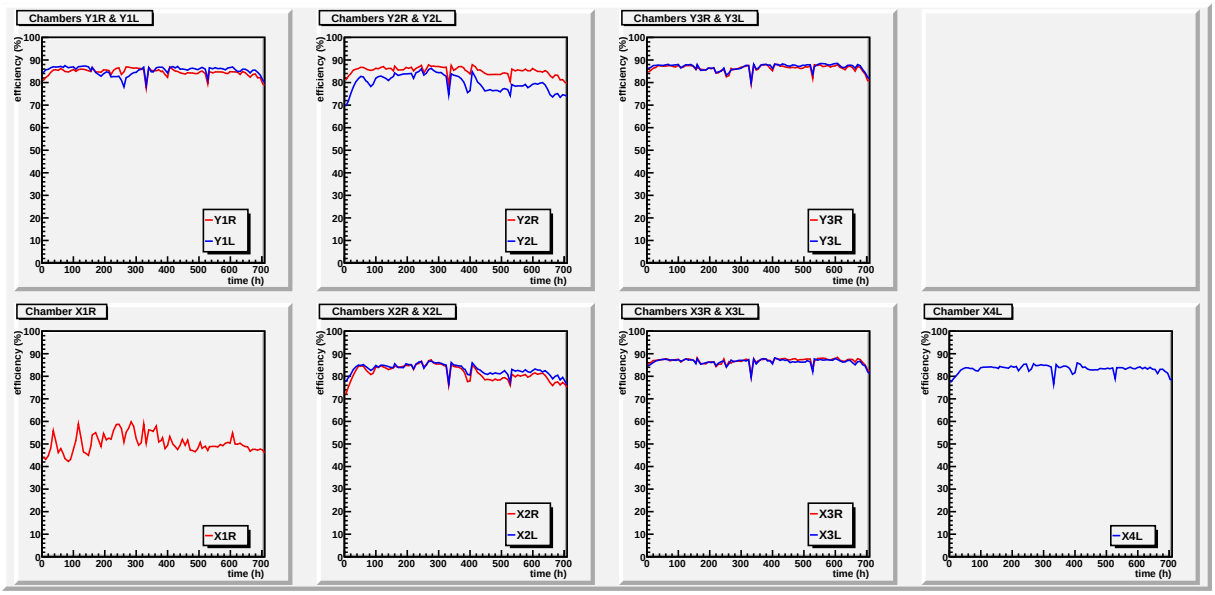
Stack 7, plank 20





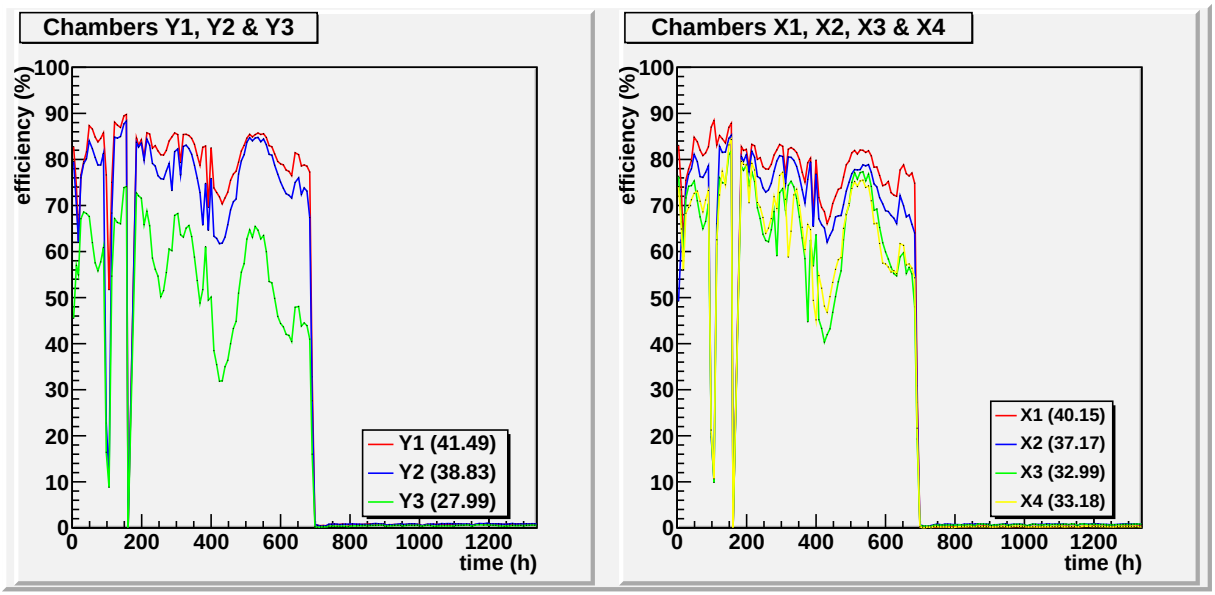
Stack 7, plank 80

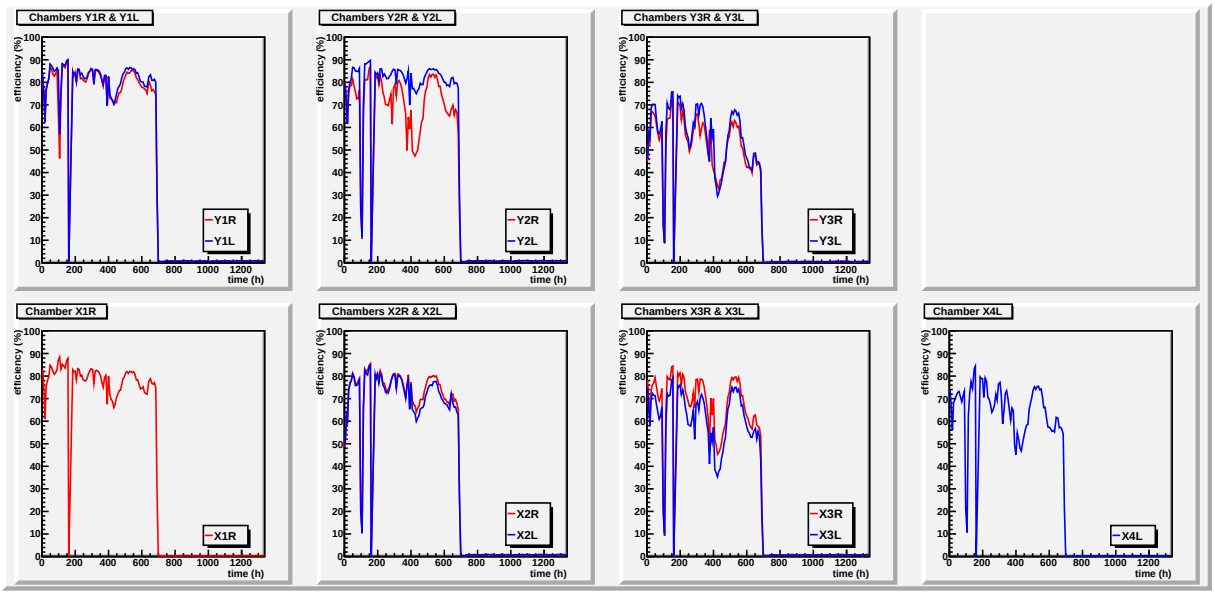




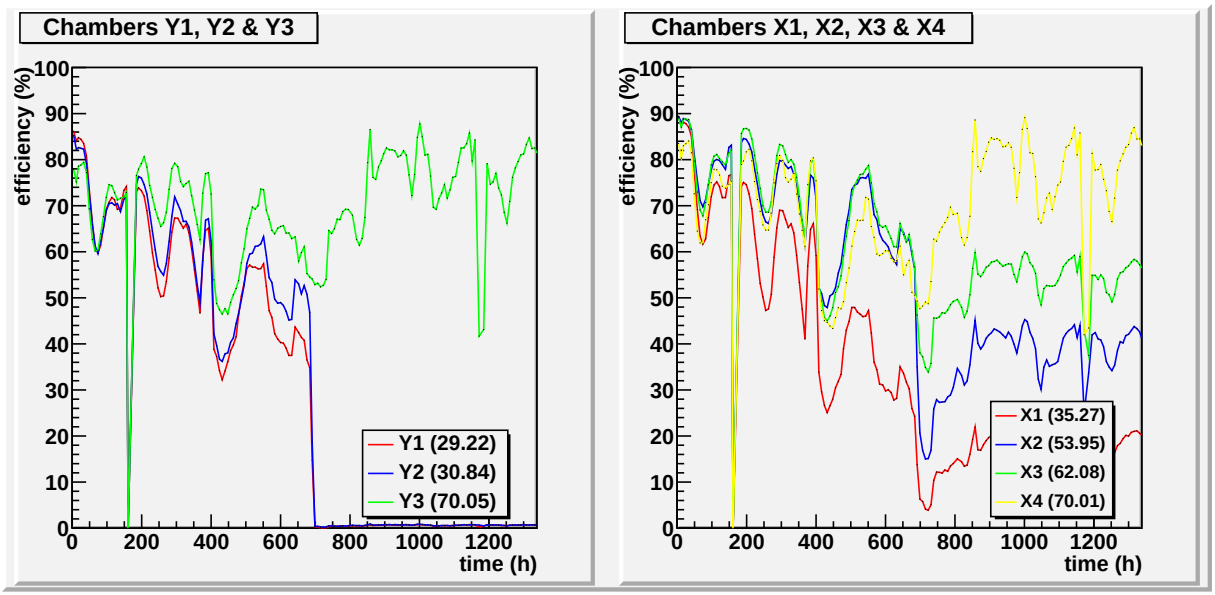
B.4 Efficiency vs. time of stack 8 (started 22 December, 2009)

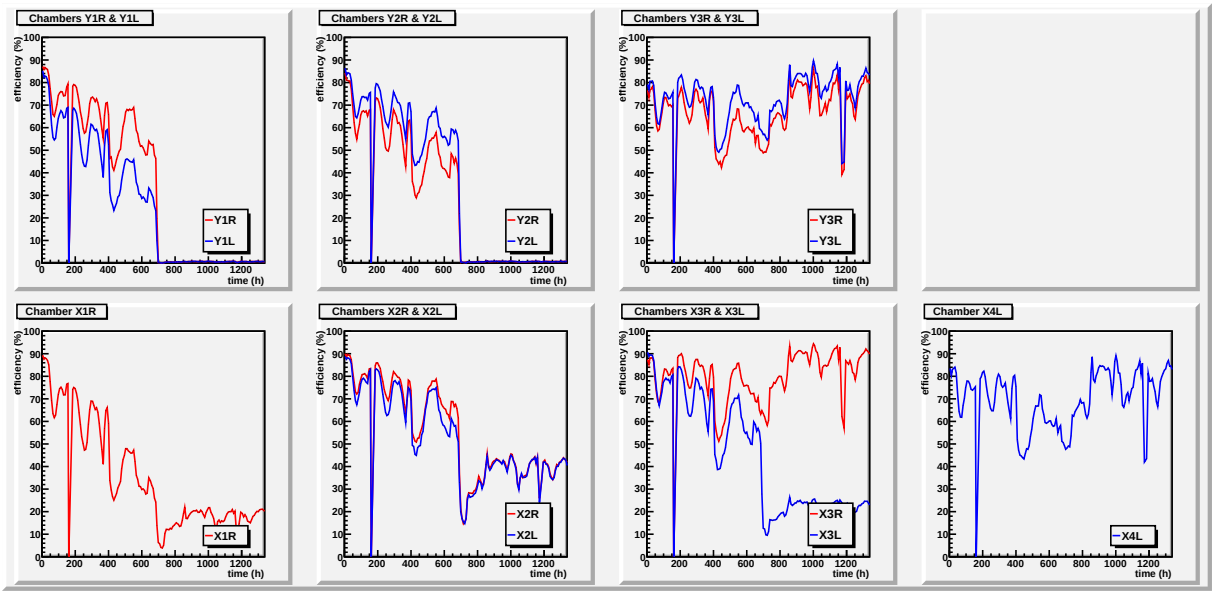
Stack 8, plank 2





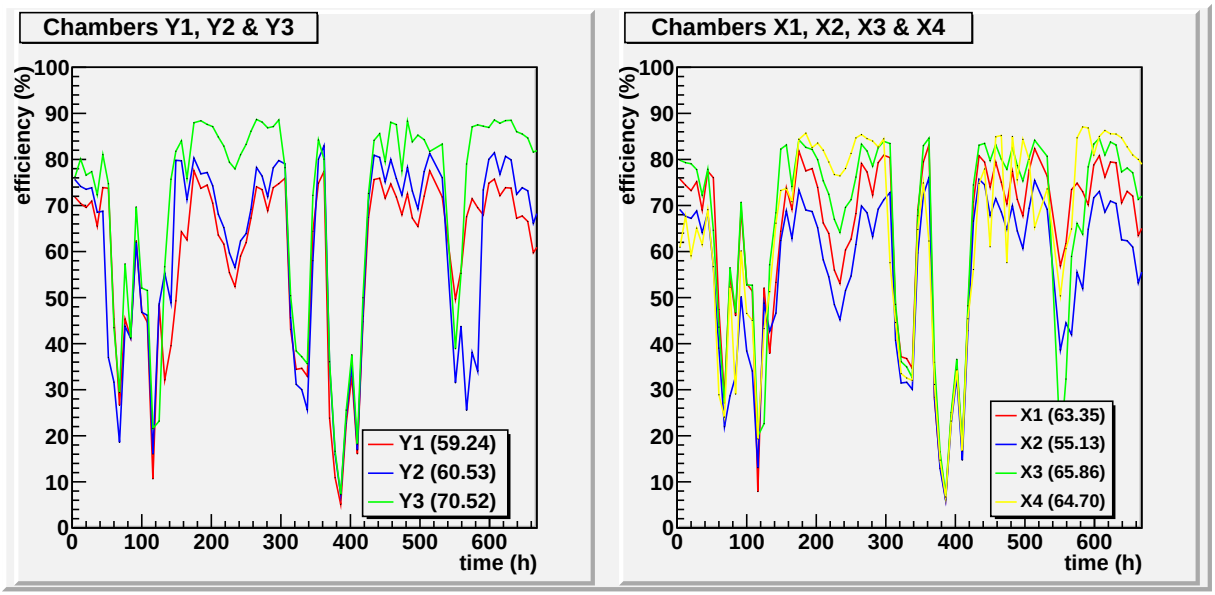
Stack 8, plank 7

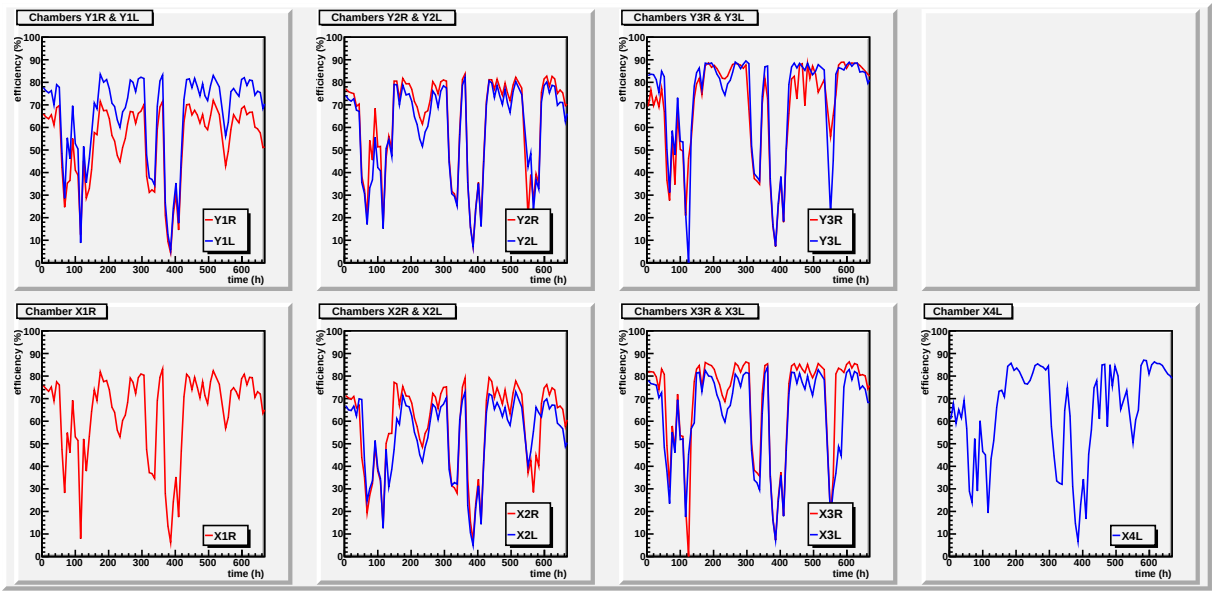




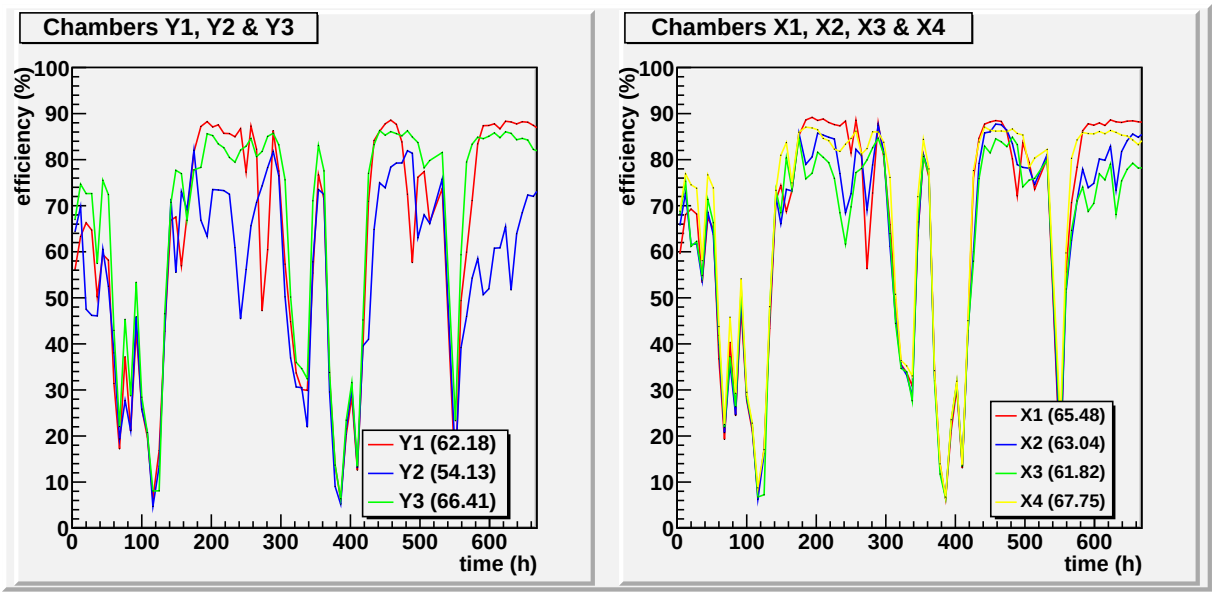
B.5 Efficiency vs. time of stack 9 (started 25 February, 2010)

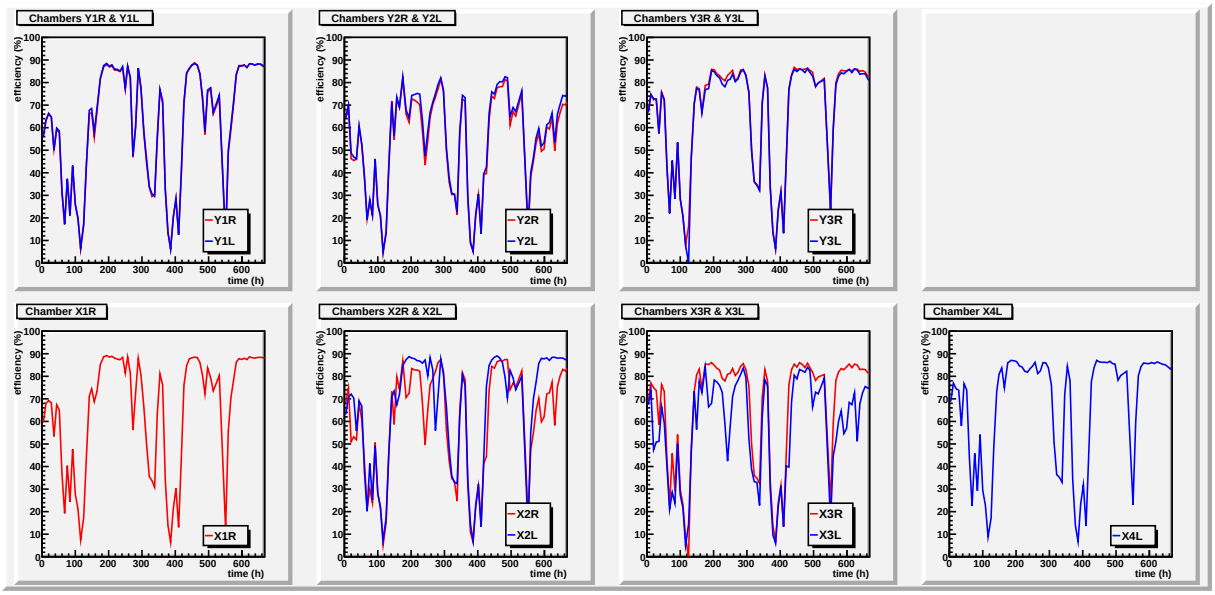
Stack 9, plank 1



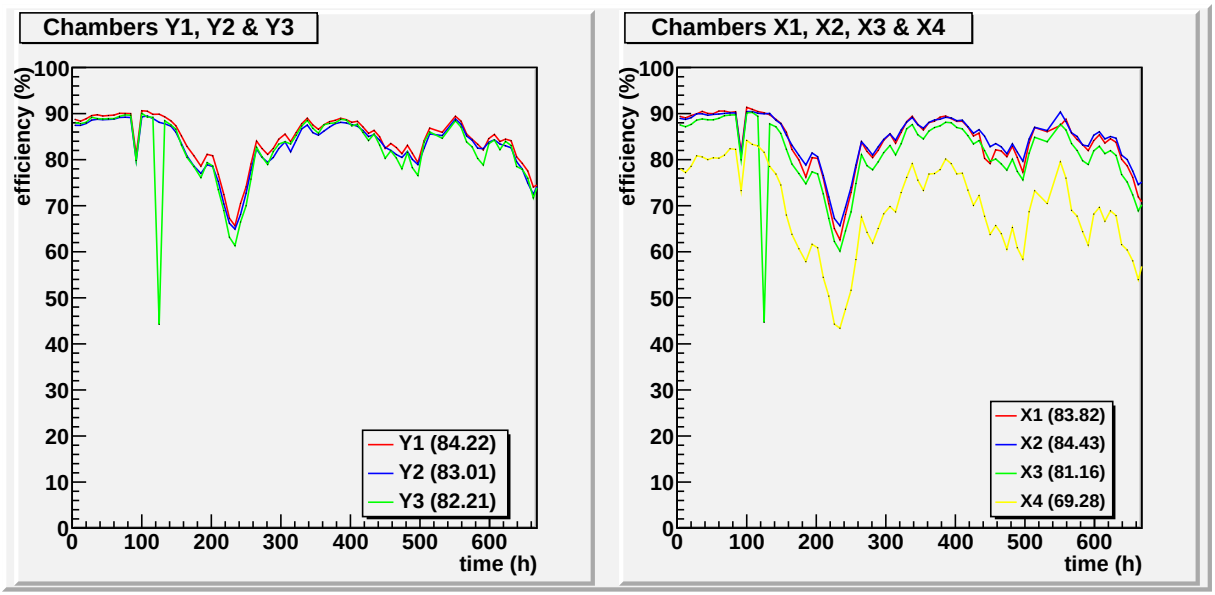


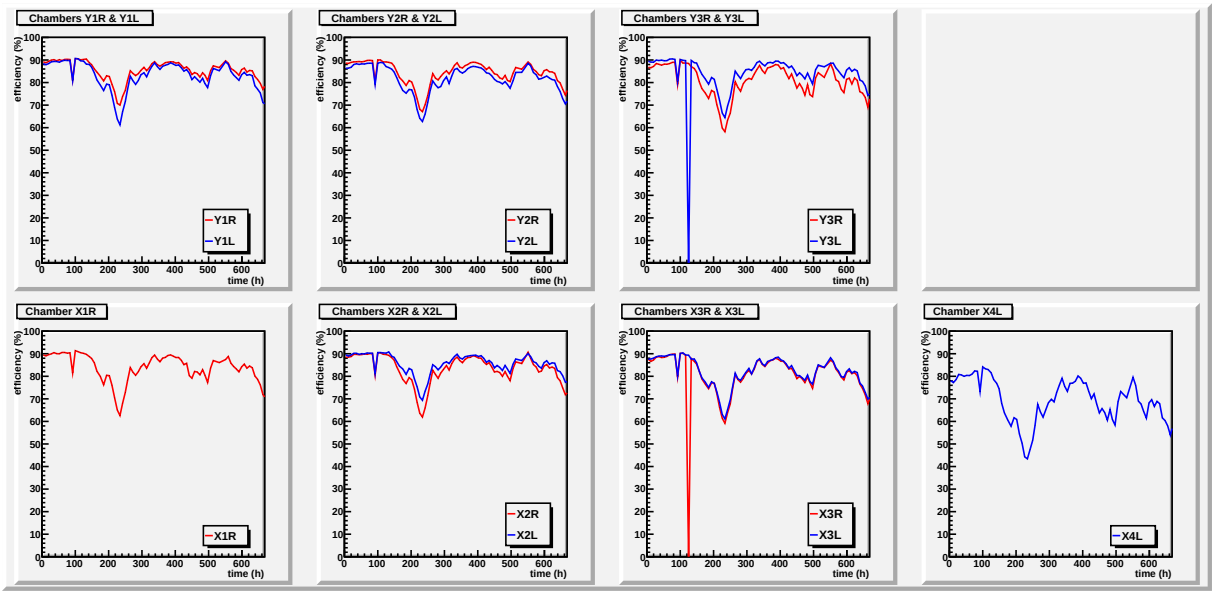
Stack 9, plank 37



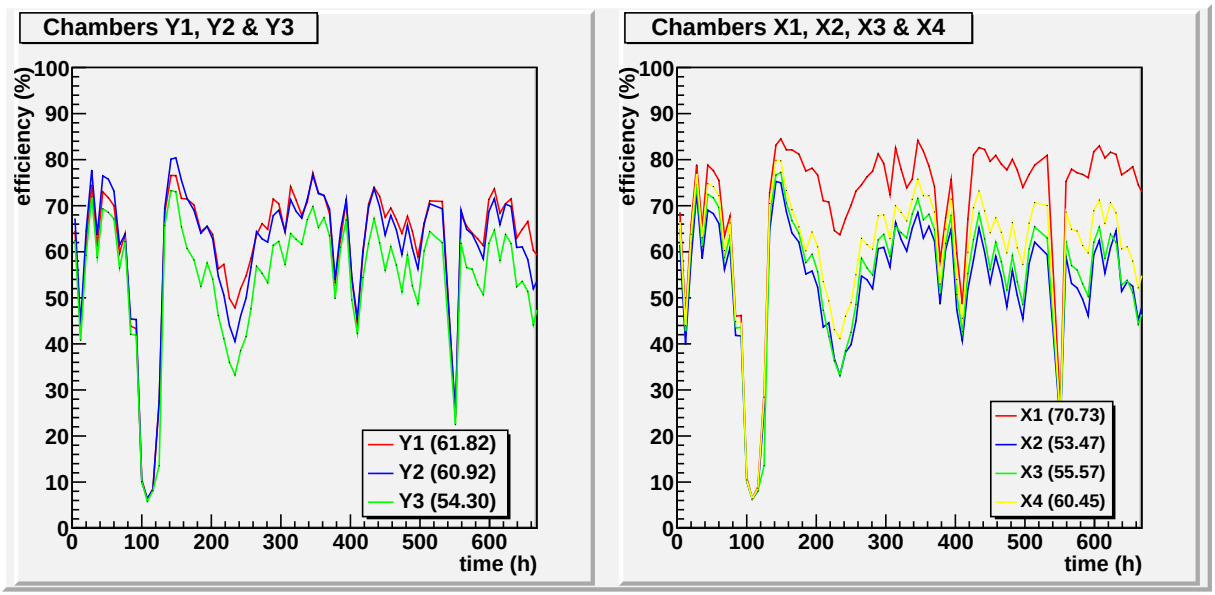


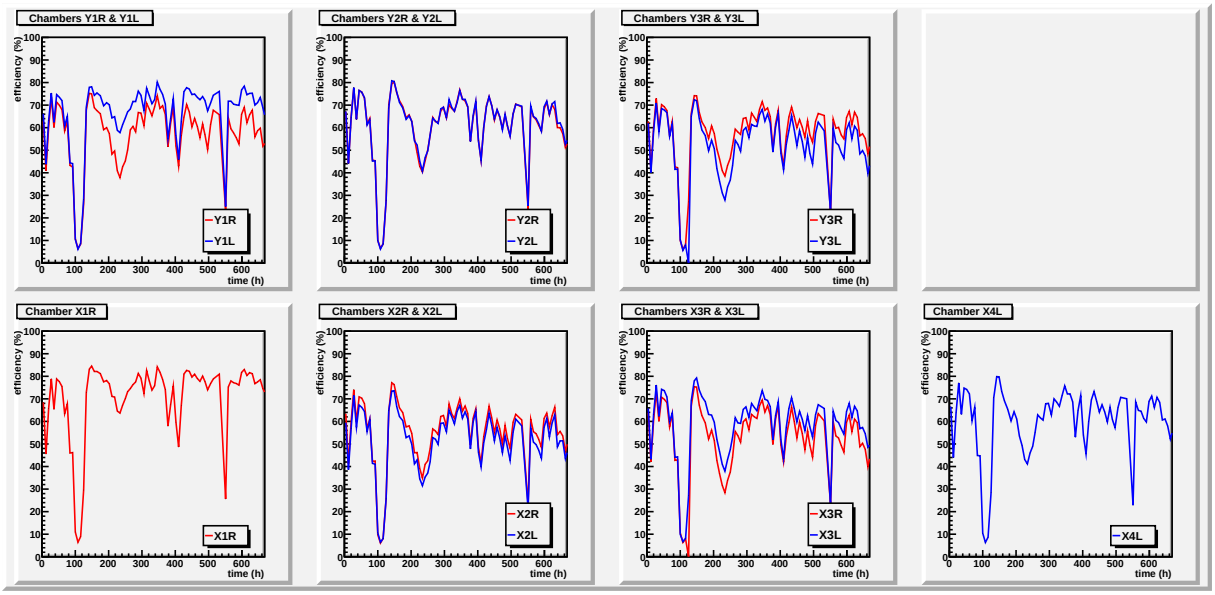
Stack 9, plank 47



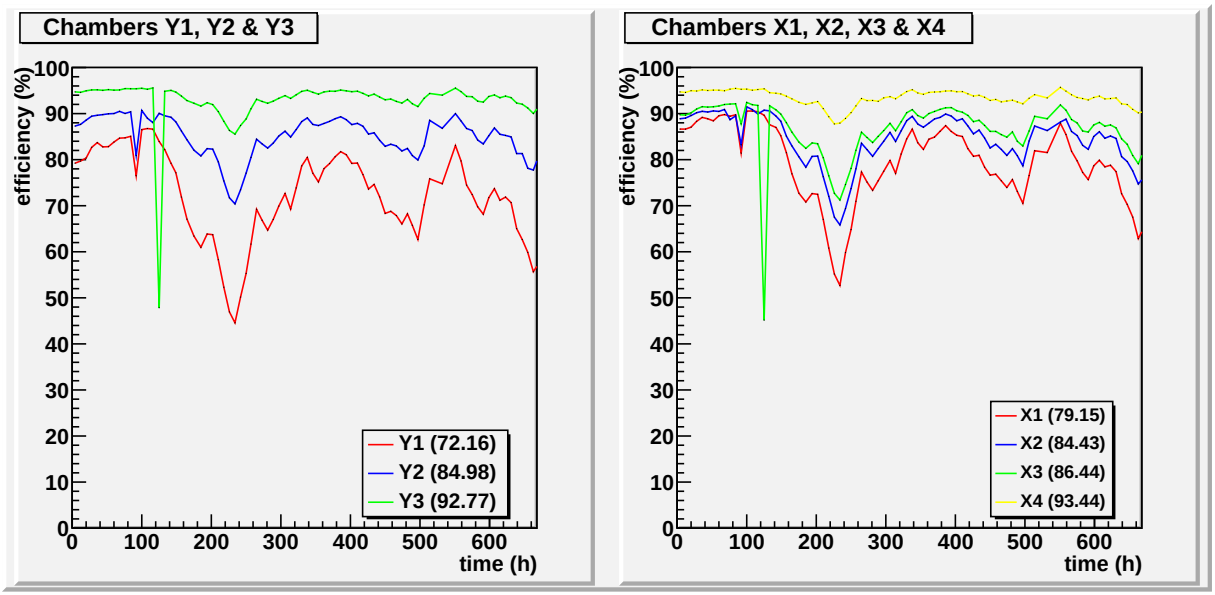


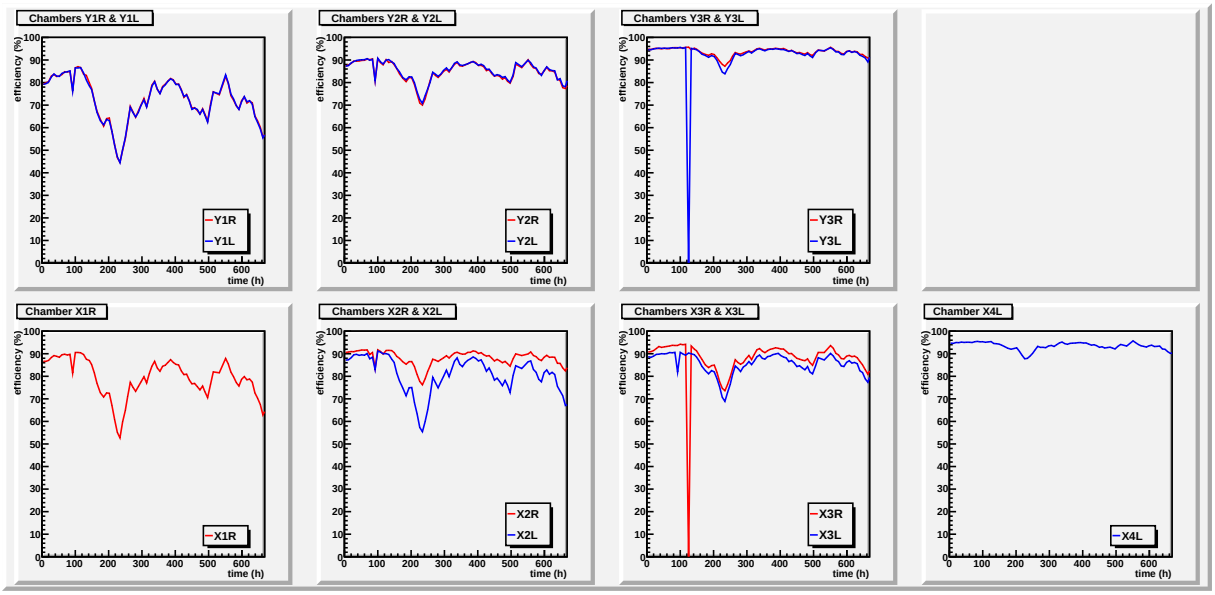
Stack 9, plank 49



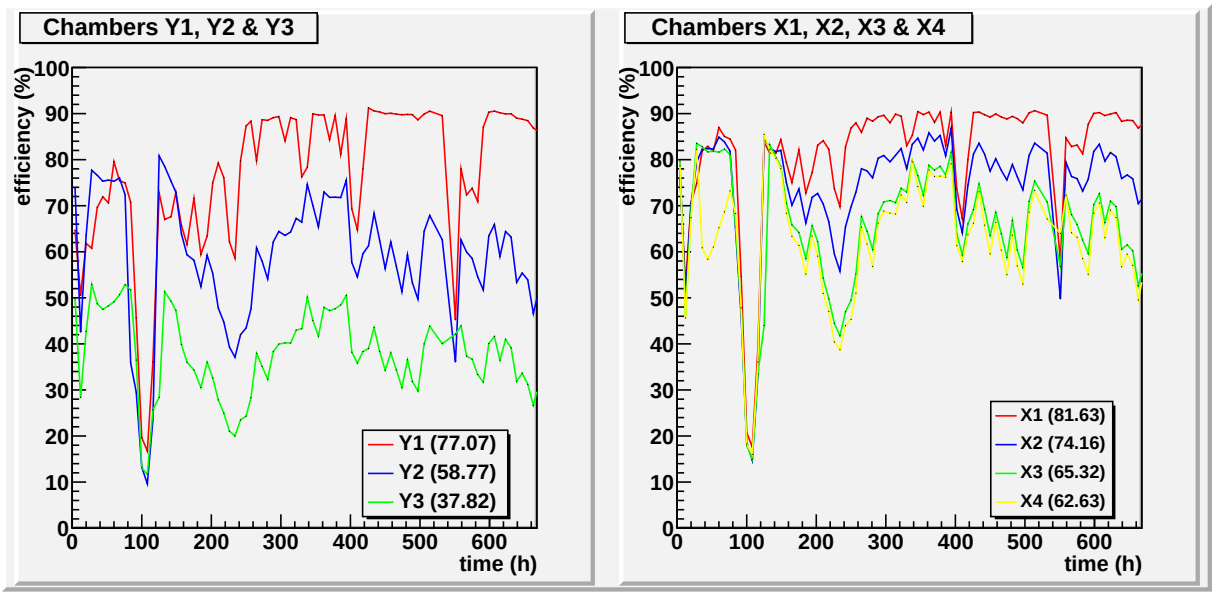


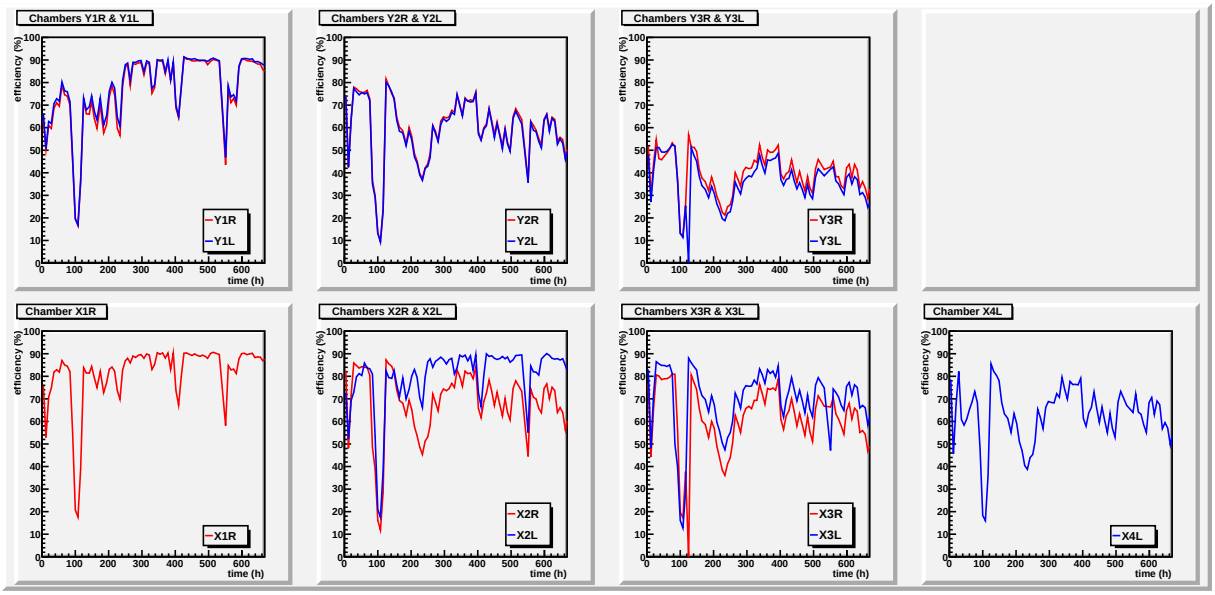
Stack 9, plank 51





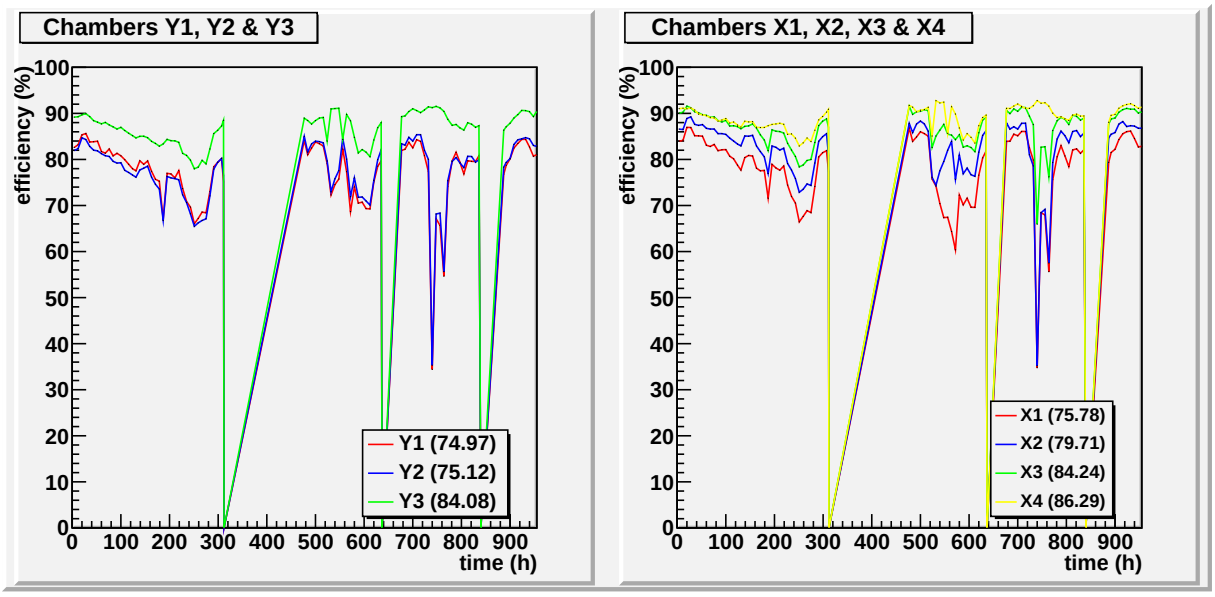
Stack 9, plank 53

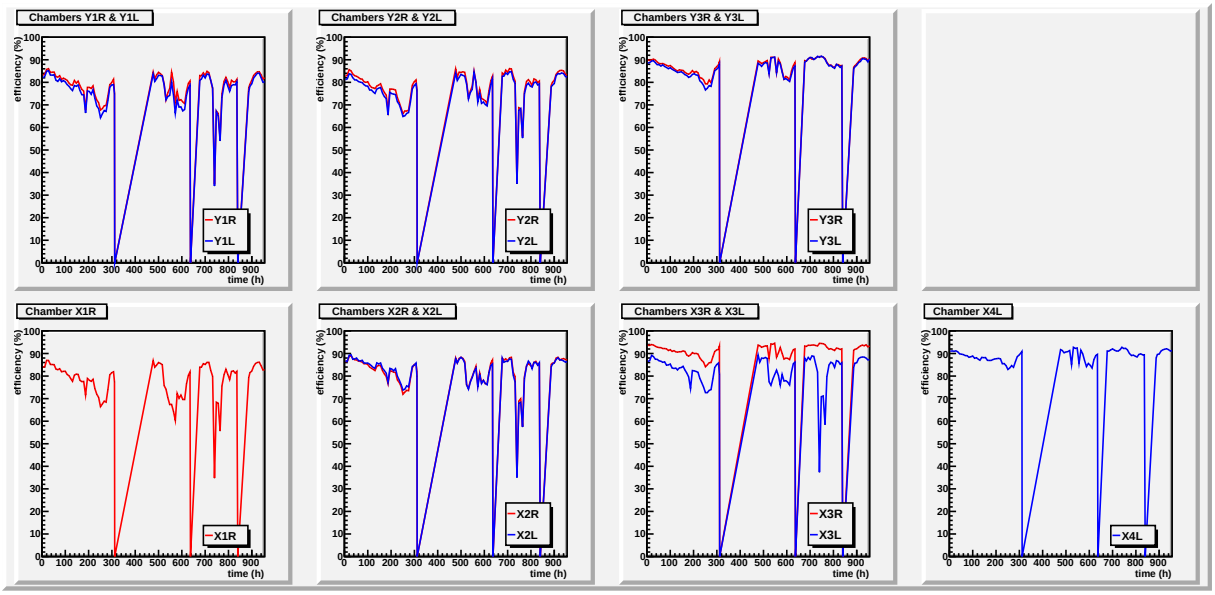




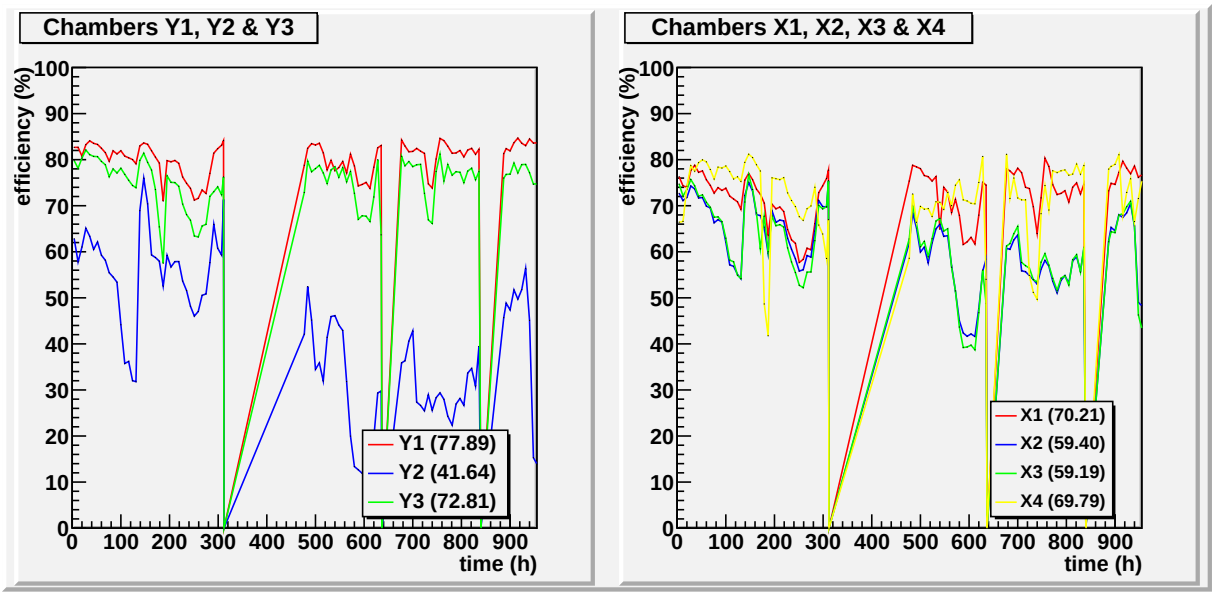
B.6 Efficiency vs. time of stack 10 (started 31 March, 2010)

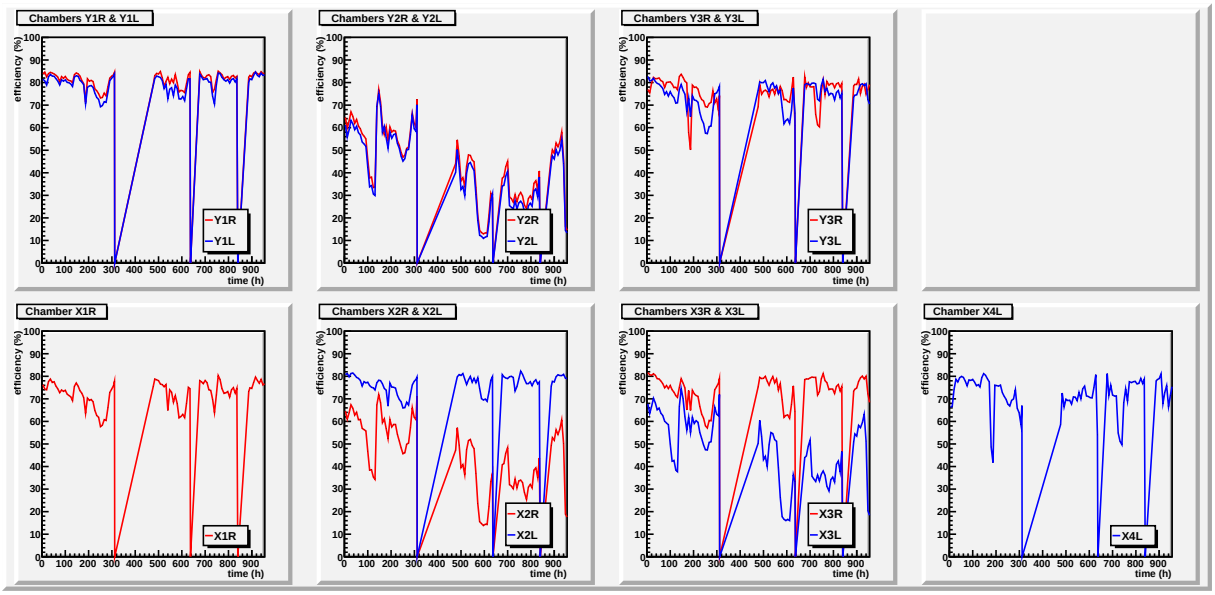
Stack 10, plank 41



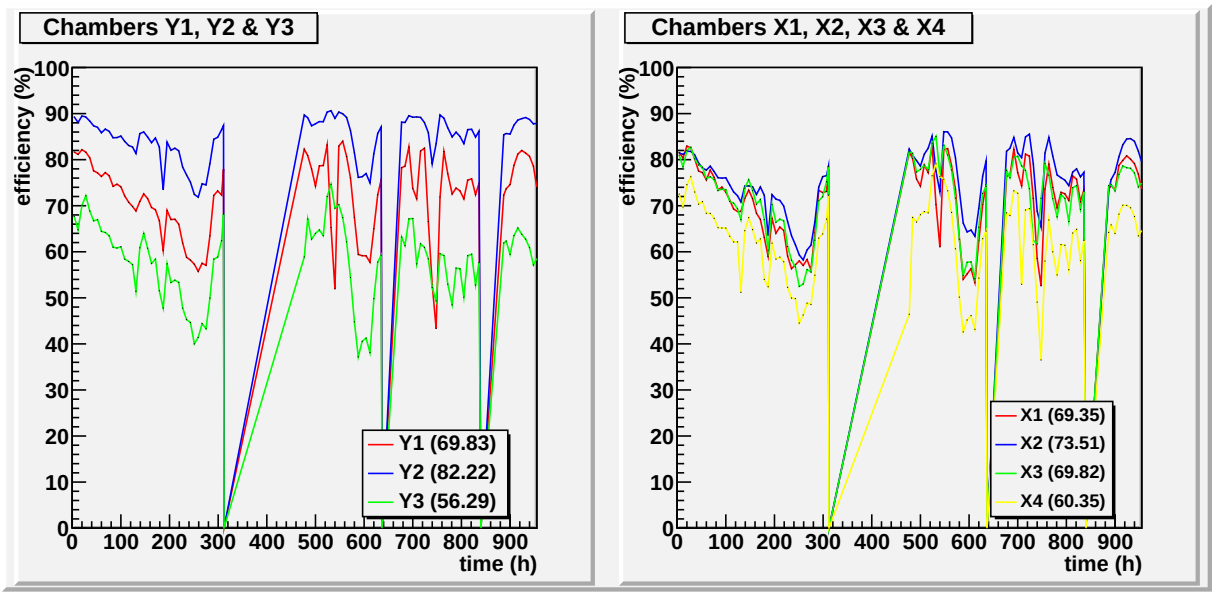


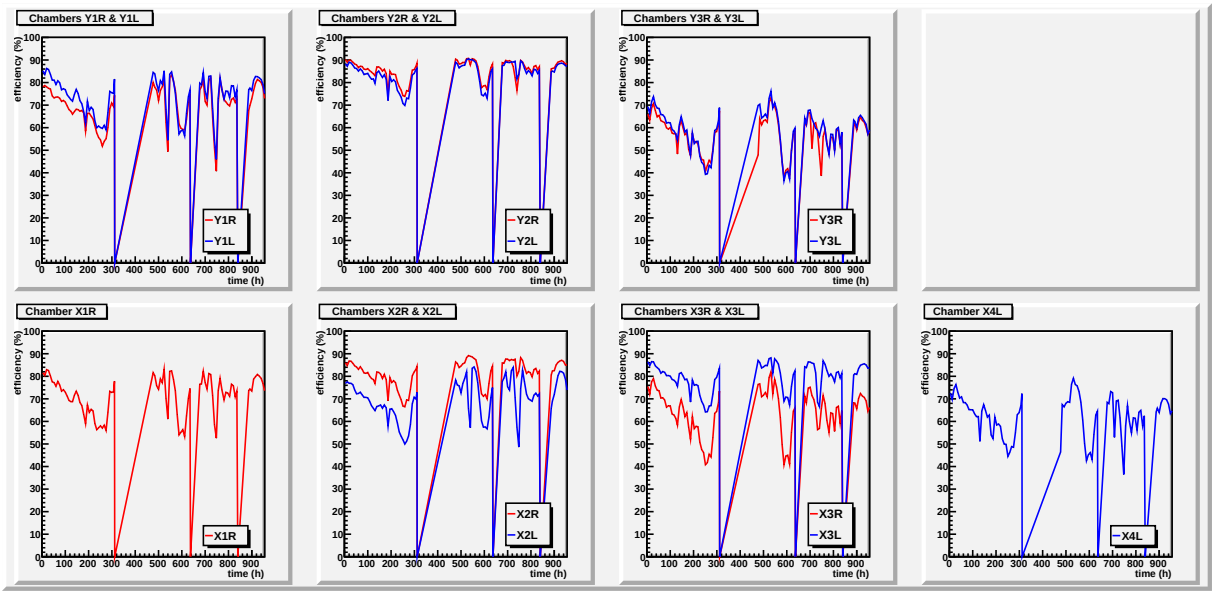
Stack 10, plank 42



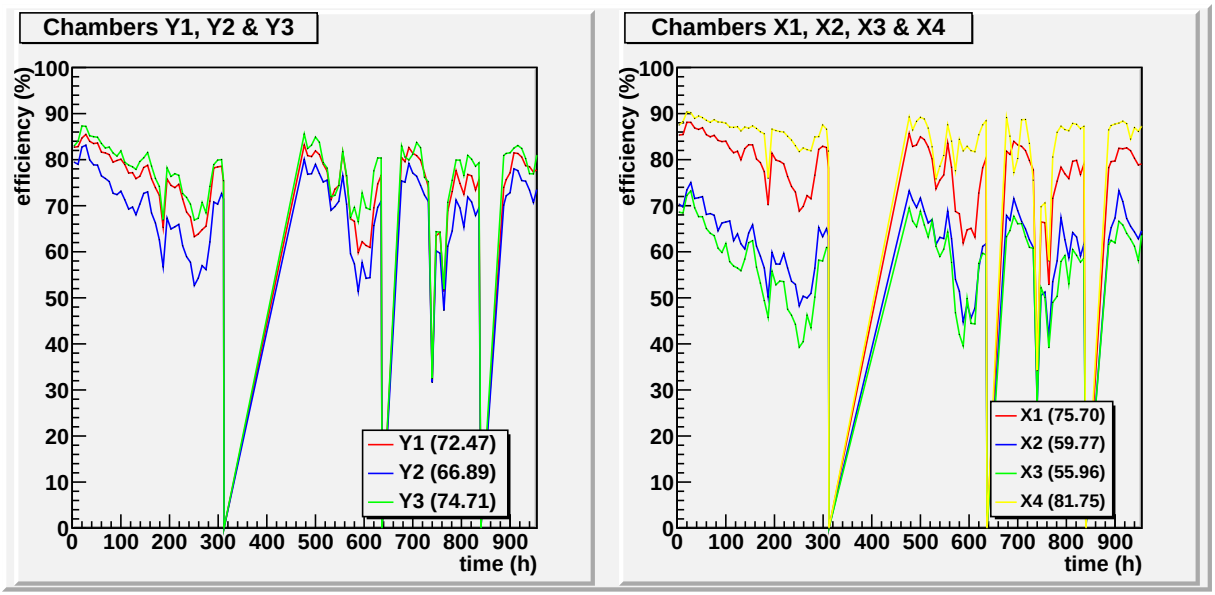


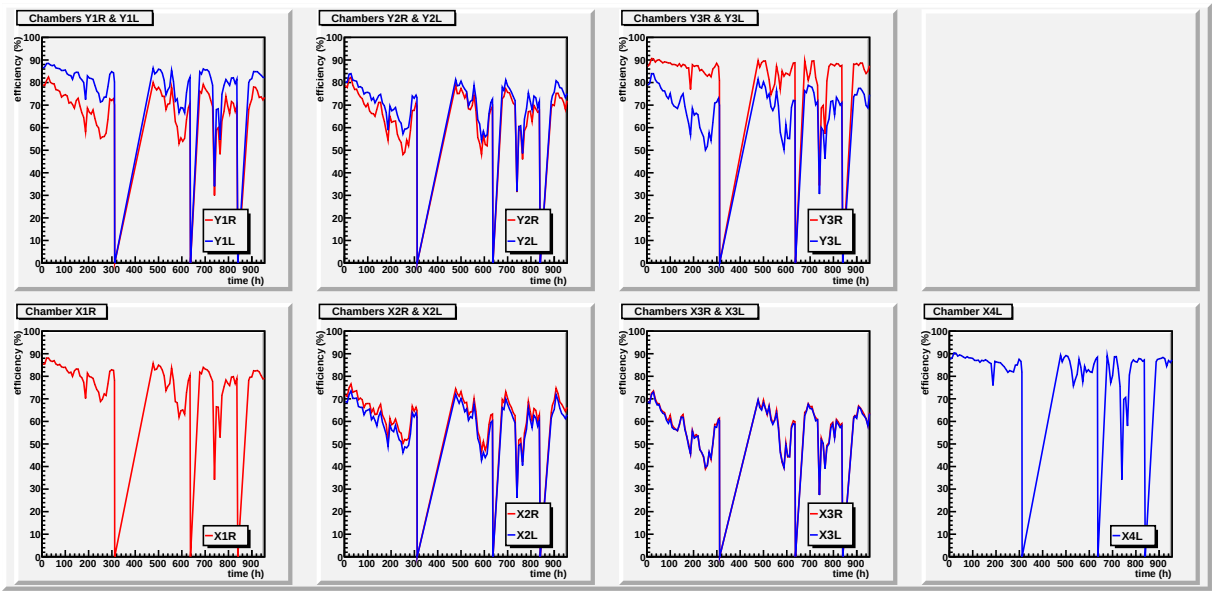
Stack 10, plank 45



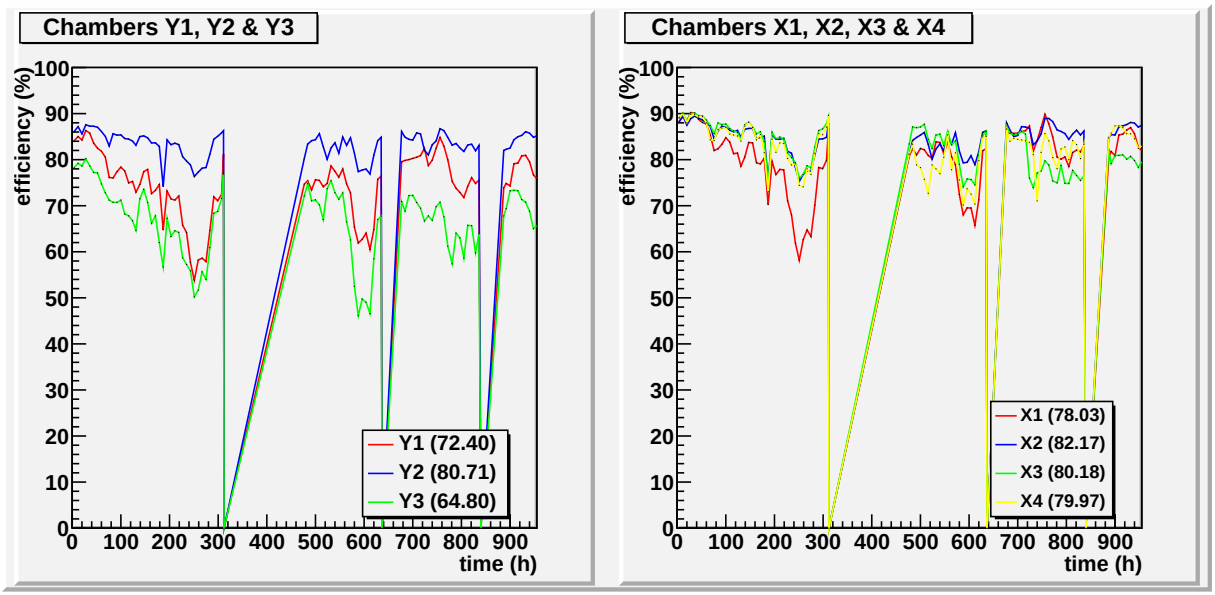


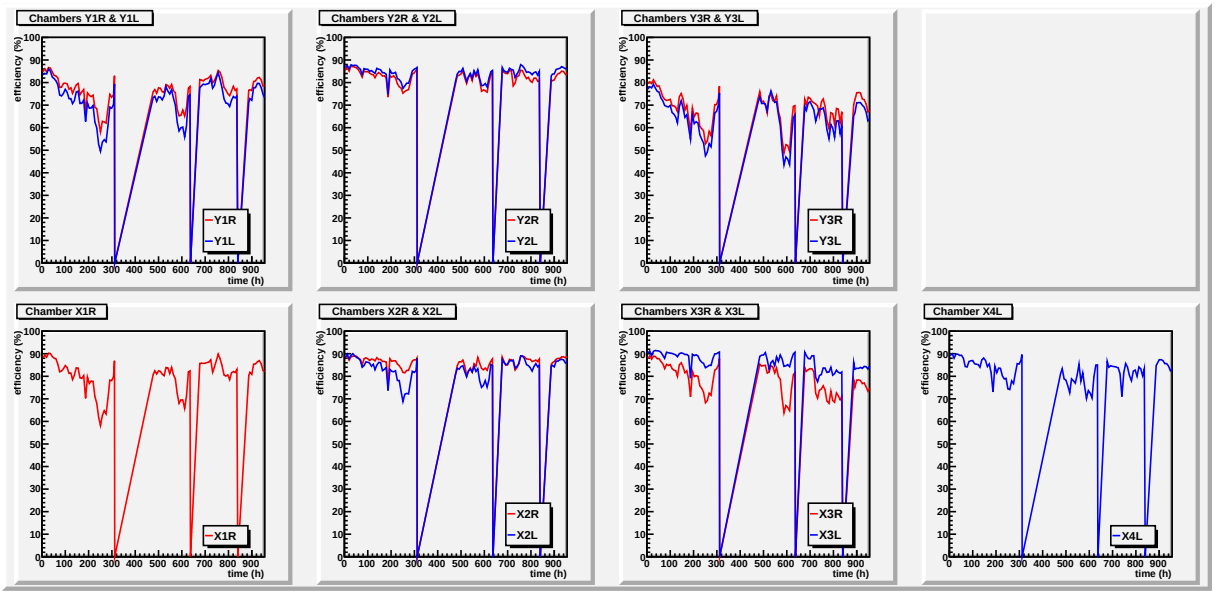
Stack 10, plank 58





Stack 10, plank 84

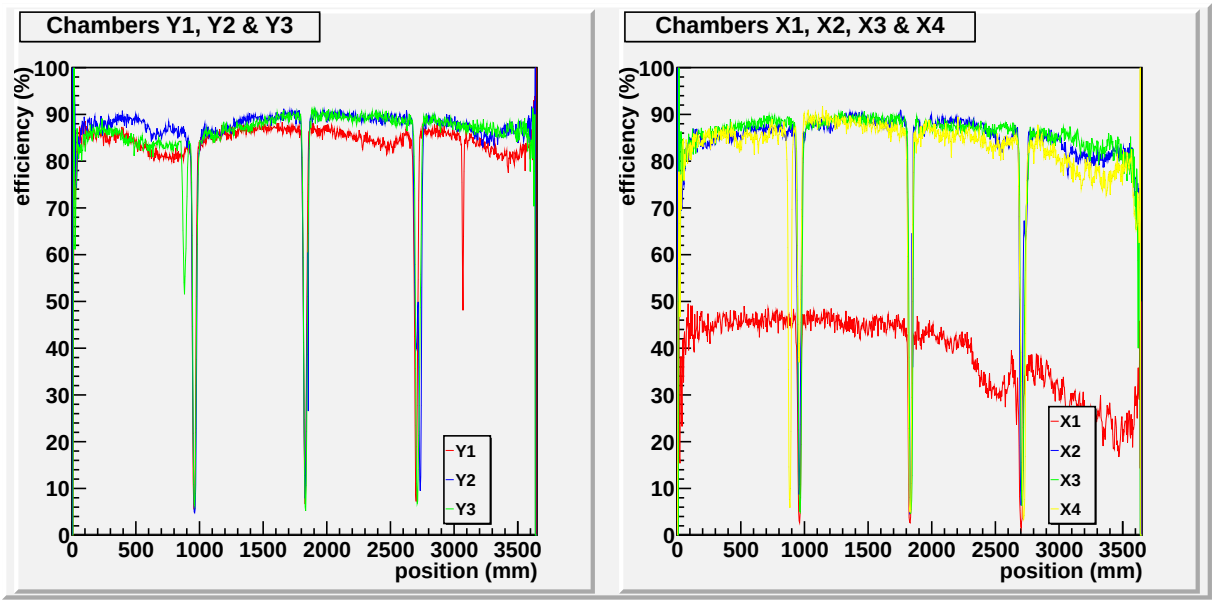




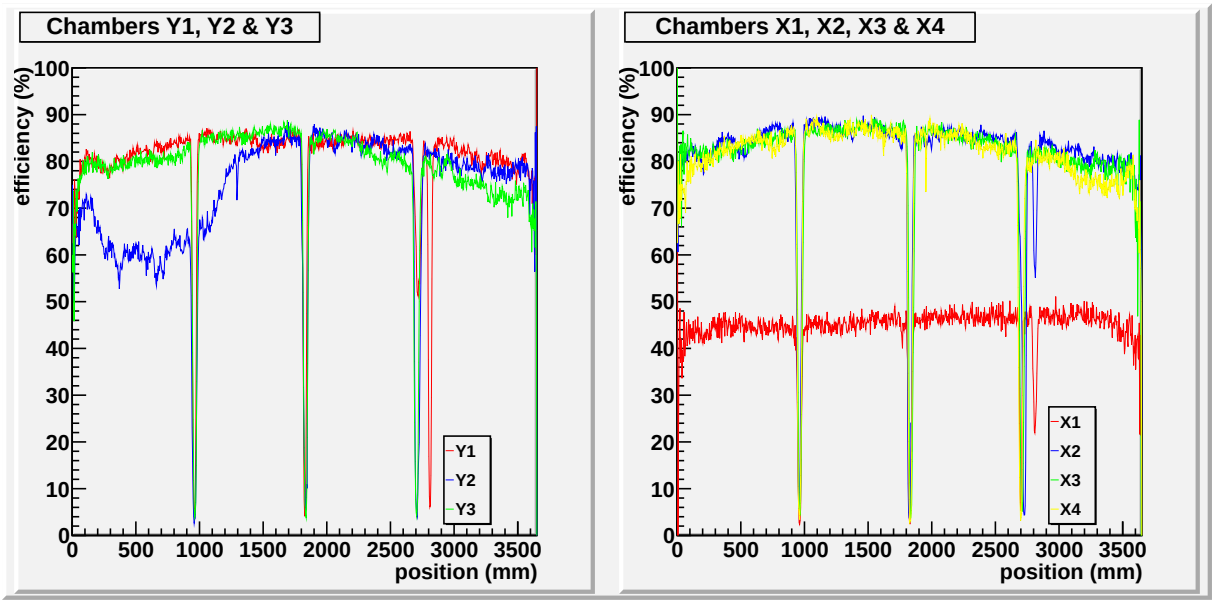
C Efficiency vs. position figures

C.1 Efficiency vs. position stack 5

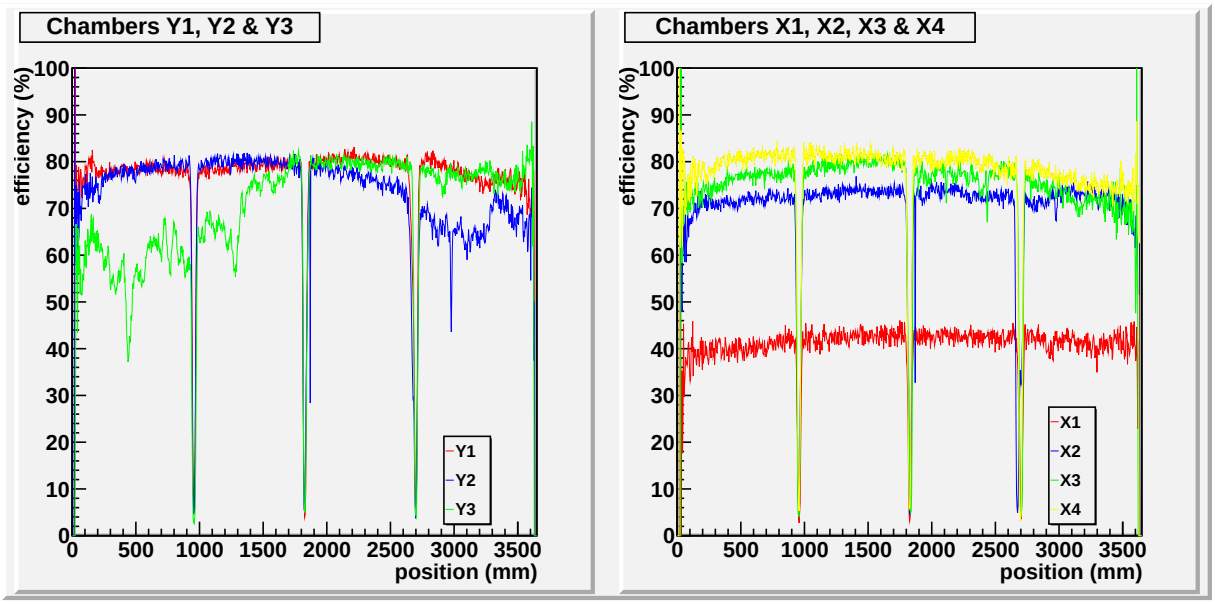
Stack 5, plank 34



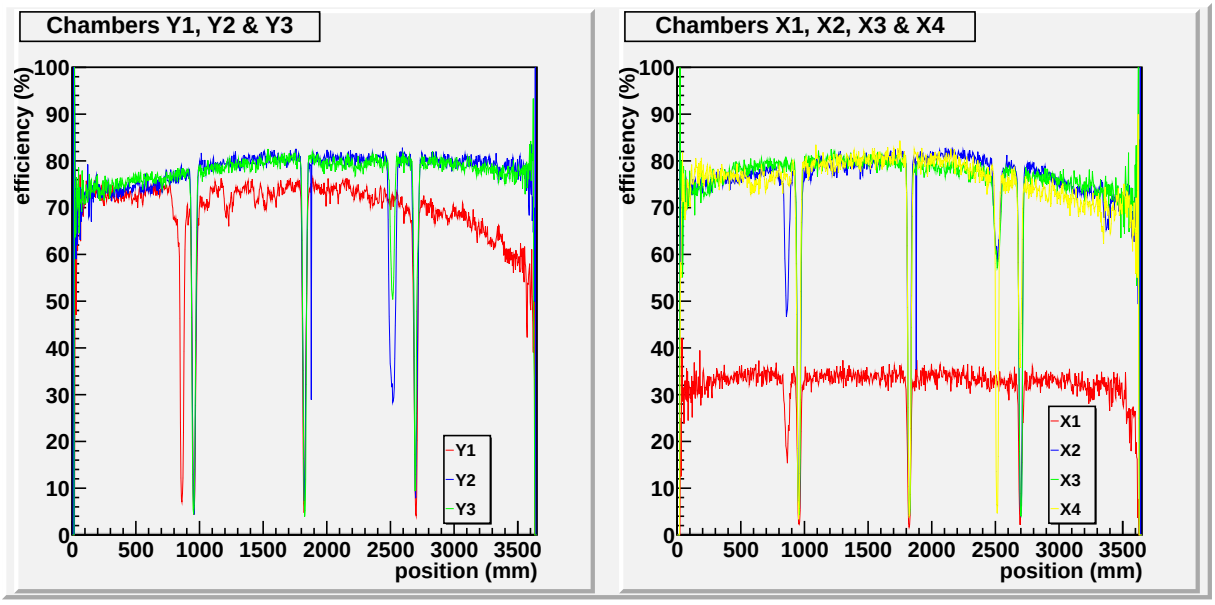
Stack 5, plank 35



Stack 5, plank 38

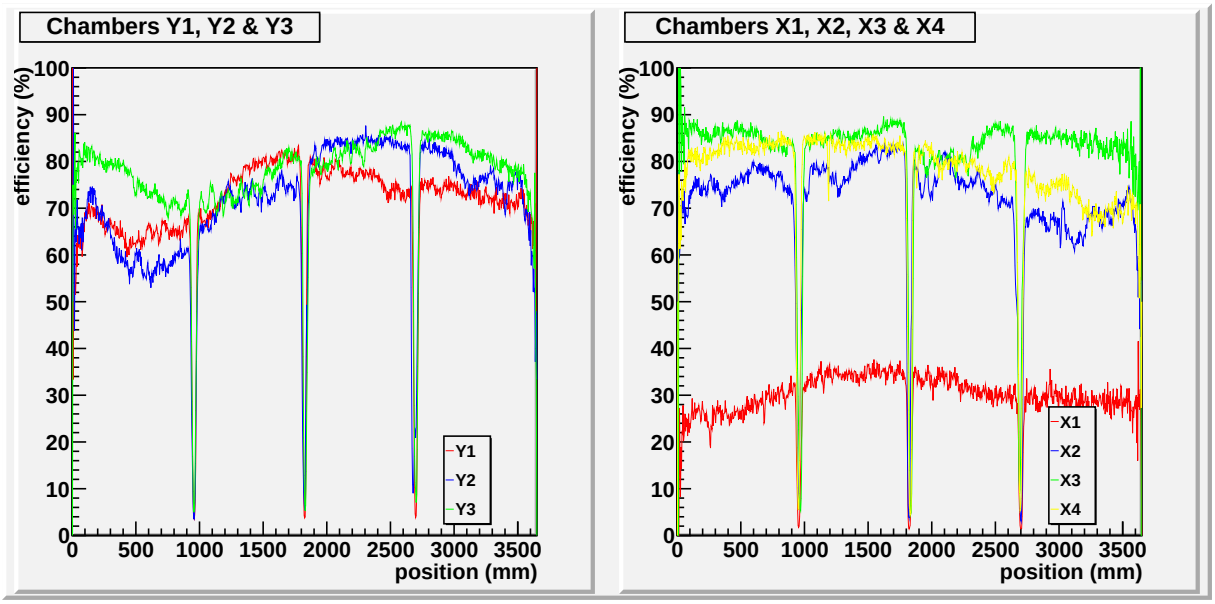


Stack 5, plank 78

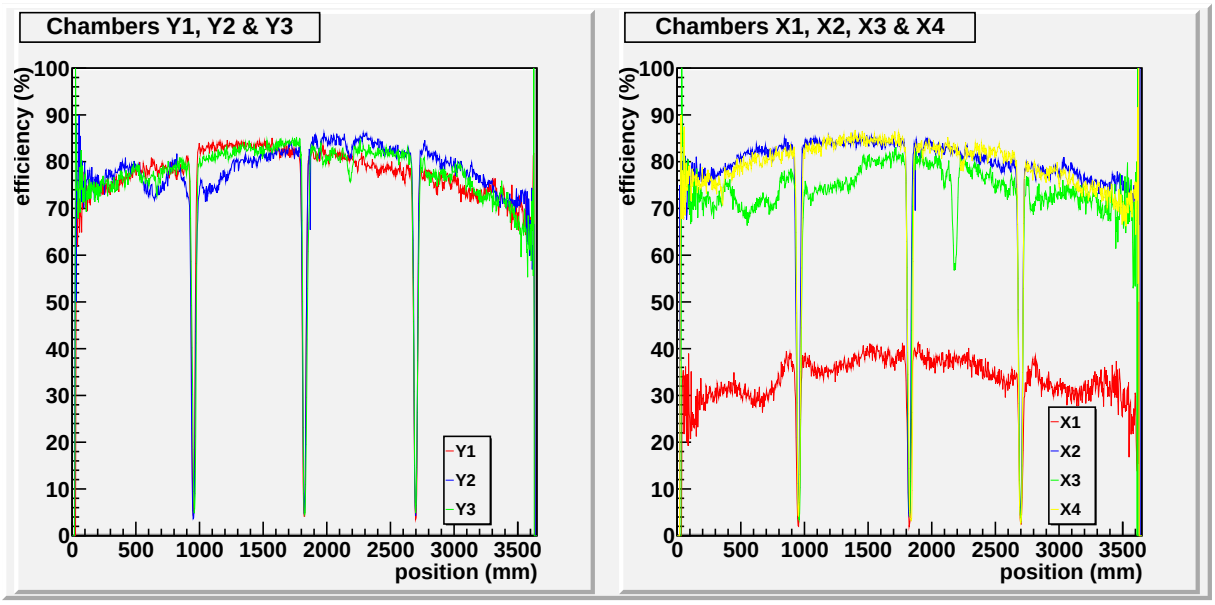


C.2 Efficiency vs. position stack 6

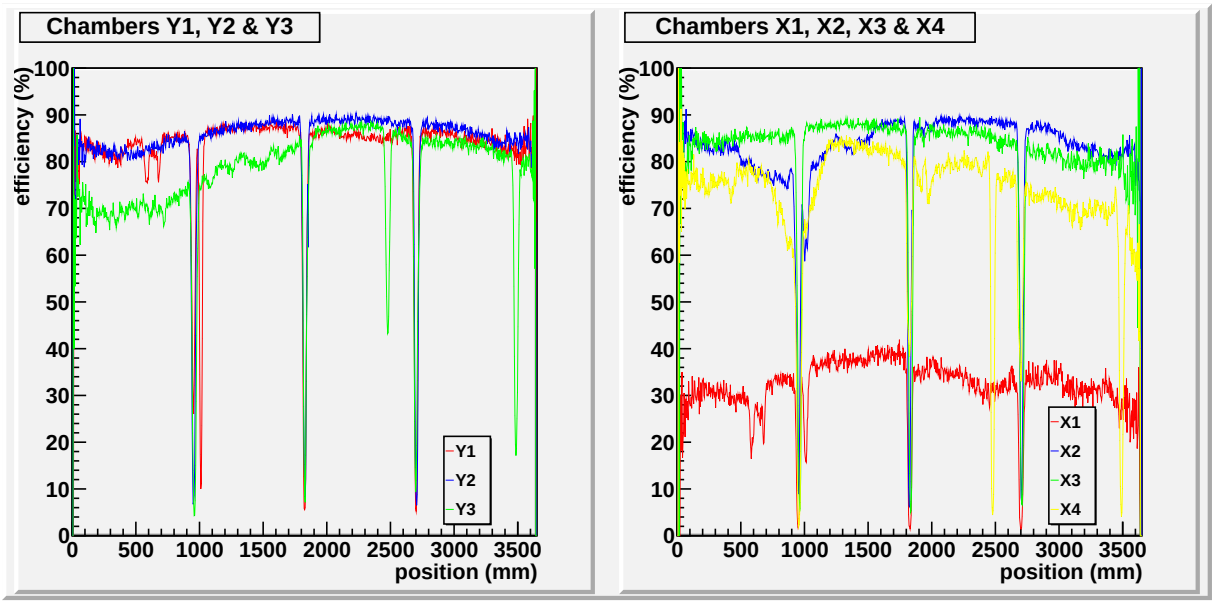
Stack 6, plank 13



Stack 6, plank 21

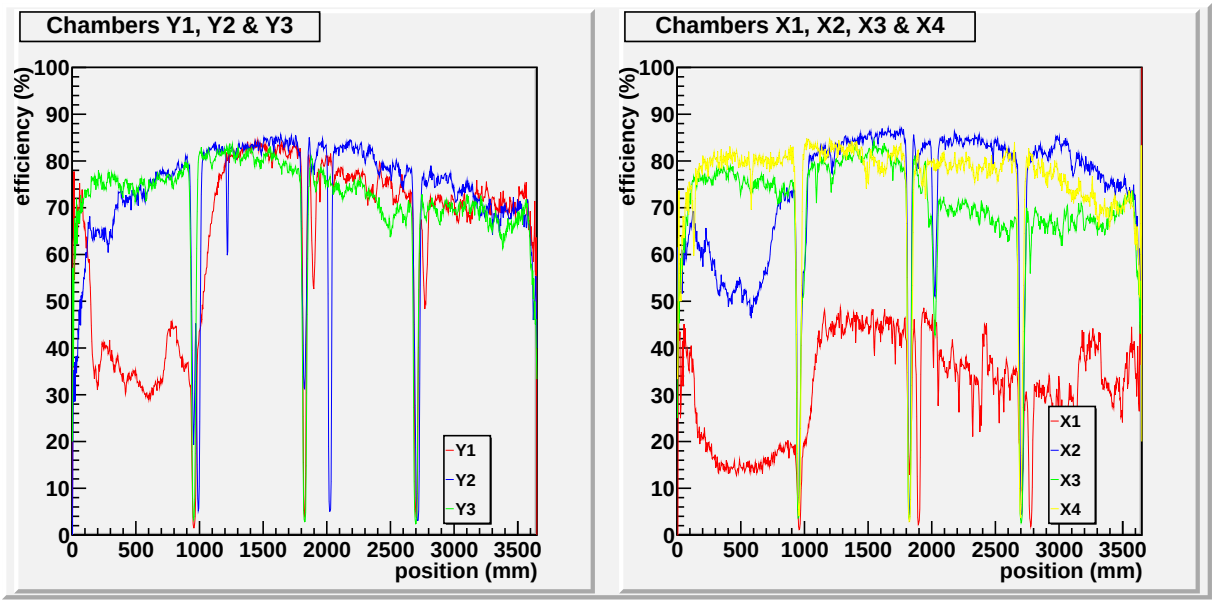


Stack 6, plank 33

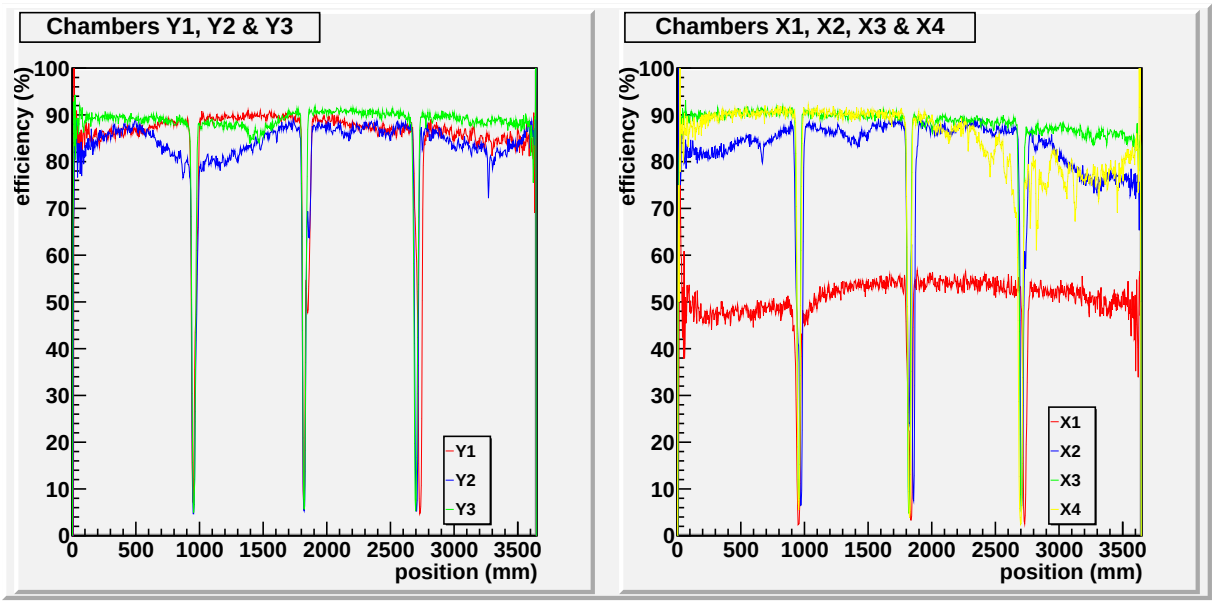


C.3 Efficiency vs. position stack 7

Stack 7, plank 20

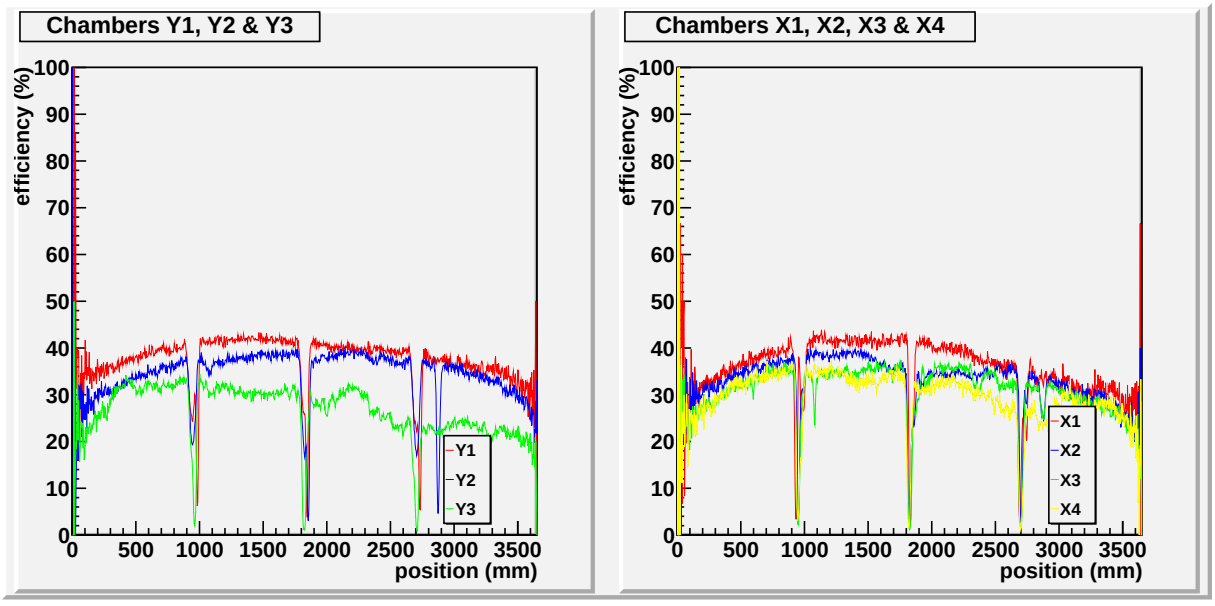


Stack 7, plank 80

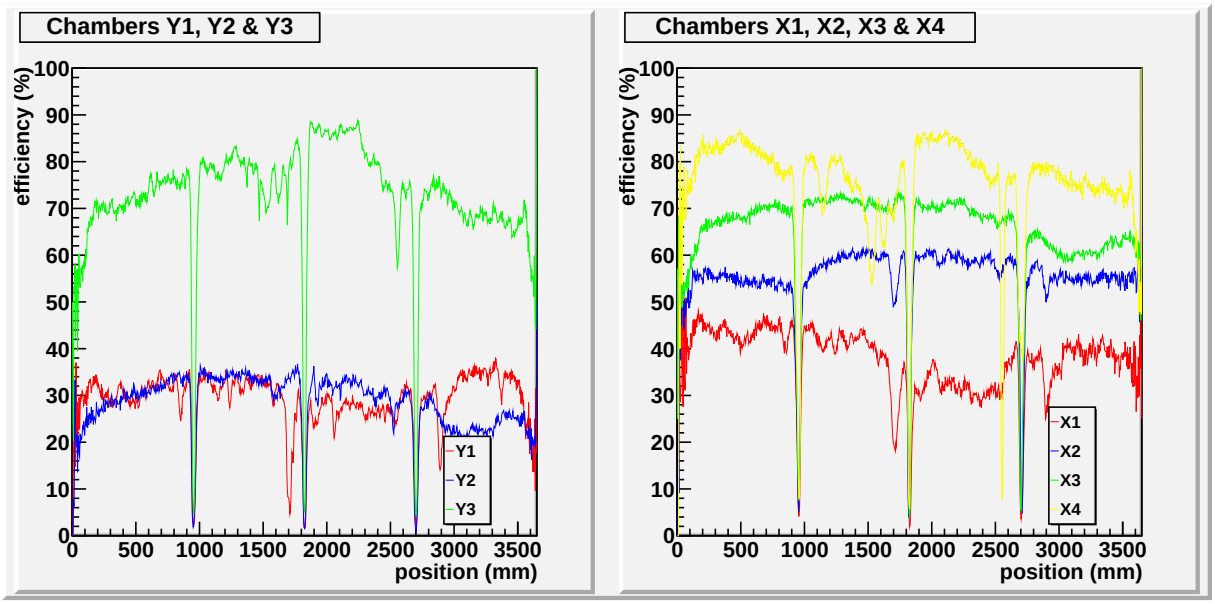


C.4 Efficiency vs. position stack 8

Stack 8, plank 2

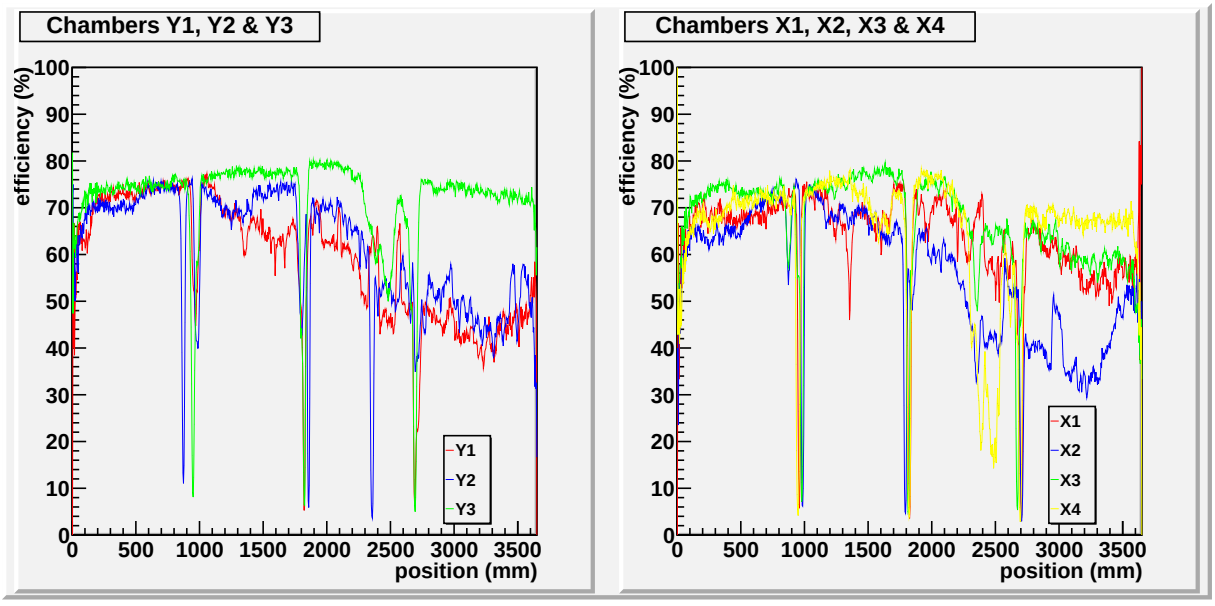


Stack 8, plank 7

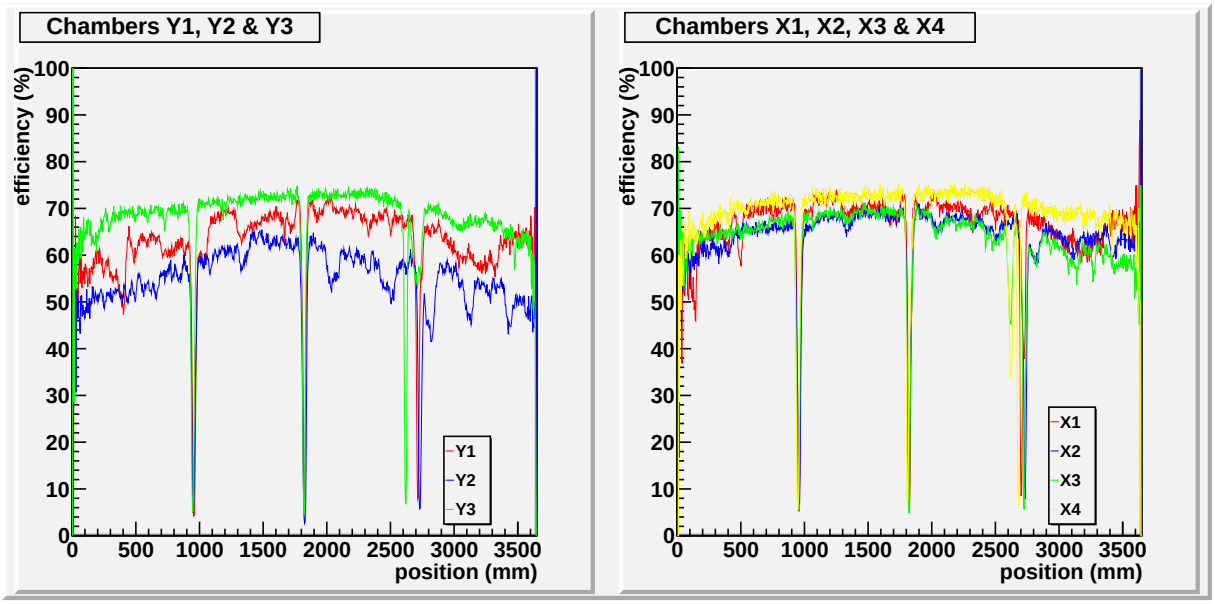


C.5 Efficiency vs. position stack 9

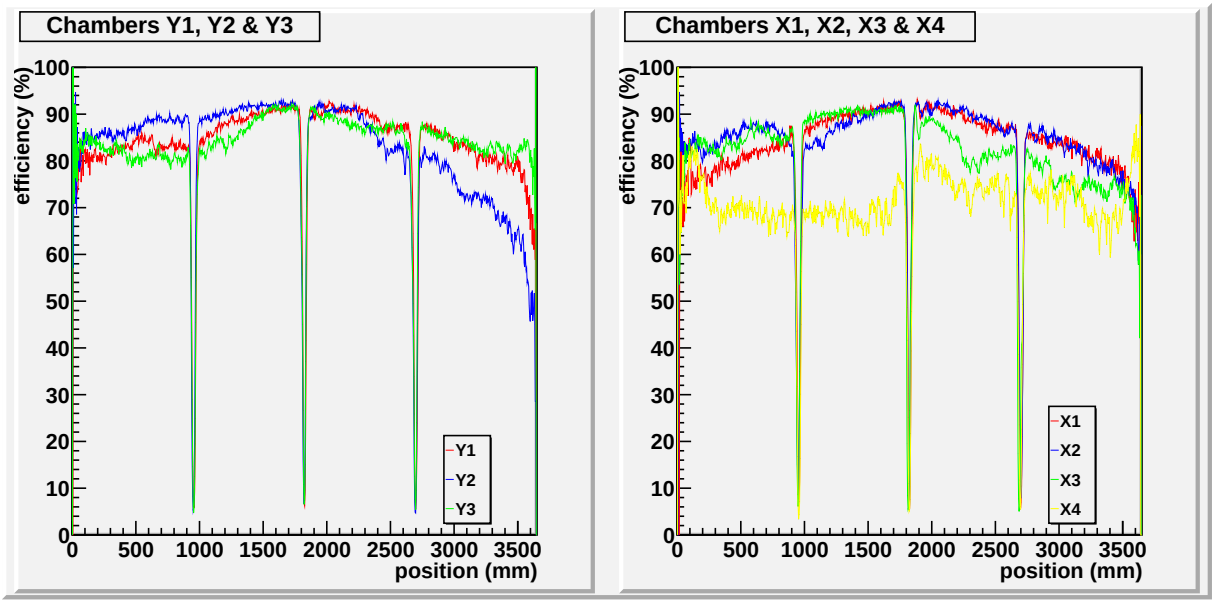
Stack 9, plank 1



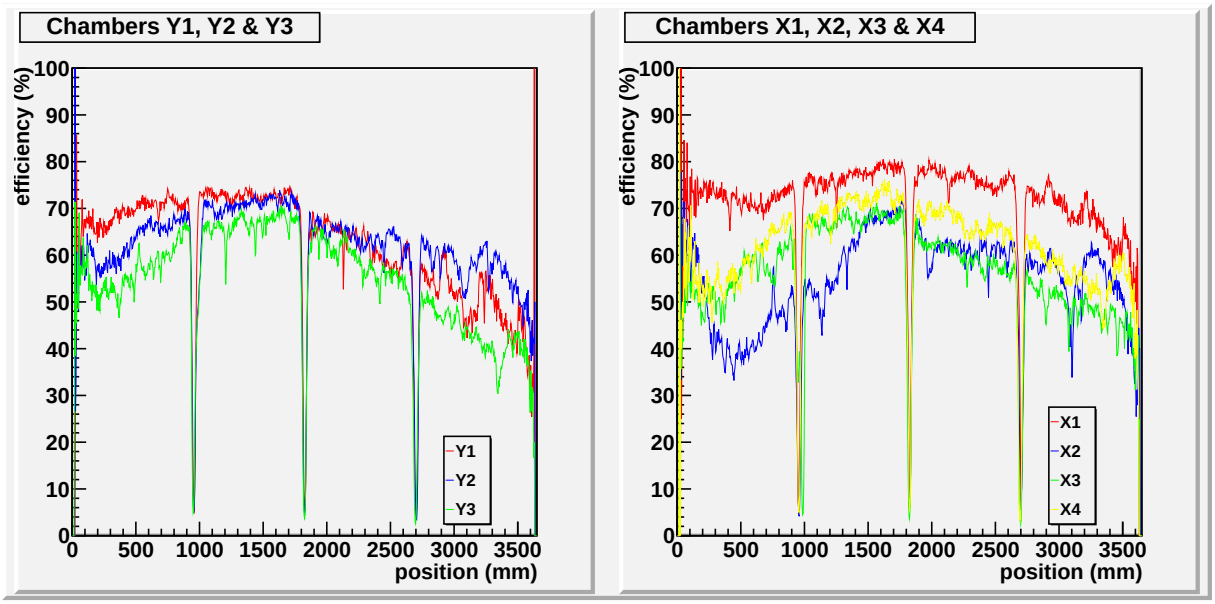
Stack 9, plank 37



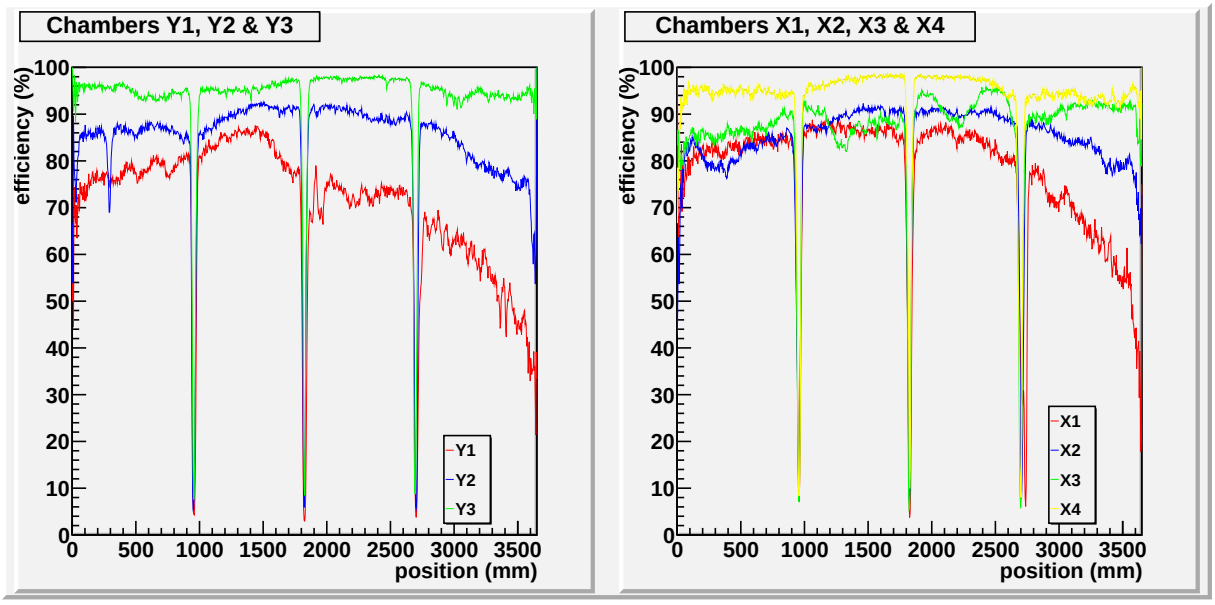
Stack 9, plank 47



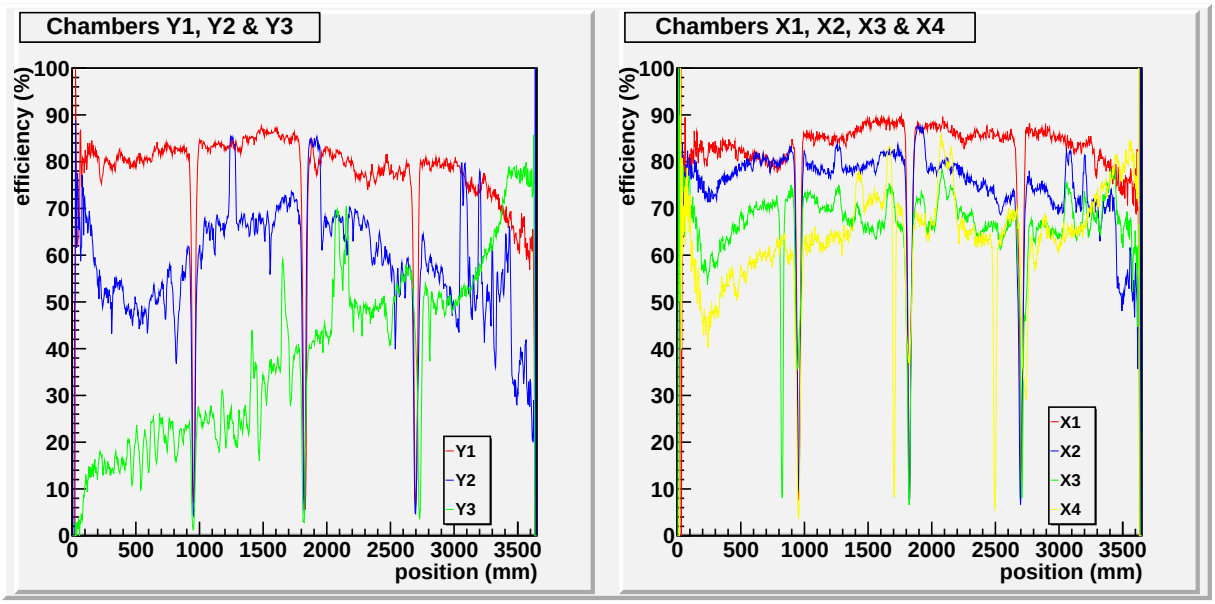
Stack 9, plank 49



Stack 9, plank 51

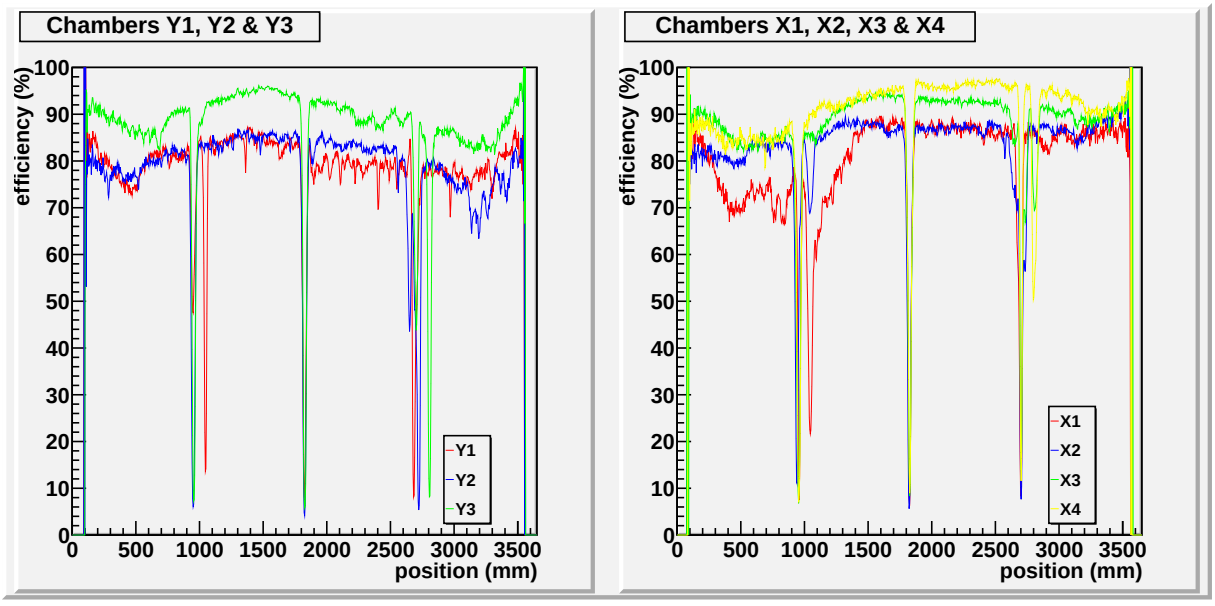


Stack 9, plank 53

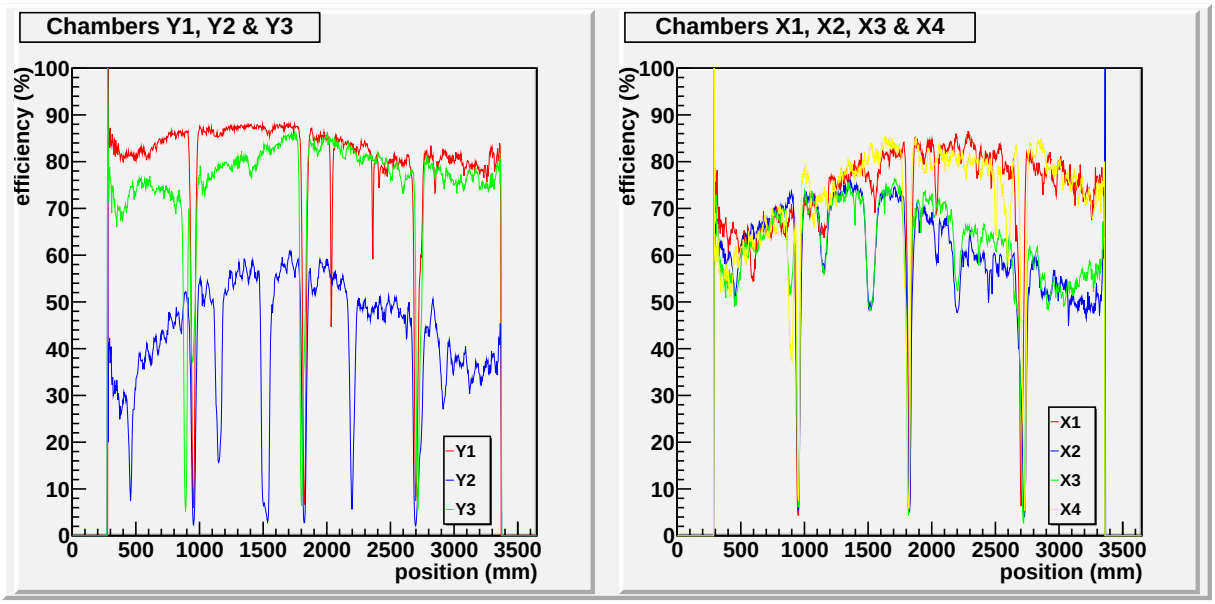


C.6 Efficiency vs. position stack 10

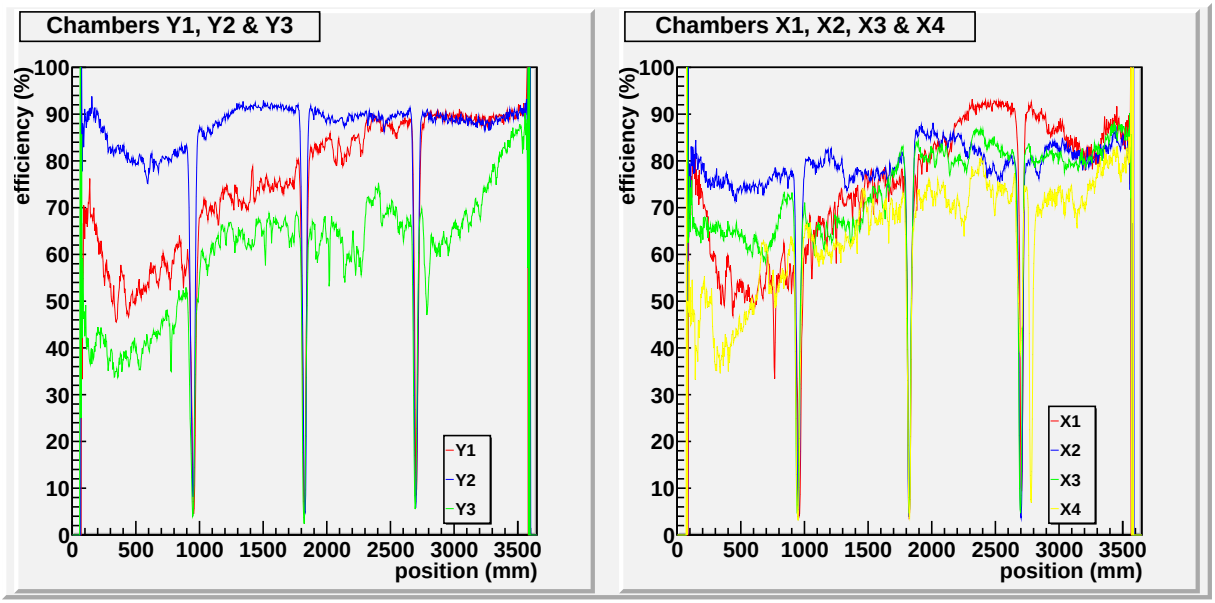
Stack 10, plank 41



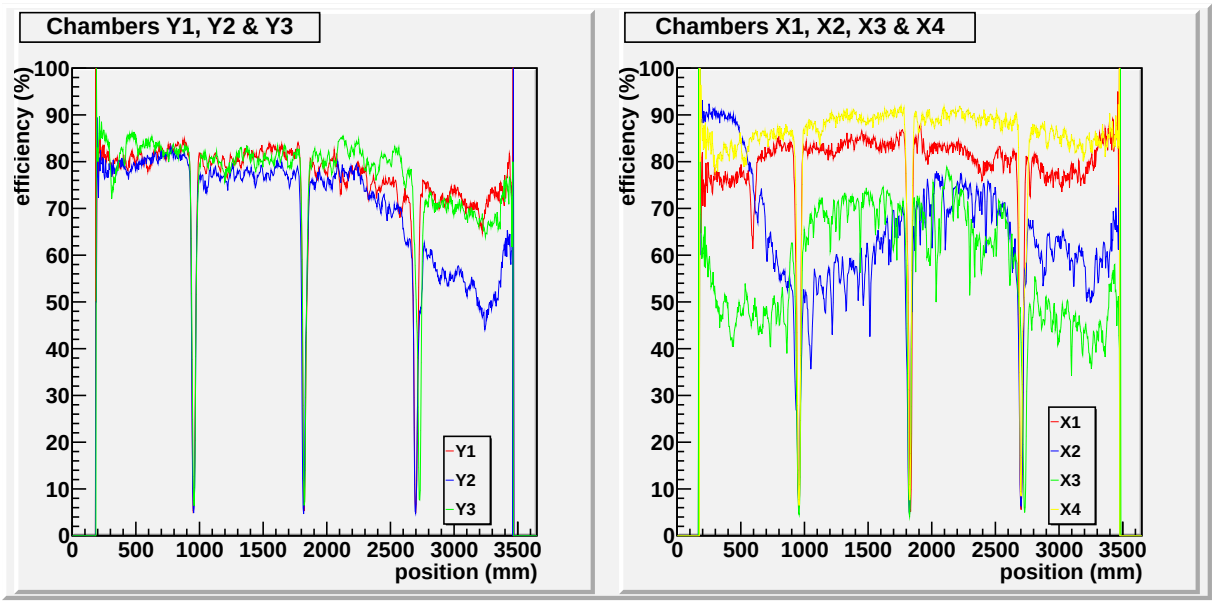
Stack 10, plank 42



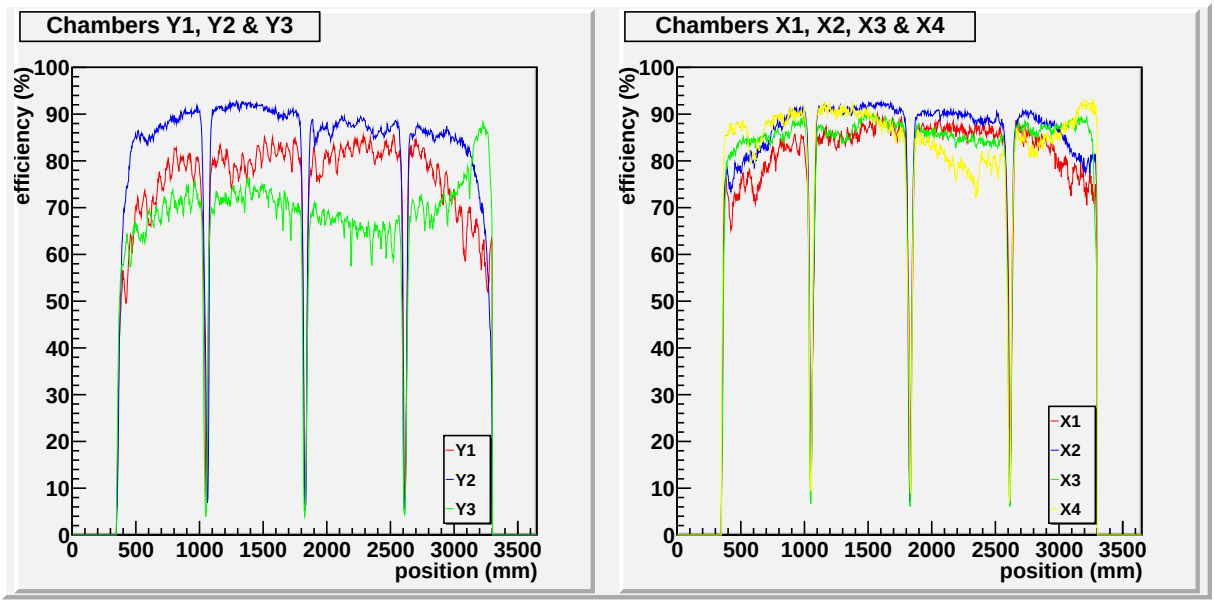
Stack 10, plank 45



Stack 10, plank 58



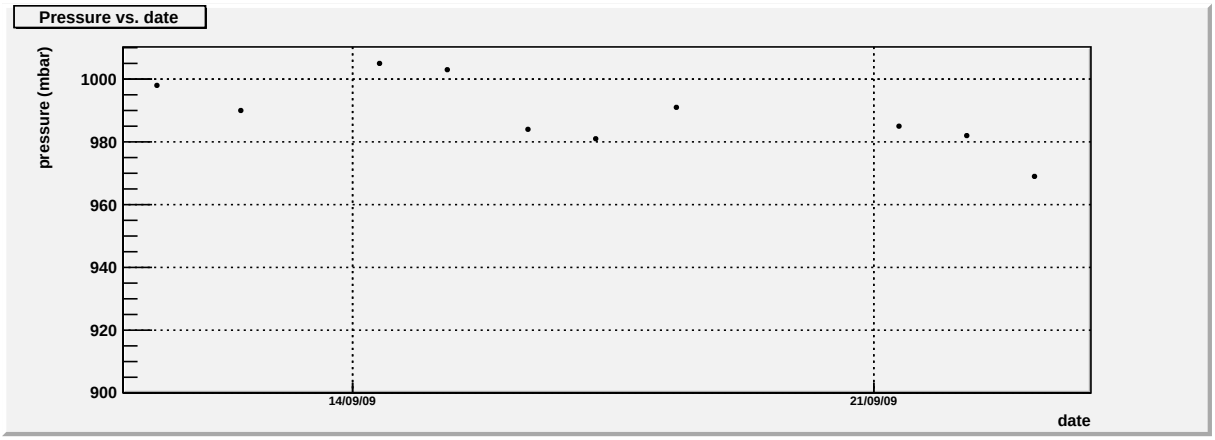
Stack 10, plank 84



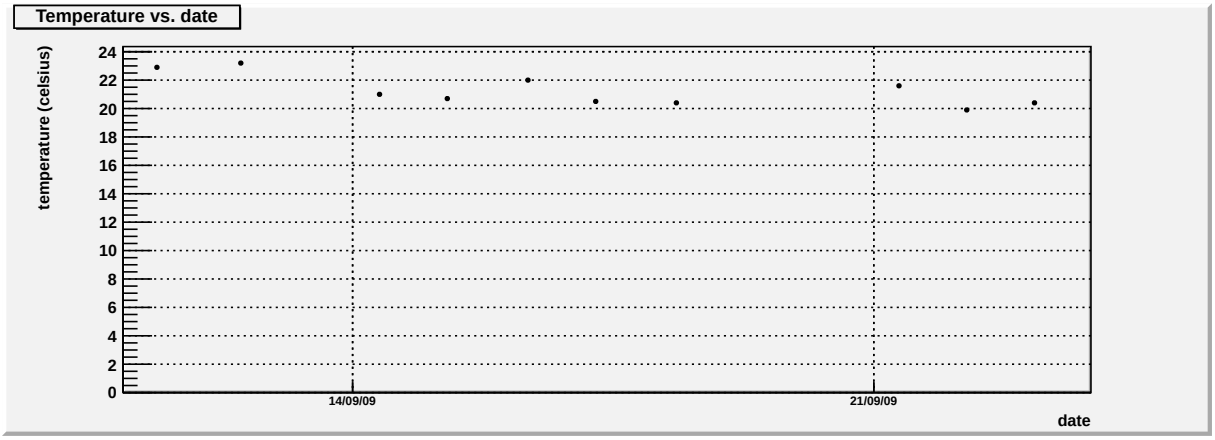
D Pressure and temperature data

Stack 5

Pressure

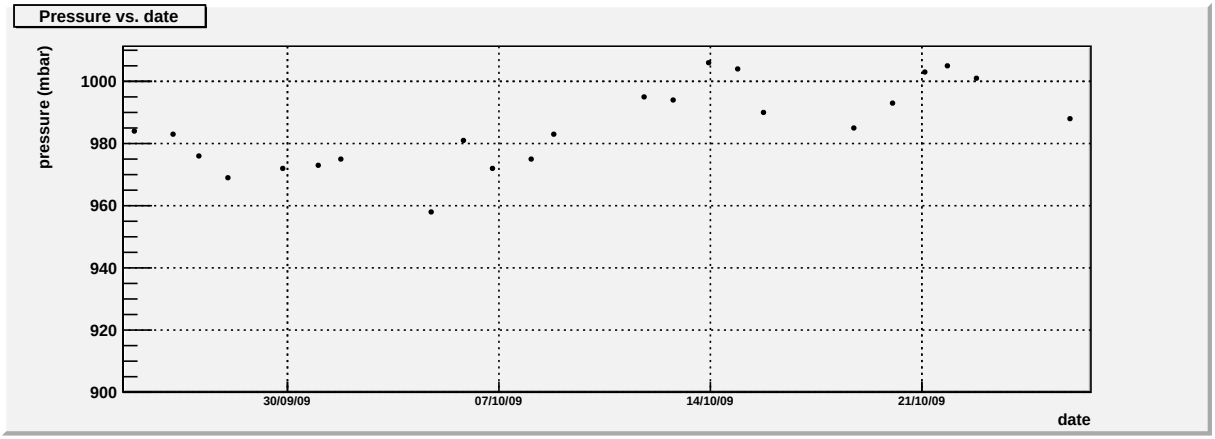


Temperature

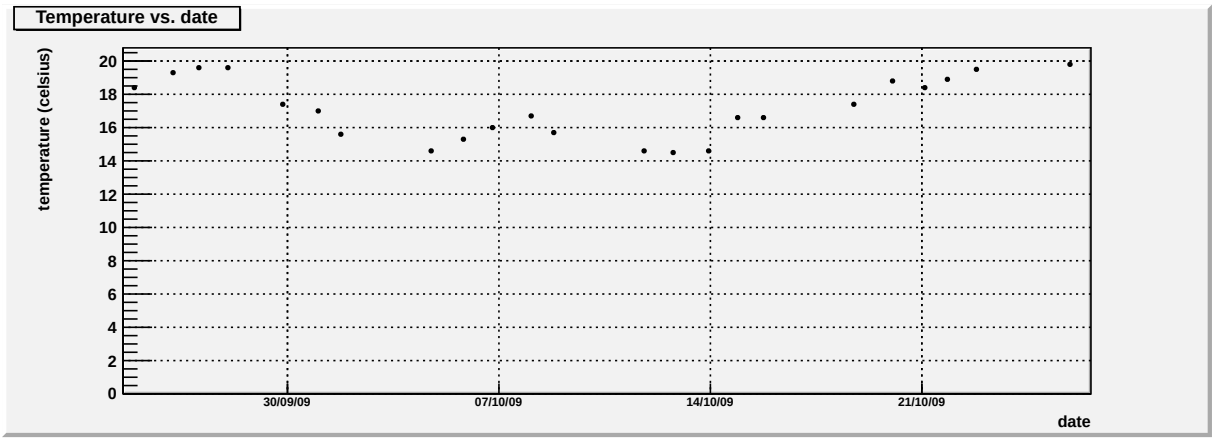


Stack 6

Pressure

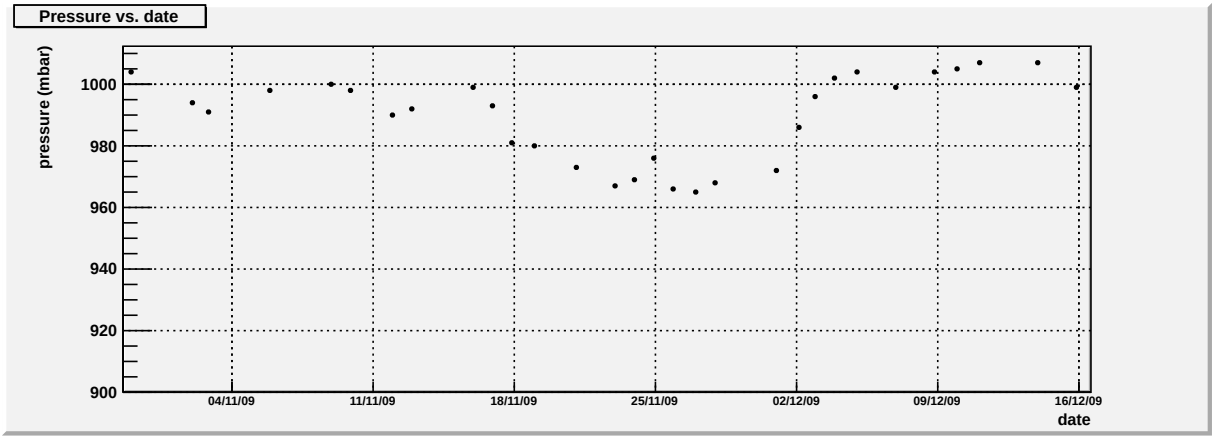


Temperature

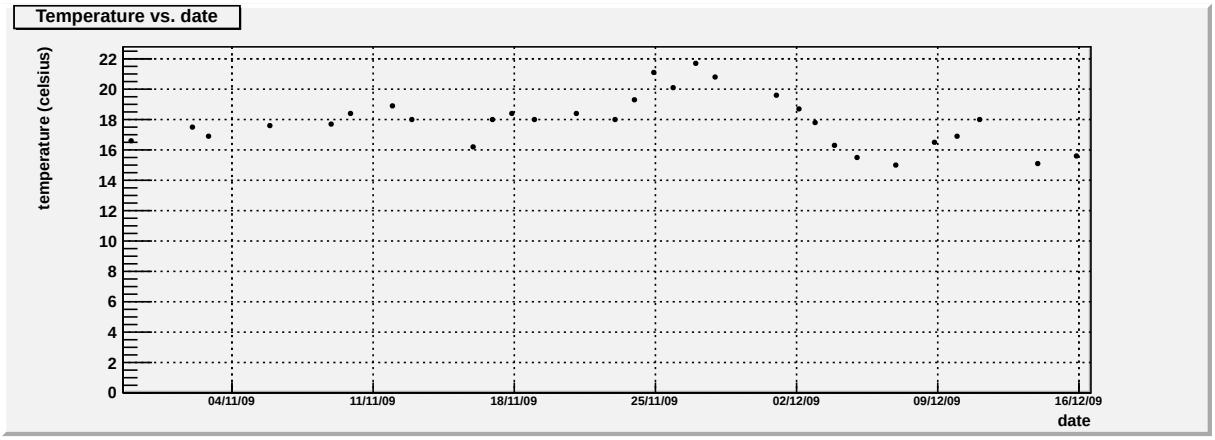


Stack 7

Pressure

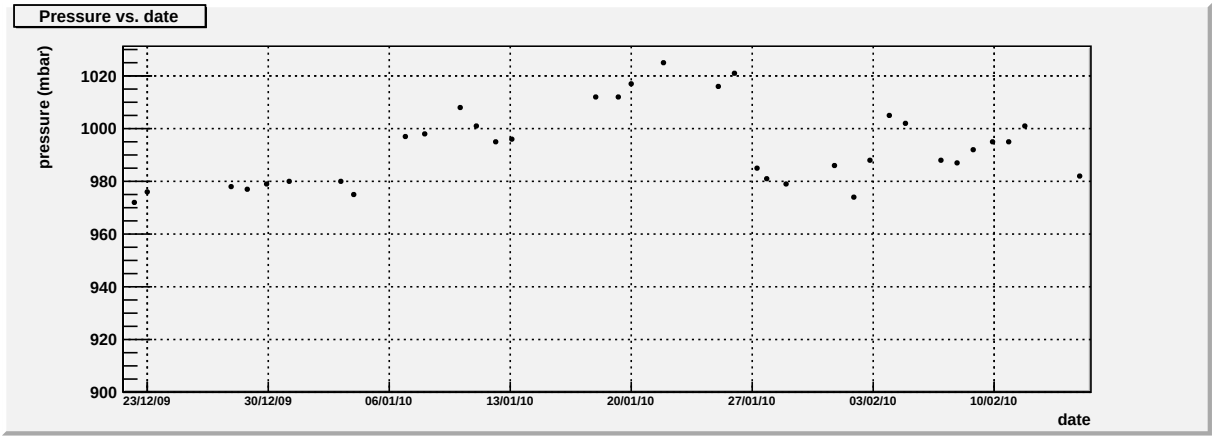


Temperature

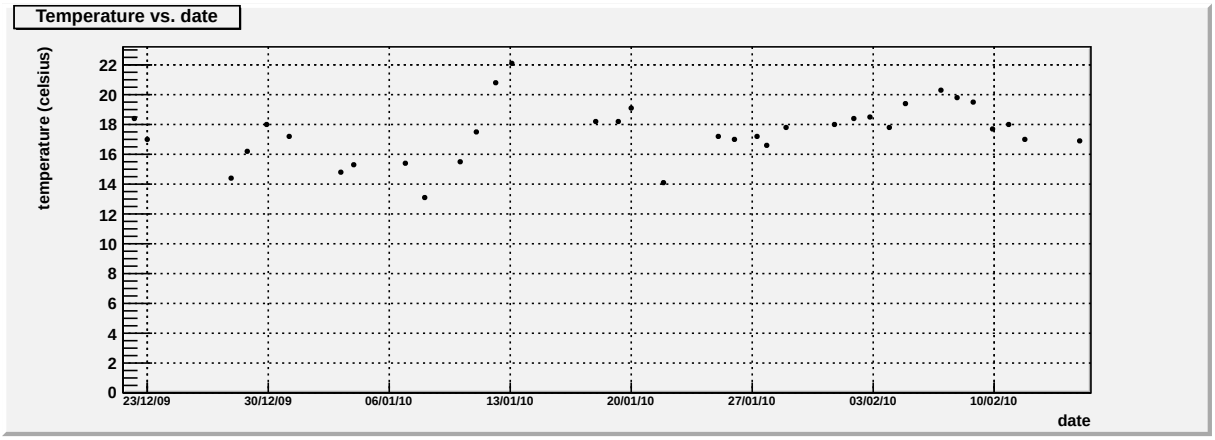


Stack 8

Pressure

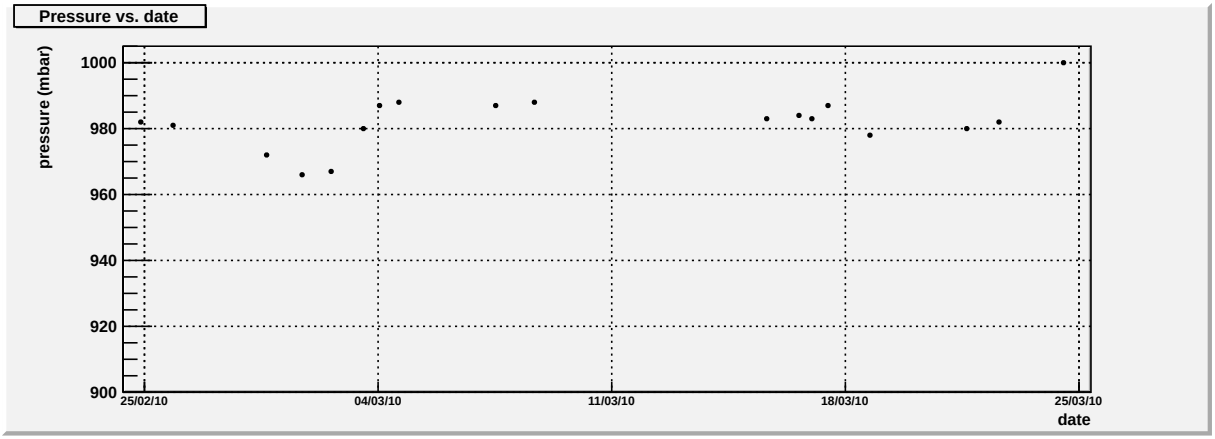


Temperature

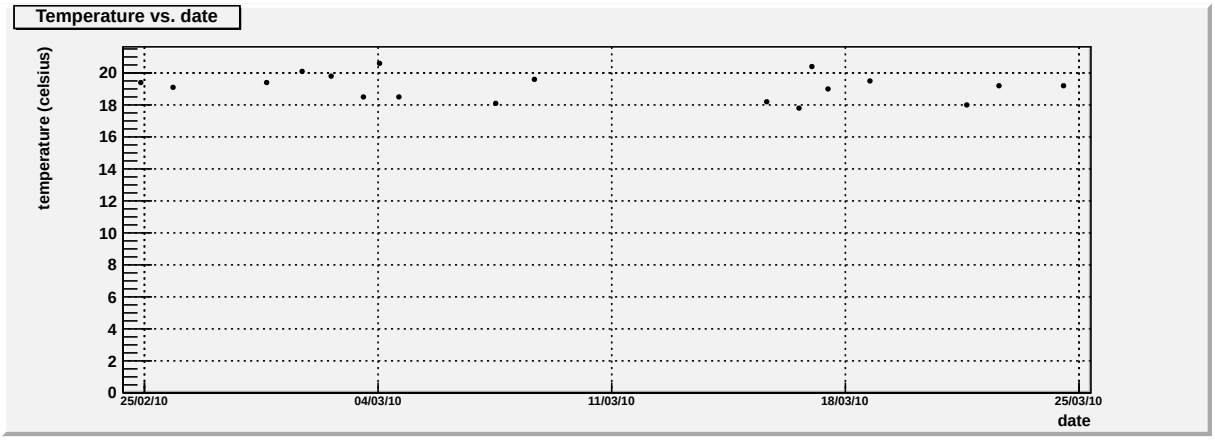


Stack 9

Pressure

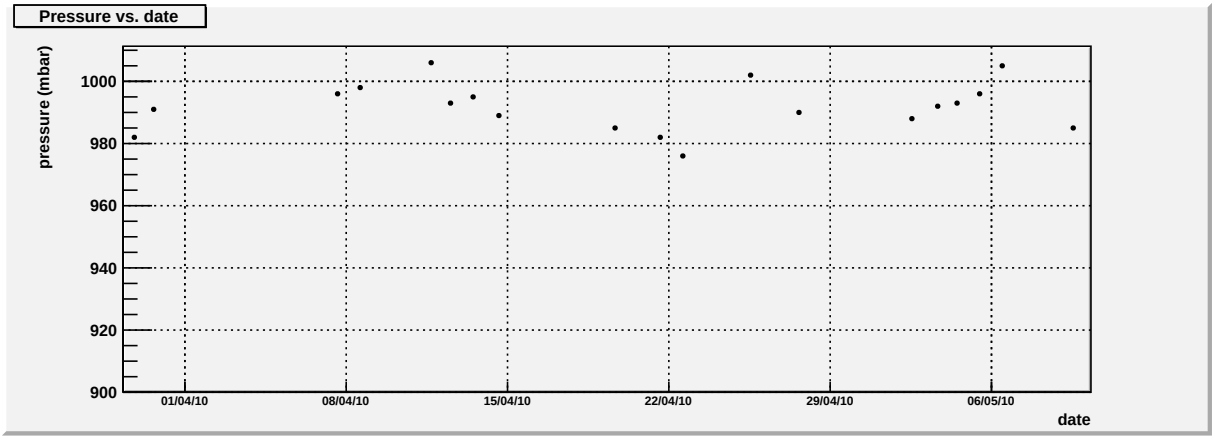


Temperature



Stack 10

Pressure



Temperature