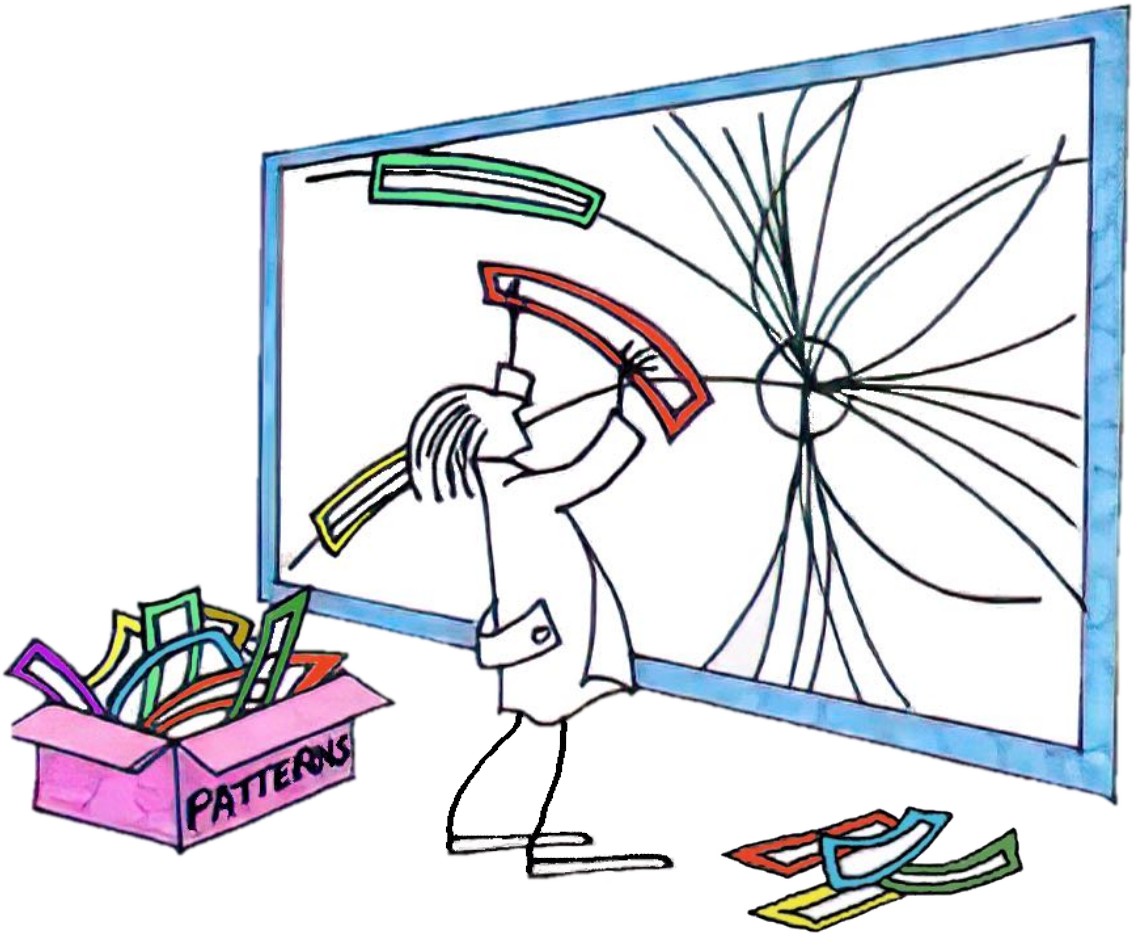
UNIVERSITÀ DEGLI STUDI DI PAVIA
DOTTORATO DI RICERCA IN FISICA – XXXII CICLO

A hardware based tracker for the ATLAS experiment: commissioning and trigger studies

Simone Sottocornola



Tesi per il conseguimento del titolo

Università degli Studi di Pavia
Dipartimento di Fisica

DOTTORATO DI RICERCA IN FISICA – XXXII CICLO

A hardware based tracker for the ATLAS experiment: commissioning and trigger studies

Simone Sottocornola

Submitted to the Graduate School of Physics in partial fulfillment of the requirements for the degree of

DOTTORE DI RICERCA IN FISICA

DOCTOR OF PHILOSOPHY IN PHYSICS

at the

University of Pavia

Supervisor: Prof. Andrea Negri

Cover: Pictorial representation of the pattern matching tracking algorithm, at the base of the FTK processor working principle.

A hardware based tracker for the ATLAS experiment: commissioning and trigger studies

Simone Sottocornola

PhD thesis - University of Pavia
Geneva, Switzerland, January 2020

In memory of my grandpa,
who taught me the beauty
of looking inside of things.

ABSTRACT

During the Run-2 of the Large Hadron Collider (LHC) the instantaneous luminosity exceeded the nominal value of 10^{34} $cm^{-2}s^{-1}$ and the number of overlapping proton-proton interactions per bunch crossing increased up to a mean value of 50. These conditions are expected to become even worse during the future runs of the LHC, posing a challenge to the trigger systems of the experiments that have to manage rates while keeping a good efficiency for interesting physics events.

In order to achieve the required online data reduction in the trigger and data acquisition system, essential to maintain the full discovery potential at such high luminosities and pileup, the LHC experiments need to optimize silicon detector information. As an example, the reconstruction of the track trajectories close to the interaction points can allow to distinguish and subtract the contributions of each pileup collision.

Because of its fine resolution and granularity, tracking information is critical for distinguishing which events, selected by the first trigger level (L1), should be kept for further processing. However, extensive tracking in such environment is prohibitively expensive in terms of processing time per event or computing cores needs. Therefore, full event tracking can only be performed at low rates ($\approx$ few kHz), while the track reconstruction is performed sparingly in specific regions of interest (ROI) which have already been identified as potentially interesting by the L1 trigger. This approach has limitations in several cases. Firstly, there is a limit to either the number or size of ROIs processed by the High Level Trigger (HLT), which forces additional non-tracking cuts to be applied, resulting in reduced efficiency or higher thresholds for the objects considered. Secondly, there are cases where global event information, such as the location of the hard interaction vertex or the number of primary vertices in the event, are useful for object selections or corrections to other detector quantities. Both these problems are particularly critical for the trigger selection of signatures containing third generation fermions, such as τ or b-jets, for which tracking information are fundamental to keep the selection performances high.

In order to facilitate the use of tracks in the HLT, the ATLAS experiment planned the installation of a hardware processor dedicated to online tracking: the Fast Tracker (FTK). It is a hardware based tracking system, currently in its commissioning phase, designed to perform full scan tracking at the full L1 trigger rate (100 kHz), providing recon-

structed tracks to the ATLAS HLT in a mean latency of about 100 μs, adequate for the online trigger selection.

FTK is a very complex system, composed of approximately 450 electronic boards based on two different standards (VME and ATCA). It counts about 10000 links, including internal interconnections and connections to the other ATLAS subsystems. FTK exploits Associative Memory (AM) and Field Reprogrammable Gate Array (FPGA) for the pattern recognition and track fitting procedures, allowing for a huge parallelization of the tracking process. The custom chips and electronic boards are developed by a consortium of 10 institutes. The final system will be composed of 8000 memory chips and 2000 FPGAs.

This thesis work is focused on the FTK project, and especially on its commissioning phase. The first chapter of this thesis is dedicated to an introduction of the LHC and ATLAS experiment. In the second chapter, the tracking problem at the high energy physics experiments is treated, with particular focus on the solutions that the various experiments have developed (or are developing). In the third chapter, an introduction to the FTK system is provided. FTK is described in some details, focusing on the problems that we encountered during the commissioning of the system and the adopted solutions, together with an updated status of the art.

In the part of the thesis describing my personal contributions, three main topics are treated.

In chapter four, the results of a study on the different boards power consumptions is presented, together with a full characterization of the FTK cooling system. This study was particularly critical in order to prove the ability of the FTK housing infrastructure to cope with the very high power dissipation of the FTK boards.

In chapter five, the FTK online software is presented. Particular focus is given to the peculiar requirements that a complex system as FTK pose to the development of the online software infrastructure. The problems we encountered during the development of the software framework, as well as the issues we had to face for integrating FTK into the ATLAS common infrastructure, are presented.

The last chapter is dedicated to a study meant to the creation of a new single-τ trigger chain, able to exploit the FTK characteristics. In particular, a trigger chain, able to increase the signal acceptance for the search of the H^+ charged Higgs boson, predicted by many Beyond Standard Model theories, is presented. This trigger chain exploits the FTK tracks to increase the τ selection efficiency at low p_T values, allowing to increase the signal acceptance for the $H^+ \to \tau\nu$ physics channel.

CONTENTS

1 THE ATLAS DETECTOR AT THE LHC

The Large Hadron Collider (LHC) is a circular particle accelerator designed to collide protons at a collision energy of 14 TeV. The ATLAS experiment is one of the two multi-purpose experiments that takes advantage of the collisions provided by the LHC. It has been conceived to pursuit an ambitious physics program, where the first milestone was the discovery of the Higgs boson, achieved in 2012 [1, 2]. This chapter introduces the accelerator complex of CERN and describes the main aspects of the ATLAS detector at the LHC.

1.1 THE LARGE HADRON COLLIDER COMPLEX

The Large Hadron Collider [3] is the main particle accelerator of the European Organization for Nuclear Research complex (CERN) located near Geneve, Switzerland. LHC is the most powerful proton-proton collider in the world, in which the Standard Model of elementary particle is being tested and in which the new high energy domain could open new windows on high energy physics. It consists in a two-ring semiconducting hadron accelerator and collider, placed in the former LEP tunnel, with a 27 km circumference, lying between 45 m and 175 m underground. Its design allows the acceleration of beams of protons up to an energy of 7 TeV per beam and heavy ions (^{208}Pb) up to an energy of 2.76 TeV per nucleon. In this case the energy per proton remains 7 TeV, but since the nucleus includes neutrons, the average energy goes down. Such energies are essential to achieve the research goals of LHC, which include:

- studies on the origin of spontaneous symmetry breaking and the Higgs boson properties;
- study of the bottom physics, in particular concerning the CP violation and rare decays;
- study of the top physics, which includes a more precise measuremens of the top mass, width and coupling and the search for rare and non standard decays;
- studies about the compositeness of quarks and leptons;
- searches for physics beyond the SM that could appear at the TeV scale, among which searches for SUSY particles and, eventually, the study of their properties;
- serendipity discoveries on unexpected phisycs.

To achieve the LHC goals, and for the full exploitation of its discovery potential, four major experiments are installed around the accelerator ring. The LHC complex, along with the points where the four major LHC experiments are situated, is pictorially represented in Figure 1.1. These experiments, in alphabetical order, are:

- ALICE [4] (**A** **L**arge **I**on **C**ollider **E**xperiment) is a general-purpose detector for heavy-ion collisions designed to address the physics of strongly interacting matter and the quark-gluon plasma at extreme values of energy density and temperature in nucleus-nucleus collisions.
- ATLAS [5] (**A** **T**orroidal **L**HC **A**pparatu**S**) is a general-purpose experiment focusing on high-p_T physics.
- CMS [6] (**C**ompact **M**uon **S**olenoid) is the other general-purpose experiment, with the same goals as ATLAS.
- LHCb [7]: dedicated to b-physics studies, like precision measurements of CP violation and rare B-meson decays.

In addition to the four major experiments, three more experiments are making use of the LHC collisions. The LHCf [8] experiment is meant to simulate cosmic rays in laboratory conditions. The TOTEM [9] (**TOT**al cross section, **E**lastic scattering and diffraction dissociation **M**easurement at the LHC) experiment is dedicated to the precise measurement of the proton-proton interaction cross section, as well as to the in-depth study of the proton structure. The MoEDAL [10] (Monopole & Exotics Detector at the LHC), is designed to search for magnetic monopoles and similar exotic searches such as the Dyon or Stable Massive Particles.

1.1.1 *The accelerator chain*

Before reaching the LHC, the protons are speed-up by a chain of accelerators, shown in Figure 1.2, that gradually increase the particles energy. The proton journey starts in a bottle of pure hydrogen gas, with impurities lower than a part per billion. The molecular hydrogen is break down into atomic hydrogen and the electron is stripped, by means of an intense electric field, leaving a sample of pure protons. They are passed to the Linear Accelerator 2, the first accelerator of the chain, that accelerates them to the energy of 50 MeV by means of radiofrequency cavities. The protons beam leaving the LINAC2 is then injected in the **P**roton **S**ynchrotron **B**ooster. It consists in four superimposed rings placed in a single magnet, which accelerate the beam up to 1.4 GeV of energy. The PSB feeds into the **P**roton **S**ynchrotron, the first synchrotron built at CERN, that further accelerate the protons up to an energy of 25 GeV. The last step before LHC is then the **S**uper

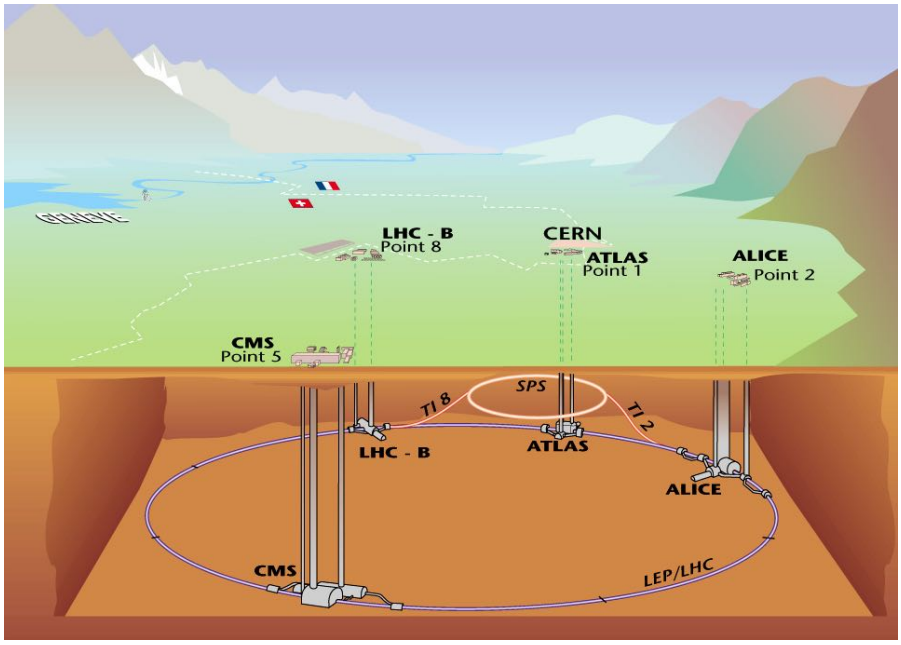


Figure 1.1: Overall view of LHC experiments.

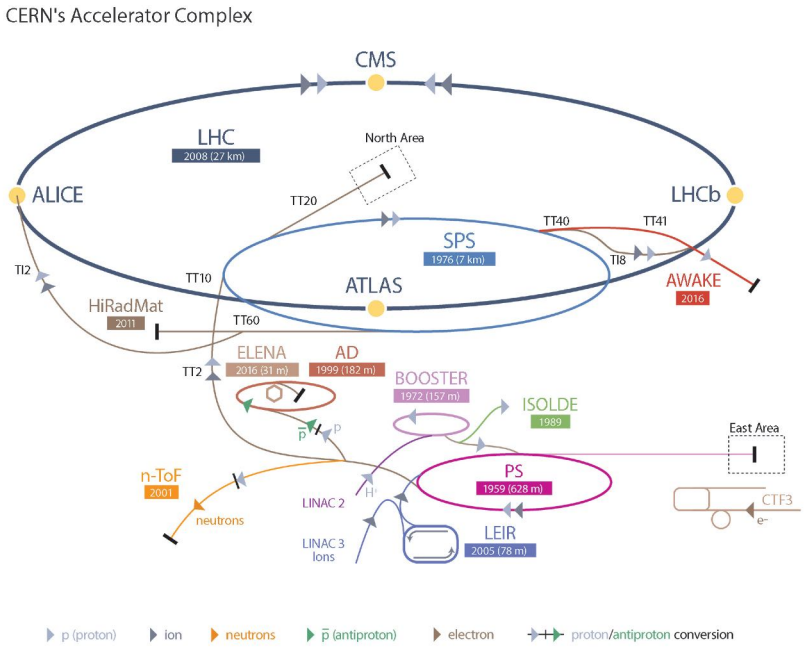


Figure 1.2: The LHC accelerator complex at CERN. The protons are pre-accelerated in the LINAC2, PSB, PS and SPS before their final injection in the LHC ring.

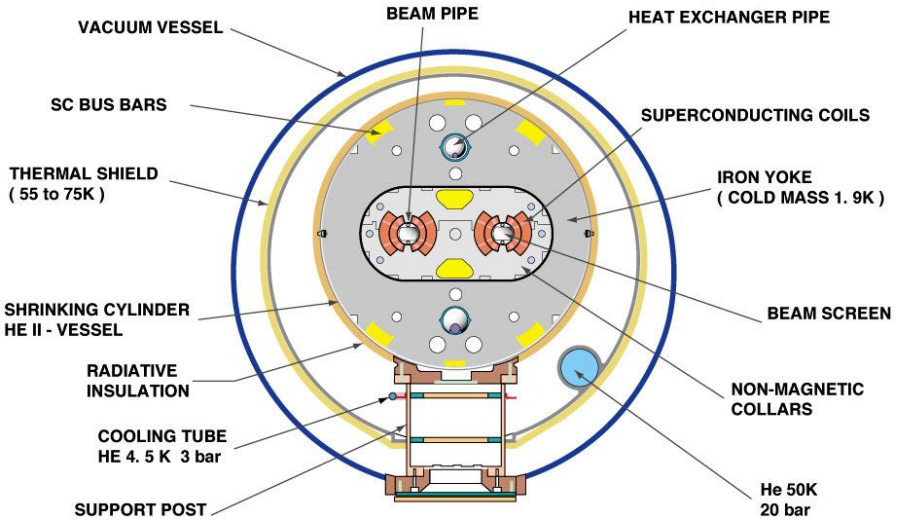


Figure 1.3: Cross-section of an LHC dipole.

Proton **S**ynchrotron. Here the protons are accelerated up to 450 GeV and they are ready to be injected in the LHC accelerator.

1.1.2 *The LHC*

Since LHC is a particle-particle collider, it can not have both beams sharing the same phase space in a single ring, like particle-antiparticle collider have. The construction of a proton-antiproton collider has been excluded, due to the impossibility of creating a high density beam of antiproton with the high efficiency required by the project. For this reason LHC has two rings with counter-rotating beams, which rely on superconducting magnets. The LHC rings are equipped with 1232 dipole, 392 quadrupole, 688 sextupole and 168 octupole superconducting magnets. The dipoles, spread across the whole length of the rings, provide a magnetic field of 8.33 T. To reach a magnetic field of this amplitude, a current of 11 850 A is needed. The wires that form the coils of the magnets are constructed by a niobium-titanium alloy and they are cooled down to their nominal operational temperature of 1.9 K. The coolant used is super-fluid helium, which has the highest thermal conductivity of any known substance. Furthermore, in the superfluid state it is characterised by a complete absence of viscosity, which enables it to penetrate in the complex shape of the coils, making its use as a refrigerant very attractive. Dipoles, the main magnets used, are 14.3 m long and weight 35 tons. Each dipole magnet contains two sets of coils producing opposing magnetic fields, bending the two beams of particles with same charge in opposite directions, while simultaneously focusing them. Figure 1.3 shows a sketch of the cross section of a dipole.

The LHC magnet system has a stored electromagnetic energy of approximately 600 MJ, and, considering also the total beam current of

0.548 A, yield a total stored energy of about 1 GJ. This stored energy must be absorbed safely at the end of each run, or in case of a malfunction or emergency. The beam dumping system and the magnet system provide additional limits for the maximum attainable beam energies and intensities.

The acceleration of the particles at LHC is carried out by 8 radio-frequency (RF) superconducting resonating cavities per each beam, able to deliver 2 MV, with a $5\,MVm^{-1}$ accelerating field, at a frequency of about 400 MHz. This set-up provides 35 640 potential wells in the RF field, called RF-buckets, that corresponds to a spacing in time for the bunch of 2.5 ns. However only one tenth of the buckets is used, making the minimum bunch spacing to be $\Delta t = 25\,ns$. These RF cavities can supply the protons, during the ramp phase from 450 GeV to 7 TeV, with about 485 keV per revolution. Furthermore, they are used to stabilize the bunches in buckets and replenish the energy lost due to synchrotron radiation.

By design, each LHC beam consist of 2808 bunches, each containing $1.15 \cdot 10^{11}$ protons. The bunches travel at approximately the speed of light, with a rotating frequency equal to 11 245 kHz and a period of 9×10^{-5} s. The spatial distance between two consecutive bunches is 7.5 m. The bunches are grouped in trains and, as said before, out of the about 3600 slots available for bunches around the LHC, only 2808 are occupied at most; the empty ones are useful too, e.g. the detectors can study the occupancy due to noise when no collisions happen.

The collisions between the particles occur at the four interaction points where the experiments are located. To ensure that the collisions do not happen with a big spread around the nominal interaction points, the two beams are designed to collide with a sufficiently large crossing angle. The beams are squeezed as much as possible at the collision point in order to increase the chances of a collision, as shown in Figure 1.4. The protons that do not interact continue to circulate around the LHC ring. Over time the beams degrade, due to collisions and interactions of the particles with the beam pipes leading to beams with less protons and thus less likely to interact in a collision that yields interesting physics. Therefore, after approximately 15 hours the beams are dumped in a radiation-shielded block deep underground and the LHC ring is refilled. A fill requires 4 minutes and 20 seconds for each ring and the protons reach their maximum energy after 20 minutes.

1.1.3 *Center of mass energy*

One of the most important parameters describing a particle accelerator is the center of mass energy of the collision it provides. The maximum proton momentum obtainable in a proton-proton collider, where the

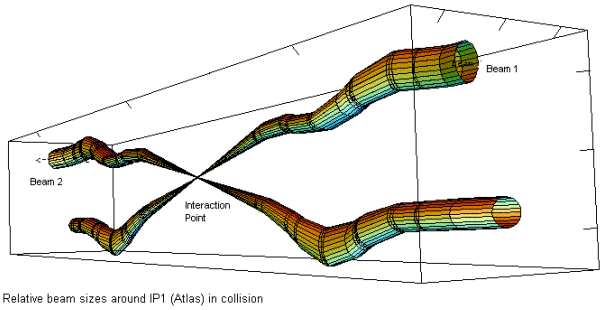


Figure 1.4: Intersection of the two LHC beams. The accelerated beams have transverse dimensions of the order of a few mm, and when they approach the collision point, they get squeezed to approximately 16 µm.

energy loss due to synchrotron radiation is negligible, is constrained, for a given radius, by the maximum magnetic field attainable:

$$p(TeV) = 0.3B(T)R(km) \tag{1.1}$$

The power loss by the emission of synchrotron radiation, i.e. the electromagnetic radiation emitted by charged particles when accelerated on a curved trajectory, can be expressed by:

$$P = \frac{e^2 c}{6\pi\epsilon_0 R^2}\gamma^4 = \frac{e^2 c}{6\pi\epsilon_0 R^2}\frac{E^4}{(mc^2)^4} \tag{1.2}$$

where ϵ_0 is the vacuum permittivity, γ is the relativistic gamma factor, R is the radius of the circle, E, m and e are, respectively, energy, mass and charge of the particle. Due to the quartic dependence of the synchrotron radiation on the mass of the charged particle that has to be accelerated, the choice of a proton collider instead of an electron one (where the proton mass is 2000 times the electron one), ease the reaching of the high energies required to study the physics of interest. However, since protons are composite particles, the amount of the parton energy is only a fraction of the total proton energy. The centre of mass energy, $\sqrt{s'}$, available to create new particles is therefore less than the proton beam centre-of-mass energy $\sqrt{s}$, and can be calculated by the equation:

$$\sqrt{s'} = \sqrt{sx_1x_2} \tag{1.3}$$

where x_1 and x_2 are the energy fractions carried by the two colliding partons. These values are governed by the **P**arton **D**istribution **F**unction (PDF), i.e. the probability of finding a parton with a fraction x of the proton momentum, that is not uniquely determined. The PDF is extracted from deep inelastic data according to different theoretical assumptions. The laboratory system, which in a collider is the centre of mass of the beams system, is thus different from the parton one, which gets a longitudinal boost in the direction of the beam axis depending on the x_1 and x_2 values.

In the proton collisions we can distinguish between two different types of interactions. The collisions between the two protons occurring at "large distances", i.e. when the two protons interact as elementary particles, are called *soft* collisions. In these collisions the particles in the final states have low transverse momentum, $< p_t > \approx 500\,MeV$, and typically are events of little interest. Much more interesting are the collisions at "small distances", i.e. the head-on collisions between partons, that are called *hard* collisions. In this case the particles in the final states can have large transverse momentum p_T, which facilitates their identification.

1.1.4 *Luminosity*

Along with the centre of mass energy, one of the most important parameters of a collider is the instantaneous luminosity ($\mathcal{L}$). Measured in $cm^{-2}\,s^{-1}$, it relates the interaction rate, $\delta R/\delta t$, to the cross-section, σ, of the process through the formula

$$\frac{\delta R}{\delta t} = \mathcal{L} \cdot \sigma \tag{1.4}$$

Given a process with a known cross-section, the event rate for this interaction can be estimated with the formula above. Alternatively, if the luminosity is known, a cross-section can be measured using the observed event rate. The instantaneous luminosity expresses the number of beam particle pairs passing by each other per unit area and per unit time. It can be calculated as shown in Equation 1.5 from the number of beam particles in each of the two colliding bunches, n_1 and n_2 respectively; from the number of bunches, n_b, in the accelerator and their revolution frequency, f_r; and from the transverse overlap of the colliding bunches, $\Sigma_x \cdot \Sigma_y$.

$$\mathcal{L} = \frac{n_b f_r n_1 n_2}{2\pi \Sigma_x \Sigma_y} \tag{1.5}$$

Integrating the instantaneous luminosity over the accelerator active time the integrated luminosity is obtained, relating the total number of produced events, N_{tot} to the cross-section:

$$N_{tot} = \sigma \cdot \int \mathcal{L}\,dt \tag{1.6}$$

1.1.5 *LHC operating parameters*

As previously mentioned, one of the main goal of the LHC physics programme is the study of extremely rare processes. The Higgs Boson is the most classical example, with its production obviously mandatory in order to study in detail its physical properties.

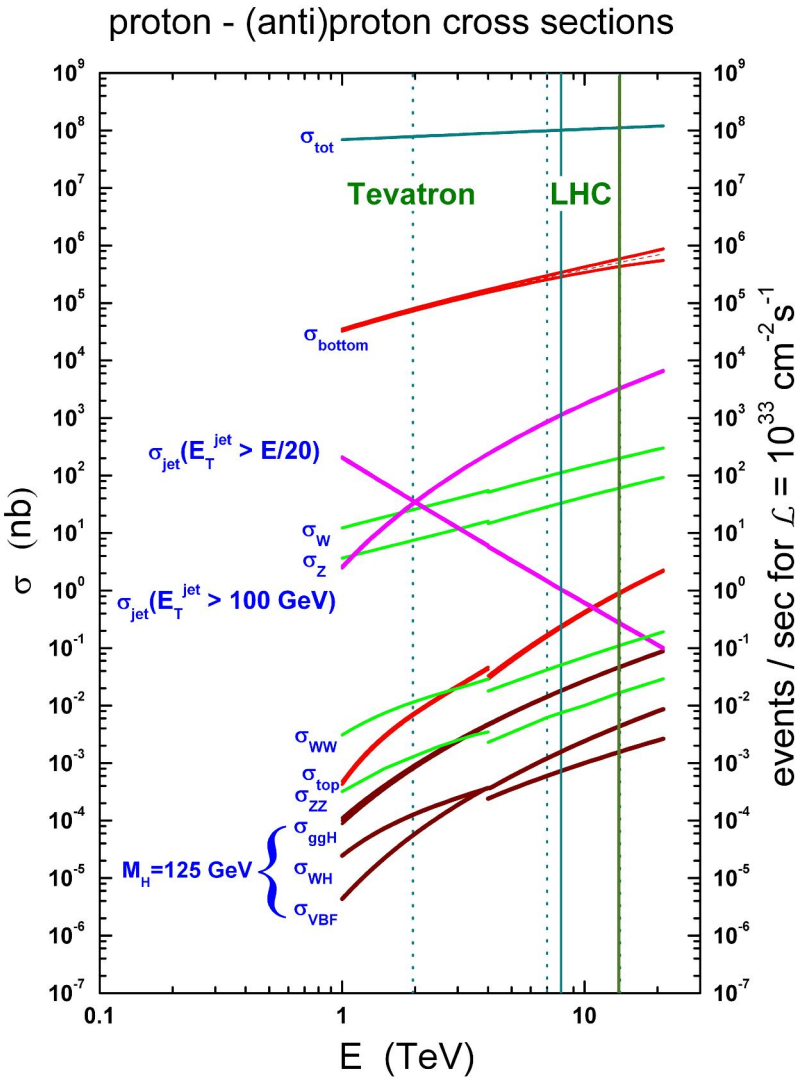


Figure 1.5: Various Standard Model cross-sections at the Tevatron and LHC colliders as function of the centre-of-mass energy ($\sqrt{s}$).

	2010 (peak)	2012 (peak)	2016 (peak)	2018 (peak)	Design
Particle energy E(TeV)	3.5	3.5	6.5	6.5	7
Particles per bunch $n_{1,2}$	1.2×10^{11}	1.6×10^{11}	1.2×10^{11}	1.15×10^{11}	1.15×10^{11}
Number of bunches N_b	348	1331	2220	2556	2808
Bunch spacing (ns)	50	50	25	25	25
Luminosity $\mathcal{L}(cm^{-2}s^{-1})$	2×10^{32}	5.4×10^{33}	14×10^{33}	21.0×10^{33}	10^{34}
pile-up(μ)	3.8	27	45	70	25

Table 1.1: Evolution of the LHC operating parameters at the ATLAS interaction point.

In Figure 1.5 the cross sections, i.e. the probability for a given reaction to occur, of various processes at a hadron collider are shown. The rate of a reaction, i.e. the mean number of events per second, is defined by the equation

$$R = \sigma\mathcal{L} \tag{1.7}$$

where σ is the reaction cross section and $\mathcal{L}$ is the luminosity of the accelerator, which can be calculated from the machine parameters by the equation 1.5, as has been shown in section 1.1.4. The number of events of a given reaction is then $N = \mathcal{L}\Delta t\sigma$, where $\mathcal{L}\Delta t$ is the integrated luminosity, with Δt the time interval of data collection. From Figure 1.5 is clearly visible the rarity of such interesting processes and the importance of the luminosity in high energy experiments. As an example, at the peak luminosity of $\mathcal{L} = 1 \times 10^{34}\,cm^{-2}s^{-1}$, that represent the LHC design luminosity, a Higgs boson event would occur with an average frequency of $\approx 0.25\,Hz$.

Another phenomenon needs to be taken into account: the *pile-up*. It represents the number of superimposed hard collisions occurring in a single bunch crossing. The average number of superimposed events, μ, can be estimated as

$$\mu = \mathcal{L}\sigma_{pp}\overline{\Delta t} \tag{1.8}$$

where $\overline{\Delta t}$ is the average bunch spacing. More than one superimposed event in a single bunch crossing leads to additional complications in the analysis of the events.

In Table 1.1 some of the operational LHC parameters of the past runs at the ATLAS interaction points are shown.

Protons are not the only particles accelerated in the LHC. Lead ions for the LHC start from a source of vaporised lead and enter Linac 3 before being collected and accelerated in the Low Energy Ion Ring (LEIR). They then follow the same route to maximum energy as the protons.

1.2 THE ATLAS DETECTOR

ATLAS is a general purpose experiment aimed at the exploration of a vast range of physics scenarios. It is designed to measure the

particles produced in pp collisions at the LHC at the highest energy and luminosity ever reached with a particle accelerator. ATLAS is the biggest LHC detector, with a diameter of 25 m and a length of 46 m weighting over 7000 t and covering almost 4π of solid angle. ATLAS also features two magnetic systems: a solenoid in the center and a toroid in the external region. More details about these systems will be provided in Section 1.2.2.

The ATLAS design has been optimized to cover all the main research goals of LHC, already presented in Section 1.1. All the benchmark physics goals can be turned into a set of general requirements for the detectors, that can be summarized as follow:

- efficient tracking at high luminosity for high p_T lepton momentum measurements, electron and photon identification, τ lepton and heavy flavour identification, and full event reconstruction capability at lower luminosity;
- very good electromagnetic calorimetry for electron and photon identification and measurements, completed by full-coverage hadronic calorimetry for accurate jet and missing transverse energy E_T^{miss} measurements;
- high precision muon momentum measurements, with the capability to guarantee accurate measurements at the highest luminosity using the external muon spectrometer alone;
- large acceptance in pseudorapidity (η) with almost full azimuthal angle (ϕ) coverage everywhere.

The resulting layout is an "onion" shaped detector, with the different subsystems placed surrounding the interaction point. Figure 1.6 shows a sketch of the detector layout, including its various subsystems. Table 1.2 summarises the coverage and resolution of the subdetectors ATLAS consists of.

1.2.1 *Coordinate system*

The ATLAS reference system is a Cartesian right-handed coordinate system with its origin placed at the nominal interaction point (IP) in the centre of the detector. The x axis points towards the centre of the LHC ring, the y axis points upwards towards the surface, and the z axis points to the anti-clockwise beam direction. The y axis is slightly tilted with respect to the vertical direction due to the general tilt of the tunnel. The global coordinate system is shown in Figure 1.7. The azimuthal angle ϕ is measured around the beam axis, and ranges from $-\pi$ to $+\pi$ with respect to the x axis. The polar angle θ is measured with respect to the z axis, ranging from 0 to π.

Given that the colliding partons have unknown momenta in each collision, the total momenta of the parton couple along the z axis is

Detector	Subdetector	Location	Resolution	Coverage (η)
Inner Detector	IBL	Barrel	$\sigma p_T/p_T = 0.05\% \oplus 1\%$	± 3
	Pixel	Barrel		± 2.5
		Endcap		$1.7 - 2.5$
	SCT	Barrel		± 1.4
		Endcap		$1.4 - 2.5$
	TRT	Barrel		± 0.7
		Endcap		$0.7 - 2.5$
E/M Calorimeter	LAr	Barrel	$\sigma_E/E = 10\%/\sqrt{E} \oplus 0.7\%$	± 1.475
	LAr	Endcap		$1.375 - 3.2$
Hadronic Calorimeter	TileCal	Barrel	$\sigma_E/E = 50\%/\sqrt{E} \oplus 3\%$	± 1.7
	HEC	Endcap		$1.5 - 3.2$
	FCal	Endcap	$\sigma_E/E = 100\%/\sqrt{E} \oplus 10\%$	$3.1 - 4.9$
Muon Spectrometer	MDT	Barrel	$\sigma_{p_T}/p_T = 10\%$ at $p_T = 1$ TeV	± 2.7
	CSC	Endcap		$2.0 - 2.7$
	RPC	Barrel		± 1.05
	TGC	Endcap		$1.05 - 1.92$

Table 1.2: Design resolution of the ATLAS subdetectors and their geometrical (η) coverage.

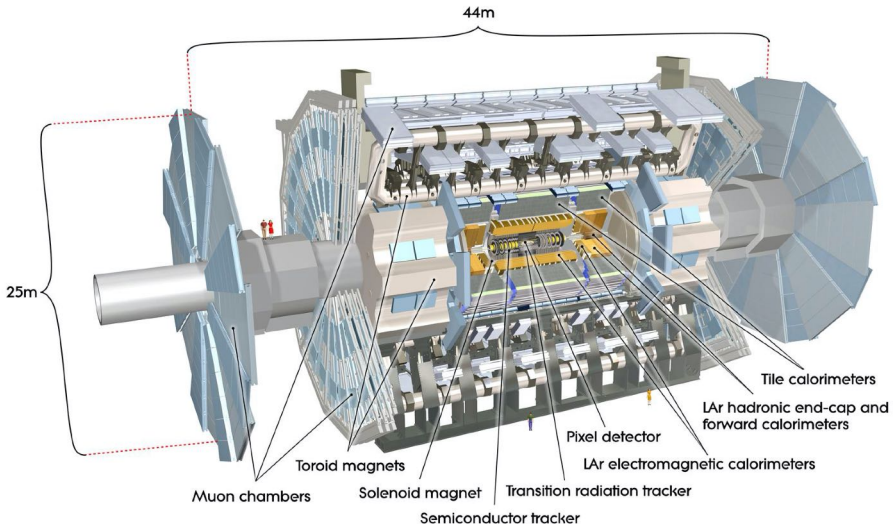


Figure 1.6: View of the ATLAS detector, showing the different subdetectors and layers around the proton beam axis. The collisions occur in the centre of the detector. The main detector components are indicated.

not known; thus, conservation of momentum between the initial and final states can not be applied along the z axis. However, momentum and energy are conserved into the transverse plane, defined as the projection on the xy plane, and are boost-invariant along the z axis, where the centre-of-mass of the colliding partons moves:

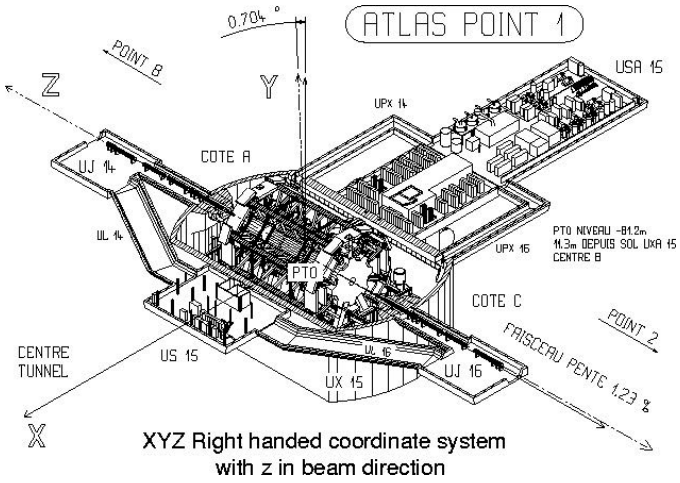


Figure 1.7: The coordinate system of the ATLAS detector.

$$\begin{aligned} E_T &= E \sin\theta \\ p_T &= p \sin\theta \end{aligned} \tag{1.9}$$

A variable commonly used at particle accelerators to describe the polar distribution of particles is the rapidity, y:

$$y = \frac{1}{2} \ln\left(\frac{E + p_z}{E - p_z}\right) \tag{1.10}$$

where p_z is the z component of the momentum of the particle with energy E. For a given boost along the z axis, all particles see their rapidity changed by the same offset. Therefore, rapidity differences are invariant to such boosts, and thus independent of the event-by-event movement of the partonic centre-of-mass. Since the azimuthal angle ϕ is also invariant with respect to these boosts, the angle of emission of a particle from an interaction point is often given as the coordinate pair (y, ϕ). Consequently, the angular separation of two particles, $(y_2 - y_1, \phi_2 - \phi_1)$, is invariant with respect to boosts along the beam axis.

In the relativistic limit, and assuming particles with negligible mass and $E \simeq p_z$, Equation 1.10 is simplified to

$$\eta = -\ln\left(\tan\frac{\theta}{2}\right) \tag{1.11}$$

where η is the pseudorapidity which describes the angle θ of a particle with respect to the beam axis.

The pseudorapidity is the most commonly used variable for physics analyses since it is independent from the mass of the particle, relying exclusively on its polar position. ATLAS covers the pseudorapidity region $|\eta| < 4.9$. However, physics analyses usually focus on objects reconstructed in the pseudorapidity region $|\eta| < 2.5$, where particles

have a chance to be measured by all main ATLAS subsystems, i.e., the tracker, the calorimeters and the muon system which are described in some detail below.

Using pseudorapidity and the azimuthal angle, the distance between two particles can be expressed as

$$\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2} \tag{1.12}$$

where $\Delta\eta = |\eta_1 - \eta_2|$ and $\Delta\phi = |\phi_1 - \phi_2| - 2\pi\kappa$, with $\kappa = +1$ if $|\phi_1 - \phi_2| > \pi$ and $\kappa = 0$ otherwise.

In order to describe the trajectory of a charged particle moving inside a magnetic field, a set of 5 track parameters can be used:

$$(d_0, z_0, \phi, \eta, q/p_T). \tag{1.13}$$

d_0 is the signed transverse impact parameter, the distance of the trajectory from the z axis at the point of its closest approach, while z_0 is the z coordinate of the trajectory at the same point. q/p_T, where q is the charge of the particle, is measured by the radius of the curvature. All of them, together with the η and the ϕ at the origin, define the five parameters describing the helix traced by the movement of a charged particle in a magnetic field.

1.2.2 *Magnet system*

The measurement of momenta of charged particles is based on their deflection inside a magnetic field. In the case of the ATLAS experiment, the magnetic field is provided by the Magnet System [11] which consists of:

- A Central Solenoid (CS), coaxial with the beam axis and provides a 2 T axial magnetic field for the inner detector, while minimising the radiative thickness in front of the barrel electromagnetic calorimeter.
- A Barrel Toroid (BT) and two End-Cap Toroids (ECTs), which produce a mean toroidal magnetic field of 0.5 T and 1 T for the muon detectors in the central and endcap regions, respectively.

The complete system has a diameter of 22 m and a length of 26 m. Figure 1.8 shows a pictorial representation of the Magnet System.

Since the Central Solenoid, displayed in Figure 1.9a, is positioned in front of the electromagnetic calorimeter, the layout is carefully optimised to keep the material thickness as low as possible. In order to achieve the desired calorimeter performance, the solenoid assembly contributes a total of ~0.66 radiation lengths at normal incidence [12]. This requires, in particular, that the solenoid windings and LAr calorimeter share a common vacuum vessel, thereby eliminating two

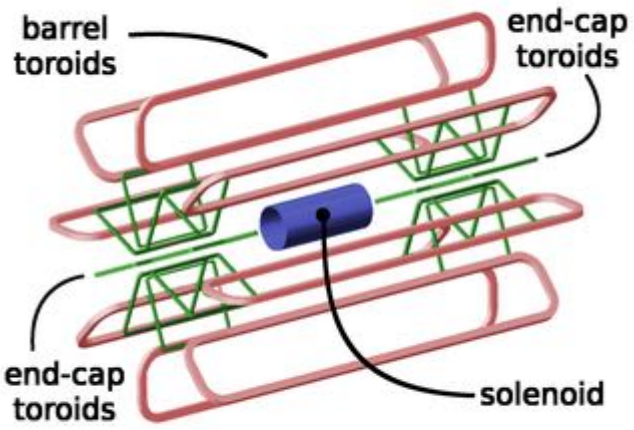


Figure 1.8: Schematic view of the complete ATLAS Magnet System. In the centre, the Central Solenoid provides the magnetic field for the Inner Detector while the 8 barrel and 2×8 end-cap of the coils toroids provide the magnetic field to the Muon Spectrometer.

vacuum walls. An additional heat shield consisting of 2 mm thick aluminium panels is installed between the solenoid and the inner wall of the cryostat. The single-layer coil is wound with a high-strength Al-stabilised NbTi conductor inside a 12 mm thick Al support cylinder. The inner and outer diameters of the solenoid are 2.46 m and 2.56 m and its axial length is 5.8 m. The flux is returned by the steel of the ATLAS hadronic calorimeter and its girder structure. The electromagnetic forces are counteracted by the combination of the coil and warm-to-cold mechanical support, which maintains the concentricity of the windings. The CS coil is designed to be as thin as possible without sacrificing the operational safety and reliability. Minimum coil material and an adequate safety margin for operation are obtained by distributing the stress uniformly between the coil components, while keeping the maximum strain due to the magnetic forces below 0.1 % in the principal stress components. All solenoid services pass through an S-shaped chimney at the top of the cryostat, routing the service lines to the corresponding control dewar.

The cylindrical volume surrounding the calorimeters is filled by the magnetic field of the Barrel Toroid. The BT is the largest component of the ATLAS detector, consisting of eight separate air-core coils each with 120 turns. This design was chosen because it minimises the multiple scattering of the muons as they travel through the detector [13]. The eight coils encased in individual racetrack-shaped, stainless-steel vacuum vessels, shown in Figure 1.9b. The coil assem-

bly is supported by eight inner and eight outer rings of struts that provide mechanical stability. The conductor and coil technology is based on winding pure Al-stabilized Nb/Ti/Cu into pancake-shaped coils, followed by vacuum impregnation. Services are brought to the coils through a cryogenic ring linking the eight cryostats to a separate service cryostat, which provides connections to the power supply, the helium refrigerator, the vacuum systems and the control system.

Finally, the two End-Cap Toroids consist of a single cold mass built up from eight flat, square coil units and eight keystone wedges, bolted and glued together into a rigid structure to withstand the Lorentz forces. Each coil has an axial length of 5 m with a total assembly weight of 239 tons.

Due to the magnetic forces, the ECT magnets are pulled into the BT and the corresponding axial forces are transferred to the BT cryostats via axial transfer points linking both magnet systems. The ECT cryostats have a classical turret for services.

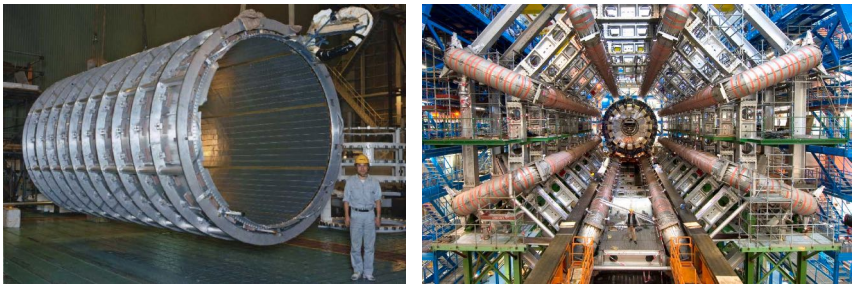


Figure 1.9: (a) The ATLAS solenoid providing the magnetic field in the inner detector. (b) Photograph of the toroid magnet in the barrel region of ATLAS.

All the magnets are indirectly cooled by forced flow of helium at 4.5 K through tubes welded on the casing of the windings. The CS is cooled via a deware coupled to the refrigerator, whereas the BT and ECT in addition have cold helium pumps to guarantee appropriate cooling by a forced helium flow.

1.2.3 *Inner detector*

The Inner Detector (ID) [14] is the innermost subdetector of ATLAS, placed just 50.5 mm from the Interaction Point (IP), which is further reduced to 33.25 mm with the addition of the Insertable B-Layer (IBL) [15] in 2014. It combines high-resolution detectors at the inner radii with continuous tracking elements at the outer radii, all contained inside the CS. The ID is designed to operate in the range $|\eta| < 2.5$, where approximately 1000 particles are expected to emerge from the collision point every 25 ns under nominal conditions. This very large track density requires fine-granularity detectors in order to

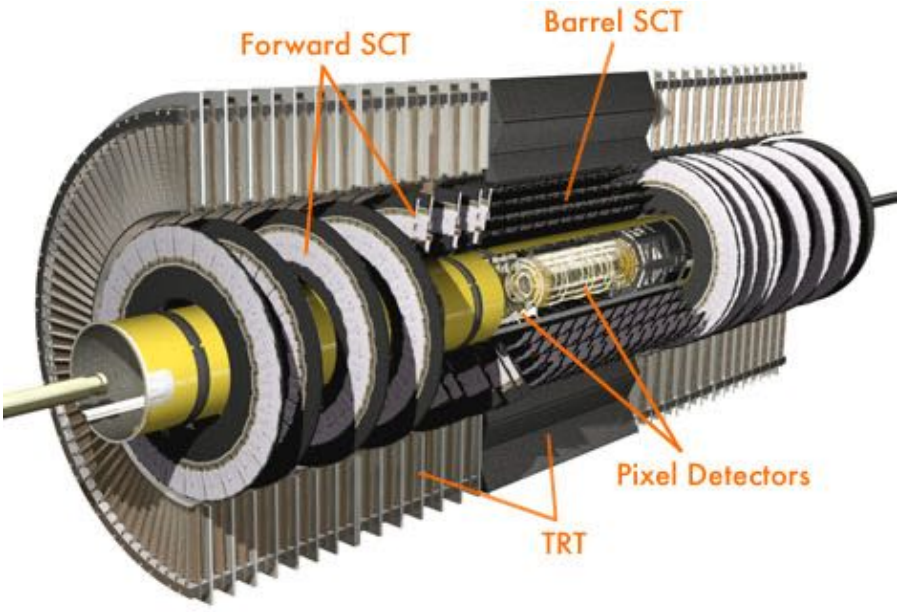


Figure 1.10: Schematic view of the layout of the Inner Detector of ATLAS. The IBL layer, at a radius of 33.25 mm, is not depicted.

be able to provide high-precision for both momentum and vertex measurements. Furthermore, its proximity to the IP requires radiation resistant components while its placement in front of the calorimeters requires the minimisation of the amount of material to avoid the possible degradation of the energy measurements.

The layout of the Inner Detector is shown in Figure 1.10 while Figure 1.11 shows a three-dimensional cut-away view of the ID. It consists of concentric layers (in the barrel, parallel disks in the endcaps) of three types of detectors, that operate independently but complement each other to achieve a moment resolution of $\sigma_{p_T}/p_T \simeq 0.05\% \oplus 1\%$. At inner radii, high resolution pattern recognition capabilities are available using discrete space points from the silicon pixel layers and the stereo pairs of silicon microstrip (SCT) layers. At larger radii, the transition radiation tracker (TRT) comprises many layers of gaseous straw tube elements interleaved with transition radiation material. With an average of 36 hits per track, it provides continuous tracking to enhance the pattern recognition and improve the momentum resolution over $|\eta| < 2.0$. Each of these systems is described in the following subsections.

1.2.3.1 *The Pixel and IBL detectors*

The Pixel Detector is made of three layers of silicon pixels, as shown in the schematic view of Figure 1.11. It provides tracking information and is also largely responsible for the ability of the detector to reconstruct secondary vertices. It is organised of three cylindrical layers in the barrel region and three concentric discs in the endcaps covering

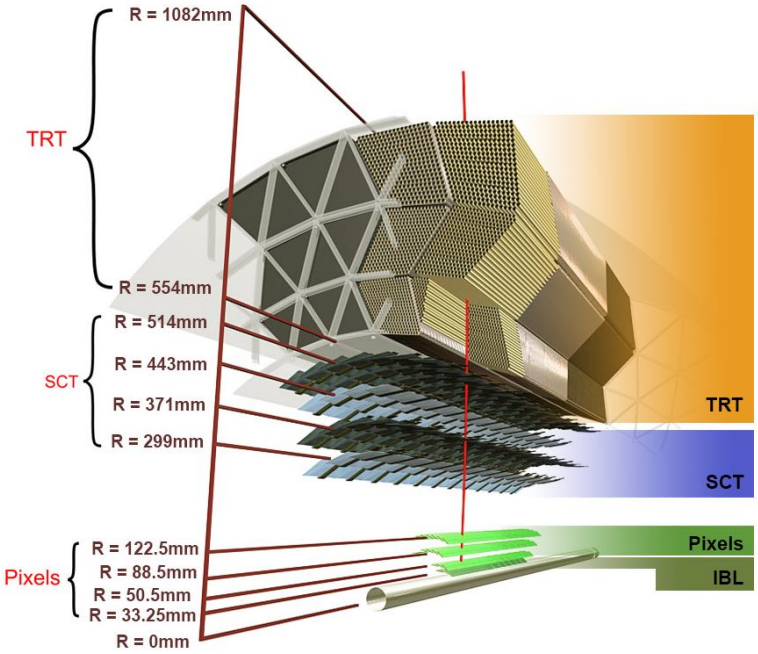


Figure 1.11: The sensors and the structural elements of the Inner Detector.

the range up to $|\eta| < 2.5$. The innermost layer[1], named B-Layer, is located just 50.5 mm from the interaction point to provide the optimal resolution on the impact parameters. The other two layers are placed at 88.5 mm and 122.5 mm from the interaction point. The three layers contain approximately 80.4 million readout channels, about 80 % of the total readout channels of ATLAS.

The pixel detector consists of 1744 identical modular units, with an external dimensions of $19 \times 63\,\mathrm{mm}^2$. The nominal pixel size is $50 \times 400\,\mu\mathrm{m}^2$. There are 47 232 pixels on each sensor, but for reasons of space there are four ganged pixels in each column of the front-end chip, thus leading to a total of 46 080 readout channels. The silicon sensors are 250 µm thick, using oxygenated n-type wafers with readout pixels on the n^+ implanted side of the detector. This design, involving double-sided processing of higher cost and complexity, has been chosen because of:

- the n^+ implants allow the detector to operate with good charge collection efficiency after type inversion, even when it operates at full depletion voltage;
- highly oxygenated material has been shown to give increased radiation tolerance to charged hadrons, with improved charge collection after type inversion and lower depletion voltage.

Figure 1.12 shows the structure of a pixel module.

1 The IBL is not considered.

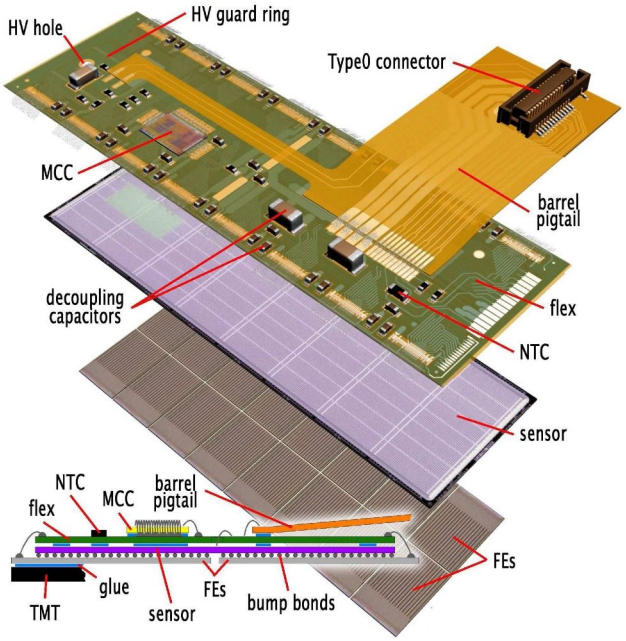
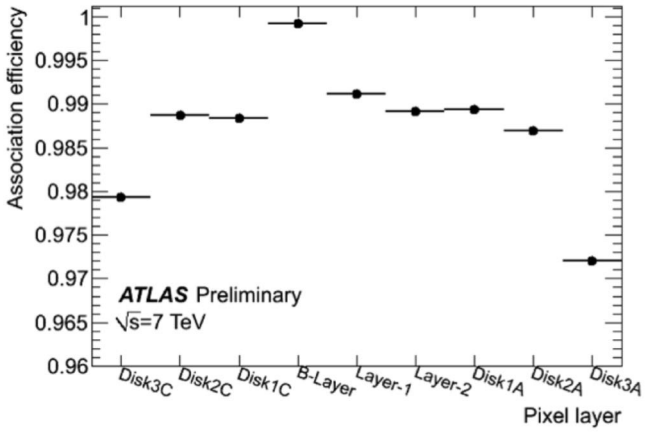


Figure 1.12: Pixel module structure of the ATLAS tracker.

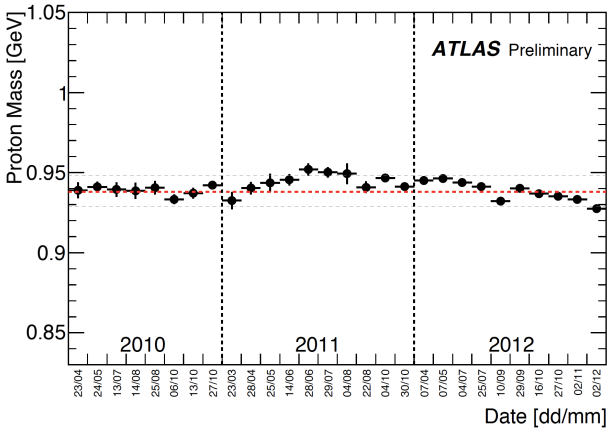
To measure the charge deposited in each pixel, units of Time-over-Threshold (ToT) are used, measured in clock cycles. ToT is the time that the analogue signal on the pixel has spent over a predefined threshold value. This information is proportional to the charge released in the pixel, which in turn depends on the energy and charge of the particle interacting with the pixel. At normal incidence, the pixel detectors spatial resolution is measured to be of $\sim 12\,\mu m$ in the $R-\phi$ plane, and $115\,\mu m$ in the longitudinal direction, with approximately 80 % of the tracks having a single pixel hit.

Figure 1.13a shows the hit to track association efficiency for different layers of the Pixel Detector during the Run1 data taking period. As can be seen from the plot, the section with the higher efficiency is the B-Layer, the one closer to the interaction point. Moreover, all the other sections show very good efficiencies. Figure 1.13b illustrates the stability of the dE/dx calibration for the Run1 data taking period through the stability of the proton mass, calculated from the momentum measured in the ID and the specific energy loss measured in the Pixel detector.

In 2014, the innermost pixel layer was added in the form of an "Insertable B Layer" (IBL), visible in Figure 1.11, placed at a radius of 33.25 mm from the beam-pipe. This insertion improves the b-tagging abilities of the ATLAS tracker. IBL consists of 14 axial staves tilted by 14° with a 20 % overlap in $r-\phi$ and no overlap in the z axis. Each stave contains 20 modules, 4 single chip modules with 3D sensors on each end and 12 double chip modules with planar sensors on the centre. All sensors consist of 26 880 pixel cells with a granularity of



(a)



(b)

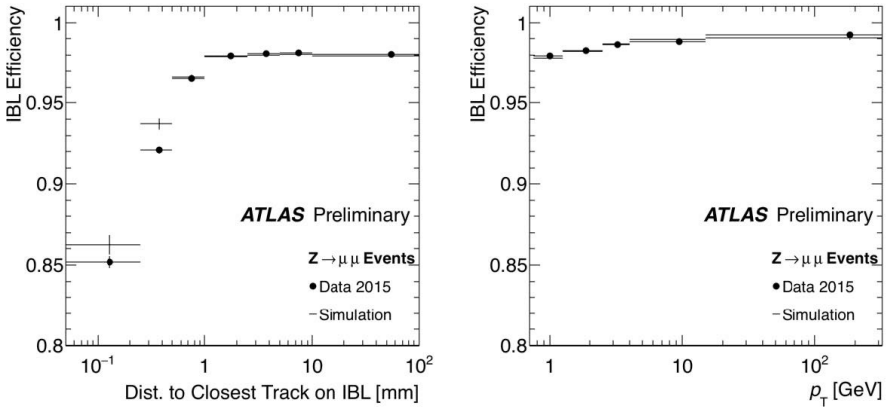
Figure 1.13: (a) Hit to track association efficiency in different sections of the Pixel Detector during Run-1. (b) Illustration of signal stability by reconstructing the proton mass from pixel dE/dx and Inner Detector.

250 µm × 50 µm (excluding the edge pixels) organised in arrays of 80 × 336 pixels.

The efficiency of associating an IBL hit to a reconstructed particle track is shown in Figure 1.14. The efficiencies are presented as a function of the tracks distance to the closest track on the IBL module (Figure 1.14a) and the particle p_T (Figure 1.14b). As can be seen, the IBL efficiencies are close to the pixel ones, presented in Figure 1.13.

1.2.3.2 *Semiconductor tracker*

The SCT system is designed to provide eight precision measurements per track in the intermediate radial range, contributing to the measurement of momentum, impact parameter and vertex position, as well as providing good pattern recognition by the use of high granularity.



(a) (b)

Figure 1.14: Efficiency of associating an IBL hit to a reconstructed particle track as a function of the tracks distance to the closest track on the IBL module (a) and the particle p_T (b) for selected $Z \to \mu\mu$ events in collision data (circles with error bars) and simulation (lines). The efficiency is computed as the fraction of selected tracks with $p_T > 0.7\,GeV$ and $|\eta| < 2.5$ having an associated IBL cluster. Disabled and unstable IBL modules have been removed.

The SCT are arranged in eight strip layers and provide an almost hermetic coverage with at least four precision space-point measurements over the fiducial coverage of the inner detector. It consists of 15 912 sensors which, for reasons of cost and reliability, make use of a classic single sided p-in-n technology with AC coupled readout strips. The strip pitch is determined by the required digitising precision, granularity, particle occupancy and noise performance. A strip pitch of 80 µm with two 6 cm long sensors has been chosen for the rectangular barrel sensors, while, radial strips of constant azimuth with mean pitch of ~ 80 µm have been chosen for the trapezoidal end-cap sensors, consisting in a total of 768 active strips of 12 cm length per sensor. The spatial resolution of the SCT detectors is measured to be of ~ 17 µm in the $R - \phi$ plane, and ~ 580 µm in the longitudinal one. Tracks can be distinguished if they are separated by more than 200 µm.

The measured efficiency for each barrel and each endcap disk at the end of Run1 is shown in Figure 1.15. These measurements use data recorded in a proton-proton run with low pile-up, with less than one interaction per bunch-crossing on average. The overall efficiency is 99.74 ± 0.04 %, where the error is systematic (the statistical error is negligible).

1.2.3.3 *Transition radiation tracker*

The outermost part of the ID is costituted by the Transition Radiation Tracker (TRT) [16]. In the barrel region the TRT detector is arranged

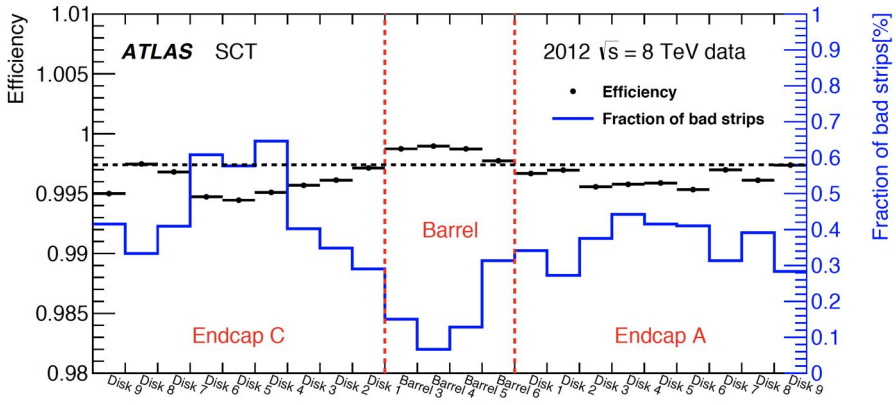


Figure 1.15: Mean intrinsic hit efficiency for each layer of the SCT measured in proton-proton data at $\sqrt{s} = 8\,\text{TeV}$ in 2012. The dashed line indicates the overall mean value. The blue line and right-hand axis show the fraction of strips in the layer which are disabled (dead or noisy).

in 73 layers interleaved with fibres providing transition radiation for electron identification, while, in the end-cap one, the TRT is made of 160 planes interleaved with foils. Thanks to this layout, all charged tracks with $p_T > 0.5\,\text{GeV}$ and $|\eta| < 2.0$ transverse at least 36 straws, compensating for the bad accuracy of 130 µm which is an order of magnitude larger than the other sub-detectors. The TRT are constituted by polyimide drift straw tubes of 4 mm diameter containing a mixture of xenon, tetrafluoromethane and carbon dioxide. The straw tube wall is made of two 35 µm thick multi-layer films bonded back to back, developed in order to achieve good electrical and mechanical properties with minimal wall thickness. The bare material is a 25 µm thick polyimide film, coated on one side with a 0.2 µm Al layer and protected by a 5 µm thick graphite-polyimide layer. The anode is made of 31 µm diameter tungsten wires plated with 0.5 µm gold, supported at the end by an end-plug and directly connected on both sides to the front-end electronics. The straws are oriented parallel to the beam direction in the barrel and radially in the end-caps, therefore the TRT only provides information in the $R - \phi$ plane.

The TRT provides pion-electron discrimination since the electron identification is enhanced by the detection of transition-radiation photons in the straw tubes. The TRT plays a central role in electron identification, cross-checking and complementing the Electromagnetic Calorimeter, especially at energies below 25 GeV. In addition, the TRT contributes to the reconstruction and identification of track segments from photon conversions down to 1 GeV and of electrons which radiate a large fraction of their energy in the silicon layers [17].

Figure 1.16 shows the straw efficiency for an Ar-based mixture in the end-cap wheels as a function of the track-to-wire distance. As can be seen from the figure, the straw efficiency profile presents a plateau

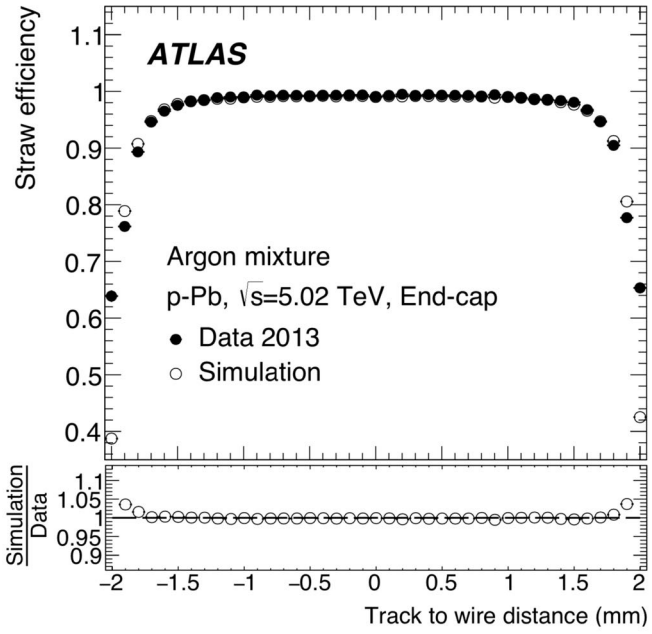


Figure 1.16: Straw efficiency in the end-cap wheels filled with an Ar-based mixture versus track-to-wire distance in the data (solid circles) and simulation (open circles).

close to 100 % efficiency and very good agreement between simulation and data.

1.2.4 *Calorimeters*

The ATLAS Calorimetry System, shown in Figure 1.17, consist of a number of sampling detectors with full φ-symmetry and coverage around the beam axis. The ATLAS calorimetric system is mainly meant to:

- measure the energy of electrons, photons and jets;
- determine the missing transverse energy (E_T^{miss}) as the negative of the vectorial sum of the energy depositions in the calorimeter, where the deposited energy is given the direction from the interaction point towards the energy deposition; obviously, the determination of (E_T^{miss}) requires excellent energy measurement resolution and complete hermeticity;
- identify particles exploiting the transverse and longitudinal shape of the particle shower and the longitudinal energy leakage;
- determine the spatial isolation of particles, which is extremely useful for background suppression, especially in various sources for rare processes. Furthermore, it is also valuable in less rare processes, e.g., a lepton from an electroweak boson decay.

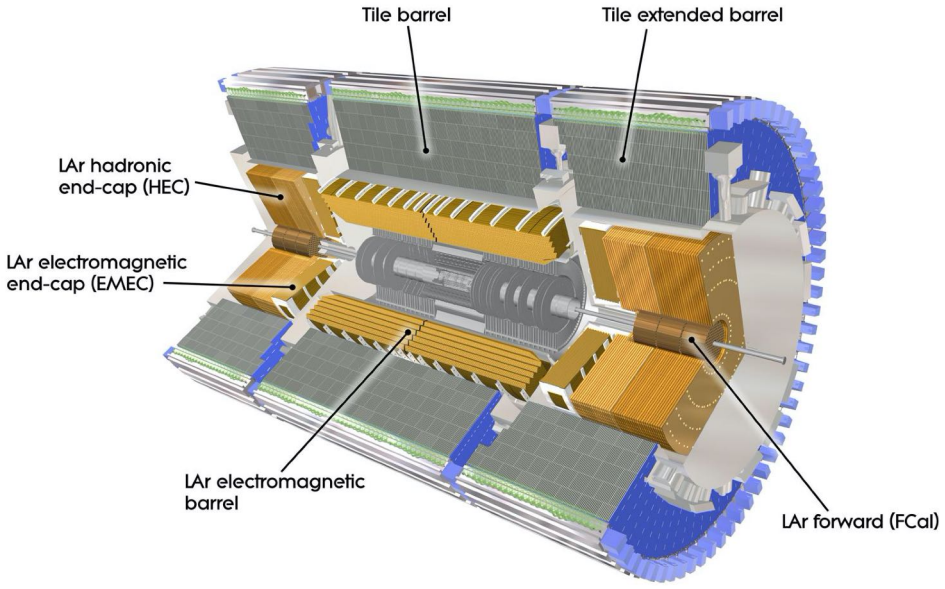


Figure 1.17: Cut-away view of the ATLAS calorimeters.

The Calorimeter System is divided in three subsystems, the Electromagnetic Calorimeter (ECal), the Hadronic Calorimeter (HCal) and the Forward Calorimeter (FCal). In order to provide good containment for electromagnetic and hadronic showers, calorimeter depth is a fundamental design consideration. The total thickness of the EM calorimeter is $> 22\ X_0$ (radiation lengths) in the barrel and $> 24\ X_0$ in the end-caps. The approximate $9.7\ \lambda$ (interaction lengths) of active calorimeter in the barrel (10λ in the end-caps) are adequate to provide good containment for high energy jets.

1.2.4.1 *Electromagnetic calorimeter*

The main requirements for the ATLAS electromagnetic calorimeter are:

- good electron identification capability from 1 GeV up to 5 TeV;
- very good energy resolution over the range 10 GeV to 300 GeV;
- good θ resolution for the measurement of showers;
- very good γ-jet, e-jet and τ-jet discrimination.

The ECal is divided into a barrel part, covering $|\eta| < 1.475$, and two end-cap components, $1.375 < |\eta| < 3.2$, each housed in their own cryostat.

The barrel calorimeter consists of two identical half-barrels, separated by a small gap of 4 mm, while each end-cap calorimeter is mechanically divided into two coaxial wheels.

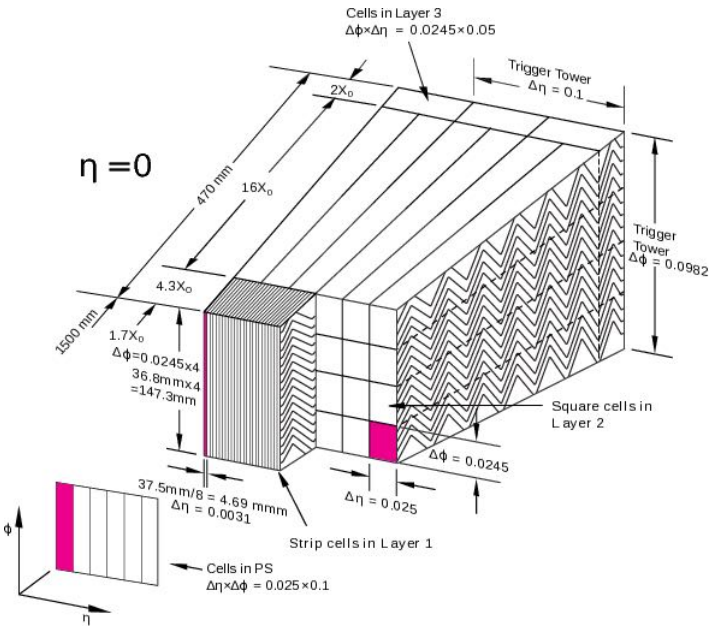
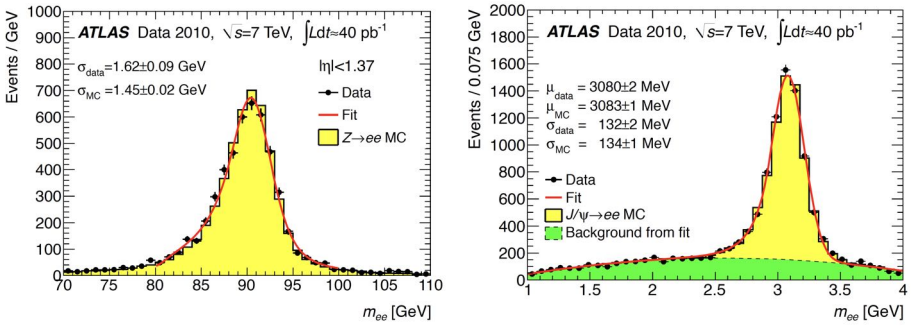


Figure 1.18: Electromagnetic calorimeter granularities.

The ECal is a lead-LAr detector with accordion shaped electrodes and lead absorber plates over its full coverage. In order to improve the mechanical strength of the absorber, the lead plates are glued to two stainless-steel sheets of 0.2 mm thick, using a resin-impregnated glass fibre. The readout electrodes consist of three conductive copper layers separated by insulating polyimide sheets and are located in the gaps between the absorbers. The two outer electrodes layers are at the high-voltage potential, while the inner one is used for reading out the signal via capacitive coupling. The accordion shape of electrodes and absorber plates has been chosen in order to provides complete φ symmetry without azimuthal cracks, and a fast extraction of the signal at the rear or at the front of the electrodes. In the barrel, the accordion waves are axial and run in φ, whereas in the end-caps, the waves are parallel to the radial direction. The lead thickness in the absorber plates has been optimised as a function of η in terms of ECal performance in energy resolution.

Over the region devoted to precision physics, $|\eta| < 2.5$, the ECal is segmented in three sections in depth, while, the end-cap inner wheel is segmented in two and it has a coarser lateral granularity than for the rest of the acceptance. The granularities of each section are detailed in Figure 1.18. Furthermore, in the region of $|\eta| < 1.8$, a presampler detector, made of an active LAr layer of 1.1 cm thickness in the barrel and 0.5 cm in the end-cap region, is used to correct for the energy lost by electrons and photons upstream of the calorimeter. All these features lead to a very uniform performance in terms of linearity and resolution.



(a) (b)

Figure 1.19: Reconstructed dielectron mass distribution for $Z \to ee$ (a) and $J/\psi \to ee$ (b) decays and comparison with simulation.

The signals from the EMCal are extracted at the detector inner and outer faces and sent to preamplifiers located outside the cryostats. The preamplifier output is formed by bipolar shapers, sampled every 25 ns, and stored in analogue memories using Switching Capacitors Arrays (SCA) during the level-1 trigger latency. If the level-1 trigger is validated, the corresponding samples are extracted from the SCA, digitized and read out to the data acquisition system.

The design resolution of the ECal is given by the formula [18]:

$$\frac{\delta E}{E} = \frac{10\%}{\sqrt{E}} \oplus 0.7\% \tag{1.14}$$

At the end of the Run1 data taking period, a detailed study was performed to determine the performances of the ATLAS ECal detector using the decays of the Z, W and J/ψ particles [19]. The EM showers that develop in the calorimeters are reconstructed as clusters of calorimeter cells that contain a large fraction of the deposited energy. Some energy is not contained in the cluster and some is lost before or after the calorimeter. For these reasons, corrections need to be applied. Calibration constants are calculated from MC simulation as a function of η, energy and shower depth. The overall energy scale is set with reconstructed mass distributions from $Z \to ee$ events (an example of which can be seen in Figure 1.19a), and cross checked using the electron E/p distribution in $W \to e\nu$ events. After set the energy scale using the calibration constants, the calorimeter energy resolution can be studied from data. The resolution is parametrized with the following formula:

$$\frac{\delta E}{E} = \frac{a}{\sqrt{E}} \oplus \frac{b}{E} \oplus c \tag{1.15}$$

where a, b and c are η-dependent parameters called *sampling term*, *noise term* and *constant term* ,respectively. Mass measurements in di-electron events are compared to simulation and the lineshape of the

Subsystem	η-range	Effective constant term c_{data}
EMB	$\|\eta\| < 1.37$	$1.2\,\% \pm 0.1\,\%(stat)^{+0.5\,\%}_{-0.6\,\%}(syst)$
EMEC-OW	$1.52 < \|\eta\| < 2.47$	$1.8\,\% \pm 0.4\,\%(stat) \pm 0.4\,\%(syst)$
EMEC-IW	$2.5 < \|\eta\| < 3.2$	$3.3\,\% \pm 0.2\,\%(stat) \pm 1.1\,\%(syst)$
FCal	$3.2 < \|\eta\| < 4.9$	$2.5\,\% \pm 0.4\,\%(stat)^{+1\,\%}_{-1.5\,\%}(syst)$

Table 1.3: Energy resolution constant term for all the ECal partitions.

$J/\psi \to ee$ decay is used, as presented in Figure 1.19b, to verify that the sampling and noise terms are modeled correctly in simulation. Finally, the width of the $Z \to ee$ decay mass distribution is then used to extract an *effective constant term*, c_{data}, including the calorimeter constant term as well as the effect of inhomogeneities and residual mis-calibration. The resulting values of c_{data}, for all the different η ranges are presented in Table 1.3.

1.2.4.2 *Hadronic calorimeters*

The main requirements for the hadronic calorimeter are:

- jet detection capability up to $|\eta| < 5$ for efficient tagging of the forward jets and good E_T^{miss} reconstruction;
- good resolution for both single jet reconstruction and jet-jet mass spectroscopy.

The ATLAS hadronic calorimeters cover the range $|\eta| < 4.9$ using different techniques best suited to the widely varying requirements of the physics processes of interest and of the radiation conditions over this large η-range. Over the range $|\eta| < 1.7$, the steel scintillating-tile technique is used for the barrel and extended barrel tile calorimeters, and for partially instrumenting the gap between them with the intermediate tile calorimeter (ITC). Over the range $1.5 < |\eta| < 4.9$, LAr calorimeter was chosen. The hadronic end-cap calorimeter (HEC) extends to $|\eta| < 3.2$, while the range $3.2 < |\eta| < 4.9$ is covered by the high density forward calorimeter (FCAL). Both the HEC and FCAL are integrated in the same cryostat as that housing the EM end-caps. An important parameter in the design of the hadronic calorimeter is the thickness. The calorimeter has to provide good containment for hadronic showers and to reduce punch-through into the muon system to a minimum. The total hadronic calorimeters thickness is 11λ at $\eta = 0$, including about 1.5λ from the outer support. Close to 10λ of active calorimeter are adequate to provide good resolution for high energy jets. Together with the large η coverage, this will also guarantee a good E_T^{miss} measurement, important for many physics signatures.

The Tile Calorimeter [20] is a sampling calorimeter using steel as absorber material and scintillating tiles as the active one. Figure 1.20 shows a sketch of a TileCal module. It is located behind the liquid argon electromagnetic calorimeter and is subdivided into a central barrel, of 5.8 m length, and two extended barrels of 2.6 m length, with an inner radius of 2.28 m each and an outer radius of 4.25 m, corresponding in a total radial depth of approximately 7.4λ. In the calorimeter the tiles are placed radially, normal to the beam line and staggered in depth, with a periodic structure along z. They are 3 mm thick, while the total thickness of the steel plates in one period is 14 mm. The two sides of the scintillating tiles are read out by wavelenght shifting fibres into two separate photomultipliers, which provide redundancy and sufficient information to partially equalize signals produced by particles entering the scintillating tiles at different impact positions. In order to partially recover the energy lost in the crack regions of the detector, the gap region between the barrel and the extended barrel is instrumented with special modules, consisting of steel-scintillator sandwiches and thin scintillator counters in the sectors where the available space in the gaps is limited. The electronics and readout of the tile calorimeter are highly integrated with the mechanical structure. The photomultiplier tubes and all the front-end electronics are mounted in 1.4 m long aluminium units, called drawers, which are inserted inside the support girder at the rear of each module. Finally, the calorimeter is equipped with three calibration systems, a charge injection, a laser and a ^{137}Cs radioactive source, which are able to test the optical and digitised signals at various stages and are used to set the PMT gains to a uniformity of $\pm$3%.

The Hadronic End-cap Calorimeter is a copper-LAr flat-plate design sampling calorimeter, which covers the range $1.5 < |\eta| < 3.2$ It shares each of the two liquid-argon end-cap cryostats with the electromagnetic end-cap and the forward calorimeters. The HEC consists of two wheels in each end-cap cryostat, a front wheel (HEC1) and a rear wheel (HEC2), each of them of 2.03 m of outher radius and containing two longitudinal sections. The HEC1 wheel is built out of 24 copper plates, each 25 mm thick, while HEC2, farther from the interaction point, uses 16 plates of 50 mm thick. In both wheels, the 8.5 mm gap between consecutive copper plates is equipped with three parallel electrodes, splitting the gap into four LAr drift spaces of about 1.8 mm, resulting in a sampling fractions of 4.4% and 2.2% for HEC1 and HEC2, respectively. The space between the electrodes is maintained using a honeycomb sheet. The central electrode is the readout one, which carries a pad structure covered by a high-resistivity layer, which define the lateral segmentation of the calorimeter. The other two electrodes carry surfaces of high resistivity to which high voltage is applied.

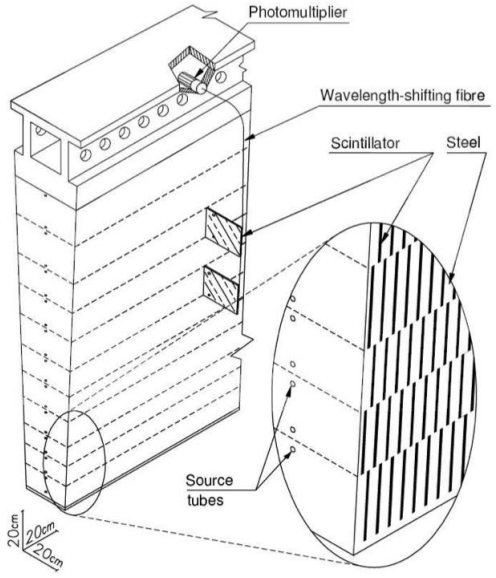
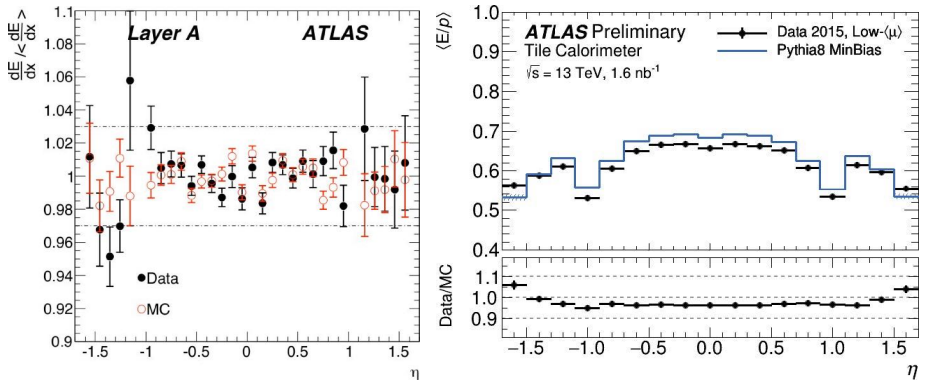


Figure 1.20: Diagram of a TileCal module.



(a) (b)

Figure 1.21: (a) Uniformity of the cell response to cosmic muons expressed in terms of the normalised truncated mean of dE/dx, as a function of η for radial layer A. (b) Energy over momentum E/p as a function of η for isolated hadron track using 2015 data.

The design resolution of the hadronic calorimeter in the barrel and endcap regions is parametrized with the following formula:

$$\frac{\delta E}{E} = \frac{50\%}{\sqrt{E}} \oplus 3\%. \tag{1.16}$$

The performance of the methods used to reconstruct and calibrate objects and correct for problematic regions of the detector was evaluated with single isolated particles [20]. Single isolated muons were used to check the electromagnetic scale across the detector, using the ratio of the energy deposited to the path length traveled by the muon in a cell. For muon energies below 100 GeV this ratio is approx-

imately constant. The uniformity of the cell response to cosmic muons expressed in terms of the normalised truncated mean of dE/dx, as a function of η for one of the layers is shown in Figure 1.21a. The response is integrated over all cells in each bin and the results for data and Monte Carlo are normalized to their averages. The performances of the ATLAS hadronic calorimeters were also studied with single isolated charged hadrons, using the ratio of the energy measured in TileCal, to the momentum of their track reconstructed in the inner detector E/p. The mean E/p as a function of is shown in Figure 1.21b for data taken in 2015 and Monte Carlo simulations. Most of the data and MC agree within 5 %.

1.2.4.3 *Forward calorimeter*

The Forward Calorimeter is a particularly challenging detector owing to the high level of radiation it has to cope with. In ATLAS, the FCAL is integrated into the end-cap cryostat, with a front face of about 4.7 m from the interaction point. Compared to layouts with a forward calorimeter situated at much larger distances from the interaction point, the survival of such a calorimeter in terms of radiation resistances is clearly more difficult. On the other hand, the integrated FCAL provides clear benefits in terms of uniformity of the calorimetric coverage as well as reduced radiation background levels in the muon spectrometer. The FCAL consists of three sections, one of which is made of copper, while the others are made of tungsten. In each section, the calorimeter consists of a metal matrix with regularly spaced longitudinal channels filled with the electrode structure consisting of concentric rods and tubes parallel to the beam axis. The rods are at positive high voltage while the tubes and matrix are grounded. The LAr in the gap act as sensitive medium. This geometry allows for an excellent control of the gaps which are as small as 250 µm in the first section. The nominal resolution for jets of the FCal is

$$\frac{\delta E}{E} = \frac{100\%}{\sqrt{E}} \oplus 10\%. \tag{1.17}$$

1.2.5 *Muon spectrometer*

The muon spectrometer [21] forms the outer part of the ATLAS detector and it is designed to detect charged particles, mainly muons, exiting the barrel and end-cap calorimeters, to measure their momentum in the pseudorapidity range $|\eta| < 2.7$, and also to trigger on them in the region $|\eta| < 2.4$. It exploits the magnetic deflection of tracks in the large superconducting air-core toroid magnets, instrumented with separate trigger and high-precision tracking chambers.

Because the particles momentum measurement is based on the measure of the track bending inside a magnetic field, the parameter

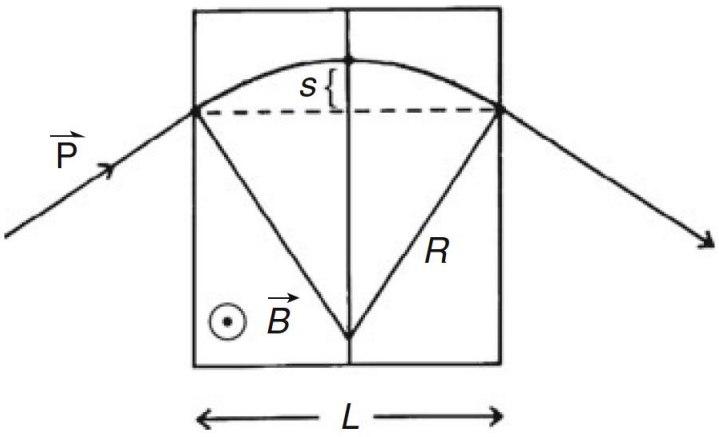


Figure 1.22: Sagitta measurement in magnetic spectrometer.

that has to be measured is the deviation from a straight line, which is known as the sagitta s of the track, shown in Figure 1.22.

Due to the Lorentz force, a charged particle moving in a magnetic field B follows an helicoidal path with bending radius given by the equation $p_T = qBR$. If the magnetic field is present only in a region of lenght l, which is the case for the muons traveling in the ATLAS detector, the bending angle is given by:

$$\sin(\alpha/2) = \frac{l/2}{R} = k\frac{Bl}{2p_T} \rightarrow \alpha \approx k\frac{Bl}{p_T} \tag{1.18}$$

and the sagitta can be calculated from:

$$s = R - R\cos(\alpha/2) \approx R\frac{\alpha^2}{8} \approx \frac{kBl^2}{8} \tag{1.19}$$

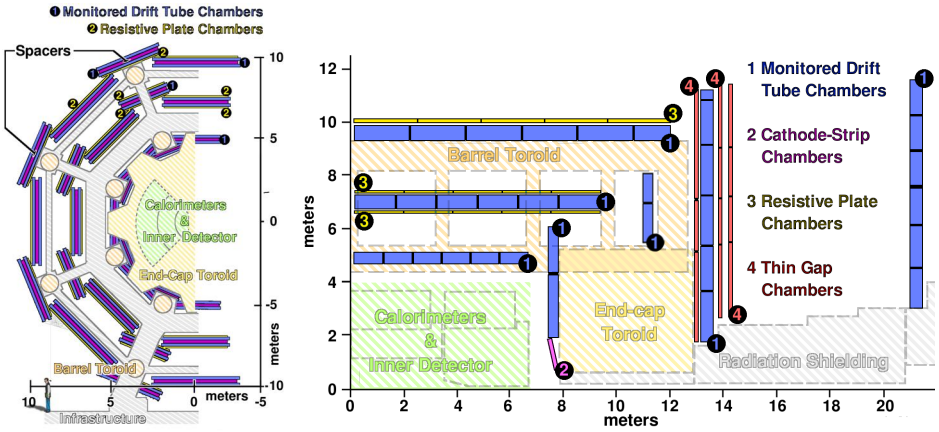
as long as $\alpha << 1$.

In order to measure the sagitta, and therefore the p_T, the following requirements are essential:

- a strong and large magnet with a well-known field strength;
- several precise measurements of the particle position along its trajectory inside and/or outside the magnetic field performed with high resolution tracking detectors.

The performance in terms of bending power is characterised by the field integral $\int B dl$, where B is the field component normal to the muon direction and the integral is computed along an infinite momentum muon trajectory, between the innermost and outermost muon-chamber planes.

The barrel toroid provides 1.5 to 5.5 T m of bending power in the pseudorapidity range $|\eta| < 1.4$, and the end-cap toroids approximately 1 to 7.5 T m in the region $1.6 < |\eta| < 2.7$. The bending power is lower



(a) (b)

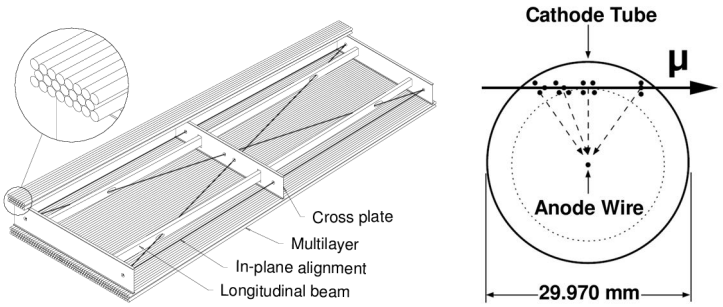
Figure 1.23: (a) Cross-sections of the barrel muon system perpendicular to the beam axis (non-bending plane) and (b) in a plane containing the beam axis (bending plane).

in the transition regions where the two magnets overlap. For magnetic field reconstruction, the goal is to determine the bending power along the muon trajectory to a few parts in a thousand. For this reason the field is continuously monitored by a total of approximately 1800 sensors distributed throughout the spectrometer volume. Their readings are compared with magnetic-field simulations and used for reconstructing the position of the toroid coils in space, as well as to account for magnetic perturbations induced by the tile calorimeter and other nearby metallic structures.

The chambers in the barrel are arranged in three concentric cylindrical shells around the beam axis at radii of approximately 5 m, 7.5 m, and 10 m. In the two end-cap regions the muon chambers are placed perpendicular to the z-axis, located at distances of $|z| \approx 7.4$ m, 14 m, and 21.5 m from the interaction point. Figure 1.23 give cross-sections in the planes transverse to, and containing, the beam axis.

The largest area of the Muon Spectrometer is covered by Monitored Drift Tubes (MDT) which fill the range $|\eta| < 2.7$ ($|\eta| < 2.0$ for the innermost layer), providing a precision measurement of the track coordinates in the principal bending direction of the magnetic field. A MDT chamber consists of multiple layers of tubes, each of them acting as an autonomous gaseous detector. A view of an MDT chamber and a tube can be seen in Figure 1.24.

The MDT tubes are made of aluminum, 30 mm in diameter and 400 µm thick, acting as the cathode. The tubes are filled with a gas mixture consisting of 93%Ar and 7%CO_2 pressurised at 3 bar. In their centres, a gold-plated tungsten-rhenium wire at a voltage of 3080 V acts as the anode. A particle traversing the tube ionises atoms in the gas mixture. The resulting free electrons drift towards the anode, accelerated by the electric field inside the tube, creating an avalanche



(a) (b)

Figure 1.24: (a) Schematic view of the mechanical structure of an MDT chamber. (b) Cross section of an aluminium tube.

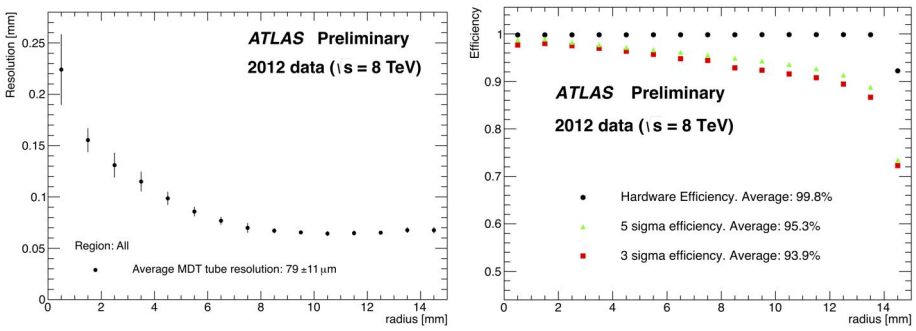
that is collected by the wire. The arrival of the avalanche is measured as a voltage drop on the anode wire. If a predefined threshold is reached, the signal is recorded. The timing of this signal determines the drift time of the electrons. Given the radii of the tubes, the maximum drift time is approximately 700 ns. To translate the drift time into a spatial position, a space-to-drift-time relation is used.

The average resolution is about 80 µm per tube and 35 µm per chamber. Figure 1.25a shows the MDT spatial resolution measured with 2012 data.

In Figure 1.25b, the MDT efficiency versus the drift radius is, instead, presented. The hardware efficiency measures if there is a tube response when a muon passes by, and 3(5) sigma efficiency are defined by the number of hits with their segment track residual value falling inside the range of 3(5) times of tube resolution divided by the number of total hits.

At large pseudorapidities, Cathode Strip Chambers (CSC), which are multiwire proportional chambers with high granularity, operating with a carbon dioxide, argon and tetrafluoromethane gas mixture are used. In the CSCs the precision coordinate is obtained by measuring the charge induced on the segmented cathode by the avalanche formed on the anode wire. Spatial resolutions of ~40 mm are achieved by segmentation of the cathode and by charge interpolation between contiguous strips. Due to their small electron drift time (30 ns) and good time resolution (7 ns), in order to withstand the demanding rate and background conditions they were chosen for the innermost plane over $2 < |\eta| < 2.7$.

The trigger system covers the pseudorapidity range $|\eta| < 2.4$. Resistive Plate Chambers (RPC) are used in the barrel and Thin Gap Chambers (TGC) in the end-cap regions. RPCs are gaseous detectors providing an excellent time resolution of 1 ns, but with a poor spatial resolution, typically of 1 cm. The basic RPC unit is a narrow gas gap formed by two parallel resistive bakelite plates separated by insulating spacers and operated with tetrafluoroethane mixed with small



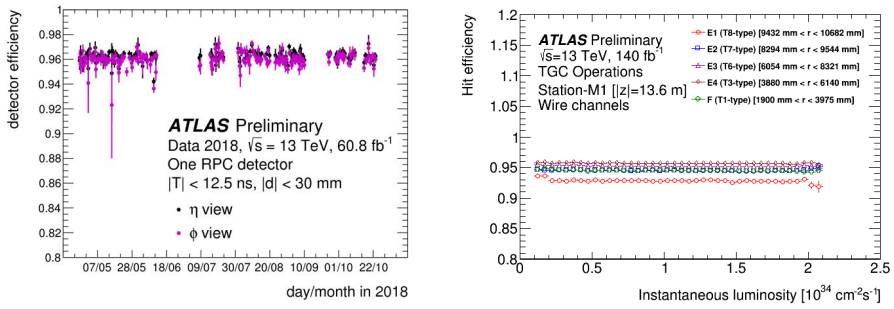
(a) (b)

Figure 1.25: (a) MDT tube resolution versus radius (distance of the particle trajectory from the wire) for all MDT chambers. The error bar is a combination of systematics and statistical errors. The tube resolution is derived from the segment track residual distributions for corresponding radius range, and the average is the result of the whole radius range. (b) MDT tube efficiency versus radius of all MDT chambers.

amounts of sulphur hexafluoride. Primary ionisation electrons are multiplied by a high, uniform electric field and the signal is readout via capacitive coupling of metal strips on both sides of the detector. The TGCs, instead, are multi-wire proportional chambers in which the anode wires provide the trigger information while the readout strips, orthogonally arranged with respect to the wires, are used to measure the second coordinate. They are operated with a gas mixture of carbon dioxide and n-penthane. The chamber time resolution is 4 ns, while their spatial resolution varies from 2 mm to 6 mm. The trigger chambers for the muon spectrometer serve a threefold purpose: providing bunch-crossing identification, providing well-defined p_T thresholds, and measuring the muon coordinate in the direction orthogonal to the one determined by the precision-tracking chambers.

In Figure 1.26, the RPC and TGC detector efficiencies, for the 2018 data taking period, are shown.

The overall performance over the large areas involved, particularly at the highest momenta, depends on the alignment of the muon chambers with respect to each other and with respect to the other subdetectors. The accuracy of the stand-alone muon momentum measurement requires a precision of 30 μm on the relative alignment of chambers both within each projective tower and between consecutive layers in immediately adjacent towers. The internal deformations and relative positions of the MDT chambers are monitored by approximately 12 000 precision-mounted alignment sensors, all based on the monitoring of deviations from straight lines.



(a) (b)

Figure 1.26: (a) RPC efficiency as a function of time for one readout panel in the η (black) and ϕ (magenta) views. Each point corresponds to a different ATLAS run recorded in 2018 during pp collisions. (b) Averaged hit efficiency for particular chamber type in one of the ATLAS TGC station ($|z| = 13.6\,m$) as a function of instantaneous luminosity in Run2.

1.2.6 *Trigger and Data Acquisition System*

The Trigger and Data Acquisition system (TDAQ) plays one of the most crucial role in the whole experiment. The interaction rate in ATLAS is about 10^9 $event/s$, but, due to the technology and resource limitations, only a part of the collected data can be stored for further analysis, typically 10^3 $event/s$. In order to reduce the events rate of about 10^6 *while* maintaining the maximum efficiency for the physics, data from events not physically interesting, i.e. minimum bias events, need to be rejected. Moreover, due to the trigger decision time of few ms, which is much bigger than the $25\,ns$ interval of bunch crossing, a multi-level organisation is required. In order to cope with such requirements, a trigger consisting in two levels of consecutive event selection has been developed: the Level-1 trigger (L1) [22] and the High Level Trigger (HLT) [23]. The Level-1 trigger performs the first level of event selection using a hardware-based system implemented on custom electronics. It determines Regions-of-Interest (RoI) in the detector, based on coarse granularity information received from the calorimeter and the muon detectors, reducing the event rate from the LHC bunch crossing rate of approximately $40\,MHz$ to $100\,kHz$. The RoIs formed at Level-1 are sent to the second subsystem, the HLT, which performs the second level of event selection. The HLT is a software-based system, implemented on commercial CPUs and networking hardware. It performs the final online selection of events based on algorithms resembling the offline selection criteria. This selection reduces the rate from the Level-1 output rate of $100\,kHz$ to approximately $1\,kHz$ with average processing time of about $200\,ms$. The HLT trigger refines the decisions made at the previous level and, if necessary, it applies additional selection criteria. The data acquisition

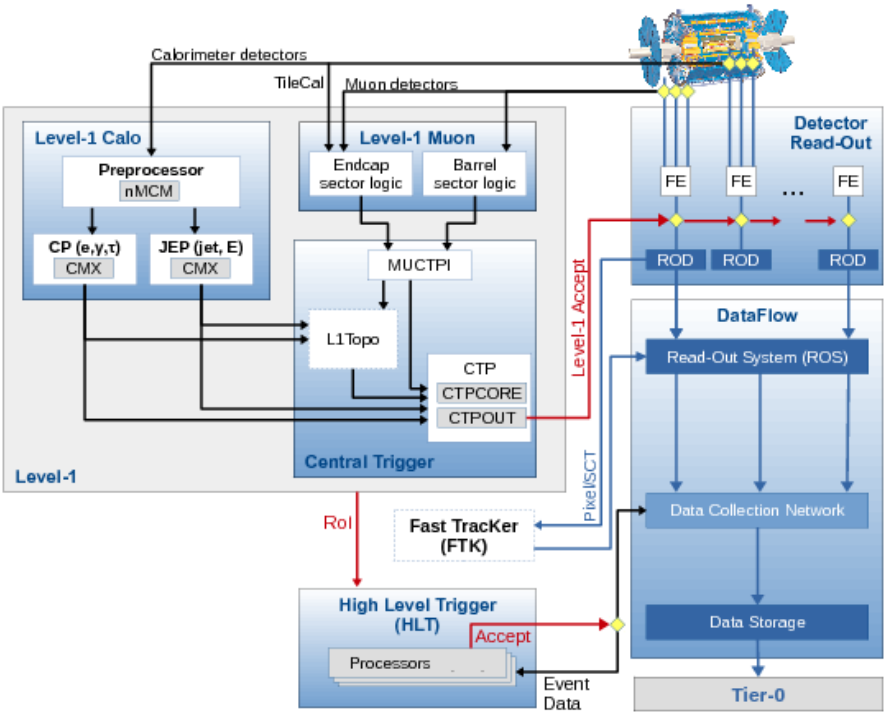


Figure 1.27: Schematic layout of the ATLAS trigger and data acquisition system.

system receives and buffers the event data from the detector-specific readout electronics, at the L1 trigger accepted rate. Figure 1.27 shows a schematic overview of the ATLAS trigger and data acquisition system.

1.2.6.1 *Level-1 trigger*

The Level-1 trigger operates at 40 MHz and it is able to reach a decision within a latency of less than 2.5 µs. It bases its selections on the search for signatures from high-p_T muons, electrons, photons, jets and τ-leptons as well as events with large missing transverse energy (E_T^{miss} or MET), and large total transverse energy. The architecture of L1 (shown in Figure 1.27) is based on three subsystems: the calorimeter trigger (L1Calo), the muon trigger (L1Muon) and the Central Trigger Processor (CTP). CTP receives information from L1Muon and L1Calo consisting of multiplicities for electrons, photons, τ, hadrons, jets and muons along with flags for total transverse energy, total jet transverse energy and missing energy. An acceptance signal (L1Accept - L1A) is then generated [24], derived from a collection of trigger inputs according to the L1 trigger menu. The latter contains 512 items, each of which is a combination of one or more trigger inputs. The L1A consist of a logical OR of all available trigger items. CTP generates an 8-bit trigger type word indicating the trigger type. At every L1A, CTP sends information to the Region-Of-Interest Builder (RoIB).

Because of the increased instantaneous luminosity and collision frequency obtained from LHC during Run 2, and because even more difficult conditions are expected during the future runs, two major hardware updates to the Level-1 trigger have been performed.

The first is the inclusion of a Topological Processor trigger (L1Topo). L1Topo accepts as input Trigger OBjects (TOB) that comprise the location, η and type of object identified. Based on this information, the full topology of the event can be reconstructed, enabling selections which are often applied at the analysis level.

The second major upgrade (which will be deployed only before the beginning of Run 3) is the inclusion of the Fast TracKer (FTK) system [25], currently in its commissioning phase. FTK will use data from the Pixel and SCT detectors to perform global track reconstruction after each Level-1 trigger, providing the High Level Trigger with early access to the tracking information. FTK moves the track reconstruction into a hardware-based system using precomputed patterns stored in custom associative memory chips for the pattern recognition. Instead of a computationally intensive helix fit, the FPGA-based track fitter performs a fast linear fit and the tracks are made available to the HLT. This system will allow the use of tracks at much higher event rates in the HLT than is currently affordable using CPU systems. More details about FTK will be given in Chapter 3.

1.2.6.2 *Level-1 calorimeter trigger*

The L1Calo trigger processes the signals from the calorimeters to identify electrons, photons, τ leptons, jets, and missing transverse energy with a maximum latency of 2.5 µs. To achieve the required latency, the 200 000 channels of the calorimeter are summed into 7168 trigger towers with a granularity of 0.1×0.1 in $\Delta\eta \times \Delta\phi$ in the range $|\eta| < 2.5$, while a coarser granularity is used beyond that range. These analogue signals are received by the Preprocessor, where they are digitised and correlated to the bunch crossing they belong to. The output of the Preprocessor is forwarded and used as the source of the Cluster Processor (CP) and the Jet/Energy-sum Processor. The results of these two modules are then forwarded to CTP for the final decision.

The Cluster Processor identifies electrons, photons, hadronically decaying τ leptons, and jets using both the electromagnetic and the hadronic calorimeters. To achieve this, a sliding window technique is used which examines 4×4 trigger towers, as shown in Figure 1.28. The technique used identifies clusters of 2 x 2 trigger towers in size in both calorimeters, known as Regions-of-Interest (RoI). These towers must contain a local maximum compared to their eight nearest neighbours, ensuring that candidates are not counted multiple times as a consequence of overlapping window arrangements.

For electrons and photons, the horizontal (2 x 1) or vertical (1 x 2) cluster sums with the highest energy in the electromagnetic calorimeter

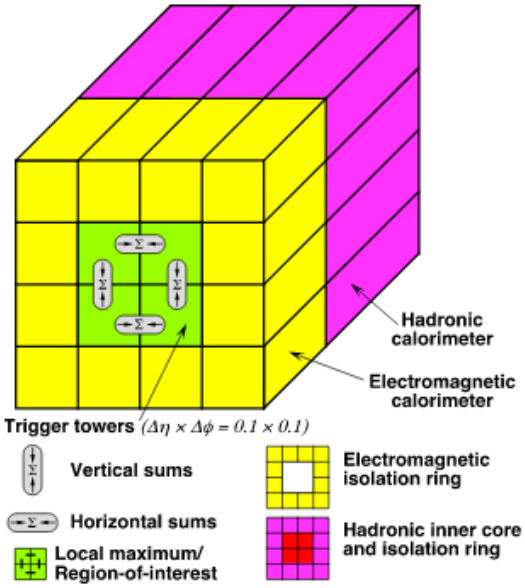


Figure 1.28: Schematic view of the trigger towers used as input to the L1Calo trigger algorithms.

must be greater than a predefined threshold. Further requirements, such as electromagnetic and hadronic isolation, can also be placed by requiring the energy in the ring surrounding the cluster to be below a predefined threshold. This helps to discriminate between single objects and broader hadronic jets. For tau lepton candidates, the EM cluster sum is combined with the hadronic core of trigger towers to produce a hadronic cluster.

The Jet/Energy-sum Processor uses coarser granularity information to identify jets and to calculate the MET, the total energy and the scalar sum of the total jet energy. The jet elements are formed by summing over 2×2 trigger towers and then sum over the depth of the electromagnetic and hadronic calorimeters. The algorithm uses the same sliding window technique as the Cluster Processor, calculating cluster sums of 2×2, 3×3 or 4×4 jet elements in size. These sums are compared to stored jet thresholds, different for each cluster size. The total and missing transverse-energy triggers cover the region out to $|\eta| < 4.9$ which is the limit of the coverage of FCal. Special care, however, is required after the range $|\eta| > 3.2$ where the granularity is much coarser.

1.2.6.3 *Level-1 muon trigger*

The muon trigger consists of three parts:

- the RPC system, triggering muons in the barrel region ($|\eta| < 1.05$);
- the TGC system, triggering muons in the endcap region ($1.05 < |\eta| < 2.4$);

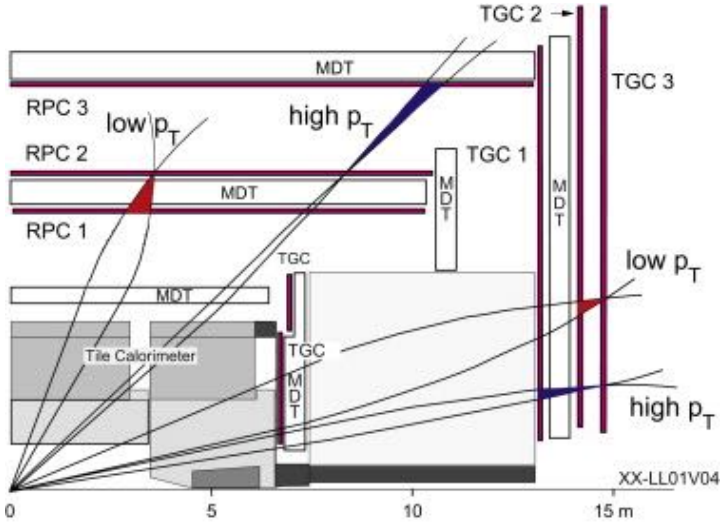


Figure 1.29: The layout of L1Muon showing the requirements for the different muon triggers.

- the Muon interface to central trigger processor (MUCTPI), that combines the information from the two systems and propagates the information to CTP and L1Topo.

Although different electronics are used in the RPC and TGC systems, both implement the same logic based on coincidence matrices. These matrices search for patterns of hits in successive layers consistent with high-p_T muons originating from the IP. The coincidences are valid if the hits involved lie within a common trajectory envelope of tracks above a given p_T threshold.

Both systems accept two different types of trigger: low-p_T and high-p_T. In the RPCs, a low-p_T trigger occurs when coincidence hits are present in at least three out of four layers in the external chambers. A high-p_T trigger further requires the existence of coincidence hits in one of the two inner chambers. Similarly, in the TGCs, the low-$p_T$trigger requires coincidence hits in at least 3 out of 4 layers while for a high-p_T trigger a further requirement is the existence of two more hits in the inner TGC. Figure 1.29 summarises the muon trigger architecture and the trigger requirements.

1.2.6.4 *High-level trigger*

The ATLAS HLT is a distributed software system performing the final online selection of events in real-time. The HLT selection are performed considering all the available detector informations within the RoIs, at full granularity and precision, reducing the required bandwidth to about 5 % of the bandwidth necessary for a full event reconstruction. If the selection is successful, the full event is then reconstructed. Originally, the RoI and the full-event processing where preformed in two separate farms, the Level-2 (L2) and the Event

Filter (EF), but after 2013 the two systems have been merged into a single homogeneous farm counting about 40 000 cores[26]. This merge allowed for a better resource sharing while minimising the network resources used, since a progressively increasing fraction of the event is fetched as needed. Finally, all the data fragments coming from different detectors are collected in a single file, which is transferred to the Data Logger for its storage in the local memories. The final event rate is of approximately 1 kHz.

2

THE TRACK RECONSTRUCTION AT LHC

The reconstruction of the particle trajectories from the observed detector hits is one of the most complex and computational demanding parts of the LHC beam-beam collisions data analysis. Moreover, due to its importance in particle identification and momentum and vertex estimation, as well as for pile-up suppression, the track reconstruction process plays a fundamental role in the online selection of the events. Besides the numerous examples that show the importance of tracking during the trigger phase, the prominent one is related to the study of the electroweak symmetry breaking process.

The electroweak coupling is proportional to the mass, thus, heavy fermions (in particular b quarks and τ leptons) are more likely to appear in the final states. High trigger efficiency for these processes requires sensitivity to the hadronic decays of the heavy fermions, that are hidden by the enormous background coming from QCD-produced light quark and gluon jets. This can only be suppressed using tracking information.

Tracks originating from a secondary vertex or not pointing to the beam line identify b quark jets, while τ jets can be separated from the background using the number of tracks in a narrow signal cone and the number of tracks in a larger isolation region. The efficiency of muon and electron triggers can also be further improved by using the tracking information. The traditional method for background suppression is the application of an isolation requirement by the calorimeters. However, at high luminosity, the increased energy deposit density results in either decreased lepton identification efficiency or increased background contamination. This effect can be greatly reduced by applying a track-based isolation using only tracks originating from the same vertex.

This chapter focuses on the strategies adopted or under development in the high energy physics experiments to achieve the reconstruction of the tracks. In Section 2.1 a general overview of the track reconstruction process in the LHC environment is given, focusing on the different track fitting and pattern recognition methods. In Section 2.2 a brief summary of the current algorithm limitations is presented, introducing some of the solutions the experiments are going to develop and adopt in Section 2.3.

2.1 TRACK RECONSTRUCTION

Track reconstruction at the LHC has to cope with the following conditions:

- high particle multiplicity due to the high collision energy and luminosity;
- large final state particle momentum range, from a few hundred MeV up to several hundred GeV;
- multiple scattering in non-sensitive material, such as detector frames, supports and cooling structures;
- secondary interaction activities to be recognized, measured and included in the final event reconstruction;
- high event rates leading to a large amount of data which need to be selected and analyzed in the most efficient way.

The track reconstruction usually consist of two steps: track finding (pattern recognition) and track fitting. The track finding and fitting steps are closely correlated, and usually used iteratively. As an example, the information obtained in a track fit can be used to recalibrate the measurement data.

Due to the topic centrality, a large variety of algorithms have been developed in the past fifty years, for both the steps [27]. In the following some of the most used ones will be presented, starting with a few more details about the track finding and fitting steps.

2.1.1 *Track finding*

In the track finding step, the pattern recognition problem is solved and the signals generated by charged particles in the tracker detectors are grouped into track candidates. In order to help fast processing, in general this procedure is performed using low resolution information, e.g. by grouping nearby hits in low granularity collections. The track finding algorithms can be separated in two main branches: *global* and *local* methods.

The global methods try to identify all the possible tracks at the same time, treating all the hits in a similar way. However, this can be very inefficient in terms of computational speed. Moreover, global methods require a detailed description of the detector layout, in order to let the algorithm know where the next measurement point is expected. Many detector layouts provide sufficiently continuous measurements so that the sheer proximity of hits makes it already likely that they belong to the same track. For this reason, local track finding methods are usually faster than the global ones.

In the local methods a single track candidate is processed at each time. Local methods usually need a *seed* as starting point, and they try to find additional measurements belonging to the track candidate iterating on the given detector hits. However, the results of these algorithms usually depend on the seed used. The local algorithms calculate the position at which the next measurement is expected based on the measurements already included in the track. If the seed track is wrong, the track can not be found. The local track finding methods are essentially based on the following three elements:

- a parametric track model, which connects a particle trajectory with a set of track parameters and provides a method of transport, i.e. extrapolation along the trajectory;
- a method to generate track seeds, i.e. rudimentary initial track candidates formed by just a minimal set of hits which serve as a starting point for the track following procedure;
- a quality criterion, which allows distinguishing good track candidates from fake tracks, so that the latter can be discarded.

In real life, a combination of global and local methods is used. For example, first a fast local method can be applied to find easy tracks with only few missing measurements. On the remaining measurements a global method can be applied. Moreover, it is also possible to apply a global method to create a track hypothesis needed for a local method. A global method can be used on the outer part of the tracking device to find track segments in this region, each consisting of only a few hits. The track segments can then serve as input for a local track finding method which step by step adds hits always closer to the center of the detector (towards regions where the track density is larger).

In the next paragraph, two examples of global track finding algorithms are presented. Since it provides both track finding and track fitting, the most used local track finding algorithm, the *Kalman filter*, will be instead treated toward the end of this Section (Subsection 2.1.3).

2.1.1.1 *Global pattern recognition*

The most widely used global pattern recognition methods are *histogramming* and *Hough-transformation*.

Thanks to its simplicity, the histogramming is the most used global track finding method in the high energy physics experiments. The points of a straight-line track present a fixed angle with respect to one of the axis. The measurements belonging to such a straight-line track will, thus, peak in a histogram filled with the angles of an event measurements. This concept can be easily generalized to helical tracks, using a proper transformation.

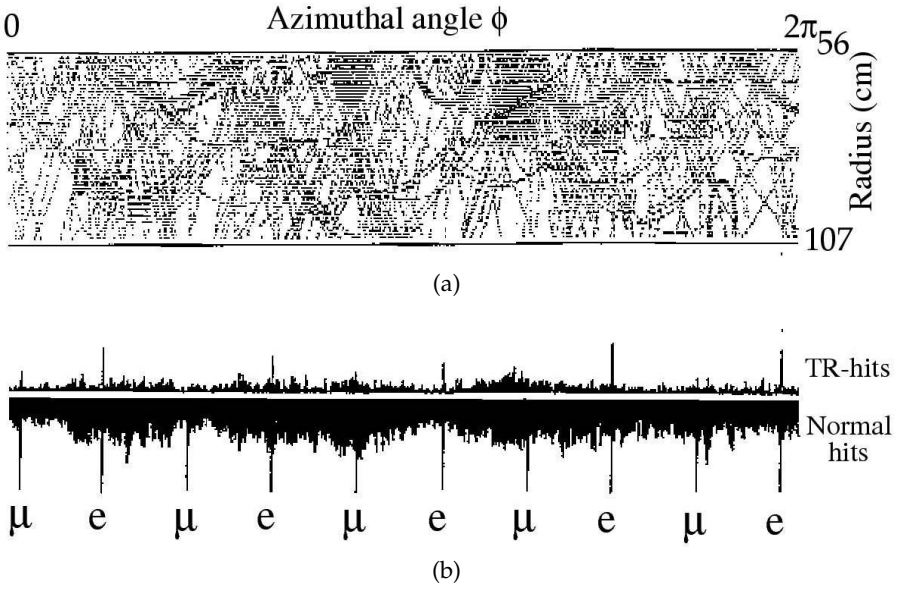


Figure 2.1: Histogramming for pattern recognition in the ATLAS TRT detector: Rϕ view of all the TRT hits (a) and histogram for muon and electron tracks with pile-up background (b). The upper part of the histogram contains the ϕ position of high threshold hits generated by transition radiation from electrons. The lower part contains normal tracking hits in the TRT. The histogramming technique allows to identify high momentum tracks and to distinguish between different kind of particles.

An example of such a histogram can be seen in Figure 2.1, where this technique is used to identify high momentum tracks in the ATLAS Transition Radiation Tracker. However, this simple histogramming does only work if the tracks are in good approximation straight-line or helical. If they are not, the histogrammed variable will not be constant and no clear structure will be visible in the histogram.

Histogramming can be regarded as a discrete implementation of the Hough transform [28]. In Figure 2.2, a sketch of the algorithm principle is shown. In the most general case, each measurement position is transformed from detector space into track parameter space, as can be seen from the upper part of the Figure. Each of these detector space points are represented in the parameter space by a curve (or in general a surface). After all the detector space points belonging to one track have been transformed, the point in which all the parameter space curves cross (the red dot in the lower part of the Figure) corresponds to the track parameter of that track. Completion of the pattern recognition task is thus converted into finding those points in the parameter space where many such curves or surfaces intersect. The Hough-transformed measurements can again be histogrammed and the track parameters of the track candidate extracted from the peaks in the histogram.

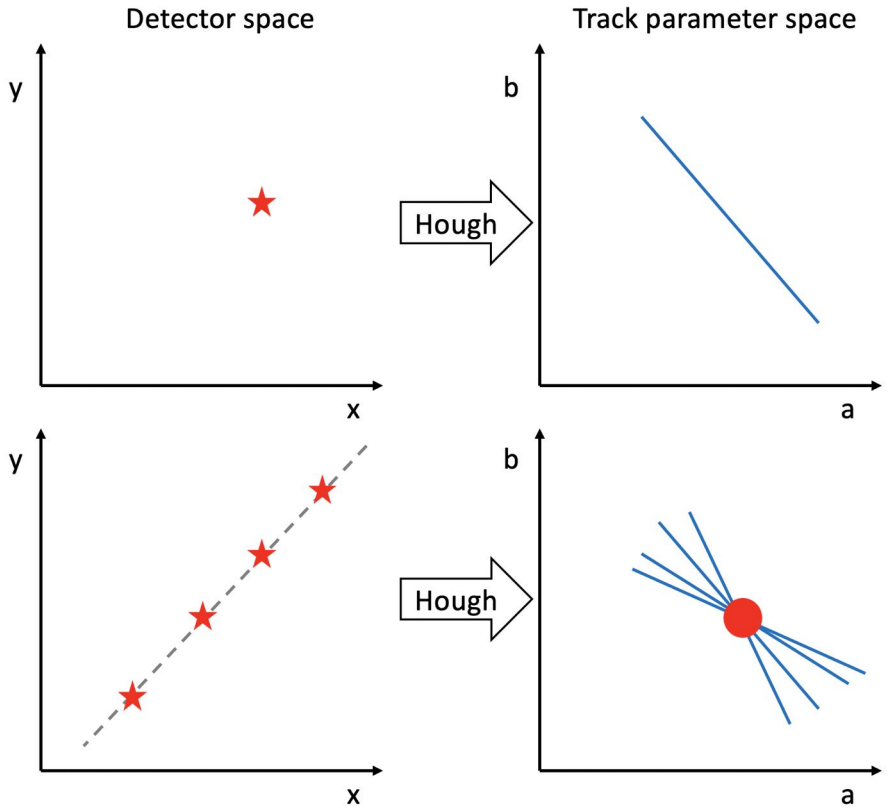


Figure 2.2: Sketch of the Hough transformation principle.

2.1.2 *Track fitting*

In the track fitting step, the best estimation of the track parameters is derived by error minimization, and the quality of the fit is computed. The track fit requires the knowledge of:

- the geometrical layout of the detector and the resolution of the sensors which provide the track measurements and their associated errors;
- a stochastic model of material effects, such as multiple scattering and energy loss;
- a track model, depending on the magnetic field.

From the physics point of view, there are a few aspects that are tightly connected with the quality of track fitting, such as:

- invariant masses must be determined with the best precision and well-estimated errors;
- secondary vertices must be fully reconstructed to estimate life-times on the order of 10^{-13} s, for example in the case of B mesons or τ leptons;

- an optimal track fitting requires a deep knowledge of the behavior of the detector as well as of the magnetic field, a precise treatment of multiple scattering and energy loss, and a track model which describes the trajectory of a charged particle.

In the following, the LHC experiments parametrization of the tracks will be briefly introduced, followed by an overview of the track fitting stage and a description of the *global fit* method (as an example of the fitting algorithms).

2.1.2.1 *Track parametrization*

The track parametrization in most of the LHC experiments assumes a homogeneous magnetic field with field direction parallel to the beam axis.

A charged particle in a static magnetic field $\vec{B}(\vec{r})$ without electric field moves under the influence of the Lorentz force $\vec{F}_L$:

$$\vec{F}_L = q\vec{v} \times \vec{B} \tag{2.1}$$

where $\vec{v}$ is the velocity of the particle and q its charge. The trajectory $\frac{\partial \vec{p}}{\partial t}$ has to obey the relativistic equation of motion:

$$\frac{\partial \vec{p}}{\partial t} = c^2 k q \vec{v}(t) \times \vec{B}(\vec{r}(t)) \tag{2.2}$$

where k is a proportionality factor, $\vec{r}(t)$ the spatial position of the particle at time t in the laboratory frame and c the velocity of light. This ordinary differential equation can be transformed into a geometrical form, which uses the path lenght s as a variable:

$$\frac{\partial \vec{r}^2}{\partial s^2} = \frac{kq}{|\vec{p}|} \frac{\partial \vec{r}}{\partial s} \times \vec{B}(\vec{r}(s)) \tag{2.3}$$

If one now assumes a homogeneous magnetic field along the z-axis, the equation 2.3 can be solved by a helix:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}(s) = \begin{pmatrix} x_0 + R_H \cdot (\cos\left(\alpha_0 + h s \cdot \cos\frac{\lambda}{R_H}\right) - \cos\alpha_0) \\ y_0 + R_H \cdot (\sin\left(\alpha_0 + h s \cdot \sin\frac{\lambda}{R_H}\right) - \cos\alpha_0) \\ z_0 + s \cdot \sin\lambda \end{pmatrix} \tag{2.4}$$

where $\vec{r}_0 = (x_0, y_0, z_0)^T$ is the starting point $s = s_0$, $\lambda = \sin^{-1}(\partial z/\partial s)$ is the slope angle, $R_H = \frac{|\vec{p}|\cos\lambda}{|kqB_0|}$ the radius of the helix and $h = -sign(qB_0)$ its sense of rotation. Using the helix solution one has to make a choice for the five dimensional track parameters $\vec{x}$ to be fitted. As an example, ATLAS uses several parametrizations in parallel. The perigee parameters d_0, z_0, ϕ_0, θ, and q/p are commonly used to express the track state next to the origin of the global coordinate system using the perigee point $P = (x_P, y_P, z_P)$, which is defined as the point of closest approach of a track to the global z axis. These parameters are defined as:

- $d_0 = \pm\sqrt{x_P^2 + y_P^2}$ signed transverse impact parameter;
- $z_0 = z_P$ longitudinal impact parameter;
- ϕ_0 azimuthal angle of the track direction at P;
- $\theta \rightarrow \cot\theta = \frac{p_z}{\sqrt{p_x^2+p_y^2}} = \frac{p_z}{p_T}$ polar angle of the track direction at P;
- $\frac{q}{p} = \frac{q}{|\vec{p}|}$ charged signed inverse momentum.

2.1.2.2 *The fitting stage*

For a given set of measurements the track parameters need to be estimated, i.e. a track fit needs to be performed.

Each fit requires an underlying *track model* describing the mapping $\vec{f}$ from the track parameters $\vec{x}$ into the measurement space. As seen in the previous paragraph, a helix may serve as a track model if considering a homogeneous magnetic field.

A track can be described at any point by a 5-component vector of track parameters $\vec{x}$, called the *state vector*, determined according to the constraints given by the equation of motion. The relation between the state vector and the observed n-measurements, described by the *measurement vector* $\vec{m}$, is

$$\vec{m} = \vec{f}(\vec{x} + \vec{\epsilon}) \tag{2.5}$$

where $\vec{\epsilon}$ is the vector of measurement errors and $\vec{f}$ is a deterministic function of $\vec{x}$:

$$\vec{f} : \vec{x} \rightarrow f_i(\vec{x}) \; i = 1, ..., n \tag{2.6}$$

The *covariance matrix* V of the measurements is defined as

$$cov(\vec{\epsilon}) = V \tag{2.7}$$

with the *weight matrix* G being

$$G = V^{-1} \tag{2.8}$$

The main task of the track fitting is to find a meaningful mapping $\vec{F}$ without bias and with minimum variance for the fitted parameters:

$$\widehat{\vec{x}} = \vec{F}(\vec{m}) \tag{2.9}$$

where $\widehat{\vec{x}}$ is the estimate of the state vector. The expectation value of the fitted vector $\vec{x}$ is, thus, supposed to be the true value $\vec{x}_{true}$

$$E\left(\widehat{\vec{x}}\right) = \vec{x}_{true} \tag{2.10}$$

with covariance matrix defined as

$$C(\vec{x}) = E\left((\widehat{\vec{x}} - \vec{x}_{true}) \cdot (\widehat{\vec{x}} - \vec{x}_{true})^T\right) \tag{2.11}$$

The fit methods can be divided into two groups:

- **Hard assignment methods**: the track is fitted considering a hard assignment of the hits to the track, i.e. a hit either belongs or does not to the track. These are also called classical methods and some of them are: the Global Fit [27], the Kalman Filter (KF) [29], and the Gaussian-sum Filter (GSF) [30].

- **Soft assignment methods**: several competing hits can contribute to the track, each of them with a respective assigned weight. The tracks can therefore share hits among each other. These methods are also called *adaptive methods*, but, since they are not significantly used by the LHC experiments, they will not be treated in this thesis work.

2.1.2.3 *The Global Fit*

The Global Fit is based on the Least Squares Method (LSM). In this method the errors are considered Gaussian distributed. The Global Fit is simpler and faster than other methods. The LSM-estimate $\widehat{\vec{x}}$ of the state vector is the value which minimizes the following function:

$$M(\vec{x}) = (\vec{m} - \vec{f}(\vec{x}))^T G (\vec{m} - \vec{f}(\vec{x})) \tag{2.12}$$

The weight matrix G in this case contains also multiple scattering effects which are included in the correlation among measurements. If the track model can be approximated by a linear model in the neighbourhood of the measurements, the function $\vec{f}$ can be written as an expansion around the point $\vec{x}_0$

$$\vec{f}(\vec{x}) = \vec{f}(\vec{x}_0) + H \cdot (\vec{x} - \vec{x}_0) + O((\vec{x} - \vec{x}_0)^2) \tag{2.13}$$

with

$$H = \frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}} \qquad (\vec{x} = \vec{x}_0) \tag{2.14}$$

Differentiating $M(\vec{x})$ with respect to $\vec{x}$ and putting $\frac{\partial \vec{M}(\vec{x})}{\partial \vec{x}} = 0$ yields the estimate of the vector state:

$$\widehat{\vec{x}} = \vec{x}_0 + (H^T G H)^{-1} H^T G (\vec{m} - \vec{f}(\vec{x}_0)) \tag{2.15}$$

with $(H^T G H)^{-1}$ the covariant matrix of $\widehat{\vec{x}}$. This procedure can be iterated to find also lower order estimates of the vector state. Among the different properties of the LSM for a linear model, it is worth to underline:

- if the measurement vector is unbiased, then the LSM-estimate of the state vector is also unbiased:

$$E(\widehat{\vec{x}} - \vec{x}_{true}) = \vec{0} \tag{2.16}$$

- the LSM-estimate is consistent;

- several methods were developed in order to reduce the effect of measurements that deviate from the expected behavior on the evaluation of the LSM-estimate. One example is to modify the error matrix ensuring a correct propagation of the errors. If the estimate is insensitive to this effect, the LSM-estimate is considered robust against outliers.

In the case of non-Gaussian errors and a non-linear model the LSM-estimate is asymptotically unbiased and consistent.

2.1.3 *Kalman Filter*

The Kalman Filter (KF) is a local track finding method. It performs the track finding as well as the track fitting by including new measurements step by step and updating the track parameters after each inclusion. It was originally developed to estimate the unobservable states of a stochastic model evolving in time (dynamic system). KF is based on a succession of prediction and filter steps. In the prediction step the current state vector is extrapolated to a future time, taking into account multiple scattering interaction and possible energy loss. In the filter step the extrapolated state vector is updated with the information of the closest local measurement to the predicted state. The distance is usually computed using the χ^2-statistic. This decision is purely local, and therefore does not take into account that other tracks can generate measurements even closer to the true one. In the case of low track density this poses no particular problem, but in the case of high track density global decision rules can be considered as an alternative. The application of the KF to the track reconstruction problem is straightforward when the track is modeled as a dynamic system:

- The state of the track at a specific surface k is given by the 5-component state vector $\vec{x_k}$. The evolution in time, i.e. the trajectory of the particle between two adjacent surfaces, is described by the following equation:

$$\vec{x}_k = \vec{f}_{k-1}(\vec{x}_{k-1}) + \vec{w}_{k-1} \tag{2.17}$$

  where $\vec{f}$ is a deterministic function which propagates the state vector from the surface $k-1$ to k and $\vec{w}_{k-1}$ includes the effect of the material on the trajectory. The latter is also called "noise process" in dynamic system language.

- The relation between the measurement $\vec{m}_k$ and the track state vector $\vec{x}_k$ at a surface k holds:

$$\vec{m}_k = \vec{h}_k(\vec{x}_k) + \vec{\epsilon}_k \tag{2.18}$$

with the measurement error $\vec{\epsilon}_k$ and $\vec{h}_k$ the function that maps the track parameters on the measurements, often a simple projection on a subset of the state vector. In other words, the measured quantities are expressed as functions of the state vector, corrupted by a measurement noise.

If $\vec{f}$ and $\vec{h}$ are linear functions, the Equations 2.17 and 2.18 can be rewritten as:

$$\vec{x}_k = \vec{F}_{k-1}\vec{x}_{k-1} + \vec{w}_{k-1} \tag{2.19}$$

and

$$\vec{m}_k = \vec{H}_k(\vec{x}_k) + \vec{\epsilon}_k \tag{2.20}$$

respectively. If $\vec{f}$ and $\vec{h}$ are non-linear, they can be approximated by their first-order Taylor expansion. The Jacobians $\vec{F}$ and $\vec{H}$ are computed as:

$$\vec{F}_{k-1} = \frac{\partial \vec{f}}{\partial \vec{x}} \qquad (\vec{x} = \vec{x}_{k-1}) \tag{2.21}$$

$$\vec{H}_k = \frac{\partial \vec{h}}{\partial \vec{x}} \qquad (\vec{x} = \vec{x}_k) \tag{2.22}$$

The two KF track fitting steps are performed as follow:

- **Prediction**: using the information from all the measurements up to $k-1$ and the magnetic field extrapolation, an estimation of the future state vector along with its covariance matrix is given:

$$\hat{\vec{x}}_{k,pred} = \vec{f}_{k-1}(\hat{\vec{x}}_{k-1}) + \vec{w}_{k-1} \tag{2.23}$$

$$C_{k,pred} = F_{k-1}C_{k-1}F_{k-1}^T + Q_{k-1} \tag{2.24}$$

  where Q_{k-1} is the covariance matrix of w_{k-1} and is assumed to be known, and C_{k-1} is the covariance matrix defined in Equation 2.11.

- **Filtering (update)**: the predicted state vector $\hat{\vec{x}}_{k,pred}$ is updated with a local measurement $\vec{m}_k$. The updated state vector on a detector surface with its covariance matrix can be written as following:

$$\hat{\vec{x}}_{k,update} = \hat{\vec{x}}_{k,pred} + K_k(\vec{m}_k - \vec{h}_k(\hat{\vec{x}}_{k,pred})) \tag{2.25}$$

$$C_{k,update} = (I - K_kH_k)C_{k,pred} \tag{2.26}$$

  where K_k is the Kalman gain matrix:

$$K_k = (C_{k,pred}^{-1} + H_k^TV_k^{-1}H_k)^{-1}H_k^TV_k^{-1} \tag{2.27}$$

  and V_k is the covariance matrix of $\vec{m}_k$.

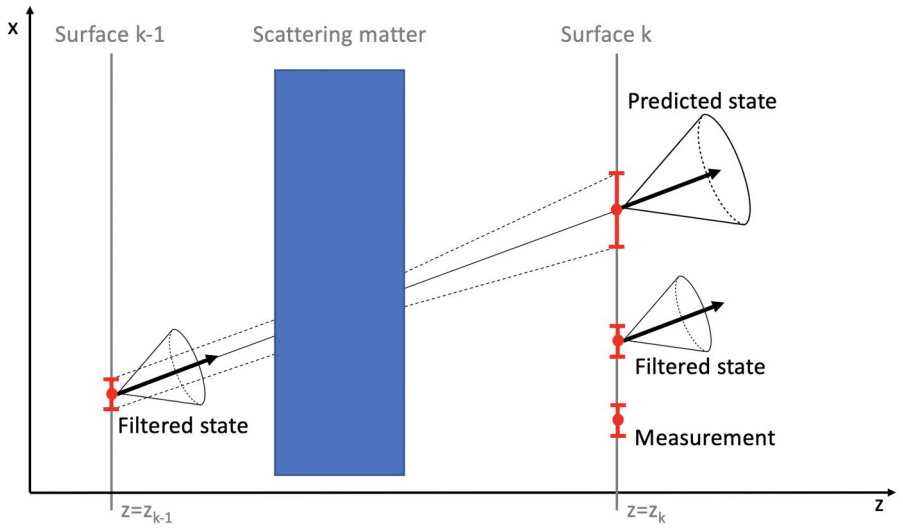


Figure 2.3: Schematic representation of the KF two-steps process. Starting form the filtered state on the surface $k-1$, the application of the prediction step provides the predicted state on surface k. Finally, the filtering step updates this state with the measurements information available, providing the final filtered state on surface k.

A schematic representation of both steps is given in Figure 2.3. If the dynamic system is strictly linear and $\vec{w}_k$ and $\vec{\epsilon}_k$ are Gaussian distributed, the KF is the optimal filter. In the case of non-Gaussian errors, the KF is the best linear filter.

When the last measurement is reached, the final estimate $\widehat{\vec{x}}_n$ contains the full information of the measurements $\vec{m}_1, ..., \vec{m}_n$. The information can then be passed back to all previous estimate state vectors in a finalization step, called smoother. The smoother step can also be done after each prediction/filter step using all measurements collected up to the present time. It can be implemented running two filters in opposite direction and combining both predictions with the local measurements. The advantages of the Kalman filter/smoother with respect to other fitting methods are:

- KF can be used in track reconstruction not only as fitting procedure, but also as track finder, thanks to its recursive nature;
- the computational cost is reduced because it is proportional to the number of surfaces crossed by the track and no large matrices have to be inverted;
- the estimated track parameters closely follow the real track;
- the linear approximation of the track model needs to be valid only between adjacent layers and not for the entire length of the track.

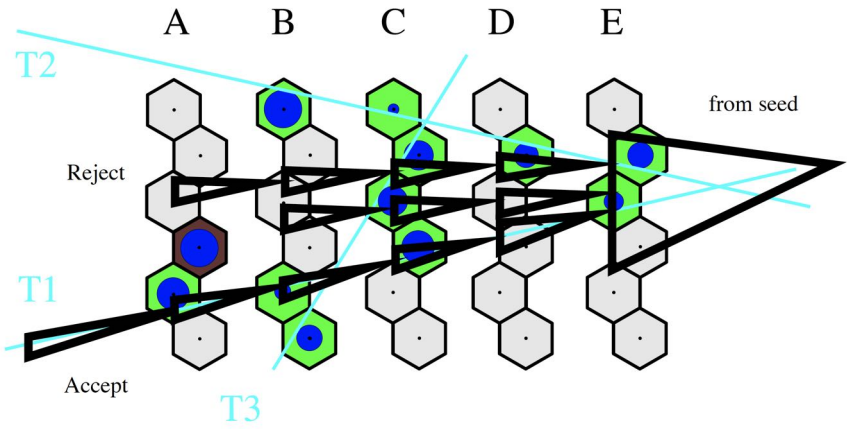


Figure 2.4: Sketch of the track search using a combinatorial Kalman filter. In the sketch the track search is performed along different branches starting from a given track seed.

It has to be noted that the KF needs an initial state vector to start, the so-called *seed*. Generally the seed is built where the track density is low or the granularity of the detector is high.

2.1.3.1 *Combinatorial Kalman Filter*

The recursive feature of the Kalman filter can be exploited to introduce track finding directly into the fitting process. In this case one starts with a seed and tries to follow the track candidate through the detector. The prediction on the next active detector surface is used to search for compatible measurement and the fit can in this case be split into several branches belonging to the different combinations. A selection criterion must be applied to the paths to keep the number of branches reasonable. One will drop paths which lead into regions where no compatible measurements can be found anymore while taking into account the possible detector inefficiencies and eventually add a selection based on the goodness of the fit. Figure 2.4 gives a simple example for such a Kalman-based track search.

2.1.4 *The ATLAS ID track reconstruction process*

The LHC experiments adopt almost the same strategy for track reconstruction: a combination of local and global pattern recognition algorithms and Kalman filters. After a pattern recognition stage, a track fit is finally used on the selected hits to perform the final estimate of the track parameters. Considering the ATLAS ID track reconstruction process as an example, it consists of several sequences, each of them exploiting different strategies [31]. The main sequence is referred to as inside-out track finding, which consists of the following different phases:

- **Data preparation and space point formation**: the initial step of the ID reconstruction consists of the cluster and drift circle creation in the silicon detectors and the transformation of these clusters into 3D space points. Clusters are formed by finding connected cells in the pixel and strip detectors. A neural network based cluster splitting module for the pixel detector [32] is used to identify and resolve merged clusters from close-by particles in dense environments. Space points are built by using the module surface and applying the coordinate transformation from the measured local to the global coordinate system.

- **Space-point seeded track finding**: track finding starts with the formation of space point triplets (seeds). To reduce the number of potential seeds, initial cuts are applied and special care is given to reduce the usage of space points in multiple seeds. Seeds that pass the initial requirements are then provided as input to a track finding algorithm that uses a combinatorial Kalman filter to complete the track candidates reconstruction within the silicon detector.

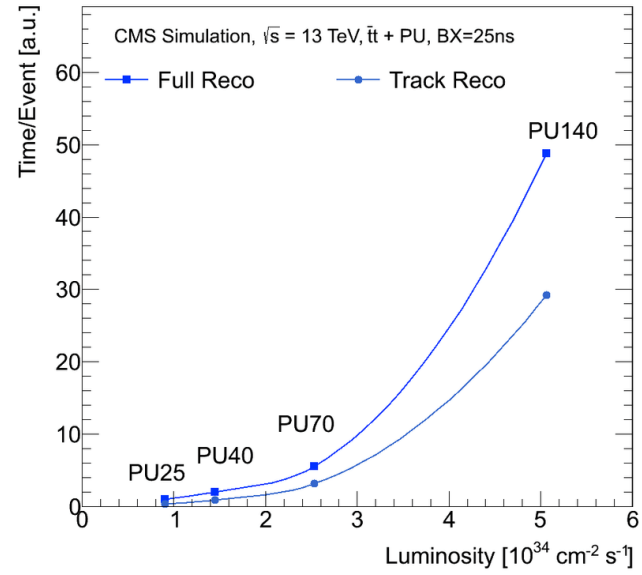
- **Ambiguity solving**: track candidates are then further processed in an ambiguity solving module whose goal is to eliminate track candidates from random hit combinations or track duplicates, which can be identified by hits shared with other track candidates. The ambiguity solving relies on a scoring function applying positive scores for unique measurements and good fit quality, while penalising missing measurements where they would be expected (holes) or shared measurements with other track candidates.

- **TRT extension**: tracks that successfully pass the ambiguity solving stage and are within the coverage of the TRT detector are finally extended into the TRT. A successful TRT extension increases the momentum resolution significantly by exploiting the longer lever arm for field integration.

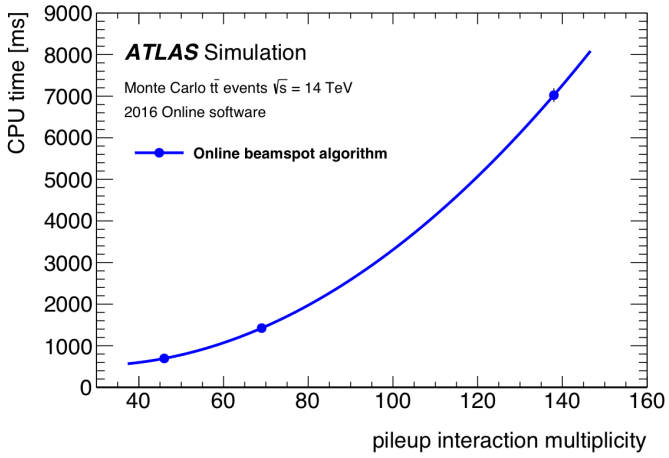
In the pattern recognition the fraction of layers with ambiguities grows with increasing pileup multiplicities and the task becomes combinatorially complex. The number of candidates can become much larger than the number of final tracks. Therefore, the computational cost of running the full KF-based pattern recognition may become prohibitive. This problem will be treated in more details in the next subsection.

2.2 THE TRACKING PROBLEM

As previously presented, the tracking process is one of the most computational demanding phase of both the offline and online analysis



(a)



(b)

Figure 2.5: (a) CMS track reconstruction time vs luminosity for $t\bar{t}$+pile-up samples with 25 ns bunch crossing. Samples with different average pile-up are used as reported on the plot. The timing of full CMS reconstruction (Full Reco) and of the iterative tracking sequence (Track Reco) are shown [33]. (b) The ATLAS trigger track reconstruction time for the beamspot reconstruction algorithm for 14 TeV $t\bar{t}$ Monte Carlo simulated with 46, 69 and 138 interactions per bunch crossing, measured on a 2.4 GHz Intel Xeon CPU. The used software version corresponds to the 2016 online trigger system. A second-order polynomial is used to fit the points.

of the LHC data. The standard reconstruction methods, presented in the Section 2.1.4, are currently performed on large CPU farms counting thousands of cores (e.g. the ATLAS HLT farm counts more than 40 000 CPU cores). Despite the big number of cores available,

the provided computational power is not enough to perform full track reconstruction for all the events. Already at the current LHC luminosity, the track reconstruction is performed at trigger level on a subset of preselected events only, and only inside preselected RoI (see Section 1.2.6) corresponding to no more than the 10 % of the total detector volume.

The tracking conditions will be even more critical during the future LHC runs, and particularly during the High Luminosity LHC era, where the collider luminosity is expected to drastically increase [34]. The new luminosity and increment in energy will lead to a higher number of pile-up interactions at each bunch crossing (up to $\mu = 200$), resulting in an increase in the number of tracks to be computed. Moreover, due to the combinatorial nature of tracking, the CPU time required for reconstruction doesn't scale linearly, but exponentially with the number of tracks, as can be seen from Figure 2.5. Assuming power density limitations from Moore's law, such a large increase is not sufficiently compensated by an increase in CPU clock frequency. For this reason, the reconstruction model currently used in most of the HEP experiments, based on traditional computers both for offline and online processing, cannot be sustained in the future without compromises between timing and physics performance that will impact the final sensitivity of the experiments.

In order to overcome this problem, a change in the reconstruction strategy needs to be performed, moving to a parallelization of the process and to the exploitation of co-processor systems more suitable than CPUs for the computational task. While some of the trigger-selection tasks executed in the farm necessarily require the flexibility and complexity that only a general-purpose CPU can provide, some other mechanical and repetitive computations can be very conveniently performed by specialized logic. The fast development that systems such as Graphical Processing Units (GPU) and Field Programmable Gate Arrays (FPGA) had in the last years, and the reduction in the production costs of Application Specific Integrated Circuit (ASIC) make their use in tracking applications very appealing. In Figure 2.6, a timing comparison between the use of CPUs and GPUs for the track seeding phase of the track reconstruction process is presented. Bringing that comparison as an example, moving specific computational tasks from CPUs to specialized electronic systems, can drastically reduce the time required for some of the reconstruction processes.

Beside the differences in the approaches that the experiments are developing/adopting, the common idea can be summarized in:

- parallelization of the work and subdivision of the detector volume in different "towers". Each tower will receive only the data coming from its selected $\eta - \phi$ region, splitting the computational cost between the region;

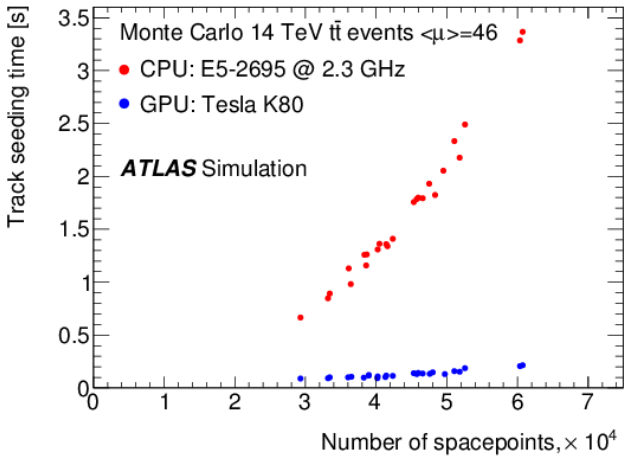


Figure 2.6: Timing of the ATLAS ID track seeding algorithm for the full detector. The red dots represent the standard HLT ID algorithm, running on a single CPU core, the blue dots show the same logic algorithm ported to GPU. The timing is shown as a function of the number of space points in an event.

- decomposition of the work in different substeps, exploiting the use of the processor systems that better fit the required task;
- merging of the different tower results and removal of the duplicate tracks.

2.3 ALTERNATIVE SOLUTIONS

In the following, some of the tracking methods adopted in the past, or currently under development for the future LHC runs, will be presented. Each of the presented algorithms tries to overcome the tracking CPU limitation problem making use of different approach.

2.3.1 *The Silicon Vertex Tracker*

One of the first application of a fully parallelized template matching method [35] can be found in the Silicon Vertex Tracker (SVT)[36], an hardware processor dedicated to the 2D reconstruction of charge particle trajectories for the level-2 of the CDF trigger [37]. This processor was based on a custom electronic system, the Associative Memory (AM), able to perform pattern matching on the whole memory content in a single clock cycle, overcoming the slowness of the standard template matching algorithms. The working principle of the SVT system can be summarized in two consecutive phases:

- a fast comparison between the detector hits and the precomputed tracks (patterns), able to find the candidate tracks;

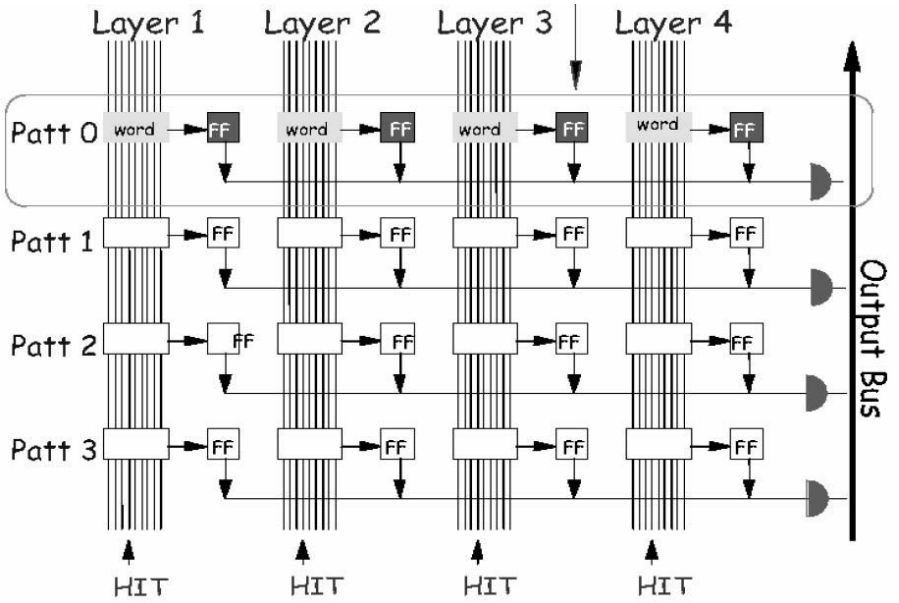


Figure 2.7: Sketch of an Associative Memory chip basic circuit.

- the track fitting on the matched hits, for the computation of the track parameters.

The pattern recognition phase is based on the AM chips, a highly parallel template matching system. When a track passes through a tracker, it leaves a unique pattern of hits on each detector layer. The principle of template matching pattern recognition is to store legitimate patterns for the possible interesting tracks (e.g. the tracks over a given p_T threshold) in a memory. Hits from an event are then compared to these pre-stored patterns. If one of the stored patterns contains all or a large percentage of detector hits, a track candidate (called road) is found. The AM is able to compare each set of hits with all the stored patterns in parallel, providing the high speed required for the trigger application of the template matching method. A sketch of the AM circuit is presented in Figure 2.7. With the AM, the pattern recognition is completed as soon as the last hit of an event is read. The found roads can then be immediately sent out sequentially for the track fitting phase. The SVT AM chips were able to store a total of 384 000 patterns, enough to cover more than the 95 % of the phase space for tracks with $p_T > 2\,GeV$. The SVT system was able to provide two-dimensional tracks with offline-like resolution within the 20 µs latency of the second level of the CDF three-staged trigger, processing the whole 50 kHz of L1 accept events rate.

The ATLAS Fast Tracker system, that will be treated in details in the next chapter, is a modern revamping of the SVT ideas. In order to avoid repetitions, details about the AM working principle and on the track fitting stage will be treated there.

2.3.2 *The Cellular Automaton algorithm*

The ALICE experiment solution for the online tracking relies on the use of GPU-accelerated algorithms based on the Cellular Automaton principle and the Kalman filter. The ALICE primary tracking detector is a *Time Projection Chamber* (TPC), that given the nature of the studied collisions ($Pb - Pb$), is characterized by a number of recorded hits one order of magnitude higher than the ones recorded by the other LHC experiments during $p - p$ collisions. In contrast to this high hit multiplicity, ALICE records heavy ion collisions at a much lower rate ($\approx 100\,kHz$), allowing the possibility to exploit the online reconstruction of all the tracks.

The track reconstruction algorithm starts with a combinatorial search for track candidates based on the Cellular Automaton method [38]. This method creates short three-hit track segments, called *tracklets*, in the neighbouring detector planes and links them into long tracks, as can be seen in Figure 2.8. The local pieces of trajectories are created from hits which are located nearby to each other, thus eliminating less probable hit combinations at the local level. The combinatorial processing is composed of two steps:

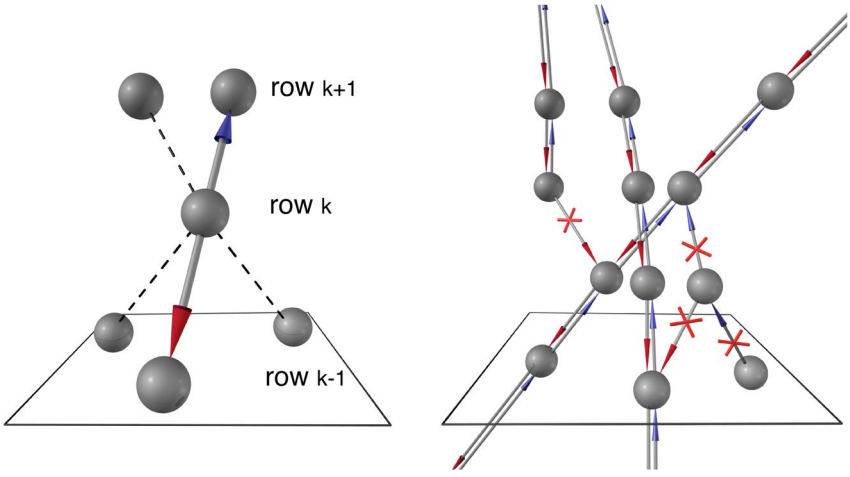
- Neighbour finder: for each hit in the k detector row, the best pair of neighbouring hits from rows $k+1$ and $k-1$ is found, as shown in Figure 2.8a. The neighbour selection criteria require that the hit and its two best neighbours form a straight line. The links to the best two neighbours are stored, and the step is completed when for each hit the best pair of neighbours is found.

- Evolution step: reciprocal links are determined and saved, and all the other links are removed, as shown in Figure 2.8b.

Every saved one-to-one link defines a part of the trajectory between the two neighbouring hits. Chains of consecutive one-to-one links define the tracklets. Because of the strong evolution criteria, each hit can only belong to one tracklet. Moreover, thanks to this hit abundance, during the pattern recognition phase it is not necessary to follow multiple track hypotheses. Therefore, all successive steps after the seeding have a linear run time with respect to the event size, as shown in Figure 2.9a [39]. Being essentially local and parallel, the cellular automaton method avoids exhaustive combinatorial searches, even when implemented on conventional computers. Since the algorithm operates with highly structured information, the amount of data to be processed durin the track search is significantly reduced.

When the tracklets are created, the sequential part of the reconstruction can take place. This phase is implemented as:

- Tracklet construction: the tracklets are created by following the hit-to-hit links as described above. The geometrical trajectories



(a) (b)

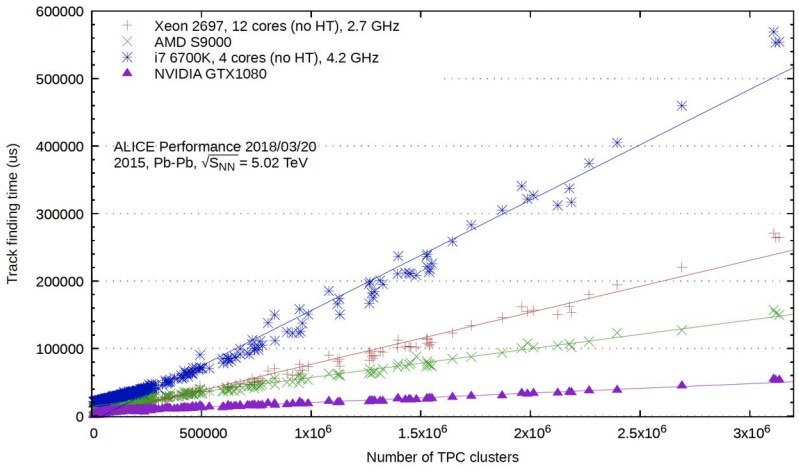
Figure 2.8: Illustration of the Cellular Automaton algorithm. (a) Neighbour finder. (b) Evolutional step.

are fitted using a Kalman Filter with a χ^2 quality check. Each tracklet is extended in order to collect hits close to its trajectory.

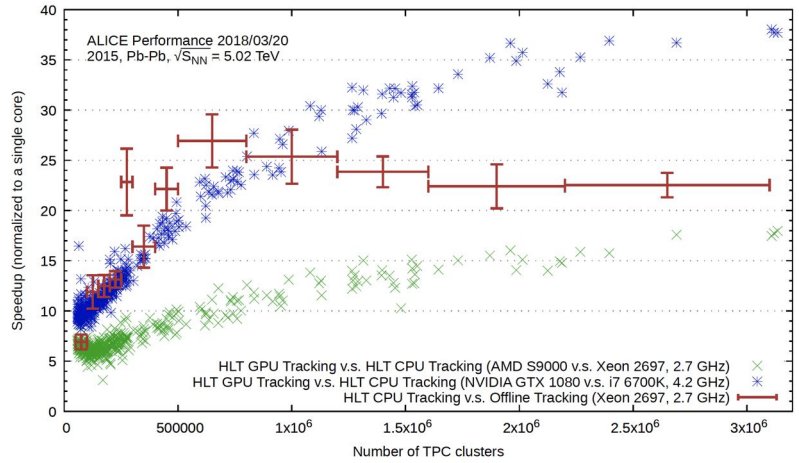
- Tracklet selection: some of the track candidates can have intersected parts. In this case the longest track is saved, while the shorter is removed. Finally, a last quality check is applied to the reconstructed tracks, including cuts on the minimal number of hits and on the momentum.

All the reconstructed tracks are eventually sent to the ALICE HLT for the final event selection. In order to take full advantage of the parallelism of the cellular automana algorithm, the track reconstruction procedure is executed on GPUs. This choice accelerates the track reconstruction by more than a factor 10 compared to a CPU solution running on four CPU cores. This can be observed in Figure 2.9b. The figure contains the profile for the speedup of HLT tracking versus online tracking in red. The speedup of a GPU versus a CPU core running the online tracking is therefore the product, which yields a speedup of up to 800 with a modern GPU and PbPb events versus a single core running the current online tracking.

In Figure 2.10 the efficiency and resolution of online and offline tracker are compared on Monte-Carlo data [40]. The online tracking efficiency is absolutely compatible to the offline. The online version even features a lower clone rate due to the approach with segment finding and merging, and slightly better efficiency for secondaries since the Cellular Automaton seeding applies only a loose vertex constraint. In terms of resolution, the offline version is always superior.



(a)



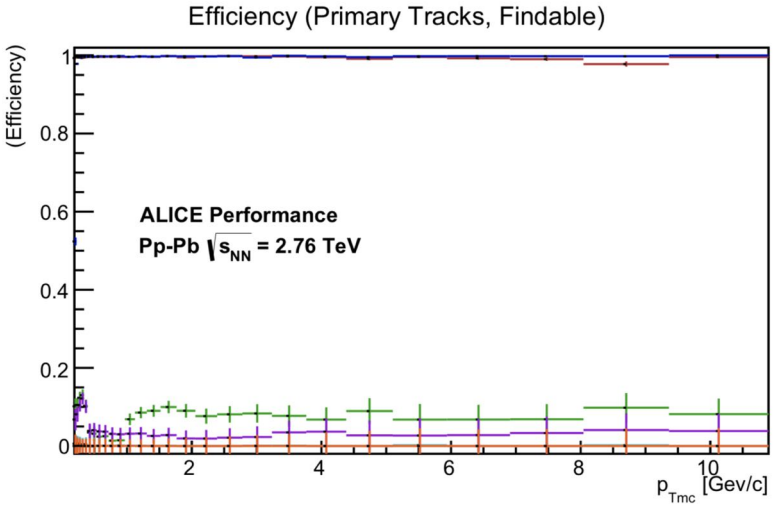
(b)

Figure 2.9: (a) Run time of the ALICE HLT TPC Tracking algorithm on CPU and GPU as a function of the event size. This plot shows that the tracking duration is almost linear both on CPU and GPU, irrespectively of the processor model. (b) Speedup of the ALICE TPC tracking on GPU normalized to a single CPU core.

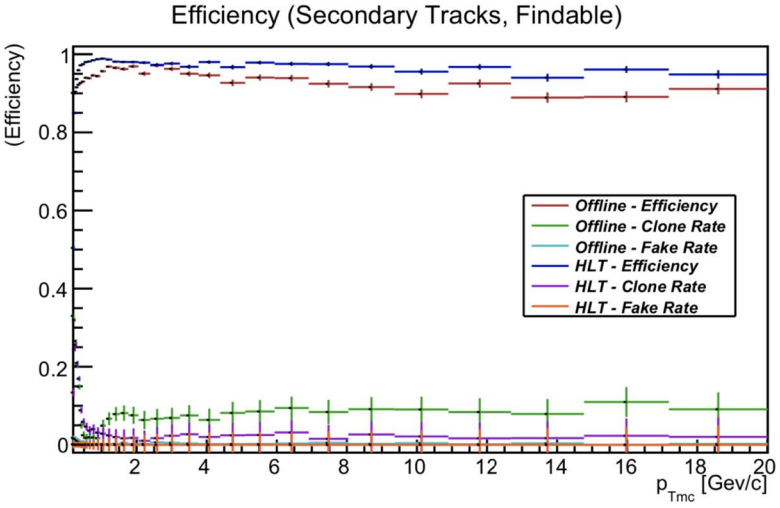
2.3.3 *The Artificial Retina algorithm*

The artificial retina algorithm, first proposed in 1999 [41], is a very fast, massively parallel, track reconstruction algorithm. The algorithm working principle takes inspiration from what is believed to be the low-level mechanism used by the mammalian visual brain areas to recognize lines and edges. This mechanism allows to recognize specific patterns in incoming data with throughput and latency performances largely superior to what has been achieved in artificial systems to date.

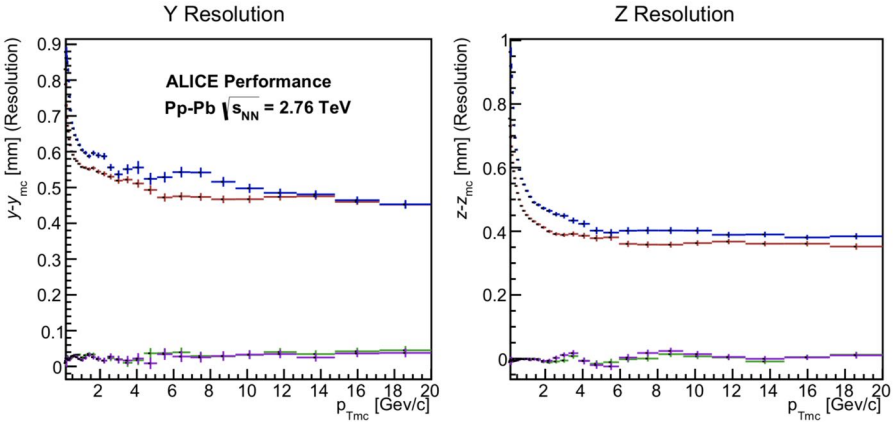
The algorithm working principle can be simply explained considering straight tracks traversing an array of n parallel detector layers. Given the coordinate of the hits in the detector, the algorithm will



(a)



(b)



(c)

Figure 2.10: Resolution and efficiency of ALICE online and offline tracking for TPC data only [40].

provide an estimate of the track parameters for each of the tracks that has generated the hits. Considering one transverse view only, a track can be described by two parameters, the coordinate of its intersection with the first and the last layer of the detector: P and Q, respectively. The two-dimensional parameter phase space is divided into a grid of receptive fields, called *cells*, labeled with a pair of parameters (P_i, Q_j). Each cell corresponds to a mapped track. The coordinates of intersection of a mapped track with the detector layers can then be defined as $t_l(P_i, Q_j)$ where l is the detector layer. The different layers are called receptors. For each incoming hit, the algorithm computes the excitation intensity of the cell corresponding to (P_i, Q_j)

$$R_{ij} = \sum_{l,r} exp\left(\frac{-d(x_r^{(l)}, t_l(P_i, Q_j))^2}{2\sigma^2}\right) \tag{2.28}$$

where $d(x_r^{(l)}, t_l(P_i, Q_j))$ is defined as the distance between the hit position x_r on layer l and the receptor $t_l(P_i, Q_j)$. The sum runs over all hits in all layers and the excitation R_{ij} is computed for all cells. The parameter σ is adjusted to optimize the sharpness of the receptors response. After all the hits have been processed, tracks are identified as local maxima in the cell space using a local cluster-finding algorithm. Since the response of each receptor is a smooth function of the coordinate of the hits, the excitation level can be used as a weight and the position of the center of the cluster can be obtained by interpolation. In this way, the precision on the track parameters is typically better than the pitch of the grid. Moreover, the needed computations can be performed in parallel over the array.

The retina algorithm is at the base of the *Track Processing Unit* (TPU), a new tracking processor implemented on commercially available FPGAs, proposed for the upgrade of the LHC-b experiment [42]. A visual representation of the TPU architecture can be found in Figure 2.11. Each cell is implemented as an independent logic block, called *engine* that performs autonomously all the necessary operations. The hits are made available from the detector into a custom switching network, that delivers them to all relevant engines in parallel, duplicating them as necessary. Local maxima are found in parallel in all the blocks, with some exchange of information between neighbors. The coordinates and excitation level of the local maxima, and the excitation level of their nearest neighbors are outputted sequentially. Finally, a parallel linearized fitter stage extracts track parameters from the cluster informations, making the reconstructed tracks available to the TDAQ system.

A feature of this approach, that deserves to be noted, is the peculiar organization of the overall system bandwidth. In the traditional trigger systems, the bandwidth is progressively reduced during processing, reaching slowly the desired output level. In the artificial retina algorithm, as shown in Figure 2.12, this is not the case. Due to the multiple

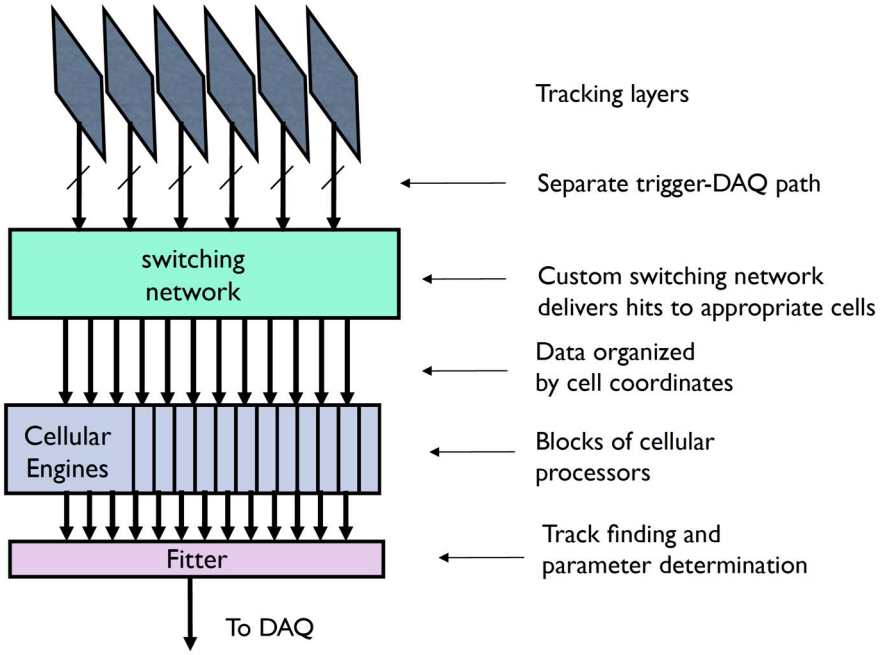


Figure 2.11: TPU architecture overview.

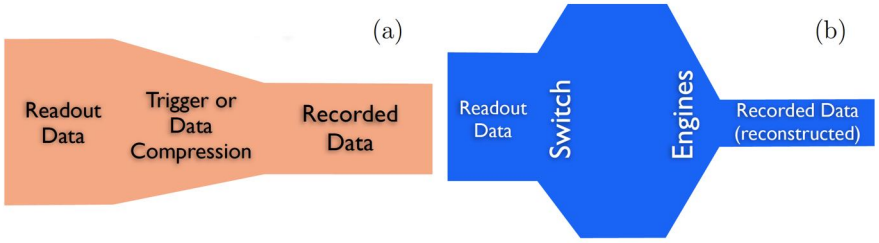


Figure 2.12: Sketch of the bandwidth flow in a generic trigger system (a) and in the Artificial Retina architecture (b).

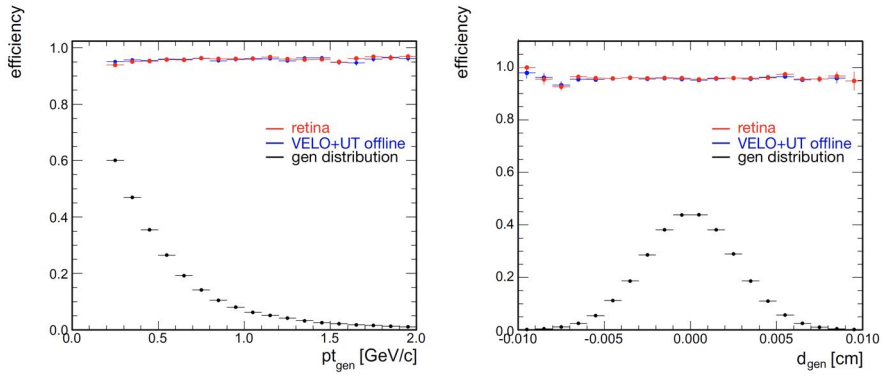


Figure 2.13: Tracking reconstruction efficiency of the retina algorithm (in red) and of the offline VELO+UT algorithm (in blue), as function of p_T and d. The distribution of the considered parameter is, also, reported in black. Luminosity of $L = 3 \times 10^{33}\,cm^{-2}s^{-1}$.

copies of the same data that are allowed to be produced, and are essential for the parallelization of the process, the bandwidth increases significantly shrinking down only at a later stage. This approach has only recently become technically feasible due to the progress of telecommunication technology.

Thanks to this system LHC-b is expected to be able to perform full track reconstruction, with offline-like performance, at the full 40 MHz collision rate. The evaluation of the performances has been performed using a detailed C++ simulation of the retina algorithm interfaced with the default LHC-b simulation [43]. In Figure 2.13 the retina simulated performance results are shown as a function of the p_T and d parameters, together with the efficiency of the current offline LHC-b track reconstruction algorithm. The retina algorithm shows very high efficiencies in reconstructing tracks, about 95% for generic tracks, which is comparable to the offline tracking algorithm. Resolutions on tracking parameters determined by the retina are comparable with those of the offline reconstruction.

2.3.4 *The tracklet algorithm*

In order to reduce the L1 output rate to the planned 750 kHz, also the CMS experiment plans to include track information at an early stage for the HL-LHC period [44]. The CMS track trigger is expected to receive data from the tracker front-ends at 40 MHz collision rate, and must reconstruct track in approximately 4 µs, in order to correlate tracks with other physics objects and to achieve a trigger decision. Before the start of the Run-4, CMS will substitute the whole Inner Tracker detector with a new silicon tracking system, consisting in a silicon pixel detector and a so called *Outer Tracker* (OT), a silicon microstrip detector already designed to provide tracking information

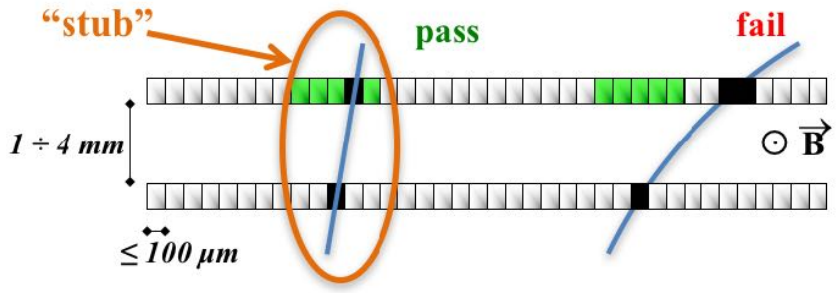


Figure 2.14: Sketch of the p_T modules that will be part of the new Outer Tracker for the upgrade of the CMS Inner Detector. The correlation of the two signals in the sensors allows the rejection of low-pT tracks using the selection window represented in green.

at the measurement stage. The CMS OT will have the unique feature of p_T modules that will provide p_T discrimination at the level of the front-end readout electronics. The p_T modules main idea is to correlate the signals coming from two closely-spaced sensors, in order to check their compatibility with the hypothesis of both coming from a track with a p_T above a certain threshold. The modules will be composed of two silicon sensors with a gap between the mid-planes of the active volumes. Both sensors will be read out by a common front-end electronics, capable to combine signal pairs. The compatibility of the signals is defined in hardware through a selection window. If the signal pairs are inside the selection window, they are combined into *stubs*. Stubs are then dispatched to the L1 trigger system at every bunch crossing for the trigger selection. All the other hits in the OT modules are instead stored in the front-end pipelines and read out only when a trigger is received. A simplified illustration of the concept is shown in Figure 2.14.

The selected stubs will be required to be consistent with a $p_T > 2\,GeV$ track originating from the interaction point. Since most minimum bias events have low p_T tracks, this will provide a factor of 10% data reduction, already before the application of other selection criteria. Moreover, the p_T modules will provide a precise measurement of the displacement along the beam line (z-axis), which will enable primary vertex reconstruction at the L1 trigger level.

In order to take full advantage of this new p_T modules also the tracking reconstruction algorithms will be completely redesigned, moving to a new *tracklet* algorithm [45], completely implemented in off-the-shelf FPGAs. Figure 2.15 shows a sketch of the working principle of this new algorithm. The tracklet approach starts with forming track seeds (tracklets) from pairs of stubs in adjacent detector layers. The tracklet is an initial estimate of the tracklet parameters calculated from these two stubs using the interaction point as a constraint. A tracklet candidate is required to be consistent with a track originating within

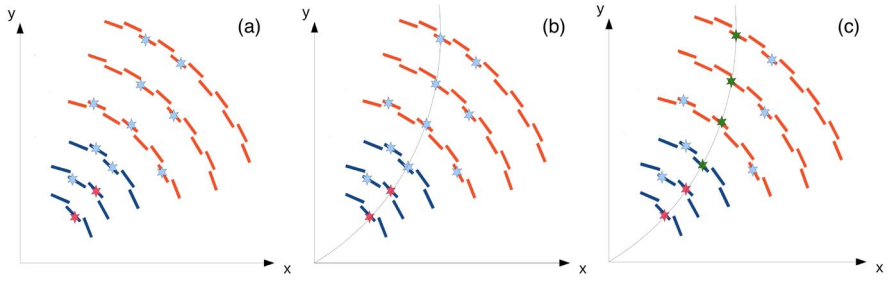


Figure 2.15: In the first step (a) of the algorithm a pair of stubs (red) in adjacent layers are combined to form a tracklet. The trajectory of the tracklet is projected (b) to the other layers. Stubs in the other layers that are close to the projection (green) are selected as matched (c) to the tracklet. Final track parameters are calculated using all associated stubs.

$|z_0| < 15\,\mathrm{cm}$. The seeding is performed for several combinations to provide good coverage of the entire pseudorapidity (η) range of the detector. The track parameters of the tracklets are then projected to other layers and disks to search for consistent stubs. When the tracklets are projected to other layers, the search for matching stubs occurs in predetermined searching windows, derived from residuals between projected tracklets and stubs. The projection of the tracklets occurs both inwards and outwards, i.e. to and from the interaction point. If a stub is found consistent with the original tracklet parameters, the matched stub is included in the track candidate and the difference between the projected tracklet position and the matched stub position is stored. A linearized χ^2 fit is performed using all stubs in the track candidate, i.e. the stubs used to make the original tracklet plus the matched stubs. The track fit uses pre-calculated derivatives and the projection-stub differences. The linearized χ^2 fit corrects the initial tracklet parameters giving the final track parameters. Because seeding is performed for multiple seeding combinations, a single track may be found several times. Duplicated tracks are removed by comparing the number of independent and shared stubs in pairs of tracks.

To address the challenging amount of data and limited available processing time, the tracklet hardware configuration massively parallelizes the data processing. The main parallelization is the division of the detector into sectors in the $r-\phi$ plane. The current project uses 28 ϕ sectors. To allow for more time for data processing, the whole system is time-multiplexed by a factor of 6. This choice is driven by a balance of cost, efficiency and needed processing power. The tracklet algorithm is implemented in the firmware subdivided in different processing steps. Currently all processing steps are synchronized to a single master 240 MHz clock. By construction, the system is fully pipelined and operates at a fixed latency: when a new event arrives the previous event moves to the next processing step. Because of that,

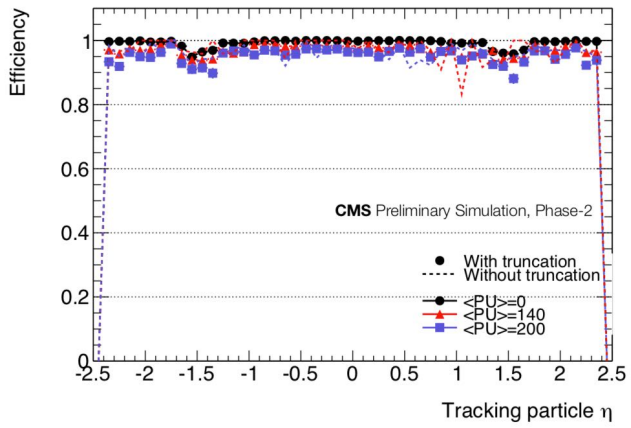


Figure 2.16: Efficiency of the tracklet algorithm as a function of η for muons. Results are shown for three different average pileup scenarios: 0 (black), 140 (red), and 200 (blue). Efficiencies are shown with (filled points) and without (dashed lines) truncation included.

a given step can only perform a fixed number of operations. If the time limit is reached, the processing on the remainder of the data is truncated. The effect of this truncation on the system is minimal, and the total tracking efficiency, as expected from an integer-based C++ emulation of the algorithm, is higher than 95 % [46], as shown in Figure 2.16.

3

THE FAST TRACKER SYSTEM

The LHC is currently in a shutdown phase, during which maintenance and upgrades to both the accelerator systems and its experiments will be performed. The shutdown periods are scheduled regularly, lasting for 2 years and followed by 4 years of data taking sessions. The current shutdown, started in 2019, is called *Long Shutdown* 2 (LS2), and the 4 years data taking period that will follow is called *Run 3* (from 2021 until 2024). The Fast TracKer (FTK) system is among the upgrades of the ATLAS TDAQ system of the LS2.

FTK is an hardware based tracking system designed to perform full scan tracking, reconstructing all the tracks with $p_T > 1$ GeV at the full L1 trigger rate (100 kHz). The reconstructed tracks will be provided to the HLT with an average latency of 100 µs, in time for the online trigger selection of the events [47].

The FTK is a very complex system, composed of a total of about 450 electronic boards based on two different standards: VME (VERSA-BUS Module Eurocard [48]) and ATCA (Advanced Telecommunications Computing Architecture [49]). The system counts about 500 input links bringing the data from the silicon detectors to the FTK system, and a total of about 10 000 links among the FTK boards themselves.

FTK processes data from the pixel and SCT detectors as well as the IBL pixel detector, and moves the track reconstruction process into a massively parallel processing system, overcoming the CPU constraints of online tracking, already presented in Section 2.2. Its working principle relies on the same concept as the CDF Silicon Vertex Trigger (SVT), presented in Section 2.3.1.

The FTK track reconstruction process is executed in three separate steps:

- reorganization of the ID data in different processing regions, for the parallelization of the tracking process;
- very fast, low resolution pattern recognition, executed by Associative Memories (AM);
- two step full resolution fit, executed by FPGAs.

In Figure 3.1, a sketch of the FTK architecture is presented. The system is composed of 6 different kinds of electronic boards, each of them designed to accomplish a different sub-task of the tracking process [50].

The **Input Mezzanine** (IM) and the **Data Formatter** (DF) boards are designed to receive the hits from the 12 layers of the ATLAS Pixel and

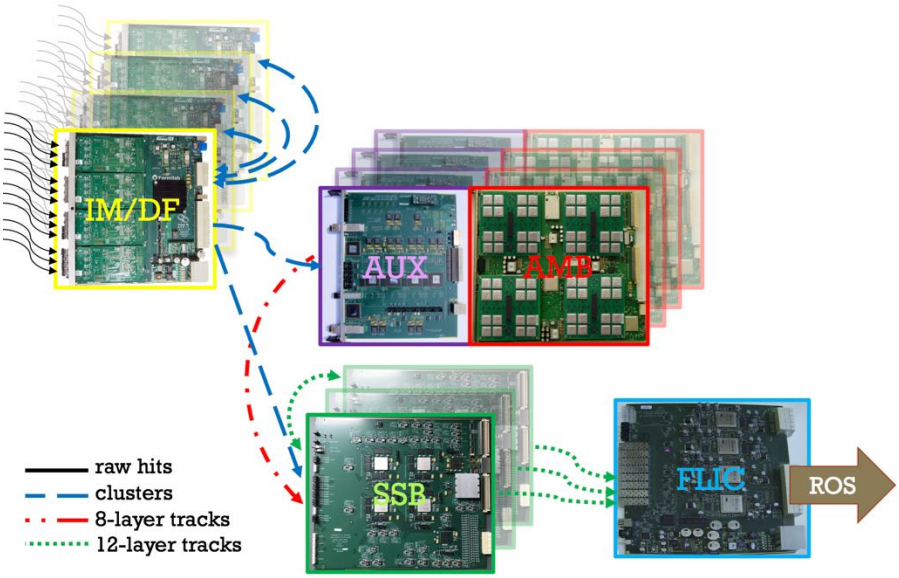


Figure 3.1: Sketch of the internal architecture of FTK. The lines represent the communication links between different boards [50].

SCT detectors and to group them into clusters of nearby hits, in order to reduce the data size. Moreover, the clustered hits are sorted into different $\eta - \phi$ regions, before being sent to the downstream boards for processing. This processing stage is required for the parallelization of the tracking process. Dividing the data in different regions let the processing boards work in parallel on their own portion of detector data, drastically reducing the processing time of each event. Since the data sorting requires a huge exchange of information between the different DF boards, these boards exploit the ATCA technology as housing infrastructure.

The **Processing Unit**s (PU), composed of the **Auxiliary Card** (AUX) and the **Associative Memory Board** (AMB) pairs, receive the given regional data for only 8 of the 12 ID layers. They then perform the pattern matching procedure and a coarse fit on the found candidate tracks.

The AUX is responsible for processing the received hits by grouping them in coarser resolution segments, and to send them to the AMB for the pattern recognition stage. The AMB compares the received segments to Monte Carlo (MC) track patterns stored in the Associative Memories. The AMs are the core component that allows FTK to cope with the rigid time constraints given by the online environment, performing the pattern recognition in parallel on all the memory lines and finding the candidate pattern (if any) in a single clock cycle. These components are able to process hundreds of millions of candidate tracks nearly simultaneously as the silicon detector data pass through FTK. When the AM finds a MC track pattern that matches the data, returns it back to the AUX that performs a fast goodness of fit. Due

to the size and to the high density of AM chips contained in each AMB, the board would not fit in an ATCA shelf. Because of this, the PU boards, as well as the SSB boards described in the following, are housed in VME crates.

The **Second Stage Board**s (SSB) combine the received 8-layer tracks, provided by the PUs, with the hits from the remaining four layers of the ATLAS ID, provided by the DF boards, and perform a full twelve-layer track fit.

Finally, the **FTK-to-Level2 Interface Crate** (FLIC) boards collect all the twelve-layer tracks coming from the different processing regions and send the final tracks to the HLT for the trigger selection.

In the firsts two sections of this chapter, the different FTK hardware components and their functionalities, together with FTK housing infrastructure, will be described in some detail. In Section 3.3, the commissioning status of the FTK system will be treated, with particular focus on the different strategies used to integrate such a complex system in the ATLAS data taking infrastructure. Finally, in Section 3.4 the performances of the already commissioned part of the system will be presented, followed by a quick summary of the outcome of the review FTK underwent in the last months, in Section 3.5.

3.1 FTK ARCHITECTURE

In Figure 3.2, a sketch of the FTK dataflow is presented. FTK is composed of a huge number of electronic elements organized in pipelines connected by thousands of specialized links: 16 400 Associative Memory chips, i.e. custom designed ASICs, are used to perform pattern matching, while about 2000 FPGAs are used for the detailed track fitting and the other data manipulation tasks. In the following a description of the various FTK functionalities will be given in some detail.

3.1.1 *The data formatting*

After each L1 trigger accept, the ATLAS Inner Detector sends the silicon hits to the DAQ Read-Out-System (ROS) by the Read-Out-Drivers (RODs). The transmission of the data is performed via optical fibers, using the S-LINK protocol [51]. In order to provide the FTK input with a copy of the ID data, the old HOLA (High-speed Optical Link for ATLAS) cards, responsible for the data transmission from the ID RODs to the ROS, have been substituted with new Dual-output HOLAs.

The FTK input is composed of 32 DF boards and their 128 IM mezzanine cards. A DF board, completed with its 4 IMs, is visible in Figure 3.3. The IMs are connected to 4 S-LINK fibers coming from

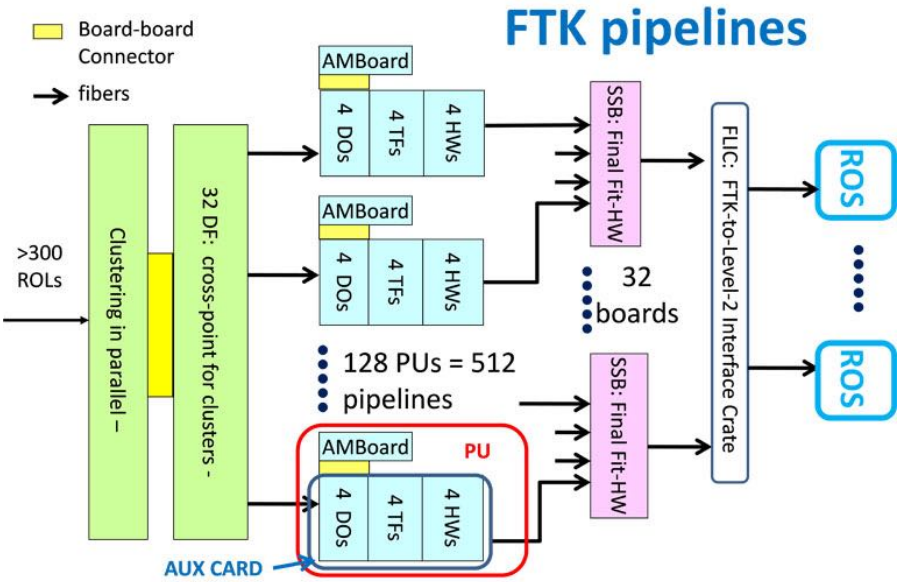


Figure 3.2: Sketch of FTK seen as a set of 512 pipelines up to the final 12-layer fit in the SSBs. The black arrows show the connections trough fibers and the yellow rectangles show high frequency connections between mezzanine cards and motherboards or between two boards in the same crate slot.

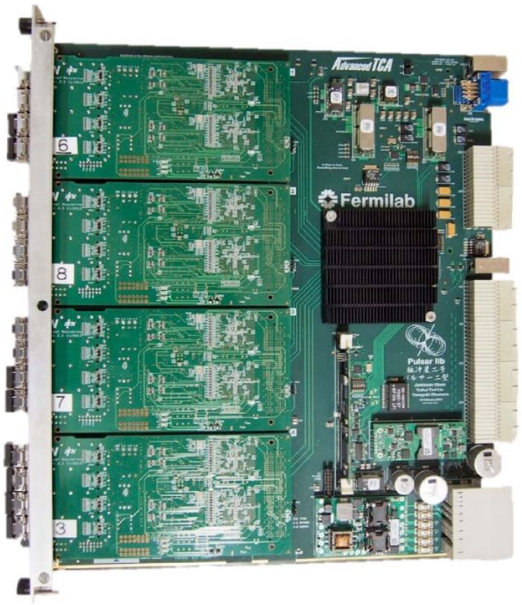
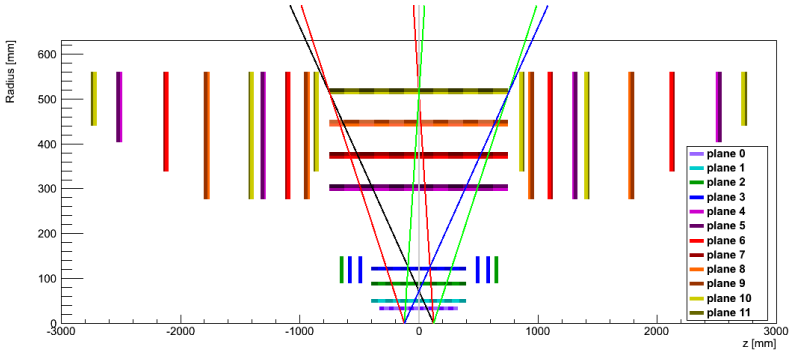
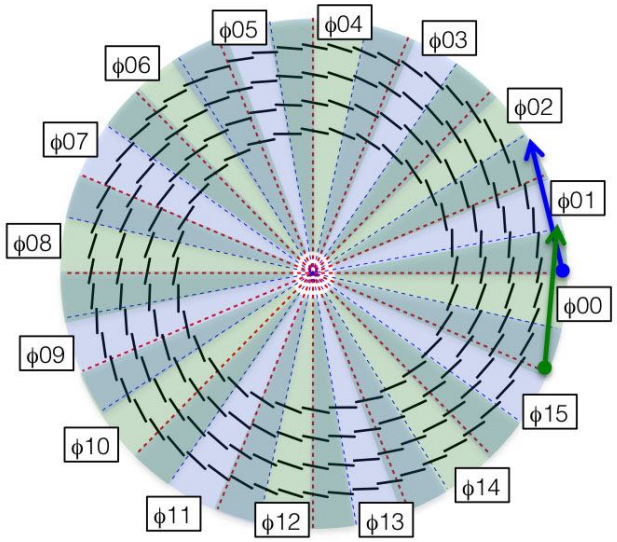


Figure 3.3: Picture of a FTK DF board. The 4 different Input Mezzanines (cards number 3, 6, 7, and 8) are also visibles. This board, together with the IMs, is responsible for the reorganization and formatting of the ID data for the FTK processing.

the ID RODs (256 from pixel and 256 from SCT). After receiving the input data, the IMs perform one or two-dimensional cluster finding in the SCT or in the Pixel/IBL detectors, respectively. Clustering is



(a)



(b)

Figure 3.4: (a) The assignment of barrel layers and end-cap disks to FTK logical layers. Layers 0-3 are pixels, the rest are SCT. The division into 4 η regions with the appropriate overlap is indicated by the thin colored lines: two endcap regions, one to the left of the black line and the other to the right of the blue line; and two barrel regions, one between the red lines and the other between the green lines.
(b) The division into the 16 detector ϕ coordinate. The green line shows the coverage of the sector ϕ00 and the blue line the coverage of sector ϕ01. The sectors overlap to avoid inefficiency due to low curvature tracks.

performed with the dual purpose of reducing the amount of data to be processed by the downstream FTK boards and of determining the cluster center in order to improve spatial resolution. After the processing, the cluster centroids are sent to the Data Formatter main board through a High Pin Count (HPC) FMC connector [52].

The DFs reorganize the ID data into projective $\eta - \phi$ towers and distribute the cluster centroids to the given PUs the data belongs to, where pattern matching and track fitting take place. The division of the

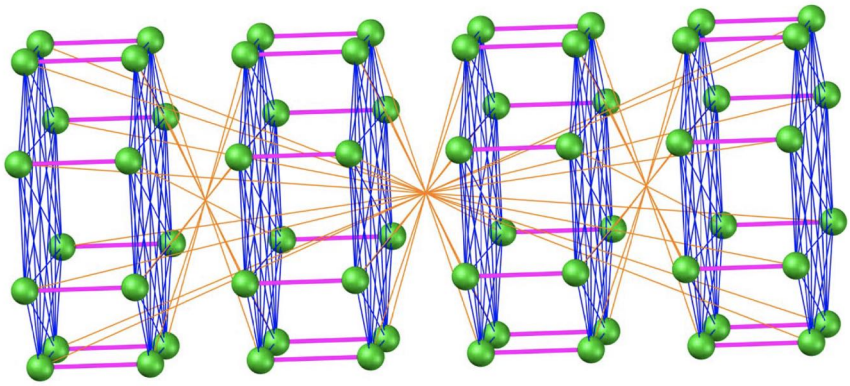


Figure 3.5: The interconnections for data sharing among the 64 $\eta - \phi$ towers within 4 DF crates. A green ball represents one FTK tower. Blue lines represent data sharing between the upper FPGAs or lower FPGAs in a shelf. The red lines represent data sharing between an upper and a lower FPGA, and the yellow lines represent inter-shelf data sharing.

FTK system into $\eta - \phi$ towers is done to achieve maximum processing parallelization. In total, 64 $\eta - \phi$ towers are used, 4 in η, as shown in Figure 3.4a, and 16 in ϕ, as shown in Figure 3.4b. The regions in ϕ overlap to minimise the inefficiency due to the finite curvature of low p_T tracks. The regions in η overlap due to the finite dimensions of the portion of space along the z axis in which interactions take place.

There are 8 $\eta - \phi$ towers in a crate, with a 10° overlap between neighbouring crates in ϕ. The data of a given tower are processed in the DF by a single specific FPGA. Two processing FPGAs are housed in each DF board, with a total of 32 DFs housed in 4 ATCA crates, called shelves. Figure 3.5 shows the different interconnections used for the sharing of the data between the different DFs.

Three different types of internal data-sharing paths are used: a local bus that connects the FPGAs on the same Data Formatter board, multiple point-to-point high speed serial links in the full-mesh ATCA backplane connecting the shelf FPGAs, and fiber optic transceivers used for the inter-shelf connection.

A Rear Transition Module (RTM) is used to send the data downstream as well as to perform the inter-shelf data sharing. RTM boards support up to eight QSFP+ [53] and six SFP+ [54] optical transceivers, half of them directly connected to the upper/lower FPGA. Through the RTM the ID hits from a given tower are sent to the PUs (SSBs) for the first (second) stage tracking.

3.1.2 *First stage processing*

The PU boards, housed in the VME core crates, are designed to perform the pattern matching and the first stage track fitting procedures.

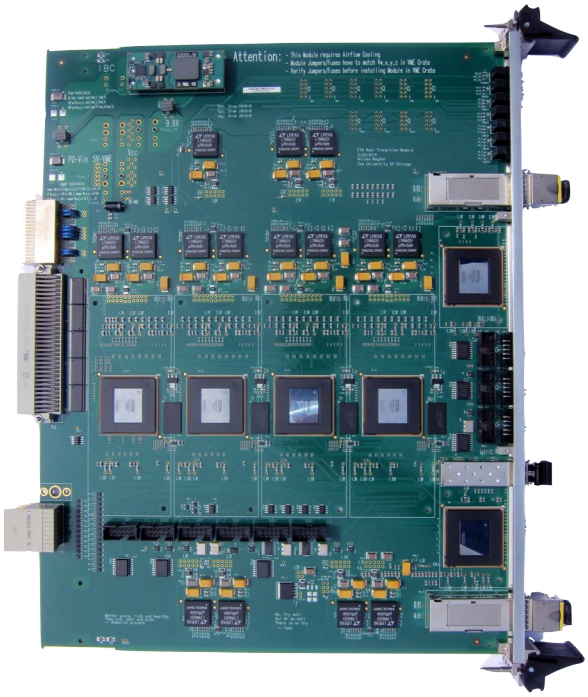


Figure 3.6: Picture of an FTK AUX card. This board is responsible for the first stage processing of the tracks.

Each PU consists of an Associative Memory VME card with a large auxiliary board behind it, with a total of 128 pairs, each of them analyzing the data of half an FTK tower.

The 3 pixel hits and the first 5 SCT layers hits are received from the DF by the AUX card. A picture of an AUX card can be seen in Figure 3.6. The input element of the AUX is the Data Organizer (DO) FPGA. Here, a smart database allows the rapid storage of hits in a candidate track, called *road*, and sends the hits (called *Super-Strip* or SS) to the AMB. In order to reduce the possible number of tracks to be matched inside the AMs, the SS are built-up from the DF hits by merging groups of hits in coarse resolution series. More details about the pattern matching phase will be given in Subsection 3.1.3. The level of reduction in resolution of the SS is optimized to compensate the finite number of patterns that can be stored inside the AMs.

When the AMB finds a road with hits on at least 7 of the 8 layers, the road number is sent to the AUX which then retrieves all the hits in the road. Rapid hit retrieval is possible because hits are stored in the DO by SS address, and a road consists of a single Super-Strip in each layer. The hits, the road number, and the sector number are then ready for the fitting stage, that take place in the Track Fitter (TF) AUX FPGAs.

Track fitting is done rapidly by replacing a helical fit with a simple calculation that is linear in the local hit position in each silicon layer. The helix parameters and χ^2 components are, therefore, estimated

from the linear calculation. It consists of a set of scalar products of the hit coordinates and pre-calculated constants that take into account the detector geometry and alignment. If the region covered by a set of constants is small, the helix parameter resolution obtained using this method is nearly as good as that obtained with a helical fit. The TF has access to a set of constants for each detector sector, which consists of eight physical silicon modules, one for each layer.

The fitting process is divided in two stages. The first-stage fit, executed in the AUX board, computes only the χ^2 of the candidate tracks (not the helix parameters). This is required in order to perform a fast selection of the tracks to be sent to the second-stage fitting. The second-stage fit, performed inside the SSB boards, computes all the helix parameters. More detail on the second-stage will be given in the Subsection 3.1.4.

In the first-stage fit, only 8 silicon layers are considered, with 11 measured coordinates in total, assuming hits on all layers, with the 3 pixel layers providing two coordinates each. Since there are 5 helix parameters, the fit has 6 degrees of freedom, each of which gives a function of the coordinates that should equal zero. The χ^2 is the sum of the squares of those functions:

$$\chi^2 = \sum_{i=1}^{6} \left(\sum_{j=1}^{11} S_{ij} x_j + h_i \right)^2 \tag{3.1}$$

where S_{ij} and h_i are precalculated constants for a particular sector, and x_j are the hit coordinates.

Following each road fitting, duplicate track removal, executed by the Hit Warrior or HW function, is carried out among those tracks which pass the χ^2 cut. In this first stage, only tracks within the same road are compared. If a track shares more than a predetermined number of hits with a previously stored track in that road, the track with the higher quality factor is retained. The quality factor depends on the χ^2 and the number of pixel and SCT layers with hits. After all the tracks in a road have been processed, the remaining tracks are transferred to the Second Stage Board.

3.1.3 *Pattern matching*

The FTK pattern matching process is executed in the Associative Memory system. It consists of 3 different components: the Associative Memory chip (AMchip), an ASIC designed and optimized for the pattern matching application; the Associative Memory VME Board (AMB), the main board housing the FPGA needed for data processing; the Local Associative Memory Boards (LAMB), the AMB mezzanine used to hold the AMchips. Each AMB houses 4 different LAMBs, that carry in turn 16 AMchips each. In Figure 3.7 an AMB, together with its 4 LAMBs and the 64 AMchips, is shown.

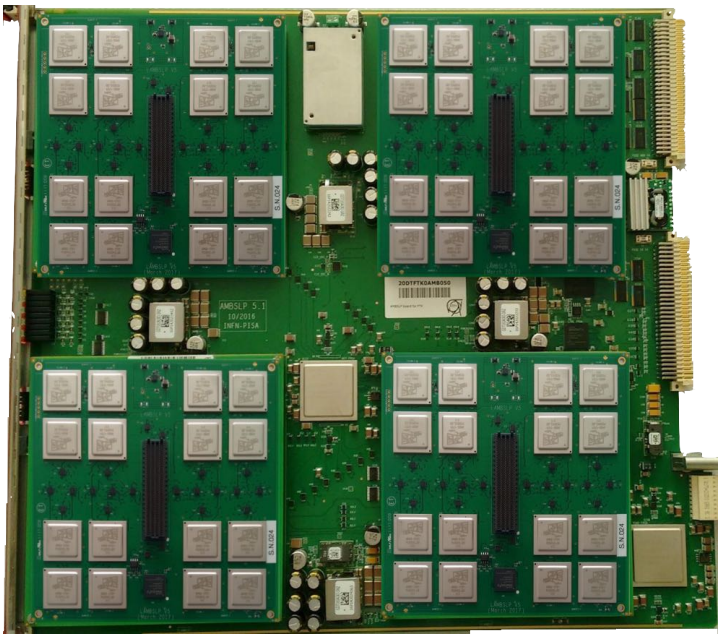


Figure 3.7: Picture of an AM Board. In the picture it is possible to see the AMB, the 4 LAMBs, and the 64 AMchips. The LAMBs in the picture show their internal face, in order to make it possible to see the AMchips. When the LAMBs are plugged, the AMchips are facing the AMB.

The data to be processed are received by the AMB from the AUX through a high frequency ERNI P3 connector. A network of high speed serial links is instead used for the distribution of the data on the AMB itself. The distribution bus is composed of 12 input serial links, carrying the silicon hits from the P3 connector to the LAMBs, and 16 output serial links, carrying the road numbers from the LAMBs back to the AUX. The hits for each event, organized into 8 buses, 1 for each of the 8 detector layers, are sent to the LAMBs through an SMD connector, fed partially in parallel, partially serially.

A set of FPGAs forms the Control logic of the AMB. When the Control logic starts to process an event, hits are popped in parallel from all the input FIFOs and are simultaneously sent to the four LAMBs, that in turn distribute them to their 16 AMchips with a four-fold fan-out. An End Event (EE) word, which includes the event tag, separates hits belonging to different events. The data in different streams have to be synchronized to guarantee that hits from the same event are being processed by the AM patterns.

The AMchip is able to process two events at the same time, using a two step pipeline: hit loading and pattern matching with readout. While hits from one event (event N) are being downloaded into the LAMBs, locally matched roads from the previous event (event $(N-1)$) are collected from the LAMBs and sent to the AUX. When the EE word is received on a hit stream, no more words are popped from

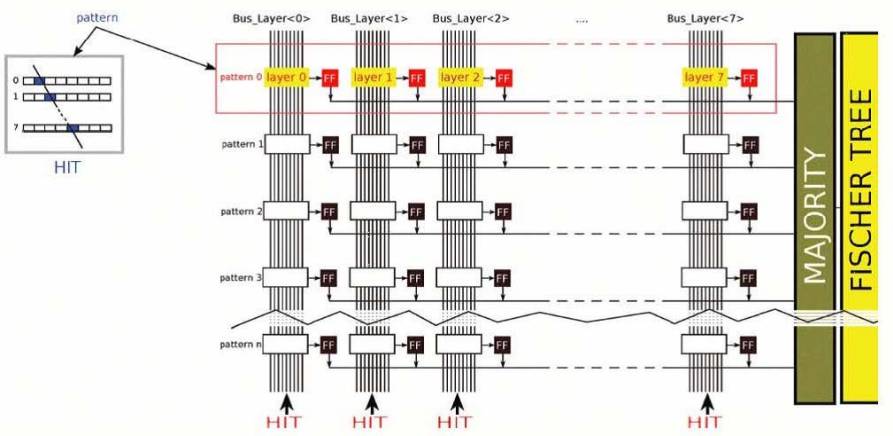


Figure 3.8: The AMchip array.

the relevant FIFO until the EE word containing the same event tag is received on all 12 hit streams. At that time, event N is fully loaded into the AM chips. Once the LAMBs have sent the last roads for event $(N-1)$ to the AUX, including the 16 EE words on all the 16 road serial links, event $(N-1)$ is considered fully processed. At that time, the Control chip sends INIT to push events forward in the pipeline: event $(N-1)$ is deleted by the AMB being fully processed, the hit information for event N is copied into the pattern matching and readout part of the AMchip, the hit portion of the pattern bank is reset for a new event, and the read-out of the roads from event N starts to complete the processing of event N. Finally, the hits from a new event, $(N+1)$, are popped out of the input FIFOs and sent to the AM chips.

In Figure 3.8, a sketch of the AMchip logic is presented. The AMchip is based on an array of Content Addressable Memory (CAM) cells. Columns are used to distribute the hit information over vertical buses called *Search Bitlines*, whereas rows are used for the write lines (the signals that enable the write operation) and for the match lines (the signal that identifies a match). Each bitline bus consists of 36 lines: the 18 bits in a hit and the 18 corresponding inverted bits. Each row in the array corresponds to one pattern. A row is organized in sub-blocks of 18 CAM cells, called a layer block. Each layer block stores the coarse resolution hit position in that layer for a pre-calculated track trajectory. A pattern is composed of 8 layer blocks, so it can identify a track crossing up to 8 detector layers. Each pattern contains the logic necessary to compare the stored position with actual hit position for each hit in one event. A pattern matches when all, or almost all, of the layers match hits in the input data for one event.

The event hit positions are received over 8 input buses of 15 bits each. This limits the maximum number of positions to 32 000 for each layer. This might seem a strict limit, however since different AM chips, or different groups of AM chips, can independently process data from

different parts of the detector, 15 bits are enough to cover an area of 12x18 pixels along the r, ϕ and z directions.

The positions stored in each layer block are encoded in the following way. Of the 18 CAM cells, each storing one bit, 12 are used to store the 12 most significant bits of the word, while the other 6 are arranged in 3 pairs and used as ternary cells storing either 0, 1 or X values. The X value means *don't care* (DC), so the hit present on the hit bus will match the stored word regardless of the values of the bits set to X. The use of the DC feature, as in ternary CAMs, allows to have variable size patterns. Normally the least significant bit of each layer corresponds to a fixed area on the detector. This area is the width of the matching window implemented by one pattern. When a bit is set to the DC value, the effective pattern size for that bit is doubled because it will match two numbers. With 3 DC bits, it is possible to enlarge the coincidence window up to a factor of 8 for each pattern and for each layer independently. In this way patterns can be tailored to maximize the acceptance for valid tracks, while reducing the probability of matching spurious hit combinations. In other words, each pattern has a better signal-to-noise ratio. Therefore, AM patterns are employed in a smarter and more efficient way.

Input data in the columns are compared in parallel with the data stored inside the layer blocks. If a layer block matches all 18 bits (accounting for DC bits in ternary cells), a Set-Reset Flip-Flop (SR-FF) is set to the high logic value. As can be seen in Figure 3.8, the majority block counts the number of SR-FFs set to 1. If this number is equal or greater to 6, the data are transferred to the AM readout block which is able to generate the address of the matched pattern by using a priority list based on a modified Fischer tree [55].

Finally, the addresses of the matched patterns are sent back to the AUX, ready for the first-stage fitting procedure.

3.1.4 *Second stage processing*

The second-stage processing is performed by the 32 Second Stage Boards, an example of which is shown in Figure 3.9. Each SSB receives through an RTM, the output from 4 AUX cards and the hits on the additional layers from the Data Formatter system for the 2 $\eta - \phi$ towers associated with those AUX cards. This second stage is needed to reduce fake tracks which may occur at the large rate due to high luminosity, and to improves helix parameter resolution (especially z_0), using all 12 detector layers.

First, the Extrapolator function (EXT) is performed, which extrapolates the positions of the hits from the 8-layer track information in the other 4 layers, to be used in the final 12-layer track fitting. The extrapolator uses a linear approximation in order to minimize the time spent for this operation.

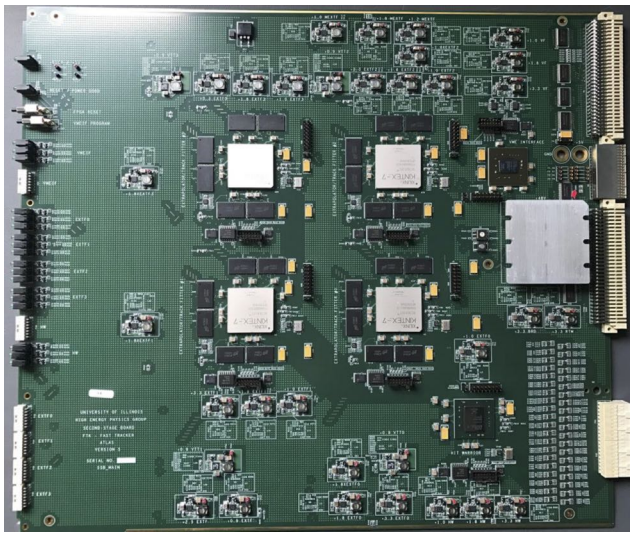


Figure 3.9: Picture of a FTK Second Stage Board. This board is responsible of the second-stage processing of the candidate FTK tracks.

After the extrapolation, the second-stage fitting is performed by the TF FPGAs. The TF uses a linear correlation for the helix parameters $\widetilde{p}$ computation, following the formula:

$$\widetilde{p_i} = \sum_{l=1}^{N} c_{il} x_l + q_i \tag{3.2}$$

where c_{il} and q_i are constants and x_l are the N cluster local coordinates. A minimum of three hits in the four layers not used in the first-stage is required.

Finally, duplicate track removal is again applied by the HW to the tracks that pass the χ^2 test, but unlike for the AUX duplicate removal procedure, tracks in all roads are used in the comparison. Because of the need to remove duplicate tracks across tower boundaries due to the tower overlap, it is more efficient to concentrate the HW function in those SSBs that send the final tracks to the FLIC cards. Because of this, there are two types of SSBs in the system, the Preliminary SSB (pSSB) and the Final SSB (fSSB), that alternate in SSB slots within a core crate and have identical Extrapolator and TF functions but different HW functions. A sketch showing the intercommunication principle between the two different SSB kinds is shown in Figure 3.10.

The pSSB sends its tracks to the $+\phi$ neighbouring fSSB in the same core crate for output to the FLIC. It also sends its tracks to the neighbouring fSSBs (excluding the fSSB just mentioned) solely for use in duplicate track removal.

The fSSB receives tracks from its own TF and the $-\phi$ neighbouring pSSB for output to a FLIC. These tracks are sent out only if they are not duplicated with any tracks from two ϕ neighbouring pSSBs (excluding the pSSB just mentioned) and two ϕ neighboring fSSBs.

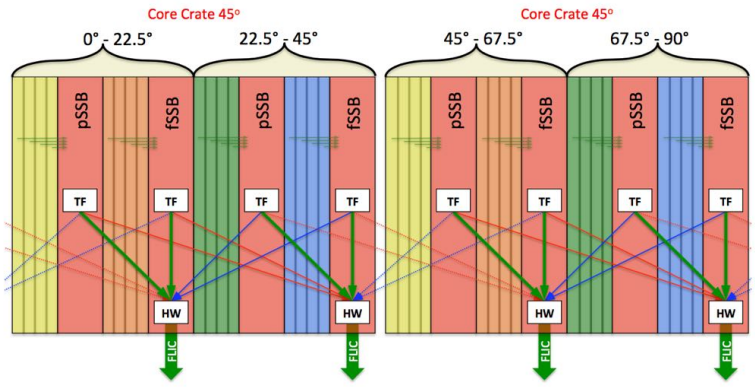


Figure 3.10: Sketch of the inter-module connections between the SSB Track Fitter and Hit Warrior, used for the duplicate removal procedure. Each SSB is connected to 5 other SSBs, and every fSSB has a connection to the FLIC.

3.1.5 *The FTK to HLT interface*

After an event is processed by the core crates and the track information is produced, the data is sent to the 2 FLIC boards for the transmission to the HLT. Data reach the FLIC boards from the fSSBs through SFP links as encapsulated records. Each core crate has two data links to one of the FLICs, one for each 22.5°ϕ region. The FLIC combines the two data streams from each core crate, formats it into ATLAS ROD fragments, and sends this information directly to the ROSs as a single data stream. The event data across the entire FLIC is identified by the Level-1 ID (L1ID), along with other information which is part of the data record header sent by the fSSBs. Finally, the High Level Trigger requests to the ROSs these fragments individually and assembles them as necessary, using them for the final trigger decision.

In addition, the FLIC provides monitoring and diagnostic capability. The system has an interface to the TDAQ infrastructure to facilitate monitoring and control. For monitoring, Spybuffers, i.e. special board circular buffers, are implemented so that the overall performance of the system can be monitored. Being one of my contributions to the project, more details on the Spybuffer logic will be given in Chapter 5. For diagnostics, the FLIC has the ability to process both fixed pattern data and user-defined data through the processing chain.

3.2 FTK INFRASTRUCTURE

The almost 450 FTK boards are housed in 13 crates, with a total of 7 racks required to contain them, placed on the second floor of one of the service halls of ATLAS (USA15). Each rack contains two VME crates (or ATCA shelves), together with one power supply and the required cooling infrastructure.

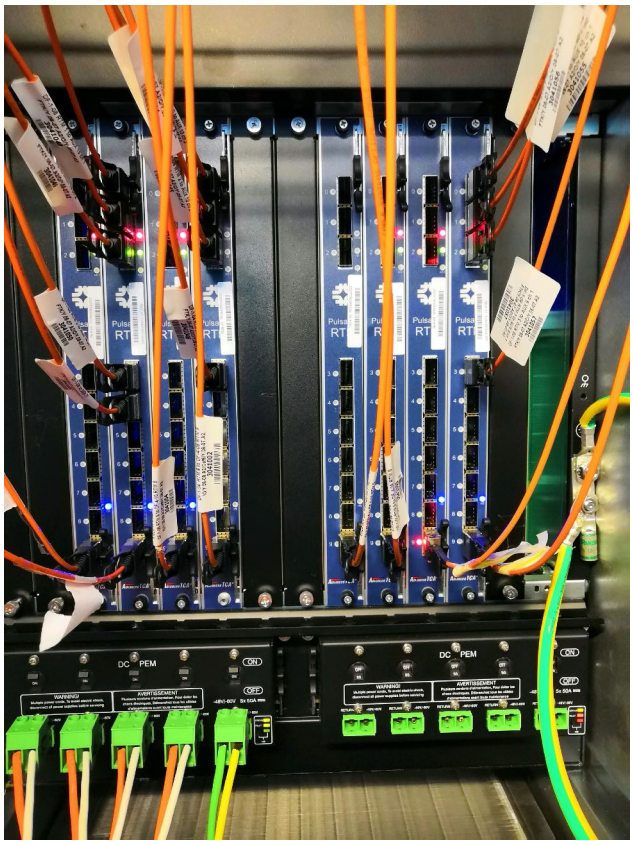
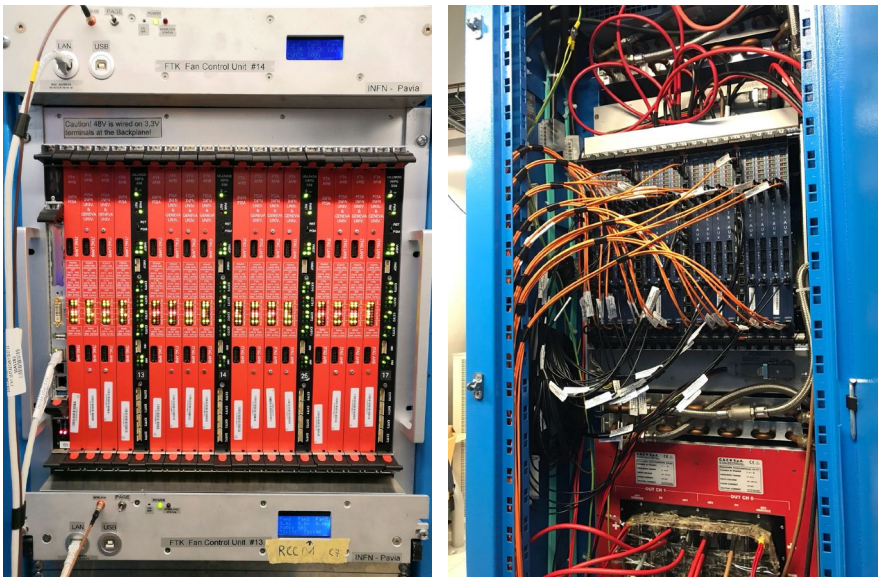


Figure 3.11: Picture of one of the FTK DF shelves in its final configuration. The shelf contains 8 DF boards and 64 IM cards. At the moment in which the picture was taken, not all the DFs were fully cabled.

As mentioned in the previous section, the FTK boards are based on the two standards VME and ATCA. The IM/DF and the FLIC boards occupy 5 ATCA shelves. One shelf is used by the two FLIC boards, while the other 4 shelves are filled with the 64 (256) DF (IM) boards. In Figure 3.11 a DF ATCA shelf fully loaded is shown. The choice of using the ATCA technology for this kind of boards was due to facilitate the massive reorganization of the silicon hits from the readout optical fibers to the $\eta - \phi$ tower organization of FTK. This data mapping can take advantage from the characteristics of the commercial ATCA shelves with full-mesh backplanes, developed by the telecommunication industry to incorporate the latest trends in high speed interconnect technologies, next-generation processors, and improved reliability, availability and serviceability.

The 128 PUs and 32 SSB boards are instead housed in the remaining 8 VME crates, each of them covering 45° in azimuth. The choice of the VME standard in this case relies on the fact that the Associative Memory boards need a full 9U height to hold all of the AM chips, and because the Auxiliary Cards behind each AMB is 280 mm deep in order to hold the Data Organizers and first-stage track fitters.



(a) (b)

Figure 3.12: (a) Picture of one of the FTK VME crates in their final configuration. It is possible to see the 16 AMBs (red boards), the 4 SSBs (black boards), and the controller Single Board Computer (SBC) in slot 0. The 16 AUXs and 4 SSB RTMs are not visible, in the back of the crate.
(b) Picture of the back of a FTK VME crate. The 16 AUX boards (blue boards) and the 4 SSB RTMs (black boards) are visible. In the picture it is possible to see the QSFP cables bringing the hits from the DFs to the AUX and SSB boards (orange cables), and the SFP cables bringing the roads from the AUX to the SSB boards (black cables). The back of the rack power supply is also visible (red box).

Figure 3.12 shows the board organization of a VME crate. Each crate house 16 Processing Units, 2 per FTK tower, and 4 SSBs.

Noteworthy is also the cooling system, required to dissipate the high power consumed by the VME boards during operations. This system consists of custom made fan trays with improved cooling capacity. Each rack contains 2 pairs of this custom fan trays, with one tray just over and one just at the bottom of each VME crate, as can be seen in Figure 3.13. More details on the boards power consumptions and on the cooling infrastructure will be given in the next chapter (Section 4).

3.3 COMMISSIONING OF THE FTK SYSTEM

The FTK system is currently undergoing its commissioning phase. Planned to have half of the system fully integrated in the last part of the Run2 data taking session, FTK is still being commissioned, with only two *slices* (chains of FTK boards composed of a single board per kind) being integrated and taking data with ATLAS. Moreover, the

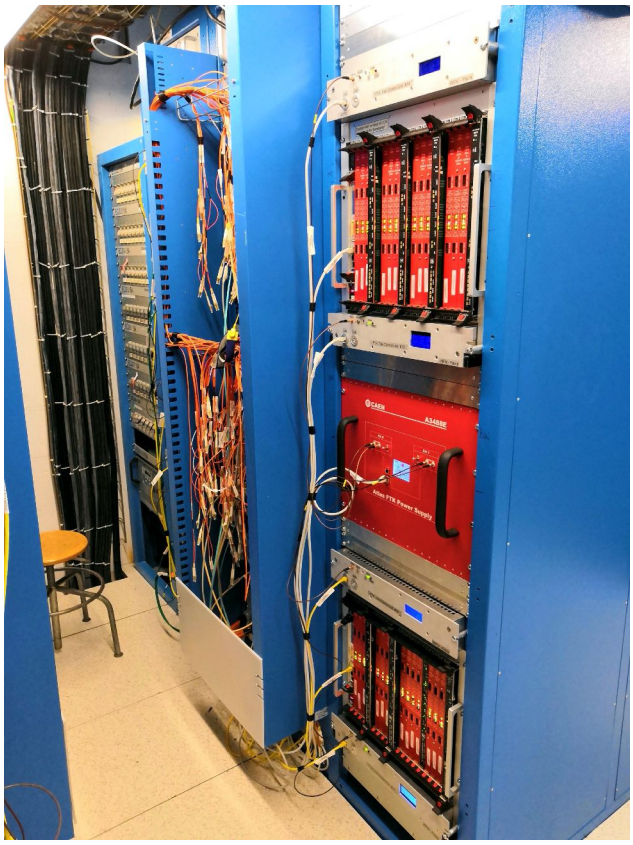


Figure 3.13: Picture of a full VME rack. In the photo the two VME crates are visible, together with the FTK Power Supply (PS), the red box, providing the required power to the crates and to the fan trays, and the 4 custom fan trays (the metallic boxes with the small LCD screen on their right).

tracks produced with these slices were not used by the HLT for trigger decisions, but only stored for the study of the FTK performances.

The commissioning delays are mainly due to firmware (FW) issues, not predictable in the laboratory system while testing the various components with simulated data. Real data can be corrupted, missing fragments can be present, and a huge diversity of issues can occur. The reliability of the system can be tested only on data taking conditions, e.g. integrating portions of the system in the ATLAS data taking sessions. In the following, the different FTK commissioning strategies will be presented, preceded by a brief overview of the production and installation status of the hardware components.

3.3.1 *Hardware production and installation*

At the moment of writing, almost all of the required FTK boards, excluding half of the AMBs, have been produced.

All the IM, DF and FLIC boards have been produced, shipped to CERN and installed in the dedicated ATLAS Service Area racks. All

FTK board	Required boards	Produced boards	Boards at CERN	Installed
IM	128	128 + spares	128	128
DF	32	32 + spares	32	32
AUX	128	128 + spares	64	64
AMB	128	71	64	64
SSB	32	32 + spares	8	8
FLIC	2	2 + spare	2	2

Table 3.1: Summary of the production, shipment and installation status of the different FTK boards at writing time.

the AUX boards have also been produced, with half of the 128 final required boards already shipped at CERN and installed. The remaining half is expected at CERN by the end of the year.

The production status of the AMB and SSB boards is, on the other hand, late with respect to the initial schedule, due to different reasons.

The AMB boards, and in particular the AMchip, production was planned to be executed in two batches. Half of the boards and chips were planned to be produced before the end of the Run2. These boards were actually produced and delivered, and are currently installed in the VME crates. The second batch of production was instead expected to be executed by the start of the LS2. Due to a review that the FTK project is undergoing, this production batch was delayed. It is expected to start by the end of the summer 2019, in order to get all the missing boards ready before the beginning of Run3 (October 2021).

The production of the SSB boards was also stopped, in this case because of a review of the SSB boards themselves. This review was required in order to improve the reliability of this kind of boards, that were showing problems with the first prototypes and which required a re-design of the power distribution and some heat sink modifications. The review has been terminated, and a new version of the SSB (SSB V5) has been produced. Some of the already produced old SSBs (SSB V4) have been modified in order to be compliant with the new specifications. Currently all the required 32 boards have been produced or modified, and their shipping to CERN is undergoing.

In Table 3.1 a summary of the production status of all the FTK boards, at the moment of writing, is shown.

As can be seen from the table, all the boards that have been received at CERN have already been installed in their respective crates, and the cabling has been performed.

3.3.2 *Commissioning status*

The commissioning of the FTK system is performed at CERN in two separate places. Before being installed in the ATLAS USA15 cavern, the

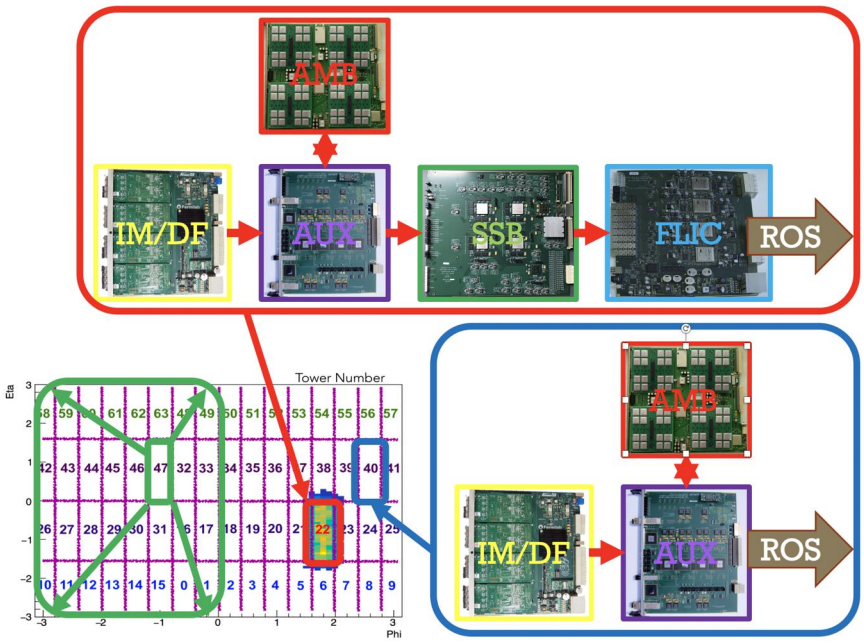


Figure 3.14: Tower distribution of the 3 FTK slices that were commissioned during 2018. A sketch of the composition of the SliceA (red box) and Slice2 (blue box) is also presented. Slice47 (green in the tower distribution plot) started its commissioning as a single tower, but will be expanded to cover the *Hemisphere*, corresponding to half of the full FTK system.

boards arrived at CERN undergo testing in a ground level laboratory, called *Lab4*.

Here two racks, containing 2 VME crates and 2 ATCA shelves, are used to test the shipped boards, checking for possible damages occurred during shipment. Once the boards have been tested and are considered good for operation, they are moved to the ATLAS USA15 cavern. Moreover, a couple of each kind of FTK boards are also permanently installed, used for commissioning purpose (e.g. testing the intercommunication between DFs, the communication between different kind of boards, etc...). Lab4 is also used for the testing of both the firmware and software releases before the official deployment of the code.

The commissioning of the FTK system during the Run2 was based on Slices. A slice represent a chain of FTK boards, with one board per kind. During 2018 3 different slices were commissioned:

- **SliceA**: a full FTK slice, composed by one board per kind (1DF (with 4 IMs) - 1AUX - 1AMB - 1SSB - 1 FLIC).
- **Slice2**: a reduced FTK slice, composed by one board per kind up to the AUX (1DF (with 4 IMs) - 1AUX - 1AMB).

- **Slice**47: a full slice, like SliceA, used as a starting point for the expansion of the system (by adding more boards).

The distribution in the detector space of the slices, together with a sketch of the composition of the above mentioned slices, can be seen in Figure 3.14.

The commissioning of these slices was performed following progressive steps. First, the correct processing of the boards was tested by running on *pseudo-data*. This are simulated ID events directly injected in the IMs and, from them, to the board chain. This simulated events are useful for testing the FTK processing under controlled conditions, avoiding the problems that the real data can have, like missing or corrupted fragments.

The second step is the so called *parasitic-mode*. In this case the FTK slices process real data (with or without a prescale on the events) but without joining the ATLAS data taking session. The FTK slices in this case sniff on the ID output, but the produced tracks are not written to the ATLAS event stream. Being out of the ATLAS run, FTK doesn't interfere with the ATLAS data taking, even if problems to the system occur. Running on parasitic mode is particularly important in order to train the system to run on real data and to make the system reliable and able to survive to the possible problems that the real data can have.

Finally, the last commissioning step is the full integration of the slice inside the ATLAS data taking session. In this case, the FTK produced tracks can be written in the ATLAS event stream, but a failure of the FTK system can degrade the overall ATLAS performances, for example introducing delays in the DAQ system. In order to avoid this problem, a method for the removal of FTK from the data taking when a major problem occurs has been developed. More details about this will be given in the Chapter 5.

SliceA was the first FTK slice to be included in an ATLAS run. It represent the FTK tower 22, covering the detector region of $-1.5 < \eta < 0$ and $1.6 < \phi < 2.0$. At the end of 2018, SliceA was fully integrated in the ATLAS data-taking architecture, processing data without prescale producing 12-layer tracks that were written to the ATLAS event stream. An example of an FTK recorded event is shown in Figure 3.15.

Before the end of the Run, also Slice2 was included in the ATLAS data-taking session. This slice covers the FTK tower 40, corresponding to a detector region $0 < \eta < 1.5$ and $2.4 < \phi < 2.8$. Slice2 is a reduced slice, used for the commissioning of the upstream boards (up to the PU) while the downstream ones (SSB-FLIC) were undergoing debugging. Slice2 was able to process and output 8-layer tracks, that, as the ones of SliceA, were wrote to the ATLAS event stream for the study of the FTK performances. In order to cope with the missing part of the FTK board chain, the AUX board of Slice2 uses a modified

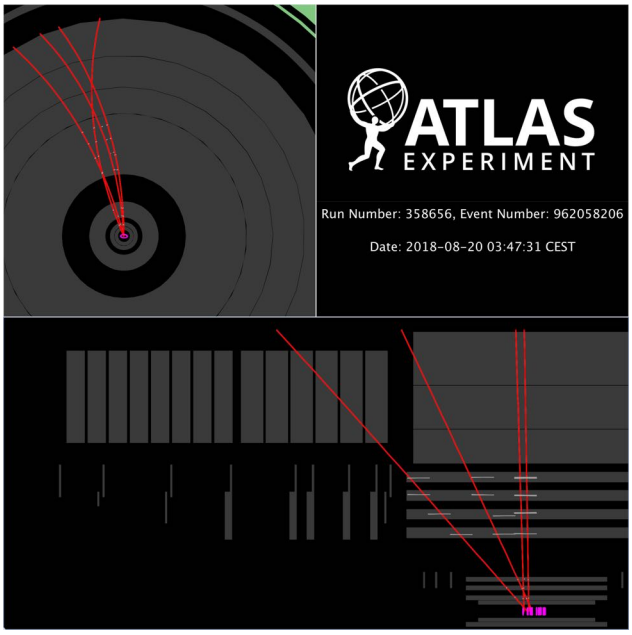


Figure 3.15: Event display for proton-proton collision including the FTK information produced by SliceA, recorded on 20th August 2018, showing the ATLAS inner detector in transverse projection (top), and lateral projection (bottom). The red lines show trajectories corresponding to the track parameters obtained from an offline fit to the cluster positions determined by FTK. The FTK cluster positions are shown in white points and, for SCT strips in the lateral projection, lines. The primary vertices determined from tracks reconstructed offline are shown in purple.

version of the firmware, that enable it to directly write its 8-layer tracks to the ROS without passing through the FLIC.

Together with SliceA and Slice2, also the third slice, Slice47 underwent commissioning during 2018. This slice, initially covering the FTK tower 47, was meant to be the starting point for the expansion of the FTK system, with the goal of covering the FTK *hemisphere*, representing half of the final FTK system. With this slice, the intercommunication between DFs, together with the communication with more PUs, were tested. Due to problems in establishing the intercommunications between DFs, no other towers were included during 2018.

The 2019 FTK commissioning is mainly focused on solving the DFs intercommunication problems, in the scaling up of the system, and in the bit-by-bit comparison between functional simulation of FTK and the processed data at each of the processing steps. Currently the testing of the intercommunication between DFs is proceeding in parallel to the scaling-up of the PU numbers. A new slice composed of a single DF and 4 PUs is under testing. The bit-by-bit comparison tools have been developed, and an automation of the comparison procedure is ongoing development.

Because of the lack of collisions during the current shutdown period, a new feeding system for FTK, called **QuestNP**, is also undergoing commissioning. This system is composed of electronic boards able to feed the FTK inputs with "real" data, collected before the end of Run2, overcoming the limitations that the direct IM feeding has (e.g. the memory available on the IM for the storing of ID events is limited, and it is not possible to simulate the data arrival time delays, typical of the data coming from the real ID detector). The QuestNP system is based on C-RORC cards (PCI Express Read-Out Receiver Card), that will interface with standard PCs. The recorded data will be retrieved via software from the application host PC, transferred to the QuestNP system via PCIexpress and sent to FTK via S-link, mimicking the ID system. All the C-RORC cards are available and already cabled, and the system is currently under testing.

The FTK system is expected to be fully commissioned before the beginning of Run3.

3.4 FTK PERFORMANCES

As presented in the previous section, during 2018, 2 of the 64 FTK towers collected proton-proton collisions and FTK commissioning data were analysed. In this section, the FTK performances based on data from these two slices will be presented. Before showing the performances, a brief introduction on the FTK simulation will be provided.

3.4.1 *FTK simulation*

The simulation of the FTK system is fundamental for the study of the system performances, but also for the bit-by-bit comparison of the FTK processing steps, already mentioned in Section 3.3, and for the computation of the FTK configurations, as the constants used in the fitting stage. Two types of simulation are used to model the FTK track processing. A full functional emulation called **FTKSim** has been implemented in C++ code. This full simulation is used to produce the sectors constants and patterns and to validate the firmware using bit-by-bit comparisons to FTK tracks.

Unfortunately, this full emulation is too slow (approximately 600 seconds per event) to be useful for large-scale Monte Carlo sample productions. A fast simulation has therefore been developed. This fast simulation uses efficiency weights and track parameter smearing to quickly emulate the FTK tracking performance. Track parameter resolutions are extracted from samples of muons simulated with FTKSim and separated into bins of track p_T, η, and number of IBL hits. For each bin, double Gaussian resolution functions are fit to the distribution of each of the five track parameter residuals, i.e.,

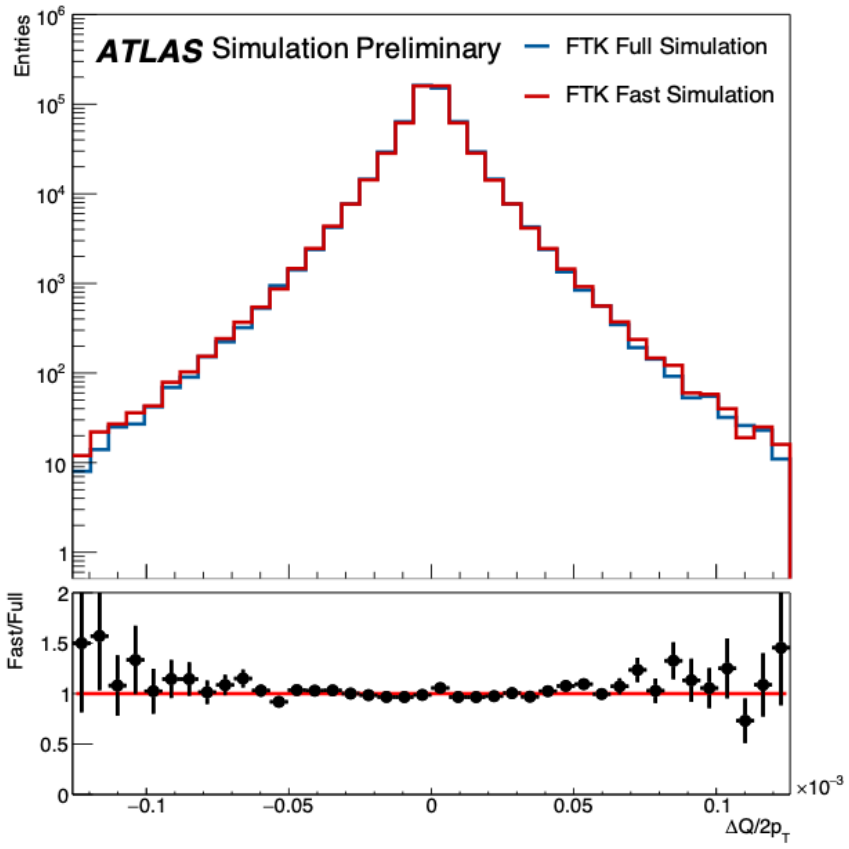


Figure 3.16: The resolution of the FTK track curvature, from functional emulation of FTK using FTKSim (FTK Full Simulation) and a parameterized uncertainty model (FTK Fast Simulation). The difference between the FTK reconstructed and true values of track charge (ΔQ) divided by twice the track transverse momentum is plotted. A fully simulated sample of muons with longitudinal impact parameter $|z_0| < 110\,\text{cm}$ and pseudo-rapidity $|\eta| < 2.5$ is used. The resolution for the fast simulation is modeled as a double gaussian derived from full simulation.

the reconstructed minus true values for each parameter. The fast simulation is validated by applying smearing functions to a simulated muon sample and comparing the resulting residuals to those obtained from full simulation. An example showing the modeling of the track curvature is presented in Figure 3.16. As can be seen from the figure, the fast simulation provides good modeling of the residuals in both the core and tails of the resolution function.

3.4.2 *FTK Slice2 performances*

The tracking performance results from the Slice2 are presented in this section. For all results, a single Data Formatter (DF) board provides

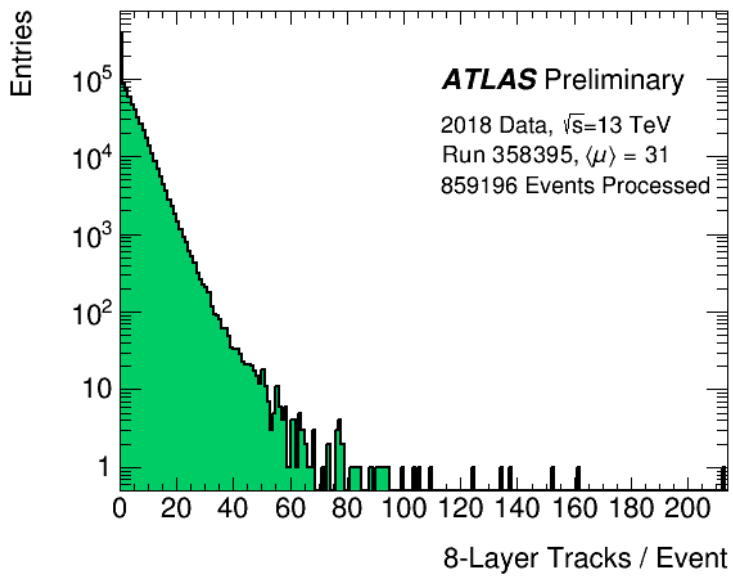


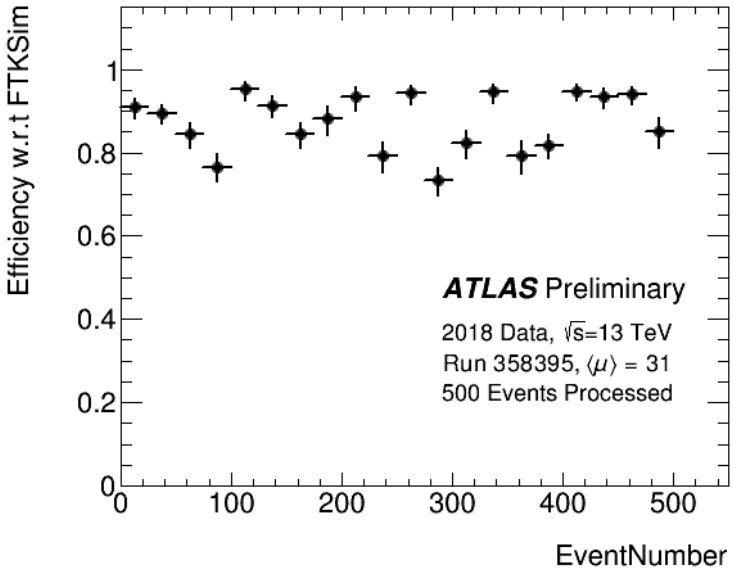
Figure 3.17: Results from a subset of 859 196 events from Run 358 395, a 13 TeV pp run which began on August 2018 with the FTK Slice2 (FTK tower 40). The plot is showing the number of 8-layer tracks produced per event.

partial coverage of tower 40. The data covers a subset of 859 196 events from Run 358 395, a 13 TeV pp run taken on August 2018.

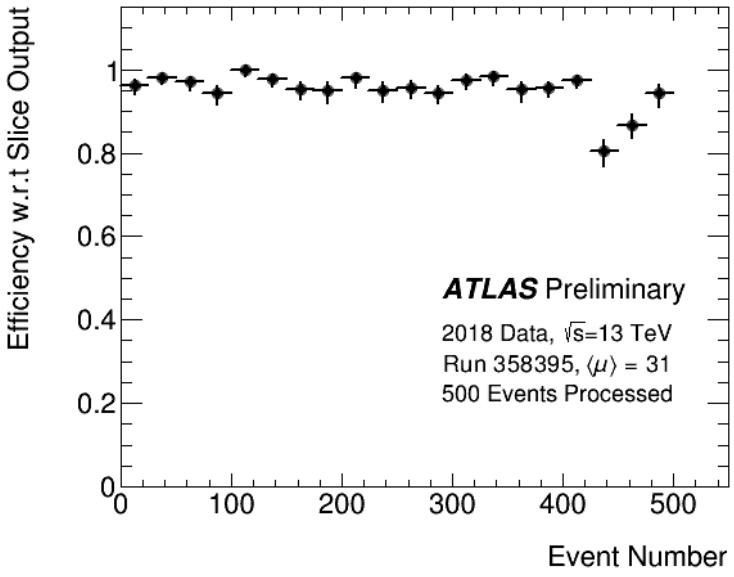
Figure 3.17 shows the distributions of the 8-layer tracks per event collected by the Slice2. The most probable number of tracks per event is 1, but events with as many as 100 tracks are collected. As will be shown in the next section, the number of 8-layers tracks per event is much larger that the one of 12-layer tracks (recorded by SliceA). This is due to the missing of the second-stage fitting, executed by the SSB board, in this slice processing. A considerable number of bad tracks is not dropped.

This is confirmed by Figure 3.18. Figure 3.18a shows the fraction of FTK tracks matched to the tracks produced by running the FTK functional simulation. The efficiency varies from the 95 to 70 %, showing the presence of a certain number of bad tracks. On the other hand, Figure 3.18b shows the fraction of tracks produced by running FTK functional simulation (FTKSim) matched to the FTK Slice2 output tracks. In this case the efficiency is almost everywhere higher than the 95 %, showing that almost all the expected tracks from FTK have been produced.

Figure 3.19 shows the p_T distribution of the 8-layer tracks generated by the FTK functional simulation (FTKSim) that were matched to tracks output by the FTK Slice2. The p_T distribution is peaked just above the FTK p_T threshold of 1 GeV and falls off rapidly at high p_T, as expected.



(a)



(b)

Figure 3.18: (a) Fraction of FTK Slice2 output tracks that are matched to tracks produced by running FTK functional simulation (FTKSim) on RAW data. (b) Fraction of tracks produced by running FTK functional simulation (FTKSim) on RAW data that are matched to FTK Slice2 output tracks.

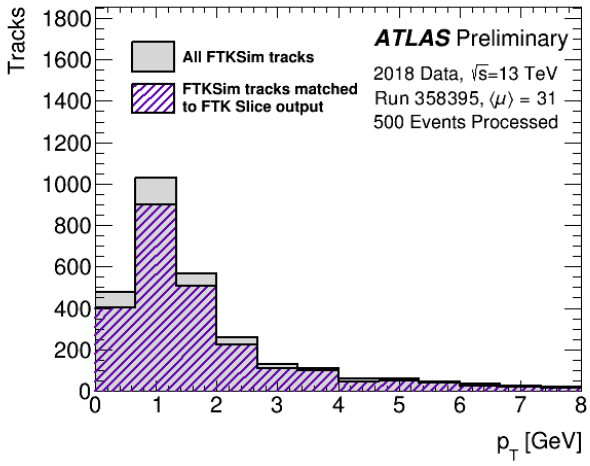


Figure 3.19: The p_T distribution of the 8-layer tracks generated with the FTK functional simulation (FTKSim) that were matched to tracks output by the FTK Slice2. Because the FTK slice output tracks do not contain track parameter information, the value of p_T is taken from FTKSim. The distribution of matched simulated tracks (grey) is compared to all simulated tracks for the same events (purple).

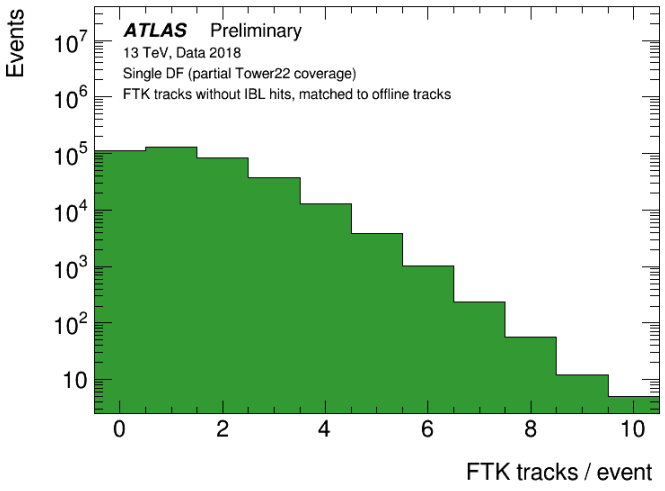
3.4.3 *FTK SliceA performances*

The tracking performance results from the SliceA are presented in this section. For all results, a single Data Formatter (DF) board provides partial coverage of tower 22. The data cover a subset of 380 000 events from Run 364 485, a special high pile-up run with average interactions per bunch crossing $\langle\mu\rangle = 82$, collected in October 2018. FTK tracks with Insertable B-layer hits were excluded, due to a FTK module ordering problem that caused incorrect hit positions in the run. The cause is understood and the fix is being implemented. FTK tracks are matched to offline tracks within $\delta R < 0.02$.

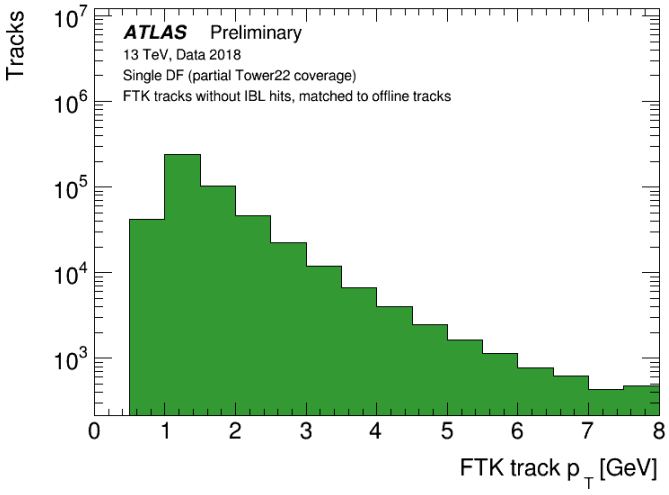
Figure 3.20 presents the distribution of the 12-layer tracks collected by the Slice. The most probable number of tracks per event is 1 and events with as many as 10 tracks are collected. The p_T distribution is peaked just above the FTK p_T threshold of 1 GeV and falls off rapidly at high p_T, as expected.

Figure 3.21 shows the fraction of FTK tracks matched to selected offline tracks as function of the FTK track transverse impact parameter d_0. More than 98% of tracks with small d_0 are matched to offline tracks, decreasing to 95% for tracks with $d_0 = 0.5$ mm. This indicates that the majority of tracks reconstructed by FTK are good tracks corresponding to real particles, as opposed to fake tracks from random combinatoric assignments of hits.

Figure 3.22 shows the track parameter resolutions for p_T and ϕ distributions. These results demonstrate that FTK is reconstructing



(a)



(b)

Figure 3.20: Results from a subset of 380 000 events from Run 364 485, a special high pile-up run with average interactions per crossing $\langle\mu\rangle = 82$, collected in October 2018 with the FTK sliceA (FTK tower 22). The plots are showing: (a) the number of tracks produced per event; (b) the track transverse momentum (p_T).

tracks with the correct momentum and directions. The resolutions are well-modelled using full functional emulation of FTK. The FTK refit, which uses a full helix fit rather than a linear approximation, and which also includes additional corrections to the hit position including the Lorentz angle effect, improves the track parameter resolutions by 10–20%.

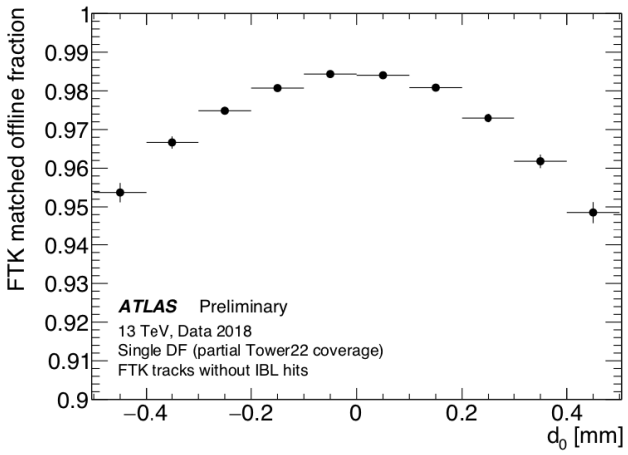


Figure 3.21: Fraction of FTK tracks matched to selected offline tracks, as a function of the FTK track transverse impact parameter d_0, from the FTK SliceA (FTK tower 22).

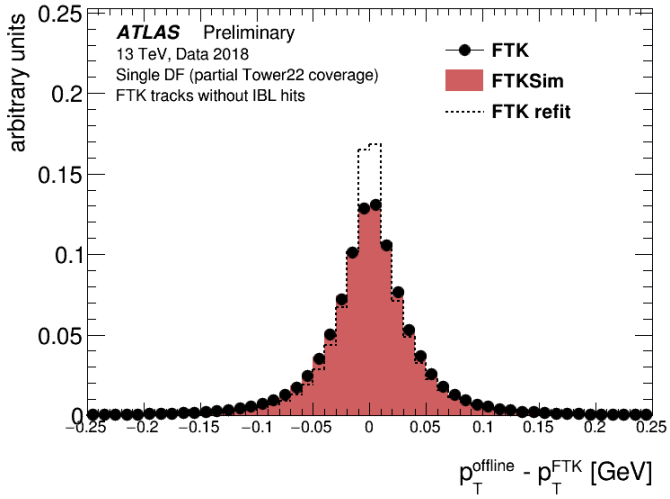
3.5 FTK REVIEW AND DESCOPING

The initial commissioning plan of the FTK system, presented in the FTK Technical Design Report, was targeting the full integration of half of the FTK system in the ATLAS data taking infrastructure by the end of Run2. This plan has been demonstrated to be too optimistic.

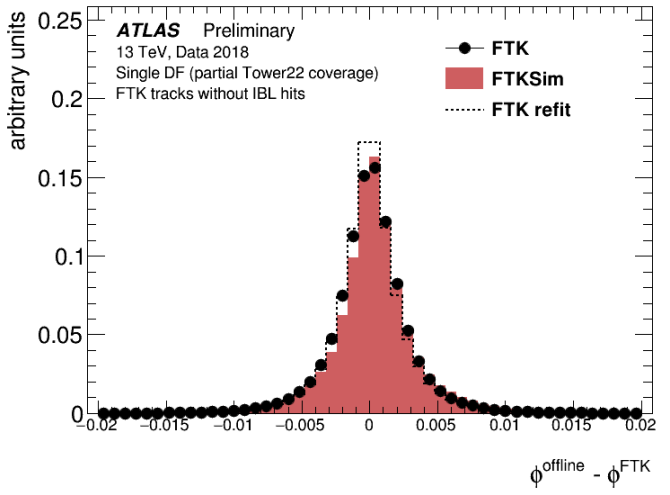
Because of the delays the project accumulated, mainly due to the underestimation of the additional FW development required, that became evident only during the commissioning phase of the system, an ATLAS internal review of FTK was put in place by the ATLAS management in July 2018. This followed concerns with the progress of the project during Run2, including comments from the LHCC referees encouraging ATLAS management to find additional resources on both project management and firmware development to avoid additional delays.

Due to the lack of resources, it was not possible to guarantee all the personpower identified as required by the review panel for the success of the project (especially FW developers). In order to overcome this limitation, a descoping of the project has been proposed and accepted.

Given that the higher priority trigger signatures, for which the use of FTK tracks will be more important during the next Run have an expected rate of few tenth of kHz, the baseline processing rate for FTK was moved from 100 kHz to 35 kHz. This reduction in processing rate introduces simplifications in the FW development, as well as increasing the reliability of the current FW implementation. As an example, running at a lower rate reduces the resource utilization in the FPGAs. This resources can then be used to enlarge the FIFOs size, increasing the possibility of FTK to survive to synchronization issues. Moreover, decreasing the rate would let FTK cope with some of the



(a)



(b)

Figure 3.22: FTK vs offline track residuals from the FTK SliceA for (a) p_T and (b) ϕ. The difference between the FTK and matched offline track p_T are shown for the FTK slice (black dots), the functional emulation of the FTK slice using FTKSim (shaded red histogram), and FTK refit (dashed line).

problems discovered during the commissioning of the system, and which were requiring the redesign of some FW blocks to be solved, saving considerable firmware development.

As an example, running under 45 kHz makes possible to execute the second-stage fitting on the SSB boards on a single set of FPGAs, reducing some of the complications of the process. Moreover, the new rate would allow DF processing without the redesign of the switch network.

Finally, the review panel recommended to adopt the SLINK protocol for the SSB/FLIC interface, and to move the interSSB overlap removal

task from FTK to the HLT (the CPU overhead for this is negligible). Being the FLIC additional firmware and hardware cost prohibitive, due to resource limitations, the FLIC board has been removed from the project, moving directly in the SSB the duty of writing the FTK tracks to the ROS.

Following this descoping, the FTK system is expected to be commissioned, with a floating of 5 months, by the beginning of Run3.

4

FTK VME INFRASTRACTURE AND COOLING

As presented in the previous chapter, three FTK processing boards (AMB, AUX and SSB) are housed in VME crates. The FTK VME crates are planned to be fully populated, with 16 PUs, 4 SSBs and one Single Board Computer each (see Figure 3.12). Due to the high boards density and to the power consumption of these boards, the crate temperature is a potential issue. The electronic components, on which the FTK boards rely, have a nominal temperature threshold over which they can't operate safely. If operated over the temperature limit, bit inaccuracy problems or even damages to the components can occur. Moreover, ensuring the board operation at temperature below these thresholds can extend their lifetime. Therefore, limiting the temperature in the crates and maintaining these temperatures as low as possible are among the most important goals of the VME infrastructure design.

In this chapter, details about the VME infrastructure configuration and of the cooling issue will be provided, together with the results of the cooling studies aimed to provide a complete characterization of the FTK VME environment.

First, an analysis of the power consumption of the VME boards will be presented (Section 4.1). This analysis has been fundamental in order to provide an estimation of the power dissipated by the FTK boards during the operational conditions of the Run3.

In the second section (Section 4.2) an estimate of the maximum temperature that the FTK VME boards can stand will be provided.

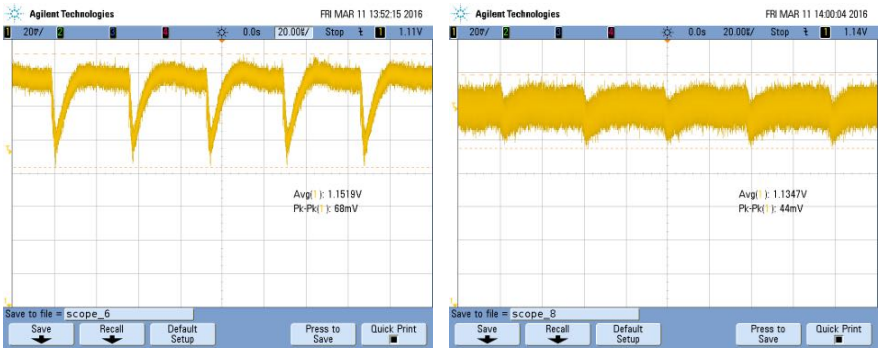
In Section 4.3, a detailed description of the cooling system of the FTK VME racks will be presented, focusing on the peculiarities that differentiate the FTK requirements from the standard ATLAS ones.

Finally, in Section 4.4, the results of the cooling studies will be presented.

4.1 VME BOARDS POWER CONSUMPTION

While a coarse prediction of the power consumed by the different board components is provided by the manufacturers, a direct measurement on the programmed boards is, anyway, required.

In fact, the FPGA power consumption is proportional to the specific configuration with which it runs. The resource utilization (in typical FPGA designs, some of the resources are not used after the configuration and thus they don't consume any power) and the switching



(a) (b)

Figure 4.1: Effect on the voltage ripple of the module dummy hit. The quiescent current is increased in order to reduce the effect of the current peak. In Figure (a), without the use of dummy hit a quiescent current of 3 A per LAMB results in a peak to peak variation of 68 mV. In Figure (b), using the dummy hit the quiescent current is increased to 38 A per LAMB, reducing the peak to peak voltage variation to 44 mV. In both the figures, the time scale is 20 $\mu V/div$, and the vertical scale is 50 mV/div.

activity (the number of signal transitions in a clock period) are the main contributors to the power variation [56].

Also the power consumption of the AM chip require a careful measurement. In this element, the power consumption is not related to its configuration, but to the composition of the data undergoing processing and to their rate.

A measurement campaign of the FTK VME boards power consumption has been carried out, as a starting point for the study of the VME temperature profile.

Being FPGA based, the power consumption of the AUX and SSB boards during operation can be considered fixed (with respect to a given FW version). Therefore, the variations upon the different running conditions are small. The measurements have been performed by programming the two boards with the latest stable FW versions (end of 2018), and running them in standalone (not connecting them to the up-stream/down-stream boards) while processing pseudodata.

The AUX boards, running in loop mode on data directly loaded on the FPGAs fifos, have been measured to consume about 70 W.

The SSB boards, fed with data coming from an emulator connected to the board, has been measured to consume about 160 W. Since no big changes in the FPGA resources utilization or in the switching activities are expected to be introduced on these boards during the last part of commissioning, these values can be considered to be close to the final ones.

The estimation of the power consumption of the AMB is more complicated. This board relies its processing on two different components:

FPGAs and AM chips. While, as for the AUX and SSB case, the AMB FPGAs are characterized by an almost fixed power consumption (for a given FW version), the AM chips show drastic variations. As already mentioned, these variations are related to the characteristics of the data undergoing processing, and in particular to the input data rate and to the Bit Flip number (BF), i.e. the number of bits that flip in the AM memory lines during the pattern matching procedure. The total AMB power consumption can be, therefore, computed as:

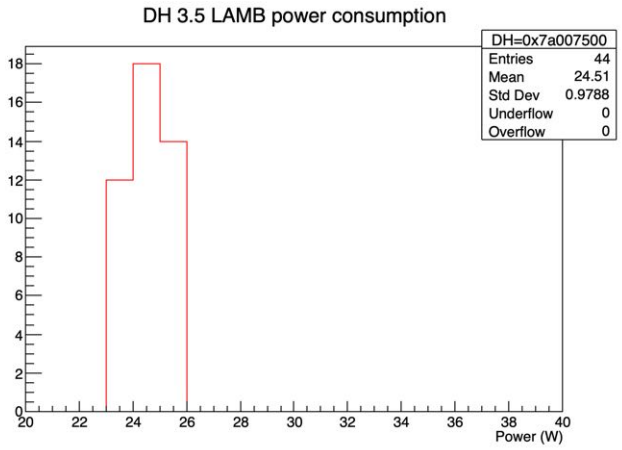
$$\mathrm{AMB}_{\mathrm{power}} = \mathrm{AMB}_{\mathrm{core}} + 4 \cdot (\mathrm{LAMB}_{\mathrm{core}}) \tag{4.1}$$

where $\mathrm{AMB}_{\mathrm{core}}$ represents the power consumption of the AM mother board alone (FPGAs component), while $\mathrm{LAMB}_{\mathrm{core}}$ represent the power consumption of the LAMB boards, which is data dependent (AM chip component). The $\mathrm{AMB}_{\mathrm{core}}$ value has been measured to be of about $100\,W$. The $\mathrm{LAMB}_{\mathrm{core}}$ is in turn the sum of two different contributions. In order to compute the expected power more considerations on them are required.

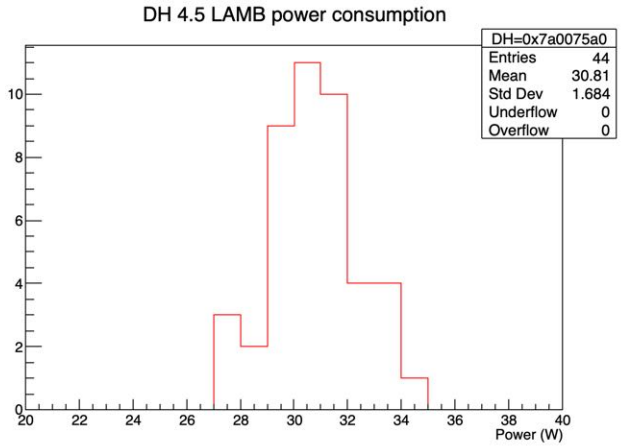
The first contribution comes from the *dummy hit* value. Because of the high data fan-out in the memory bank, the current on the AM chip net is not constant, but varies from a quiescent current of $20\,A$ to a peak current of $220\,A$ in a few hundreds of nanoseconds. A very powerful DC-DC converter is used to power each LAMB system, but even with this device is not possible to avoid the effect of the fast large peak of current, resulting in a drop of the voltage level. In order to mitigate this problem, a dummy word is sent to the AM chips while in IDLE, i.e. while it is not processing data, helping to increase the quiescent current and, thus, reducing the voltage drop. In Figure 4.1, an example of the behaviour of the voltage drop with and without the use of the dummy hit is presented.

The dummy hit is a 32 bit word that is sent to the AM chip, alternated to the IDLE word, during the IDLE state. This dummy word doesn't fire any pattern on the chip, but can be chosen so that the number of bits that flip in the AM lines at each clock cycle can be tuned as desired, managing the IDLE power consumption of the chip. Being configurable, the dummy hit can be adapted to the user requirements. Choosing a dummy-hit value with a lower level of bits flip can reduce the IDLE power consumption, helping to keep the board temperatures under control. On the other hand, this would reduce its effectiveness on the mitigation of the voltage drop.

A study on the IDLE power consumption of the AMB, for different values of the dummy hit has been performed. In Figure 4.2, the LAMB power consumption distribution for two different values of the dummy hit are shown. The choice of the value of the dummy hit is a compromise between the increase of the quiescent current of the chip net, requiring a big bit flip value, and the necessity of reducing the total power consumption of the board. Two different values of



(a)



(b)

Figure 4.2: LAMB power consumption for a value of dummy hit of 0x7a007500 (corresponding to a bit flip of 3.5) (a) and a value of dummy hit of 0x7a0075a0 (corresponding to a bit flip of 4.5) (b). During this measurements no data were sent to the AM chips. The power consumption is only due to the dummy hit value.

dummy hit has been identified as optimal for operations, each of them better fitting one of the two characteristics described above. The lower dummy hit value is 0x7a007500, corresponding to a bit flip value of 3.5:

$$0x7a00 \rightarrow 0b0111 - 0b1010 - 0b0000 - 0b0000$$
$$0x7500 \rightarrow 0b0111 - 0b0101 - 0b0000 - 0b0000$$

ID layer	Number of clusters			
	Barrel		Endcap	
	Average	Max tower	Average	Max tower
Pixel Lo	280	320	280	340
Pixel L1	240	290	240	280
Pixel L2	220	270	190	220
SCT Lo	240	250	260	280
SCT L1	240	270	230	250
SCT L2	270	280	220	230
SCT L3	270	280	220	230
SCT L4	240	250	200	220

Table 4.1: Expected average and maximum number of clusters for the different ID layers used by the PU. These values are computed through simulations of the nominal data flow for the full FTK system (64 towers) at $\mu = 57$.

$$
\begin{aligned}
&0b0000 - 0b0000 - 0b0000 - 0b0000 \qquad BF = 0\\
&\Downarrow\\
&0b0111 - 0b1010 - 0b0000 - 0b0000 \qquad BF = 5\\
&\Downarrow\\
&0b0111 - 0b0101 - 0b0000 - 0b0000 \qquad BF = 4\\
&\Downarrow\\
&0b0000 - 0b0000 - 0b0000 - 0b0000 \qquad BF = 5
\end{aligned}
$$

$$\text{Total BF} = 14/4 = 3.5$$

where the first and last 16 bit words are the IDLE word.

This dummy hit is characterized by a mean IDLE power consumption of about 25 *W*, as can be seen from Figure 4.2a. The second dummy hit value identified is 0x7a0075a0, corresponding to a bit flip value of 4.5. This dummy hit provides a mean IDLE power consumption of about 31 *W*, as shown in Figure 4.2b. In the following, the lowest bit flip value will be considered the standard for operations.

The second component contributing to the LAMB power consumption is the mean number of bit flips inside the data and the rate at which these data are sent to the AM chips for comparison. Considering a maximum number of modules per FTK tower of 53 and a maximum number of SSID per module of 88 (these are the expected value for the Run3 FTK running conditions), the maximum value of SSID in the data is 4664, corresponding to a 13 bits word. Considering continuous data with random SSID distributed flat between 0 - 8191 (using 13

bits), the power consumption of the LAMB has been measured to be about 48 W. This value has been measured feeding the AMB with continuous pseudodata, ensuring no dummy hit contribution to the power consumption value.

In order to have a conservative estimation of the rate, the worst case scenario for the Run3 running condition has to be considered. In Table 4.1, the nominal FTK data flow, considering all the 64 FTK towers and a pile-up value of $\mu = 57$, is presented. The maximum number of clusters among towers and layers expected for these conditions is 340 clusters/layer. This value has been chosen to be used for the computation of the power consumption, giving a 22 % margin over the mean value on the different layers. It is important to notice that the cluster number doesn't depend on the PatternBank used, but only on the geometry of the FTK towers and on the pileup. Using a linear approximation for the scaling of the cluster number with the pileup, a number of 360 clusters/layer is considered for the $\mu = 60$ case, while 480 clusters/layer are considered for the $\mu = 80$ case.

It is then possible to perform the computation of the expected AMB total power consumption for the two pile-up cases $\mu = 60$ and $\mu = 80$. Here, the FTK design rate of 100 kHz is considered. Assuming a mean number of 360 clusters to be processed every 10 µs at a $\mu = 60$, and given a 100 MHz clock cycle for the AM chips, the 36 % of the cycles are spent processing data, with a power consumption of 48 W. The remaining cycles are spent in IDLE mode, with the board consuming the 25 W given by the dummy hit. The resulting total AMB power consumption, at a pileup $\mu = 60$ is:

$$\begin{aligned} AMB_{power} &= AMB_{core} + 4 \cdot (LAMB_{core}) = \\ &= 100\,W + 4 \cdot (48\,W \cdot 0.36 + 25\,W \cdot 0.64) \approx 233\,W \end{aligned} \tag{4.2}$$

with a mean LAMB power consumption of about 33 W. Considering, instead, a possible pileup of $\mu = 80$, the total AMB power consumption grows to 244 W, with a LAMB consumption of 36 W.

Given the output of the FTK review, and taking into account an FTK rate reduction from 100 kHz to 35 kHz, the board power consumption is reduced to 212 W (28 W per LAMB) at both $\mu = 60$ and $\mu = 80$ (in this case the dummy hit value becomes predominant in the power consumption computation). However, given that the review outcome doesn't prevent the possibility for FTK to run at full rate if resources will be found, the 100 kHz processing rate at $\mu = 60$ case will be taken as reference in the rest of the chapter.

The total VME rack power can be computed as the sum of the different boards powers, plus the fan tray system already mentioned in Section 3.2 and treated in more details in Section 4.3. As will be shown later, the fan tray system has a power consumption of 1.7 kW for crate (couple of bottom/upper fan trays). In Table 4.2, a summary of the various VME components power consumption is shown. Considering a 90 % efficiency for the power supply, the total VME rack power is:

FTK VME component	Power (W)	Component number
AUX	70	32
SSB	160	8
AMB	212-244	32
Fan tray couple	1700	2
Full rack	$\approx 16\,000$	

Table 4.2: Table summarizing the power consumption of the different FTK VME components of each FTK rack.

FTK VME board	Max operational temperature (°C)	Max temperature (°C)
AUX	85	125
SSB V_4	85	100
SSB V_5	100	125
AMB	80	100

Table 4.3: Table summarizing the maximum operational temperature and the maximum temperature that the boards can stand without suffering of damages. Operating this boards under the operational limit can increase the components life time.

$$\begin{aligned} \mathrm{Rack}_{power} = (((\mathrm{AUX}_{power} + \mathrm{AMB}_{power}) \cdot 16 + \mathrm{SSB}_{power} \cdot 4) \cdot 2 + \\ + \mathrm{Fan}_{power} \cdot 2) \cdot \mathrm{PS}_{eff} \approx 16\,kW \end{aligned} \tag{4.3}$$

The standard VME cooling, coupled with the standard rack water cooling system, is not enough powerful to dissipate a total power of $\approx 16\,kW$. Custom solutions has therefore been adopted, and will be presented in the following sections.

4.2 TEMPERATURE LIMITS

Each board component has a given temperature range in which its safe operation is guaranteed. The maximum temperature threshold is generally provided by the productor of the given component. Operating an electronic board over the temperature threshold can lead to accuracy and performances issues, and eventually in damages to the component itself. Moreover, the operational temperature can reduce the life time of the electronic component [57]. Keeping the temperatures low during operation can let the FTK boards to drastically increase their life time, reducing the probability of failures. Given the high number of FPGAs (about 2000 elements) and of ASICs (about 8000 AM chips) elements that compose the FTK system, reducing the component failure risk is of primary importance for the project itself.

Figure 4.3: Temperature profile of some of the LAMBs during a cooling test. In the test, 16 AMB were running in high power conditions. In this test, the FW temperature protection of two of the LAMBs used has been triggered. From the plot it is possible to see the quick increase of the temperatures of the LAMB 2 (AMB 13 and AMB 1), followed by a quick decrease due to the stop of the dataflow triggered by the AMB FW protection. The protection threshold is set to 80 °C. All the other LAMBs were instead running stably at about 70 °C.

For AUX and SSB the temperature limit is driven by the processing FPGAs. The AUX board is equipped with 6 commercial **Altera Arria-V** FPGAs. This FPGA is rated to an operational temperature limit of 85 °C [58]. No damages to the component are anyway expected for temperatures up to 125 °C. Concerning the SSB board, as mentioned in Section 3.3.1, two different versions are produced, each of which mounts a different FPGA version. The V4 SSB mounts 4 **Xilinx kintex-7** commercial FPGAs. This FPGAs are rated for an operational temperature of up to 85 °C [59]. The temperature threshold for the V5 SSBs is higher. This version of the board is equipped with the same FPGA, but its industrial version that is rated for temperatures up to 100 °C.

The AMB is equipped with 4 FPGAs. The HIT and ROAD chips are commercial **Xilinx Artix-7** FPGAs, rated to an operational temperature threshold of 85 °C [60]. The AMB data control logic is instead based on two **Xilinx Spartan-6** FPGAs, that present the same temperature thresholds [61]. For both the FPGA versions, no damages are anyway expected for temperatures up to 125 °C.

Beside the FPGA limits, the most temperature sensible elements of the AMB are the AM chips. Being a custom elements, an exhaustive data sheet is not available. Estimations have rated the AM chips to be damage-safe up to 100 °C, but a conservative operational temperature threshold has been set to 80 °C. Table 4.3 summarizes the operational and maximum temperatures for the different VME boards.

In order to prevent possible damages, a protection system has been put in place. This system relies on two different components. The first component is a FW protection, able to immediately stop the board dataflow in case of the measured temperature is found to overcome a given threshold. The FW protection is already available on the

SSB FPGAs, for both the board versions, and on the AMB FPGAs. A further FW protection has been included in the AMB, monitoring the temperature of the AM chips. While the FPGA temperatures are directly measured by the components themselves, the AM chips measurement relies on temperature sensors placed around the AM chips, on the LAMBs. This protection stops the data to be sent to the AM chips in a whole LAMB, letting the temperature to decrease. In Figure 4.3, a test in which the FW AMB protection was triggered is shown. In two of the LAMBs it is possible to see the behaviour of the temperature increasing until the threshold, and starting to decrease soon after, thanks to the stopping of the dataflow on the specific LAMB.

While it is planned for the future, currently a FW protection on the AUX board is not yet in place. It has anyway to be noted that no temperature problems are expected for this board, for the following reasons:

- the power consumption of the AUX is much lower than the other boards one;
- the AUX boards are placed in the back of the crate, while the other high power consuming boards are placed in the frontal part of the crate. As will be shown in Section 4.4, the back part of the crate is characterized by much lower temperatures;
- a SW protection has been put in place.

In addition to the FW component, the second one is a SW protection, managed by the ATLAS DCS (Detector Control System) and able to shutdown the whole rack in case of problems. The DCS is a highly distributed system that is used to supervise and monitor the ATLAS experiment infrastructure. It is used to:

- put the detector hardware into selected operational conditions;
- continuously monitor and archive the detector hardware run-time parameters;
- automatically perform corrective actions if necessary;
- provide an user interface to manage the detector hardware.

The DCS is used to control and monitor the FTK infrastructure, retrieving data such as the board temperature, the air temperature in the crate, in the fan trays and in the racks, the fan speeds and the Power Supply parameters. In addition, the DCS manages the cooling system in response to the retrieved data, as well as managing the PS emergency interlocks. Exploiting this, the DCS has been programmed to read the rack air temperature, and to trigger the PS interlock in case the read value overcome a given threshold. The interlock shuts

down the power to the rack, keeping only the power to the cooling system (the fan trays). The rack air temperature thresholds at which the DCS triggers the shutdown was preliminary set to 32 °C, but, as will be shown in Section 4.4, an optimized version has been studied during the last set of cooling tests.

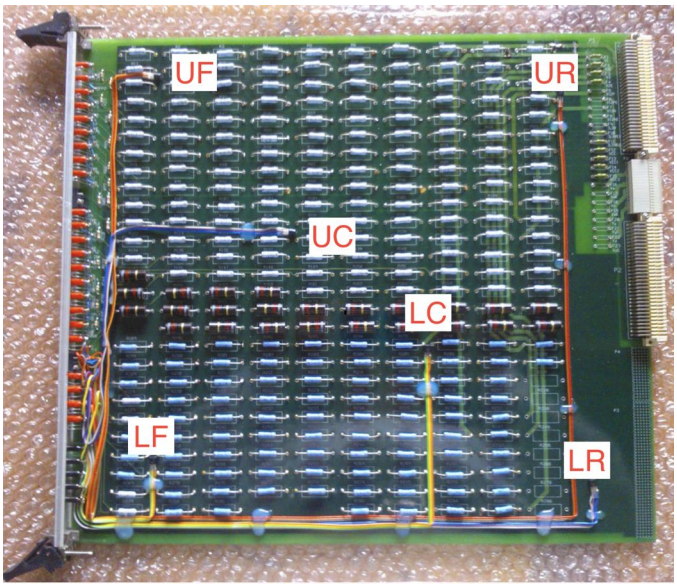
Finally, in order to overcome the lack of a FW protection on the AUX boards, the DCS has been also programmed to monitor the AUX FPGA temperatures, and to shutdown the given crate in case one of them is found to lay above the operational temperature threshold.

4.3 COOLING INFRASTRUCTURE

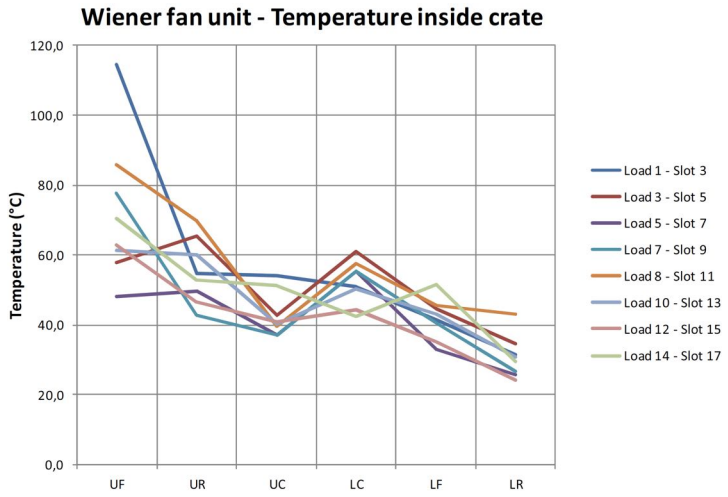
As shown in the previous sections, the VME cooling infrastructure has to face challenging requirements due to the very high power density that has to be dissipated and the temperatures inside the rack that have to be taken under the electronic components operational limits.

In order to be able to dissipate a power of more than $\approx 16\,kW$, six 1 U heat exchangers are mounted in each VME rack. The heat exchangers are cooled by mean of a cold water flow injected at 14 °C, with a pressure $\Delta P = 4.5\,bar$. These heat exchangers, in the given configuration, are proven to be able to dissipate $3.8\,kW$ of power each [62]. A so high number of heat exchangers, on the other hand, increases drastically the resistance to the flowing of the air inside the rack. Together with the high board density (increasing the air flow resistance as well), this poses the requirement of powerful fans, able to help the cooling system by increasing the air flow and its uniformity between the different crate slots.

Four custom made fan trays, composed of 9 powerful fans (able to run with a speed above $10.5\,krpm$) has been developed and produced by the Pavia ATLAS group. The choice of custom fan trays is driven by the fact that the commercial available ones don't provide enough air flow through the rack components, as demonstrated in studies performed during the development phase of the FTK VME cooling infrastructure. In Figure 4.4, one of the results of these studies, executed using high performance commercial Wiener fans, is shown. Since at the moment of the test the FTK boards were not yet available, load boards (VME boards composed of a series of resistors only) have been used to test the temperature of the system in a realistic condition. During this test, 15 load boards were housed in the VME crates. Each board was able to dissipate $298\,W$ of power, for a total per crate of $4470\,W$ (to be compared with the $5660\,W$ expected in the worst FTK running conditions, as presented in Section 4.1). As can be seen from the figure, some of the sensors were exceeding the 80 °C limit imposed by the AM chips. These commercial fan trays, characterized by 6 fans able to run at a maximum speed of $5\,krpm$, were not powerful enough to cope with the high resistance of the air flow inside the rack. This



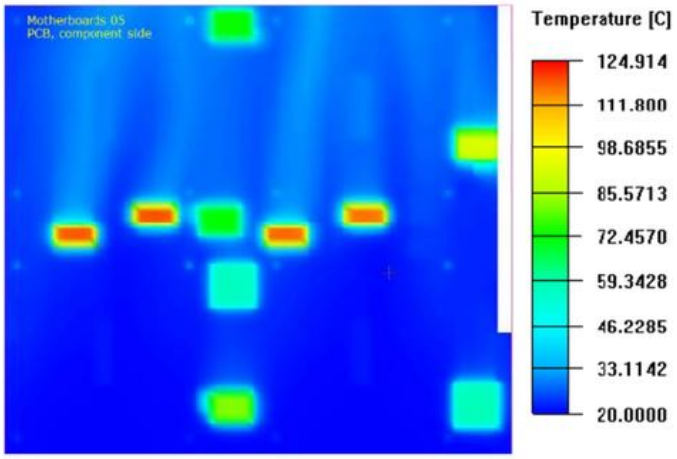
(a)



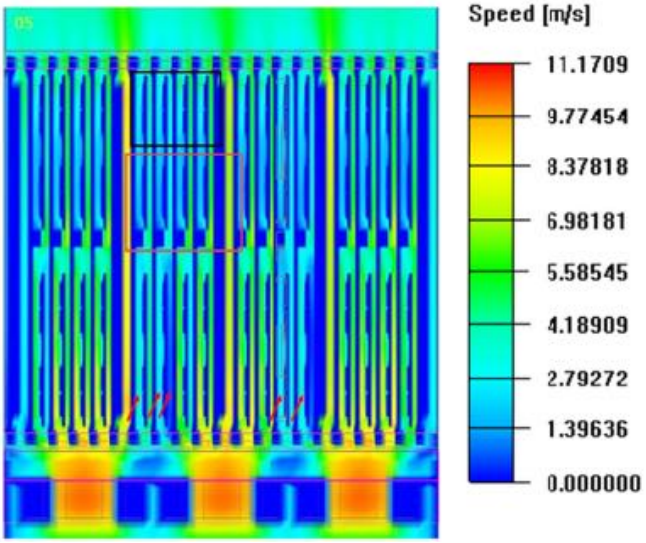
(b)

Figure 4.4: Results from a preliminary cooling study of the VME infrastructure, performed during the design phase of the FTK cooling system. In Figure (a), a picture of one of the 15 load boards housed in each VME crates is shown, providing the names and positions of the different temperature sensors. Figure (b) shows the temperatures measured by the different sensors for 8 of the used slots. As can be seen, some slots have sensors reading temperatures higher than the threshold limit of 80 °C.

was confirmed by mean of a computational simulation performed to study the air flux inside the VME crate [63]. Even if the results of the simulation are not reliable enough due to the oversimplification of the system (the results were considered too conservative), it predicted very high temperatures in some regions of the boards ($\approx 120\,^{\circ}\mathrm{C}$) and a lack of temperature uniformity in the slots, due to the not high



(a)



(b)

Figure 4.5: (a) Temperature simulation of the air flow on a prototype of AMB board. The image shows a view of the AMB side, where the electronic components are placed. The hot spots are the board power supplies. (b) Frontal view of a VME crate air flow simulation. The boards are not cooled down evenly.

enough air flow produced by the fans. In Figure 4.5, some of the results from the simulation are reported, where it is possible to note the high temperature reached by some of the boards components and the distribution of the air velocity inside a crate.

In the final FTK VME cooling system composition, each VME rack is equipped with the 6 heat exchangers, the standard rack turbine, and by four of the custom fan trays, placed just under/above the crates. Figure 4.6 shows a sketch of the final VME rack configuration, together with a real picture of a fully populated rack.

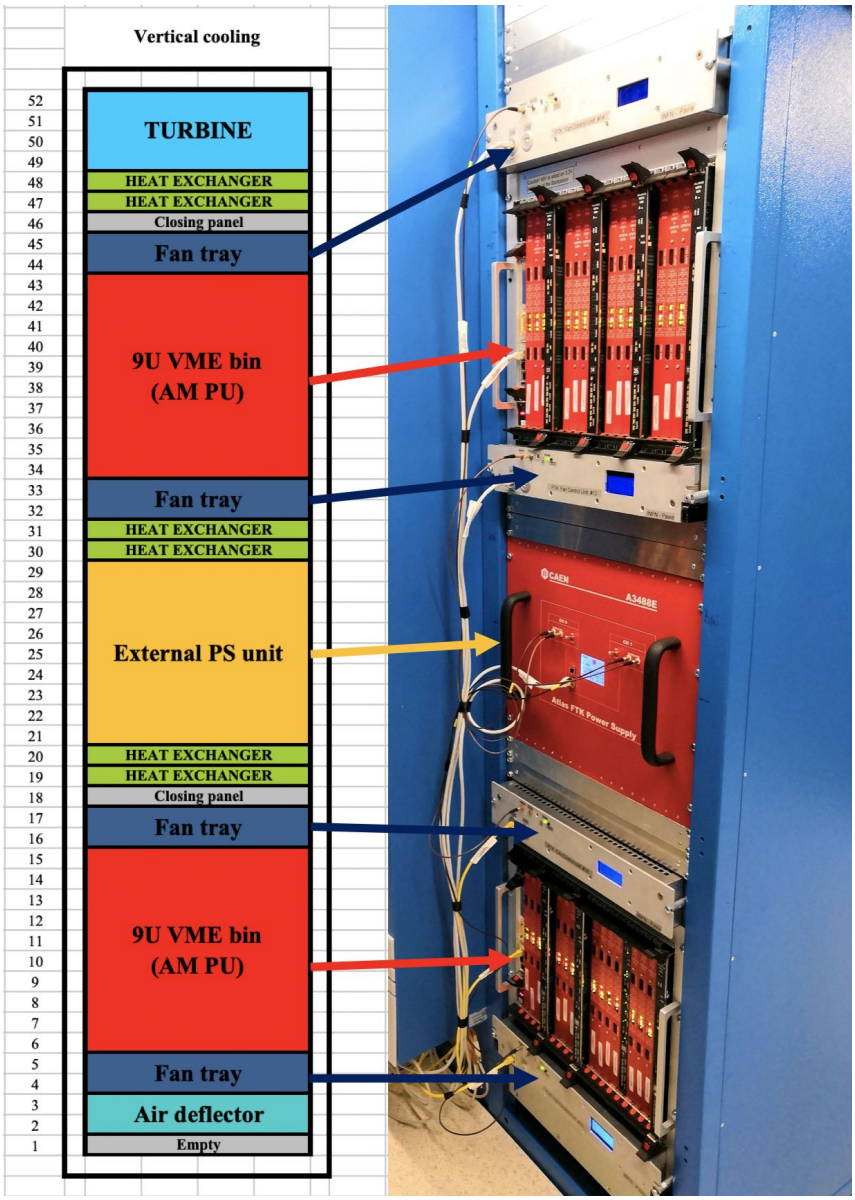


Figure 4.6: Sketch representing the VME rack configuration, together with a picture of a fully populated rack.

The 9 fans of each fan tray are organized into 3 rows of 3 fans each, two rows in the front of the rack, where the AMB and SSB boards are located, and one row on the back, where the AUX cards are placed. The mechanics are different for the bottom and top fan tray versions, in order to leave 1 U of open space between the fans and the bin for the bottom version.

Electronic control and monitoring are implemented with an Arduino microcontroller, Mega 2560 type, equipped with a local display for every fan tray. Local/remote ON/OFF is available, and four speeds (40 %, 60 %, 80 %, 100 %) are settable independently within the three rows of fans. Six temperature sensors are positioned inside the fan tray

to monitor the environmental temperature. The fan speed selection is controlled, through the Arduino, by the DCS system already presented in the previous section. The speed of the fans is automatically managed and increased/decreased by the DCS with the increase/decrease of the temperatures measured inside the fan trays.

The speed for the worst temperature condition was initially set to 100 %-100 % (bottom-top tray), but as will be shown in the next section, it has been optimized after the cooling studies.

For testing purpose only, a chiller, able to cool down the water in entrance to the rack cooling system (i.e. to the heat exchangers), was provided for the FTK racks. Also this component has been tested in the last cooling studies, but, as will be shown in the next section, due to the small improvements in the cooling performances and the risk of condensation that colder water adds to the system, its use has been discarded.

Even if some preliminary studies on the final cooling system were executed at the very beginning of the commissioning of FTK, a more precise and complete study has been performed in the last year. The study was meant to certify the ability of the cooling system to maintain the temperature of the boards under the limits, and to study optimizations on the system parameters (as the temperature threshold for the SW shutdown and the optimal fan speed configuration). These study will be presented in the next section.

It is important to note that the fall-back solution for a negative outcome of the study was to equip additional racks in which distribute some of the boards (reducing the crate board occupancy and, thus, the cooling requirements). This solution would led to a not negligible increases in the project costs.

4.4 COOLING STUDIES

A series of cooling tests were performed, reproducing the board working conditions expected for the Run3 of the LHC data taking period. The cooling studies have been performed over a two years period, but for the sake of compactness, in the following only the final results will be presented.

In the first part of this section, some details about the testing conditions will be provided, as the configuration of the boards used (both the physical positions of the boards in the crates and the FW/data configurations), the configurations of the cooling system, and the scripts developed to study the temperatures and the power consumptions.

In the second part of the section the results of the study about the temperatures reached inside the racks will be presented, with some considerations about the goodness of the cooling system.

Finally, the optimizations of the cooling system that have been identified during the studies will be shown. Some of these optimizations

have already been used during the last tests, while other will be made available by the start of the new LHC Run.

4.4.1 *Cooling test setup*

The cooling tests have been performed using one of the FTK racks placed in the USA 15 ATLAS cavern. The rack was fully equipped, with a FTK power supply, two VME crates fully loaded of boards, and the final version of the cooling system (in the same configuration as the one presented in the previous section). A picture of the rack used in the cooling tests can be found in Figure 4.6. In Figure 4.7, details about the boards positions inside the crate are provided (Figure 4.7a), together with a closer image of one fully loaded VME crate (Figure 4.7b).

For the tests, the AUX cards were made running in loop mode on a sample of pseudodata directly loaded in the input FIFOs. The power consumption of this board was of about 70 W, as presented in Section 4.1.

Both the two versions of SSB boards were used for the tests. However, because of the output of the SSB review, that requested minor hardware modifications on the V4 SSBs (as presented in Section 3.3.1), only one of that board version was available and, thus, used for the tests. All the SSBs were running on data provided by an emulator, directly connected to the boards by the respective RTM cards. The power consumption of these boards was about 160 W.

The AMB boards were run in loop mode on different set of pseudodata. The set of pseudodata to be used for the given test was selected during the board configuration procedure. The different pseudodata sets were produced with different predefined bit flip values, in order to select a different power consumption of the AM chips for each of the measurements. The number of bit flips chosen was of 6, 6.5 and 7. This choice was made in order to reproduce the power consumptions expected by the Run3 conditions, described in Section 4.1. The value of the dummy hit, instead, was kept constant during all the tests. The value chosen was 0x7a007500, corresponding to a bit flip of 3.5. Figure 4.8 presents the measured values of the LAMBs power consumption for the three different pseudodata sets used for the tests.

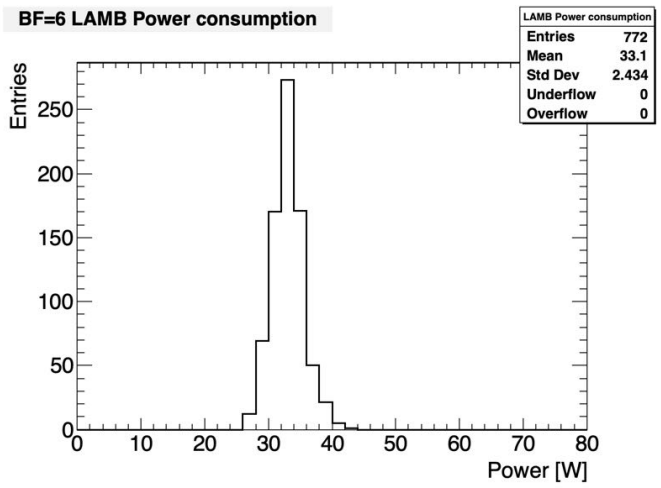
In Figure 4.8a and 4.8b, the pseudodata sets corresponding to a bit flip value of 6 and 6.5 are presented. These two sets of pseudodata correspond to the Run3 100 kHz FTK running conditions with a pile-up value of $\mu = 60$ and $\mu = 80$, respectively. The pseudodata set shown in Figure 4.8c, containing bit flip 7 pseudodata, exceed the worst case power scenario expected for the future running conditions. However, it has been considered interesting in order to evaluate for possible margins available in case of unexpected variations of the total rack power.

Slot	Bottom VME crate	Top VME crate
1	sbc-ftk-rcc-02	sbc-ftk-rcc-01
2	AMB+AUX	AMB+AUX
3	AMB+AUX	AMB+AUX
4	AMB+AUX	AMB+AUX
5	AMB+AUX	AMB+AUX
6	SSB-V5	SSB-V5
7	AMB+AUX	AMB+AUX
8	AMB+AUX	AMB+AUX
9	AMB+AUX	AMB+AUX
10	AMB+AUX	AMB+AUX
11	SSB-V5	SSB-V5
12	AMB+AUX	AMB+AUX
13	AMB+AUX	AMB+AUX
14	AMB+AUX	AMB+AUX
15	AMB+AUX	AMB+AUX
16	SSB-V4	SSB-V5
17	AMB+AUX	AMB+AUX
18	AMB+AUX	AMB+AUX
19	AMB+AUX	AMB+AUX
20	AMB+AUX	AMB+AUX
21	SSB-V5	SSB-V5
Rack 05-07		

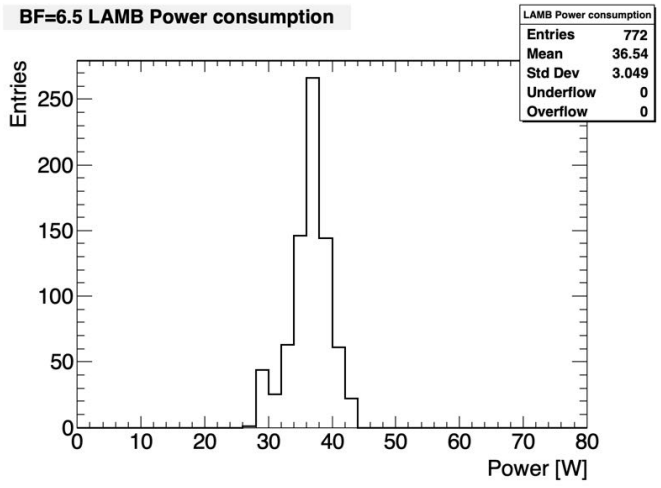
(a)

(b)

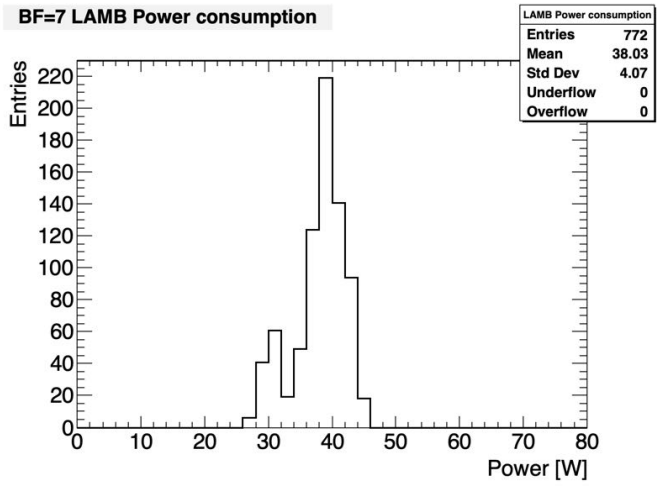
Figure 4.7: (a) Sketch of the FTK rack (rack 5-7) used during the cooling tests. The slots of each crate in which the 16 PUs and the 4 SSBs are placed are shown. Only one V4 SSB was used during the tests, placed in slot 16 of the bottom crate. (b) Picture of the top VME crate used during the cooling tests. The red boards are the 16 AMBs, while the black ones are the 4 SSBs. In slot 1 is it possible to see the Single Board Computer used for the control of the boards.



(a)



(b)



(c)

Figure 4.8: Distribution of the LAMB power consumption for the different values of bit flip (BF) pseudodata: (a) 6, (b) 6.5, and (c) 7.

The configuration of the boards for the tests was executed in two different ways, one for the PUs and one for the SSBs. The PU configuration was performed using the standard ATLAS online software framework, that will be described in more details in the next chapter. The SSB configuration, instead, was executed via a custom command line tool. As mentioned earlier, the SSB boards were fed via an emulator directly connected to the SSB RTMs. The emulator SW is not integrated in the FTK framework, thus, a python script was developed and run in parallel to the PUs partition.

At the time of the tests, the boards temperature information was only accessible through custom command line tools, not integrated in the FTK online SW. In order to cope with this, for the measurement session a python wrapper of boards standalone tools has been developed. This wrapper was able to read directly from the boards FPGAs the temperature registers and to perform the conversions required. Each measurement was performed in intervals of about 300 s. In order to have a direct measurement of the AMB power during the tests, the AMB DC/DC converters values of current and voltage were also measured. This was not required for AUX and SSB, that show, as presented in the first section of the chapter, stable power consumption.

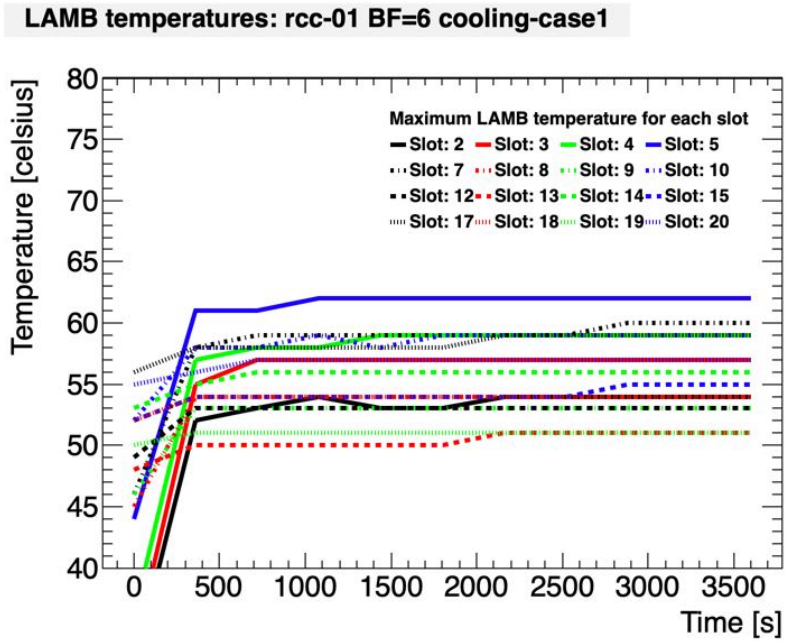
The temperature of the fan trays and of the rack air was also measured. The data in this case were directly retrieved from the DCS. The DCS was also used to provide the values of the power provided during the tests by the rack power supply. This information was useful in order to perform a cross check between the power expected from the AUX and SSB boards and the measured power consumed by the AMBs.

During the tests the SW temperature safety threshold has proven to be too low (originally set as a safely best guess). The SW safety system, monitoring the rack air temperature and shutting down the PS in alarming conditions (as presented in Section 4.2), was frequently triggered when the board temperatures were still under control. In order to overcome the problem, for the time of the tests this safety system was disabled.

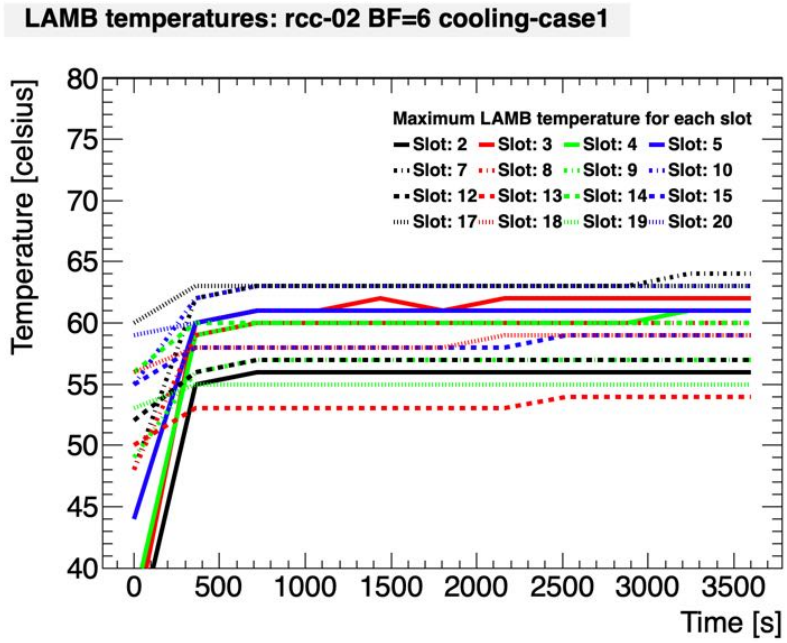
Also the SW controlling the speed of the fans was disabled. As will be shown in the last section of this chapter, the 100 %-100 % speed of the fan has been proved not to be the optimal one (for the worst temperature cases). Moreover, the automatic variation of the fan speeds was contributing to misleading results. In order to overcome this problem, the fan speed was manually set at the optimal configuration (60 %-100 %).

4.4.2 *Cooling tests in standard cooling configuration*

The set of results presented in this section refers to a standard cooling configuration, i.e. the one described in the previous subsection.



(a)



(b)

Figure 4.9: Temperature profile of the hottest LAMB of each AMB for the top VME crate (a) and bottom VME crate (b). The AMB operational temperature limit is 80 °C. In this configuration all the temperatures are well under control.

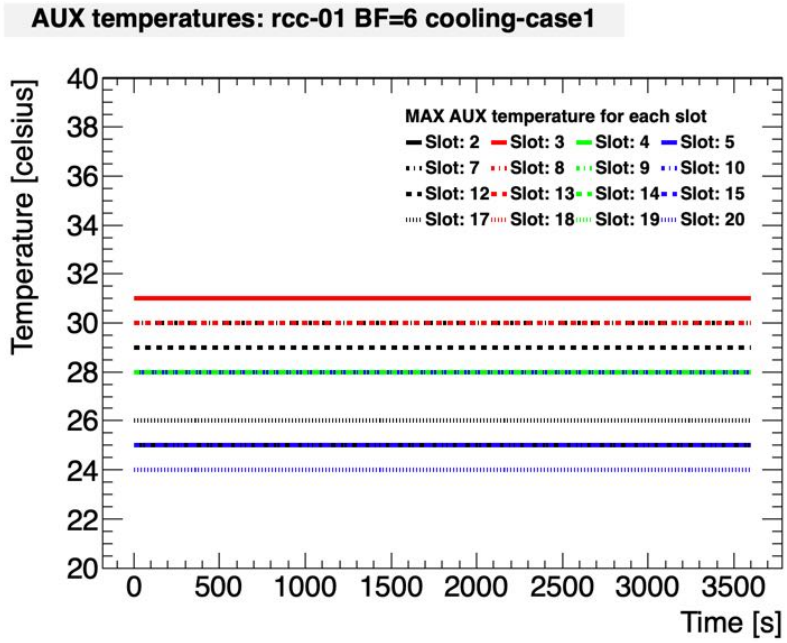
In Figure 4.9, the temperature profile for the AMB boards, measured during the test, is provided. This plots refer to a bit flip 6 pseudodata configuration. The temperature limit for this board is 80 °C. In the plot, the maximum temperature of all the 4 LAMBs of each slot AMB

is presented. Figure 4.9a refers to the measurements of the upper VME crate (rcc-01, as the name of the SBC controlling that crate) while Figure 4.9b refers to the bottom one (rcc-02). This convention will be kept for the rest of the chapter. From both the plots it is possible to note an initial rapid increase of the maximum temperatures, due to the heating up of the boards. This increase stops just after the second measurement, where the plateau is reached. In this configuration, the temperatures are well under the operational temperature limit.

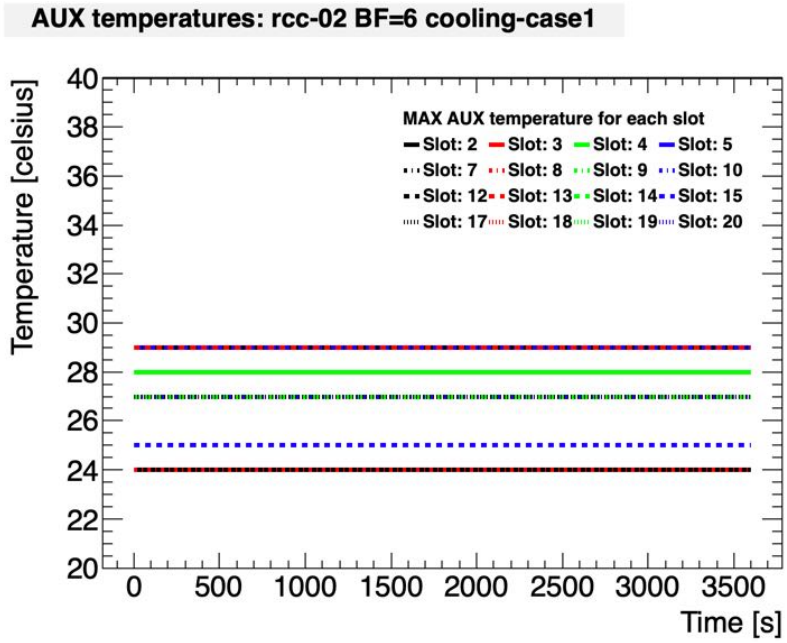
From the comparison of the two crate results, the upper crate temperatures are slightly lower than the lower crate ones. This is expected due to the position of the heat exchangers inside the rack. As can be seen in Figure 4.6, only two heat exchangers will cool down the air coming from the upper VME crate, while 4 of them are placed between the lower and the upper crate. A solution to this problem would be to move one of the heat exchangers that are surrounding the PS to the very bottom of the rack, just after the air deflector. Since the difference in the temperatures between the two crates is, anyway, small, this is not considered an issue.

Figure 4.10 shows the same measurements performed on the AUX cards. Due to the low power consumption of this kind of boards, the temperatures are very stable. Being the first measurements made after the end of the AMB first measurements, the AUX card only shows the plateau of the temperature (no initial increase can be seen). To be noted that these boards are placed in the back side of the crates. This means that they don't suffer from the heating released by the more power consuming AMB and SSB boards. Also in this case, the temperatures are well under control. Since no differences in the AUX temperatures with respect to this test have been observed, in the following the AUX results will not be presented.

Figure 4.11 shows the same measurements performed on the SSBs. Also in this case the measurements were performed after the AMB ones, thus, only the plateau can be observed. The single V4 SSB used in the test is placed in slot 16 of the bottom crate (rcc-02). All the temperatures are well below the operational limit, for both the versions of SSBs. The SSB placed in slot 21 shows temperatures of about 6 °C higher than the other boards. This is due to a non perfect distribution of the air flow inside the crate. To better understand this problem, Figure 4.12 shows the same test measurements, but including the temperatures of all the 4 SSB FPGAs rather than only the hottest one. As can be observed from the plots, the hottest FPGA of each crate is always the number 2 of slot 21. This FPGA is placed on the top back corner of the SSB, the one more difficult to be reached by the air flow produced by the fans. Interesting is also the fact that the two upper FPGAs of each slot (FPGAs number 2 and 3), are usually hotter than the lower FPGAs (number 0 and 1). Moreover, between the lower FPGAs, the one placed in the back of the board is usually hotter than



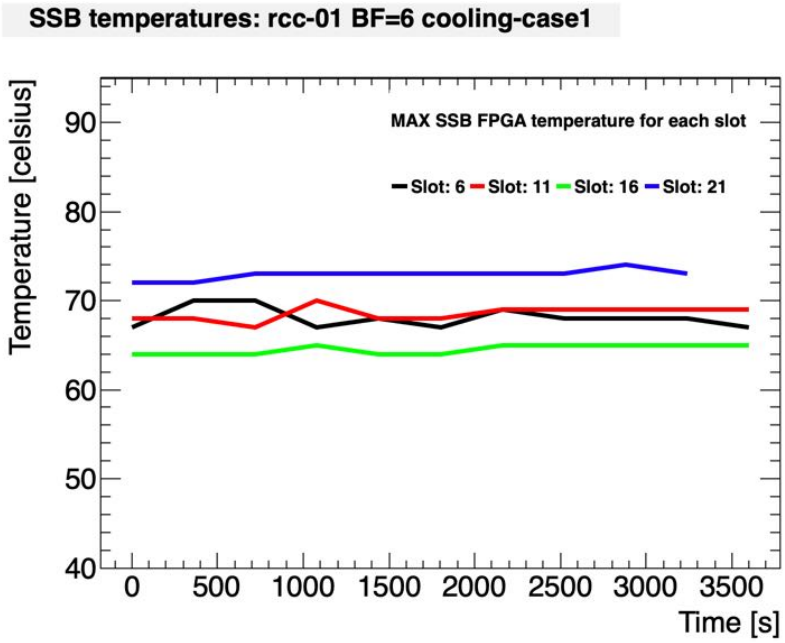
(a)



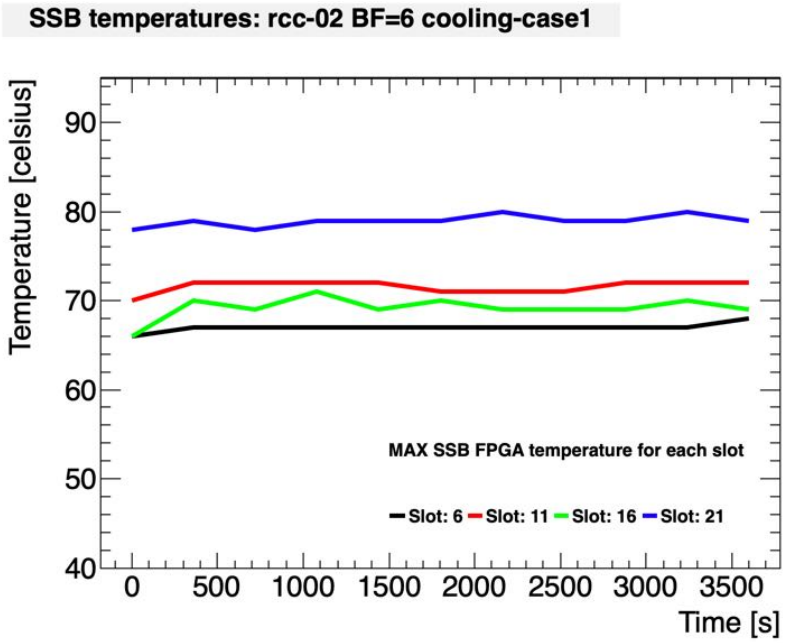
(b)

Figure 4.10: Temperature profile of the hottest FPGA of each AUX for the top VME crate (a) and bottom VME crate (b). The AUX operational temperature limit is 85 °C. In this configuration all the temperatures are well under control.

the front one. Besides not much can be made in order to improve the air flow distribution, these results are interesting for choosing the final position of the boards inside the crate. A V4 SSB, for example, has not

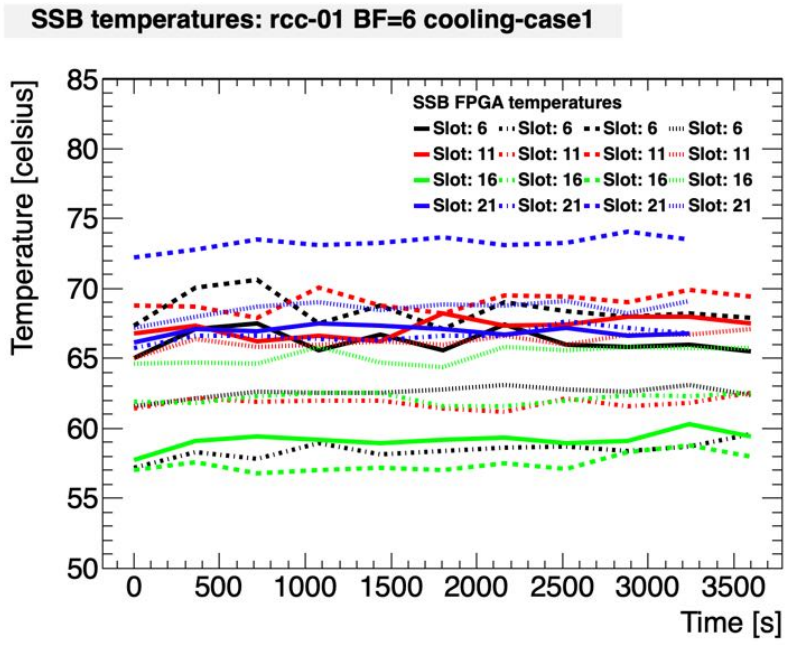


(a)

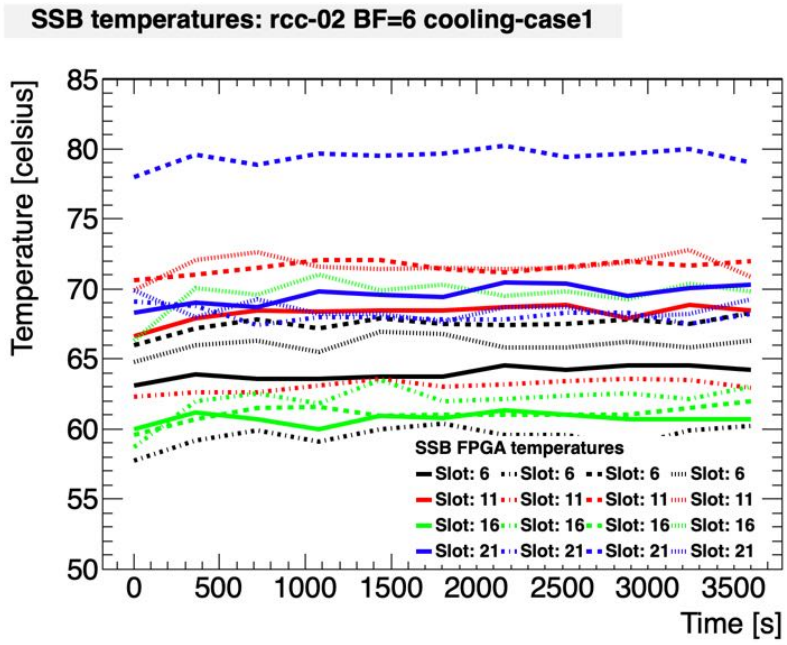


(b)

Figure 4.11: Temperature profile of the hottest FPGA of each SSB for the top VME crate (a) and bottom VME crate (b). The SSB operational temperature limit is 85 °C for the V4 boards and 100 °C for the V5 ones. The only V4 SSB of the test is placed in slot 16 of the bottom crate (rcc-02). In this configuration all the temperatures are well under control.



(a)



(b)

Figure 4.12: Temperature profile of the 4 FPGAs of each SSB for the top VME crate (a) and bottom VME crate (b). In the plot, each dashed line correspond to a different FPGA. The FPGA number of which each line correspond is the same as it order of appearance in the legend (e.g. continuous line: FPGA 0).

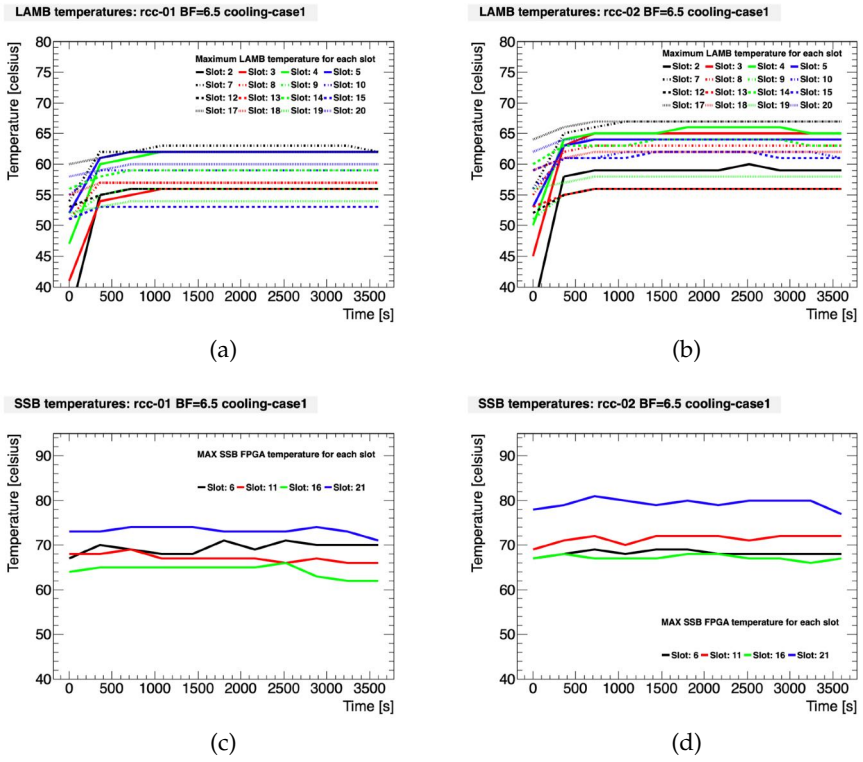


Figure 4.13: Profile of the maximum LAMB temperatures (a),(b) and of the maximum SSB FPGA temperatures (c),(d) for the BF 6.5 AMB configuration. All the temperatures are well under the respective board operational limits.

to be placed in slot 21, where the 80 °C measured are very close to the operational temperature limit for that board.

Figure 4.13 shows the results obtained for the second pseudodata configuration of the AMBs (BF 6.5). The LAMB maximum temperatures are slightly increased with respect to the BF 6 case, but still under control. The SSB temperatures are instead the same as the ones measured in the previous test. This confirm that the heat is correctly dissipated, and the AMB temperature increase doesn't interfer with the SSB temperatures.

Finally, Figure 4.14 shows the results obtained for the last configuration case, BF 7. Also in this configuration, the LAMB temperatures are slightly higher with respect to the previous test, but still well below the alarming threshold. Concerning the SSBs, as for the previous case the temperatures are not changed.

As shown from the plots, these tests proved the ability of the FTK cooling system to keep the boards temperatures under control. Moreover, margin over the Run3 worst case scenario running conditions have been found, as confirmed in the test that used BF 7 pseudodata.

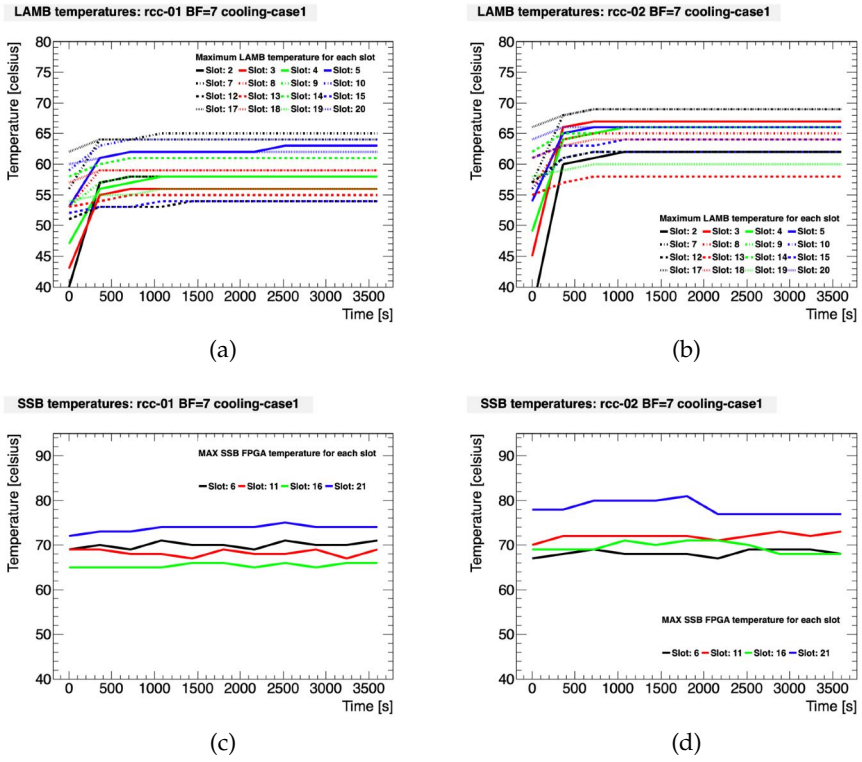


Figure 4.14: Profile of the maximum LAMB temperatures (a),(b) and of the maximum SSB FPGA temperatures (c),(d) for the BF 7 AMB configuration. All the temperatures are well under the respective board operational limits.

4.4.3 *Cooling tests in improved cooling configuration*

As presented in the previous sections, keeping the temperatures of the boards components under their operational temperature limit is not the only goal of the FTK cooling system. A working environment characterized by lower temperatures can increase the life time of the electronic components, reducing the probability of both processing errors and failures. In order to study the possibility of decrease the temperatures inside the crates measured with the standard cooling configuration (presented in the last section), two more tests have been performed.

In the first test, the water flow of the cooling system has been increased. The flux was moved from the standard 4.5 bar to 5 bar, lowering the difference of temperature between the input and output water and slightly increasing the cooling capacity of the system.

In Figure 4.15 and Figure 4.16, the results from the tests, performed with AMB pseudodata of respectively BF 6 and 7, are shown. As can be seen comparing the plots of Figure 4.9 and Figure 4.11 with the ones in Figure 4.15, the plateau of the maximum temperatures are slightly decreased, for both the LAMBs and the SSB FPGAs. A similar

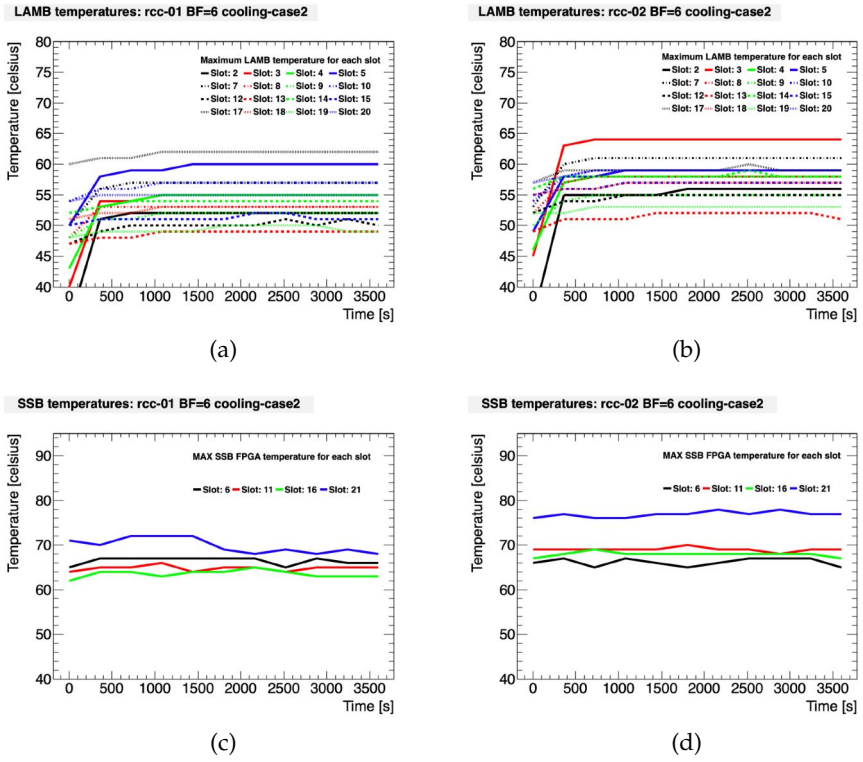


Figure 4.15: Profile of the maximum LAMB temperatures (a),(b) and of the maximum SSB FPGA temperatures (c),(d) for the BF 6 AMB configuration. The test was executed with an increased water flux of the FTK cooling system of 5 bar.

result can be observed comparing the plots of Figure 4.14 with the ones of Figure 4.16.

The second test was performed by reverting the water flux to the standard 4.5 bar and enabling the water chiller. The chiller was installed on the FTK racks input water pipes for testing purpose only (as described in Section 4.3). During the tests it was used to decrease the input water temperature of about 2.5 °C, bringing it from the standard 14 °C to 11.5 °C. A further decrease of the water temperature was not possible because of the risk of condensation inside the rack. The decrease of the input water temperature has the con of increase the rack dew point. Consequence of this is the formation of dew around the water pipes inside the rack, that can drop into the electronic boards. The consequences for the electronic boards of that possibility are obviously catastrophic.

To avoid such a possibility, during the tests all the FTK racks, apart the one under testing, were shut down, while the one used was kept under strict observation. Despite this, dew formation has been observed during the test, as can be seen from the Figure 4.17. A possible solution for this problem would be the use of dried air inside the rack (e.g. by including an air dryer inside the rack). This would

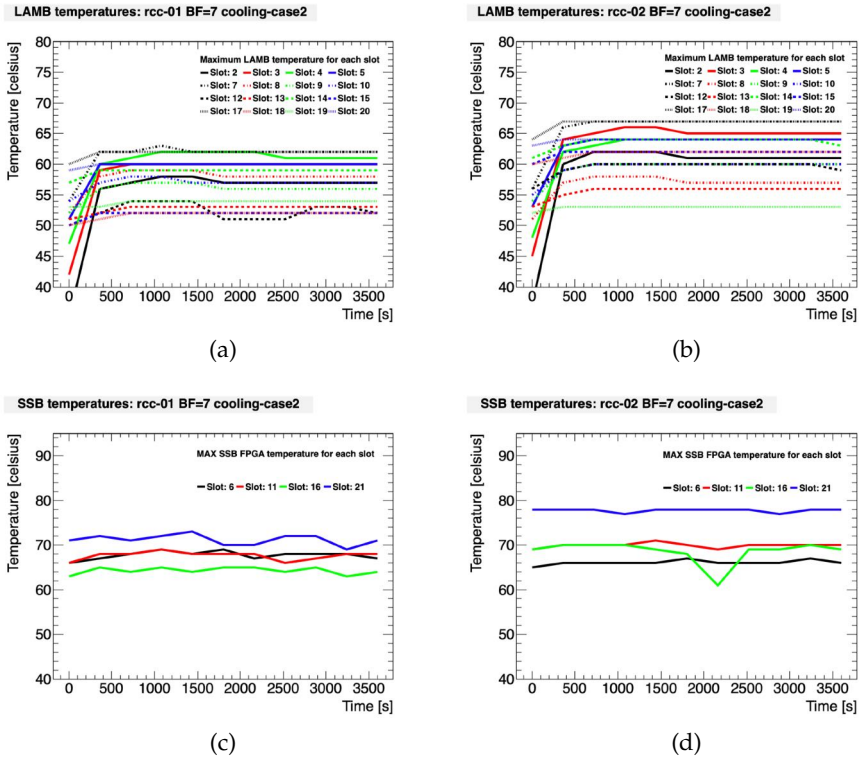


Figure 4.16: Profile of the maximum LAMB temperatures (a),(b) and of the maximum SSB FPGA temperatures (c),(d) for the BF 7 AMB configuration. The test was executed with an increased water flux of the FTK cooling system of 5 bar.

help to decrease the dew point and to keep it under control. For the test execution, however, a drier was not used.

In Figure 4.18 and Figure 4.19, the results of the tests performed in this cooling configuration, running the AMB with pseudodata of respectively BF 6 and 7, are shown. From these tests it is possible to see that the maximum temperatures of both the AMB and SSB boards are further decreased by about 1 °C. A more clear view of the outcome of the tests can be obtained by looking at the Figure 4.20, presenting the histograms of the temperatures for all the LAMB measurements performed using pseudodata of BF 7 in the standard configuration test (Figure 4.20a), in the increased water flux test (Figure 4.20b), and in the reduced water temperature test (Figure 4.20c). The histograms show a mean reduction of the boards temperature of about 2 °C for the higher water flux test and a reduction of about 3 °C for the chiller case, compared to the standard cooling configuration.

Since the use of the chiller drastically increases the risk of dew inside the racks the chiller cooling configuration has been discarded. On the other hand, since the high water flux configuration doesn't present relevant cons, it has been choose as preferred cooling configuration for the FTK final setup.

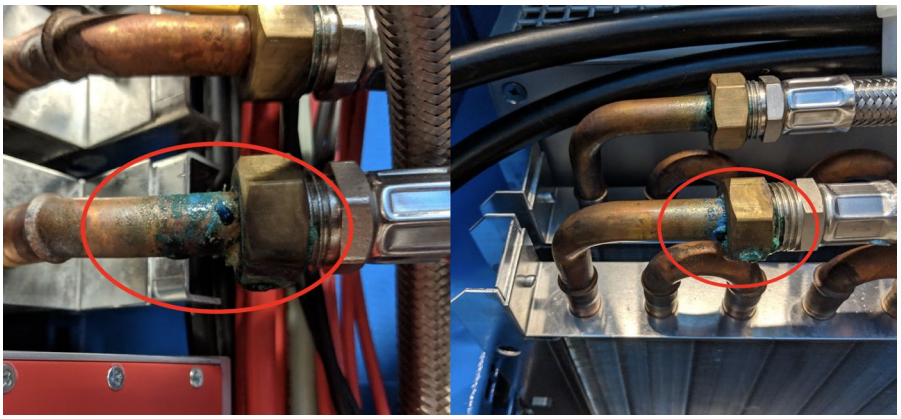


Figure 4.17: The Figure shows an example of the formation of dew happened inside the testing FTK rack during the execution of the chiller configuration cooling test. The pipes visible in the figure are the ones bringing the water in input to the heat exchangers.

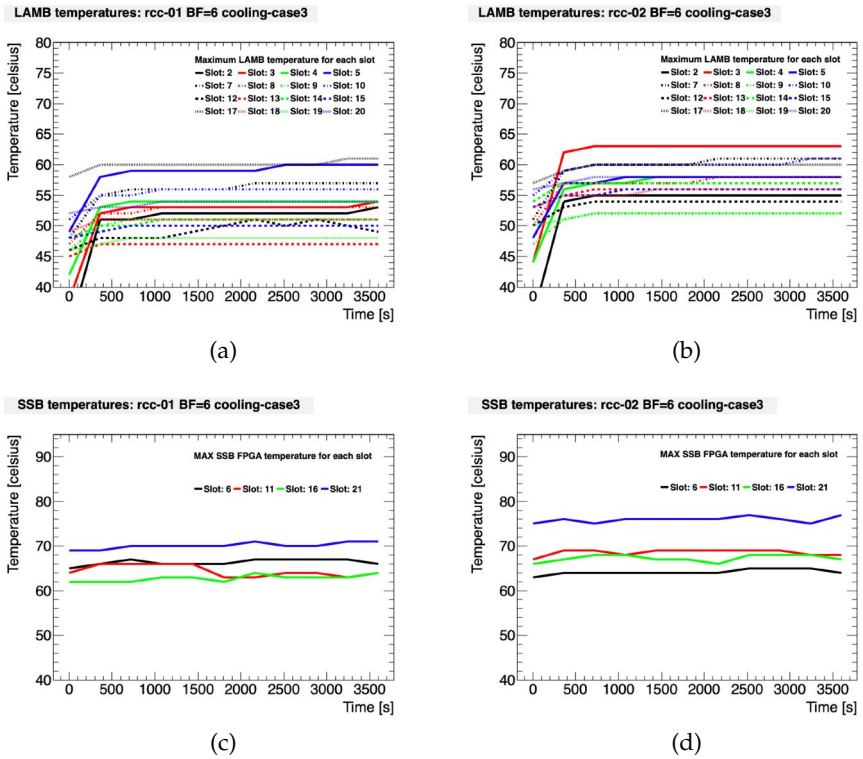


Figure 4.18: Profile of the maximum LAMB temperatures (a),(b) and of the maximum SSB FPGA temperatures (c),(d) for the BF 6 AMB configuration. The test was executed after enabling the water chiller.

4.4.4 *Cooling tests in reduced board multiplicity configuration*

As mentioned before, a fall back solution for the reduction of the board temperatures was the reduction of the number of boards housed inside

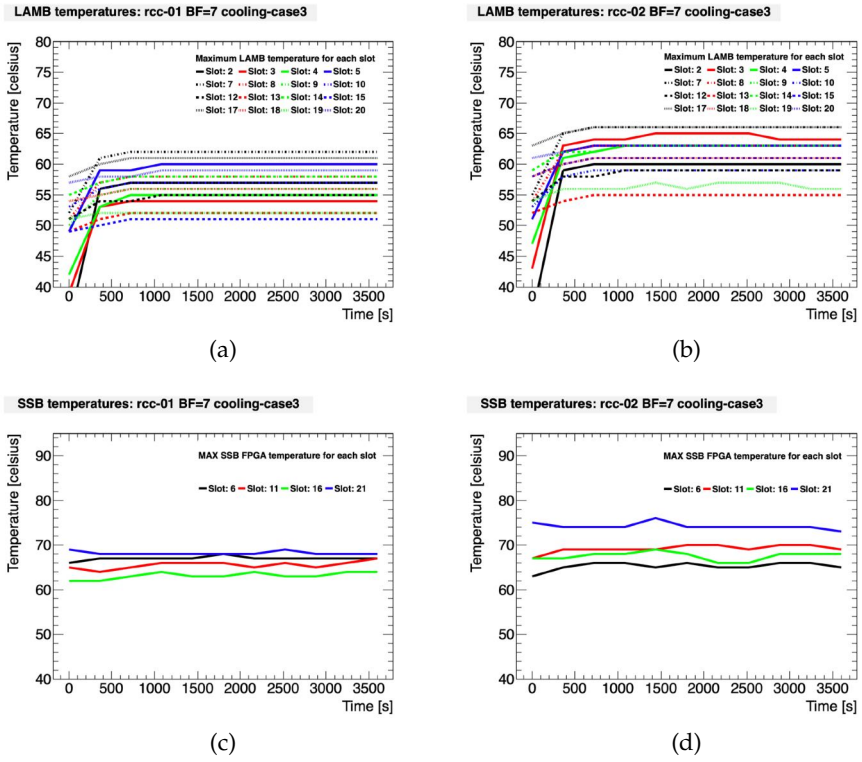


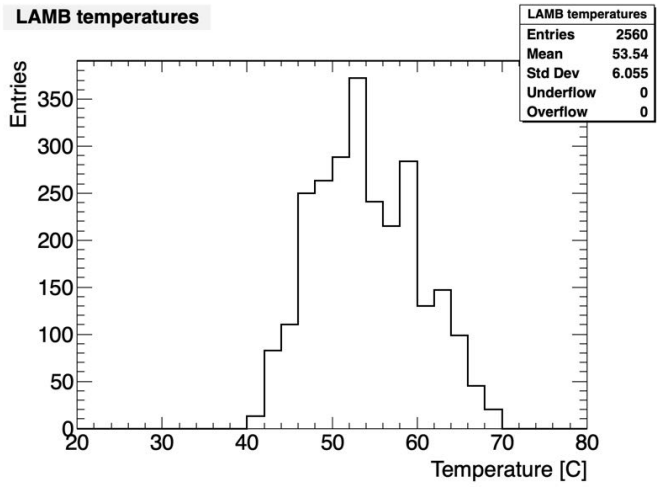
Figure 4.19: Profile of the maximum LAMB temperatures (a),(b) and of the maximum SSB FPGA temperatures (c),(d) for the BF 7 AMB configuration. The test was executed after enabling the water chiller.

each VME rack. This solution would require the purchase and the full equipment of two additional racks. Each crate would require to house 12 PUs and 3 SSBs (instead of 16/4 PUs/SSBs), reducing the total power to be dissipated by the cooling system. In order to study the gain obtainable with this setup, cooling tests in which the hotter 4 PU slots and the hottest SSB slot were removed have been performed. In Figure 4.21, the results of these tests are presented. Only the test in which AMB pseudodata of BF 7 are shown. As can be seen from a comparison with Figure 4.14, the maximum temperature of the boards are drastically reduced. However, this reduction is mostly due to the removal of the boards from the hotter slots. By comparing the temperatures slot by slot, it is possible to note that temperatures are only slightly reduced.

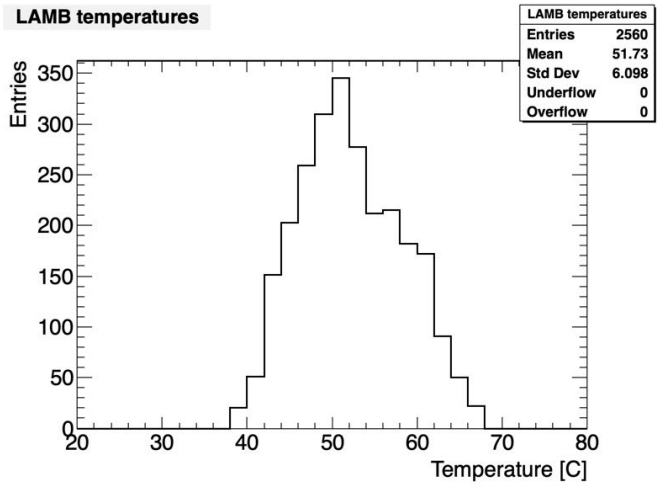
Since the results from the complete crates configuration are good, this fall-back solution has been considered not required.

4.4.5 *Cooling tests in second rack*

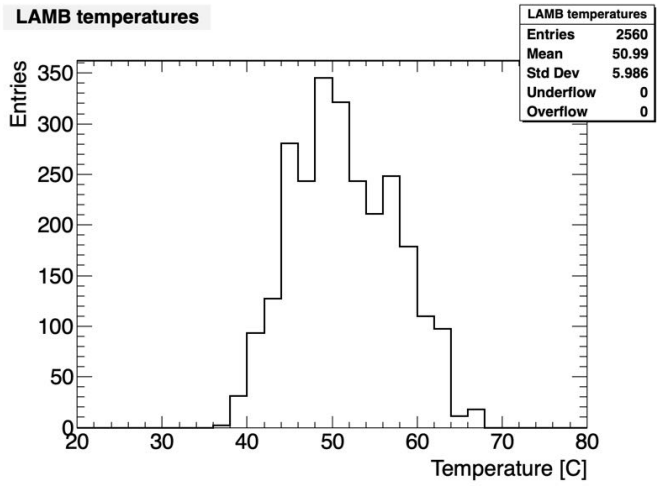
Since all the tests presented previously were executed on a single rack, another test able to study the possible variations of the boards tem-



(a)



(b)



(c)

Figure 4.20: Distribution of the LAMB temperatures for the standard cooling configuration test (a), the increased water flux case (b), and the chiller cooling case (c). All the tests were performed using the BF 7 AMB pseudodata.

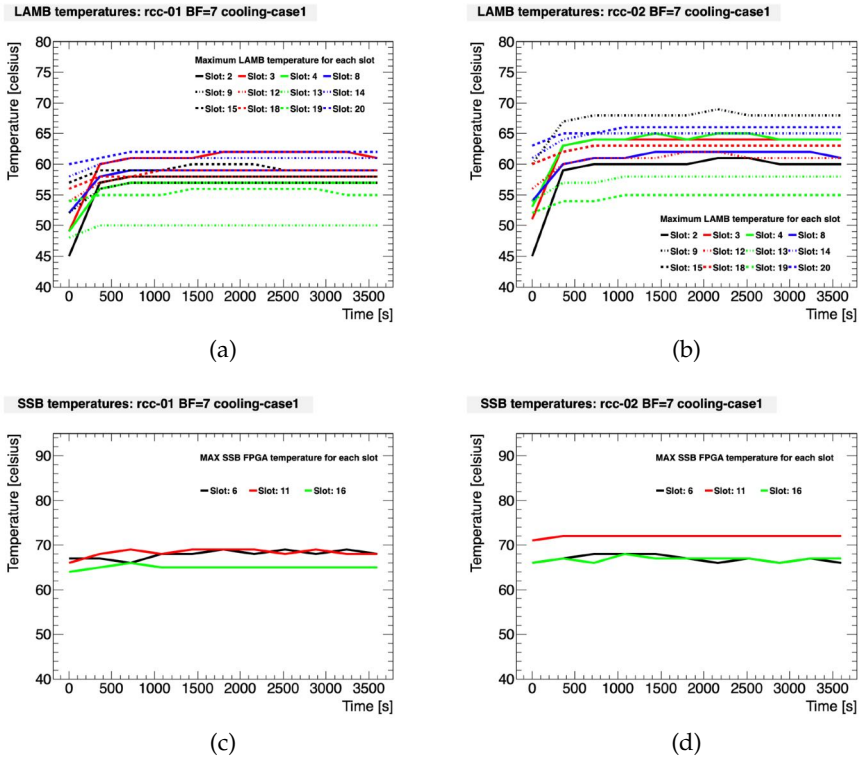


Figure 4.21: Profile of the maximum LAMB temperatures (a),(b) and of the maximum SSB FPGA temperatures (c),(d) for the BF 7 AMB configuration. In this tests only 12 PUs and 3 SSBs where housed in each VME crate.

peratures among the racks was required. Due to the lack of available boards, the test was performed with only 8 PUs and one SSB per crate, as shown in Figure 4.22. In order to have coherent results, the same test was performed on two different racks: the standard cooling test rack and the new FTK rack 7-9.

Figure 4.23 shows the distribution of the LAMB temperatures for the new rack 7-9 (Figure 4.23a) and for the standard cooling test rack (Figure 4.23b). The measured mean LAMB temperature of the new rack is 2 °C higher than the one of the standard rack. In order to be conservative, a variation of 5 °C has been considered as rack to rack variation for the final analysis of the cooling test results.

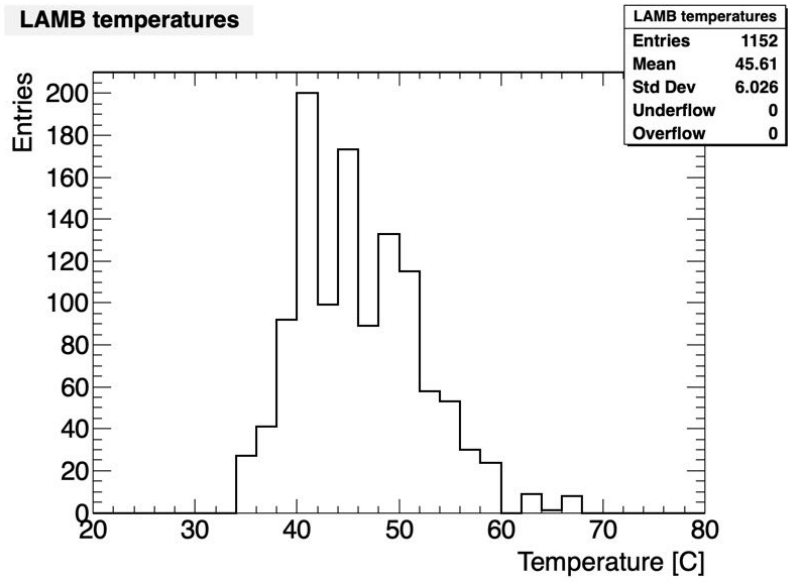
4.4.6 *Fan speed optimization study*

A parameter that can play a role in the cooling of the boards is the speed at which the fans run. Varying the fan speed can cause differences in the air flux inside the racks, and it can increase/decrease turbolences. Moreover, reducing the speed of the fans can increase their lifetime. Tests, changing each time the settings of the fan speeds, have been performed in order to study the best fan speed configuration.

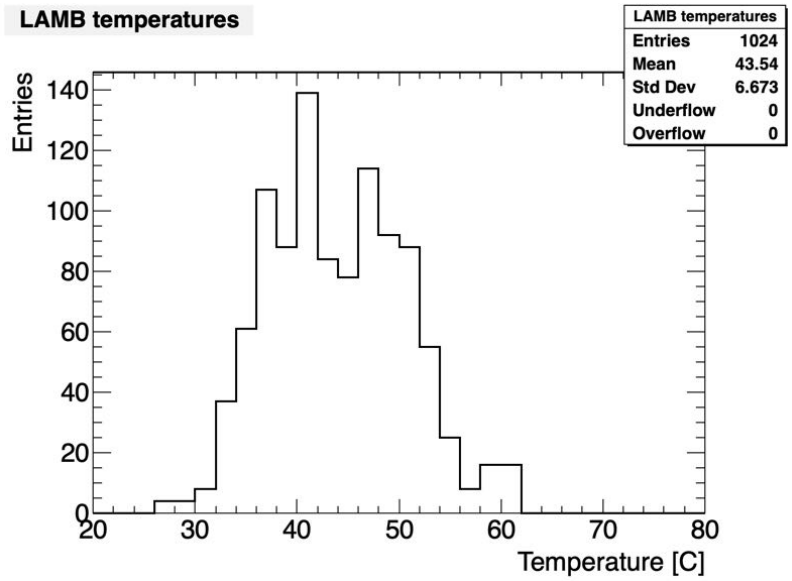
Figure 4.22: Picture of one of the VME crate used for the reduced configuration cooling tests. The crate is load with 8 PUs, one SSB (slot 21, black board), and the Single Board Computer. In order to avoid the dispersion of the air flow from the missing slots, blinds have been used (metallic covers on the slots where the boards are missing).

The tests were performed in the cooling rack, using 16 PUs and 4 SSBs in each crate and using the speed configurations (top-bottom crate fans, same configuration for crate fan couples) 100 %-100 %, 80 %-100 %, 60 %-100 %, 100 %-80 %, and 100 %-60 %. The tests were also repeated for different AMB power consumptions levels. Only the results showing the LAMB temperatures for the BF 6 AMB pseudodata configuration will be presented. The tests not presented shown similar results.

In Figure 4.24 and Figure 4.25, the results of the BF 6 tests are presented. In all the tests performed, the lowest temperature was measured in the fan speed configuration 60 %-100 %. Running the bottom fan tray at a reduce speed with respect to the top one, seems to helps to avoid the creation of turbolences and to increase the uniformity of the air flux in the crate volume. Moreover, running the bottom fans at 60 % of their maximum speed can significantly increase their lifetime. For this reason, the final fan speed of the FTK lower fans will be fixed to 60 %. The fan speed of the top fans, instead, will be adjusted automatically by the DCS (as in the past), moving from the lower 60 % level to the maximum 100 % level as function of the temperatures read.



(a)

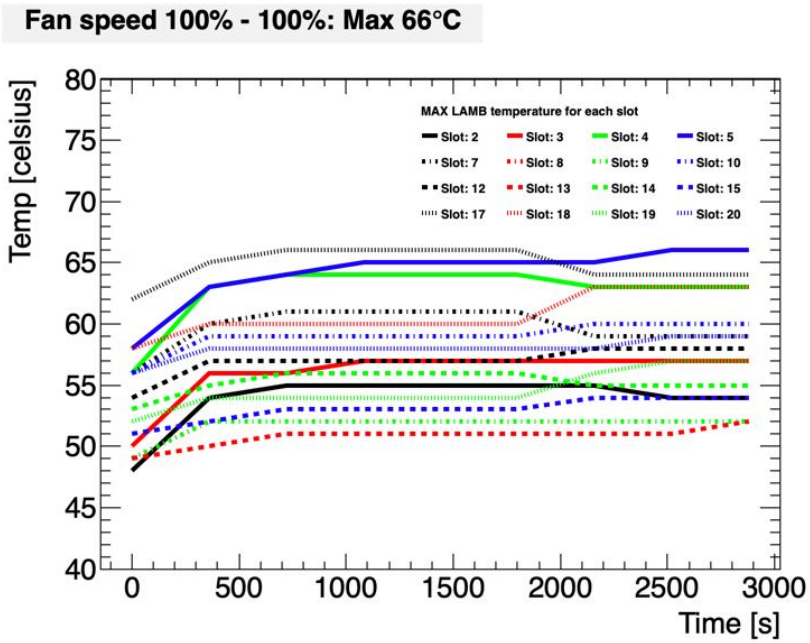


(b)

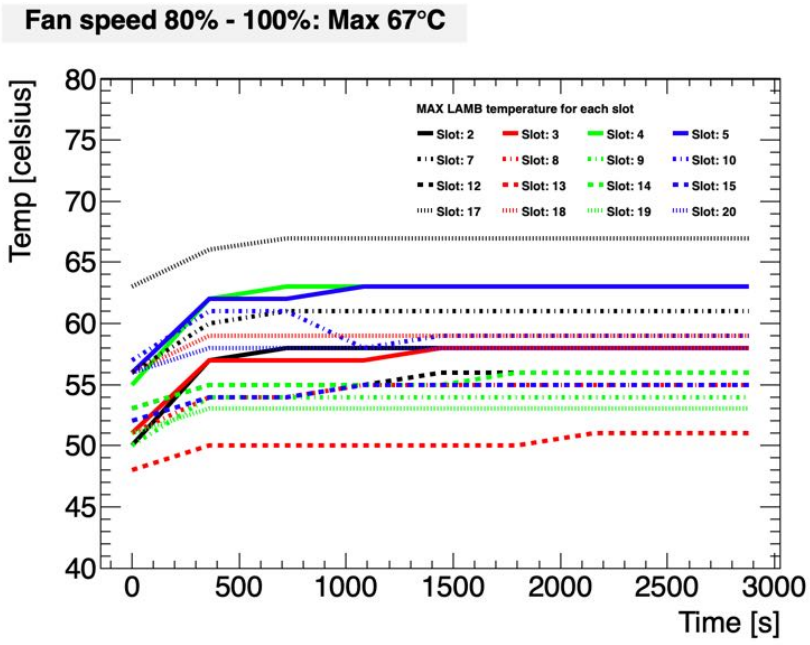
Figure 4.23: Distribution of the LAMB temperatures for the AMB BF 6 pseudodata configuration for the new rack 7-9 (a) and the standard cooling rack (b). The VME crates on each rack were equipped with 8 PUs and 1 SSB. A variation of 2 °C has been observed between the two racks.

4.4.7 *Temperature threshold for SW safety system*

As presented in Section 4.2, a SW safety system has been put in place, able to shutdown automatically the rack PS in case the temperature of the air inside the rack would overcome a given threshold. In the absence of real tests, that threshold was initially set to 32 °C. In order

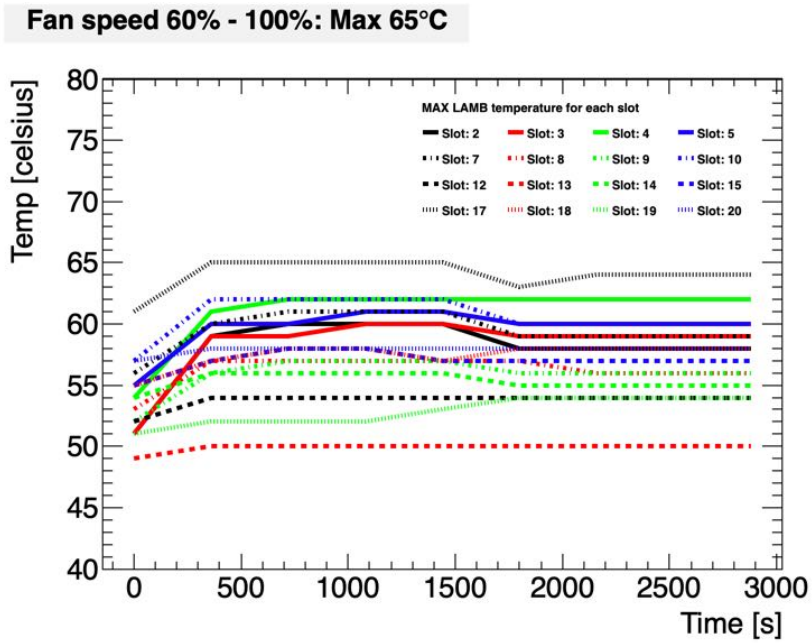


(a)

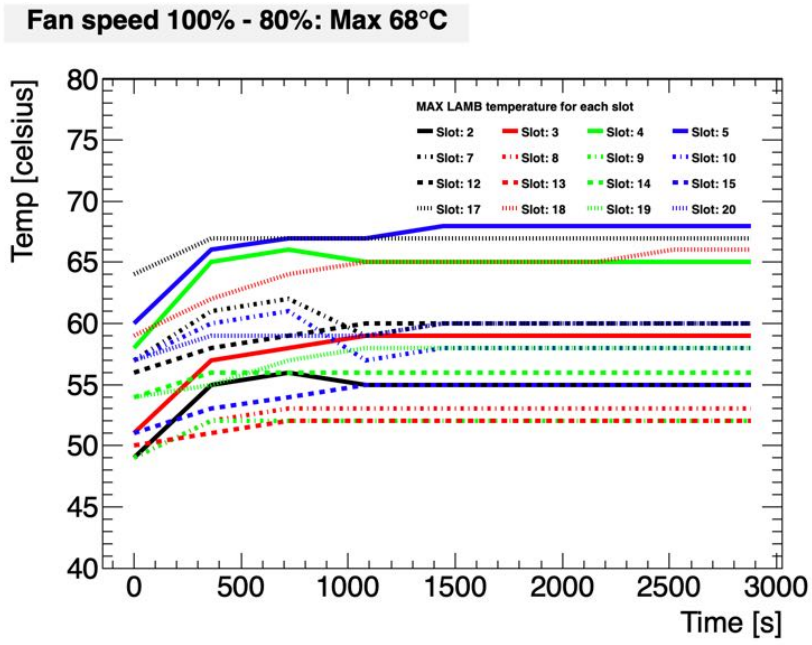


(b)

Figure 4.24: Profile of the maximum LAMB temperatures for the BF 6 AMB pseudodata, in a fan speed configuration of 100 %-100 % (a) and 80 %-100 % (b).



(a)



(b)

Figure 4.25: Profile of the maximum LAMB temperatures for the BF 6 AMB pseudodata, in a fan speed configuration of 60 %-100 % (a) and 100 %-80 % (b).

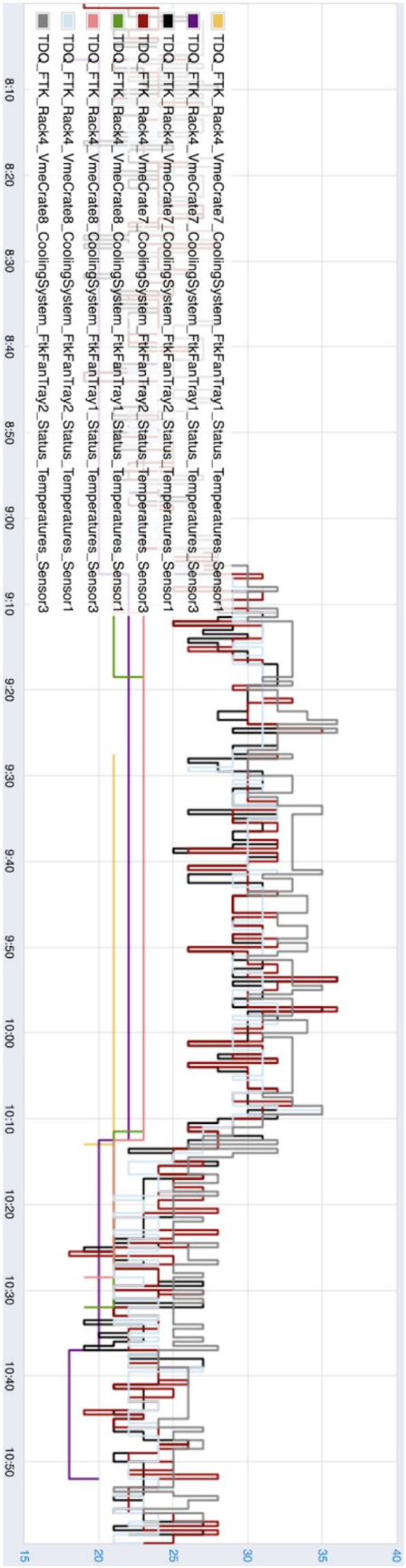


Figure 4.26: Profile of the air temperature measured inside the fan trays by the DCS system. The measurements were performed running 16 AMBs with BF 7 pseudodata. The maximum measured temperature was 36 °C.

BF conf	LAMB power (W)	Max LAMB temp (°C)	Max SSB temp (°C)
6	33	62	78
6.5	36	65	78
7	38	67	79

Table 4.4: Table summarizing the results obtained during the high water flux configuration cooling tests. This cooling condition is the chosen one for the final FTK configuration.

to find a more precise value for this threshold, and to avoid useless shutdown of the FTK racks (that would be obviously problematic during data taking) a study on the rack air temperature has been executed.

For these tests, the fully loaded FTK cooling test rack has been used. Pseudodata with different BF values were used, checking in the meantime for the temperature of the air inside the rack. Figure 4.26 shows the temperature profile of the rack air temperature measured inside some of the fan trays, obtained running the AMBs on the worst case scenario BF 7 pseudodata. The maximum value of the air temperature obtained in this test was of 36 °C. The maximum corresponding temperature for the LAMBs was of 70 °C. Since the test running condition is expected to be higher than the maximum condition expected for the Run3 data taking session, a proposed final threshold for the SW safety system was of 38 °C. This would leave some margin over the maximum air temperature measured, still keeping the boards in a safe working condition.

4.4.8 *Cooling studies summary*

The cooling studies presented in this section end up in a complete characterization of the FTK VME cooling system. The main outcome of the studies is that the FTK cooling infrastructure has proven to be able to keep all the boards temperatures under the operational limits, in all the expected FTK working conditions.

In Table 4.4, the maximum temperatures measured in the high water flux cooling condition (the one identified for the use in the FTK data taking session) are summarized.

Allowing for up to 5 °C rack to rack variations, the maximum AMB temperature expected for the worst running condition scenario (FTK 100 kHz at $\mu = 80$, corresponding to pseudodata with BF 6.5) is of 70 °C, well under the 80 °C AM chip temperature limit. This running condition correspond to a LAMB power consumption of about 33 *W*. Considering the descoping of FTK to a 35 kHz processing rate, with an expected LAMB power consumption of 28 *W*, the margin over the temperature threshold is even bigger.

The SSB maximum temperature measured was of 83 °C (allowing also in this case a 5 °C rack to rack variation margin). This temperature refers to only one crate slot, that seems problematic for the cooling air to be reached. Excluding this slot, the higher temperatures is reduced to 74 °C. The V-4 SSBs, that present a lower operational temperature limit, will not be placed in this slot.

The AUX cards, as expected, don't present any concern about the temperatures they reach.

Beside the very good results obtained, some optimizations have been identified. The fans speed has been optimize to a 60 %-100 % configuration, able to improve the air flow inside the crates and to increase the fans lifetime. The SW safety air temperature threshold has also been optimized, moving to a more operational safe value of 38 °C.

Other minor optimizations, that will be put in place before the end of the Long Shutdown are foreseen. During the tests, some portions of the racks not properly seal have been identified. The air was able to escape from the rack, introducing turbolences in the rack air flow. These "holes" will be properly sealed using plexiglass panels. Moreover, the use of 6 heat exchangers, able to dissipate up to about 22 kW, is probably excessive. Removing one or two of them could lead to a better air distribution, while not interfering too much with the board temperatures. Further studies concerning these aspects will be executed in the next months.

5

FTK ONLINE SOFTWARE

The FTK system, presented in Chapter 3, is a very complex system. It is composed of hundreds of electronic boards and thousands of links, connecting the FTK boards and FTK system itself with the other parts of the experiment. Configuring and monitoring such a system is very challenging. Each board has to load a different set of configurations (defined by the portion of detector data it is intended to process), many links has to be set up and verified (many operations are required, and have to be executed in the proper order), and the system has to be constantly monitored for faults and for the quality of the data produced. Moreover, some of the system configuration may fail and have to be repeated, or the system can get stuck while processing data (because of data errors and/or FW bugs). The control system has, therefore, to be reliable and able to survive and recover for unexpected problems. Finally, FTK has to be integrated in the ATLAS data taking framework, as all the ATLAS subsystems have to.

The FTK online software (SW) has to cope with all these duties. It has to provide a stable and reliable environment in which the FTK boards can be programmed and configured, can start and stop their processing, and in which every part of the system is monitored and is integrated in the ATLAS framework.

During my PhD, the FTK online SW development has been one of my primary activities. I directly contributed in the development and optimization of all the different aspects of the project: from the development of the core SW, the development of the configuration and monitoring tools, to the coordination of the board specific development tasks.

We initially developed the FTK online SW making use of the existing tools available from the common ATLAS framework, adapting them to the FTK use-case or, sometimes, making a completely new use of them. This choice lead us to provide a stable online software environment since the beginning of Run2, easing the commissioning of the system since its first stage, while working on the identification of bottlenecks and criticalities. The gain in knowledge on the system peculiarities and requirements, and the identification of its limitations, guide us today in its optimization.

In this chapter the FTK online software will be described in some details, focusing on the problems and requirements specific of this project and on the solutions that we developed (or are developing). In the next subsection, a brief general presentation of the components and duties of the common ATLAS online software framework will be

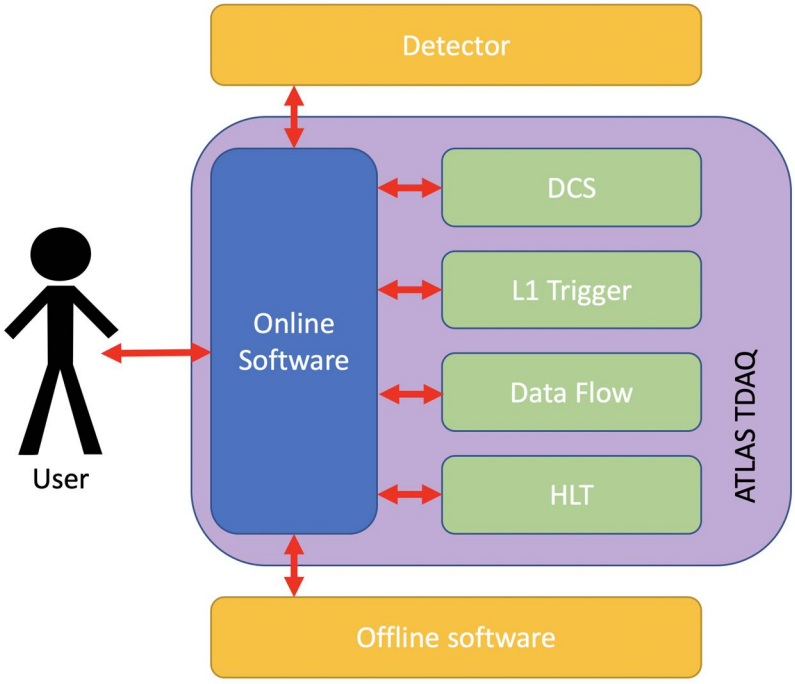


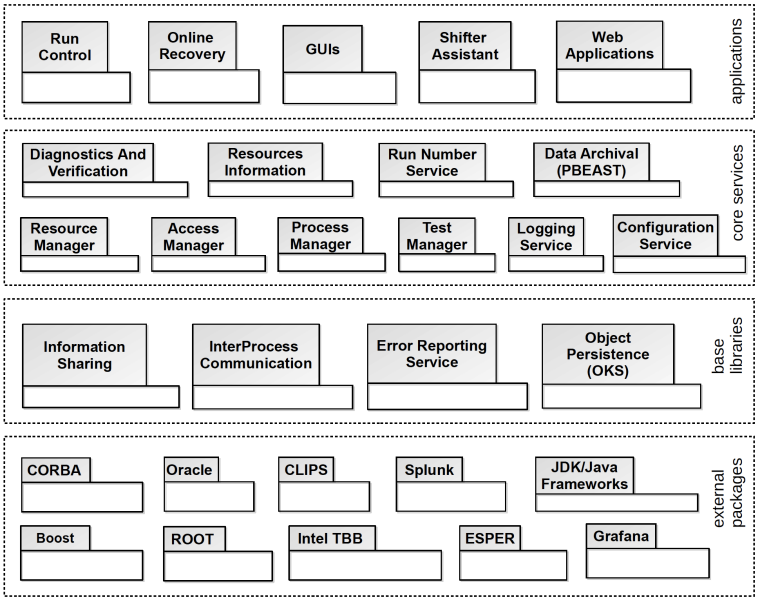
Figure 5.1: Context diagram of the ATLAS online software.

presented. This will lead us to introduce how the integration of FTK in this framework has been performed. In the rest of the chapter will be treated in some details all the problems we encountered and the FTK specific solutions we adopted, for all the various components of the online software.

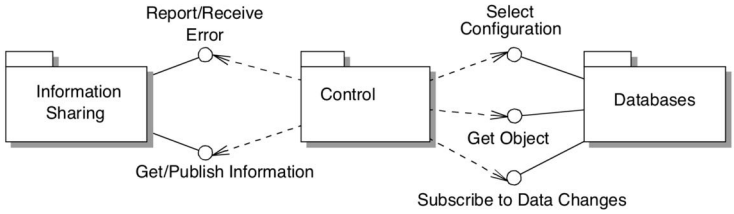
5.1 ATLAS ONLINE SOFTWARE FRAMEWORK

The ATLAS Online Software encompasses the software to configure, control, and monitor the TDAQ system, with the only exclusion of the processing and transportation of physics data [64]. It is a customizable framework which provides essentially the 'glue' that holds the various sub-systems together. It does not contain any elements that are detector specific as it is used by all the various configurations of the TDAQ and detector instrumentation. It co-operates with the other sub-systems and interfaces to the Detector Control System, the LVL1 Trigger, the DataFlow, the High Level Trigger processor farms, the Offline Software, and to the human users. A schematic representation of the ATLAS online SW context is presented in Figure 5.1.

The ATLAS online software ranges from high level applications to low level packages and it is designed following a layered component model, as shown in Figure 5.2a. At the very bottom are common libraries and packages to deal, for instance, with threads, in house developed object persistency system and the libraries for the CORBA based inter process communication [65]. Higher up are a set of services,



(a)



(b)

Figure 5.2: (a) Control and Configuration software: high-level architecture. (b) Internal interaction between the Online Software packages.

like the configuration service or the process management. Above these layers is the so-called application layer, with the run control, the online recovery system, and a set of graphical user interfaces (GUIs) allowing the operator to act on the system.

The online software architecture is based on a component model and consists of three high-level components, called packages. Each of these packages is responsible for a clearly defined functional aspect of the whole system, containing all the sub-packages required. The first component is Control, which provides services to marshal the TDAQ and ATLAS subsystems through their start-up and shutdown procedures, so that they are performed in an orderly manner. It is responsible for the synchronization of the states of a run in the entire ATLAS system and for process supervision. In order to handle the control of complex hierarchical systems, the online SW, through the Run Control (RC) application, provides a Finite State Machine (FSM) which controls the subsystems firmware and software operations, distributing the commands to and synchronizing operations between the different parts of the detector.

The Databases package contains sub-packages for the configuration of the TDAQ system and detectors. Configuration database services are provided for holding the large number of parameters and configurations which describe the system topology, hardware components, software components, and running modes. It is responsible to support system configuration description and access to it, record operational information during a run and access to this information. Boundary classes provide read/write access to the conditions storage.

Finally, the Information Sharing package contains classes to report error messages, to publish states and statistics, to distribute histograms built by the sub-systems of the TDAQ system and detectors, and to distribute events sampled from different parts of the experiment's data flow chain. During data taking, access is provided to monitor information like statistics data, sampled data fragments to be analysed by monitoring tasks, histograms produced in the TDAQ system, and also to the errors and diagnostic messages sent by different applications.

The interaction between the online software packages is shown in Figure 5.2b.

5.2 FTK ONLINE SOFTWARE FRAMEWORK

The FTK online SW is an extension of the TDAQ online SW, meant for the integration of the FTK system in the common ATLAS framework, described in the previous section. It is used to define the SW objects representing the FTK physical components, to manage their configuration and operation, and to monitor the system.

We decided to structure the FTK online SW in 12 distinct packages, as presented in Figure 5.3. Four of these packages constitute the FTK common software framework, directly developed and maintained by the FTK online SW team. The remaining packages define the board specific software, which is developed and maintained by the specific board developers (under the online SW team support and direction).

The *ftkcommon* package is the base package defining the FTK SW framework. This package contains the definitions of the classes directly inheriting from the ATLAS framework (over which the different board objects are based), and the objects directly linked to the TDAQ code. Moreover, the configuration objects common to more than one board, as the *PatternBank* (container class for the patterns to be loaded on the AM chips) and *Constant* (container class used to store the 8 and 12 layer fitting constants) C++ objects, are defined in this package. It also define all the logging macros, the FTK specific error handling functions, and, in general, all the low level objects and functions used to manage the integration of FTK in the ATLAS framework.

The two *ftkvme* and *ftkdqm* packages are used to define the VME access functions and the base monitoring objects, respectively.

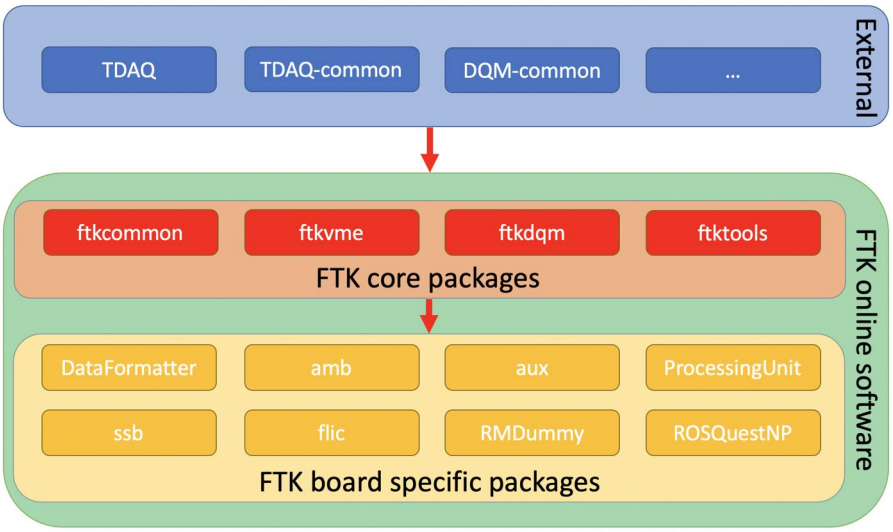


Figure 5.3: Sketch of the FTK online SW structure. The FTK online SW is composed of 12 different packages: 4 of them form the core SW, while the others are FTK HW components specific.

Finally, the *ftktool* package contains all the common tools and applications required from the project, as the ftk_maker (presented in Section 5.4) and FTK_Solo (presented in Section 5.6.3) tools or the recovery machinery (described in Section 5.5.5).

The other packages contain the board specific SW, defining the classes and methods used to configure, run and monitor the given FTK board.

The *DataFormatter* package contains the code related to the IM and DF boards. The communication protocols of these boards are also defined in this package. The *aux*, *amb* and *ProcessingUnit* packages take care of the PU boards couple, with the two aux and ambslp packages defining the respective low level board libraries and tools, while the ProcessingUnit package defines the common controller class. The *ssb* and *flic* packages collect the code for the SSB and FLIC boards, respectively. Finally, the *ROSQuestNP* package is used to manage the integration of the QuestNP boards in the FTK SW framework, while the *RMDummy* package defines a "fake board" emulator used for testing and developing purposes.

Since the FTK SW is based on the common ATLAS framework, the accessory tools (as the building management tools and the versioning tools) have to be the ones used by the TDAQ online SW.

The building of the FTK SW releases is managed by means of the CMake package [66]. CMake is an open-source, cross-platform family of tools designed to build, test and package software. CMake is used to control the software compilation process using simple platform and compiler independent configuration files, and generate native makefiles and workspaces that can be used in the compiler environment. Initially, the FTK software building was based on the

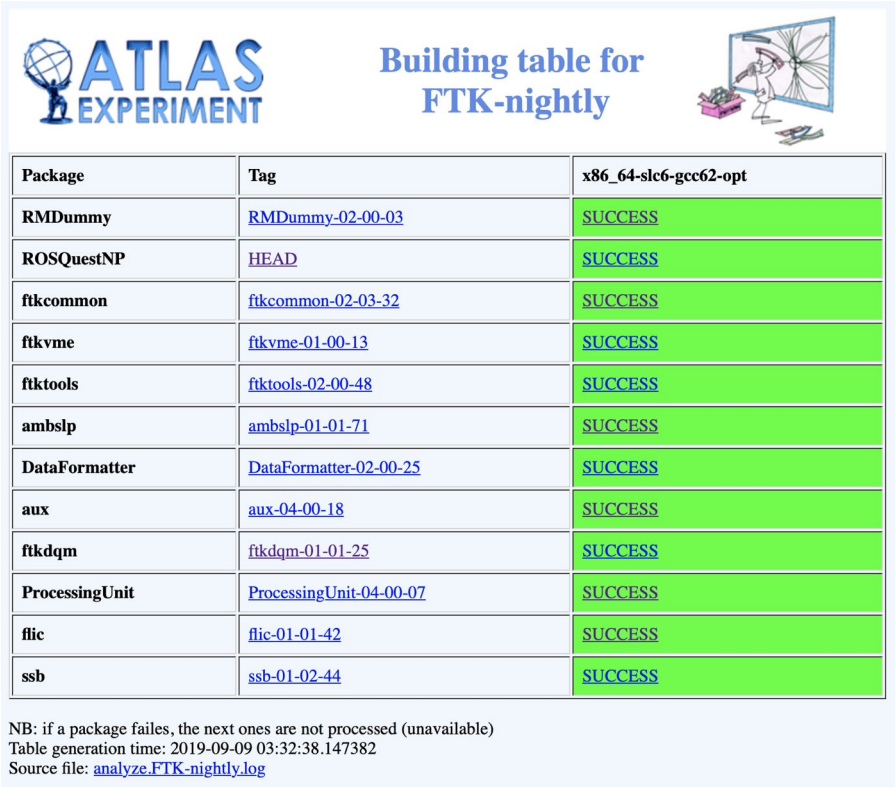
Configuration Management Tool (CMT) package. The migration to CMake followed the one of the TDAQ online SW. FTK completed its migration at the beginning of 2017.

The versioning software has been migrated to a new platform in recent times. Subversion (SVN) was the standard versioning software for ATLAS until 2018. Following the migration of the whole ATLAS SW to GitLab [67], also the FTK code has been migrated.

The FTK code relies its deployment on software releases. Bash and Python scripts for the automation of the release building process have been developed (and underwent major updates concurrently to the CMake and GitLab migrations described above). We decided to rely the collection of the SW to be used for a new release on the collection of tags, made through the versioning software. Each package manager provides a candidate version of its package code, which is added to the release tag collector and tested by the FTK expert on-call. Tests able to validate the new software are performed in Lab4 on a FTK special slice, using a nightly release built with the new release tags. When a candidate release has been validated, it is deployed at P1 and is used for commissioning, replacing the previous release. During 2018, 11 online SW releases were deployed for FTK, 4 of which were major releases (containing substantial changes to the FTK online SW framework), and many patches were installed. This big number of releases allows to note the vivacity of the online SW development for the FTK project.

In order to ease the SW testing, we decided to make available also different nightly releases. The code used to build them is, again, based on the tag collection method. Three different nightlies are compiled every night via a cronjob, using the same automatic scripts used for building the official releases. The FTK-nightly release is a nightly build of the SW tagged in a special tag collector. As mentioned above, it is used for the validation of the code before an official release is built. Moreover, it is used for the validation of the patch code, before its installation, or by the board developers to test their changes running the code on real boards. The FTK-nightly-head is a nightly release build on the master of each FTK package. This release is mainly used to check that no building errors have been introduced in the master during daily development. Finally, FTK-nightly-dev is a special nightly release build on specific branches of the FTK packages. It is mainly used to test the code during major changes to the FTK core SW, or when changes undergoing development in branches need to be tested before the merging. All these nightly releases are available only in Lab4.

Notification emails are automatically sent to the code developers summarizing the results of each nightly and release building. A script, able to analyze the results of each build, has been developed in python and it is integrated in the release building scripts. It is able to send



ATLAS EXPERIMENT

Building table for FTK-nightly

Package	Tag	x86_64-slc6-gcc62-opt
RMDummy	RMDummy-02-00-03	SUCCESS
ROSQuestNP	HEAD	SUCCESS
ftkcommon	ftkcommon-02-03-32	SUCCESS
ftkvme	ftkvme-01-00-13	SUCCESS
ftktools	ftktools-02-00-48	SUCCESS
ambslp	ambslp-01-01-71	SUCCESS
DataFormatter	DataFormatter-02-00-25	SUCCESS
aux	aux-04-00-18	SUCCESS
ftkdqm	ftkdqm-01-01-25	SUCCESS
ProcessingUnit	ProcessingUnit-04-00-07	SUCCESS
flic	flic-01-01-42	SUCCESS
ssb	ssb-01-02-44	SUCCESS

NB: if a package failes, the next ones are not processed (unavailable)
Table generation time: 2019-09-09 03:32:38.147382
Source file: analyze.FTK-nightly.log

Figure 5.4: Example of a FTK-nightly building table web page. A new building table is automatically generated for every release/nightly build. The building table summarizes the release name, the packages contained, the tags used and a status report on the specific building result. Both the tags and the building status are links to the SW repository and log files, respectively.

individual notifications to the manager of each package that failed the building process, pointing to the logs to ease the debugging. Building tables are also automatically produced, summarizing the status of each package, the tags used for the given build, and links to all the logs. The building tables are accessible via a web page. In Figure 5.4, an example of FTK-nightly building table is shown.

The development of a continuous integration procedure for the FTK code, based on the GitLab pipelines and able to perform a complete building of the master code at every commit, is also planned. This is expected to further speed up the debugging of the code. Its deployment is foreseen for the last months of the 2019.

The migration of the building architecture of the FTK code to a new operative system, that is undergoing deployment in the ATLAS nodes this year, is still missing. ATLAS is currently moving from the scientific linux 6 OS to the more recent CENTOS7 OS. The FTK plan is to perform the migration to the new OS by the end of the 2019.

Another open point for the FTK SW environment is the disposal of the Andrew File System (AFS). The FTK SW releases, currently

installed in AFS, will be moved to the CernVM File System (CVMFS). A defined date for the disposal of AFS is not yet available, but the FTK release migration to CVMFS is expected to be completed before the beginning of 2020.

No other changes are expected to the FTK and, in general, to the ATLAS online SW environment until the end of Run3.

5.3 FTK CONTROL SYSTEM

The ATLAS Control software is responsible of performing the initialization and shutdown of the ATLAS firmware and software, of dispatching commands to TDAQ and subdetectors elements and of synchronizing operations between them, as well as performing monitoring and error handling. In order to handle the control of complex systems, the control software is based on a FSM, in which the FTK system needs to fit its operations. In this section, the FTK control software will be described in some details.

The configuration database of the FTK system, i.e. the database describing which boards have to be controlled, the parameters and the different constants and patternbanks that have to be used as well as the link mapping, will be instead described in Section 5.4.

The ATLAS Control SW is based on different levels of controller applications, forming a well defined hierarchical structure. The FSM commands, by instance, are not directly sent to all the different applications, but they are propagated through the controller layers following the application hierarchy, reaching at the end all the leaf applications meant for the control of the hardware. In order to minimize the development of code in the first part of the commissioning (when the specific needs of the system were still not clear), an available TDAQ application, called *Rod Crate Daq* (RCD) [68], has been reused as leaf application for FTK.

The RCD provides a complete framework to implement data acquisition functionality at the boundary between the detector specific electronics and the common part of the data acquisition system. Based on a plugin mechanism, it allows selecting and using common services (like data output and data monitoring channels) and developing libraries to control, monitor, acquire data and/or emulate detector specific electronics. Providing also event building functionality, the RCD has been developed as the main data acquisition tool for the first phase of detector commissioning, becoming the natural choice to be adopted for the FTK boards control application.

The plugin modules of the RCD application, describing the hardware or software components, are controlled through methods corresponding to the TDAQ FSM transitions. These plugin objects are called *ReadoutModules* (RM). A different RM class has been developed

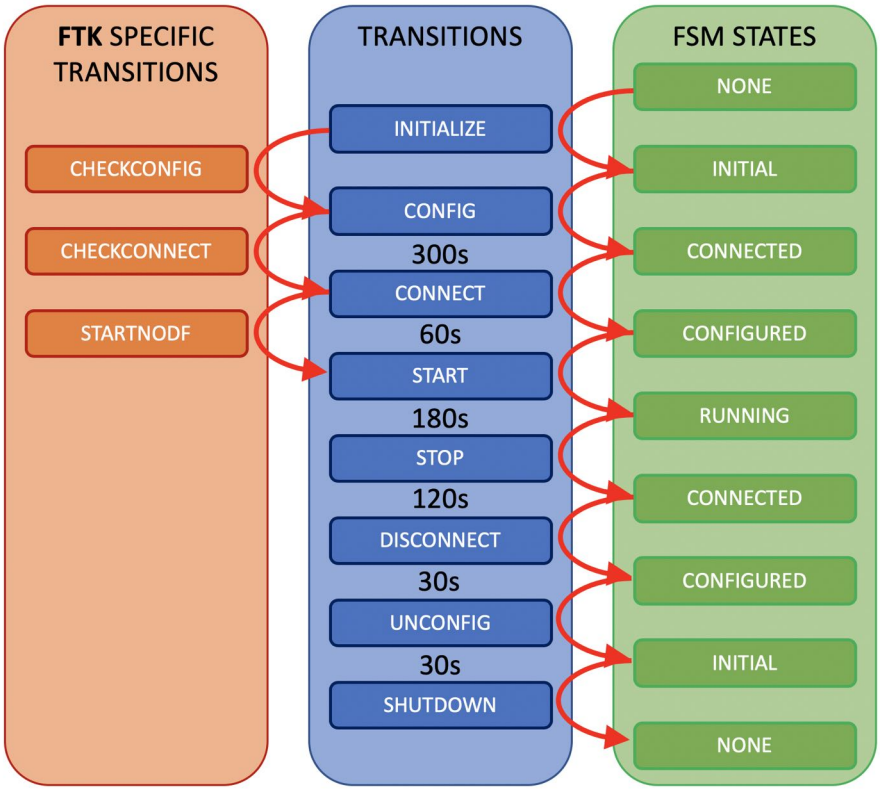


Figure 5.5: Sketch representing the FSM states and transitions available for ATLAS and all its subsystems. The average ATLAS transition times are shown under each of the transitions. The FTK specific sub-transitions are also shown. These sub-transitions are executed by the FTK subsystem only, and are executed before the given common transition, to which they are associated to, takes place. More details about the FTK specific transitions will be given in Subsection 5.5

for each of the FTK boards, each of them inheriting from an FTK base class. These plugins are loaded by the RCD application at run time. They are loaded dynamically, and do not need to be linked to the main application.

One RCD application is used to control all the components of each crate/shelf, with the application running on its controller machine. Each board is defined in the RCD by one RM (excluding the PU case, in which the PU RM represents the AUX-AMB couple). The number and types of the RMs that an RCD has to contain are defined in the configuration database, together with the values of its parameters. The RM of each board type defines all the methods required for the configuration of the specific board type it is associated with. The principal methods are the ones corresponding to the FSM transitions, defined as virtual in the RM C++ base class (which manages the interactions with the TDAQ infrastructure) and overriden in the FTK derived classes to fit the project use cases.

In Figure 5.5, the ATLAS FSM states and transitions, made available for the configuration and shutdown of all the ATLAS subsystems, are shown.

The *INITIALIZE* and *SHUTDOWN* transitions are only used to setup/stop the SW infrastructure. The FTK RCD applications, by instance, are started during initialize and killed during shutdown.

The RM plugins are loaded at the beginning of the *CONFIG* transition. The RCD application accesses the configuration database checking for the enabled plugins, and the proper RM constructors are called. The RM setup() function is called, with each of the RMs accessing themself the database in order to identify the configurations associated to the boards they represent. The following method called is config(), which starts the configuration of the boards. In this method, checks on the configurations already loaded on the boards are performed. Due to the large amount of time required for the data loading on some of the FTK boards (in particular, the VME boards), this step can only occur in special "non data taking" runs. In a normal run, the boards are expected to have the correct versions of patternbanks/constants already loaded, and only checks are performed. More details on the configuration loading step will be given in section 5.5.1. Other than the configuration loading, further operations are performed, depending on the board type. In general, all the operations required to configure the boards have to fit inside the *CONFIG* transition, through the config() RM method, and the *CONNECT* one that follows, via the connect() method. During these two transitions the RCD application, and in particular the RM, communicates with the boards, accessing the FPGAs registers to prepare them for processing. Different communication protocols are used. The VME boards use the VME protocol, while the ATCA ones use IPBus [69]. Other protocols are used for the inter board communication, e.g. the dispatching of the commands from the DF to its mezzanine IM boards is based on the I^2C protocol [70]. Moreover, all the links between the different FTK boards are setup, together with the links connecting FTK to the other subsystems.

The *START* transition is the last before the beginning of the data processing. This transition correspond to the prepareForRun() method of the board RMs. During this transition the configuration of the boards has to be completed, the links have to be setup, and the board have to be ready for processing, waiting for the data that will start to arrive when the running state will be reached.

The other transition methods are required to stop the boards. The stop() method, called to execute the *STOP* transition, is used to block the processing of the FPGAs. A last call to the monitoring functions, that will be described later, is also forced at the beginning of this transition, in order to collect a final snapshot of the system. The disconnect() function, associated to the *DISCONNECT* transition, is mainly used to shut down the boards links. Finally, the unconfig()

method, associated to the *UNCONFIG* transition, is used to bring the boards in the idle state.

Two monitoring methods are defined in the RMs plugin, publish() and publishFullStat(). This methods are periodically called by the RCD application to monitor the board status. More details will be presented in Section 5.6.

Other methods, that are used for different FTK specific duties, are also defined in the RMs. The functions used to perform the recovery procedure (which will be described in Section 5.5.5) are an example of them.

All the RM methods described above are directly called by the controller RCD, which, in turn, responds to external calls (e.g. from the TDAQ software or from the user or system expert).

For the management of SW issues, an error handling infrastructure common to all the ATLAS subsystems is provided. Apart for external libraries, the error handling is performed through C++ exceptions. Four different level of errors are made available from the TDAQ code: *info*, *warning*, *error*, and *fatal* (*debug* is also available, but it is only used for debugging purposes). Exceptions are raised in the code every time an operations fails, usually because of board access or link setup problems, and they can be caught at different levels. If the issue can be cured, this is done at the immediate upper level, usually by repeating the failed operation. If the problem can not be cured, the exception is sent to the further code level, where more informations are available for the definition of the error severity. Besides for the fatal severity level, the application execution remains anyway unchanged (operation may in any case be skipped following the standard *throw-catch* C++ behaviour) after an error is raised. If, instead, a fatal error is thrown, the controller application is automatically set in the ERROR state, and needs to be restarted. Only errors that are preventing FTK from running or that involve a wrong FTK configuration, as shown in Section 5.5.1), are identified as fatal errors.

5.4 FTK CONFIGURATION DATABASE

The configurations of the ATLAS systems is managed through a package called Object Kernel Support (OKS) [71]. The OKS system is based on an object model that supports objects, classes, associations, methods, data abstraction, inheritance, polymorphism, object identifiers, composite objects, integrity constraints, schema evolution, data migration, active notification and queries. The OKS system stores database schema and data in portable Extensive Markup Language files and allows different schema and data files to be merged into a single database. The OKS schema files define the classes and the properties of these objects. Data files are instead used to initialize these objects

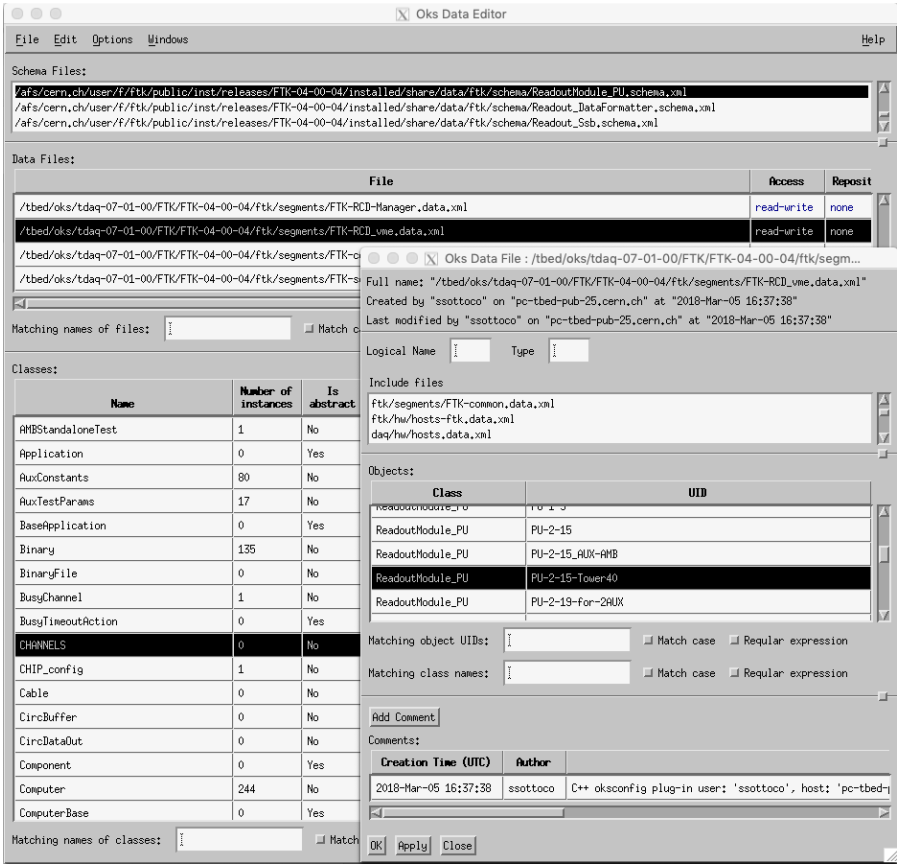


Figure 5.6: Example view of the OKS Data Editor tool for the configuration management of the OKS data files. In this example, the different PU RMs contained in the one of the VME RCD can be seen.

and to set their properties. From the OKS description, C++ objects are automatically generated from DAL.

OKS includes GUIs to design database schema and to manipulate OKS objects, as the OKS Schema Editor and the OKS Data Editor tools. Moreover, python APIs are available and can be used to automatize the database creation. The configurations of all the ATLAS subsystems have to be defined by mean of this package.

The FTK online SW has been integrated in OKS since the beginning of the commissioning phase. Schemas, defining the classes representing the different FTK components and the properties of these C++ objects, have been created and are automatically updated at every FTK software release. A schema file has been created for each of the boards RMs. The OKS data files, used to initialize the FTK components with the correct attributes (they specify all the attributes of a given RM, defining a given FTK configuration), were initially created by the FTK experts by "hand", by mean of standard text editors or by the two OKS Editor tools cited above. As an example, a picture of the OKS Data Editor tool is shown in Figure 5.6. These tools allow the graphical

creation/manipulation of the OKS data files. Beside the simplicity of usage of the OKS tools, with the advancement of the commissioning and the growing-up of the FTK system they have started to become cumbersome.

In the database, each FTK electronic board is defined by an OKS object (class). Each of these objects contains dozens of parameters, defining, e.g. the link between the boards, the constants and PatternBanks that have to be loaded (and that are different for each board), the FW specific configurations, etc. Moreover, some of the boards C++ objects are defined by different sub-classes, e.g. by the objects defining the different FPGA configurations, each of them with different specific parameters. This lead to thousands of OKS objects to be created (for the final FTK configuration). In order to cope with this huge number of OKS objects, the use of automatic configuration generators for FTK is mandatory.

For this purpose an FTK tool, called **ftk_maker**, has been created. This tool, developed starting from the daq *partition_maker* package, uses the OKS python binding to create and manage the OKS configurations for FTK. It is able to generate a full set of configurations for FTK, to update already available data files and to generate all the TDAQ infrastructure configurations required to run the FTK code. This tool is based on python dictionaries in which standard configurations for all the FTK boards and components are defined. The dictionaries map the FTK boards and the links between them, the PatternBanks and constants to which a given board is associated, and the default values of the configuration parameters. Another python dictionary, that can be generated directly from the tool, can be used to manage which boards have to be included in the configuration, for which type of run the generation have to be performed and to eventually change the different parameters values. Tools for the boards checksum computation are also integrated, and called automatically from the tool storing the computed values inside the generated OKS data files. More details about this will be given in Section 5.5.1.

The development of this tool has been completed in the first part of 2019 and is currently the standard tool for the FTK configurations generation.

Another type of database used by ATLAS, and in which FTK have to be integrated, is the condition database. This database is used to record the experimental conditions at the time the experiment event data were collected. Moreover, it is characterized by time windows, offers a time booking for all these conditions. This database is particularly useful to FTK, by instance, for the storing of the patternbanks and constants files. While the location of these files is defined in the OKS configuration, their content has to be stored in the condition database. The ATLAS condition database is based on COOL [72], a framework developed to support persistency for several relational technologies (Oracle, MySQL

and SQLite), based on the POOL Relational Abstraction Layer (RAL) [73] and on the SEAL libraries [74]. It provides specific software components to handle the versioning of the conditions data.

The integration of COOL in the FTK online software is one of the main tasks for the current Long Shutdown SW development. Currently, the different FTK boards configurations are stored locally and the configurations in which FTK run are not booked. Moving to the use of the condition database, all this configuration data will be stored in the database, and all the informations for a given run will be available for offline analysis. The migration to COOL started at the end of the Run2. Currently, only the COOL integration of the SSB configurations has been performed, but it still has to be deployed. Development is ongoing also for the AMB and AUX boards, with the plan to migrate all the VME boards by the end of the year.

5.5 THE FTK ONLINE SW CHALLENGES

The development of the FTK online SW had to face many issues, some of which identified only with the experience gained from the commissioning of the system.

The online SW has to be reliable. The FTK applications have to be able to survive for the entire duration of the ATLAS runs. Moreover, they have to be able to cope with an infinite variety of unexpected problems, as board access problems or concurrency issues, configuration or monitoring problems, board processing troubles, etc. The limited resources of some controller machines poses additional complication to the software. Memory leaks can have dramatic outcomes, as the death of the application in the middle of a run.

FTK has also to respect some precise requirements, posed by the ATLAS environment. The time available for the completion of each FSM transition is limited, and it is driven by the other subsystems. The operations of all the ATLAS subsystems are synchronized. Before moving to the next transition, all the subsystem have to complete the previous one, and have to reach the proper FSM state. If a subsystem requires more time to complete a transition than the other subsystems, it introduces delays in the whole ATLAS infrastructure. This poses the requirement for the FTK transitions times to lie in the shadow of the other ATLAS subsystems. Being the configuration of FTK very complicated, this is challenging and it required the development of ad hoc solutions.

Moreover, the limited number of transitions that ATLAS provides presents another challenge for the FTK configuration. Configuring the FTK system requires complex procedures, with many different operations to be executed following a well-defined order and to be repeated in case of error. The use of a FSM helps to ensure that the operations follow the predefined order, but fitting these operations

in the 5 transitions provided by the ATLAS FSM for the subsystem configuration is very challenging.

Beside the common problems that every online SW project has to cope with, in the following the FTK specific issues and the solutions developed by the FTK online SW team will be presented.

5.5.1 *Configurations loading*

The configuration loading of the VME boards is a particularly critical operation. FTK needs to load on its FPGAs and AM chips a huge quantity of data (the constants and PatternBanks used for the fitting and pattern matching procedures), which is of the order of 1 GB for each board. This operation is challenging for two main reasons.

First, due to the amount of data and to the limit of the VME bus and its exploitation by our SW, the required loading time is much bigger than the available time for the ATLAS configuration transitions. Loading the PatternBanks on the AMs takes about 5 minutes per board (resulting in 80 minutes for a total of 16 PUs to be configured through the same controller application), compared with the about 3 minutes it takes to configure the entire ATLAS detector. Optimizations are possible, but they can't overcome the problem. As an example, the use of the block transfer for the VME data transfer, which would guarantee a 20 MB/s speed (the theoretical maximum speed is of 40 MB/s) is not everywhere exploited, but would not reduce anyway the configuration time to an acceptable value.

Secondly, the SBC on which the controller application runs is limited in both memory and CPU power, which makes it impossible to implement a full paralellelization of the loading operations for all the crate boards. The VME RCDs have to run on the crates SBC. These machines are limited in both CPU power and memory, with 1.9 GHz 4 Cores, 4 GB RAM, and no Hard Disk available. The parallel loading of the configuration files on the SBC memory would lead to a crash of the application.

To cope with these problems we decided to decouple the data loading step from the data taking operations. During the configure transition of the data taking runs, the FTK online software only checks for the correctness of the content of the board memories through the computation of checksums (a sketch of the loading logic is shown in Figure 5.7). These checksums are computed both via FW, on the full board memory content, and SW, on the expected configuration data, and compared with each other. In case of mismatches (e.g. because of errors in the configuration loading procedure), the software prevents FTK from starting the data taking session. In the special interfill configuration runs, the configuration data are, instead, allowed to be loaded on the boards.

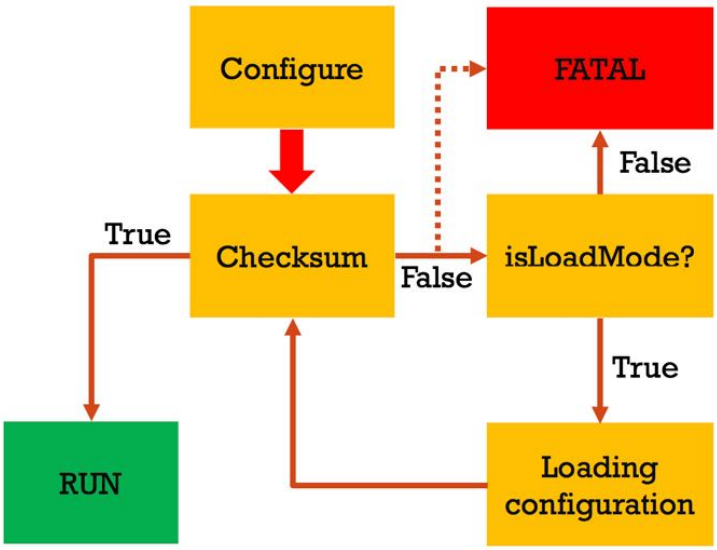


Figure 5.7: FTK configuration loading logic. The loading of the configuration on the boards is performed only during a "configuration run" identified via the isLoadMode flag.

In order to prevent the possibility to enable the loading mode during a data taking run, a simple OKS switch between the two run types wouldn't be safe enough. We decide to manage the run type selection through an environmental variable, FTK_LOAD_MODE. This variable can be defined only in the partition object, i.e. the root configuration file, that can only be modified in private FTK runs. This makes impossible to run in ATLAS in the configuration loading setup.

A function defined in the ftkcommon package is used to check for the presence of the environmental variable. This function is called by all boards RMs, at the beginning of the configure transition. If FTK is in "loading mode", the configuration loading takes place.

If a board is found to have FW and SW checksums not matching in an ATLAS run and, thus, having a wrong configuration loaded, a FATAL error is raised and FTK can not take part to the data taking session (the data produced by the FTK system would be invalid). In the meantime the board will be fully reconfigured.

5.5.2 *FTK sub-transitions*

As presented in Section 5.5, ATLAS provides 3 distinct transitions for the start-up of its subsystems: *configure*, *connect* and *start*. These transitions are not enough to fit all the required FTK operations.

The FSM transition are fundamental for the time ordering of the different SW objects operations, and, thus, of the FTK boards configuration operations. The boards configuration is required to be done in

separate steps (requiring access to the boards, as already mentioned in Section 5.5) that are driven via SW. The use of multiple transitions guarantees that all the boards execute the given operations before moving to the next transition, and, thus, the next configuration step. The FTK configuration requires more than 3 transitions.

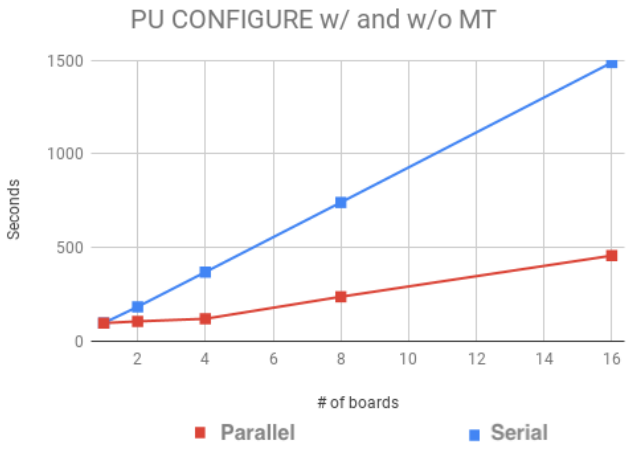
As an example, the setup of the links between the FTK boards requires a multistep handshake. The link initialization procedure requires firstly a reset of the upstream transmitter, followed by a reset of the downstream receiver and finally a check about the link status on the receiver side. If the checking on the status of the receiver is performed while the other board transmitter is resetting, race conditions can occur and the link alignment can fail.

Another example comes from the starting of the IM/DF boards. In order to avoid backpressure problems, the IM/DF boards have to start to accept data only after all the downstream boards have been correctly started. This requires to distribute the prepareForRun() method to different sub-transitions for the IM/DF RM and the other boards ones.

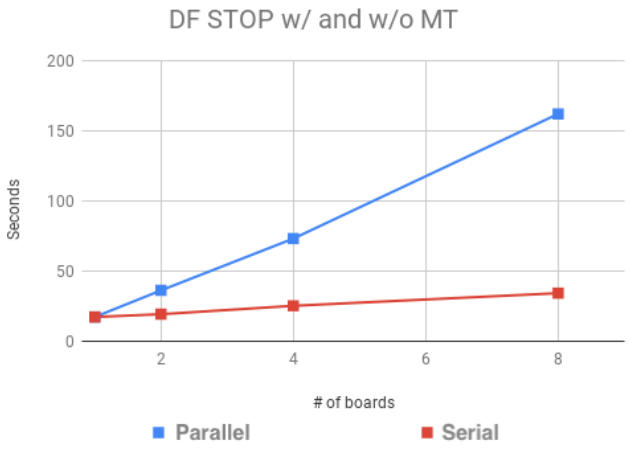
Moreover, checks on the FW version loaded on the board FPGAs are required and have to be performed in a separate transition, before any board starts its configuration operations. The expected FW version for each FTK FPGA is stored in the configuration database. If the FW version on which a given FPGA is programmed mismatch with the expected one, the run has to be restarted and the board reprogrammed. In order to do this, when a mismatch is found a FATAL error is raised and the application is automatically moved to the ERROR state. No further transitions can be called. During the commissioning has been noted that configuring a board multiple times without passing through the stopping procedures (that reset the FPGAs) can lead to problems, curable only by powercycling the card. This has to be avoided. With the use of a separate transition it is possible to ensure that all the FPGAs of all the application boards have performed the FW check before starting the configuration, thus avoiding the problem.

As shown in Figure 5.5, to fit all the required operations 3 FTK specific sub-transitions have been introduced, one for each of the standard start-up transitions. The sub-transition *checkconfig* has been dedicated to the FW checking. The sub-transition *checkconnect* is dedicated to the link alignment checking, while the transition *startNoDF* is designed to call the prepareForRun() method inside the IM/DF RM (in the other RM prepareForRun() is called during the standard start transition).

These sub-transitions are defined in OKS and the sub-transition commands are dispatched to the FTK online SW via the controller application. The sub-transitions are executed before the standard transition to which they are associated to take place. For the ATLAS point of view, the FTK sub-transitions are part of the main transitions.



(a)



(b)

Figure 5.8: Measured transition times for one VME crate (a) and one ATCA shelf (b) as a function of the number of enabled boards. The measurements have been performed with and without the parallelization of the given transition. Only the measurements of one of the transitions has been reported.

5.5.3 *FTK transition times*

As presented at the beginning of this section, in order to not introduce delays in the start-up of the ATLAS system, the FTK transition times have to be kept within the other subdetectors ones. Transition times and scaling measurement campaigns have been performed in the last two years, revealing excessive execution times for most of the FTK transitions, especially while running with multiple boards or in full crate configurations.

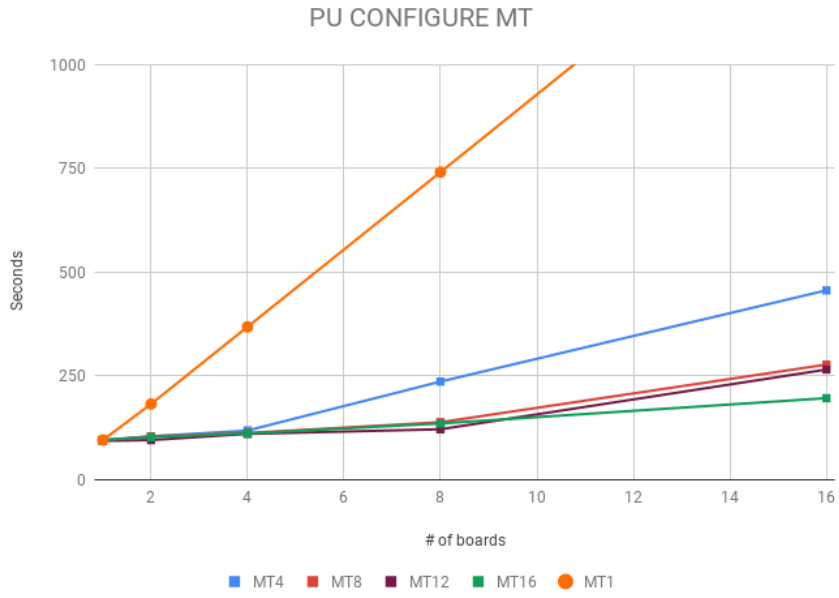


Figure 5.9: Transition times of the PU configure transition as a function of the number of enabled boards, for different values of the maximum number of thread spawned.

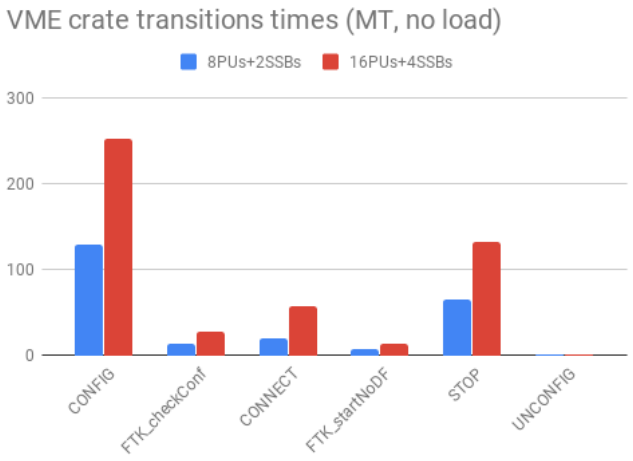
The RCD provides the possibility of running all its RMs in parallel only for the configure transition. The operations of all the other transitions are executed serially. This lead to two obvious problems:

- the FTK transition times scale linearly with the number of RM enabled in the RCD;
- the resources available on the controller machine (as CPU and bandwidth) are not fully exploited.

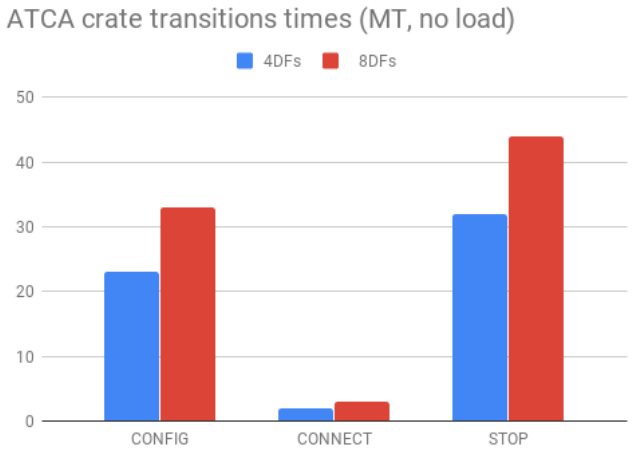
In order to reduce the transition times, we decided to extend the RCD functionalities, enabling the possibility of running in parallel on separated threads, i.e. through MultiThreads programming (MT), all the different RM transition methods and monitoring functions. A new RM FTK C++ base class has been developed. This class, from which all the board specific RM classes have to inherit, provides a thread pool able to manage the thread spawning for the different RMs of the given RCD. The transitions to be parallelized and the maximum number of thread to be spawned for each transition are configurable from OKS RM attributes.

The migration to the new framework has been completed during the end of the 2018.

A measurement campaign on the FTK transition times has been repeated during the 2019, meant to verify the gain in performances obtained with the introduction of the FTK SW parallelization. The



(a)

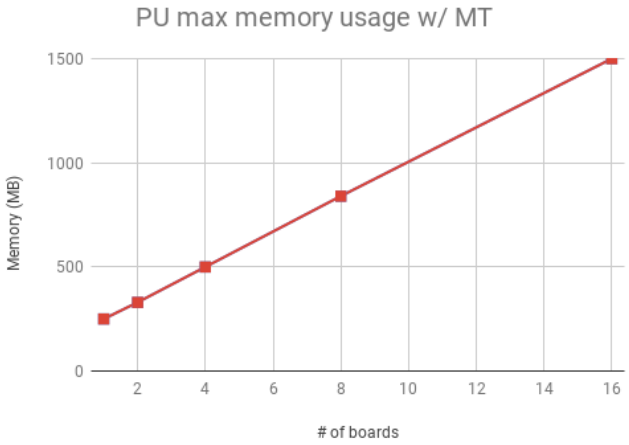


(b)

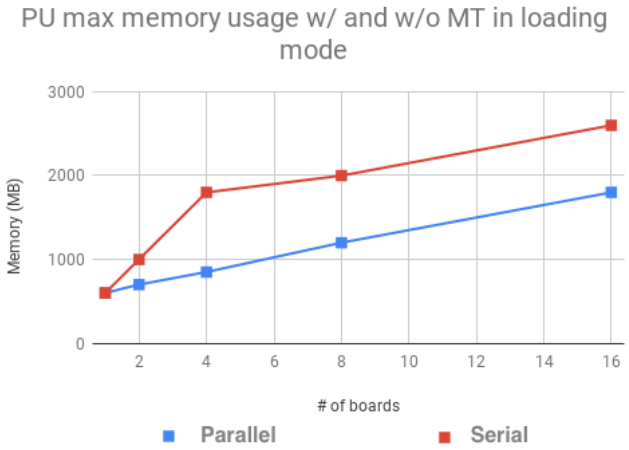
Figure 5.10: Measured transition times for one VME crate (a) and one ATCA shelf (b). Only the transitions with execution times greater than 1 s are shown. The crate (shelf) configurations of 8+2 (4) and 16+4 (8) boards are shown. All the measured FTK transition times are smaller than the ATLAS ones.

ATLAS transition times, posing the limit for the time available to the execution of the different transitions in FTK, are shown in Figure 5.5.

Figure 5.8 shows the configure (stop) measured transition time as a function of the number of boards enabled in a VME crate (ATCA shelf). It is possible to notice the drastic reduction in time moving from a serial to a parallel transition execution. In Figure 5.8a, it is also possible to notice an almost perfect scaling for up to 4 boards, followed by a linear scaling for up to 16 boards. This behaviour, not visible in Figure 5.8b, is due to the limitation to the number of spawned threads that was set to the number of CPU cores available in the SBC (4 cores) to limit the memory usage.



(a)



(b)

Figure 5.11: Measured maximum memory usage for the VME crate for the standard (a) and loading mode (b) configuration.

Optimization studies have also been performed. In Figure 5.9, the configure transition time for the PUs as a function of boards enabled for different values of the maximum number of threads spawned is shown. As can be seen, limiting the number of thread to the CPU cores number is not the optimal solution. This is due to the non saturation of the CPU during the transition, that can be better exploited adding additional threads. The final outcome of these studies shown that the best thread configuration is removing the thread number limit at all, as expected.

In Figure 5.10, the final transition times for the VME (ATCA) crate (shelf) are shown. As can be seen from the plots, the FTK transition times, exploiting the MT transition configuration, are under the ATLAS limits.

Finally, the maximum memory usage of the RCD application for the VME and ATCA components has been measured. This information is particularly critical for the VME case, due to the limited SBC resources (SBC has 4 GB of memory). In Figure 5.11a, the maximum memory usage for a VME RCD as a function of the number of enabled boards is presented (for MT transitions). As expected, the maximum memory scales linearly with the number of boards. More critical is the loading mode configuration, where the quantity of memory required for the configuration loading is higher. In Figure 5.11b, the maximum measured memory usage is shown, for both the parallel and serial case. Due to memory limitations, the maximum number of threads has been fixed to 6. As can be observed, the serial case requires less maximum memory for the RCD operation. Anyway, limiting the number of threads can let the maximum memory usage to be kept under control, while speeding up the loading operations.

5.5.4 *Board access interference*

During the commissioning period, issues deriving from concurrent accesses to the boards during the configuration/monitoring processes were observed. The problems were accentuated after the parallelization of the different RM functions.

These issues were due to non atomic board access operations. The board configuration procedures and monitoring require write-read accesses to some of the board registers, accesses to be repeated in case of errors. These procedures are particularly critical for that components that require the dispatching of the operation inside the board itself, as the access to the AM chips or the access to the IM through the master DF board. These kind of operations require not to be interrupted by other concurrent requests.

If during one of these procedures a new board access is requested, for example by a monitoring operation, the configuration procedure of the component may fail, or data error be raised. In order to solve these issues, we introduced the use of mutexes for the board access serialization, blocking all the board accesses when another non atomic operation is ongoing on the given board.

Moreover, multiprocess accesses have also to be considered. Wrong PatternBanks have been observed to be written in the AM chips due to concurrent board access from the Run Control application and the DCS monitoring one. In order to overcome concurrency issues from different applications, semaphores were adopted. In the AM-DCS case, a board register semaphore has been introduced. A board register is set when the AMB is undergoing configuration. The DCS checks that register, and stop its monitoring until the "lock" has been released. Despite this solution solved the problem, blocking the DCS board monitoring for a long amount of time may be dangerous (the DCS

has to check for the board temperature, that has always to be kept under control, as discussed in Chapter 4). For this reason, a SW system semaphore is under development.

The solutions presented above fixed the concurrency issues, but introduced some constraints. The access serialization slows down both the monitoring operation and the board responsiveness to the control software. The SW solutions are fast and easy to be implemented, but a proper solution would be a more complicated management of the accesses at fw level. In the future we plan to work on this direction.

5.5.5 *FTK automatic procedures*

The ATLAS online software framework provides automatic procedures used to disable and recover faulty subsystem components while running: the Stopless-Removal and Stopless-Recovery procedures [75]. Exploring these procedures is mandatory to minimize the impact of FTK on the ATLAS data-taking while maximizing the run-time within the ATLAS system. Due to the system complexity, the development of such automatic procedures was particularly challenging for FTK.

The Stopless-Removal procedure allows to exclude a part of the readout that is slowing down the data taking session. The Event Builder (EB) is the TDAQ system responsible of the data collection from all the different ATLAS readout channels [76]. If the data coming from one of the channels is missing, after a timeout the ROS generates an empty fragment to substitute it. If the channel fragments are persistently late or missing, e.g. because of a fault in the given subsystem, the timeout introduce a delay in the whole ATLAS data taking session. This delay can be overcome providing the EB with an empty fragment immediately generated by the readout system itself. The Stopless-Removal procedure is meant for this purpose. When an issue causing delayed or missing fragments is detected, the given subsystem can trigger the removal procedure who puts the given link in *discard* mode, allowing the Read Out System (ROS) to generate empty fragment for the subsystem itself and, thus, avoiding the introduction of delays.

The triggering of the removal procedure requires a deep knowledge on the output status and on the possible failure conditions, not yet available for a system under commissioning. Due to the complexity in monitoring the FTK output, we decided to base the issue identification on the first downstream component, the ROS. We developed a watchdog python application that monitors the content of the FTK ROSes via IS, looking for missing/late fragments. The removal procedure is automatically triggered by the watchdog application when these quantities are found to be more than a configurable threshold. An automatic call to the FTK_Solo tool is, also performed. This is required in order to ensure the retrieval of the FPGA monitoring informations, mandatory for identifying and debugging the cause of the problem.

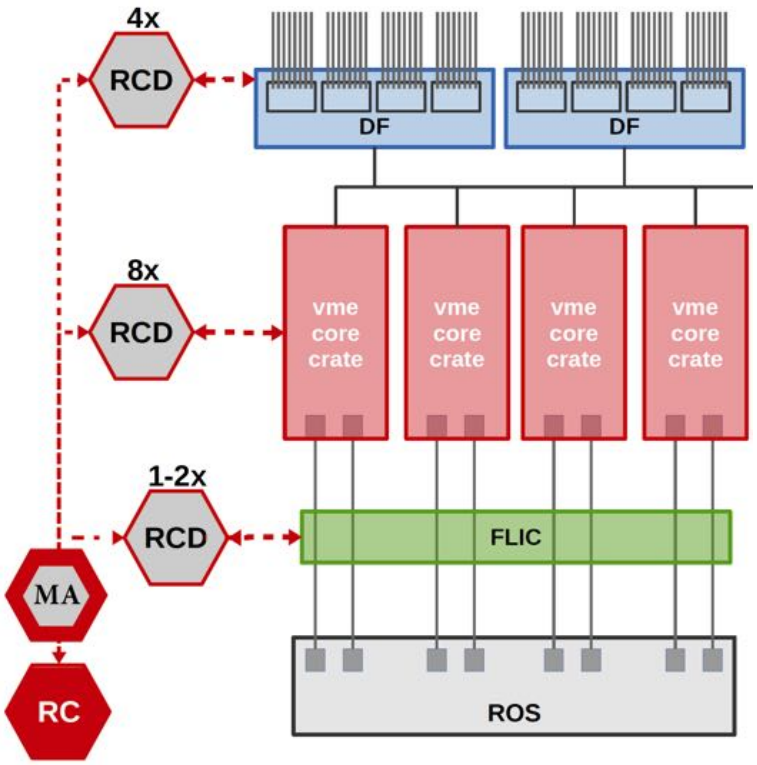


Figure 5.12: Sketch of the stopless-recovery procedure infrastructure of FTK. The Manager Application (MA) manages the communications between the board controller applications (RCDs), dispatcing the recovery transition actions and managing the internal FTK FSM.

More challenging was the development of the FTK Stopless-Recovery procedure. This procedure allows a user to re-activate components that had been previously disabled during the run (from a stopless removal procedure). We developed an ad-hoc Manager Application (MA) responsible to manage the communication between the different board controllers. It also controls an FTK internal FSM, required to synchronize all the operations needed for the system reconfiguration. A sketch of the FTK Stopless-Recovery infrastructure can be seen in Figure 5.12. The recovery procedure is started by an FTK expert via a specific command sent to the MA. The MA starts the procedure by initializing an internal FSM and dispatching the recovery transitions to the board controller applications, waiting for the acknowledge and the operation-ended messages before asking for the next transition. In order to recover the system, all the operations executed in the normal transitions are performed also in the recovery ones. Starting from STOP, a copy of all the different transitions are executed in the predefined order until the running state is reached. When the recovery procedure is completed by all the boards, the MA manages the communications with the ATLAS Central Hint and Information Processor

(CHIP), which recovers the disabled readout link between FTK and ATLAS, ending the recovery procedure.

5.6 MONITORING OF THE FTK SYSTEM

Monitoring the FTK internal data flow and the quality of the output data is fundamental. Without a good monitoring infrastructure, the FTK system would act as a black box. Problems would not be detectable, or at least not before analyzing the output data. Moreover, without having access to the board informations and the board registers as well as to the portion of the data produced by the different board chains, the debugging of the issues and the bugs identification would be almost impossible.

The FTK monitoring is performed at different levels: board data flow, board data quality, system output data quality and high level data quality. While the last two levels are still under development, the board data flow monitoring and the board data quality monitoring are fundamental for the commissioning phase of the project and have started to be developed since the beginning of the commissioning phase.

In this Section, more details about the FTK monitoring infrastructure and tools will be provided.

5.6.1 *Board data flow monitoring*

The board data flow monitoring is responsible to check the status of a given FTK board (e.g. the board is stuck - processing is ongoing). Monitoring this is of particular importance during the data taking, as it is used to understand if the system is correctly processing data or if problems occurred. In case of problems, the FTK system have to be removed from the run in order to not introduce delays or write wrong data to the ATLAS data stream, and can be subsequentially recovered (using the recovery procedures presented in Section 5.5.5).

Information on the status and processing of the FTK boards is stored in specific board registers, read by a monitoring thread of the board RCD through the VME/IPBus protocols. Two different monitoring methods are available at the RM level, settable with two different reading rates. Thanks to the parallelization of the FTK framework, each RM can now count on its own threads for the monitoring, drastically reducing the minimum reading rate the system can sustain. Moreover, also the two monitoring functions of each RM can currently run in parallel on two different threads. However, due to concurrency issues happening while accessing in parallel a single board, the access to the given board has been serialized with the use of mutexes, reducing the effectiveness of this further parallelization. Figure 5.13 shows the behaviour of the publish and publishFullStat functions execution for

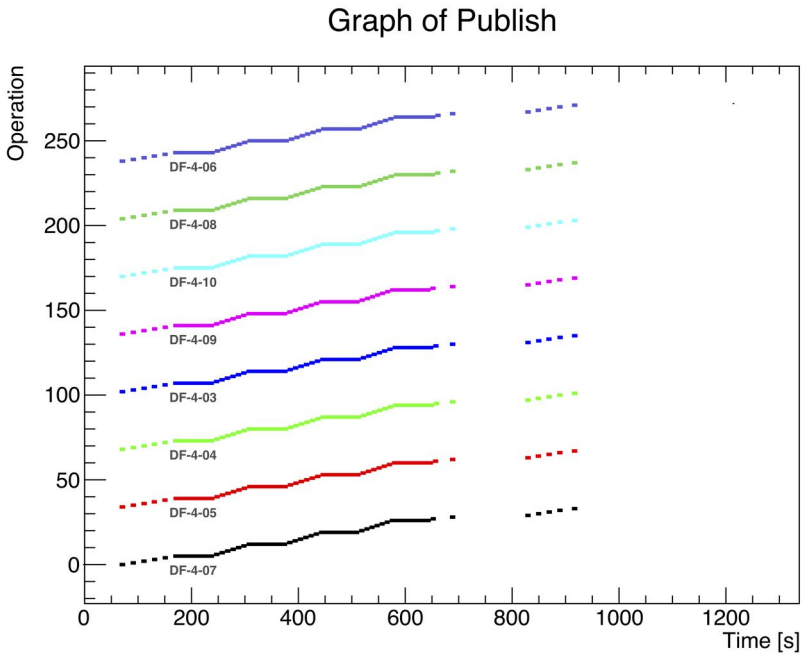


Figure 5.13: Plot of the time execution of the two RM monitoring functions for the DF boards. In the plot, the y value represent the execution of a given operation, i.e. the execution of one of the monitoring functions. The faster operations (the short lines on a given y value) represent the execution of the publish function, while the longer operations represent the execution of publishFullStat.

the DF RM. As can be seen from the plot, the function execution is parallelized between the different boards. However, due to the mutexes described above, the single board parallelization is blocked, resulting in a serialization of the monitoring functions.

Specific board registers are read by the given thread on a frequency that depends on the urgency of the stored information (e.g. some information will be used in the future to trigger the Stopless-Removal procedure, requiring a fast monitoring cycle).

In order to allow for expert data access, this information is made available through the ATLAS information [77] and histogram [78] services. The monitoring functions publish the content of the read register under the form of vectors or histograms, depending on the service used.

Finally, post mortem access is given through the use of high level monitoring applications, as the Grafana application [79]. This application allows to retrieve the same data described above, that have been persificated. An example of an FTK Grafana dashboard showing board data-flow information can be seen in Figure 5.14.

Through the tools and services presented, the online and offline monitoring of all the dataflow information, as the number of events processed or the number of tracks fitted from each board, but also

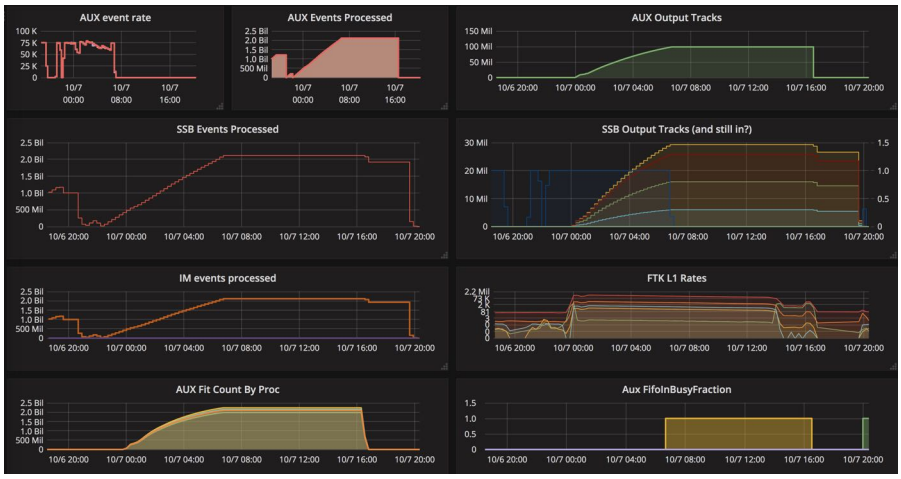


Figure 5.14: Example of high level monitoring of board data-flow variables for FTK (Grafana dashboard)

informations about the status of the FPGAs, the level of the FIFO occupancy, etc, is possible.

5.6.2 *Board data quality monitoring*

The board data quality monitoring is responsible to check if an FTK board is producing a reasonable output. It is particularly critical for debugging the FW, representing a snapshot of the data contained in the board during the processing.

The information on the data quality are stored in spybuffers: board circular buffers in which copies of the internal processing data are stored. The typical use-case of the spybuffer is to monitor the input/output of a FW module prior to an error occured. Since the FW debugging requires the collection of board spybuffers for the same event from different boards, a freeze mechanism, able to block the monitoring buffer writing as soon as an interesting error occurs, has been developed. This freeze operation triggers the board FW to deliberately stop updating its spybuffers (without affecting the processing FIFOs). In order to access the spybuffer information, an online spybuffer readout system has been developed, making a new use of the ATLAS event monitoring (EMON) service [80].

The spybuffer content is read out periodically from the boards through VME/IPBus protocol via one of the two monitoring threads of the board control software. Once spybuffers are read out from each board, they are wrapped inside the ATLAS event format via a custom FTK-EMON interface. An identifier integer is used to fully define their origin (i.e. stores the board type, board number, input/output buffer, lane number). The encoding of the spybuffer identifier is presented in Figure 5.15.

Board internal	Subdetector ID	Data type	Reserved	I/O	Board type	Board number
31-24	23-16	15	14	13-12	11-08	07-00

Figure 5.15: Bit allocation for the 32 bit FTK spybuffer identifier encoding.

Data are then made available to the EMON service. The FTK-EMON interface was designed to allow the possibility to temporarily store spybuffers from different boards while waiting for client requests, and to allow data sampling based on selection criteria. The choice of using the EMON service and the ATLAS event format for the spybuffer wrapping allowed us to make use of the persification infrastructure already available from the common ATLAS SW, without having to develop a new system from scratch.

The current spybuffer readout logic has, however, some limits. Due to the limitations given by some of the boards internal buses (e.g. I^2C for the IM/DF intercommunication, as presented in Section 5.5.4) the spybuffer readout for some of the boards is very slow (e.g. about 1 minute for one DF board). Moreover, because of the mutexes required to solve the board concurrent accesses problem, the board data-flow monitoring is blocked for the whole spybuffer readout time, even if it has higher priority. In order to cope with this problem an optimization of the readout logic is under development.

A single board register will be filled with a spybuffer error status word by the FW every time a freeze will occur. This status register will be read out by the board data-flow monitoring thread and will be used as a trigger for the full spybuffer reading. The choice of reading the full spybuffer or not will be based on a set of predefined error-state selection criteria defined in the board configuration. With this logic the spybuffer readout will be performed only when a given error will occur.

Another optimization under development is the addition of a new board register spybuffer to the board FW. Currently, the freeze signal blocks the overwriting of the spybuffer content, leaving the board registers untouched and able to continue to monitor the boards. While having the board monitoring working after a freeze is mandatory, the information stored in the board register at the moment of a freeze, i.e. when an error occurs, can be very helpful for debugging. In the current implementation, the registers information at the moment of the freeze is lost. The register spybuffer will instead act as the normal spybuffer, storing a copy of the board registers information when triggered by a freeze. The board registers spybuffer will be added as a tail to the spybuffer event, and published, together with the original one, as a data blob.

For the retrieving from EMON of the published spybuffers, a python tool, called *ftk_spy_dump*, has been developed. This tool is able to

```
Any additional errors in the system:
  DF-IM-crtShelf-2-slot-4:
    No additional errors found.

Rates measured by different boards:
  AUX-crate-4-slot-19:
    0.0 kHz
  AMB-crate-4-slot-19:
    0 kHz (last 0.1s)
  SSB-crate-4-slot-16:
    0.00 kHz

Events through each board:
  DF-IM-crtShelf-2-slot-4:
     855932328 events out
  AUX-crate-4-slot-19:
     855952312 events out
  AMB-crate-4-slot-19:
     855952313 events out
  SSB-crate-4-slot-16:
     855952311 events out

Additional information:
Road counts:
  Raw        :    1123239250   1220594282   |  Avg: 1144447804.75   Tot:    4577791219
  Per Event  :          1.31         1.43   |  Avg:          1.34   Tot:          5.35
  Rate (kHz) :             0            0   |  Avg:             0   Tot:             0
Fit counts:
  Raw        :    4034418075    143987843   |  Avg: 3103674657.75   Tot:   12414698631
  Per Event  :          4.71         0.17   |  Avg:          3.63   Tot:         14.50
  Rate (kHz) :             0            0   |  Avg:             0   Tot:             0
Track counts:
  Raw        :     416347513    450547577   |  Avg:  416046326.75   Tot:    1664185307
  Per Event  :          0.49         0.53   |  Avg:          0.49   Tot:          1.94
  Rate (kHz) :             0            0   |  Avg:             0   Tot:             0

--- OUTPUT TRACKS ---
  1352315766
```

Figure 5.16: Portion of the FTK_Solo summary output from an FTK Slice run. The tool is able to produce a summary file containing all the interesting information about the status of the boards. In the image, the measured boards rate, the event processed, and various processing counters are shown.

dump the spybuffer content in a file, accessible for offline analysis. The analysis of the spybuffers is instead executed by board specific tools, able to perform the unwrapping of the data and to analyze their content.

5.6.3 *The FTK_Solo tool*

The spybuffer monitoring method presented in the previous section is currently available, but its deployment is quite recent (end of 2018). Moreover, some of the tools required for the parsing of the spybuffers retrieved from EMON are still under development. In order to make possible the retrieving of this information, mandatory for the commissioning of the system, until the EMON integration will be fully completed, an interim tool, *FTK_Solo*, has been developed.

FTK_Solo is a python wrapper of board standalone tools, able to perform the reading of the monitoring information directly accessing the boards content. The tool is able to access the configuration database of given partition, parse the information describing the enabled FTK components, log on the proper controller machines and perform the data retrieval from the board content via VME or IPBus.

The tool is also able to process the retrieved information. It creates a summary file containing all the relevant dataflow values for each of the enabled boards, checks for possible backpressure and for its origin, and saves the spybuffers collected in a organized folder tree. An example of the tool summary output is presented in Figure 5.16. In order to ease the information retrieval, the whole tool output is automatically copied in a specific region of the FTK project workspace. Moreover, an automatic call to the tool has been included in the removal procedure, as described in Section 5.5.5, in order to ensure the retrieval of the spybuffers for the debug of the removal cause.

Due to its characteristics, and being the only available tool for the spybuffer retrieval, it is massively used in the commissioning of the FTK system. Despite its success, two main problems require its disposal:

- concurrency issues from the direct access to the boards;
- scalability issues with the number of boards to be monitored.

As described in section 5.5.4, concurrent access to the boards by different applications can lead to concurrency issues, e.g. resulting in the corruption of the data. Being a standalone tool that leaves outside the controller application, it is not possible to manage the board access through SW mutexes, and a system semaphore would drastically slow down the execution of the control software operations.

Moreover, the tool present serious problems of scalability. An optimization was performed at the end of last year, allowing the tool to directly access the configuration database for the board information retrieval (before this, the boards to be monitored and the different controller nodes were collected in an internal dictionary, difficult to be maintained with the scale up of the number of boards). Despite this, the tool execution remains very slow. A single thread is used to perform all the operations, resulting in dozens of minutes for reading the information of a single crate.

The natural solution to these problems would be to decouple the information retrieval from its analysis, moving the first to the use of the ATLAS whiteboards, i.e. the IS and OH services, as well as the EMON service for the spybuffers retrieval. These ATLAS tools are developed to be fully scalable, and all the FTK information are already published in them. A new tool, able to retrieve the information from IS and EMON, is under development. The replacement of FTK_Solo with this new tool is one of the main topic for the development of the online SW monitoring for the current Long Shutdown period.

5.6.4 *Bit level checks*

The collection of the boards spybuffers is not only important for debugging purpose, but it is also fundamental for the validation of

the given board output. While studies on the goodness of the full FTK chain output can be done directly analyzing the produced data, intermediate step validation, i.e. the validation of a given board output, has to be performed comparing the board spybuffers to a functional emulation of the given board. This procedure is called *bit level check*.

The boards bit level checks are performed with the use of board specific C++ functional emulators. These tools, different for each of the FTK boards, are able to completely emulate the whole board processing, as well as emulate the processing of a given board fw module, producing in output the data expected to be produced by the hardware.

In order to perform this validation, two different spybuffers are required (and available) for each of the FTK boards: input and output spybuffers. The input spybuffers contain an exact copy of the data as they are received from the board, before it starts its processing. They are the starting point for the bit level emulation, used to feed the emulation tools. The output spybuffers, instead, contain a copy of the output of the board, as it is before being sent downstream. The board specific emulation tool is fed with the input spybuffers, after they have been arranged in a specific format, and produce the emulated board output as result. This output emulation is finally compared with the board output spybuffers, and a bit level validation of the board output is performed.

Some of the boards, as the AUX board, present the possibility of perform bit level checks also on a given fw module. Input and output spybuffers for each module are available, and a complete step by step emulation can be performed. This kind of checks are particularly useful in order to debug the intermediate processing steps of the board. Being able to perform a full analysis of all the board processing steps can drastically ease the fw debugging, letting to spot issues or inefficiencies coming from a given portion of the processing.

Currently, the tools required to perform the bit level check procedure are available for all the FTK boards, and validations are periodically performed. However, some of the operations required, as the preparation of the spybuffer data or the execution of the board emulation, are of cumbersome execution. Different standalone tools are used by each of the FTK boards, and the tests have to be performed by the given board expert. An automation of the procedure is under development, with the goal to integrate bit level check execution on all the FTK runs. This new procedure will base the hardware emulation on FTKSim, already presented in Section 3.4.1. Currently this tool is only able to perform an emulation of the full FTK chain, since it is not possible to emulate a given step of the processing chain. Work is ongoing to integrate the standalone emulation tools in FTKSim, in order to have a single and easy to use framework to perform the bit level checks.

The new system will retrieve the required boards spybuffers from EMON every time they will be made available from the RC software, it will automatically run the boards FTKSim emulation, and will compare the two outputs, summarizing the bit level accuracy of the given board. The automation of the bit level check procedure is expected to be ready for the beginning of the Run3.

5.7 CONCLUSION

The online SW is one of the most critical items of the FTK project. As presented in this chapter, it is in charge of controlling the different FTK components, to monitor their status and to integrate the FTK subsystem inside the ATLAS environment.

At the current state, the FTK online software can be considered fully integrated in the ATLAS data acquisition system. The FTK online SW design goal was to make use of the available ATLAS tools, while spotting problems and bottlenecks of the given implementation. The adopted solutions have been presented, together with some open problems and the solutions that will be developed in the next months.

The monitoring has undergone big development in the last period, motivated by the advancement of the commissioning. In particular, the board data flow and the board data quality monitoring have reached a mature state, with a few optimizations still possible.

6

TAU TRIGGER STUDIES

Beside the activities already described, I performed some studies in order to increase the tau trigger acceptance based on the additional information provided by FTK. This leads to the definition of a new single-tau trigger chain with improved performance with respect to the one in use in the experiment. An estimation of the gain in acceptance and statistical significance that the application of such a trigger chain would have in a real physics search, and in particular for the search of a charged Higgs boson, will be shown in the following.

The presence of a charged Higgs boson is a feature of models with two Higgs doublets, in particular of the Minimal Supersymmetric Standard Model, which will be briefly introduced in Section 6.1. In Section 6.2, the ATLAS tau trigger system will be presented, in order to introduce the new FTK-based trigger chain in Section 6.3.2. Finally, in Section 6.4, the improvements in events selection and in statistical significance that the inclusion of the new FTK-based trigger chain can provide will be presented. In particular, this study is focused on the application of the new trigger chain to the search for a charged Higgs boson in the decay channel $H^+ \to \tau\nu$.

6.1 BEYOND THE STANDARD MODEL

The Standard Model is one of the great successes of the 20 th century physics. However, some good experimental and theoretical reasons make us believe that it is not the ultimate theory.

The first hint comes from the so called **Hierarchy Problem** [81]. In Quantum Field Theory (QFT), the Higgs mass receives corrections from self interactions, gauge loops, and fermion loops (especially from the top quark). These loops are quadratically divergent and are proportional to $\int d^4k(k^2 - m^2)^{-1} \sim \Lambda^2$ for some cutoff scale Λ. Explicitly,

$$\delta m_H^2 = \frac{\Lambda^2}{32\pi^2}\left[6\lambda + \frac{1}{4}(9g^2 + 3g'^2 - y_t^2)\right] \tag{6.1}$$

If $\Lambda >> 10\,\mathrm{TeV}$, the quantum correction to the Higgs mass is much larger than the mass itself ($\delta m_H^2 >> m_H^2$). In conclusion, the Higgs mass is quadratically sensitive to any mass scale of new physics. The implication of the hierarchy problem is that new physics is expected at the TeV scale, in order to eliminate the large loop contributions from above that scale.

A second hint comes from the **Matter-Antimatter Asymmetry** [82]. Presumably, the Big Bang created matter and antimatter in equal amounts, but, as evident, the known universe is currently mainly made of matter. An interaction able to violates the conservation of baryon and lepton number had to happen. The Standard Model is not able to generate an amount of Charge Parity CP violation, the effect for which matter does not exactly behave in the same way as anti-matter under CP transformation, which can account for the observed matter dominance. Hence, the matter anti-matter asymmetry is expected to be due to non-SM phenomena.

Other hints are also coming from **cosmological observations**. Measurements of the cosmic microwave background show that the SM can explain about 5 % of the energy content in the universe. It is estimated that about 27 % is constituted by dark matter and 68 % by dark energy [83]. The existence and properties of dark matter are studied from its gravitational effects on visible matter, radiation and large-scale structures of the universe. Dark energy is used to explain the acceleration of the expansion of the universe. Many theories predict that the dark matter particles would be light enough to be produced at the LHC. If they would be created, they would escape through the detectors unnoticed. However, they would carry away energy and momentum. Therefore, their presence could be inferred from the amount of missing energy and momentum in the events.

A final SM open problem is related to the **Neutrino masses** [84]. In the SM, neutrinos are massless and cannot change flavor. However, the observed number of electron neutrinos arriving from the sun is much smaller than the number predicted by the Standard Solar Model. This problem is known as the solar neutrino problem. To solve this problem neutrinos are required to oscillate between flavors, and this is only possible if they have mass.

New models have been proposed to replace or extend the SM in order to address these problems. Between them, the most notable is the supersymmetric model (SUSY) [85]. This class of theories extends the SM by introducing a new symmetry to it, which relates bosons to fermions.

Assuming there is one supersymmetric generator Q_α, an anti-commuting spinor, and applying it to fermions and bosons, respectively, yields:

$$\begin{aligned} Q_\alpha \, |\mathrm{fermion}\rangle &= |\mathrm{boson}\rangle \\ Q_\alpha \, |\mathrm{boson}\rangle &= |\mathrm{fermion}\rangle \end{aligned} \tag{6.2}$$

From this it results that each SM particle has an associated supersimmetric partner. This new particle is characterized by the same quantum numbers as the SM one, but with spin that differs by $\frac{1}{2}$. All the new particles, together with their SM counterparts are arranged in the so called supermultiplets, containing both bosons and fermions, with equal masses.

In order to give masses to the different fermions, in particular to the up-type and down-type quarks (as well as for the leptons), and, also, to preserve the renormalizability of the theory, at least two Higgs doublets are required within SUSY. The two Higgs doublets must have weak hypercharge $Y_{H_1} = -1$ and $Y_{H_2} = 1$:

$$H_1 = \begin{pmatrix} H_1^0 \\ H_1^- \end{pmatrix} \qquad \text{and} \qquad H_2 = \begin{pmatrix} H_2^+ \\ H_2^0 \end{pmatrix} \tag{6.3}$$

The minimal SUSY model that extends the SM is called *Minimal Supersymmetric Standard Model* (MSSM). This model introduces the least possible number of new fields while preserving supersymmetry and being compatible with the SM. The first point of this model is to choose in which supermultiplets the SM fields should be embedded. A common supermultiplets for fermions and gauge bosons can't be created, since the firsts belong to the fundamental representation of the gauge group, while the seconds to the adjoint one. Because of this, the SM fermions are placed in chiral multiplets, separated for quarks and leptons, while gauge bosons in the vector ones. Finally, the remaining degrees of freedom are filled up by the SM particles superpartners, the new particles of the model. These particles consist of scalar partners of the SM quarks and leptons. Their name is the same as the one of the SM particles their are associated to, but with an *s* prefix indicating "scalar". The left- and right-handed chiral states of quarks and leptons have different gauge transformation properties, so each of them must have its own complex scalar partner.

What have been done for the chiral multiplets needs to be done also for the vector one, that, apart from gauge bosons, has to contain also fermions. These new fermions are called gauginos, and they belong to the adjoint representation of the gauge group. Their names are gluinos, charginos and neutralinos.

6.1.1 *The Higgs Sector of the MSSM*

The only missing part for the completion of the MSSM is the definition of the Higgs sector. As described above, a single Higgs doublet is not enough to provide masses to both up-type and down-type quarks. Moreover, a second doublet is also required for the gauge anomaly cancellation. Because of this, and being the Higgs boson a scalar, another chiral multiplet is required, and the Higgs has to be accompanied by a fermionic superpartner, called higgsino. Therefore, the MSSM Higgs sector comprises two scalar Higgs doublets with opposite hypercharge (as presented in Equation 6.3) and four higgsinos: two neutral and two charged.

After the electroweak symmetry breaking, five different spin-0 particles are present:

- Two neutral scalar particles: h and H^0

- Two complex scalar particles: $H^{\pm}$
- A pseudoscalar particle: A

Two free parameters characterize the MSSM Higgs sector: $\tan\beta$ and m_A. The first parameter, $\tan\beta = \frac{v_2}{v_1}$, represents the ratio of the vacuum expectation values of two Higgs fields. It satisfies $(v_1 + v_2)^2 = v^2$, with $v \approx 246\,GeV$. The second parameter is conveniently chosen to be the mass of the pseudoscalar particle. From this two parameters, the different masses can be expressed as:

$$m^2_{h,H^0} = \frac{1}{2}\left[m^2_A + m^2_Z \pm \sqrt{(m^2_A + m^2_Z)^2 - 4m^2_Z m^2_A \cos^2 2\beta}\right] \quad (6.4)$$

$$m^2_{H^\pm} = m^2_A + m^2_W \quad (6.5)$$

At the tree level the following constraints are imposed on the Higgs boson masses:

- $m^0_H > \max m_A, m_Z$
- $m^\pm_H > m_W$
- $m_h \leqslant \min m_A, m_Z|\cos 2\beta| \leqslant m_Z$

Finally, the vacuum expectations values of the natural fields are defined as follows:

$$\langle H^0_1\rangle = \frac{v_1}{\sqrt{2}} \qquad \text{and} \qquad \langle H^0_2\rangle = \frac{v_2}{\sqrt{2}} \quad (6.6)$$

Interesting for the rest of the chapter is also the coupling of the charged Higgs boson with the τ lepton, given by the following:

$$g_{H^+\tau^-\bar{\nu}_\tau} = \frac{-i}{\sqrt{2}v}\left(m_\tau \tan\beta\,(1+\gamma_5)\right) \quad (6.7)$$

$$g_{H^-\tau^+\nu_\tau} = \frac{-i}{\sqrt{2}v}\left(m_\tau \tan\beta\,(1-\gamma_5)\right) \quad (6.8)$$

6.1.2 *The charged Higgs boson*

In a generic model incorporating a second Higgs doublet (2HDM) four different ways for the Higgs boson to couple to the fermions are possible. In type-I, all fermions couple to one of the Higgs doublets. The MSSM implementation of the 2HDM is of type II, and the dependence of production cross-section and branching fractions for the charged higgs is determined by this fact. In the lepton-specific type, all leptons couple to one of the Higgs doublets, and the quarks to the other. In the flipped-model, up-type quarks and charged leptons couple to the same doublet.

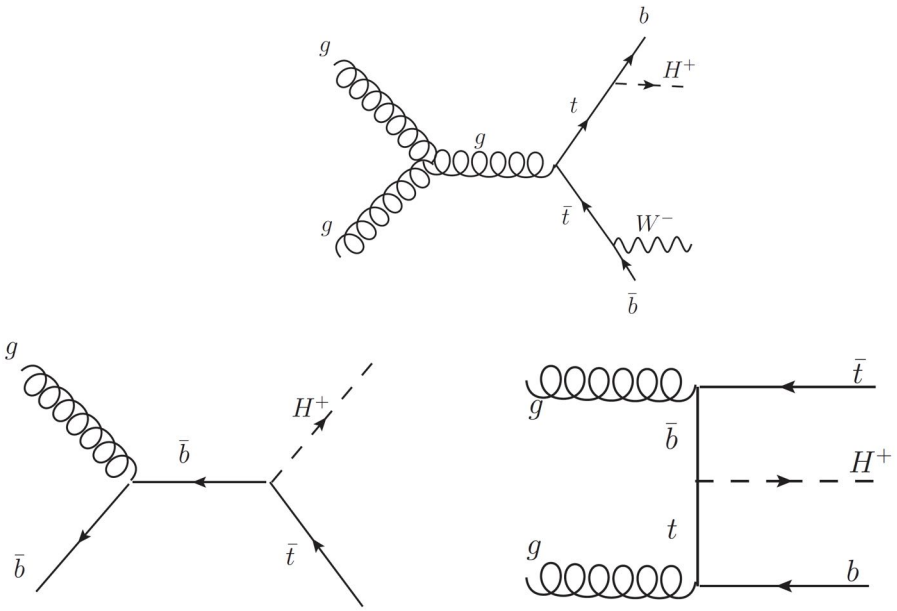


Figure 6.1: Example of leading-order Feynman diagrams for the production of a charged Higgs boson through top-quark decay (upper picture), 5FS production (left picture) and 4FS production (right picture).

In the type-II model, the model to which we will refer to in the rest of the chapter, the production and decays of the charged Higgs boson partly depend on it mass ($m_{H^\pm}$).

For low $H^\pm$ masses, and in particular for masses lower than the top quark mass, the primary $H^\pm$ production mechanism is through decay of a top quark, through the decay channel $t \to bH^+$. At the LHC, the leading production source for a low mass H^+ is via $t\bar{t}$ production.

For higher masses, in particular for $m_{H^\pm} > m_t$, the associated production with a top quark, via the $gb \to tH^+$ or $gg \to tbH^+$ channels, is the leading $H^\pm$ production mode at the LHC. This two production channels are called 5-flavor scheme (5FS) and 4-flavor scheme (4FS), respectively.

In Figure 6.1, the Feynman diagrams illustrating the leading-order production mechanisms of a charged Higgs boson are presented.

In contrast with the 4FS scheme, in the 5FS one the b-quark is considered as an active flavour inside the proton. While the 4FS and 5FS cross sections agree when computed to all orders in perturbation theory, any finite order summation yields different results in the two schemes due to different ordering of the perturbation expansion. At the leading order, the two schemes predictions can differ significantly. In order to avoid dependence on the chosen approximation cross sections, the calculations in the 4FS and 5FS approximations are combined and matched [86]. The matched cross section is given by:

$$\sigma_{matched} = \frac{\sigma_{4FS} + w\sigma_{5FS}}{1 + w} \tag{6.9}$$

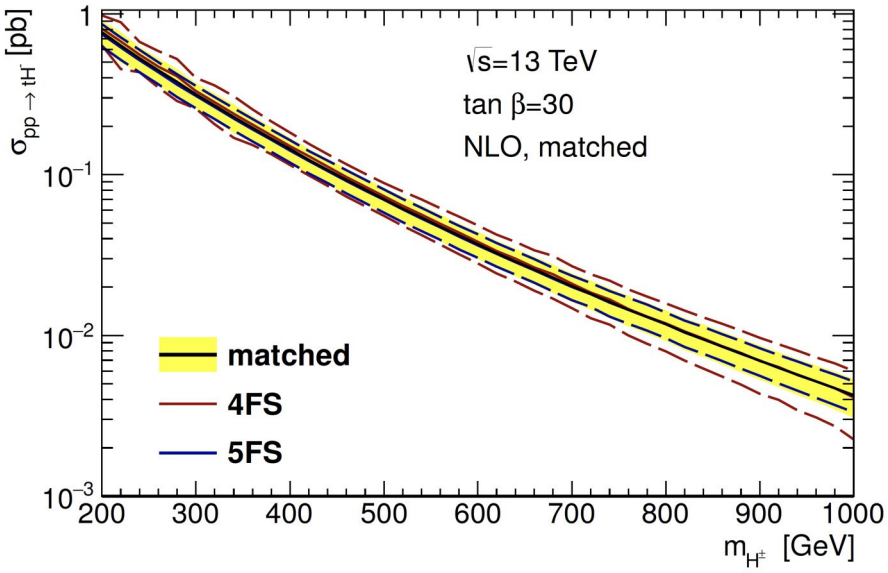


Figure 6.2: Production cross section for the heavy charged Higgs boson (i.e. resulting from the associated production with a top quark), as a function of m_{H^+}, in a type-II 2HDM with $\tan\beta = 30$ [87]. The predictions are shown for the left (5FS) and right (4FS) diagrams of Figure 6.1.

where $w = \log \frac{m_H^\pm}{m_b} - 2$, with m_b the b-quark mass, is the matching logarithmic term. Finally, the theoretical uncertainties are combined as:

$$\Delta\sigma_{matched} = \frac{\Delta\sigma_{4FS} + w\Delta\sigma_{5FS}}{1+w} \tag{6.10}$$

In Figure 6.2, the production cross section for a type-II 2HDM charged Higgs boson in the 5FS and 4FS production schemes is presented, for pp collision at a center of mass energy $\sqrt{s} = 13\,TeV$.

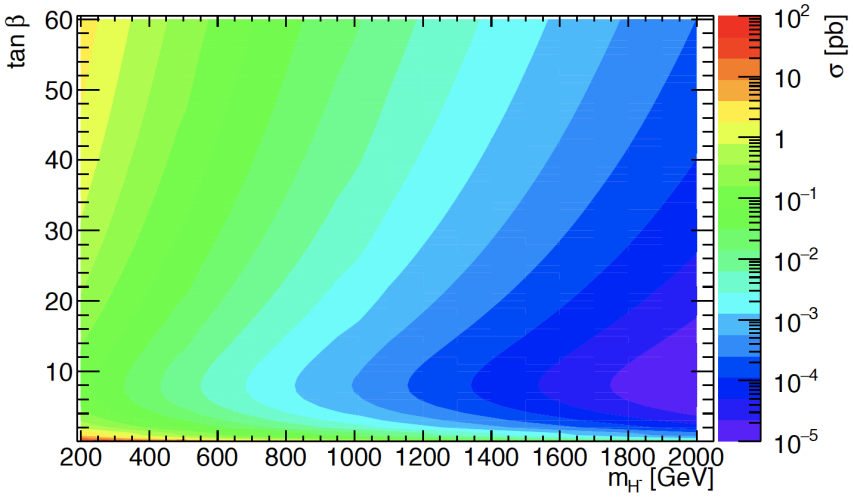
The production and decays of the charged Higgs boson are also controlled by the parameter $\tan\beta$, already defined in the previous subsection. In Figure 6.3, a two-dimentional plot of production cross section as a function of $\tan\beta$ and m_{H^+} for 4FS is presented.

Concerning the $H^\pm$ decays, the decay channel $H^+ \to \tau\nu$ is the dominant one at low masses, while the channel $H^+ \to t\bar{b}$ is the preferred one at high m_{H^+}.

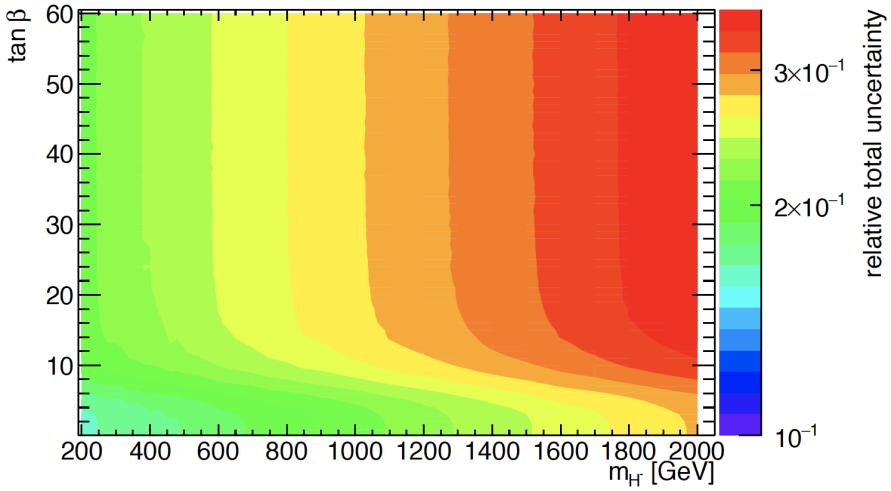
The partial decay width for $H^\pm \to l^\pm\nu$ is given by the following:

$$\Gamma(H^\pm \to l^\pm\nu) = \frac{G_F m_{H^\pm}}{4\sqrt{2\pi}} m_l^2 \tan^2\beta \left(1 - \frac{m_l^2}{m_{H^\pm}^2}\right)^3 \tag{6.11}$$

where l is a lepton with mass m_l and G_F is the Fermi constant. In Figure 6.4, the branching fractions for $H^\pm$ decays into SM particles are displayed, for $\tan\beta$ values of 10 and 50, as a function of the charged Higgs mass. For $\tan\beta > 3$ and low masses $H^\pm$, the main decay channel is $H^+ \to \tau\nu$. Above the top-quark mass, the branching



(a)



(b)

Figure 6.3: Two-dimensional plot of the charged Higgs boson production cross section (a) and average relative uncertainty (b) as a function of $\tan\beta$ and $m_{H^\pm}$ values in the 4FS of the 2DHM [88].

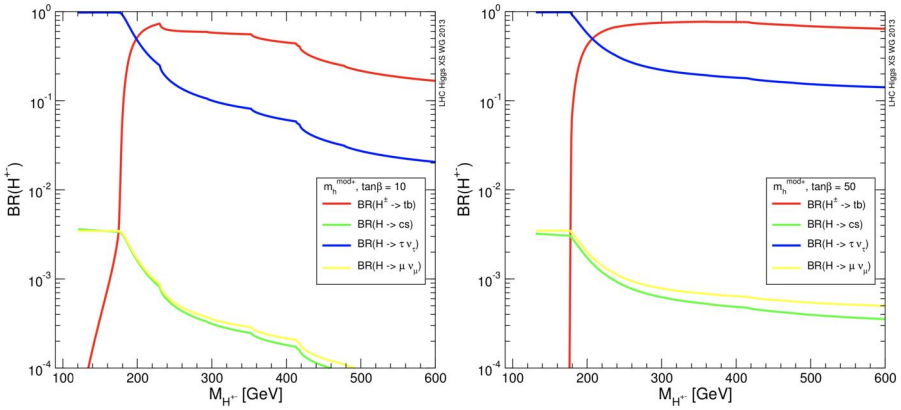


Figure 6.4: Branching fractions of the charged Higgs boson as a function of $m_{H^\pm}$, for $\tan\beta$ values of 10 (left) and 50 (right).

	Decay mode	Branching ratio Γ_i/Γ
Leptonic decays	$\tau^\pm \to e^\pm + \nu_e + \nu_\tau$	17.82%
	$\tau^\pm \to \mu^\pm + \nu_\mu + \nu_\tau$	17.41%
	total leptonic decays	35.23%
Hadronic decays	$\tau^\pm \to \pi^\pm + \nu_\tau$	10.83%
	$\tau^\pm \to \pi^\pm + \pi^0 + \nu_\tau$	25.52%
	$\tau^\pm \to \pi^\pm + 2\pi^0 + \nu_\tau$	9.30%
	$\tau^\pm \to \pi^\pm + 3\pi^0 + \nu_\tau$	1.05%
	total 1-prong decays	46.70%
	$\tau^\pm \to 2\pi^\pm + \pi^\mp + \nu_\tau$	8.99%
	$\tau^\pm \to 2\pi^\pm + \pi^\mp + \pi^0 + \nu_\tau$	2.70%
	total 3-prong decays	11.69%
	other decays	6.38%
	total hadronic decays	64.77%

Table 6.1: Decay channels and their branching ratios for leptonic and hadronic tau decays. The row "others" includes both decays to more than 3 charged hadrons and decays involving at least one kaon.

fraction BR($H^+ \to \tau\nu$) can still be substantial (at least 10 %) depending on the value of $\tan\beta$.

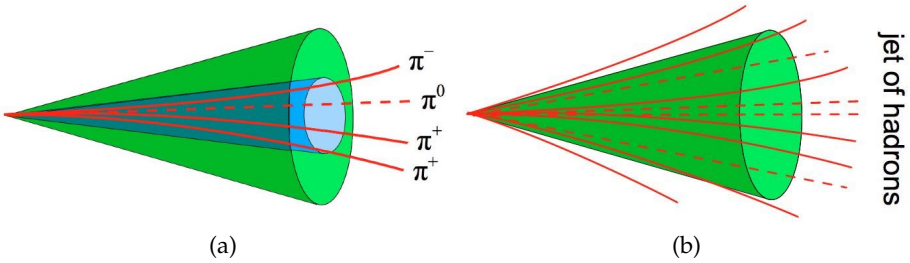
Since the goal of this chapter is to study the improvements in the event selection using a new single-tau trigger chain, only the $H^+ \to \tau\nu$ decay channel will be considered in the following.

6.2 THE HADRONIC TAU TRIGGER

In Table 6.1, the most common tau decay channels, together with their branching ratios, are shown. Because of its high mass, the third generation τ lepton is the only lepton that can decay into either hadrons or another lepton plus neutrinos.

While the leptonic tau trigger selection relies on the identification of the electron and muons produced in the final state, which are indistinguishable from prompt electrons and muons, the identification of the hadronic decaying taus has to face unique challenges. These decay signatures are very similar to that of jets of hadrons, as is shown schematically in Figure 6.5. The extremely large production cross section of multi-jet events make them a major background.

Hadronic tau decays are classified into two different categories, based on the number of charged particles in the final state: *1-prong* and *3-prong* decays. While decays into more than three charged particles are possible, their branching ratio is small and they are hence typically



(a) (b)

Figure 6.5: (a) Sketch of a three-prong τ decay. The decay products clustered in a narrow cone (in blue). (b) Sketch of a jet of hadrons. The particles are distributed throughout a large cone.

not considered for trigger identification purposes. The hadrons in the final state of tau decays are reconstructed as jets of particles.

For the separation of τ from jets, the decay topology is used. The τ signal features 1 or 3 charged particles in a very narrow cone with little or no activity in a surrounding isolation cone, whereas jets typically have activity distributed throughout the isolation cone. Because of this, tracking information is fundamental for the identification process.

6.2.1 *Level-1 tau trigger*

Since no tracking information is available at the first trigger level, the L1 tau trigger selection relies on calorimetric information. This selection is performed by the *L1Calo* trigger system, already presented in Section 1.2.6.2.

For each of the τ candidates, two distinct calorimetric regions are defined, using trigger towers in both the electromagnetic and hadronic calorimeters: the so called *core* and *isolation* regions. The core region is defined as a square of 2×2 trigger towers, corresponding to a 0.2×0.2 area in the $\Delta\eta \times \Delta\phi$ space. The isolation region is, instead, defined as an annulus surrounding the core region, defined by the towers lying between 2×2 and 4×4 around the core region.

The trigger selection relies on two thresholds based on the amount of transverse energy deposited in the two calorimetric regions. For the core region, the E_T of the τ candidate is calculated as the sum of the transverse energy deposited in the two most energetic neighbouring towers of the EM calorimeter, and in all the 4 core HAD cells. For the isolation region, the sum of the transverse energy deposited in all the isolation region cells is considered.

While an energy-independent lower E_T threshold is applied on the core region, in the isolation region an energy-dependent upper E_T threshold, given by the following equation, is applied for energies $E_T < 60\,GeV$:

$$E_T^{EM_{isol}}[GeV] \leqslant \left(\frac{E_T[GeV]}{10} + 2\right) \tag{6.12}$$

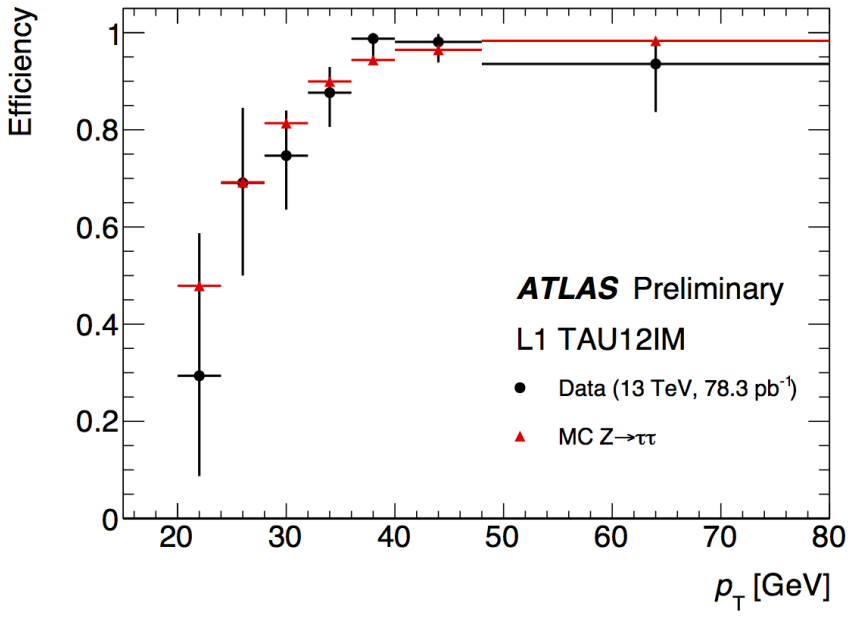


Figure 6.6: The Level-1 tau trigger efficiency measured in data and compared to simulation, with respect to offline reconstructed tau candidates transverse momentum with p_T above 20 GeV. The online tau candidates are reconstructed at L1 of the ATLAS trigger and are required to have a transverse energy of 12 GeV.

For energies $E_T > 60$ GeV, no E_T requirement is applied.

Because all of the calorimetric cells in a trigger tower are combined without the use of sophisticated clustering algorithms and without τ specific energy calibrations, the L1 energy resolution is significantly worse than the offline one. Moreover, the coarse energy and geometrical position granularity limits the precision of the measurement. These effects lead to a significant signal efficiency loss for low-E_T tau candidates, as can be seen from Figure 6.6.

In case of combined $\tau + e/\mu/\tau$ triggers, further reductions in rate are achieved by additional kinematic and geometric selections implemented with the new *L1Topo* system [89]. These algorithms are based on the angular selection $\Delta R = \sqrt{(\Delta\phi)^2 + (\Delta\eta)^2}$ applied to the L1 objects above given E_T thresholds contained in the lists of L1 objects provided by L1Calo. The L1Topo algorithms are used to find either $\tau + e/\mu/\tau$ pairs not back-to-back (with an angular separation $\Delta R < 2.9$) and jet candidates that are not overlapping (with an angular separation $\Delta R > 1$). In Figure 6.7, the efficiency of this new trigger system is presented. The deployment of L1Topo has been completed in 2018.

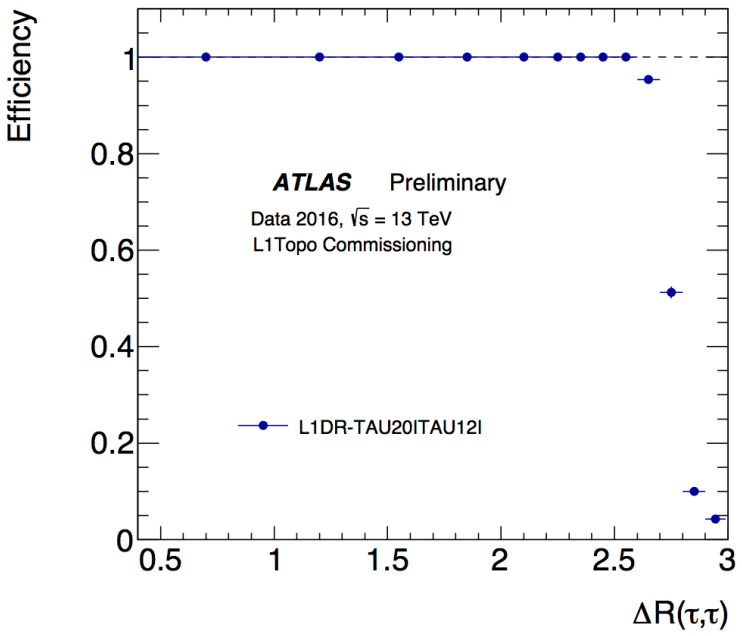


Figure 6.7: Efficiency of the L1Topo selection as function of the $\Delta R(\tau, \tau)$ between offline tau candidates in di-tau events. The efficiency is computed in data events collected in 2016 and selected by a di-tau trigger with the same energy and isolation selections, but no ΔR cut. The L1Topo selection is fully efficient for di-tau events reconstructed offline with $\Delta R < 2.5$.

6.2.2 *HLT tau trigger selection*

The HLT tau trigger selection is executed in three successive steps, each of them applying a different selection for the rate reduction [90]. The order of the steps follows their CPU requirements, with the less CPU consumptive selection executed at the beginning of the HLT, while the most demanding one is executed at the end, when the rate has already been reduced. In order of execution, and thus of computing power required, the different selection steps are: *calo-only preselection*, *track preselection* and *offline-like selection*.

In the following subsections, more details about these steps will be provided.

6.2.2.1 *Calo-only preselection*

In the first HLT step, a quick selection is performed using calorimetric information only. The RoI identified at L1 (the RoI concept has been presented in Section 1.2.6) are used as a seed for the execution of topo-clustering algorithms. Unlike for the L1 selection, data at full detector granularity is retrieved for the HLT selection, and the bunch-by-bunch pile-up corrections are applied. This results in reconstructed topo-clusters energies very close to the offline ones. The energy of the

τ candidate is calculated in a cone of $\Delta R < 0.2$ around the barycentre of the jet seed. A dedicated τ energy calibration (TES) is, also, applied to improve the precision of the energy measurement.

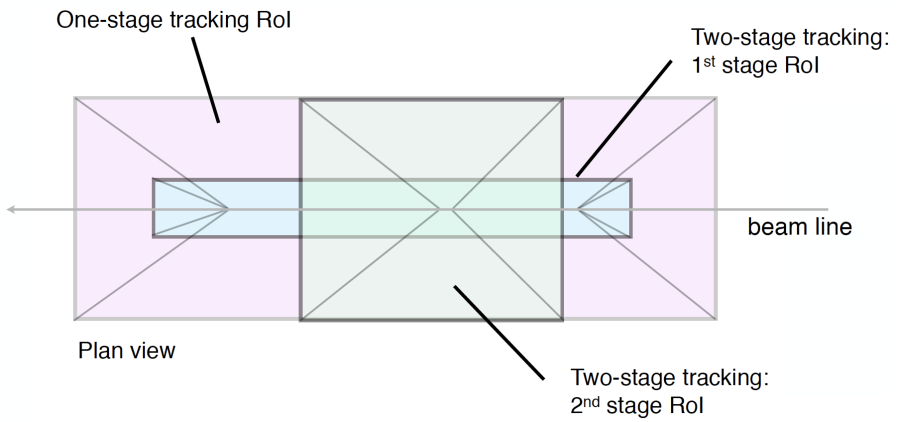
After the calo-only τ reconstruction, a selection on the minimum p_T of the τ candidate is also applied and only the remaining candidates pass to the next stage of the HLT tau trigger.

6.2.2.2 *Track preselection*

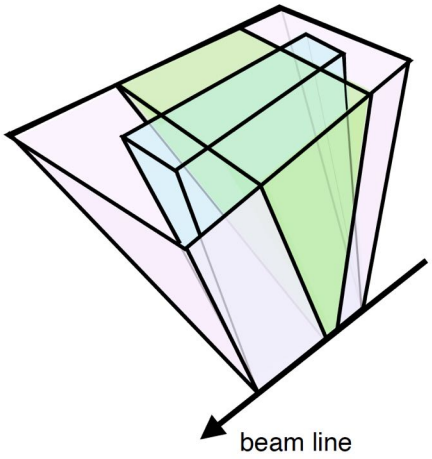
The second HLT stage is based on tracking information, reconstructed into the *two-stage fast tracking*, a trigger-specific pattern recognition algorithm [91]. The algorithm is divided in two steps. First, the leading track (with $p_T > 1\,GeV$) is sought in a narrow $\Delta R < 0.1$ cone around the selected L1 RoI, along the entire beamline. The candidate tau is rejected if no track is found. In the second step, the tracks reconstruction algorithm is executed in a wider cone, $\Delta R < 0.4$ around the RoI center, but only along a small portion of the beamline ($|z| < 10\,mm$ with respect to the leading track). In Figure 6.8, a schematic illustration of the tracking regions is provided.

As can be seen in Figure 6.9, this two-step approach is able to drastically reduce the CPU power required by a single-stage track reconstruction performed in the whole RoI along the whole beamline, as it would be required since the z position of the candidate τ is unknown (the two step tracking overcome this requirement). As can be seen from Figure 6.8, the reconstruction volume of the two-stage tracking is, in fact, sensibly smaller. The configuration of the two-stage fast tracking algorithm in terms of track minimum transverse momentum and RoI size has been optimised to yield the highest tracking efficiency within the available CPU resources.

After the two-stage fast tracking procedure is completed, all the reconstructed tracks with $p_T > 1\,GeV$ are associated to the candidate τ, and the track multiplicity is computed. Two different track multiplicities are considered: the track multiplicity in the *core* region, defined as $\Delta R < 0.2$, and the *isolation* region, defined as $0.2 < \Delta R < 0.4$ (both calculated around the candidate τ direction). The τ candidates that don't satisfy the multiplicity requirements of $1 \leqslant N_{tracks} \leqslant 3$ in the core region and $N_{tracks} \geqslant 1$ in the isolation region are discarded. This track multiplicity requirement is able to achieve enough rate reduction to make possible to run more CPU intensive algorithms on the accepted events, while keeping high efficiency for real τ. However, the track multiplicity requirement has been abandoned for most of the trigger chains during 2017 (e.g. is not used for the single-τ or di-τ chains). This choice was taken because of trigger inefficiencies for the selection of high p_T τs observed at the beginning of Run2, that was primarily due to the requirement on the number of reconstructed tracks at the fast tracking stage.



(a)



(b)

Figure 6.8: Sketch illustrating the RoI from the single-stage and two-stage tau lepton trigger tracking in plan view (x-z plane) along the transverse direction (a) and in perspective view (b).

6.2.2.3 *Offline-like selection*

For all the τ candidates that have passed the track preselection, a more precise computation of their associated tracks is executed, and a final selection is performed through a Boost Decision Tree (BDT) algorithm.

The recomputation of the track parameters is executed using a precision-tracking algorithm, very close to the one used for the offline track reconstruction. In this step, the tracks identified by the two-stage fast tracking algorithm are used as a seed for the new algorithm. Using the new track information, as well as calorimetric information, two sets of variables are computed and used for the BDT identification process. The set of variable to be computed depends on the τ track multiplicity in the core region ($\Delta R < 0.2$), and, thus, to the 1-prong or 3-prong tau decay types. The definition of these variables follows

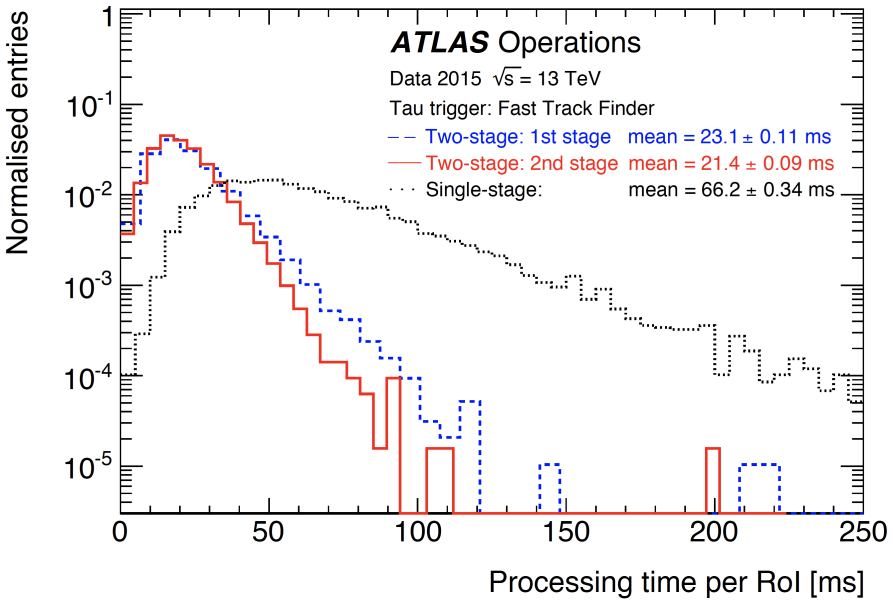


Figure 6.9: The ID trigger tau tracking processing time for the fast-track finder comparing the single-stage and two-stage tracking approach.

Variable	Description	1-prong	multi-prong
f_{cent}	Central energy fraction	•	•
$f^{-1}_{leadtrack}$	Leading track momentum fraction	•	•
R_{track}	Track radius	•	•
$\lvert S_{leadtrack}\rvert$	Leading track d_0 significance	•	
f^{track}_{iso}	Tracks p_T in isolation region	•	
ΔR_{Max}	Maximum ΔR		•
S^{flight}_T	Transverse flight path significance		•
m_{track}	Track mass		•
$f^{track-HAD}_{EM}$	Fraction of EM energy from $\pi^{\pm}$	•	•
f^{EM}_{track}	Ratio of EM energy to track p_T	•	•
$m_{EM+track}$	Track-plus-EM-system mass	•	•
$p_T^{EM+track}/p_T$	Ratio of track-plus-EM-system to p_T	•	•

Table 6.2: Discriminating variables used as input to the tau BDT identification algorithm for 1-prong and multi-prong τ candidates

closely their offline counterparts [92]. The BDT variables are presented in Table 6.2.

All the variables are feed to the BDT, which allows to compute an identification score. In Figure 6.10, the BDT algorithm performances for τ candidates reconstructed offline as either 1-prong and 3-prong and passing both the HLT p_T and track multiplicity requirements are presented. The BDT signal efficiency performances have been evaluated on a simulated sample of $Z \to \tau\tau$ events, while the background rejection on simulated QCD di-jet events. The performance

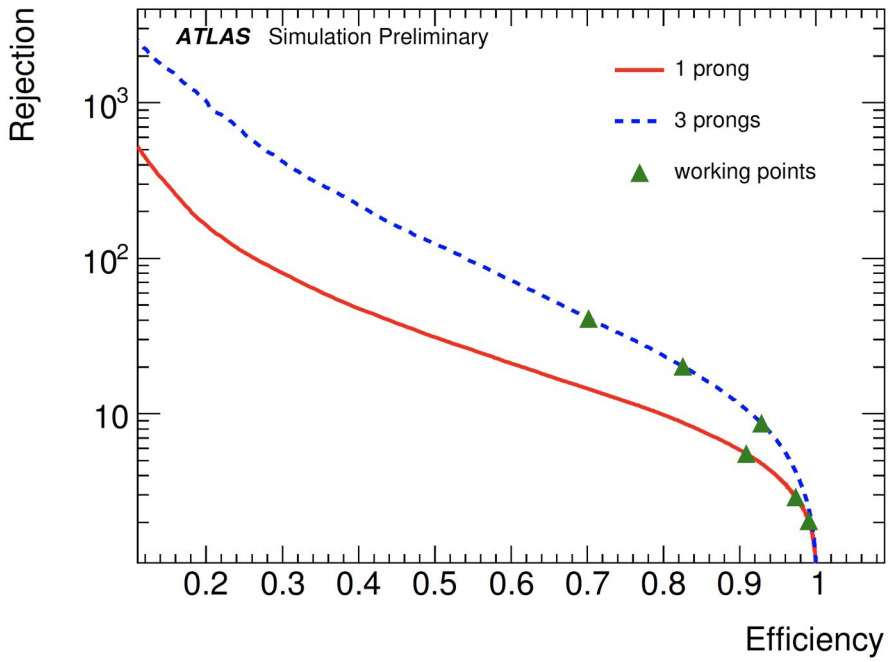


Figure 6.10: Performance of the BDT algorithm in terms of background rejection as a function of the signal efficiency for 1-prong and 3-prong τ candidates reconstructed offline as 1-prong and 3-prong τ and passing the HLT p_T and track multiplicity requirements. The working points correspond to the *tight*, *medium* and *loose* identification criteria in order of decreasing background rejection.

has been evaluated for true τ candidates and also for τ satisfying the offline tau identification criteria, in order to ensure maximal overlap between online and offline identification criteria. Different working points on the BDT identification score have been defined, tuned separately for 1-prong and multi-prong candidates: *loose*, *medium* and *tight*. The working point used in all the tau HLT trigger chains is the medium one, that yields an efficiency of 96 % (82 %) for true 1-prong (3-prong) τ, that are reconstructed offline as 1-prong (3-prong) τ and have passed the HLT p_T and track multiplicity requirements.

6.2.3 *Trigger algorithms timing*

In Table 6.3, the time required to process each of the HLT tau trigger algorithms is presented. As can be seen from the table, the majority of the processing time is spent to execute the fast tracking and precision tracking algorithms. Between the two, the precision tracking requires less time for its execution since it uses as seed the tracks already computed at the previous stage.

The full algorithm sequence requires about 90 ms to run on the ATLAS TDAQ farm, but the full sequence is only executed for the τ candidates that survived all the intermediate steps, as the calo-only

HLT tau trigger step		Mean [ms]	RMS [ms]
Calo-only preselection:	Topo-clustering	7	3
	τ reconstruction	1	0
Track preselection:	First stage fast tracking	32	16
	Second stage fast tracking	27	14
	τ reconstruction	1	0
Offline-like selection:	Precision tracking	21	12
	τ reconstruction and BDT	1	0

Table 6.3: Summary of the execution times of all the steps of the HLT tau trigger as measured in data (collected in October 26^{th} 2016) at an instantaneous luminosity of about $1.2 \times 10^{34}\,cm^{-2}s^{-1}$ and pile-up of 40.

preselection and the track preselection. Moreover, for each event all computed information is cached so that other tau triggers do not need to compute it again.

The selection cuts applied in the different steps have been optimized in order to keep the event rate under control. Since the computational resources required to process the whole selection chain is noticable, an important reduction in rate has to be executed in the first selection steps. Selection thresholds too low would drastically increase the number of candidate taus to be processed, and the HLT computing farm would not be able to process all the events. It has however to be noted that increasing the trigger p_T thresholds may result in loosing access to signatures with soft taus in the final state.

From what said before, it is possible to observe the impact that the use of the FTK system would have on the tau trigger system. Having all the event tracks already available at the beginning of the trigger selection would allow to avoid the execution of the tracking algorithms, reducing the time required for the τ event selection from the current mean 90 ms to only 10 ms, corresponding to a 89 % time and resource utilization reduction. Considering, instead, a more relaxed case in which the precision tracking step would be maintained, in order to provide reconstructed tracks with offline-like quality, the execution time would be of 31 ms, corresponding to a 66 % resource utilization reduction.

6.2.4 *Tau trigger menus*

The ATLAS tau trigger menu is composed of a single-tau high p_T trigger chain, and many combined $\tau + X$ trigger chains, where X stays for an electron, a muon, a second τ, or E_T^{miss}.

Table 6.4 summarizes all the main tau trigger chains, available during 2018. Besides the p_T thresholds, the baseline tau trigger selection includes the L1 isolation requirement and the medium identification

Trigger	Trigger selection (p_T)		Trigger rate	
	L1 (GeV)	HLT (GeV)	L1 (kHz)	HLT (Hz)
Single-τ	100	160	1.3	33.5
di-τ	20-12	25-35	5.7	20.4
$\tau + e$	12-15	25-17	4.2	11.8
$\tau + \mu$	12-10	25-14	2.2	9.9
$\tau + E_T^{miss}$	20-45	35-70	2.8	17.1

Table 6.4: Primary tau triggers used in the 2018 pp data taking period. The trigger rates are reported for an instantaneous luminosity of $2.0 \times 10^{34}\,cm^{-2}s^{-1}$.

criterion, as described in the previous subsection. The requirements on the track multiplicities have been, instead, disabled in the second part of the Run2 data taking period. Moreover, due to L1 rate limitations, the combined triggers $\tau + e/\mu/\tau$ and $\tau + E_T^{miss}$ require the presence of an additional jet candidate at L1 with transverse momentum above 25 and 20 GeV, respectively. No jet requirement is applied at the HLT because no additional rate reduction is needed. It is important to note that the p_T thresholds are chosen as a trade-off among the maximization of the events selection and the available bandwidth and CPU resources available.

6.3 FTK-BASED SINGLE-TAU TRIGGER CHAIN

As presented in the previous section, two main aspects contribute to the choice of the tau trigger thresholds. The first is, obviously, the bandwidth limit. The HLT has to be able to reduce the output event rate to a sustainable value, making it possible to store the selected events for offline analysis. The amount of bandwidth available for a given trigger chain is the outcome of a trade-off between the different signature trigger groups, with the goal of splitting the available bandwidth between all the different signatures, in order to maximize the physics results.

The second limit comes from the computational resources available. As presented in Subsection 6.2.3, the hadronic tau HLT selection is very demanding in terms of computational resources. If the p_T selection thresholds would be decreased too much, too many events would require to be processed by all the HLT selection steps, and the available HLT CPU farm would not be able to cope with them.

In this section, a preliminary study aiming to the creation of a new single-tau trigger chain, able to exploit the FTK peculiarities in order to overcome this second limit is presented.

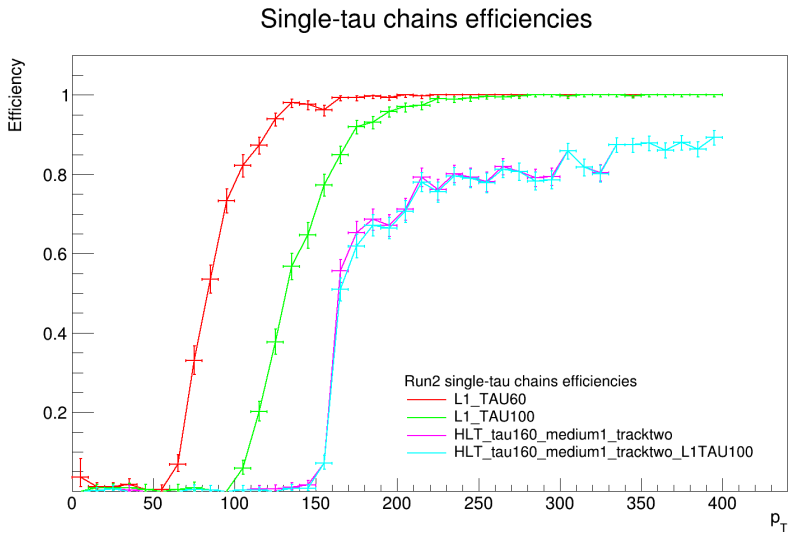


Figure 6.11: Efficiency of the Run2 L1 and HLT single-tau trigger chains as a function of the offline tau p_T. The efficiencies have been calculated on a sample of simulated $\tau\overline{\tau}$ events.

The FTK system can be exploited at the HLT level in many different ways: it can provide vertex reconstruction, improve the pile-up suppression, improve the MET reconstruction, and can free HLT resources. Since specific approaches are already under studying, in this thesis work I explored the possibility of creating a new single-tau trigger chain that would increase the signal acceptance for low-p_T taus by lowering the HLT thresholds, while keeping the chain output rate at an acceptable level.

This new chain uses, for the event selection, track multiplicity requirements based on the FTK produced tracks, removing any need for track computation at the HLT level. Thanks to this, the computational requirements are drastically reduced. Exploiting the freed resources, the HLT farm can be used to process more events by lowering the trigger thresholds and, thus, allowing for the selection of low p_T taus that would not be accessible otherwise.

6.3.1 *Run2 single-tau trigger chains*

Two different single-τ trigger chains have been used during the Run2 data taking period: *HLT_tau160_medium1_tracktwo* for the 2015-2016 period, and *HLT_tau160_medium1_tracktwo_L1TAU100* for the 2017-2018 one. In Figure 6.11, the L1 and HLT trigger efficiencies of these two chains, calculated on a $\tau\overline{\tau}$ simulated sample (Drell-Yan production), are presented.

The main difference of the two chains relies in the change of the L1 p_T threshold. This threshold was raised from 60 GeV to 100 GeV. The choice has been taken in order to free some of the L1 bandwidth,

that was not used at the HLT level. As can be seen from the plot, the increase of the L1 threshold doesn't results in any significant reduction of the HLT efficiency.

Since the goal of the trigger chain under study is the reduction of the HLT thresholds, also the L1 one have to be kept as low as possible. For this reason, in the following the lower L1 trigger chain, with a p_T cut of 60 GeV, will be considered. Since it was used until the half of Run2, it is assumed in the following that lowering the L1 threshold to this level is sustainable for the system.

6.3.2 *New FTK single-tau trigger chain*

The starting point of the new-FTK based tau trigger chain is the reduction of the HLT p_T thresholds. The use of a L1 trigger chain with thresholds at 60 GeV, as presented in the previous subsection, allows for a reduction of the HLT ones down to 80 GeV. Two different new trigger chains have been created, exploiting the same trigger selection requirements (that will be described in the following), but with a different tau p_T thresholds: $p_T > 80$ GeV and $p_T > 100$ GeV.

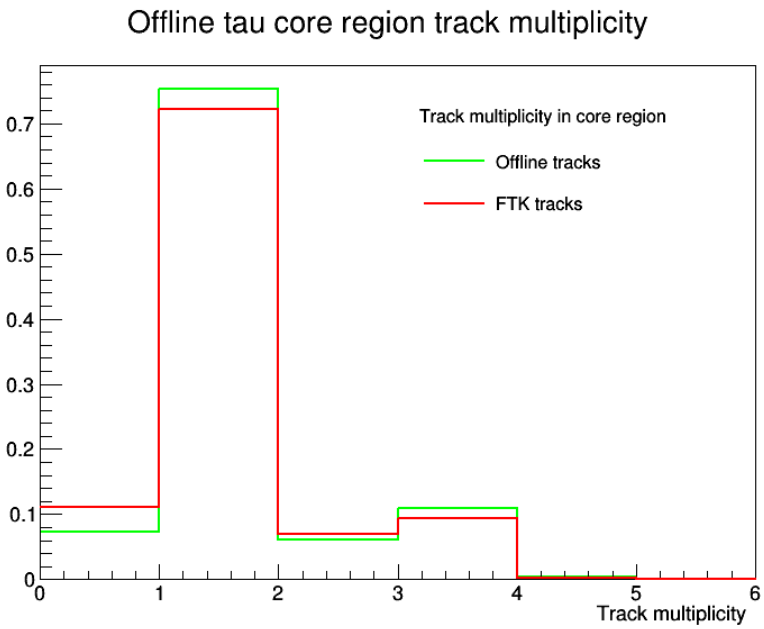
The lowering of the τ candidate p_T threshold increases the number of candidate taus to be processed by the HLT farm. In order to maintain a sustainable rate at the HLT level, while leaving enough computing resources to process all the events, a fast selection based on track multiplicity requirements, using the FTK provided tracks, has been implemented for the new chain. For this scope, two different space regions have been defined around the τ direction: a core and an isolation region. The core region has been defined as $\Delta R < 0.1$ with respect to the τ direction, while the isolation region is defined as $0.1 < \Delta R < 0.4$.

Only tracks within $\Delta R < 0.4$ are considered. The tracks are associated to the given tau candidate if they pass the following requirements:

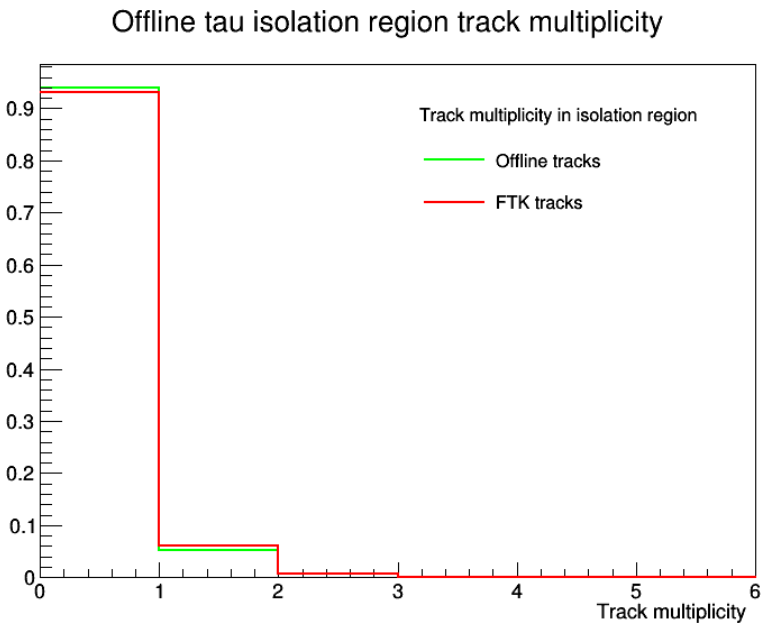
- $p_T > 2$ GeV;
- $|d_0| < 2$ mm;
- $|z_0| < 100$ mm;
- the tracks in the τ cone are accepted if the z coordinate of their vertex is less than 2 mm far away from the leading track.

Moreover, the tracks are required to lay in the detector acceptance. Only tracks with $\eta < 2.5$ or not lying between $1.37 < |\eta| < 1.52$ are considered for the trigger selection. After all the tracks have been associated to the candidate tau, the track multiplicity of the two regions is computed.

In order to emulate FTK tracks, the FTK fast simulation tool, presented in Section 3.4.1, has been used. This tool is fed for each event

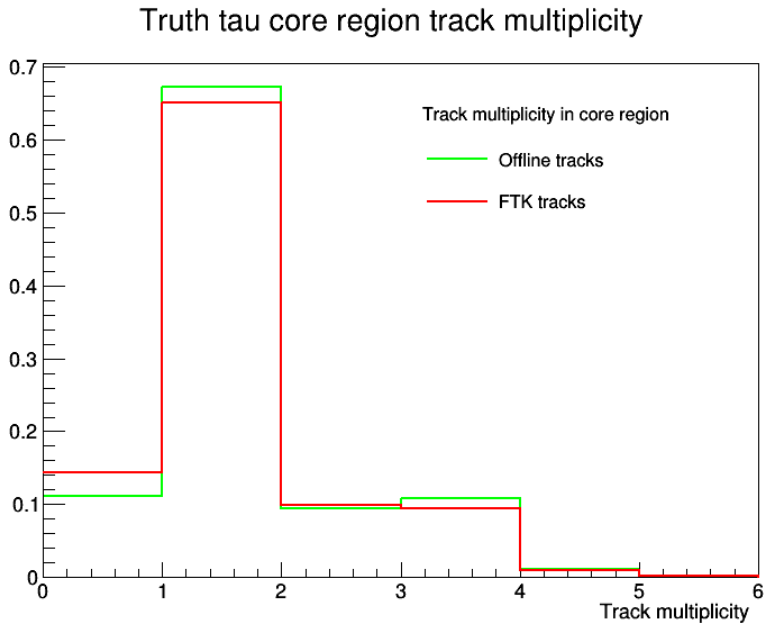


(a)

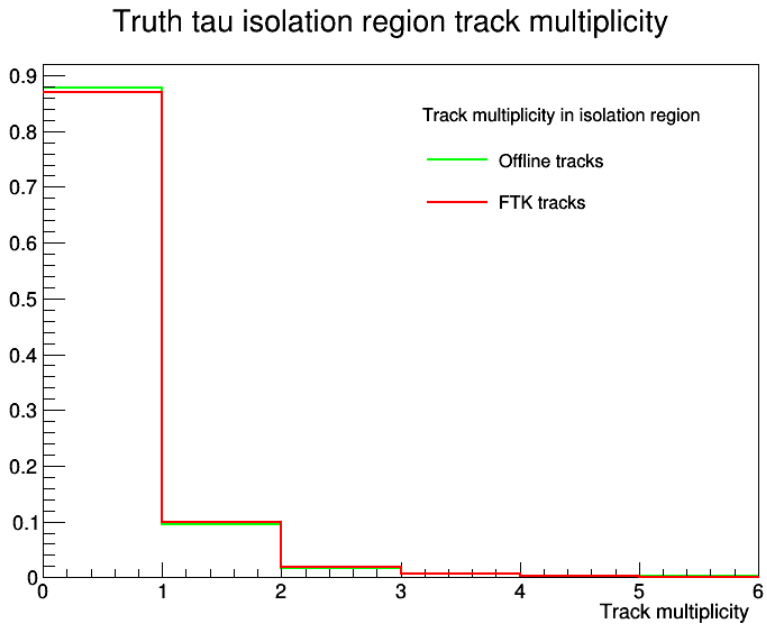


(b)

Figure 6.12: Offline tau track multiplicity in the core (a) and in the isolation (b) regions for FTK and offline tracks, calculated from $\tau\bar{\tau}$ MC samples.



(a)



(b)

Figure 6.13: Truth tau track multiplicity in the core (a) and in the isolation (b) regions for FTK and offline tracks, calculated from $\tau\overline{\tau}$ MC samples.

with all the Inner Detector offline tracks, it deletes some of them (emulating the FTK track reconstruction efficiency), and smears the remaining ones, providing a set of tracks with the same efficiency and quality as the ones expected by FTK.

In Figure 6.12, the track multiplicity calculated using FTK and offline tracks for the candidate taus of a simulated sample of $\tau\overline{\tau}$ (through Drell-Yan production), is presented. Only taus with $p_T > 20\,GeV$, laying in the η detector acceptance and passing an offline medium BDT requirement have been considered. As it can be seen from the plots, very good agreement has been found for the track multiplicities inside the isolation region. Considering the core region, the number of candidate taus with no tracks associated is slightly bigger for the FTK case. This is due to the FTK track reconstruction efficiency that is slightly worse than the offline one, resulting in the missing of some of the tracks that should be associated to the candidate tau. As will be presented later, this issue has been overcame by not requiring any lower limit on the core track multiplicity for the trigger selection.

A further check has been performed by studying the truth tau (instead of the offline ones), available in the MC samples used. In Figure 6.13, the same two plots are presented, but computing the FTK and offline track multiplicity on truth tau only. As can be seen from the plots, the results are similar.

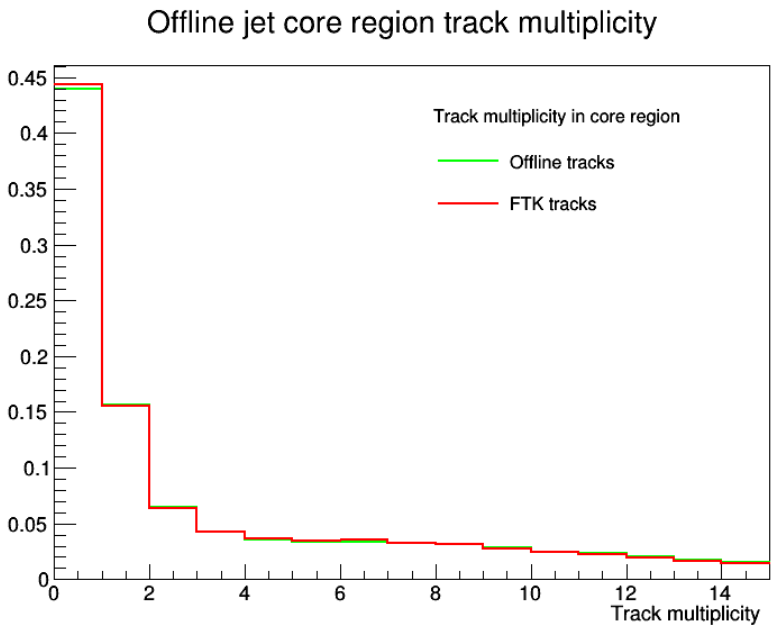
Finally, in Figure 6.14 the same track multiplicity is presented for QCD jets, the major background for the tau trigger selection. The plots have been produced running on a simulated sample of di-jets events. Also in this case, a good agreement has been found between FTK and offline tracks.

In order to maximize the new chain background rejection and the signal selection efficiency, the following requirements have been defined:

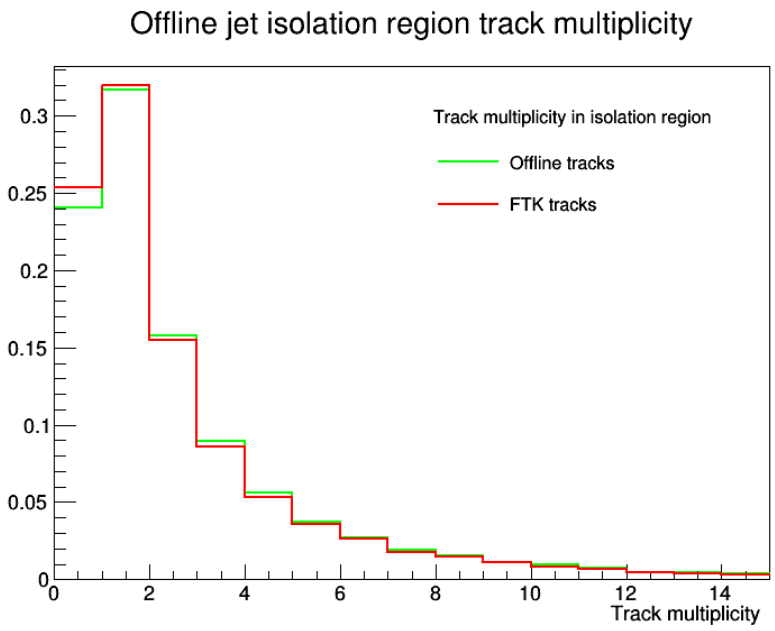
- track multiplicity in core region: $0 \leqslant N_{tracks} < 4$;
- track multiplicity in isolation region: $0 \leqslant N_{tracks} < 2$.

To further improve the background rejection, a second selection has been added to the new trigger chain, based on the BDT score and executed only on the tau candidate passing the track multiplicity requirements. Since no HLT information is available in the MC samples, the offline BDT loose criteria has been used for this purpose. This choice is motivated by two reasons.

First, the offline loose BDT criteria shows a background rejection similar to the one of the online BDT medium criteria (the working point to be used for the online BDT selection). As a confirmation of this, a comparison study of the two selection criteria has been performed. In this comparison, the number of events containing at least one online medium candidate tau or an offline loose candidate tau have been counted. The study has been performed running on



(a)



(b)

Figure 6.14: Offline jet track multiplicity in the cone (a) and in the isolation (b) regions for FTK and offline tracks, calculated from di-jets MC samples.

Sample	HLT medium criteria	Offline loose criteria
di-jet	12	12
$H^+ \to \tau\nu$ (200 GeV)	2794	2839
$H^+ \to \tau\nu$ (500 GeV)	6206	6330
Enhanced bias	103	108

Table 6.5: Number of events containing at least one τ candidate passing the online medium BDT criteria and the offline loose BDT criteria, over 10 000 processed events for each sample. The study has been performed using on simulated di-jet and $H^+ \to \tau\nu$ samples, as well as a sample of enhanced bias data.

simulated di-jet and $H^+ \to \tau\nu$ samples, and on an enhanced bias data sample (Run number 360 026, see Section 6.3.4). More details about the simulated samples used will be given in Section 6.4.2. As can be seen from Table 6.5, the number of events selected by the two requirements is in agreement within 5 %.

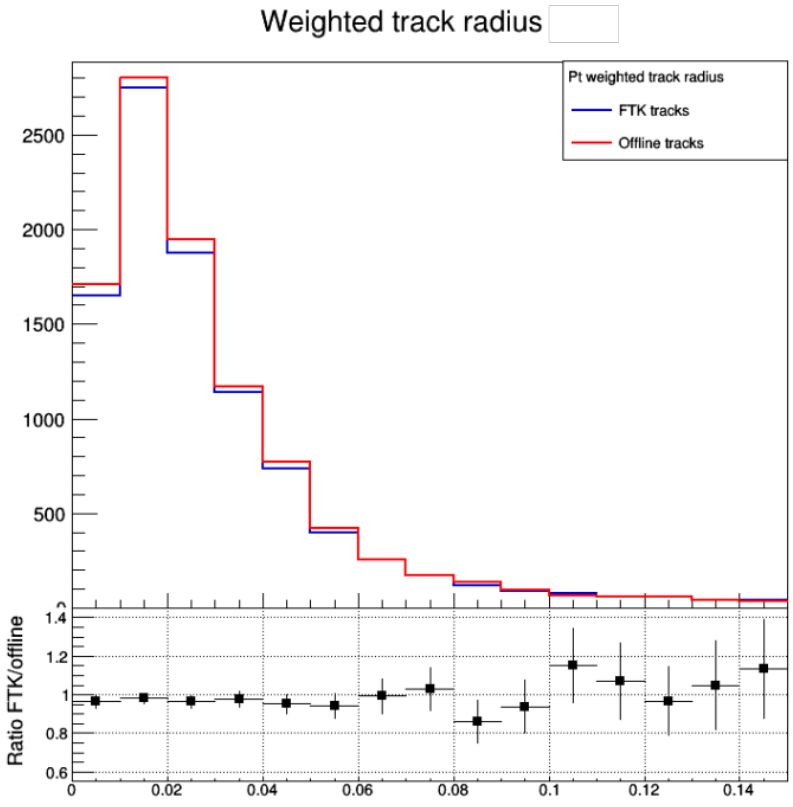
The second reason for using offline BDT score relies on the very good agreement between the tracking related offline BDT variables and the same variables computed from FTK tracks. As an example of this, the distributions of two of the variables used as a seed for the BDT selection, computed from offline and FTK tracks, are shown in Figure 6.15. As can be seen, the two variables distributions are in good agreement.

Despite the fact that the offline BDT loose selection criteria is a first order approximation of the HLT medium one, the creation of a new BDT optimized for the FTK track usage is required. This remains one of the open points from this thesis work, that will be addressed in the future. Still, for a preliminary study of the performances of this new trigger chain, the use of the offline BDT score is a good approximation.

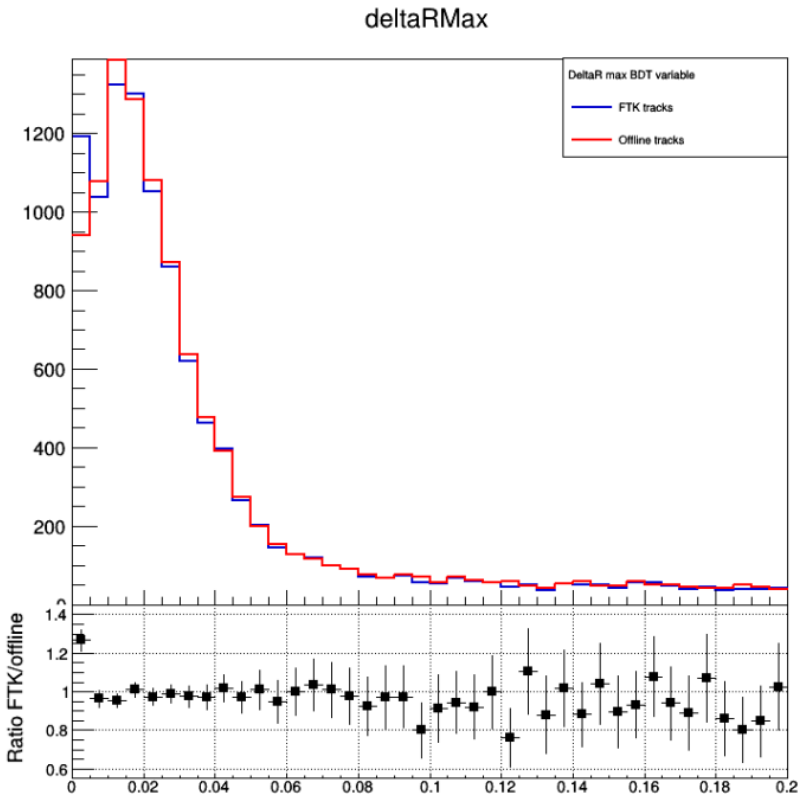
6.3.3 *Chain efficiency study*

A study of the selection efficiencies of the new FTK based trigger chains made use of a sample of simulated $\tau^+\tau^-$ (through Drell-Yan production) events, the same sample used for the study of the candidate tau track multiplicity. In Figure 6.16, the trigger efficiency of the new FTK-based trigger chain with tau p_T threshold at 80 GeV (*HLT_ftktau80_L1TAU60*) and at 100 GeV (*HLT_ftktau100_L1TAU60*), together with the L1 and the standard single-tau HLT efficiencies, are presented.

As can be seen from the plot, the new chains present an efficiency of about 80 %, which in the plateaux region is very close to the one of the standard HLT chain. The large improvement of the two FTK based chains relies in the selection of low p_T taus, given by the lowering of the p_T thresholds. This can lead to significant improvements in the



(a)



(b)

Figure 6.15: Distribution of the f_{iso}^{track} (a) and of the ΔR_{Max} (b) BDT variables computed with offline and FTK tracks.

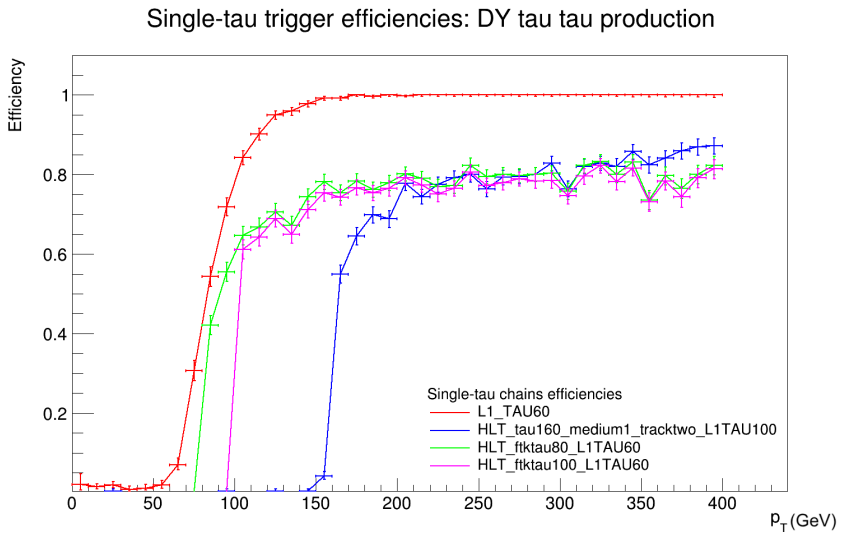


Figure 6.16: Selection efficiency of the two FTK based single-tau trigger chains, together with the efficiencies of the L1 and HLT single-tau standard chains, as a function of the offline tau p_T. The efficiencies have been calculated on a sample of simulated $\tau\bar{\tau}$ events.

trigger acceptance for a multitude of interesting physics searches. One of these will be presented in Section 6.4.

6.3.4 *Chain rate study*

To complete the characterization of the new trigger chains, a study meant to provide an estimate of the chain output rate have been performed.

The rate of events passed by any trigger selection is calculable from a sample of data recorded with zero trigger bias (for example, random trigger) [93]. However, most selections of particular interest at a hadron collider, for example high p_T leptons, come from processes whose cross sections are much smaller than the total inelastic cross section, meaning that an unfeasibly large data sample would be required. Enhanced bias data samples, on the other hand, are characterized by overweight in high p_T events which are likely to be selected by the trigger. The data are collected using a variety of L1 triggers of all signature types, combinations and p_T ranges to produce a compact dataset which has the statistical power to assess the trigger rate of algorithmic selections on the data of the type typically performed by the HLT. Enhanced bias data are taken with an invertible trigger menu such that a single weight is calculable per event which corrects for the prescales applied during the enhanced bias data taking and restores an effective zero bias spectrum.

Trigger chain	Events triggered
HLT_tau80_medium1_tracktwo_L1TAU60	3010
HLT_tau160_mediumRNN_tracktwoMVA_L1TAU100	1390
HLT_ftktau80_L1TAU60	1040
HLT_ftktau160_L1TAU60	610

Table 6.6: Table summarizing the number of events triggered by the two HLT reference chains and the two new FTK trigger chains, running on 100 000 events of an enhanced bias data sample.

The rate of the new FTK based chains has been estimated simulating their trigger selection on a sample of these enhanced bias data. For the rate computation, two different trigger chains have been taken as references: the HLT chains *HLT_tau80_medium1_tracktwo_L1TAU60* and *HLT_tau160_mediumRNN_tracktwoMVA_L1TAU100*. These two chains have a measured rate, at the time of the data taking, of 95 Hz and 44 Hz, respectively.

Table 6.6 shows the number of the events triggered running on the enhanced bias data, for the two reference chains and the two FTK ones. The trigger rate obtained for the two new chains is of 33 Hz and 19 Hz respectively. These rates are compatible with the current HLT single-tau trigger chain rate, presented in Section 6.2.4, that is of about 33 Hz. Since both the proposed trigger chains present reasonable rates, only the chain with the lower p_T threshold will be considered in the rest of the chapter.

6.4 CASE STUDY: $H^+ \to \tau\nu$ CHANNEL SELECTION IMPROVEMENTS

In the previous section, the study of a new FTK based τ trigger chain has been presented. The chain showed very good efficiency, comparable to the ones of the current single-tau HLT trigger chains, while allowing the selection of taus with p_T down to 80 GeV, maintaining at the same time a good background rejection and an acceptable output rate. The use of such a HLT trigger chain can led to an improvement in signal acceptance in physics searches where soft τ are foreseen in the final state.

Since the new chain performs a single-tau selection, we decided to study its application to the search for a charged Higgs boson in the $H^\pm \to \tau^\pm_{had}\nu$ decay channel. A search for charged Higgs boson produced either in top-quark decays or in association with a top-quark, and subsequently decaying via $H^\pm \to \tau^\pm\nu$, has been already performed using 36.1 fb^{-1} proton-proton collision data recorded with the ATLAS detector at $\sqrt{s} = 13$ TeV [94]. In this chapter the increase in signal acceptance that the integration of the new single-τ trigger

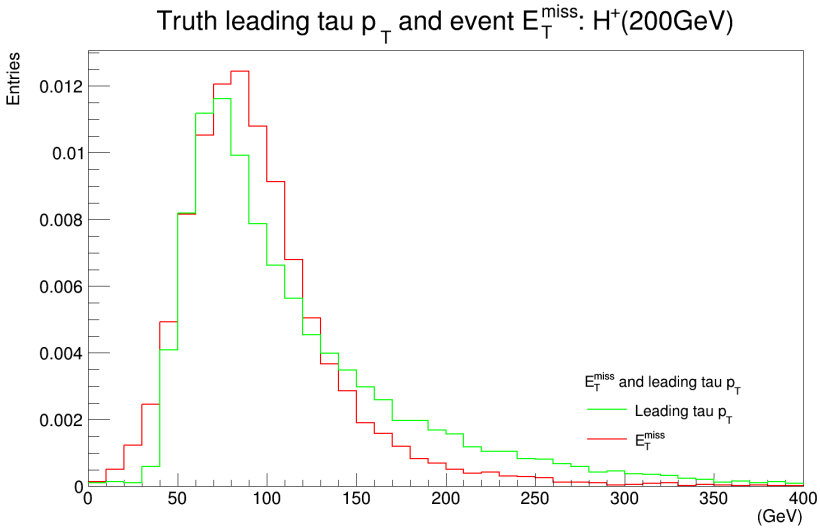


Figure 6.17: Distribution of the offline missing transverse energy and offline leading tau p_T for charged Higgs boson (200 GeV) events in the $H^+ \to \tau^+_{had}\nu$ decay channel.

chain would provide, to both the online and offline event selection, will be assessed.

6.4.1 *Trigger strategy*

In the original analysis, two different trigger strategies were considered, depending on whether the top-quark produced with the H^+ decays hadronically or semileptonically. In the τ_{had}+lepton case, the analysis relies on events accepted by single electron/muon triggers. In the τ_{had}+jets channel, the analysis is based on events accepted by a single E_T^{miss} trigger. Only this last case will be considered in the following.

The E_T^{miss} trigger used for the online event selection was based on thresholds of 70, 90 or 110 GeV, depending on the data-taking period considered, and thereby, accounting for different pile-up conditions. Since the new FTK based trigger chain is proposed for the Run3 LHC period, and since the future pile-up conditions will be comparable or worse than the Run2 ones, the last Run2 period MET threshold is taken as reference for the following considerations.

In Figure 6.17, the MET and leading tau p_T distributions, for a charged Higgs boson (200 GeV) decaying via the $H^+ \to \tau^+_{had}\nu$ channel, is presented. From the plot it is possible to see that less than half of the events lay inside the MET trigger acceptance region. The introduction of a second trigger selection based on the single-tau trigger chain was originally not considered, due to the high p_T thresholds of

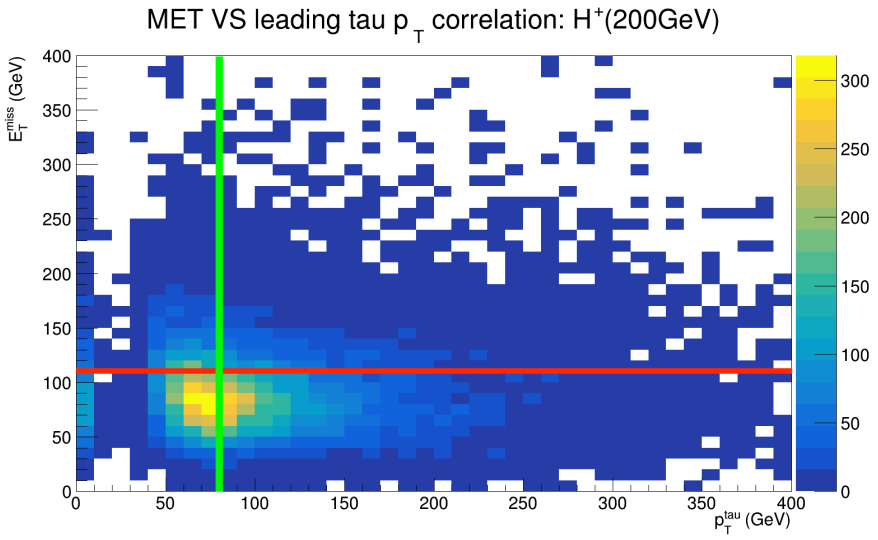


Figure 6.18: Leading tau p_T distribution vs missing transverse energy for the decay of a charged Higgs boson of 200 GeV in the $H^+ \to \tau^+_{had}\nu$ decay channel. The red line represents the 110 GeV MET threshold of the E_T^{miss} trigger chain used in the original analysis, while the green line represent the 80 GeV tau p_T threshold of the FTK based single-tau trigger chain.

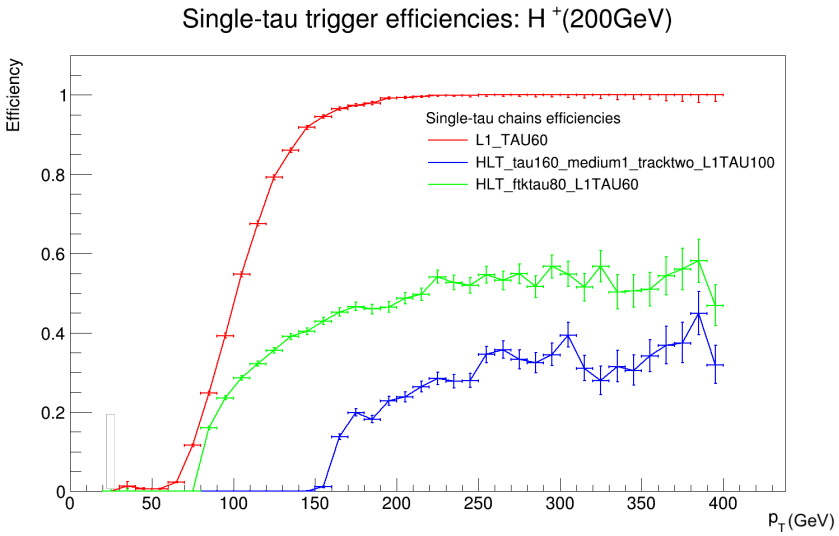
this trigger chain (160 GeV). In this case, as can be seen, only the tail of the τ distribution would lay in the trigger p_T acceptance.

Moving to a much lower single-tau p_t threshold, as in the case of the new FTK based trigger chain, the introduction of a composite events selection, based on the combination (logical OR) of the E_T^{miss} and of the low p_T single-tau trigger chains, can significantly improve the event acceptance.

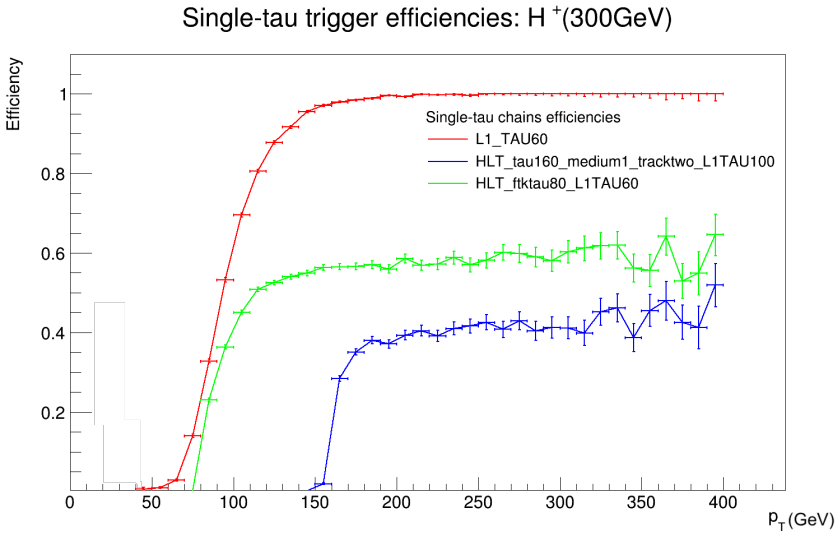
In Figure 6.18, the 2D correlation of the leading tau p_T vs the event MET is shown, for an H^+ with mass of 200 GeV. As can be seen from the plot, the bulk of the signal lays at about 80 GeV for both the MET and leading tau p_T. With the current selection, only the events appearing in the plot over the red line (representing the 110 GeV MET threshold) are within the trigger acceptance, and no significance changes to the bulk acceptance would be present with the introduction of a high p_T single-tau selection. A notable improvement is, instead, possible by introducing the FTK single-tau chain: only the events laying inside the region delimited by the red and green lines would in this case be discarded.

The trigger efficiency in the selection of signal events for the new FTK based trigger chain has been studied. Simulated samples of $H^+ \to \tau_{had}\nu$ events have been used. For the sake of concision, only the results obtained running on H^+ samples of masses 200 GeV, 300 GeV, 400 GeV and 1000 GeV are shown.

In Figure 6.19 and in Figure 6.20, the trigger efficiencies of the FTK-based single tau trigger chain is presented, together with the L1



(a)

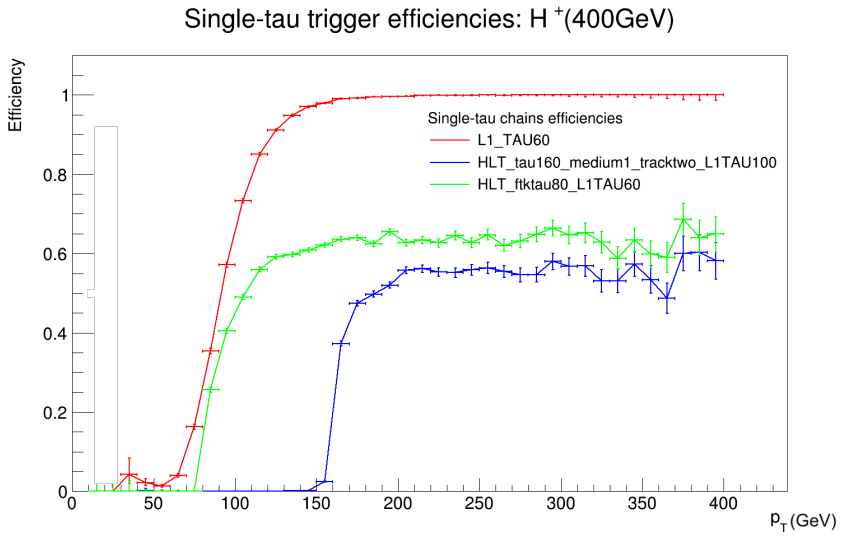


(b)

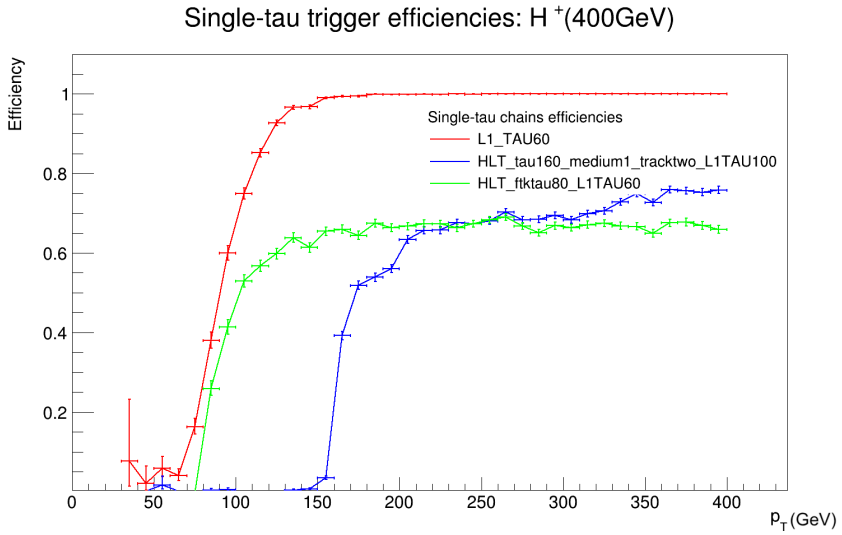
Figure 6.19: Distribution of the offline leading taus p_T for samples of simulated $H^+ \to \tau_{had}\nu$ events, for a H^+ mass of 200 GeV (a) and 300 GeV (b).

and standard single-tau ones, for a sample of 200 GeV (Figure 6.19a), 300 GeV (Figure 6.19b), 400 GeV (Figure 6.20a) and 1000 GeV (Figure 6.20b) H^+. Thanks to the HLT p_T threshold reduction, the efficiency of the new chain is notably better, especially for lower H^+ masses. Moreover, it is possible to note that the selection efficiency of all the chains deteriorate with the lowering of the H^+ mass. This is expected, and due to the fact that some of the candidate taus exit the p_T selection range of the chains.

Interesting is the fact that the FTK chains show higher efficiencies with respect to the standard chains for high p_T taus (in particular in



(a)



(b)

Figure 6.20: Distribution of the offline leading taus p_T for samples of simulated $H^+ \to \tau_{had}\nu$ events, for a H^+ mass of 400 GeV (a) and 1000 GeV (b).

Figure 6.19a). This gain in efficiency of the new FTK chain is due to two main reasons.

First, the p_T value entering the plots is the one of the leading offline tau, independently from the fact that this tau is the one that triggered the given chain or not.

Moreover, the leading tau may be outside of the detector acceptance, or it is not selected by the trigger, due to its efficiency. Because of this, some of the events may trigger on the subleading tau coming from the tau decay of the W from the top quark which is procudced alngside the H^+, which present a softer spectrum and, thus, is more common

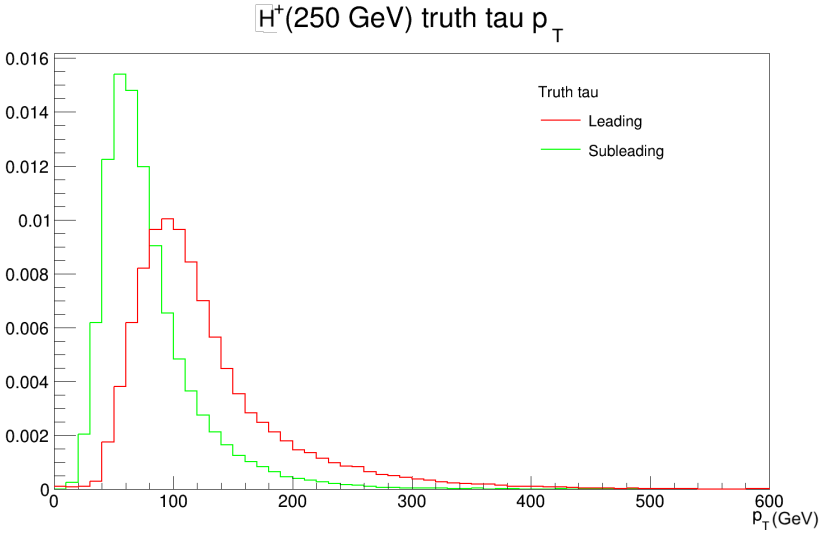


Figure 6.21: Distribution of the truth leading and subleading taus p_T for samples of simulated $H^+ \to \tau_{had}\nu$ events, with an H^+ of mass 250 GeV.

to be triggered by the lower p_T trigger chains. As a support to this, in Figure 6.21, the p_T distribution of the leading and subleading taus, for the H^+ 250 GeV $H^+ \to \tau_{had}\nu$ sample are shown.

These two contributions can be separated by studying the trigger efficiency for the events that triggered the given trigger chain on the leading tau only. In Figure 6.22, the trigger chains efficiency for the selection of the leading event tau is presented. As can be seen, over the given chain p_T threshold all the chains present a similar efficiency. The correct trigger efficiency for the $H^+ \to \tau_{had}\nu$ signal, however, is the one presented in Figure 6.19.

6.4.2 *Samples of simulated events*

Since the proposed trigger chain was developed after the end of Run2 and, therefore, was not included in any data-taking session, the study has been performed on simulated event samples only.

The signal events with $H^+ \to \tau\nu$ used in this study were generated with MadGraph5_aMC@NLO [95] in two distinct mass regions:

- in the top-quark mass region (160–180 GeV), the full process $pp \to H^+Wbb$ was generated at LO;
- in the mass range above the top-quark mass (200–2000 GeV), H^+ production in association with a top-quark was simulated at next-to-leading order (NLO).

The interference between the Feynman diagrams is taken into account in the top-quark mass region, where it is most relevant. In all cases, the matrix-element generator was interfaced to Pythia v8.186 [96] for the

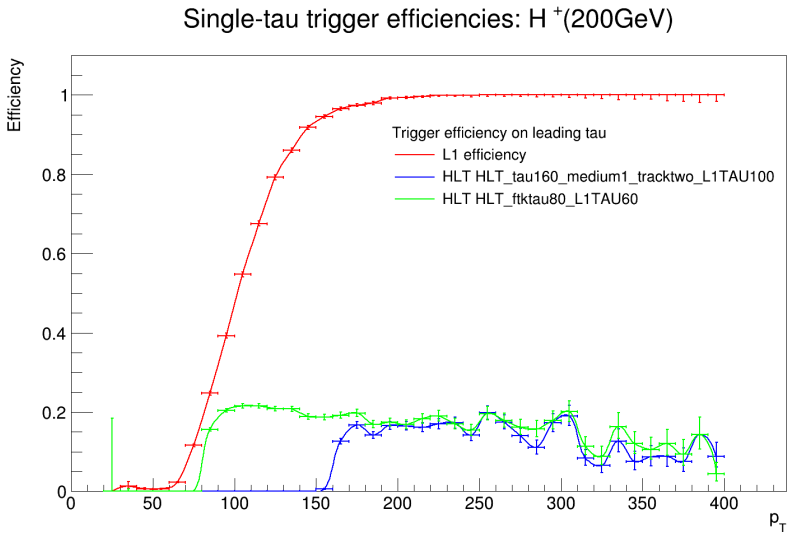


Figure 6.22: Selection efficiency of the FTK based single-tau trigger chain, together with the efficiencies of the L1 and HLT single-tau standard chains, as a function of the offline leading tau p_T. The efficiencies have been calculated on a sample of simulated $\tau\bar{\tau}$ events.

simulation of the parton shower and the underlying event, with the A14 [97] set of tuned parameters (tune). The NNPDF2.3 [98] parton distribution function (PDF) sets were used for the matrix-element generation and the parton shower.

The SM background processes include the production of $t\bar{t}$ pairs, single top-quarks, W+jets, Z/γ^*+jets and electroweak gauge-boson pairs ($WW/WZ/ZZ$), as well as multi-jet events. The $t\bar{t}$ events constitute the main background in the lower mass H^+ searches, while multi-jet events dominate for large charged Higgs boson masses. All backgrounds arising from a quark- or gluon-initiated jet misidentified as a hadronically decaying τ-lepton are estimated with a data-driven method.

The generation of $t\bar{t}$ events used the Powheg-Box v2 [99][100][101] generator, with the CT10 [102] PDF set in the matrix-element calculations. Single-top-quark events were generated in the Wt-, t- and s-channels using the Powheg-Box v1 generator, with the CT10 PDF set in the Wt- and s-channels or the fixed 4-flavour CT10f4 [103] PDF set in the t-channel. For all processes above, top-quark spin correlations are preserved (MadSpin [104] was used for top-quark decays in the t-channel). The parton shower and the underlying event were simulated using Pythia v6.428 [105] with the CTEQ6L1 [106] PDF set and the corresponding Perugia 2012 [107] tune. The top-quark mass was set to 172.5 GeV.

The sample of $t\bar{t}$ events is normalised to the next-to-next-to-leading-order (NNLO) cross-section, including soft-gluon resummation to next-to-next-to-leading-logarithm (NNLL) order [108]. The normali-

sation of the sample of single-top-quark events uses an approximate calculation at NLO in QCD for the s- and t-channels [109][110] and an NLO+NNLL calculation for the *W*t-channel [111].

Events containing either a W or Z boson with associated jets were simulated using Sherpa v2.2.1 [112] together with the NNPDF3.0NNLO [113] PDF set. Matrix elements were calculated for up to two partons at NLO and four partons at LO using Comix [114] and OpenLoops [115], and they were merged with the Sherpa parton shower [116] according to the ME+PS@NLO prescription [117]. The *W*/Z+jets events are normalised to the NNLO cross-sections calculated using FEWZ [118]. Diboson processes (*WW*, *WZ* and ZZ) were simulated at NLO using the Powheg-Box v2 generator, interfaced to the Pythia v8.186 parton shower model. The CT10nlo PDF set was used for the hard-scatter process, while the CTEQL1 PDF set was used for the parton shower. The non-perturbative effects were modelled using the AZNLO tune. EvtGen v1.2.0 [119] was used for the properties of bottom- and charm-hadron decays, except in samples generated with Sherpa. All simulated events were overlaid with additional minimum-bias events generated with Pythia v8.186 using the A2 [120] tune and the MSTW2008LO [121] PDF set to simulate the effect of multiple pp collisions per bunch crossing (pile-up). Simulated events were then weighted to have the same distribution of the number of collisions per bunch crossing as the data. All signal and background events were processed through a simulation [122] of the detector geometry and response based on Geant4 [123] and they are reconstructed using the same algorithms as the data.

6.4.3 *Offline object reconstruction and selection*

For the events that passed the two trigger chains a further event selection is performed, based on requirements applied on the offline objects. While some of the requirements are independent from the trigger chain that fired the event, cuts on the MET or on the tau p_T depend on that.

The following requirements are applied:

- at least one τ candidate, satisfying the medium BDT identification criteria;
- no electron or muons with E_T or p_T above 20 GeV;
- at least three jets with $p_T > 25$ GeV, of which at least one b tagged.

To ensure compatibility with the trigger selection, two additional cuts on the τ p_T and on the MET are performed, depending on which chain fired the given event:

- if the single-tau chain is the only one fired:

  - at least one candidate tau with the above requirements, satisfying $p_T > 100\,GeV$
  - $E_T^{miss} > 40\,GeV$
- if the MET chain is the only one fired:
  - at least one candidate tau with the above requirements, satisfying $p_T > 40\,GeV$
  - $E_T^{miss} > 150\,GeV$
- if both the chains are fired:
  - at least one candidate tau with the above requirements, satisfying $p_T > 40\,GeV$
  - $E_T^{miss} > 40\,GeV$

As presented above, most of the physics objects reconstructed in ATLAS are used for the selection: charged leptons (electrons and muons), jets (including those compatible with the hadronisation of b-quarks or hadronic decays of τ-leptons) and missing transverse momentum. In the following, more details about the reconstruction process of these abjects are provided.

The electrons are reconstructed offline by matching energy deposits that have been clustered in the electromagnetic calorimeter to a track reconstructed in the ID. They are required to have $p_T > 20\,GeV$. Moreover, they are required to lay in $|\eta| < 2.47$ and to not lay in the transition region between the barrel and end-cap calorimeters ($1.37 < |\eta| < 1.52$). In addition to this, they are required to pass a loose identification criteria, based on the offline BDT output.

Muons are reconstructed by matching an ID track with a track in the muon spectrometer. Moreover, they are required to have a $p_T > 20\,GeV$ and $|\eta| < 2.5$. Finally, a loose identification criteria is required also for the muon case.

In order to ensure that both electrons and muons originate from the primary vertex, i.e. the vertex with the highest sum of the p_T^2 of its associated tracks, their tracks are required to have a longitudinal impact parameter of $|z_0 \sin\theta| < 0.5\,mm$, and a transverse impact parameter significance of $|d_0/\sigma(d_0)| < 5$ and $|d_0/\sigma(d_0)| < 3$ for electrons and muons, respectively. Moreover, in order to reduce contamination by leptons from hadron decays or photon conversion, two different isolation requirements are applied: a calorimetric and a track isolation requirement. The calorimetric based requirement checks for energy deposits in the calorimeters, in a cone of $\Delta R < 0.2$ around the electron or muon candidate. The track isolation uses instead a variable cone size, starting from $\Delta R < 0.2$ or $\Delta R < 0.3$ (electrons and muons, respectively), that decreases with the increase of the particle p_T.

The offline jet reconstruction starts from the search for energy deposits in the calorimeters. The reconstruction is performed using the

anti-k_t algorithm [124], implemented in the FastJet package [125]. A radius parameter value of $R = 0.4$ is used. Moreover, pileup and energy corrections, together with energy and η-dependent calibrations, are performed. Jets are required to have $p_T > 20\,GeV$ and $|\eta| < 2.5$. A multivariate technique (jet vertex tagger) is applied to jets with $p_T < 60\,GeV$ and $|\eta| < 2.4$. This technique, through the use of tracking and vertexing information, allows the identification and selection of jets originating from the hard-scatter interaction [126]. For the identification of jets containing b-hadrons, a multivariate discriminant technique is, again, used. It is based on the impact parameter information and the identification of secondary and tertiary vertices within the jet [127]. A single threshold value in the discriminant output range identifies the working point of the algorithm, that for this study is 70 %. It corresponds to a rejection factors of 13 , 56 and 380 against c-jets, hadronic τ decays and jets from light quarks or gluons, respectively.

The offline reconstruction of hadronically decaying τ leptons uses as a seed the anti-k_t jets showing a deposited transverse energy $E_T > 10\,GeV$ and that presents 1 or 3 associated tracks in the inner detector. These tracks have to lay withing a cone of $\Delta R < 0.2$ around the axis of the visible decay products of the candidate object. The candidate τ_{had} are required to have a visible transverse momentum $p_T > 30\,GeV$ and to lay withing $|\eta| < 2.3$. Also in this case, the barrel/end-cap transition region ($1.37 < |\eta| < 1.52$) is excluded. A final BDT selection, similar to the one used in the online selection, is performed in order to separate the candidate τ_{had} from QCD jets. The medium working point has been chosen, which provides a reconstruction and identification efficiency of 55 % (40 %) for 1-prong (3-prong) hadronic τ decays in $Z \to \tau\tau$ events, and leading to a rejection factor of about 50–200 for jets. Moreover, in order to reduce the number of electrons misidentified as τ_{had} candidates, a likelihood-based veto is used.

The magnitude of the missing transverse momentum, E_T^{miss} [128], is reconstructed from the negative vector sum of transverse momenta of reconstructed and fully calibrated objects, with an additional term that is calculated from ID tracks that are matched to the primary vertex and not associated with any of the selected objects.

6.4.4 *Results*

In order to study the variations in signal acceptance and the possible improvements in the signal statistical significance between the use of the two new trigger chains, the analysis has been performed with 3 different trigger setups:

- MET trigger selection;
- single-tau trigger selection;

H^+ mass	MET	Single-tau	Combo	Gain
170 GeV	1888	774	2125	14 %
200 GeV	1832	718	2087	13.9 %
225 GeV	1768	728	2017	14 %
250 GeV	1693	803	1944	14.8 %
275 GeV	1606	819	1836	14.3 %
300 GeV	1499	796	1697	13.2 %
350 GeV	1239	739	1380	11.4 %
400 GeV	994	627	1095	10.1 %
500 GeV	616	427	662	7.6 %
1000 GeV	52	37	53	2.2 %

Table 6.7: Expected event yields after trigger selection for an hypothetical H^+ signal, for 36.1 fb^{-1} of simulated data, and yields gain for the two trigger chains combination over the use of the MET chain only.

- a combinations of the two (MET OR single-tau: Combo).

In Table 6.7, the number of events selected in each of the three cases, for different values of the H^+ mass, are presented. The inclusion of the new single-tau trigger chain in the event selection, in OR with the original MET chain, brings an increment of about 14 % in the number of events selected at trigger level, for H^+ masses up to about 300 GeV.

In order to provide an estimate of the final event yields after the offline selection, the complete analysis selection has also been performed. The event yields for different hypothetical H^+ signals, for all the cumulative selection steps, is presented in Table 6.8 for the use of the MET chain only and in Table 6.9 for the events selected by the combination of the MET and single-tau trigger chains. In Table 6.10, the yields gain variations are presented. As can be seen, the gain in the total number of events selected is even more notable than the gain in the triggered events, reaching an about 55 % of gain at $m_{H^+} \approx 275$ GeV. This variation is due to two main factors. The first is the inclusion in the event selection of new events that were not selected by the MET trigger chain, direct consequence from the variation presented in Table 6.7. The second contribution comes from the reduction of the offline MET selection thresholds for events triggered by both the trigger chains, as presented in the subsection 6.4.3. For the events triggered by the MET chain only, the offline cut on the E_T^{miss} was required to be higher than the trigger cut, in order to make compatible the offline selection with the trigger plateaux. Introducing a second trigger chain, this is not anymore the case, and the cut has, then, been reduced.

Finally, in order to study the gain in statistical significance, the behaviour of the analysis selection has been studied on samples of

H^+ mass	Trigger	Tau cut	Lepton cut	MET cut	Jet cut
170 GeV	1888	1112	913	496	374
200 GeV	1832	1022	867	562	398
225 GeV	1768	976	824	535	372
250 GeV	1693	941	804	529	365
275 GeV	1606	892	757	508	354
300 GeV	1499	823	696	475	328
350 GeV	1239	691	579	415	288
400 GeV	994	546	453	343	236
500 GeV	616	363	298	236	162
1000 GeV	52	28	22	23	15

Table 6.8: Expected event yields after cumulative selection cuts for an hypothetical H^+ signal, for 36.1 fb^{-1} of simulated data selected by the MET trigger chain.

H^+ mass	Trigger	Tau cut	Lepton cut	MET cut	Jet cut
170 GeV	2152	1319	1071	790	592
200 GeV	2087	1215	1020	819	576
225 GeV	2017	1166	975	810	559
250 GeV	1944	1137	961	827	567
275 GeV	1836	1070	901	801	551
300 GeV	1697	977	823	743	504
350 GeV	1380	801	668	622	427
400 GeV	1095	626	518	494	338
500 GeV	662	400	328	318	218
1000 GeV	53	31	24	24	17

Table 6.9: Expected event yields after cumulative selection cuts for an hypothetical H^+ signal, for 36.1 fb^{-1} of simulated data, selected by the combination of the MET trigger chain and the new FTK-based single-tau one.

H^+ mass	Total yelds gain
170 GeV	58.4 %
200 GeV	44.8 %
225 GeV	50.2 %
250 GeV	55.2 %
275 GeV	55.6 %
300 GeV	54.0 %
350 GeV	48.0 %
400 GeV	43.3 %
500 GeV	34.0 %
1000 GeV	8.7 %

Table 6.10: Expected event yields gain after cumulative selection cuts for an hypothetical H^+ signal, for 36.1 fb^{-1} of simulated data. The gain refers to the variation in the number of events passing all the selection criteria, comparing the case of the use of the MET trigger chain only or the combination of the MET and FTK-based single-tau chain.

Background process	MET chain	Combo chain
$W(\mu\nu) + \text{Jets}$	0	0
$W(\tau\nu) + \text{Jets}$	0	0
Single top (t-channel)	92	173
Single top (s-channel)	1403	2550
Single top (Wt-channel)	2899	5435
$W(\tau\nu)$	0	895
WW	15	15
WZ	0	13
$Z/\gamma^*(ee) + \text{Jets}$	0	0
$Z/\gamma^*(\mu\mu) + \text{Jets}$	0	0
$Z/\gamma^*(\tau\tau) + \text{Jets}$	0	0
ZZ	0	4
$t\bar{t}$	47 866	92 313
Total	52 275	101 398

Table 6.11: Expected event yields after cumulative selection cuts for the main signal backgrounds, for 36.1 fb^{-1} of simulated data.

H^+ mass	MET chain	Combo chain	Variation
170 GeV	1.63	1.86	13.7 %
200 GeV	1.74	1.81	4.0 %
225 GeV	1.63	1.75	7.8 %
250 GeV	1.60	1.78	11.4 %
275 GeV	1.55	1.73	11.7 %
300 GeV	1.43	1.58	10.5 %
350 GeV	1.26	1.34	6.3 %
400 GeV	1.03	1.06	2.9 %
500 GeV	0.71	0.68	-3.8 %
1000 GeV	0.07	0.05	-21.9 %

Table 6.12: Expected statistical significance for an hypothetical H^+ signal for 36.1 fb^{-1} of simulated data, and significance variation for the two trigger chains combination over the use of the MET chain only.

the main signal backgrounds. For this study, the same selection criteria used in the signal selection have been applied. The statistical significance has been simply calculated as $\frac{S}{\sqrt{B}}$, ignoring the systematic uncertainties on the background estimation. In order to effectively extract the signal, the original analysis made use of a BDT. The use of such a tool exceed the goal of my analysis. This excercise has been performed with an explorative function, and it is supposed to serve as input for a more detailed study that has to be executed by the physics groups.

In Table 6.11, the event yields for the selection of background events is presented. As can be seen from the table, an increase of background events has been introduced, but it is under control. In order to confirm this, the differences in statistical significance of the signal has been studied. In Table 6.12, the statistical significance for the two cases is shown, together with its variation. Introducing the new combined trigger selection, a gain of more than 11 % is visible, for H^+ masses around 275 GeV. A loss of statistical significance is, instead, visible for very high H^+ masses. It has however to be noted that these high mass regions are not the focus of this work (the selection cuts have not been optimized), and that the studied $H^+ \to \tau\nu$ channel is not anymore the prominent H^+ decay channel in that mass regions, as presented in Section 6.1.2.

CONCLUSIONS

This thesis presented a description and an updated review of the commissioning status of the new FTK processor. This system was developed to provide ATLAS with the possibility of decoupling the ID online track reconstruction from the HLT tasks. The motivations that moved ATLAS in the development of such a system, and that are pushing (or have pushed) the other LHC experiments to adopt solutions that are alternative to the standard CPU-based tracking algorithms, are investigated as well.

After an introduction and an overall description of the FTK project, the original part of the thesis covers the whole spectrum of commissioning activities.

The studies performed on the cooling system demonstrated the capability of this infrastructure to keep the FTK hardware temperature under control. Specifically, all the VME crate temperatures, measured in realistic running conditions, were largely under the operational limits. This avoided the requirement of a board density reduction in the VME crates. Moreover, a few optimizations to the cooling infrastructure have been identified. They are able to further decrease the overall temperatures and to possibly increase the lifetimes of both the project mechanical and electronic components.

The FTK online SW was developed with the goal to integrate the FTK system in the ATLAS framework while respecting all its requirements and exploiting as much use as possible of the already available tools. The studies presented in this thesis show that we achieved this goal. The online software is currently in an operational state, with the focus of the work pointed to its optimization.

Lastly, the study of a new single-tau trigger chain showed how the use of the FTK system would lead to an improvement of the trigger performances and, thus, of the overall discovery potential of the ATLAS experiment. Specifically, the gain in events acceptance for the search of an hypothetical charged Higgs boson, in the decay channel $H^+ \to \tau\nu$, has been presented. It has been demonstrated that the use of this new chain, with a reduced p_T threshold, would bring an increase of more than 10 % in both trigger event acceptance and signal statistical significance. The use of such a trigger chain would be possible only if tracking information would be available at an early HLT stage, as the case for the FTK tracks.

The next steps toward the completion of the project commissioning would have included the finalization of the hardware installation, the establishment of the dataflow with a scaled up system, and the integration of the FTK data in the HLT online decisions. This last

step would have been executed in two separate phases: firstly, using FTK to tag the events in passthrough, using the information only for offline validation; secondly, using its information for online selection, starting with ancillary functionality as the computation of the primary vertices, and lastly for the full event selection. Moreover, further trigger chain studies would have been required, for many different trigger signatures, in order to fully exploit the FTK potential.

Unfortunately, due to funding and person power reasons, the project has been closed. Concerning the hardware infrastructure, the online SW and the trigger improvements, this thesis demonstrates that the finalization of the FTK commissioning was reachable, and that the project goals were achievable. The completion of this project would have led to a notable improvement of the ATLAS discovery potential.

ACKNOWLEDGMENTS

I want to acknowledge the University of Pavia, the INFN and the CERN TDAQ group for each funding a different period of my PhD, allowing me to spend almost the whole last three years in Geneva, working in loco on my research activities. In particular, I really have to thank the Pavia ATLAS group, and especially the group leader Dr. Gabriella Gaudio for all the effort made in order to make this happen. This experience really changed my life.

A very big thanks goes also to my referees, Prof. Alessandro Cardini and Prof. Fabrizio Palla, who kindly accepted to review this thesis and who provided very valuable comments, despite the little time I gave them to read it.

Moving to the non institutional acknowledgments, I firstly want to express my sincere gratitude to my supervisor, Prof. Andrea Negri, for his guidance all over these years (that, to be honest, extends beyond the period of my PhD). Thank you very much for all your patience, for all the help and advice, and for all the hidden work that, even being not visible, a supervisor has to carry out (and I know it has been a lot!).

Again, a super big thanks has to go to Dr. Gabriella Gaudio for all the help, the advice and the nights spent reading this thesis. Without your organization and your constant pressure this thesis would still be at the second chapter.

I want to thank Dr. Nicoletta Garelli for all the support, the help, and especially the advice (in particular, personal) you gave me. All the chats we had in R1 have been so helpful. I'll always be glad that I had the possibility to work and meet a person like you.

I want to give a big thanks to Dr. Giacomo Polesello, Prof. Chiara Roda, Dr. Alberto Annovi, Dr. Paolo Mastrandrea and Dr. Monica Verducci for all the help they gave me, especially concerning the trigger studies part of this thesis. Your help has been fundamental to the success of this work.

A huge thanks goes to all the FTK people. Words can not describe how much I learned from all of you, and how much I enjoyed spending time with you all during the last three years. Working with all of you has been an honour. It is very sad that had to finish in this way.

Between them, a very special thanks has to go to Lesya Horyn. Thank you so much for all the fun, for the tons of English questions you answered, and of course for all the inspirational cute animals you sent me whenever the writing was becoming too hard. I hope to be for you as supportive as you have been for me, in the very close future.

Of course I want to send a very big thanks to Eng. Marco Pasquale Polenghi. The endless discussions about the sense of life, as well as about our interminable list of misadventures, will never be forgotten. I'm very glad to have found such a good friend. Living with you has been memorable, I'm very happy it ended.

I also want to thank all the other people who shared the last three years with me, the people of Pavia University and INFN and the other CERN people who helped to make the last three years so unforgettable. Thanks to all of you.

Finally, a big thanks has to go to my family. In the last three years we saw each other less often, but your love never became less. Thank you very much for everything.

BIBLIOGRAPHY

[1] ATLAS Collaboration. "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC." In: *Physics Letters B* 716.1 (2012), pp. 1–29. ISSN: 0370-2693. DOI: https://doi.org/10.1016/j.physletb.2012.08.020. URL: http://www.sciencedirect.com/science/article/pii/S037026931200857X.

[2] CMS Collaboration. "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC." In: *Physics Letters B* 716.1 (2012), pp. 30–61. ISSN: 0370-2693. DOI: https://doi.org/10.1016/j.physletb.2012.08.021. URL: http://www.sciencedirect.com/science/article/pii/S0370269312008581.

[3] Rudiger Voss and Amos Breskin, eds. *The CERN Large Hadron Collider, accelerator and experiments*. 2009. URL: http://www-spires.fnal.gov/spires/find/books/www?cl=QC787.P73C37::2009.

[4] K. Aamodt et al. "The ALICE experiment at the CERN LHC." In: *JINST* 3 (2008), S08002. DOI: 10.1088/1748-0221/3/08/S08002.

[5] G. Aad et al. "The ATLAS Experiment at the CERN Large Hadron Collider." In: *JINST* 3 (2008), S08003. DOI: 10.1088/1748-0221/3/08/S08003.

[6] S. Chatrchyan et al. "The CMS Experiment at the CERN LHC." In: *JINST* 3 (2008), S08004. DOI: 10.1088/1748-0221/3/08/S08004.

[7] A. Augusto Alves Jr. et al. "The LHCb Detector at the LHC." In: *JINST* 3 (2008), S08005. DOI: 10.1088/1748-0221/3/08/S08005.

[8] O. Adriani et al. "The LHCf detector at the CERN Large Hadron Collider." In: *JINST* 3 (2008), S08006. DOI: 10.1088/1748-0221/3/08/S08006.

[9] G. Anelli et al. "The TOTEM experiment at the CERN Large Hadron Collider." In: *JINST* 3 (2008), S08007. DOI: 10.1088/1748-0221/3/08/S08007.

[10] Albert De Roeck and James L. Pinfold. "The MoEDAL experiment at the LHC." In: *PoS* LHCP2016 (2016), p. 099. DOI: 10.22323/1.276.0099.

[11] "ATLAS magnet system: Technical design report." In: (1997).

[12] "ATLAS central solenoid: Technical design report." In: (1997).

[13] "ATLAS barrel toroid: Technical design report." In: (1997).

[14] "ATLAS inner detector: Technical design report. Vol. 1." In: (1997).

[15] M. Capeans, G. Darbo, K. Einsweiler, M. Elsing, T. Flick, M. Garcia-Sciveres, C. Gemme, H. Pernegger, O. Rohne, and R. Vuillermet. "ATLAS Insertable B-Layer Technical Design Report." In: (2010).

[16] Bartosz Mindur. "ATLAS Transition Radiation Tracker (TRT): Straw tubes for tracking and particle identification at the Large Hadron Collider." In: *Nucl. Instrum. Meth.* A845 (2017), pp. 257–261. DOI: 10.1016/j.nima.2016.04.026.

[17] G. Aad et al. "Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics." In: (2009). arXiv: 0901.0512 [hep-ex].

[18] *ATLAS liquid-argon calorimeter: Technical Design Report*. Technical Design Report ATLAS. Geneva: CERN, 1996. URL: https://cds.cern.ch/record/331061.

[19] The ATLAS Collaboration. "Electron performance measurements with the ATLAS detector using the 2010 LHC proton-proton collision data." In: *The European Physical Journal C* 72.3 (Mar. 2012), p. 1909. ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-012-1909-1. URL: https://doi.org/10.1140/epjc/s10052-012-1909-1.

[20] *ATLAS tile calorimeter: Technical Design Report*. Technical Design Report ATLAS. Geneva: CERN, 1996. URL: https://cds.cern.ch/record/331062.

[21] *ATLAS muon spectrometer: Technical Design Report*. Technical Design Report ATLAS. Geneva: CERN, 1997. URL: https://cds.cern.ch/record/331068.

[22] *ATLAS level-1 trigger: Technical Design Report*. Technical Design Report ATLAS. Geneva: CERN, 1998. URL: https://cds.cern.ch/record/381429.

[23] Peter Jenni, Marzio Nessi, Markus Nordberg, and Kenway Smith. *ATLAS high-level trigger, data-acquisition and controls: Technical Design Report*. Technical Design Report ATLAS. Geneva: CERN, 2003. URL: https://cds.cern.ch/record/616089.

[24] S. Ask et al. "The ATLAS central level-1 trigger logic and TTC system." In: *JINST* 3 (2008), P08002. DOI: 10.1088/1748-0221/3/08/P08002.

[25] M Shochet, L Tompkins, V Cavaliere, P Giannetti, A Annovi, and G Volpi. *Fast TracKer (FTK) Technical Design Report*. Tech. rep. CERN-LHCC-2013-007. ATLAS-TDR-021. June 2013. URL: https://cds.cern.ch/record/1552953.

[26] N Garelli, M T Morar, and W Vandelli. *Balancing the resources of the High Level Trigger farm of the ATLAS experiment*. Tech. rep. ATL-DAQ-PROC-2012-033. Geneva: CERN, June 2012. URL: https://cds.cern.ch/record/1458515.

[27] R. Fruhwirth and R. K. Bock. "Data analysis techniques for high-energy physics experiments." In: *Camb. Monogr. Part. Phys. Nucl. Phys. Cosmol.* 11 (2000). Ed. by H. Grote, D. Notz, and M. Regler, pp. 1–434.

[28] Allam Shehata Hassanein, Sherien Mohammad, Mohamed Sameer, and Mohammad Ehab Ragab. "A Survey on Hough Transform, Theory, Techniques and Applications." In: *CoRR* abs/1502.02160 (2015). arXiv: 1502.02160. URL: http://arxiv.org/abs/1502.02160.

[29] R. Fruhwirth. "Application of Kalman filtering to track and vertex fitting." In: *Nucl. Instrum. Meth.* A262 (1987), pp. 444–450. DOI: 10.1016/0168-9002(87)90887-4.

[30] R Frühwirth. "Track fitting with non-Gaussian noise." In: *Computer Physics Communications* 100.1 (1997), pp. 1–16. ISSN: 0010-4655. DOI: https://doi.org/10.1016/S0010-4655(96)00155-5. URL: http://www.sciencedirect.com/science/article/pii/S0010465596001555.

[31] T Cornelissen, M Elsing, S Fleischmann, W Liebig, E Moyse, and A Salzburger. *Concepts, Design and Implementation of the ATLAS New Tracking (NEWT)*. Tech. rep. ATL-SOFT-PUB-2007-007. ATL-COM-SOFT-2007-002. Geneva: CERN, Mar. 2007. URL: https://cds.cern.ch/record/1020106.

[32] The ATLAS collaboration. "A neural network clustering algorithm for the ATLAS silicon pixel detector." In: *Journal of Instrumentation* 9.09 (Sept. 2014), P09009–P09009. DOI: 10.1088/1748-0221/9/09/p09009. URL: https://doi.org/10.1088%2F1748-0221%2F9%2F09%2Fp09009.

[33] Giuseppe Cerati, Peter Elmer, Steven Lantz, Kevin McDermott, Dan Riley, Matevz Tadel, Peter Wittich, Frank Wuerthwein, and Avi Yagil. "Kalman Filter Tracking on Parallel Architectures." In: *Journal of Physics: Conference Series* 664 (May 2015). DOI: 10.1088/1742-6596/664/7/072008.

[34] Apollinari G., Béjar Alonso I., Brüning O., Fessia P., Lamont M., Rossi L., and Tavian L. *High-Luminosity Large Hadron Collider (HL-LHC): Technical Design Report V. 0.1*. CERN Yellow Reports: Monographs. Geneva: CERN, 2017. DOI: 10.23731/CYRM-2017-004. URL: https://cds.cern.ch/record/2284929.

[35] P. R. S. Swaroop and Neelam Sharma. "An Overview of Various Template Matching Methodologies in Image Processing." In: 2016.

[36] W. Ashmanskas et al. "Silicon vertex tracker: A Fast precise tracking trigger for CDF." In: *Nucl. Instrum. Meth.* A447 (2000), pp. 218–222. DOI: 10.1016/S0168-9002(00)00190-X.

[37] D. Amidei et al. "The CDF trigger." In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 265.1 (1988), pp. 326–335. ISSN: 0168-9002. DOI: https://doi.org/10.1016/0168-9002(88)91087-X. URL: http://www.sciencedirect.com/science/article/pii/016890028891087X.

[38] David Rohr, Mikolaj Krzewicki, and Volker Lindenstruth. "Fast online reconstruction and online calibration in the ALICE High Level Trigger." In: *2016 IEEE-NPSS Real Time Conference (RT)* (2016), pp. 1–3.

[39] David Rohr, Sergey Gorbunov, and Marten Ole Schmidt. *GPU-based Online Track Reconstruction for the ALICE TPC in Run 3 with Continuous Read-Out*. Tech. rep. arXiv:1905.05515. May 2019. URL: https://cds.cern.ch/record/2677616.

[40] David Rohr. "Tracking performance in high multiplicities environment at ALICE." In: *5th Large Hadron Collider Physics Conference (LHCP 2017) Shanghai, China, May 15-20, 2017*. 2017. arXiv: 1709.00618 [physics.ins-det].

[41] L. Ristori. "An artificial retina for fast track finding." In: *Nucl. Instrum. Meth.* A453 (2000), pp. 425–429. DOI: 10.1016/S0168-9002(00)00676-8.

[42] A Abba et al. *A specialized track processor for the LHCb upgrade*. Tech. rep. LHCb-PUB-2014-026. CERN-LHCb-PUB-2014-026. Geneva: CERN, Mar. 2014. URL: https://cds.cern.ch/record/1667587.

[43] A. Abba et al. "Simulation and performance of an artificial retina for 40 MHz track reconstruction." In: *JINST* 10.03 (2015), p. C03008. DOI: 10.1088/1748-0221/10/03/C03008. arXiv: 1409.0898 [physics.ins-det].

[44] CMS Collaboration. *The Phase-2 Upgrade of the CMS Tracker*. Tech. rep. CERN-LHCC-2017-009. CMS-TDR-014. Geneva: CERN, June 2017. URL: https://cds.cern.ch/record/2272264.

[45] Edward Bartz et al. "FPGA-Based Tracklet Approach to Level-1 Track Finding at CMS for the HL-LHC. FPGA-Based Tracklet Approach to Level-1 Track Finding at CMS for the HL-LHC." In: *EPJ Web Conf.* 150.arXiv:1706.09225 (June 2017), 00016. 11 p. DOI: 10.1051/epjconf/201715000016. URL: https://cds.cern.ch/record/2287519.

[46] Marco Trovato. "Track Trigger at the High Luminosity LHC." In: *PoS* LHCP2018 (2018), 259. 6 p. DOI: 10.22323/1.321.0259. URL: http://cds.cern.ch/record/2669228.

[47] M Shochet, L Tompkins, V Cavaliere, P Giannetti, A Annovi, and G Volpi. *Fast TracKer (FTK) Technical Design Report*. Tech. rep. CERN-LHCC-2013-007. ATLAS-TDR-021. June 2013. URL: https://cds.cern.ch/record/1552953.

[48] *The VMEbus specification manual: conforms to ANSI/IEEE Std 1014-1987, IEC 821 and 297*. Scottsdale, AZ: VITA, 1987. URL: https://cds.cern.ch/record/113323.

[49] *Advanced TCA base specification: advanced TCA*. Wakefield, MA: PICMG, 2008. URL: https://cds.cern.ch/record/1159877.

[50] Simone Sottocornola. "FTK: An Hardware based Tracker for the ATLAS Experiment." In: (2018), 41–48. 8 p. URL: https://cds.cern.ch/record/2677371.

[51] A Guirao. "READOUT OF HIGH SPEED S-LINK DATA VIA A BUFFERED PCI CARD." In: (Oct. 2002). URL: http://cds.cern.ch/record/1744711.

[52] *FMC standard products and support*. https://www.samtec.com/standards/vita/fmc. Accessed: 20-10-2019.

[53] *QSFP technical specifications document*. https://members.snia.org/document/dl/25963. Accessed: 20-09-2019.

[54] *SFP technical specification document*. http://www.snia.org/sff/specifications. Accessed: 20-09-2019.

[55] P. Fischer. "First implementation of the MEPHISTO binary readout architecture for strip detectors." In: *Nucl. Instrum. Meth.* A461 (2001), pp. 499–504. DOI: 10.1016/S0168-9002(00)01283-3.

[56] Li Shang, Alireza S Kaviani, and Kusuma Bathala. "Dynamic power consumption in Virtex™-II FPGA family." In: *Proceedings of the 2002 ACM/SIGDA tenth international symposium on Field-programmable gate arrays*. ACM. 2002, pp. 157–164.

[57] A. Amouri, F. Bruguier, S. Kiamehr, P. Benoit, L. Torres, and M. Tahoori. "Aging effects in FPGAs: an experimental analysis." In: *2014 24th International Conference on Field Programmable Logic and Applications (FPL)*. Sept. 2014, pp. 1–4. DOI: 10.1109/FPL.2014.6927390.

[58] *Altera Arria-V Data Sheets*. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/arria-v/av_51002.pdf. Accessed: 20-09-2019.

[59] *Xilinx kintex-7 Data Sheets*. https://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf. Accessed: 20-09-2019.

[60] *Xilinx Artix-7 Data Sheets*. https://www.xilinx.com/support/documentation/data_sheets/ds181_Artix_7_Data_Sheet.pdf. Accessed: 20-09-2019.

[61] *Xilinx Spartan-6 Data Sheets*. https://www.xilinx.com/support/documentation/data_sheets/ds162.pdf. Accessed: 20-09-2019.

[62] G. Thomas. "Rack Cooling Project - Final report." In: (Aug. 2002). URL: http://cds.cern.ch/record/38031.

[63] C L Sotiropoulou et al. "The Associative Memory System Infrastructures for the ATLAS Fast Tracker." In: *IEEE Trans. Nucl. Sci.* 64.6 (2017), 1248–1254. 7 p. DOI: 10.1109/TNS.2017.2703908. URL: https://cds.cern.ch/record/2295706.

[64] Andrei Kazarov et al. *The controls and configuration software of the ATLAS data acquisition system for the LHC run 2*. Tech. rep. ATL-DAQ-PROC-2019-007. Geneva: CERN, May 2019. URL: http://cds.cern.ch/record/2674871.

[65] *CORBA C++ programming reference*. https://docs.oracle.com/cd/E13211_01/wle/cref/index.htm. Accessed: 20-08-2019.

[66] *CMake open-source, cross-platform family of tools designed to build, test and package software*. https://cmake.org/. Accessed: 02-09-2019.

[67] *GitLab DevOps platform*. https://about.gitlab.com. Accessed: 02-09-2019.

[68] S Gameiro et al. *The ROD Crate DAQ of the ATLAS Data Acquisition System*. Tech. rep. ATL-DAQ-CONF-2005-020. ATL-COM-DAQ-2005-019. Geneva: CERN, 2005. URL: https://cds.cern.ch/record/844033.

[69] *The IPbus protocol*. https://twiki.cern.ch/twiki/pub/LAr/LArDemonstrator/ipbus_protocol_v2_0.pdf. Accessed: 207-09-2019.

[70] *I2C specification and user manual*. https://www.nxp.com/docs/en/user-guide/UM10204.pdf. Accessed: 207-09-2019.

[71] R Jones, Livio P Mapelli, Yu Ryabov, and I V Soloviev. "The OKS persistent in-memory object manager." In: *IEEE Trans. Nucl. Sci.* 45.4 (1998), pp. 1958–1964. DOI: 10.1109/23.710971. URL: https://cds.cern.ch/record/370902.

[72] Andrea Valassi, M Clemencic, D Dulstra, N Goyal, A Salnikov, R Trentadue, and M Wache. "CORAL and COOL during the LHC long shutdown." In: *J. Phys.: Conf. Ser.* 513.FERMILAB-CONF-13-498-CD (Nov. 2013), 042045. 6 p. DOI: 10.1088/1742-6596/513/4/042045. URL: https://cds.cern.ch/record/2025664.

[73] *POOL Relational Abstraction Layer*. https://indico.cern.ch/event/419656/contributions/1895565/attachments/874032/1225437/POOLRAL3Dworkshop.pdf. Accessed: 207-09-2019.

[74] *SEAL homomorphic encryption library*. https://github.com/Microsoft/SEAL. Accessed: 207-09-2019.

[75] G Anders, G Avolio, G Lehmann Miotto, and L Magnoni. "Intelligent operations of the data acquisition system of the ATLAS experiment at the LHC." In: (Aug. 2014). URL: https://cds.cern.ch/record/1752130.

[76] W Vandelli et al. *The ATLAS Event Builder*. Tech. rep. ATL-DAQ-CONF-2008-005. ATL-COM-DAQ-2007-047. 6, Part 2. Geneva: CERN, Nov. 2007. DOI: 10.1109/TNS.2008.2006050. URL: https://cds.cern.ch/record/1070827.

[77] *Information Service User's guide*. https://atlas-tdaq-monitoring.web.cern.ch/atlas-tdaq-monitoring/IS/doc/userguide/is-usersguide.pdf. Accessed: 20-08-2019.

[78] *OH User's API documentation*. https://atlas-tdaq-monitoring.web.cern.ch/atlas-tdaq-monitoring/OH/refman/index.html. Accessed: 20-08-2019.

[79] *Grafana Open platform for analytics and monitoring*. https://grafana.com/. Accessed: 20-08-2019.

[80] *Event MONitoring Design document*. https://atlas-tdaq-monitoring.web.cern.ch/atlas-tdaq-monitoring/EMON/design.pdf. Accessed: 20-08-2019.

[81] Fred Jegerlehner. "The Hierarchy Problem and the Cosmological Constant Problem Revisited – A new view on the SM of particle physics." In: *Found. Phys.* 49.9 (2019), pp. 915–971. DOI: 10.1007/s10701-019-00262-2. arXiv: 1812.03863 [hep-ph].

[82] J. H. Christenson, J. W. Cronin, V. L. Fitch, and R. Turlay. "Evidence for the 2π Decay of the K_2^0 Meson." In: *Phys. Rev. Lett.* 13 (1964), pp. 138–140. DOI: 10.1103/PhysRevLett.13.138.

[83] Laurent Canetti, Marco Drewes, and Mikhail Shaposhnikov. "Matter and Antimatter in the Universe." In: *New J. Phys.* 14 (2012), p. 095012. DOI: 10.1088/1367-2630/14/9/095012. arXiv: 1204.4186 [hep-ph].

[84] G. Karagiorgi, A. Aguilar-Arevalo, J. M. Conrad, M. H. Shaevitz, K. Whisnant, M. Sorel, and V. Barger. "Leptonic CP violation studies at MiniBooNE in the (3+2) sterile neutrino oscillation hypothesis." In: *Phys. Rev.* D75 (2007), p. 013011. DOI: 10.1103/PhysRevD.75.013011, 10.1103/PhysRevD.80.099902. arXiv: hep-ph/0609177 [hep-ph].

[85] John F. Gunion, Howard E. Haber, Gordon L. Kane, and Sally Dawson. "The Higgs Hunter's Guide." In: *Front. Phys.* 80 (2000), pp. 1–404.

[86] Robert Harlander, Michael Krämer, and Markus Schumacher. *Bottom-quark associated Higgs-boson production: reconciling the four- and five-flavour scheme approach*. 2011. arXiv: 1112.3478 [hep-ph].

[87] Martin Flechl, Richard Klees, Michael Kramer, Michael Spira, and Maria Ubiali. "Improved cross-section predictions for heavy charged Higgs boson production at the LHC." In: *Phys. Rev.* D91.7 (2015), p. 075015. DOI: 10.1103/PhysRevD.91.075015. arXiv: 1409.5615 [hep-ph].

[88] D. de Florian et al. "Handbook of LHC Higgs Cross Sections: 4. Deciphering the Nature of the Higgs Sector." In: (2016). DOI: 10.2172/1345634, 10.23731/CYRM-2017-002. arXiv: 1610.07922 [hep-ph].

[89] Andrew Todd Aukerman and Tae Min Hong. "ATLAS Level-1 Topological Trigger : Commissioning and Validation in Run 2." In: (Oct. 2017). URL: https://cds.cern.ch/record/2289427.

[90] *The ATLAS Tau Trigger in Run 2*. Tech. rep. ATLAS-CONF-2017-061. Geneva: CERN, July 2017. URL: https://cds.cern.ch/record/2274201.

[91] Morad Aaboud and all. "Performance of the ATLAS Trigger System in 2015. Performance of the ATLAS Trigger System in 2015." In: *Eur. Phys. J. C* 77.CERN-EP-2016-241. 5 (Nov. 2016), 317. 76 p. DOI: 10.1140/epjc/s10052-017-4852-3. URL: https://cds.cern.ch/record/2235584.

[92] *Reconstruction, Energy Calibration, and Identification of Hadronically Decaying Tau Leptons in the ATLAS Experiment for Run-2 of the LHC*. Tech. rep. ATL-PHYS-PUB-2015-045. Geneva: CERN, Nov. 2015. URL: https://cds.cern.ch/record/2064383.

[93] *Trigger monitoring and rate predictions using Enhanced Bias data from the ATLAS Detector at the LHC*. Tech. rep. ATL-DAQ-PUB-2016-002. Geneva: CERN, Oct. 2016. URL: https://cds.cern.ch/record/2223498.

[94] "Search for charged Higgs bosons decaying via $H^{\pm} \to \tau^{\pm}\nu_{\tau}$ in the τ+jets and τ+lepton final states with 36 fb^{-1} of pp collision data recorded at $\sqrt{s} = 13$ TeV with the ATLAS experiment." In: *JHEP* 09.arXiv:1807.07915 (July 2018), 139. 46 p. DOI: 10.1007/JHEP09(2018)139. URL: https://cds.cern.ch/record/2631950.

[95] J Alwall, R Frederix, S Frixione, V Hirschi, F Maltoni, O Mattelaer, H -S Shao, T Stelzer, P Torrielli, and M Zaro. "The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations." In: *JHEP* 07.arXiv:1405.0301. CERN-PH-TH-2014-064. CP3-14-18. LPN14-066. MCNET-14-09. ZU-TH

14-14 (May 2014), 079. 158 p. DOI: 10.1007/JHEP07(2014)079. URL: https://cds.cern.ch/record/1699128.

[96] Torbjörn Sjöstrand and Stephen Mrenna. "A Brief Introduction to PYTHIA 8.1." In: *Comput. Phys. Commun.* 178.arXiv:0710.3820. CERN-LCGAPP-2007-04. LU TP 07-28. FERMILAB-PUB-07-512-CD-T (Oct. 2007), 852–867. 27 p. DOI: 10.1016/j.cpc.2008.01.036. URL: https://cds.cern.ch/record/1064095.

[97] *ATLAS Run 1 Pythia8 tunes*. Tech. rep. ATL-PHYS-PUB-2014-021. Geneva: CERN, Nov. 2014. URL: https://cds.cern.ch/record/1966419.

[98] Richard D. Ball et al. "Parton distributions with LHC data." In: *Nuclear Physics B* 867.2 (2013), pp. 244–289. ISSN: 0550-3213. DOI: https://doi.org/10.1016/j.nuclphysb.2012.10.003. URL: http://www.sciencedirect.com/science/article/pii/S0550321312005500.

[99] P Nason. "A New Method for Combining NLO QCD with Shower Monte Carlo Algorithms." In: *JHEP* 11.hep-ph/0409146. BICOCCA-FT-2004-11 (Sept. 2004), 040. 29 p. URL: https://cds.cern.ch/record/792756.

[100] Stefano Frixione, Paolo Nason, and Carlo Oleari. "Matching NLO QCD computations with Parton Shower simulations: the POWHEG method." In: *JHEP* 11.arXiv:0709.2092. BICOCCA-FT-07-9. GEF-TH-21-2007 (Sept. 2007), 070. 91 p. URL: https://cds.cern.ch/record/1056860.

[101] Simone Alioli, Paolo Nason, and Carlo Oleari. *A general framework for implementing NLO calculations in shower Monte Carlo programs: the POWHEG BOX*. Tech. rep. arXiv:1002.2581. Feb. 2010. DOI: 10.1007/JHEP06(2010)043. URL: https://cds.cern.ch/record/1240986.

[102] Hung-Liang Lai, M Guzzi, Joey Huston, Zhao Li, Pavel M Nadolsky, Jon Pumplin, and C -P Yuan. *New parton distributions for collider physics*. Tech. rep. arXiv:1007.2241. MSUHEP-100707. SMU-HEP-10-10. July 2010. URL: https://cds.cern.ch/record/1278008.

[103] Sayipjamal Dulat, Tie-Jiun Hou, Jun Gao, Marco Guzzi, Joey Huston, Pavel Nadolsky, Jon Pumplin, Carl Schmidt, Daniel Stump, and C.-P. Yuan. "New parton distribution functions from a global analysis of quantum chromodynamics." In: *Physical Review D* 93.3 (Feb. 2016). ISSN: 2470-0029. DOI: 10.1103/physrevd.93.033006. URL: http://dx.doi.org/10.1103/PhysRevD.93.033006.

[104] Pierre Artoisenet, Rikkert Frederix, Olivier Mattelaer, and Robbert Rietkerk. "Automatic spin-entangled decays of heavy resonances in Monte Carlo simulations." In: *Journal of High Energy Physics* 2013.3 (Mar. 2013). ISSN: 1029-8479. DOI: 10.1007/jhep03(2013)015. URL: http://dx.doi.org/10.1007/JHEP03(2013)015.

[105] Torbjörn Sjöstrand and Stephen Mrenna. "PYTHIA 6.4 physics and manual." In: *Journal of High Energy Physics* 2006.05 (May 2006), pp. 026–026. ISSN: 1029-8479. DOI: 10.1088/1126-6708/2006/05/026. URL: http://dx.doi.org/10.1088/1126-6708/2006/05/026.

[106] Jonathan Pumplin, Daniel Robert Stump, Joey Huston, Hung-Liang Lai, Pavel Nadolsky, and Wu-Ki Tung. "New Generation of Parton Distributions with Uncertainties from Global QCD Analysis." In: *Journal of High Energy Physics* 2002.07 (July 2002), pp. 012–012. ISSN: 1029-8479. DOI: 10.1088/1126-6708/2002/07/012. URL: http://dx.doi.org/10.1088/1126-6708/2002/07/012.

[107] Peter Z. Skands. "Tuning Monte Carlo generators: The Perugia tunes." In: *Physical Review D* 82.7 (Oct. 2010). ISSN: 1550-2368. DOI: 10.1103/physrevd.82.074018. URL: http://dx.doi.org/10.1103/PhysRevD.82.074018.

[108] Michał Czakon and Alexander Mitov. "Top++: A program for the calculation of the top-pair cross-section at hadron colliders." In: *Computer Physics Communications* 185.11 (Nov. 2014), pp. 2930–2938. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2014.06.021. URL: http://dx.doi.org/10.1016/j.cpc.2014.06.021.

[109] P. Kant, O.M. Kind, T. Kintscher, T. Lohse, T. Martini, S. Mölbitz, P. Rieck, and P. Uwer. "HatHor for single top-quark production: Updated predictions and uncertainty estimates for single top-quark production in hadronic collisions." In: *Computer Physics Communications* 191 (June 2015), pp. 74–89. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2015.02.001. URL: http://dx.doi.org/10.1016/j.cpc.2015.02.001.

[110] M. Aliev, H. Lacker, U. Langenfeld, S. Moch, P. Uwer, and M. Wiedermann. "HATHOR – HAdronic Top and Heavy quarks crOss section calculatoR." In: *Computer Physics Communications* 182.4 (Apr. 2011), pp. 1034–1046. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2010.12.040. URL: http://dx.doi.org/10.1016/j.cpc.2010.12.040.

[111] Nikolaos Kidonakis. "Two-loop soft anomalous dimensions for single top quark associated production with aW-orH-." In: *Physical Review D* 82.5 (Sept. 2010). ISSN: 1550-2368. DOI: 10.1103/physrevd.82.054018. URL: http://dx.doi.org/10.1103/PhysRevD.82.054018.

[112] T Gleisberg, S Höche, F Krauss, M Schönherr, S Schumann, F Siegert, and J Winter. "Event generation with SHERPA 1.1." In: *Journal of High Energy Physics* 2009.02 (Feb. 2009), pp. 007–007. ISSN: 1029-8479. DOI: 10.1088/1126-6708/2009/02/007. URL: http://dx.doi.org/10.1088/1126-6708/2009/02/007.

[113] Richard D. Ball, Valerio Bertone, Stefano Carrazza, Christopher S. Deans, Luigi Del Debbio, Stefano Forte, Alberto Guffanti, Nathan P. Hartland, José I. Latorre, and et al. "Parton distributions for the LHC run II." In: *Journal of High Energy Physics* 2015.4 (Apr. 2015). ISSN: 1029-8479. DOI: 10.1007/jhep04(2015)040. URL: http://dx.doi.org/10.1007/JHEP04(2015)040.

[114] Tanju Gleisberg and Stefan Höche. "Comix, a new matrix element generator." In: *Journal of High Energy Physics* 2008.12 (Dec. 2008), pp. 039–039. ISSN: 1029-8479. DOI: 10.1088/1126-6708/2008/12/039. URL: http://dx.doi.org/10.1088/1126-6708/2008/12/039.

[115] F. Cascioli, P. Maierhöfer, and S. Pozzorini. "Scattering Amplitudes with Open Loops." In: *Physical Review Letters* 108.11 (Mar. 2012). ISSN: 1079-7114. DOI: 10.1103/physrevlett.108.111601. URL: http://dx.doi.org/10.1103/PhysRevLett.108.111601.

[116] S Schumann and F Krauss. "A parton shower algorithm based on Catani-Seymour dipole factorisation." In: *Journal of High Energy Physics* 2008.03 (Mar. 2008), pp. 038–038. ISSN: 1029-8479. DOI: 10.1088/1126-6708/2008/03/038. URL: http://dx.doi.org/10.1088/1126-6708/2008/03/038.

[117] Stefan Höche, Frank Krauss, Marek Schönherr, and Frank Siegert. "QCD matrix elements + parton showers. The NLO case." In: *Journal of High Energy Physics* 2013.4 (Apr. 2013). ISSN: 1029-8479. DOI: 10.1007/jhep04(2013)027. URL: http://dx.doi.org/10.1007/JHEP04(2013)027.

[118] Charalampos Anastasiou, Lance J. Dixon, Kirill Melnikov, and Frank Petriello. "High precision QCD at hadron colliders: Electroweak gauge boson rapidity distributions at NNLO." In: *Phys. Rev.* D69 (2004), p. 094008. DOI: 10.1103/PhysRevD.69.094008. arXiv: hep-ph/0312266 [hep-ph].

[119] D. J. Lange. "The EvtGen particle decay simulation package." In: *Nucl. Instrum. Meth.* A462 (2001), pp. 152–155. DOI: 10.1016/S0168-9002(01)00089-4.

[120] *Summary of ATLAS Pythia 8 tunes*. Tech. rep. ATL-PHYS-PUB-2012-003. Geneva: CERN, Aug. 2012. URL: https://cds.cern.ch/record/1474107.

[121] A. D. Martin, W. J. Stirling, R. S. Thorne, and G. Watt. "Parton distributions for the LHC." In: *The European Physical Journal C* 63.2 (July 2009), pp. 189–285. ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-009-1072-5. URL: http://dx.doi.org/10.1140/epjc/s10052-009-1072-5.

[122] G. Aad, B. Abbott, J. Abdallah, A. A. Abdelalim, A. Abdesselam, O. Abdinov, B. Abi, M. Abolins, H. Abramowicz, and et al. "The ATLAS Simulation Infrastructure." In: *The European Physical Journal C* 70.3 (Sept. 2010), pp. 823–874. ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-010-1429-9. URL: http://dx.doi.org/10.1140/epjc/s10052-010-1429-9.

[123] S. Agostinelli et al. "GEANT4: A Simulation toolkit." In: *Nucl. Instrum. Meth.* A506 (2003), pp. 250–303. DOI: 10.1016/S0168-9002(03)01368-8.

[124] Matteo Cacciari, Gavin P Salam, and Gregory Soyez. "The antiktjet clustering algorithm." In: *Journal of High Energy Physics* 2008.04 (Apr. 2008), pp. 063–063. ISSN: 1029-8479. DOI: 10.1088/1126-6708/2008/04/063. URL: http://dx.doi.org/10.1088/1126-6708/2008/04/063.

[125] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. "FastJet user manual." In: *The European Physical Journal C* 72.3 (Mar. 2012). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-012-1896-2. URL: http://dx.doi.org/10.1140/epjc/s10052-012-1896-2.

[126] *Tagging and suppression of pileup jets with the ATLAS detector*. Tech. rep. ATLAS-CONF-2014-018. Geneva: CERN, May 2014. URL: https://cds.cern.ch/record/1700870.

[127] "Performance ofb-jet identification in the ATLAS experiment." In: *Journal of Instrumentation* 11.04 (Apr. 2016), P04008–P04008. ISSN: 1748-0221. DOI: 10.1088/1748-0221/11/04/p04008. URL: http://dx.doi.org/10.1088/1748-0221/11/04/P04008.

[128] M. Aaboud, G. Aad, B. Abbott, O. Abdinov, B. Abeloos, S. H. Abidi, O. S. AbouZeid, N. L. Abraham, H. Abramowicz, and et al. "Performance of missing transverse momentum reconstruction with the ATLAS detector using proton–proton collisions at $\sqrt{s} = 13$ TeV." In: *The European Physical Journal C* 78.11 (Nov. 2018). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-018-6288-9. URL: http://dx.doi.org/10.1140/epjc/s10052-018-6288-9.