UNIVERSITÀ DI SIENA 1240

Department of Physical Sciences, Earth and Environment
PhD in Experimental Physics
XXXIV Cycle
Coordinator: Prof. Riccardo Paoletti

Improving charm CPV measurements with real-time data reconstruction

Disciplinary Scientific Sector: FIS/01
A thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

PhD Student
Federico Lazzari

Supervisor
Prof. Giovanni Punzi

Tutor
Prof. Riccardo Paoletti

2018/2021

Contents

Chapter 1

Introduction

Precise measurements of CP violation (CPV) in the charm sector play a key role in probing the Standard Model (SM), and has therefore a relevant place in the LHCb physics program (see Chapter 2). The charm quark is the only up-type quark that allows to study CPV, in fact the up quark creates π^0, that is a CP eigenstate, and the top quark decays before it can hadronize. However CPV in charm decays is expected to be equal ore less than 10^{-3} and the theoretical predictions are not straightforward. Therefore huge samples of c-hadrons decays are needed. For these reasons, CPV in charm decays remained unobserved until 2019, when the LHCb collaboration observed a significant difference in the CP asymmetries of $D^0 \to \pi^+\pi^-$ and $D^0 \to K^+K^-$ decays [1]. The result is generally believed to be compatible with a SM origin, but the present level of theoretical understanding does not allow a very precise comparison. In this context it is crucial to collect greater data samples and start a systematic exploration of all the c-hadron decay channels, to improve the existing measurement and eventually get for confirmation of CP violation.

$D^0 \to K^0\overline{K}^{*0}$ and $D^0 \to \overline{K}^0 K^{*0}$ decays are two decay channels with a predicted CP asymmetry of order 10^{-3}. The prompt decay $K^{*0} \to K^+\pi^-$ produces charged tracks pointing directly to the D^0 decay vertex, allowing to trigger it efficiently and collect large samples. During Run 2, LHCb collected $845 \cdot 10^3$ $D^0 \to K^0_S K^-\pi^+$ events and $617 \cdot 10^3$ $D^0 \to K^0_S K^+\pi^-$ events. These decays receive contributions from several resonances. The LHCb detector description is reported in Chapter 3.

In this thesis I introduce a novel analysis method to extract CP-violating parameters from individual resonances with the maximum attainable resolution without the need for a full amplitude analysis, also in the presence of significant interference effects (see Chapter 4). The result of the analysis is still blind, pending completion of the internal LHCb review - however the studies on the statistical uncertainties show that the resolutions scales up as expected by the increase of data sample, indicating that the newly introduced methodology does not lose power in comparison with a full Dalitz fit.

The current sensitivity of this measurement is not yet at the level of the expected CPV effect, but already in the current year (2022) LHCb will start acquiring significant further data, with an almost completely renewed detector (Chapter 5).

The instantaneous luminosity will increase by a factor of 5 and the amount of data collected will increase from $\sim 9\,\mathrm{fb}^{-1}$ (collected up to now) to $\sim 50\,\mathrm{fb}^{-1}$ at the end of Run 4. The increased luminosity requires a completely new trigger system. Simple quantities as the deposit of transverse energy (E_{T}) or tracks with high transverse momentum (p_{T}), usually the only information available at the first level trigger, have not enough discriminating power to ensure high trigger efficiency for hadronic signal decays. The LHCb collaboration chose to implement a full software reconstruction of every collision event in real-time, allowing to trigger directly on advanced tracks parameters. Due to the computational power required by this task, the collaboration adopted an heterogeneous solution, with the first stage of reconstruction and trigger performed on GPU, and the second level on CPU. At the same time, the collaboration has already put forward a Framework TDR for continuing operation beyond Run 4, at even higher luminosities. The aim is to collect data up to a luminosity of $\mathcal{L} = 1.5 \cdot 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$, and integrate $300\,\mathrm{fb}^{-1}$. Further real-time computing improvements are needed to fulfil this goal. For this purpose, LHCb is carrying out R&D activities related to high-performance computing accelerator platforms for the future upgrade. One solution under development is a highly-parallelized custom tracking processor based on the "Artificial Retina" architecture, and implemented on Field Programmable Gate Arrays (FPGAs) (see Chapter 6).

In this thesis I describe my work in implementing the "Artificial Retina" architecture (see Chapter 7). The goal was to develop all the "Artificial Retina" sub-system and demonstrate that it can perform tracking in the LHCb extremely high-rate environment. One aspect of the system that needs to be thoroughly tested is the Distribution Network. It is a distinctive element of the "Artificial Retina" that allows to reconstruct tracks with throughput and latency performances never attained before. It relies on a high number of optical links and it is technologically challenging, and needs to be thoroughly tested. As a result of this development work, it has been established that a full-size "Artificial Retina" system can be made to operate correctly and consistently within the peculiar LHCb Upgrade II environment.

In Chapter 8 I discuss the realisation of a reduced-size demonstrator, specific for the VErtex LOcator (VELO) detector, that will be able to function in parasitic mode already in the upcoming Run 3. This demonstrator needs the coordinates of the hits on the VELO, an information not directly available, since the VELO produces the list of active pixels, and a single hit can activate multiple pixel. Taking inspiration from the "Artificial Retina", I conceived and developed a FPGA-based clustering algorithm, to recognise groups of contiguous active pixels (Chapter 9). This clustering firmware is now fully developed and tested and it is the first piece of the project to be commissioned for physics data taking, already in the Run 3 of the LHC that is about to start.

Chapter 2

CP violation in charm decays

2.1 The Standard Model

The Standard Model (SM) of particle physics is a quantum field theory describing the fundamental constituents of matter and the interactions among them [2–4]. The symmetries of the Lagrangian and the representations of the particles under these symmetries defined this model. The gauge group of symmetry of the SM is

$$G_{SM} = SU(3)_C \otimes SU(2)_L \otimes U(1)_Y \tag{2.1}$$

where $SU(3)_C$ is the symmetry group of the Quantum Chromo-Dynamics (QCD), which describes the strong force theory, with the subscript C refers to the colour charge of the field under a transformation of this group. The $SU(2)_L \otimes U(1)_Y$ term represents the symmetry group of the electro-weak interactions as introduced by the theory of Glashow-Weinberg-Salam [4, 5], with the subscripts L and Y refer to the chirality of the weak interactions and to the hypercharge, respectively.

The fundamental building blocks of matter are the half-integer spin particles that are representations of the G_{SM} group:

$$Q^I_{Li}(3,2)_{+1/6}, \quad u^I_{Ri}(3,1)_{+2/3}, \quad d^I_{Ri}(3,1)_{-1/3}, \quad L^I_{Li}(1,2)_{-1/2}, \quad \ell^I_{Ri}(1,1)_{-1}, \tag{2.2}$$

where $i = 1, 2, 3$ runs over the generation of fermions (generation index), the index $L(R)$ indicates the left (right) chirality, and the index I denotes the interaction eigenstates. This notation makes the representations and the quantum numbers of the fields manifest. Left-handed quarks, Q^I_L , are triplets of $SU(3)_C$, doublets of $SU(2)_L$, and carry hypercharge $Y = +1/6$; right-handed up-type quarks, u^I_R, are triplets of $SU(3)_C$, singlets of $SU(2)_L$, and carry hypercharge $Y = +2/3$; right-handed down-type quarks, d^I_R, are triplets of $SU(3)_C$, singlets of $SU(2)_L$, and carry hypercharge $Y = -1/3$. Leptons are singlets of $SU(3)_C$ and are classified according to the transformation properties of their fields with respect to $SU(2)_L$. Left-handed leptons, L^I_L, are doublets of $SU(2)_L$; right-handed leptons, ℓ^I_R, are singlets of $SU(2)_L$.

In addition to fermions representation, there is a single scalar representation $(1/2, 1/2)$:

$$\Phi = \begin{pmatrix} \Phi^+ \\ \Phi^0 \end{pmatrix}, \tag{2.3}$$

which assumes a vacuum expectation value of

$$\langle\Phi\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\ \nu\end{pmatrix}, \tag{2.4}$$

often parameterized as:

$$\Phi = \exp\left[i\frac{\sigma_j}{2}\theta_j\right]\frac{1}{\sqrt{2}}\begin{pmatrix}0\\ \nu+H\end{pmatrix} \tag{2.5}$$

where σ_j are the Pauli matrices, θ_j are three real fields and H is a neutral scalar field known as Higgs boson field. The non-zero vacuum expectation generates a spontaneous breaking of the gauge group

$$G_{SM} \to SU(3)_C \otimes U(1)_{EM}, \tag{2.6}$$

where $U(1)_{EM}$ is the symmetry group of electromagnetism. This classification of leptons, quarks and bosons is schematically summarised in Figure 2.1.

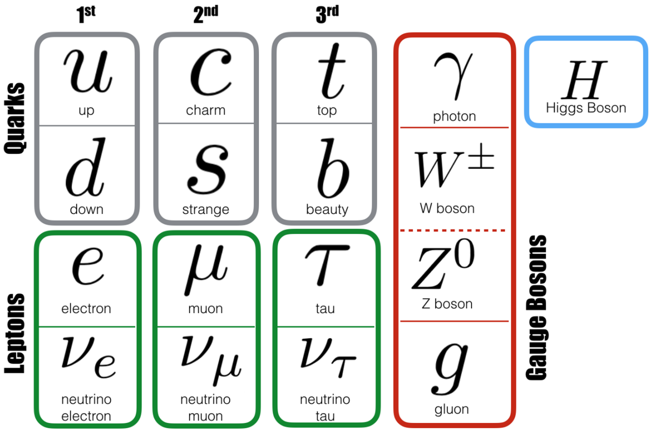


Figure 2.1: Elementary particles forming the Standard Model. The six quarks are highlighted in grey, the six leptons in green, the four gauge bosons in red and the Higgs boson in blue.

Once the gauge symmetry, the particle content, and the pattern of spontaneous symmetry breaking are defined, the SM Lagrangian is derived as the most general renormalisable Lagrangian satisfying these requirements. It can be divided in four terms

$$\mathcal{L}_{SM} = \mathcal{L}_{kinetic} + \mathcal{L}_{gauge} + \mathcal{L}_{Higgs} + \mathcal{L}_{Yukawa}. \tag{2.7}$$

The kinetic term describes interaction between quarks and gauge bosons. It is the sum of all kinetic terms of fermions:

$$\mathcal{L}_{kinetic} = i\overline{\psi}\gamma^{\mu}D_{\mu}\psi, \tag{2.8}$$

where γ^μ are the Dirac matrices, ψ is a Dirac spinor, $\overline{\psi} = \psi^\dagger\gamma^0$ is the adjoint spinor, and D_μ is the covariant derivative that replaced the standard derivative in order to maintain the gauge invariance. It is defined as

$$D_\mu = \partial_\mu + \frac{ig_s}{2}G^a_\mu\lambda_a + \frac{ig}{2}W^d_\mu\sigma_d + \frac{ig'}{2}B_\mu Y, \tag{2.9}$$

where Y, σ_d and λ_a are respectively the $U(1)_Y$, $SU(2)_W$, and $SU(3)_C$ symmetries.

The gauge term the describes the boson kinetic term, it is

$$\mathcal{L}_{gauge} = -\frac{1}{4}G^a_{\mu\nu}(G^a)^{\mu\nu} - \frac{1}{4}W^d_{\mu\nu}(W^d)^{\mu\nu} - \frac{1}{4}B_{\mu\nu}B^{\mu\nu}, \tag{2.10}$$

where $(G^a)^{\mu\nu}$ is the Yang-Mills tensor which represents the eight $(a = 1, ..., 8)$ gluon fields, $(W^d)^{\mu\nu}$ is the weak field tensor that represent three $(d = 1, 2, 3)$ gauge fields, and $B_{\mu\nu}$ is the electromagnetic tensor that represents $U(1)_Y$ gauge field B_μ.

The Higgs term describes the Higgs self-interaction, and the spontaneous electroweak symmetry breaking which allows all the SM particles to acquire mass. The Langrangian is written as

$$\mathcal{L}_{Higgs} = (\nabla_\mu\Phi)^\dagger(\nabla^\mu\Phi) + \mu^2\Phi^\dagger\Phi + \lambda(\Phi^\dagger\Phi)^2, \tag{2.11}$$

where the first term represents the kinetic energy of the Higgs field together with its gauge interactions, and the other two represent the mass term and the self-interaction term respectively.

The Yukawa term describes the coupling between fermions and the scalar field. The Langrangian is written as

$$\mathcal{L}_{Yukawa} = -Y^d_{ij}\overline{Q}^I_{Li}\Phi D^I_{Rj} - Y^u_{ij}\overline{Q}^I_{Li}\overline{\Phi}U^I_{Rj} - Y^\ell_{ij}\overline{L}^I_{Li}\Phi\ell^I_{Rj} + \text{h.c.}, \tag{2.12}$$

where $Y^{u,d,\ell}_{ij}$ are 3×3 complex matrices, $i, j = 1, 2, 3$ are the generation indexes and $\overline{\Phi} = i\sigma^2\Phi^\dagger$.

The SM is the best description of fundamental physics interactions currently available but it does not provide a complete picture. It incorporates three of the four fundamental forces, omitting gravity. Moreover, the existence of "dark matter" is not explainable in the SM context, while there are several strong hints of its existence. The reasons why there are three generations of quarks and leptons is left completly open, and so is the mass scale hierarchy. Another crucial issue with the SM in the role played by *CP* violation (*CPV*), the breaking the invariance of physical processes under the *CP* transformation. The *CP* operator combines the charge conjugation C with the parity reverse P. Under the C transformation all intrinsic quantum numbers are inverted. Under P the spatial coordinates are inverted. While the *CPV* phenomena observed in particle physics laboratories can be accommodated by the SM, it offers no fundamental explanation of it, and the resulting parameterisation is quantitatively insufficient to explain the cosmological matter-antimatter asymmetry in the Universe. Many extensions of the SM, that have not yet been experimentally observed, include additional fundamental sources of *CPV*.

Those facts have motivated several large-scale experimental campaigns aimed at precision measurements of physics processes sensitive to *CPV*, seeking to improve our knowledge of Nature. Amongst them, LHCb at CERN has been specifically designed to perform precision measurements of charm and bottom decay observables, and in that context the present thesis work has been developed. The following sections will discuss the subject in greater detail.

2.2 *CP* violation in the Standard Model

Within the SM, *CP* symmetry is broken by an irreducible complex physical phase in the Yukawa quark-term of the SM Lagrangian. *CP* symmetry is therefore preserved in strong and electromagnetic interactions, as supported by all experimental results thus far [6–8], but violated in weak interactions.

In the basis of mass eigenstates, the charged current weak interactions for quarks have the following form:

$$\mathcal{L}_{W^\pm} = \frac{-g_2}{\sqrt{2}}(\overline{u}_L, \overline{c}_L, \overline{t}_L)\gamma^\mu V_{CKM}\begin{pmatrix} d_L \\ s_L \\ b_L \end{pmatrix} W_\mu^\dagger + \text{h.c.}. \tag{2.13}$$

The V_{CKM} is the Cabibbo-Kobayashi-Maskawa (CKM) matrix [9,10]; a 3×3 unitary matrix that parametrizes complex couplings between the quark-mass eigenstates and the charged weak gauge bosons $W^\pm$:

$$V_{CKM} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix}. \tag{2.14}$$

The CKM matrix can be parameterized by three mixing angles and a complex phase:

$$V_{CKM} = \begin{pmatrix} c_{12}c_{13} & s_{12}c_{13} & s_{13}e^{-i\delta} \\ -s_{12}c_{23} - c_{12}s_{23}s_{13}e^{-i\delta} & c_{12}c_{23} - s_{12}s_{23}s_{13}e^{i\delta} & s_{23}c_{13} \\ s_{12}s_{23} - c_{12}c_{23}s_{13}e^{i\delta} & -c_{12}s_{23} - s_{12}c_{23}s_{13}e^{i\delta} & c_{23}c_{13} \end{pmatrix}, \tag{2.15}$$

where $s_{ij} = \sin\theta_{ij}$, $c_{ij} = \cos\theta_{ij}$ and δ is the phase responsible for *CPV* in flavour-changing processes in the SM [9]. It is known experimentally that $s_{13} \ll s_{23} \ll s_{12} \ll 1$, therefore it is convenient to use the Wolfenstein parametrization that instead of the parameters $(s_{12}, s_{23}, s_{13}, \delta)$ uses four new parameters (λ, A, ρ, η) [11]:

$$V_{CKM} = \begin{pmatrix} 1 - \lambda^2/2 & \lambda & A\lambda^3(\rho - i\eta) \\ -\lambda & 1 - \lambda^2/2 & A\lambda^2 \\ A\lambda^3(1 - \rho - i\eta) & -A\lambda^2 & 1 \end{pmatrix} + \mathcal{O}(\lambda^4). \tag{2.16}$$

The unitarity of the CKM matrix imposes

$$\sum_i V_{ij}V_{ik}^* = \delta_{jk}, \quad \sum_j V_{ij}V_{kj}^* = \delta_{ik}, \tag{2.17}$$

hence the orthogonality among rows and columns. This six constraints are represented as triangles in a complex $(\overline{\rho} - \overline{\eta})$ plane, all having the same area. These are known as *unitarity triangles*. The most common is triangle arising from the $V_{ud}V_{ub}^* + V_{cd}V_{db}^* + V_{td}V_{tb}^* = 0$ condition. Dividing each term by $V_{cd}V_{cb}^*$ (the best experimentally known term), one obtain that the vertices of the unitary triangle are exactly (0,0), (1,0) and $(\overline{\rho}, \overline{\eta})$. Figure. 2.2 shows this unitarity triangle in the complex $(\overline{\rho}, \overline{\eta}))$ plane.

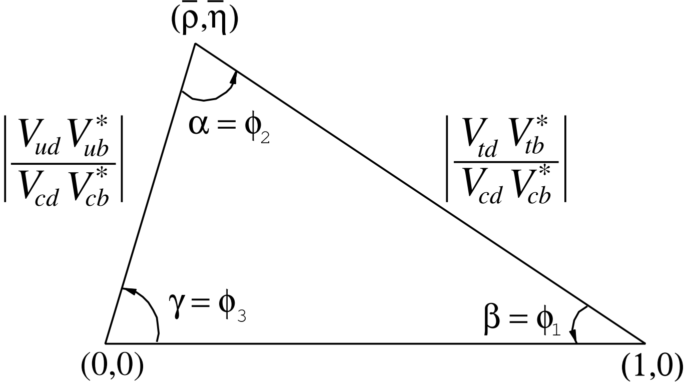

$(\bar{\rho},\bar{\eta})$
$\left|\frac{V_{ud}V_{ub}^*}{V_{cd}V_{cb}^*}\right|$
$\alpha = \phi_2$
$\left|\frac{V_{td}V_{tb}^*}{V_{cd}V_{cb}^*}\right|$
$\gamma = \phi_3$
$\beta = \phi_1$
(0,0)
(1,0)

Figure 2.2: Unitarity triangle in $(\overline{\rho}, \overline{\eta})$ plane.

The area of the unitarity triangles is equal to half of the Jarlskog invariant [12], J, defined as:

$$Im[V_{ij}V_{kl}V_{il}^*V_{kj}^*] = J \sum_{m,n \in (d,s,b)} \epsilon_{ikm}\epsilon_{jln} \tag{2.18}$$

It is a measure of *CPV* independent from the choice of the phase convention, and can be approximated by $J \approx \lambda^6 A^2 \eta$ in the Wolfenstein parametrization. *CPV* violation occurs only if $J \neq 0$, the current measurements indicate $J = (3.00^{+0.15}_{-0.09}) \times 10^{-5}$ [13].

The CKM matrix elements are fundamental parameters of the SM, their precise determination is important to make predictions and to put strong constraints on beyond Standard Model theories. The fit result for the magnitudes of all nine CKM elements are [13]

$$V_{CKM} = \begin{pmatrix} 0.97401 \pm 0.00011 & 0.22650 \pm 0.00048 & 0.00361^{+0.00011}_{-0.00009} \\ 0.22636 \pm 0.00048 & 0.97320 \pm 0.00011 & 0.04053^{+0.00083}_{-0.00061} \\ 0.00854^{+0.00032}_{-0.00016} & 0.03978^{+0.00082}_{-0.00060} & 0.999172^{+0.000024}_{-0.000035} \end{pmatrix}. \tag{2.19}$$

Most of the CKM matrix elements are determined through direct measurements, looking at tree level processes. In this way, it is possible to directly extract the value of $|V_{ij}|$. However, some elements have low precision, such as $|V_{tb}|$ and $|V_{cs}|$, or are too suppressed to be measured, such as $|V_{td}|$ and $|V_{ts}|$. In these cases, the indirect measurements are performed looking at higher-order processes. It is useful to display the various measurements and compare them in the $\overline{\rho}, \overline{\eta}$ plane. Figure 2.3 illustrates the global fit result of CKM parameters [14]. The shaded 99% CL regions all overlap consistently around the global fit region.

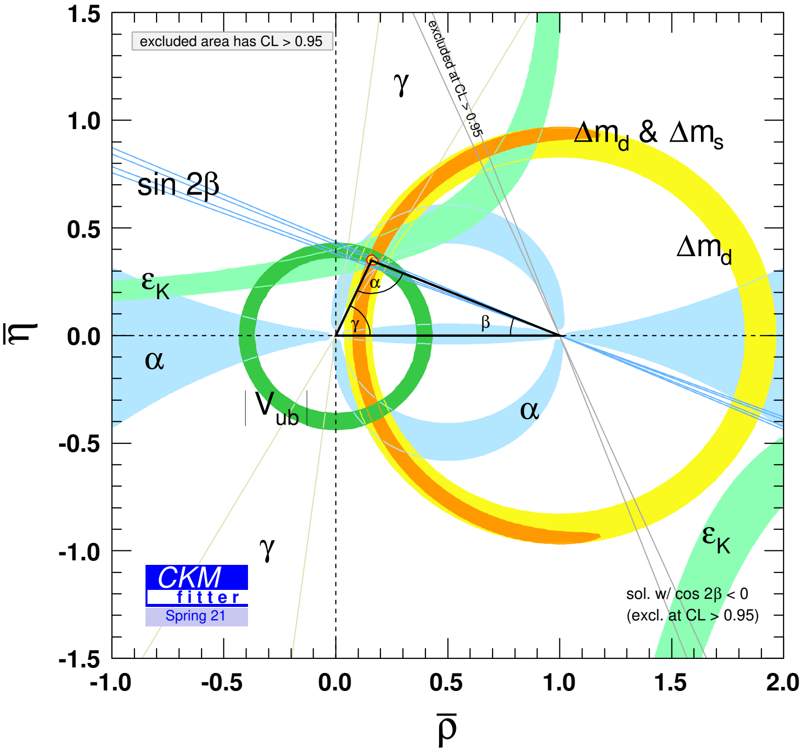


Figure 2.3: Current experimental status of the global fit to all available experimental measurements related to the unitarity triangle phenomenology. From Ref. [14].

2.2.1 Types of *CP* violation

The *CP* transformation for a *CP*-eigenstate f is $CP|f\rangle = \omega_f|f\rangle$ and $CP|\overline{f}\rangle = \omega_f^*|\overline{f}\rangle$, where ω_f is a complex phase ($|\omega_f| = 1$). Usually phases are of two types. They are called weak and strong phases. Weak phases come from complex terms in the Lagrangian appearing as complex conjugated in the *CP* conjugate amplitude. They occur only in the CKM matrix, which is part of the electroweak sector, therefore the are called weak phases. Strong phases come from final state interactions and they contribute to the amplitudes through the intermediate on-shell states in the decay process. The name derive from the strong interactions that generates hadrons in the final state. These phases arise even if the Lagrangian is real. Strong phases do not change sign under *CP* transformation.

Experimentally, *CP* violation manifests in the decay, in the mixing, and in the interference between decay and mixing. The *CP* violation in the decay is also called *direct CPV*, whereas the other two types are called *indirect CPV*.

CPV in the decay

The decay amplitudes of a generic particle P and of its antiparticle $\overline{P}$ into a final state f and to the C-conjugated final state $\overline{f}$ are defined as

$$\mathcal{A}(P \to f) := \langle f|\mathcal{H}|P\rangle \tag{2.20}$$

$$\mathcal{A}(\overline{P} \to \overline{f}) := \langle \overline{f}|\mathcal{H}|\overline{P}\rangle, \tag{2.21}$$

where $\mathcal{H}$ is the decay Hamiltonian. Considering a decay process which can proceed through several decay amplitudes, these two decay amplitudes can be written as

$$\mathcal{A}(P \to f) = \sum_k |a_k| \exp\{i(\phi_k + \delta_k)\} \tag{2.22}$$

$$\mathcal{A}(\overline{P} \to \overline{f}) = \sum_k |a_k| \exp\{i(-\phi_k + \delta_k)\}, \tag{2.23}$$

where ϕ_k are the weak phases, and δ_k are the strong phases, which do not change sign under *CP*.

The *CPV* in the decay is observed if $\mathcal{A}(P \to f) \neq \mathcal{A}(\overline{P} \to \overline{f})$. Since all observables are related to the squared amplitudes, a golden observable, sensitive to the *CPV* in the decay, is the *CP* asymmetry

$$A^{dir}(f) = \frac{\Gamma(P \to f) - \Gamma(\overline{P} \to \overline{f})}{\Gamma(P \to f) + \Gamma(\overline{P} \to \overline{f})} \tag{2.24}$$

where Γ is the time-integrated decay width of the decay process and it is proportional to the squared amplitude: $\Gamma(P \to f) \propto |\mathcal{A}(P \to f)|^2$ and $\Gamma(\overline{P} \to \overline{f}) \propto |\mathcal{A}(\overline{P} \to \overline{f})|^2$, thus

$$A^{dir}(f) = \frac{|\mathcal{A}(P \to f)|^2 - |\mathcal{A}(\overline{P} \to \overline{f})|^2}{|\mathcal{A}(P \to f)|^2 + |\mathcal{A}(\overline{P} \to \overline{f})|^2}. \tag{2.25}$$

The difference appearing in the numerator becomes:

$$|\mathcal{A}(P \to f)|^2 - |\mathcal{A}(\overline{P} \to \overline{f})|^2 = -2\sum_{i,j} |a_i||a_j| \sin(\phi_i - \phi_j) \sin(\delta_i - \delta_j). \tag{2.26}$$

It follows that *CPV* in the decay appears as a result of the interference among various terms in the decay amplitude, and it does not occur unless at least two terms have different weak phases and different strong phases.

Another way to re-write the *CP* asymmetry is

$$a^{dir} = \frac{1 - R_f^2}{1 + R_f^2}, \tag{2.27}$$

with

$$R_f = \left|\frac{\mathcal{A}(P \to f)}{\mathcal{A}(\overline{P} \to \overline{f})}\right|. \tag{2.28}$$

Therefore, the *CP* violation in the decay occurs if $R_f \neq 1$.

CPV in the mixing

Within the Standard Model, the mesons are defined as *flavoured* if they posses a non-null flavour quantum number (strangeness, charmness or bottomness). The $K^0(d\bar{s})$, $D^0(c\bar{u})$, $B^0(d\bar{b})$, and $B^0_s(s\bar{b})$ are neutral flavoured mesons that are unable to decay into lighter particles through a strong or electromagnetic interaction. The interaction eigenstates (or flavour eigenstates) in which they are produced is different from the mass ones. The mass eigentate is related to the free Hamiltonian, that drives the time evolution of the particle. Thus, it is possible for a flavoured meson to be produced with a certain flavour and then to oscillate into its antiparticle. This process is called *mixing*.

The initial state of a meson can be expressed as a linear combination of the flavour eigenstate M^0 and $\overline{M}^0$

$$|\psi(0)\rangle = a(0)|M^0\rangle + b(0)|\overline{M}^0\rangle. \tag{2.29}$$

The time evolution of this state described by the Schrodinger equation is

$$i\hbar\frac{\partial}{\partial t}|\psi\rangle = H|\psi(t)\rangle, \tag{2.30}$$

where H is the free Hamiltonian and $|\psi(t)\rangle$ is a linear superposition of $|M^0\rangle$, $|\overline{M}^0\rangle$, and all the final state $|f_k\rangle$ in which these two mesons can decay:

$$|\psi(t)\rangle = a(t)|M^0\rangle + b(t)|\overline{M}^0\rangle + \sum_k c_k(t)|f_k\rangle. \tag{2.31}$$

Aiming to find only $a(t)$ and $b(t)$, without distinguishing the final states to which the mesons decay, and considering times t much larger than the typical strong interaction scale, it is useful to apply the Weisskopf-Wigner approximation [15]. The simplified time evolution is determined by a 2×2 effective Hamiltonian $\boldsymbol{H}$ that can be written in terms of a Hermitian and an anti-Hermitian matrices:

$$\boldsymbol{H} = \boldsymbol{M} - \frac{i}{2}\boldsymbol{\Gamma}. \tag{2.32}$$

where $\boldsymbol{M}$ and $\boldsymbol{\Gamma}$ are the Hermitian mass and decay matrices respectively. The diagonal elements of the mass matrix and the decay matrix are associated with flavour-conserving transitions $M^0 \to M^0$ and $\overline{M}^0 \to \overline{M}^0$, whereas the non-diagonal elements are associated with flavour-changing transitions $M^0 \to \overline{M}^0$ and $\overline{M}^0 \to M^0$. If $\boldsymbol{H}$ is not diagonal, flavor eigenstates are not mass eigenstates. The normalised eigenstates of $\boldsymbol{H}$ are defined as

$$|M_1\rangle = p|M^0\rangle + q|\overline{M}^0\rangle \tag{2.33}$$

$$|M_2\rangle = p|M^0\rangle - q|\overline{M}^0\rangle, \tag{2.34}$$

where p and q are complex coefficients satisfying

$$|p|^2 + |q|^2 = 1 \tag{2.35}$$

$$\frac{q}{p} = \pm\sqrt{\frac{\boldsymbol{M}^*_{12} - (i/2)\boldsymbol{\Gamma}^*_{12}}{\boldsymbol{M}_{12} - (i/2)\boldsymbol{\Gamma}_{12}}}. \tag{2.36}$$

The eigenvalues of $\boldsymbol{H}$ are

$$\lambda_{1,2} = \boldsymbol{M}_{11} - \frac{i}{2}\boldsymbol{\Gamma}_{11} \pm \frac{q}{p}\big(\boldsymbol{M}_{12} - \frac{i}{2}\boldsymbol{\Gamma}_{12}\big) = m_{1,2} - \frac{1}{2}\Gamma_{1,2}, \tag{2.37}$$

where $m_{1,2}$ and $\Gamma_{1,2}$ correspond to the masses and decay widths of the two eigenstates. Usually, the mass and width differences $\Delta M = m_2 - m_1$ and $\Delta\Gamma = \Gamma_2 - \Gamma_1$ of the eigenstates are parametrised in units of the average decay width Γ, through the two dimensionless mixing parameters $x \equiv \Delta M/\Gamma$ and $y \equiv \Delta\Gamma/2\Gamma$.

The time evolution of unstable particle states is

$$|M_{1,2}(t)\rangle = e^{-im_{1,2}t}e^{-\frac{1}{2}\Gamma_{1,2}t}|M_{1,2}(0)\rangle, \tag{2.38}$$

and the time evolution of a particle that was created in its flavour eigenstate at $t = 0$:

$$|M^0(t)\rangle = g_+(t)|M^0\rangle + \frac{q}{p}g_-(t)|\overline{M}^0\rangle \tag{2.39}$$

$$|\overline{M}^0(t)\rangle = g_+(t)|\overline{M}^0\rangle + \frac{p}{q}g_-(t)|M^0\rangle, \tag{2.40}$$

with

$$g_\pm = \frac{e^{-i\lambda_1 t} \pm e^{-i\lambda_2 t}}{2}. \tag{2.41}$$

Given a particle produced in its flavour eigenstate at time $t = 0$, the probability of measuring at time t a particle with the same or opposite flavour is

$$\mathcal{P}(M^0 \to M^0(t)) = |g_+(t)|^2, \tag{2.42}$$

$$\mathcal{P}(\overline{M}^0 \to \overline{M}^0(t)) = |g_+(t)|^2, \tag{2.43}$$

$$\mathcal{P}(M^0 \to \overline{M}^0(t)) = \left|\frac{q}{p}\right|^2 \cdot |g_-(t)|^2, \tag{2.44}$$

$$\mathcal{P}(\overline{M}^0 \to M^0(t)) = \left|\frac{p}{q}\right|^2 \cdot |g_-(t)|^2, \tag{2.45}$$

with

$$|g_\pm|^2 = \frac{1}{2}e^{-\Gamma t}\left[cosh(y\Gamma t) \pm cos(x\Gamma t)\right]. \tag{2.46}$$

The *CP* violation in the mixing occurs if $\mathcal{P}(M^0 \to \overline{M}^0(t)) \neq \mathcal{P}(\overline{M}^0 \to M^0(t))$. Hence if $|q/p| \neq 1$ and that at least one of the mixing parameters x and y is non-zero.

CPV in the interference

In the case of a common final state f is shared by the M^0 and the $\overline{M^0}$ meson, the *CP* symmetry can be violated in the interference between the decay without mixing, $M^0 \to f$, and the decay with mixing, $M^0 \to \overline{M}^0 \to f$. The time-dependent decay amplitude of an initially pure M^0 state decaying to a final state f, accessible from both M^0 and $\overline{M^0}$ states, is given by

$$\langle f|H|M^0(t)\rangle = \mathcal{A}(M^0 \to f)g_+(t) + \mathcal{A}(\overline{M}^0 \to f)\frac{q}{p}g_-(t). \tag{2.47}$$

The time dependent decay rate is proportional to $|\langle f|H|M^0(t)\rangle|^2$

$$\begin{aligned}\frac{d\Gamma}{dt}(M^0 \to f) \propto |\mathcal{A}(M^0 \to f)|^2[(1-|\lambda_f|^2)cos(x\Gamma t) + (1+|\lambda_f|^2)cosh(y\Gamma t)\\ - 2Im(\lambda_f)sin(x\Gamma t) + 2Re(\lambda_f)sinh(y\Gamma t)],\end{aligned} \tag{2.48}$$

where

$$\lambda_f = \frac{q}{p}\frac{\mathcal{A}(\overline{M}^0 \to f)}{\mathcal{A}(M^0 \to f)}. \tag{2.49}$$

Analogous calculations apply for an initially pure $\overline{M^0}$ state.

The *CP* symmetry can be violated in the interference when

$$Im(\lambda_f) + Im(\lambda_{\overline{f}}) \neq 0. \tag{2.50}$$

For final *CP* eigenstates, the condition simplifies to

$$Im(\lambda_f) \neq 0. \tag{2.51}$$

2.3 *CPV* in the charm sector

The study of *CP* asymmetry in the decay of an up-type quark is possible for the charm quark. The others two up-type quark do not provide useful information: in the hadronization the up quark creates π^0, that is a *CP* eigenstate, and the top quark decays before it can hadronize. Then the study of *CP* asymmetry in *c*-hadrons (particles containing at least one charm quark) is particularly interesting, pushing multiple searches for *CPV* in several processes involving *c*-hadrons. However any evidence for *CPV* in the charm sector remained unobserved for decades. Only in 2019 the LHCb collaboration observed *CP* violation in $D^0 \to \pi^+\pi^-$ and $D^0 \to K^+K^-$ decays [1]. The result, $\Delta A_{CP} = (-15.4 \pm 2.9) \times 10^{-4}$, is generally believed to be compatible with a SM origin [16–19], but the present level of theoretical understanding does not allow a very precise comparison, due to the presence of strong-interaction effects which are difficult to compute, and the lack of experimental data beyond this single measurement.

The theoretical predictions in this field are not straightforward since the masses of *c*-hadrons, $\mathcal{O}(2\,\text{GeV}/c^2)$, belong to a range where non-perturbative hadronic physics is operative and the phenomenological approximations commonly used in the strange and bottom sectors are of little help. Lattice-QCD requires high computational power for the determination of relevant charm properties, and the computational power actually available is not enough. Also exclusive approaches that rely on explicitly accounting for all possible intermediate states, do not provides precise predictions; the c-hadrons decay in many final states, therefore precise measurements of amplitudes and strong phases are needed. All these peculiarities lead to large uncertainties in the theoretical picture of charm-dynamics.

Several experiments have contributed and are still contributing to the study of charm physics. From fixed target experiments like E691 and FOCUS, to e^+e^- machines and hadron colliders. About e^+e^- machines, the majority of the results have come from the CLEO, BaBar and Belle experiments, which operated at the $\varUpsilon(4S)$ resonance (corresponding to center-of-mass energies of approximately 10.6 GeV) producing $B^0\overline{B}^0$ and B^+B^- pairs. From 2011 also the BESIII experiment is contributing to this field, operating at a centre-of-mass energy between $3.8-4.6\,\text{GeV}$. BESIII can not collect as much data as BaBar and Belle, but collects data at charm threshold have powerful advantages over the data at $\varUpsilon(4S)$ threshold. Events produced at that energy are extremely clean and the signal/background ratio is optimal.

The experiments performed at e^+e^- colliders have the advantage of operating in a clean environment, where the level of background is low and where its easier to control systematic uncertainties. However, the production cross section for $c\overline{c}$ is much higher at hadron colliders The cross section for $c\overline{c}$ pair production is $\sigma \sim 1.3\,\text{nb}$ at the $\varUpsilon(4S)$ resonance [20], while at LHCb (pp collisions at a center-of-mass energy of 13 TeV in the range $0 < p_{\mathrm{T}} < 8\,\text{GeV}/c$ and $2 < \eta < 4$) $\sigma(pp \to c\overline{c}X) = (2940 \pm 3 \pm 180 \pm 160)\,\mu\text{b}$ [21]. With this advantage LHCb is playing a major role in the charm physics sector.

2.3.1 Contribution of LHCb on *CPV* observation in charm

Since the start of data-taking in 2011, LHCb has collected ~ 1 billion of D^0 decays which have contributed to increase physics knowledge of the charm sector. One milestone placed by LHCb is the first observation of $D^0 - \overline{D}^0$ oscillations from a single measurement [22], published in 2013.

Evidence of $D^0 - \overline{D}^0$ oscillations had already been reported by BaBar Belle and CDF using different D^0 decay channels [23–25], but only the combination of these measurements provides confirmation of $D^0 - \overline{D}^0$ oscillations with a significance greater than 5 standard deviations [26]. The LHCb result excludes the no-mixing hypothesis with a probability corresponding to 9.1 standard deviations, representing the first observation of $D^0 - \overline{D}^0$ oscillations from a single measurement.

A second milestone is the already cited observation of *CPV* in $D^0 \to \pi^+\pi^-$ and $D^0 \to K^+K^-$ decays in 2019 [1].

More in general the contribution of LHCb is perceptible looking at the world average plots [26]. Figure 2.4 shows the world average of y_{CP} (left) and A_Γ (right).

The resolution achieved by LHCb are lower than the others experiment, in the case of A_Γ the world average is dominated by LHCb measurements.

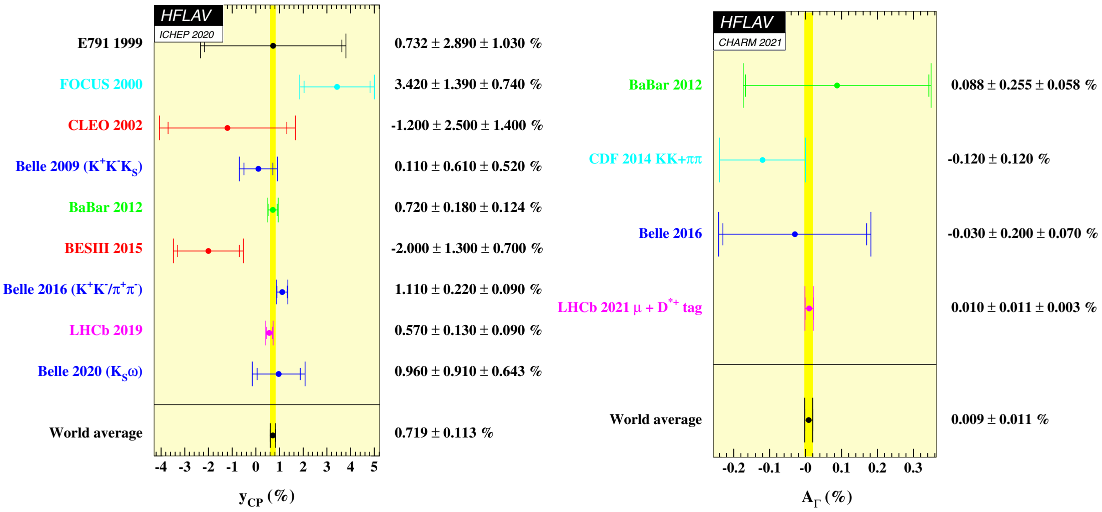


Figure 2.4: y_{CP} world average (left). A_Γ world average (right). From Ref. [26].

Figure 2.5 summarises the actual knowledge of mixing parameters and CPV in mixing and interference. On the top it shows the value of x and y mixing parameters, in no-mixing hypothesis $x = y = 0$. On the bottom it shows the value of $\mathrm{Arg}(q/p)$ and $|q/p| - 1$, in case of CPV in the interference between mixing and decay, $\mathrm{Arg}(q/p) \neq 0$ and $|q/p| - 1 \neq 0$ is an evidence of CPV in mixing. In 2021 LHCb observed a non-zero mass difference in the D^0 meson system, with a significance exceeding 7 standard deviations [27]. The data are consistent with CP symmetry and improve existing constraints on the associated parameters. Figure 2.5 quantifies the impact of this observation on world average of x, y, $\mathrm{Arg}(q/p)$, and $|q/p| - 1$. On the left hand side the plots do not include this measurements, and on the right hand side they do.

2.3.2 Future role of LHCb on CPV in charm sector

Many LHCb measurements are limited by the statistical uncertainty. For this reason, all the sub-detectors and the trigger system were updated, and this year LHCb will start to acquire new data at an increased instantaneous luminosity. The plan is to collect 50 fb^{-1} of data during Run 3 and Run 4.

The observation of CPV in charm decays was performed measuring $\Delta A_{CP} = A_{CP}(K^+K^-) - A_{CP}(\pi^+\pi^-)$ with a sensitivity of 2.9×10^{-4} [1]. With the new data, the sensitivity on ΔA_{CP} is estimated to be 7×10^{-5} [28]. Moreover the statistical sensitivity on the singles A_{CP} will reach the level of 1.5×10^{-4}, disclosing the possibility of observe CPV also in the single channels and not only on the difference. Also the other measurements on x, y, $\mathrm{Arg}(q/p)$, $|q/p|$, and A_Γ will have similar improvements.

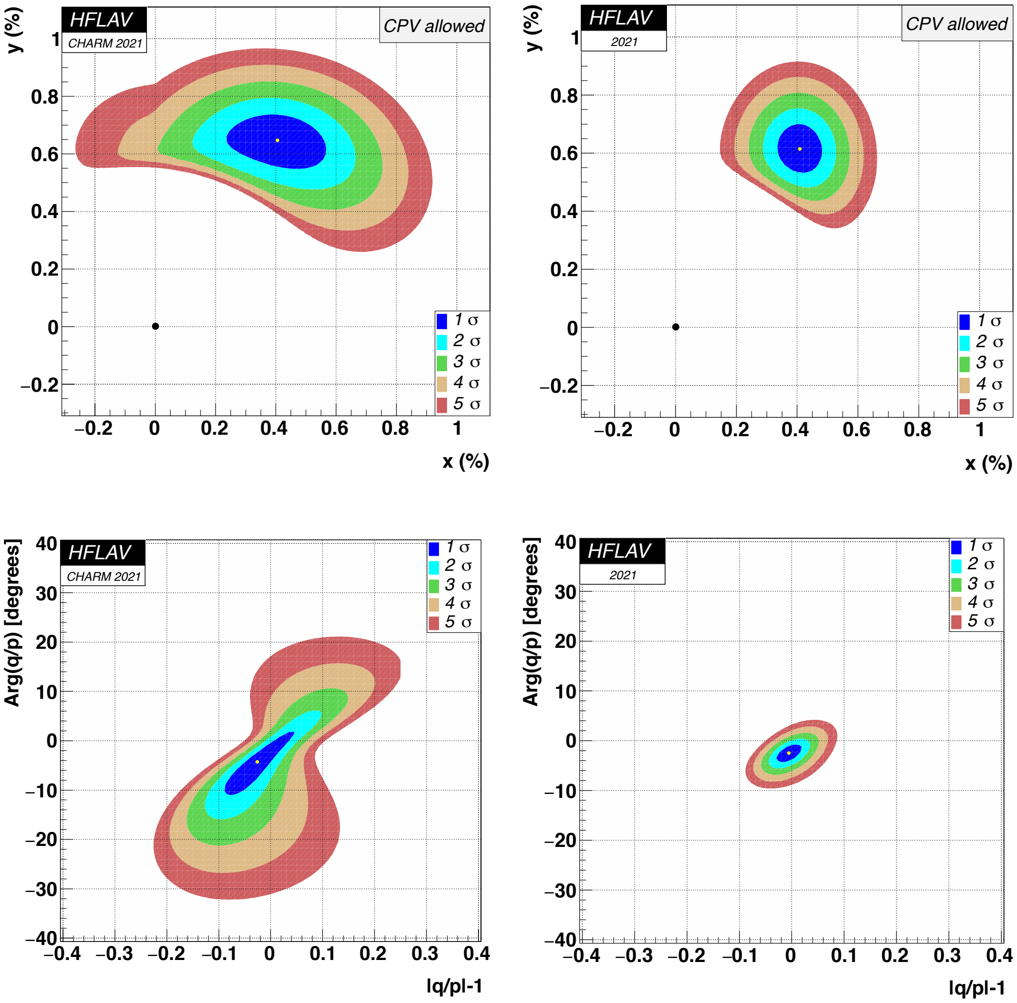


Figure 2.5: Value of x and y mixing parameters (top). Value of Arg(q/p) and $|q/p| - 1$ (bottom). Result from a global fit, including all measurements except the LHCb observation of a non-zero mass difference in the D^0 meson system (left), and including also this measurement (right). From Ref. [26].

There are strong arguments to continue flavour physics studies also after these runs. The LHCb submitted an expression of interest proposing a second upgrade [29]. The aim is to further increase instantaneous luminosity and then collect 300 fb^{-1} of data.

2.3.3 D^0 decays into two neutral kaons

As already mentioned, the *CPV* measured $D^0 \to \pi^+\pi^-$ and $D^0 \to K^+K^-$ decays, while generally believed to be compatible with a SM origin [16–19], still remains of somewhat uncertain origin, due to the present level of precision of theoretical calculations and the lack of experimental data for others decay channel, not allowing a very stringent comparison. It is therefore of paramount importance to detect and measure *CPV* effects in additional charm decay modes.

Amongst possible decay channels, the $D^0 \to K^0_S K^0_S$ one is very promising, because the majority of predictions places the value of $A_{CP}(K^0_S K^0_S)$ at the level of 10^{-3} [30–32], with 1.1% (C.L. 95%) as upper limit [33]. In 2021 LHCb published the most precise measurement of *CP* asymmetry in this channel [34], performed with data collected during Run 2. The measure is dominated by the statistical uncertainty: $A_{CP}(K^0_S K^0_S) = (-3.1 \pm 1.2 \pm 0.4 \pm 0.2)\%$, where the first uncertainty is statistical, the second is systematic, and the third is due to the uncertainty on the CP asymmetry of the calibration channel. The high statistical uncertainty is due to low trigger efficiencies on K^0_S. The K^0_S particles have a relatively high lifetime of $\tau \sim 0.9 \times 10^{-10}$ s; therefore, in LHCb they often decay outside the VErtex LOcator (VELO) acceptance, preventing them from being reconstructed in the first trigger level. As a consequence, the number of $D^0 \to K^0_S K^0_S$ decays collected by LHCb is more limited than for other D^0 decay channels.

Recent theoretical works predicts a *CP* asymmetry of order 10^{-3} [35, 36] for $D^0 \to K^0 \overline{K}^{*0}$ and $D^0 \to \overline{K}^0 K^{*0}$ decays. While this might be smaller than for the case of $D^0 \to K^0_S K^0_S$, the prompt decay $K^{*0} \to K^+ \pi^-$ produces charged tracks pointing directly to the D^0 decay vertex, allowing to trigger it more efficiently and collect larger samples. Therefore the samples currently available to LHCb are larger, making them an attractive target for a focused work.

The $D^0 \to K^0 \overline{K}^{*0}$ and $D^0 \to \overline{K}^0 K^{*0}$ decays (charge conjugate decays implied throughout this thesis, unless explicitly specified) belong to the singly Cabibbo-suppressed decays (SCS) category. Therefore the decay amplitudes of these decays involve the CKM elements $\lambda_q \equiv V^*_{cq} V^*_{uq}$, with $q = d, s, b$. Using the relation $\lambda_d + \lambda_s + \lambda_b = 0$ the amplitude of the decay d can be expressed as:

$$\mathcal{A}(d) \equiv \lambda_{sd} \mathcal{A}_{sd}(d) - \frac{\lambda_b}{2} \mathcal{A}_b(d), \tag{2.52}$$

where $\lambda_{sd} = \frac{\lambda_s - \lambda_d}{2}$. At first order in $|\lambda_b|/|(\lambda_s - \lambda_d)|$ the direct *CP* asymmetry reads

$$A^{dir}_{CP}(d) \equiv \frac{|\mathcal{A}(d)|^2 - |\overline{\mathcal{A}}(d)|^2}{|\mathcal{A}(d)|^2 + |\overline{\mathcal{A}}(d)|^2} \tag{2.53}$$

$$\equiv Im \frac{\lambda_b}{\lambda_{sd}} Im \frac{\mathcal{A}_b(d)}{\mathcal{A}_{sd}(d)}. \tag{2.54}$$

$\mathcal{A}_{sd}(d)$ and $\mathcal{A}_b(d)$ can be written as the sum of topological amplitudes; in the limit of exact SU(3) symmetry these are the *tree* (T), *colour-suppressed tree* (C), *exchange* (E), *annihilation* (A), *penguin* (P_q), and *penguin annihilation* (PA_q) amplitudes. The latter two topologies involve a loop with the indicated internal quark $q = d, s, b$.

$Im(\lambda_b/\lambda_{sd})$ defines the typical size of $|A^{dir}_{CP}(d)|$. It is a pure CKM phase, and its value is equal to $-6 \cdot 10^{-4}$.

From the branching ratios [37]

$$\mathcal{B}^{exp}(D^0 \to K^0_S \overline{K}^{*0}) = (0.9 \pm 0.2) \cdot 10^{-4} \tag{2.55}$$

$$\mathcal{B}^{exp}(D^0 \to K^0_S K^{*0}) = (1.1 \pm 0.2) \cdot 10^{-4}, \tag{2.56}$$

the value of $|\mathcal{A}_{sd}(d)|$ is extracted, empirically finding that it is small.

$|\mathcal{A}_b(d)|$ involves the large topological amplitude E. This amplitude involves no loop and a global fit to measured branching ratios supports a large value of $|E|$ [38], comparable to $|T|$. Therefore its value is enhanced.

The decomposition of $\mathcal{A}(D^0 \to K^0\overline{K}^{*0})$ and $\mathcal{A}(D^0 \to \overline{K}^0K^{*0})$ shows that they depends on exchange and penguin annihilation topologies only:

$$\mathcal{A}_{sd}(D^0 \to \overline{K}^0K^{*0}) = E_P - E_V + E_{P3} - E_{V1} - E_{V2} - PA_{PV}^{break}, \tag{2.57}$$

$$\mathcal{A}_b(D^0 \to \overline{K}^0K^{*0}) = -E_P - E_V - E_{P3} - E_{V1} - E_{V2} - PA_{PV}, \tag{2.58}$$

$$\mathcal{A}_{sd}(D^0 \to K^0\overline{K}^{*0}) = -E_P + E_V - E_{P1} - E_{P2} + E_{V3} - PA_{PV}^{break}, \tag{2.59}$$

$$\mathcal{A}_b(D^0 \to K^0\overline{K}^{*0}) = -E_P - E_V - E_{P1} - E_{P2} - E_{V3} - PA_{PV}, \tag{2.60}$$

where the subscripts P stands for the pseudo-scalar K^0 and $\overline{K}^0$, and the subscripts V stands for the vectorial K^{*0} and $\overline{K}^{*0}$. The contributions from PA_P and PA_V cannot be distinguished from each other, therefore $PA_{PV} \equiv PA_P + PA_V$. The corresponding topological diagram are shown in Figure 2.6 and 2.7.

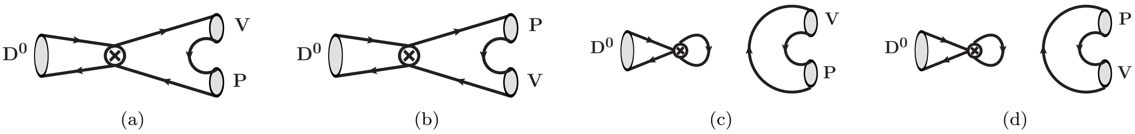


Figure 2.6: SU(3)$_F$-limit topological amplitudes E_P (a), E_V (b), PA_{Pq} (c), and PA_{Vq} (d) contributing to $D^0 \to K^0\overline{K}^{*0}$ and $D^0 \to \overline{K}^0K^{*0}$. The subscripts P stands for the pseudo-scalar K^0 and $\overline{K}^0$, and the subscripts V stands for the vectorial K^{*0} and $\overline{K}^{*0}$. The q in PA_{Pq} and PA_{Vq} labels the quark running in the loop at the weak vertex. From Ref. [35].

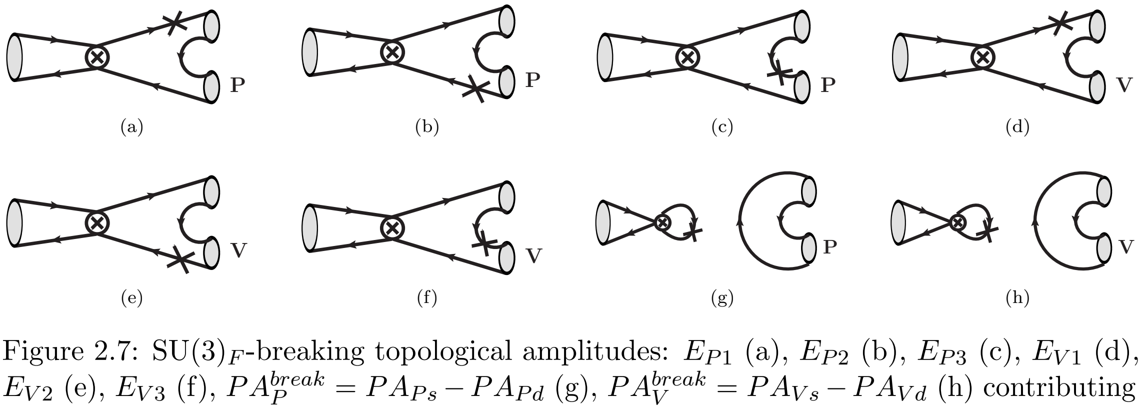


Figure 2.7: SU(3)$_F$-breaking topological amplitudes: E_{P1} (a), E_{P2} (b), E_{P3} (c), E_{V1} (d), E_{V2} (e), E_{V3} (f), $PA_P^{break} = PA_{Ps} - PA_{Pd}$ (g), $PA_V^{break} = PA_{Vs} - PA_{Vd}$ (h) contributing to $D^0 \to K^0\overline{K}^{*0}$ and $D^0 \to \overline{K}^0K^{*0}$. The subscripts P stands for the pseudo-scalar K^0 and $\overline{K}^0$, and the subscripts V stands for the vectorial K^{*0} and $\overline{K}^{*0}$. The contribution $PA_P^{break} and PA_V^{break}$ cannot be distinguished from each other. From Ref. [35].

Several SU(3)$_F$-breaking topologies are present. However the experimental data shows that their effect are small [39]. In the SU(3)$_F$ symmetry limit, Equation 2.54

can be written as:

$$A_{CP}^{dir}(D^0 \to K^0 \overline{K}^{*0}) = \frac{Im(\lambda_b)}{\lambda_{sd}} \mathrm{Im}\Big(\frac{E_P + E_V + PA_{PV}}{E_P - E_V}\Big) \tag{2.61}$$

$$A_{CP}^{dir}(D^0 \to \overline{K}^0 K^{*0}) = -\frac{Im(\lambda_b)}{\lambda_{sd}} \mathrm{Im}\Big(\frac{E_P + E_V + PA_{PV}}{E_P - E_V}\Big). \tag{2.62}$$

Extracting the value of $|E_P - E_V|$, $|E_P|$, amd $|E_V|$ from the literature [39,40], the maximum value of $|A_{CP}^{dir}|$ near the K^* resonance is

$$|A_{CP}^{dir}| \lesssim 0.003. \tag{2.63}$$

LHCb performed a time-integrated amplitude analysis [39], using *pp* collisions data corresponding to an integrated luminosity of $3.0\,\mathrm{fb}^{-1}$ collected during 2011 and 2012 (LHC Run 1) at center-of-mass energies $\sqrt{s} = 7$ TeV and 8 TeV, respectively. The Run 1 sample contains about 189k signal decays, that was about a hundred times larger than the previous amplitude study of the same modes performed by the CLEO collaboration [41]. The LHCb analysis produced a quite detailed amplitude model of the $D^0 \to K_S^0 K^- \pi^+$ and $D^0 \to K_S^0 K^+ \pi^-$ decays. While its agreement with data does not match the statistical precision of the sample, the quality of fit is broadly comparable to other amplitude analyses with similar sample sizes, and it is adequate for the use I make in the present work, as discussed later. The Run 1 analysis also included a fit with floating *CP* violating parameters, but this was not its main focus, and the probability of observing an effect was low with the statistics available at the time.

The main goal of my work is instead to measure the *CP* asymmetry of $D^0 \to K_S^0 \overline{K}^{*0}$ and $D^0 \to K_S^0 K^{*0}$ decays, separately from other components leading to the same final state, while avoiding to introduce excessive model dependencies. I put this specific resonance at the centre of my work partly for particularly large asymmetry pointed out by the theoretical studies, and partly because it has some unique experimental features (discussed later) that make it a particularly interesting subject and have been a driver for some of my analysis choices. However, the same technique can be applied to perform asymmetry measurements also in other sub-channels that are less peculiar but still worth studying - in particular the $K^{*\pm} K^{\mp}$ modes, that have recently been indicated as potentially carrying a significant *CP* asymmetry [36].

Chapter 3

The LHCb Run 2 detector

3.1 The Large Hadron Collider

The Large Hadron Collider (LHC) is a superconducting proton-proton and heavy-ion collider located at the CERN laboratory [42], on Swiss-French state border. The LHC is installed in a 27 km long circular tunnel, about 100 m underground, that previously housed the LEP. Protons are extracted from hydrogen gas and their energy are gradually increased by a series of accelerator machines, shown in Figure 3.1. Extracted protons are first accelerated by the Linac 2 up to an energy of 50 MeV, then by the Booster up to an energy of 1.4 GeV. The Proton Synchrotron (PS) and Super Proton Synchrotron (SPS) respectively accelerate them to an energy of 25 GeV and 450 GeV. Finally protons are injected in the LHC.

In the LHC, two proton or ion beams circulate in opposite directions in two separate beam pipes accelerated by radio-frequency (RF) cavities. Beams are bent by more than 1200 superconducting dipole magnets 15 m long, cooled at temperature of 1.9 K by 120 tons of superfluid helium, which generate a magnetic field of 8.3 T.

Beams collide in four points placed along the LHC ring, where the detectors of the four major LHC experiments are installed. ATLAS and CMS are general-purpose experiments, while ALICE and LHCb are specifically dedicated to heavy-ion and heavy-flavor physics, respectively. Other smaller experiments are installed along the collider.

Proton beams are split in bunches each one consisting of about 10^{11} protons, and are time-spaced for a multiple of 25 ns corresponding to a bunch-crossing rate up to 40 MHz. The maximum number of bunches per beam is 2808, so the average bunch-crossing rate is $\sim$ 30 MHz. The peak istantaneous luminosity of the LHC project design is of $\mathcal{L} = 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ at a center of mass energy $\sqrt{s} = 14\,\mathrm{TeV}$. The design parameters will be achieved in 2022 during the Run 3. The energy at the center of mass was $\sqrt{s} = 7\,\mathrm{TeV}$ in 2010 and 2011, while in 2012 it was raised to 8 TeV. After a two years shut-down, in which several upgrades and checks to magnets system were done, LHC was restarted with an energy at the center of mass of 13 TeV, maintained for the entire Run 2: from 2015 to the end of 2018.

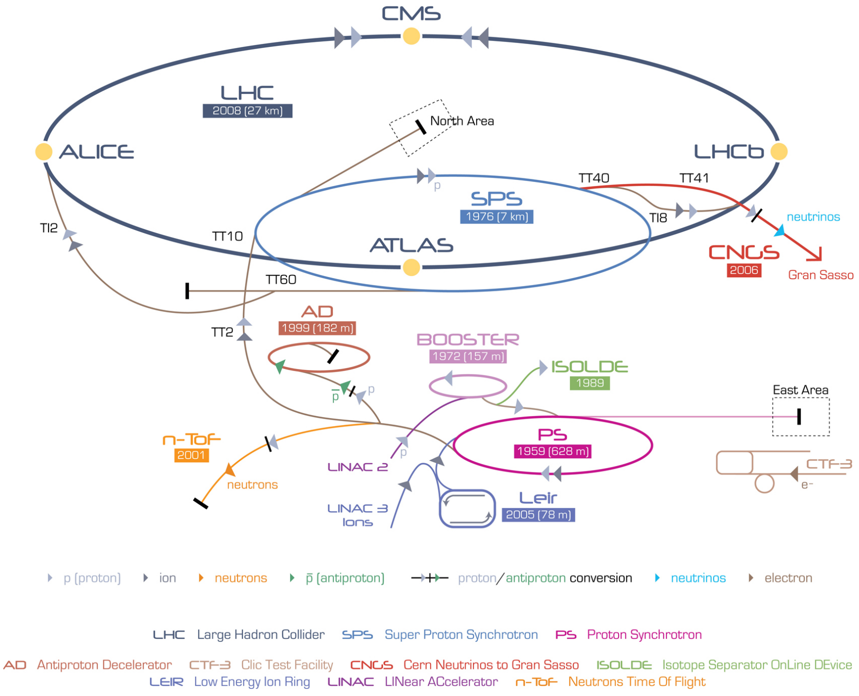


Figure 3.1: CERN Accelerator Complex.

3.2 The LHCb detector in Run 2

The LHCb detector [43, 44] is a single-arm forward spectrometer covering the pseudorapidity range $2 < \eta < 5$, designed for the study of particles containing b- or c-quarks. The LHCb detector layout, shown in figure 3.2, is motivated by the fact that at high energies both b-hadrons are produced in the same forward or backward cone, as shown in figure 3.3.

LHCb adopts a right-handed coordinate system with the x-axis pointing toward the centre of the LHC ring, the y-axis pointing upwards, and the z-axis pointing along the beam direction.

The LHCb detector is carefully designed to reconstruct heavy-flavour decays in the high-background environment of a hadron collider. The charm and beauty hadrons are highly boosted in the laboratory frame. Having lifetimes of $\mathcal{O}(0.1-1\,\text{ps})$, they can fly several millimeters before decaying. Their relatively long lifetime is a distinctive feature that can be exploited by detectors with sufficient vertex resolution. The vertex resolution is also essential for measurements such as neutral-meson oscillations and time-dependent CP asymmetries, where the lifetime of the studied hadron has to be measured with high precision. True heavy-flavour decays are discriminated from residual backgrounds also performing high-resolution measurements of the particles momentum and of their invariant masses. These requirements are achieved with an advanced charged particle tracking system and a particle identification system.

The tracking system includes a magnet and four different detectors: the VErtex

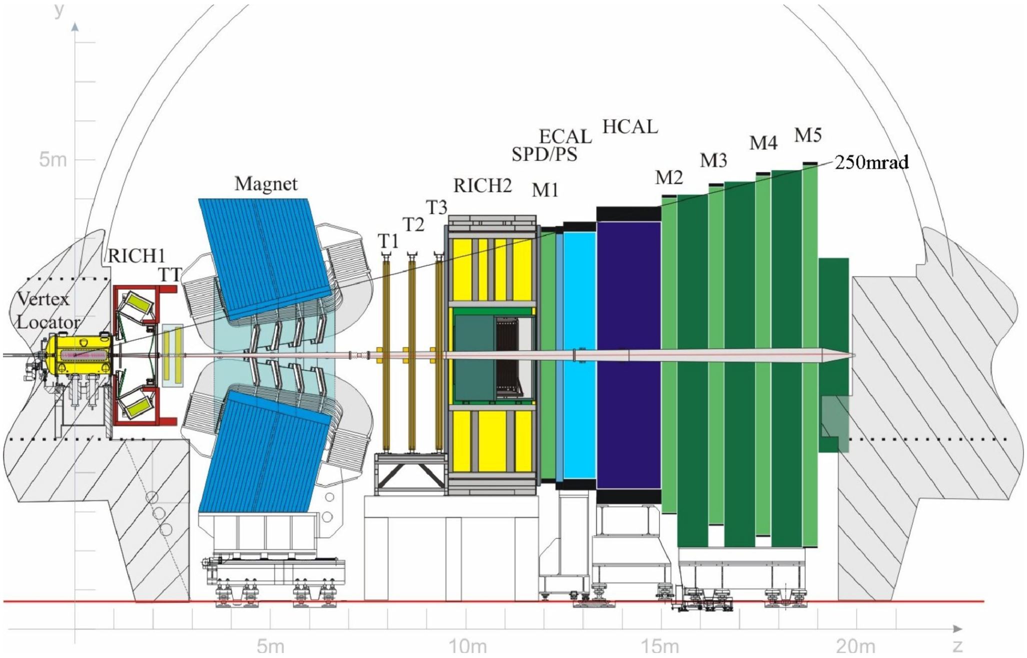


Figure 3.2: Layout of LHCb detector.

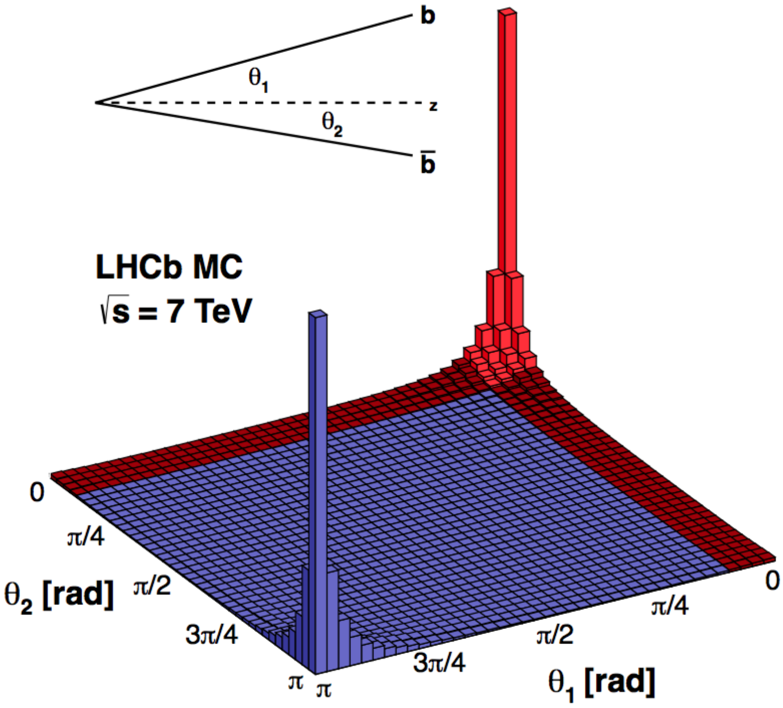


Figure 3.3: Angular correlation between b and $\bar{b}$ quarks in $b\bar{b}$ pair production, simulated with Pythia event generator.

LOcator (VELO) and the Tracker Turicensis (TT) upstream of the magnet, and the Inner Tracker (IT) and Upstream Tracker (UT) (arranged in the T1-T3 tracking stations) downstream of the magnet. The particle–identification system includes several detectors exploiting different technologies: two ring imaging Cherenkov (RICH) detectors, the calorimeter system, and the muon detectors. The calorimeter system is formed by the scintillator pad detector (SPD), the pre-shower detector (PS), the electromagnetic calorimeter (ECAL), and the hadron calorimeter (HCAL).

When the beams intersect, multiple primary pp interactions may occur causing high particle occupancy in the detector. The nominal LHC luminosity value is reduced to $\mathcal{L} = 4 \cdot 10^{32}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ in the LHCb intersection point. Lower luminosity is obtained by appropriately defocusing the beams by moving them apart transversely. This transverse separation is progressively modified during a fill, to keep the luminosity constant as the beam current decreases. The chosen luminosity value is optimised to obtain one or two inelastic interactions per bunch crossing according to trigger bandwidth, and for limit radiation damage. Figure 3.4 shows the integrated LHCb luminosity collected in Run 1 and in Run 2.

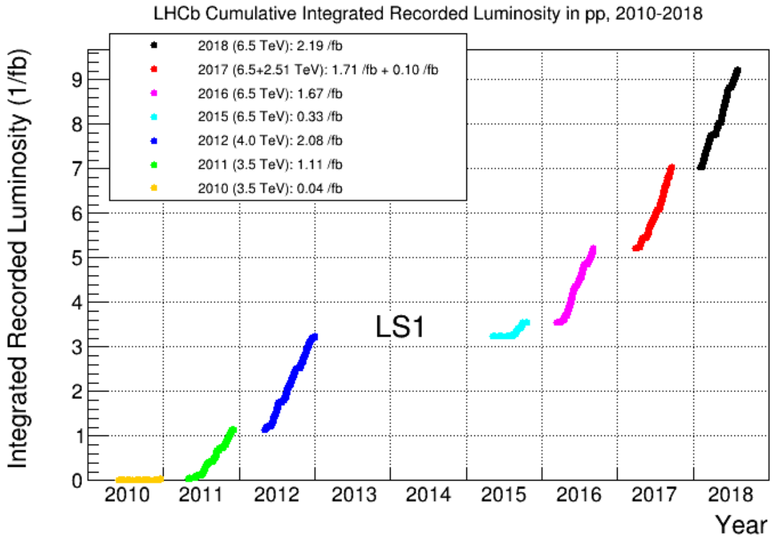


Figure 3.4: Integrated LHCb luminosity collected in Run 1 and Run 2.

3.2.1 Tracking system

The tracking system must provide accurate spatial measurements of charged particle tracks, in order to allow quantities such as charge, momentum, and vertex locations to be determined.

VErtex LOcator

The VErtex LOcator (VELO) is a silicon strip detector that measures charged particle trajectories in the region closest to the interaction point [45]. Its main purpose is to reconstruct primary and secondary vertexes with a spatial resolution smaller than typical decay lengths of b- and c-hadrons in LHCb ($c\tau \approx 100 - 500\,\mu\mathrm{m}$). It plays a fundamental role for discriminating heavy flavors signals from the underlying background.

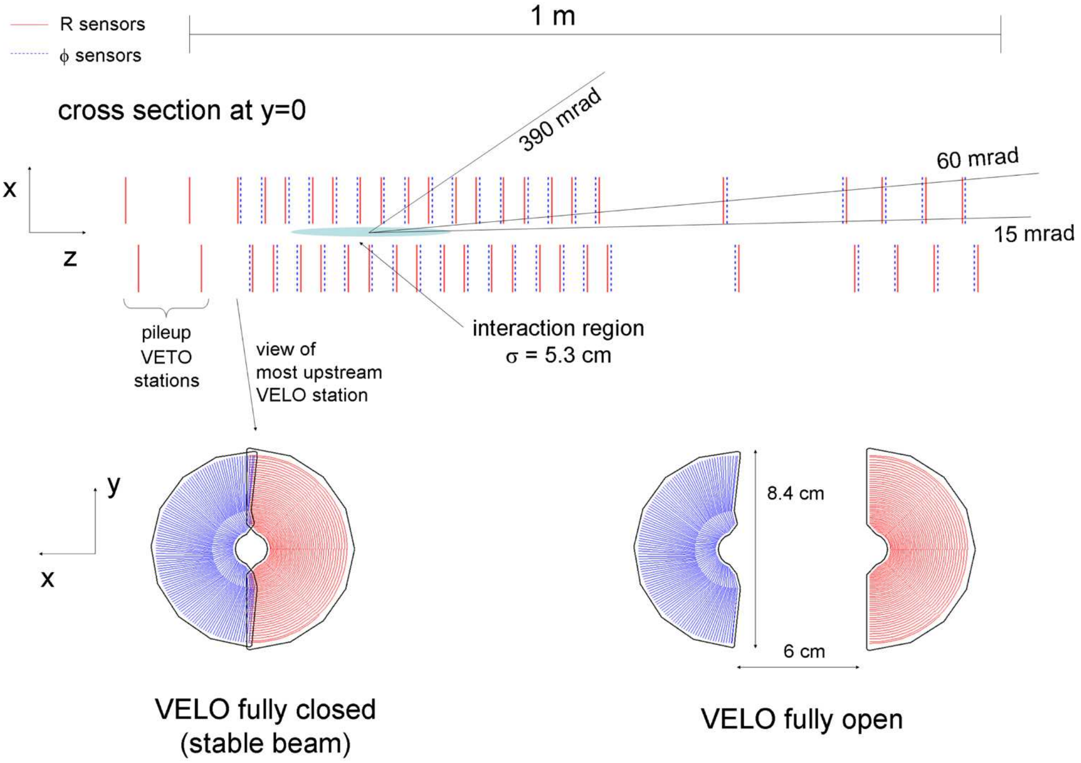


Figure 3.5: Representation of VELO detector, with a transverse view of a VELO station in closed and open configurations.

The VELO consists of 21 disk-shaped stations installed along the beam axis inside the beam pipe, both upstream ($z > 0\,\mathrm{cm}$) and downstream ($z < 0\,\mathrm{cm}$) of the nominal interaction point. Figure 3.5 shows the layout of the system. Stations placed at $z > 0\,\mathrm{cm}$ provide precise measurements of vertexes positions. While the stations at $z < 0\,\mathrm{cm}$ constitute the pile-up veto system, which provides position of primary vertices candidates along the beam-line and measures the total backward charged track multiplicity. The stations are made by two type of silicon strip sensors, the r and ϕ sensors, arranged with radial and azimutal segmentation to measure r and ϕ particle intersection coordinates. Each station is divided into two retractile halves, called modules, as shown in Figure 3.5. Each halves consists of both r and ϕ sensors. VELO veto stations consist of r sensors only. The retractile halves allow to move the sensors away from the beam, to do not damage silicon sensors during LHC injection phases, when VELO stations are "opened" and the sensors have a minimum distance of 30 mm from the beam axis, instead, when stable beams are circulating for data taking, stations are "closed" and the sensors reach a minimum distance of 5 mm from the beam axis.

Both r and ϕ sensors are centered around the nominal beam position, and are covering a region between 8 and 42 mm in radius. Their sensitive area is thick 300 μm. The r sensors consist of semicircular concentric strips that are divided in four 45° sectors to reduce occupancy. The pitch increases linearly from 38 μm at the innermost radius to 102 μm at the outermost radius. The ϕ sensors are subdivided in two concentric regions: the inner one covers a radius r between 8 and 17.25 mm, the outer one covers r between 17.25 and 42 mm with pitch linearly increasing from the center. ϕ sensors are designed with an angular tilt of +10° in the inner region and -20° in the outer region, with respect to the radial direction; for adjacent sensors, the tilt is reversed. This layout is designed to improve pattern recognition and to better distinguish noise from genuine hits.

The whole VELO is placed inside the LHC vacuum pipe in order to place the sensors as close to the primary interactions as possible. To protect the integrity of the primary LHC vacuum system, the sensors are separated from the beam volume by a 0.3 mm aluminium shield known as "RF-foil".

The individual hit resolution of the sensors is strongly correlated to the sensor pitch and projected angle, that is the angle perpendicular to the strip direction. Raw hit resolution varies from ≈ 10 μm for smallest pitch to ≈ 25 μm for biggest pitch.

Tracker Turicensis

The Tracker Turicensis (TT) is a silicon micro-strip detector [46]. It consists of four 150 cm × 130 cm layers, corresponding to the full LHCb angular acceptance, grouped in two stations separated by 30 cm along the beam line. The four layers are arranged in a x-u-v-x configuration. The first and last layer ("x" configuration) consist of vertical strips, while the "u" and "v" layers are rotated by ±5°. The slight rotation with respect to the vertical layers avoid the ambiguities that would arise with an horizontal orientation providing a measurement in y-direction as well. Figure 3.6 shows the TT stations in the x-u-v-x configuration.

Each sensor module is 500 μm thick with a sensitive region of 9.6 cm × 9.4 cm, carrying 512 readout strips with a pitch of 183 μm.

The TT has two purposes: to reconstruct trajectory of low-momentum particles that are swept away from the acceptance by the magnet, and to reconstruct long lived particles, as K^0_S and Λ^0, which decay outside the VELO region.

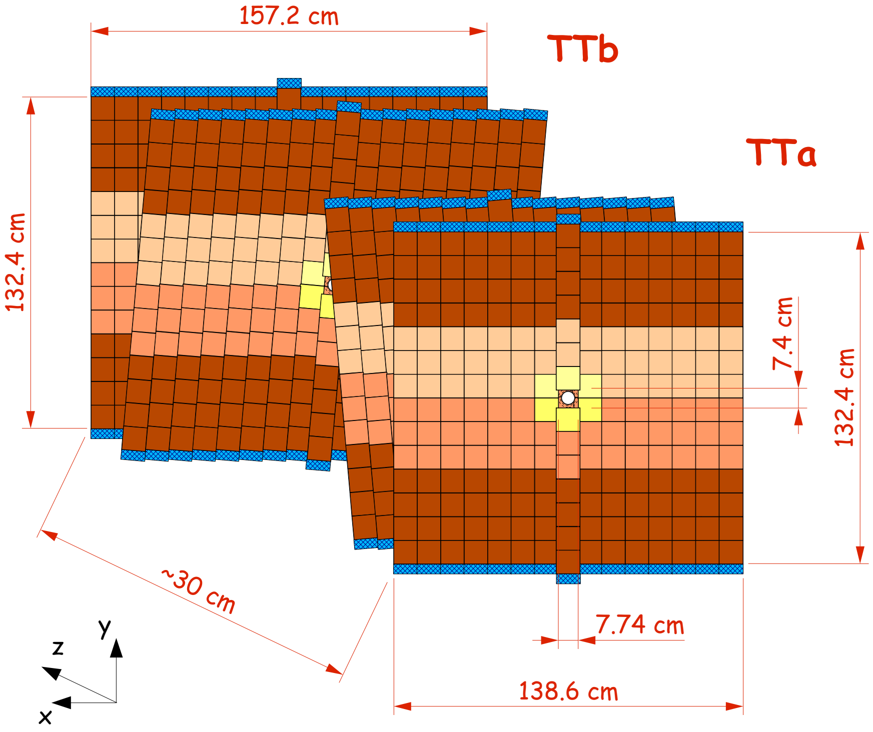


Figure 3.6: x-u-v-x configuration of TT stations.

The dipole magnet

The LHCb warm dipole magnet is placed between the TT and the T-stations, it provides bending for the measurement of the momentum of particles. It is formed by two saddle-shaped coils placed with a small angle with respect to the beam axis, in order to increase the opening window with z and follow the acceptance of the LHCb detector. Figure 3.7 shows perspective view of the magnet.

It dissipates 4.2 MW of electric power with a current of 5.85 kA in normal operating condition. The maximum magnetic field strength is above 1 T, while its integral is $\int Bdl = 4$ Tm. The field mainly develops in the y direction, hence the xz-plane can be considered with good approximation the bending plane. Before the data-taking period, a precise map of the magnetic field is obtained with Hall probes, in order to ensure good momentum resolution and consequently a good mass resolution that helps to select more efficiently processes of interest. Figure 3.8 shows the B_y component of LHCb magnetic field. A fringe field is present in the region where the tracking detectors are installed.

The LHCb magnet has a unique feature consisting into the possibility to reverse the polarity of the magnetic field (MagUp or MagDown). This allows a precise control of the charge asymmetries introduced by the detector. Particles hit preferentially one side of the detector, depending on their charges, generating large

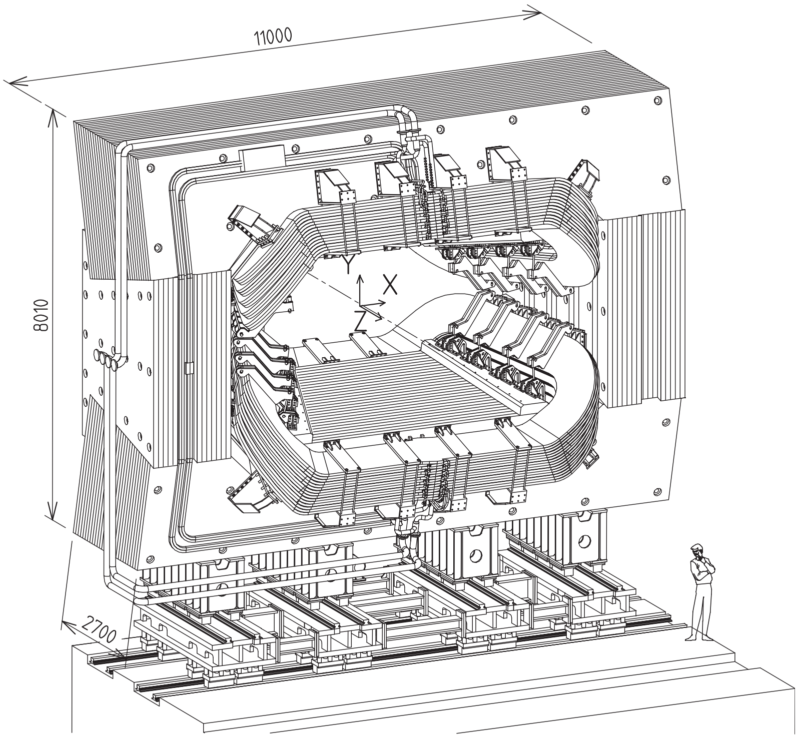


Figure 3.7: Perspective view of LHCb dipole magnet.

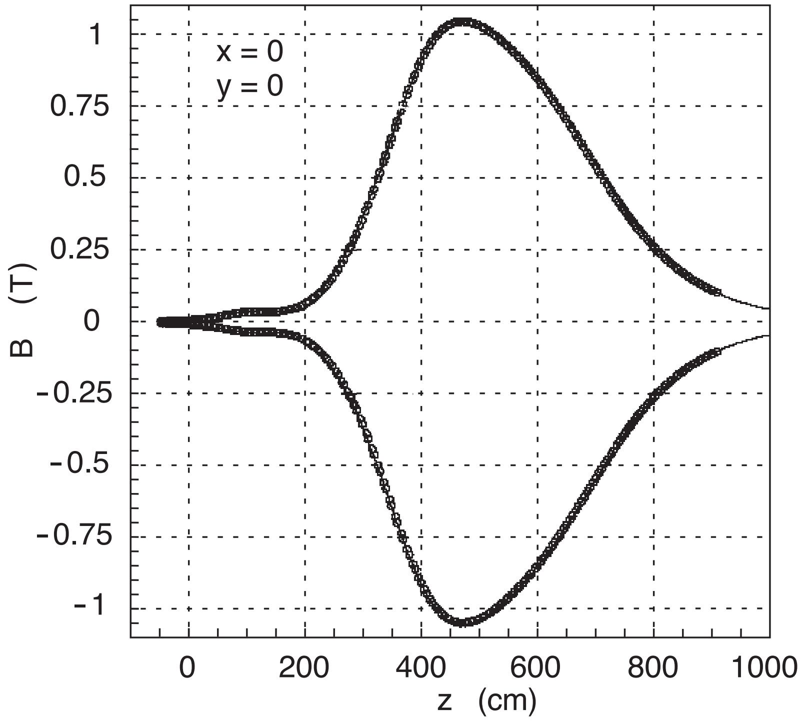


Figure 3.8: Measured B_y component of LHCb magnetic field. From Ref. [43].

detection asymmetries. If data samples collected with the two different polarities have approximately equal size and the operating conditions are stable enough, effects of detection charge asymmetries are expected to cancel.

Inner Tracker

The Inner Tracker (IT) is located downstream the dipole magnet, and it consists of 3 stations [46]. It covers an acceptance of $\sim 150-200\,\mathrm{mrad}$ in the bending plane and of $\sim 40-60\,\mathrm{mrad}$ in the yz-plane. Each station has four layers in a x-u-v-x configuration. The layers are cross-shaped, and are optimised to reconstruct tracks that passed through the magnetic field region lying near the beam axis. Figure 3.9 shows the layout of one IT layer. The IT uses the same micro-strip sensors of the TT. Single-hit resolution of this detector is of $\approx 50\,\mu\mathrm{m}$.

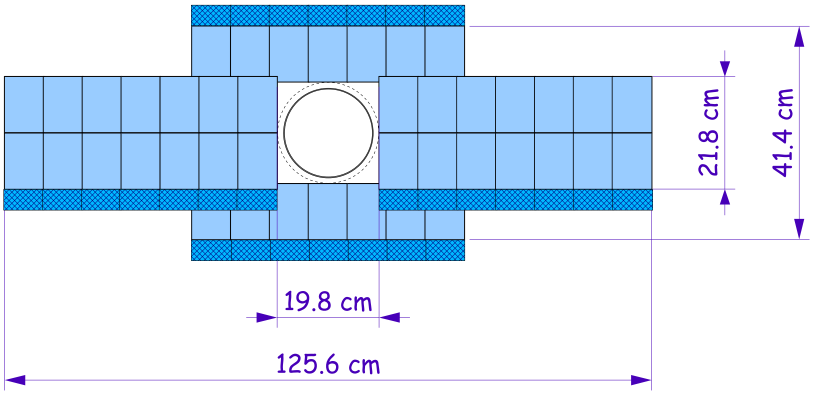


Figure 3.9: Layout of one IT layer.

Outer Tracker

The Outer Tracker (OT) is a gaseous ionisation detector consisting of straw tubes operating as proportional counters [47]. The OT is used to measure track bending in the acceptance region not covered by the IT sub-detector. The OT consist of three stations, each station is located downstream an IT station, which together form a T-station. Each OT station is subdivided in four layers x-u-v-x. Each layer is subdivided in modules, consisting of 64 straw tubes. Straw tubes are 2.4 m long with an inner diameter of 4.9 mm. They are filled with a mixture of 70% Ar and 30% CO_2 to achieve a drift time of 50 ns. The straw tubes allow to reconstruct tracks with a spatial resolution of $\approx 200\,\mu\mathrm{m}$. Figure 3.10 shows the layout of the OT together with the TT and the IT.

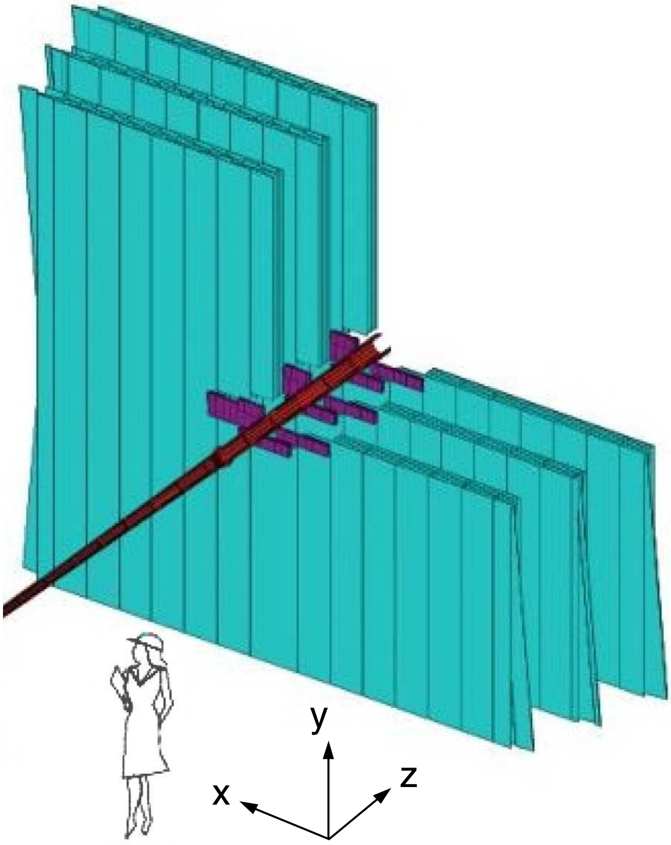


Figure 3.10: Layout of OT subdetector. The TT and the IT are highlighted in purple.

Track reconstruction

The LHCb tracking reconstruction is currently performed in stages. First, tracks are reconstructed as straight lines using the r VELO sensors. Then, hits from the ϕ VELO sensors are added to these tracks. Two different algorithms are used to combine these VELO tracks with hits in the other tracking stations: the forward and the backward. The forward method propagates VELO tracks through the magnetic field, and adds hits in the downstream tracking stations. The backward method finds straight track segment in the T-stations (track seeds) and then attempts to propagate them in the opposite direction, matching them to VELO tracks. Finally, hits from the TT are added to the track to improve the momentum resolution and reject incorrect combinations of hits.
Different types of tracks are distinguished in LHCb according to the subdetectors crossed.

- Tracks reconstructed both in VELO and T-stations subdetectors are called "long tracks", they can include also hits from the TT. They are the most relevant tracks for several LHCb analysis.

- Tracks reconstructed both in VELO and TT subdetectors are called “upstream tracks”. They are low momentum tracks swept out the LHCb acceptance by the magnetic field.
- Tracks reconstructed on TT and T-stations subdetectors are called “downstream tracks”. They are generated mainly from long-lived particles as K^0_S decaying outside the VELO region.
- Tracks reconstructed on T-stations only are called “T tracks”.
- Tracks reconstructed on VELO only are called “VELO tracks”. They are used in the primary vertex reconstruction.

Figure 3.11 shows a representation of this track classification.

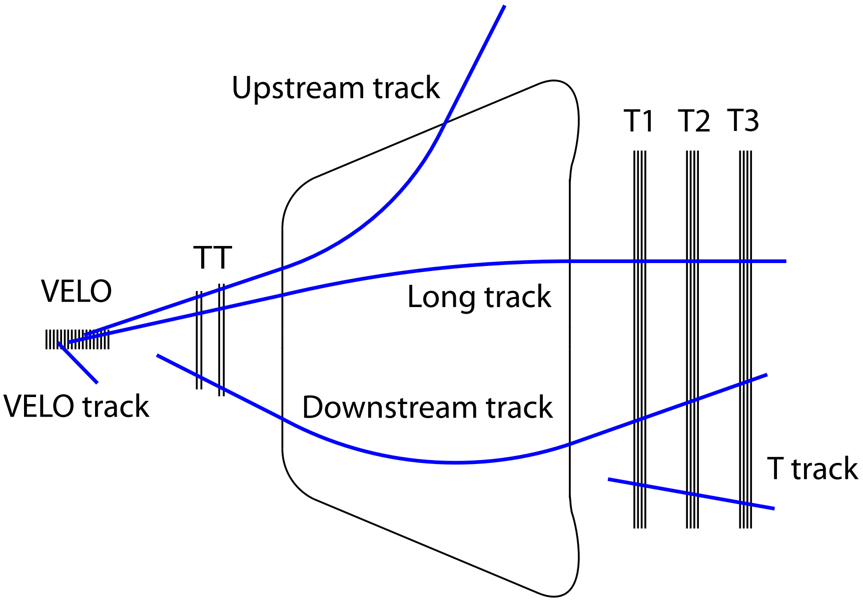


Figure 3.11: LHCb track classification.

3.2.2 Particle identification system

Particle identification plays an important role in decays studied by LHCb, classifying final states ans rejecting backgrounds. Cherenkov detectors are able to separate between charged kaons and pions, while calorimeter detectors allow identification of electrons, photons, and hadrons. Muons are identified by muon chambers. It follows a detailed explanation of these sub-detectors.

The ring imaging Cherenkov detectors

Two ring imaging Cherenkov (RICH) detectors, RICH1 and RICH2, allow to identify charged particles [48]. The two detectors configuration aims to achieve the best separation power between pion and kaon mass hypotheses, covering at the same time a large range of momenta. In particular, RICH1 aims to identify low-momentum particles ($1 - 60\,\mathrm{GeV}/c$), while RICH2 is tuned for particles with higher momenta ($15 - 100\,\mathrm{GeV}/c$).

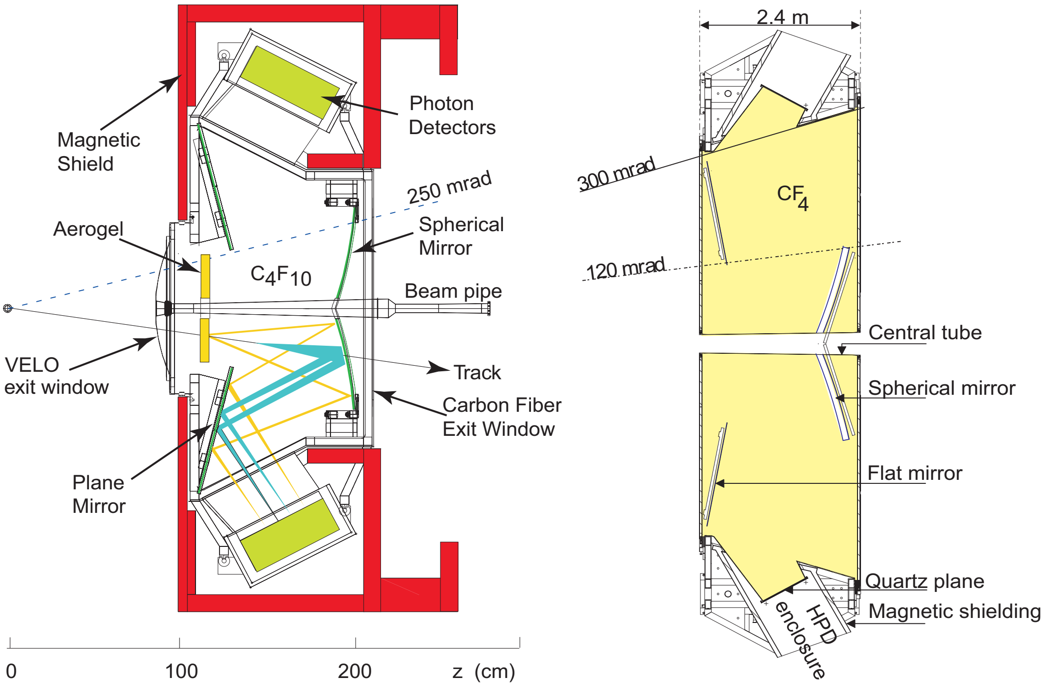


Figure 3.12: RICH1 (left) and RICH2 (right) geometries.

RICH1 is placed before the magnet, between the VELO and the TT, in order to identify the particles that are bent out from the LHCb acceptance by the magnetic field. It uses aerogel and C_4F_{10} radiators. RICH2 is placed after the last T-station and uses CF_4 as radiation medium. In both RICH detectors, a complex system of spherical and plane mirrors reflects the emitted photons outside the LHCb acceptance, where they are collected by a lattice of hybrid photon detectors (HPDs). In this way, HPDs can be shielded from the magnetic field. Figure 3.12 shows the geometry of the two RICH detectors. Figure 3.13 shows Cherenkov angles as a function of particles momentum for the different radiators used at LHCb. The $\pi - K$ separation is 90% efficient for momenta up to $30\,\mathrm{GeV}/c$.

Calorimeter detectors

Calorimeter detectors provide fast information for the low level trigger and offer identification of electrons, photons, and hadrons, together with a raw measurement

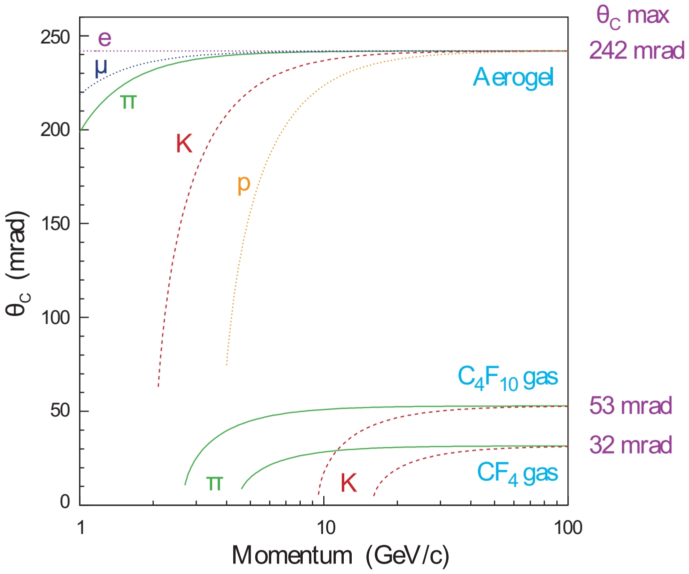


Figure 3.13: Cherenkov angles as a function of particles momentum for the different radiators used at LHCb.

of their energies and positions [49].

Calorimetric system is formed by scintillator pad detector (SPD) and pre-shower detector (PS), separated by a thin lead converter, the electromagnetic calorimeter (ECAL), and the hadron calorimeter (HCAL). All of them are placed between the first and the second muon station and cover the angular acceptance from $25 - 300(250)$ mrad in the bending (non-bending) plane.

The SPD and PS consist of two planes of scintillating pads, used at the low level electron trigger in order to reject background from charged and neutral pions and to improve electron identification. The SPD, just like a tracking detector, reveals only charged particles. Electrons and photons start showering in the lead converter, thick 2.5 radiation lengths, and produce on the PS a significantly larger signal than pions. The SPD is also used to measure the number of tracks per event, in order to veto online too crowded events.

The ECAL is made of alternated 4 mm thick scintillators tiles and 2 mm thick lead plates. The total thickness corresponds to about 25 radiation lengths, guarantees an almost complete electromagnetic shower containment and provides a good energy resolution of approximately $\sigma_E/E(\,\mathrm{GeV}) \approx 10\%/\sqrt{E(\,\mathrm{GeV})}$.

The HCAL is made of alternate 4 mm thick scintillators tiles sandwiched between 16 mm iron sheets, corresponding to about 5.6 interaction lengths. The energy resolution is $\sigma_E/E(\,\mathrm{GeV}) \approx 70\%/\sqrt{E(\,\mathrm{GeV})}$. This value is due to the poor thickness, that does not allow the complete shower containment and hence energy measurement. For this reason the HCAL is exploited only for trigger purposes and not for offline analysis.

Muon detectors

Muon detectors provide identification and transverse momentum measurement of penetrating muons for both low level and high level triggers, as well as for offline reconstruction [50]. They consist of five rectangular stations, referred to as M1-M5, placed along the beam axis and covering the angular acceptance from 20 (16) to 306 (258) mrad in the bending (non bending) plane. Each station consists of two mechanically independent halves, called A and C sides that can be horizontally moved in order to access to the beam pipe and the detector chambers, for installation and maintenance.

M1 station, which is installed between RICH2 and the calorimeter detectors, improves transverse momentum measurements for muons, since the calorimeter system introduces uncertainties due to multiple scattering. M2-M5 stations are placed downstream of the calorimeter detectors. They are interleaved with 80 cm of thick iron absorbers that select penetrating muons and result in a total thickness of 20 interaction lengths. In order to traverse the whole detector, a muon is typically required to have at minimum momentum of 6 GeV/c.

Each station is divided into four segmentated regions R1-R4, whose cells scale in the ratio 1:2:4:8 with the distance from the beam axis. All stations use multiwire proportional chamber detectors, except for the R1 region of the M1 station, where high particle density requires a radiation tolerant detector. R1 region of the M1 station uses triple gas electron multiplier detectors (triple-GEM). Figure 3.14 shows the side view of the Muon Detector and the station layout with the four regions R1–R4.

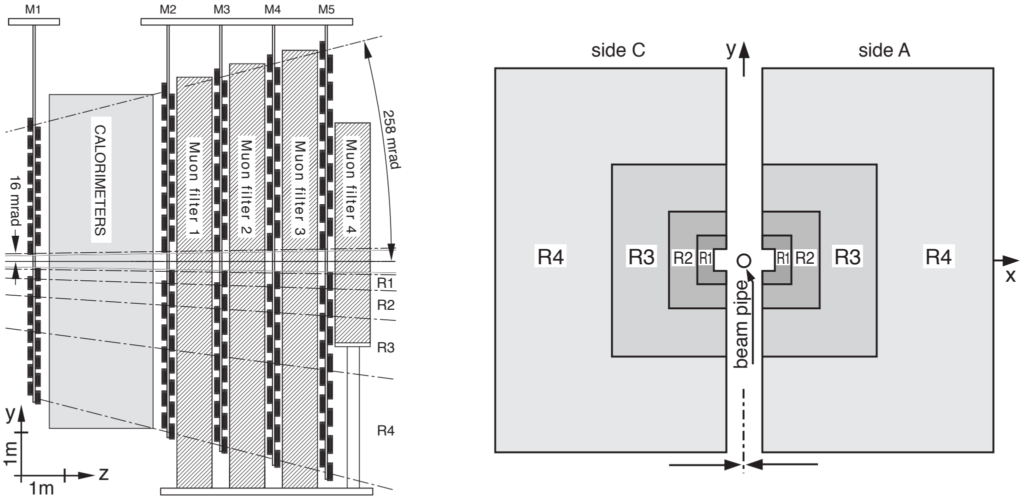


Figure 3.14: Side view of the Muon Detector (left). Station layout with the four regions R1–R4 (right).

For triggering, muon reconstruction is performed by the stand alone system that achieves an average transverse momentum resolution of $\sim 20\%$.

3.3 The LHCb trigger in Run 2

The LHCb trigger was designed to select heavy-flavor decays from the huge light-quark background, sustaining the LHC bunch-crossing rate of 40 MHz and selecting up to 12.5 kHz of data to store [51]. Only a small fraction of events, about 15 kHz, contains a b-hadron decay with all final state particles emitted in the detector acceptance. The rate of "interesting" bottom hadron decays is even smaller, of a few Hz. Corresponding values for charmed hadrons are about 20 times larger. It is therefore crucial, for the trigger, to reject background as early as possible in the data flow.

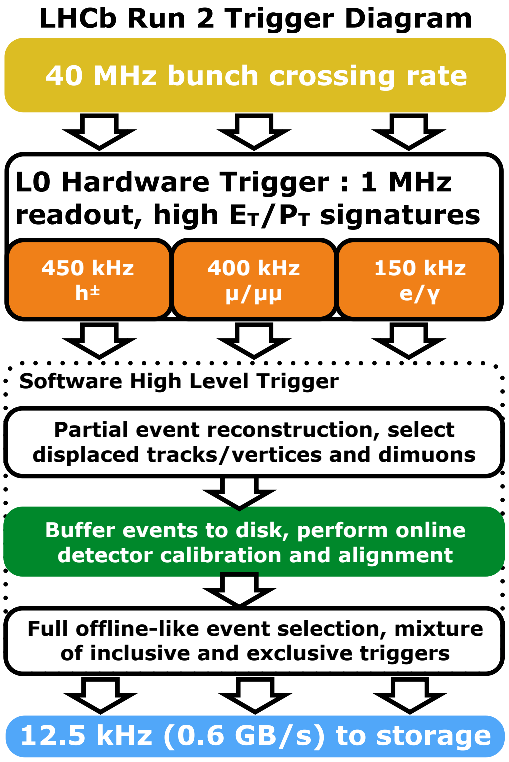

LHCb Run 2 Trigger Diagram
40 MHz bunch crossing rate
L0 Hardware Trigger : 1 MHz readout, high E_T/P_T signatures
450 kHz $h^{\pm}$
400 kHz $\mu/\mu\mu$
150 kHz e/γ
Software High Level Trigger
Partial event reconstruction, select displaced tracks/vertices and dimuons
Buffer events to disk, perform online detector calibration and alignment
Full offline-like event selection, mixture of inclusive and exclusive triggers
12.5 kHz (0.6 GB/s) to storage

Figure 3.15: Representation of LHCb trigger flow and typycal event-accept rates for each stage.

The LHCb trigger is organised into two sequential stages, the L0 trigger and the High Level Trigger (HLT). This two-level structure helps coping with timing and selection requirements, with a fast and partial reconstruction at low level, followed by a more accurate and complex reconstruction at high level. The hardware-based L0 trigger operates synchronously with the bunch crossing. It uses information from calorimeter and muon detectors to reduce the 40 MHz bunch-crossing rate to below 1.1 MHz, which is the maximum value at which the whole detector can be read out by design. Then, the asynchronous software-based HLT performs a finer selection based on information from all detectors, and reduces rate to 12.5 kHz. Figure 3.15

shows the LHCb trigger flow for Run 2, and typical event-accept rates for each stage.

The L0 trigger

The task of L0 trigger is to reduce the event rate from 40 MHz (the bunch-crossing rate) to 1 MHz, that is the maximum rate at which the full detector can be read. Data from all detectors are stored in memory buffers consisting of an analog pipeline that is read out with a fixed latency of 4 μs. The L0 decision must be available within this fixed time, therefore the L0 trigger is entirely based on custom-built electronic boards, relying on parallelism and pipelining. At this stage, trigger requests can only involve simple and immediately available quantities, like those provided by calorimeter and muon detectors. The L0 trigger consists of three independent trigger decisions, the *L0 hadron*, the *L0 muon*, the *L0 calorimeter*. Each decision is combined with the others through a logic “or” in the L0 decision unit.

The L0 hadron trigger aims at collecting samples enriched in hadronic c- and b-particle decays. Final-state particles from such decays have on average higher transverse momenta than particles originated from light-quark processes, and this property helps in discriminating between signal and background.

The L0 muon trigger uses the information from the five muons stations, to identify the most energetic muons. Once the two muons candidates with highest transverse momentum per quadrant of the muons detectors are identified, the trigger decision depends on two thresholds: one on the highest transverse momentum (L0 muon) and one on the product of the two highest transverse momenta (L0 dimuon).

The L0 calorimeter trigger uses the information from ECAL, HCAL, PS, and SPD. It calculates the transverse energy E_{T} deposited in a cluster of 2×2 cells of the same size, for both the electromagnetic and the hadron calorimeters. The transverse energy is combined with information on the number of hits on preshower and scintillator pad detectors to define three types of trigger candidates, photon, electron, and hadron.

The High Level Trigger

Events accepted at L0 are transferred to the Event Filter Farm (EFF), an array of computers consisting of more than 15,000 commercial processors, for the HLT stage. The HLT is implemented through a C++ executable that runs on each processor of the farm, reconstructing and selecting events in a way similar to the offline processing. A substantial difference between online and offline algorithms is the time available to completely reconstruct a single event. The offline reconstruction requires almost 2 s per event in average, while the maximum time available for the online reconstruction is typically 50 ms.

The HLT consists of several trigger selections designed to collect specific events, in particular, c- or b-hadron decays. Every trigger selection is specified by reconstruction algorithms and selection criteria that exploit the kinematic features of charged and neutral particles, the decay topology, and the particle identities. The HLT processing time is shared between two different levels, a first stage called High Level

Trigger 1 (HLT1) and a second stage High Level Trigger 2 (HLT2). A partial event reconstruction is done in the first stage in order to reduce the event accept rate to 30 kHz, and a more complete event reconstruction follows in the second stage.

At the first level, tracks are reconstructed in the VELO and selected based on their probability to come from heavy-flavor decays, by determining their impact parameter with respect to the closest primary vertex. At the second level, a complete forward tracking of all tracks reconstructed in the VELO is performed, and also Downstream and T tracks reconstruction is performed. Several trigger selections, either inclusive or exclusive, are available at this stage.

A key computing challenge is to store and process this data, which limits the maximum output rate of the LHCb trigger. Writing full raw sub-detector data, which are passed through a full offline event reconstruction before being considered for physics analysis, LHCb would be able to register few kHz of events. Charm physics in particular is limited by trigger output rate constraints. A new streaming strategy includes the possibility to perform the physics analysis with candidates reconstructed in the trigger, thus bypassing the offline reconstruction. In the *Turbo stream* the trigger write out a compact summary of physics objects containing all information necessary for analyses [52], discarding the rest of the event to save bandwidth and storage resources. This allows an increased output rate and thus reach higher trigger efficiencies.

Chapter 4

CPV measurement in $D^0 \to K^0_S K^\mp \pi^\pm$ decays

4.1 Scope and strategy

From the Run 2 data sample (corresponding to 5.6 fb^{-1}) I have extracted (selection described further below) a number of $D^0 \to K^0_S K^- \pi^+$ and $D^0 \to K^0_S K^+ \pi^-$ candidates[1] of $845 \cdot 10^3$ and $617 \cdot 10^3$ respectively – a sample size ~ 8 times larger than in Run 1. To separate the $D^0 \to K^0_S K^- \pi^+$ ($D^0 \to K^0_S K^+ \pi^-$) decay from the charge conjugate of the other channel $\overline{D}{}^0 \to K^0_S K^- \pi^+$ ($\overline{D}{}^0 \to K^0_S K^+ \pi^-$), I selected $D^{*+} \to D^0 \pi^+_{soft}$ prompt $D^{*-} \to \overline{D}{}^0 \pi^-_{soft}$ decays. The pion from the $D^{*\pm}$ decay has little kinetic energy, for this reason it is commonly referred to “soft” pion. Its sign gives the flavour of the accompanying D^0, and produces two separated samples: the Right-Sign (RS) set where the pion from the D^0 decay has the same charge of the “soft” pion, and the Wrong-Sign (WS) where the signs are opposite.

A simple way to achieve an asymmetry measurement of a specific resonance is to just count events in a certain region around the nominal resonance region, and then evaluating a *CP* asymmetry. However this is not very sensitive, particularly in the presence of strong interference as in the present case, with the asymmetry expected to vary point by point in the Dalitz plot. The same can be said of other global approaches, trying to generically detect charge asymmetries with no assumptions on their structure. On the other hand, a full Dalitz analysis can keep all effects into account and make best use of all available information, but it relies rather heavily on the correctness of the assumed model. In a full Dalitz analysis it is often necessary to consider several alternative models, and evaluating the associated systematic uncertain is difficult and prone to subjective judgements. It is in principle possible for small mismodelings, even in regions well separated from the resonance of interest, to influence the result in ways that is difficult to control. To avoid both extremes, I have chosen a new approach, that might be regarded as a middle ground between a full Dalitz analysis and a simple counting experiment. I want to take advantage

[1] The inclusion of the charge conjugate decays is implied.

of the peculiar shape of the Dalitz plot of the $D^0 \to K^0_S K^- \pi^+$ and $D^0 \to K^0_S K^+ \pi^-$ decays near the resonance of the $K^*(892)^0$, to measure the difference between the complex coefficient of the $K^*(892)^0$ resonance in the amplitude model of the two decays. To this purpose, I construct a custom-made observable of low dimensionality, tuned to the expected Dalitz distribution in order to be as sensitive as possible to the effects being searched. At the same time, its simplicity and low dimensionality make it very insensitive to the precise distribution assumed, allowing to detect in an unbiased way any possible *CP*-violating effect following roughly the expected pattern. In short, I try to use the theoretical model to optimise the sensitivity of this search to a good extent, but not so far as to lose robustness to small theoretical mismodelings; and I make sure that the expected value of the observable is zero in case of *CP* symmetry, independently of the modeling assumptions. The model only enters in optimising the sensitivity, and (in a limited and essentially unavoidable way) in case of detecting a significant asymmetry, in translating the value of the custom observable to a measurement of physically meaningful parameters, that can be used to advance the understanding of *CP* violation in the charm sector.

4.2 Decay model

4.2.1 *CP*-conserving part

The expected Dalitz distributions used for tuning the custom-made observable are produced using the amplitude model PDF found in the analysis performed by the LHCb collaboration with the Run 1 data [41], and follows its formalism. The amplitude model PDF uses the isobar formalism:

$$a_{K^0_S K^\pm \pi^\mp}(m^2_{K^0_S K}, m^2_{K^0_S \pi}) \propto \varepsilon(m^2_{K^0_S K}, m^2_{K^0_S \pi}) |\mathcal{M}_{K^0_S K^\pm \pi^\mp}(m^2_{K^0_S K}, m^2_{K^0_S \pi})|^2 \tag{4.1}$$

where ε is the efficiency model, and

$$\mathcal{M}_{K^0_S K^\pm \pi^\mp}(m^2_{K^0_S K}, m^2_{K^0_S \pi}) = \sum_R a_R e^{i\phi_R} \mathcal{M}_R(m^2_{K^0_S K}, m^2_{K^0_S \pi}) \tag{4.2}$$

with the sum over 2-body intermediate resonances R, where $D^0 \to (R \to (AB)_L)C$. The matrix elements $\mathcal{M}_R$ are given by:

$$\mathcal{M}_R = B^D_L(p, p_0, d_D)\Omega_L T_R(m_{AB}) B^R_L(q, q_0, d_R) \tag{4.3}$$

where p and q are the momentum of C and A (or B) in the R rest frame respectively, p_0 and q_0 are the same quantities calculated using the nominal resonance mass, d_D and d_R are the meson radius parameters, which are set to $5.0(\text{GeV}/c)^{-1}$ and $1.5(\text{GeV}/c)^{-1}$ respectively. L indicates the spin of the resonance. B^D_L and B^R_L are the barrier penetration factors for the production of RC and AB, respectively. Ω_L accounts for the angular distribution of the final state particles. T_R is the dynamical function describing the resonance, also called *lineshape*.

For the efficiency model $\varepsilon(m^2_{K^0_S K}, m^2_{K^0_S \pi})$, I adopted the same 6th order polynomial in $m^2(K^0_S K)$ and $m^2(K^0_S \pi)$ obtained in the Run 1 analysis fitting simulated events. Figure 4.1 shows the efficiency model in the same space around $m^2(K\pi)$ vs $m^2(K^0_S\pi)$ used for the others Dalitz plots. Again, this analysis methodology is insensitive to the details of this function; the corresponding systematic uncertainty will be assessed in section 4.6, where I also discuss the important topic of the possible *asymmetry* of the efficiency function.

Details on the individual lineshapes used in the amplitude model are reported in appendix A.

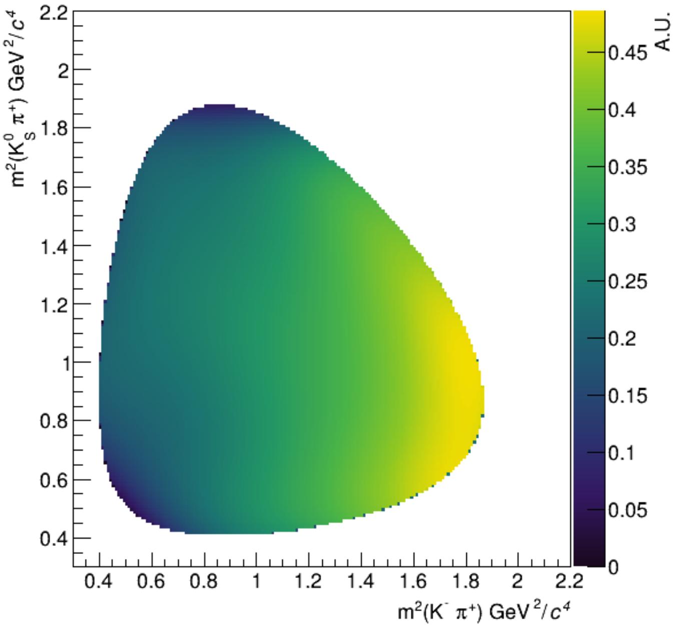


Figure 4.1: Efficiency model.

4.2.2 *CP*-violating amplitudes

To account for the possibility of a decay through a specific resonance to be *CP*-violating, the complex amplitudes of R and $\overline{R}$ must be modified. As a parametrisation, I chose to replace the amplitude and the phase parameter a_R and ϕ_R of equation 4.2 with $a_R(1 \pm \Delta a_R)$ and $\phi_R \pm \Delta\phi_R$ respectively, where the signs are set by the flavour tag. I use the convention that a positive sign produces the D^0 complex amplitudes, and the negative sign the $\overline{D}^0$ complex amplitudes.

This modification makes the Dalitz plots of the decay and its complex conjugate somewhat different. I used `GooFit` to produce Dalitz plots simulating *CP*-violation on desired resonances with specific value of Δa_R and $\Delta\phi_R$. Figure 4.2 shows an example where the Dalitz plots of the $\overline{D}^0 \to K^0_S K^+\pi^-$ and $D^0 \to K^0_S K^-\pi^+$ decays with $\Delta a_R = 0.04$ and $R = K^*(892)^0$. Their bin-by-bin difference is also shown, to highlight the differences between them. Similar plots for the WS decay channel and $\Delta\phi_R \neq 0$ are shown in Figure 4.3. All plots show a characteristic pattern around the value $m^2(K^-\pi^+) = 0.8\,\mathrm{GeV}^2/c^4$, corresponding to the $K^*(892)^0$ resonance region.

The characteristic patterns seen in the difference plots of Figs. 4.2 and 4.3 are the signatures of CP-violation in the $K^*(892)^0$ resonance, that I need to look for to detect CP asymmetry in this decay mode. With the exception of minor details, the shapes of these patterns are quite insensitive to the details of the amplitude model. In the next section I describe the method I used to build simple observables with the property of maximal sensitivity to decay asymmetries shaped according those Dalitz patterns.

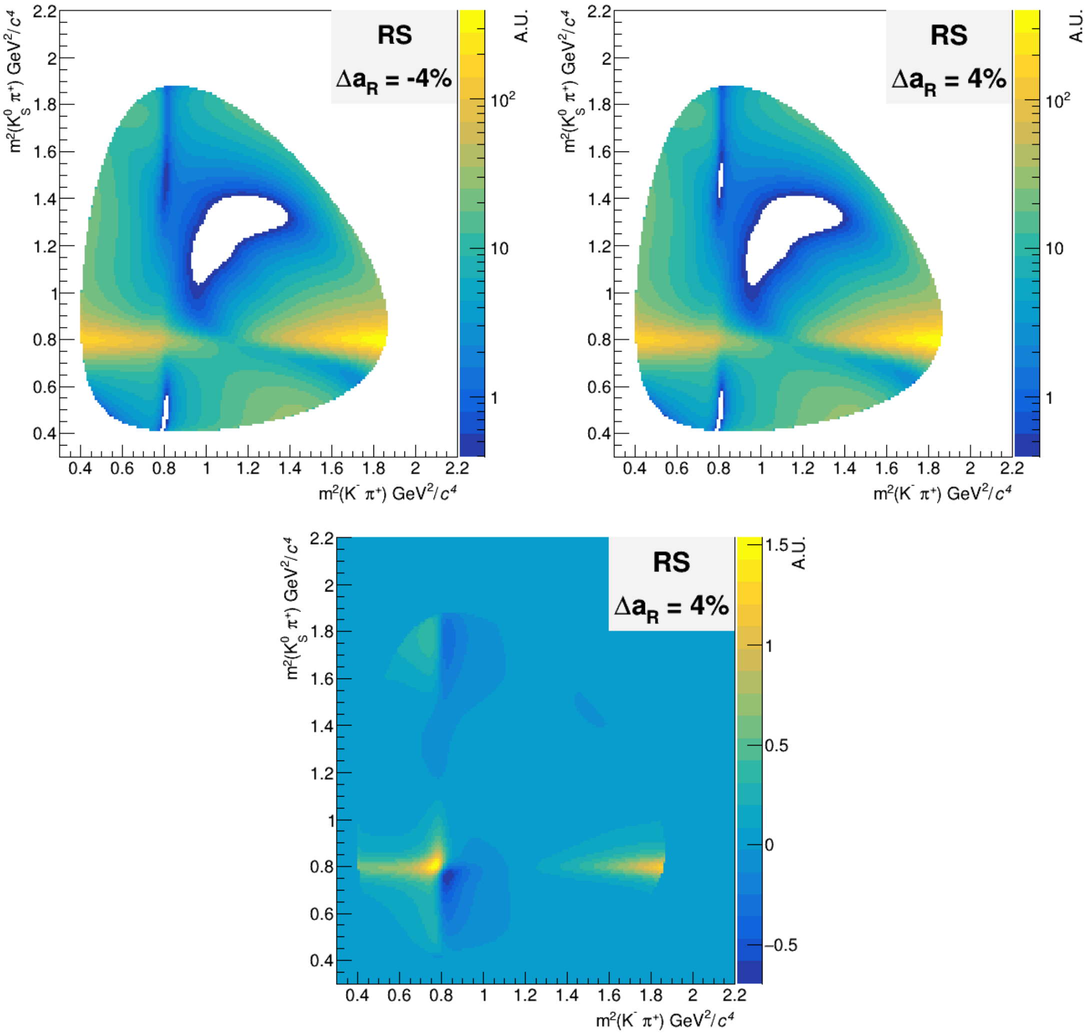


Figure 4.2: Simulated Dalitz plot of the $\overline{D}^0 \to K^0_S K^+\pi^-$ (top left) and $D^0 \to K^0_S K^-\pi^+$ (top right) decays with $|\Delta a_R| = 0.04$ and $R = K^*(892)^0$ and the differences between them obtained subtracting the first from the second.

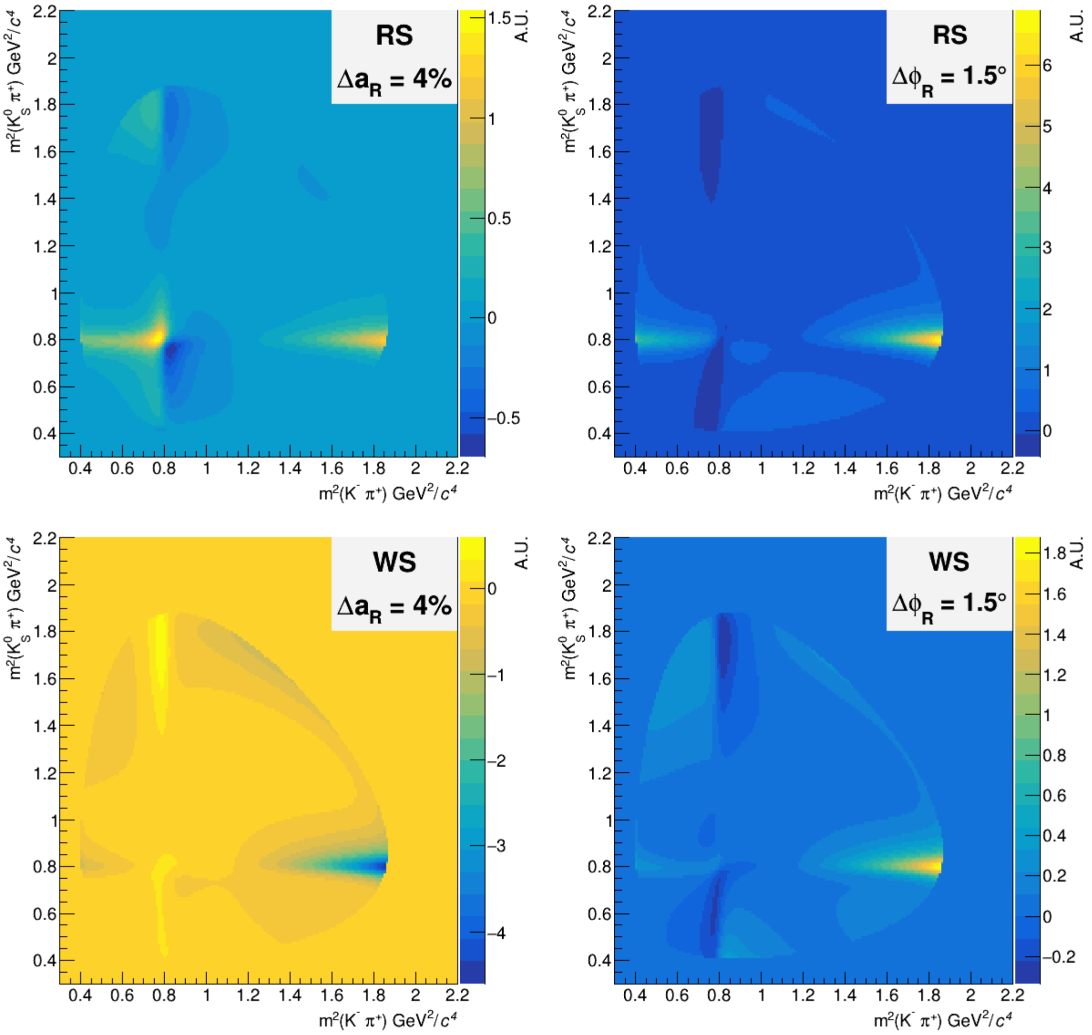


Figure 4.3: Differences between simulated Dalitz plot for RS and WS samples with $\Delta a_R = 0.04$ or $\Delta\phi_R = 1.5°$ ($R = K^*(892)^0$).

4.3 Optimised *CPV* detection

Given two data samples S^+ and S^-, distributed as $f^+(x)$ and $f^-(x)$ respectively, I define the average distribution as:

$$f(x) = \frac{f^+(x) + f^-(x)}{2} \tag{4.4}$$

and the difference between the two expected distributions according to a model m:

$$g(x) = f^+_m(x) - f^-_m(x). \tag{4.5}$$

The observable that I'm going to describe shows maximal sensitivity when $f^+(x) - f^-(x)$ have the same shape of $g(x)$.

In the specific case of this analysis the data samples S^+ and S^- are the data sample of D^0 and $\overline{D}^0$ decays of one of the two mode (RS and WS), $f^+(x)$ and $f^-(x)$ the Dalitz distribution of the data samples, $f^+_m(x)$ and $f^-_m(x)$ the Dalitz distribution generated by the amplitude model injecting *CPV* in $K^*(892)^0$ resonance.

The function $g(x)$ is identically zero in the absence of *CP*-violating effects. I also took the distributions $f^+(x)$ and $f^-(x)$ as separately normalised, so that an overall *CP* asymmetry that is uniform over the full Dalitz plot would still give $g(x) = 0$ everywhere. This means that an analysis aimed purely at detecting $g(x) \neq 0$ would be insensitive to an overall *CP* asymmetry. However, it would be totally unexpected to have exactly the same *CP*-violating amplitudes and phases in all subchannels contributing to the Dalitz distribution, and it fact it is not predicted by the latest theoretical calculations [35,36]. Conversely, the overall asymmetry is more difficult to measure accurately, due to the existence of sizeable efficiency differences due to charge asymmetries in the detection, contributing non-trivial systematic uncertainties to its determination. I have chosen to focus most of my measurement on the detection of relative asymmetries in the plot, leaving the issue of an overall global asymmetry check to a possible additional step to be performed separately.

Owing to the smallness of expected asymmetries, I assumed the distributions can be safely expanded to first order in the *CP*-violating parameter of interest:

$$g(x) = \theta g_0(x) \tag{4.6}$$

$$f^\pm(x|\theta) = f(x) \pm \theta g_0(x) \tag{4.7}$$

where θ is a real coefficient, and $\theta = 0$ implies *CP* symmetry (I will come back to the linearity assumption in a later section).

Considering now a statistical test of the hypothesis $H_0 : \theta = \theta_0$ versus the alternative H_θ given by a value of the parameter $\theta > \theta_0$. It is a known result that, if the likelihood is sufficiently regular, a most powerful test exists for small deviations of θ from θ_0 ("Locally Most Powerful test", or LMP), and it is defined by the critical region

$$s(x) > q_\alpha$$

where $s(x)$ is the Fisher score function:

$$s(x) = \left.\frac{\partial \log \mathcal{L}}{\partial \theta}\right|_{\theta=\theta_0}, \tag{4.8}$$

q_α is chosen so that $\mathbb{P}(s(x) > q_\alpha|\theta_0) = \alpha$, and α is a chosen significance level (i.e. the probability of rejecting the null hypothesis when it is true). In our case, the score functions are

$$s^\pm(x) = \left.\frac{\partial \log \mathcal{L}^\pm}{\partial \theta}\right|_{\theta=0} \tag{4.9}$$

$$= \left.\frac{\partial \log \left[\prod_{x \in S^\pm} (f(x) \pm \theta g_0(x))\right]}{\partial \theta}\right|_{\theta=0} \tag{4.10}$$

$$= \sum_{x \in S^\pm} \left.\frac{\partial \log \left[f(x) \pm \theta g_0(x)\right]}{\partial \theta}\right|_{\theta=0} \tag{4.11}$$

$$= \pm \sum_{x \in S^\pm} \left.\frac{g_0(x)}{f(x) \pm \theta g_0(x)}\right|_{\theta=0} \tag{4.12}$$

$$= \pm \sum_{x \in S^\pm} \frac{g_0(x)}{f(x)}. \tag{4.13}$$

This procedure is however intended for a single-sided test ($\theta > \theta_0$), while in our case we do not want to assume knowledge of the sign of the *CP* asymmetry. For this reason, we choose to combine the two Fisher score functions in a single statistic t that is suitable for a two-sided test, in this way:

$$t = \frac{1}{N_{S^+}} s^+(x) + \frac{1}{N_{S^-}} s^-(x) = \frac{1}{N_{S^+}} \sum_{x_i \in S^+} \frac{g_0(x_i)}{f(x_i)} - \frac{1}{N_{S^-}} \sum_{x_i \in S^-} \frac{g_0(x_i)}{f(x_i)} \tag{4.14}$$

where N_{S^+} and N_{S^-} are the sizes of the two observed data samples. This choice of coefficients provides an appropriate normalisation and makes our chose statistic independent from a possible global asymmetry in the total expected numbers of events N_{S^+}, N_{S^-}. If $\theta = 0$, t will be distributed around zero; otherwise its mean value will be a linear function of θ. This makes t a convenient statistic to use, not only for the purpose of testing H_0 but also as a way to measure the *CP* asymmetry parameters quantitatively. It should be noted that this is obtained by a straightforward calculation, with no need for any fit/numerical minimisation procedure.

In principle, this optimal sensitivity is achieved when performing the sum in Equation. 4.14 over the whole Dalitz space. However, to limit the uncertainty associated to the modelling of the complete Dalitz space and spurious contributions from other resonance asymmetries, it is more effective to restrict the integration region to the most relevant part, that is where the $g_0(x)$ function is significantly different from zero. This reduces only slightly the statistical power of the analysis compared to the absolute maximum, while ensuring a good robustness of the result, controlling the systematic uncertainty.

For the $D^0 \to K^0_S K^- \pi^+$ and $D^0 \to K^0_S K^+ \pi^-$ modes, I choose the regions defined by $m^2(K^-\pi^+) \in [0.7, 0.9]\,\text{GeV}^2/c^4$ and $m^2(K^0_S\pi^+) < 0.8\,\text{GeV}^2/c^4 \vee m^2(K^0_S\pi^+) > 1.3\,\text{GeV}^2/c^4$, corresponding roughly to the location of the $K^*(892)^0$ peak (Figure 4.4).

The function $g_0(x)$ is obtained from the four diff-plots shown in Figure 4.3, thus four different functions are defined: two for $D^0 \to K^0_S K^- \pi^+$ decays modelling

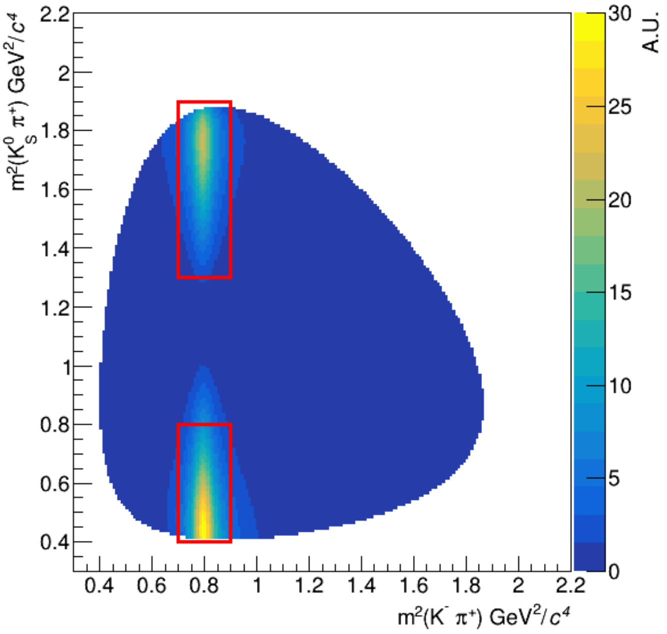


Figure 4.4: Dalitz plot of the $D^0 \to K^0_S K^*(892)^0$ decay. The red box indicates the regions used to evaluate t.

$\Delta a_{K^*(892)^0} = 0.04$ or $\Delta\phi_{K^*(892)^0} = 1.5°$, and two for $D^0 \to K^0_S K^- \pi^+$ decays modelling the same value of $\Delta a_{K^*(892)^0}$ and $\Delta\phi_{K^*(892)^0}$. I define separate observables t: t_a^{RS} t_ϕ^{RS} t_a^{WS} t_ϕ^{WS}, one for each $g_0(x)$ function.

I produced several models for different values of Δa_R and $\Delta\phi_R$; tests of the linearity of observable t response, and extraction of the corresponding physics parameters θ are discussed in Section 4.3.1.

I evaluated the sensitivity of observable t as the probability of rejecting the no-*CPV* hypothesis as a function of the amount of *CP* violation injected into the amplitude model. I produced the distribution of the Dalitz plot assuming no-*CPV*, $\Delta a_{K^*(892)^0} = 0.04$, and $\Delta\phi_{K^*(892)^0} = 1.5°$. I randomly extracted from these distributions different samples S^+ and S^- (one pair for each model assumption). The size of the samples is extracted from a Poisson distribution with mean equal to the size of the real data samples ($4.344 \cdot 10^5$ for the RS and $3.139 \cdot 10^5$ for the WS). Then I compute the values of t. To observe the statistical fluctuations, I repeated the test 5'000 times for each condition.

Figure 4.5 shows in red the t-distribution, arbitrarily taking the central values measured in Run 1 as reference points: $\Delta a_{K^*(892)^0} = 0.04$ or $\Delta\phi_{K^*(892)^0} = 1.5°$ for the RS and the WS channels. The blue distributions are relative to the no-*CPV* hypothesis. The magenta line correspond to the cutoff choosing a significance level α of 4.55% (equivalent to a deviation of 2σ). The power of the test (probability of rejecting the no-*CPV* hypothesis when it is false), is given by the integral of the red distribution above the cutoff. With these value of $\Delta a_{K^*(892)^0}$ and $\Delta\phi_{K^*(892)^0}$ and yield the power is better than 97.5%.

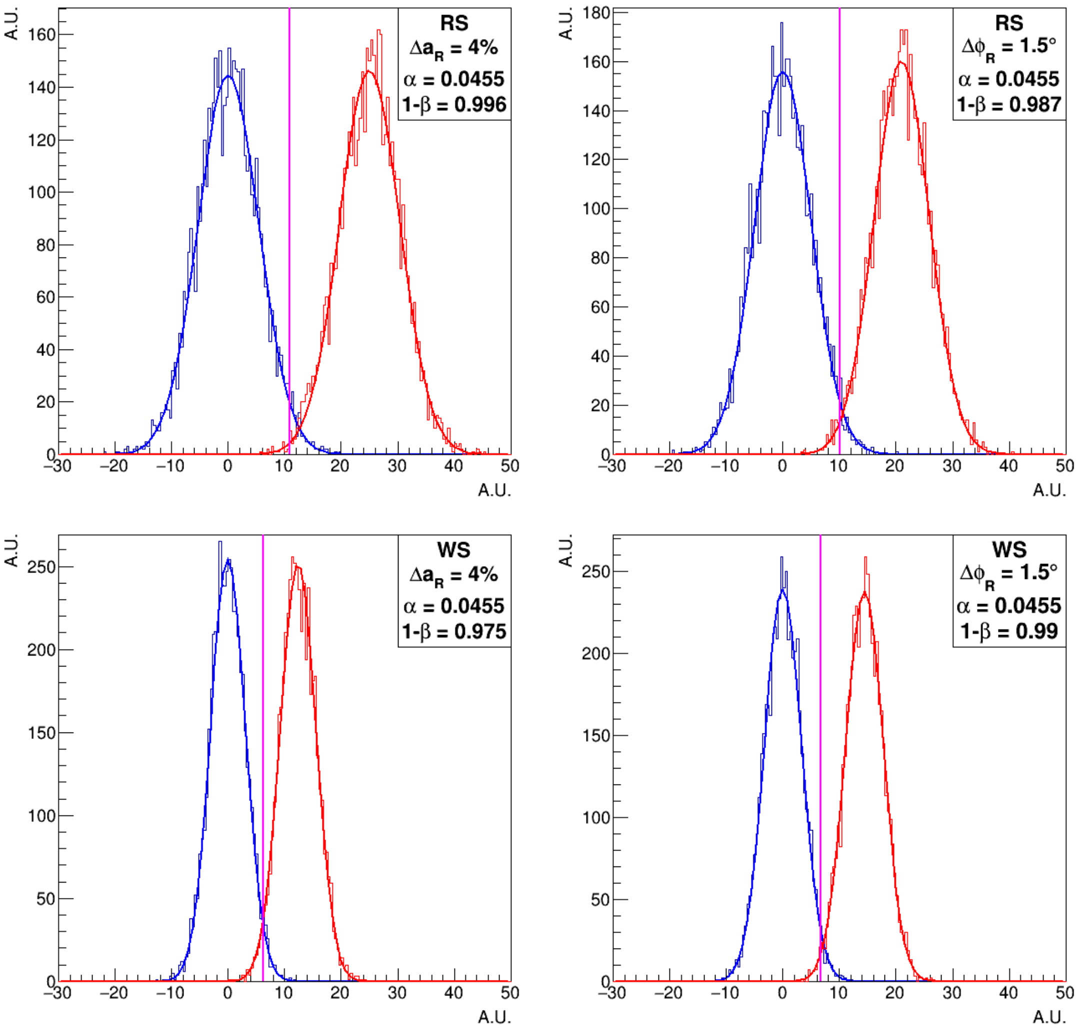


Figure 4.5: Separation between the t distributions assuming no-*CPV* (blue distribution) and assuming $\Delta a_R = 0.04$ or $\Delta\phi_R = 1.5°$ with $R = K^*(892)^0$ (red distribution). The magenta line represents the null hypothesis rejection cutoff.

4.3.1 Linearity of response

In principle t must be evaluated for many different values of the *CP*-violating parameters, changing the function $g_0(x)$ each time. However, if small changes in those parameter do not produce significant changes in the shape of $g(x)$ distribution, the measured value of the t can be easily convert to a value of Δa_R or $\Delta\phi_R$ with no need for testing different templates. This amounts to a strong simplification of the analysis – effectively turning a template-fitting procedure with numerical minimisation into a single calculation of a simple integral.

To perform the linearity check, I adopted as central values for the definition of $g_0(x)$ the same values of the previous test:

- RS, $\Delta a_{K^*(892)^0} = 0.04$,
- RS, $\Delta \phi_{K^*(892)^0} = 1.5°$,
- WS, $\Delta a_{K^*(892)^0} = 0.04$,
- WS, $\Delta \phi_{K^*(892)^0} = 1.5°$.

Then I randomly extracted samples from different Dalitz plots produced from the model, assuming $\Delta a_{K^*(892)^0}$ from 0 to 0.1 with step 0.01 or $\Delta \phi_{K^*(892)^0}$ from 0° to 2.5° with step 0.25°. For each case, I randomise 5'000 sets of S^+ and S^- samples with a size comparable to the size of the data samples and compute t.

Figure 4.6 shows the distribution of the observables t for different values of $\Delta a_{K^*(892)^0}$ and $\Delta \phi_{K^*(892)^0}$. It is possible to clearly see the t distribution shifting towards higher values, proportionally to $\Delta a_{K^*(892)^0}$ and $\Delta \phi_{K^*(892)^0}$. Figure 4.7 shows that this shift is very closely linear (within a small fraction of a sigma), even on a range extending far beyond plausible expectations for these parameters.

Table 4.1 summarises the fit results of the plot in Figure 4.7 with a linear function:

$$\langle t_a \rangle = p_0 + p_1 \cdot \Delta a$$

$$\langle t_\phi \rangle = p_0 + p_1 \cdot \Delta \phi.$$

The p_0 coefficients obtained from the fit are not zero as the result of an artificial adjustment of small non-linearities. Inverting these functions allows to convert the measurement of t in the measurement of $\Delta a_{K^*(892)^0}$ and $\Delta \phi_{K^*(892)^0}$. Dividing the standard deviation of the distribution of the observables t in no *CP*-violation hypothesis (σ_{H_0}) by the corresponding function slope provides the statistical uncertainty in the measurement of $\Delta a_{K^*(892)^0}$ and $\Delta \phi_{K^*(892)^0}$. Table 4.1 reports these values, together with the statistical uncertainty obtained in the Run 1 analysis for comparison. Based on just the yield increase, I would expect a resolution improvement by about a factor 3 if the current analysis had no loss of power with respect to the full Dalitz fit. From the table is possible to see that this is indeed the case, and I actually have more power than our naive extrapolation would predict. This could be attributed to the fact that here I am neglecting the (modest) effect of the background, and I am partly exploiting some information on the shape of the distribution as determined from the Run 1 analysis.

4.3.2 Sensitivity to a global asymmetry

Production asymmetry, different geometrical acceptance between π^+_{soft} and π^-_{soft}, detection asymmetry, and other physics or detector effects could lead to a different size of D^0 and $\overline{D}^0$ samples. The Equation 4.14 equalise the size of the data samples, however a global asymmetry could in principle lead to changes of the distribution of the observable t and its statistical power $1 - \beta$.

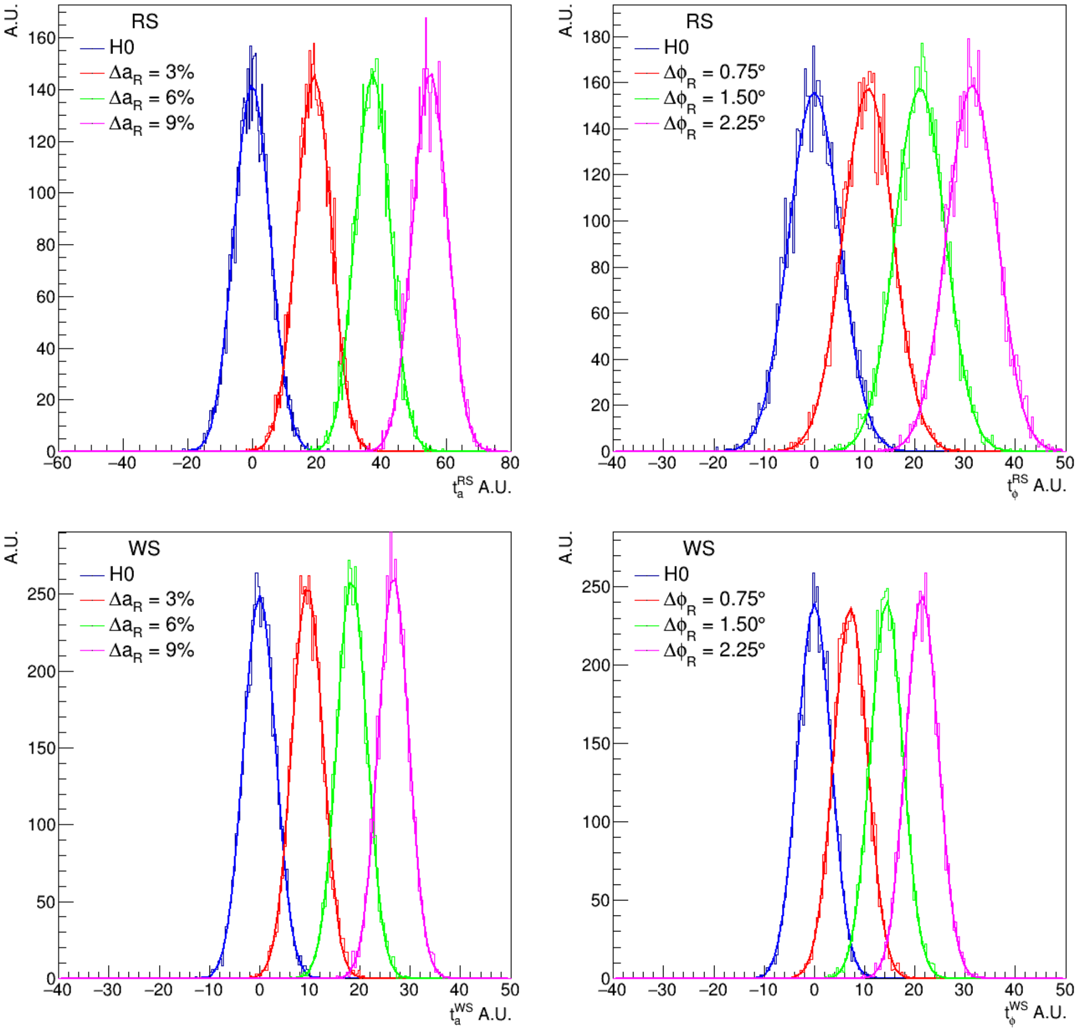


Figure 4.6: Separation between the t distributions assuming different values of $\Delta a_{K^*(892)^0}$ (left) and $\Delta\phi_{K^*(892)^0}$ (right), for RS (up) and WS (down) decay modes.

		p_0	p_1	σ_{H_0}	σ_Δ	σ_Δ Run 1
RS	a	0.489 ± 0.056	607.74 ± 0.95	5.49	0.0090	0.031
	ϕ	0.104 ± 0.088	13.926 ± 0.060	4.93	0.35°	1.6°
WS	a	0.508 ± 0.030	291.90 ± 0.50	3.12	0.011	0.024
	ϕ	0.0292 ± 0.0060	9.527 ± 0.041	3.28	0.34°	1.8°

Table 4.1: Summary of the fit result of the t versus Δ functions (Δa or $\Delta\phi$ according to the row) and comparison between the statistical uncertainty on the measure of $\Delta a_{K^*(892)^0}$ and $\Delta\phi_{K^*(892)^0}$, and the statistical uncertainty in Run 1 analysis.

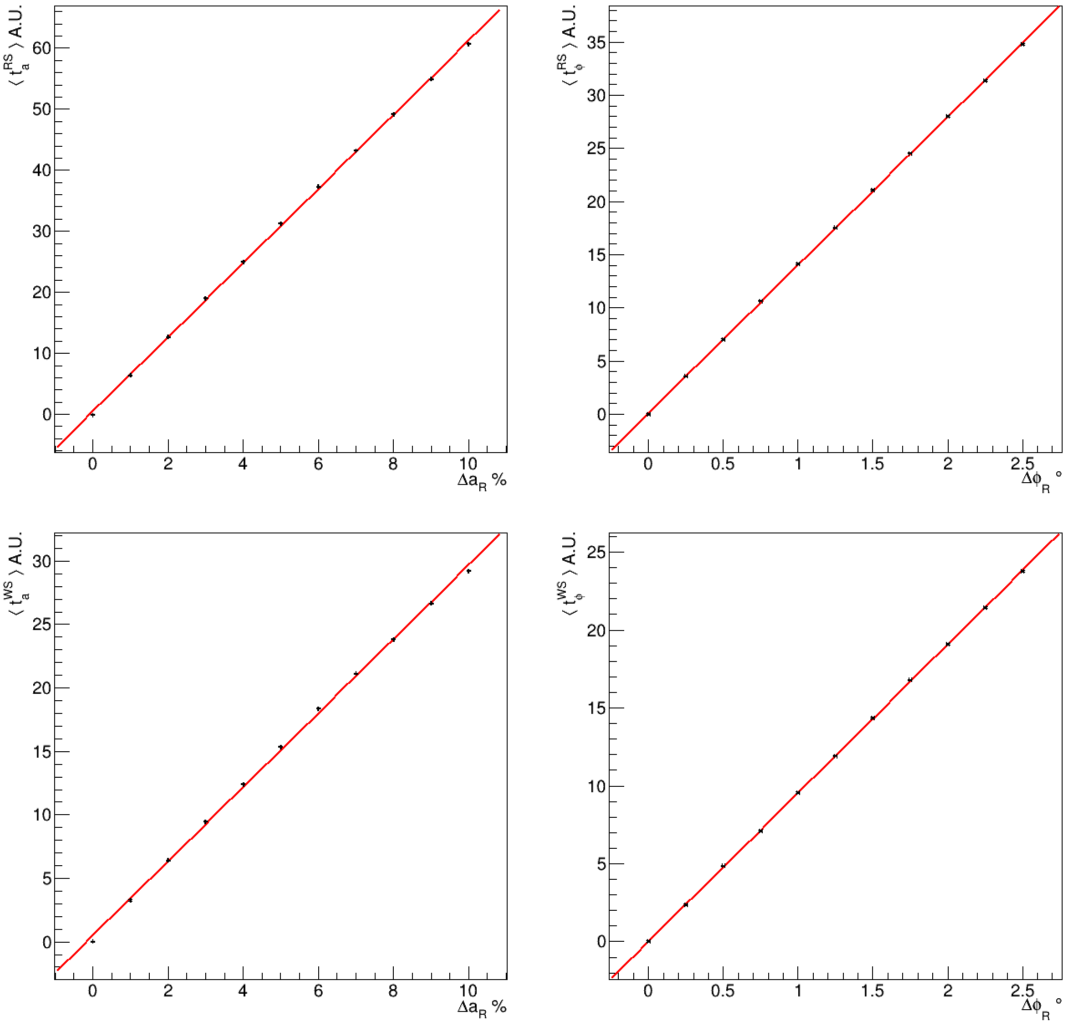


Figure 4.7: Mean of the t distributions versus the value of $\Delta a_{K^*(892)^0}$ (left) and $\Delta\phi_{K^*(892)^0}$ (right), for RS (up) and WS (down) decay modes.

To verify if the observable t is sensitive to a flat global asymmetry, I produced a set of data samples extracted with Poisson distribution around the means N_{D^0} and $N_{\bar{D}^0}$:

$$\begin{cases} \langle N_{D^0}\rangle + \langle N_{\bar{D}^0}\rangle = 2\langle N_0\rangle \\ \\ \dfrac{\langle N_{D^0}\rangle - \langle N_{\bar{D}^0}\rangle}{\langle N_{D^0}\rangle + \langle N_{\bar{D}^0}\rangle} = 0.2 \end{cases} \tag{4.15}$$

where N_0 is the size used in previous tests with symmetric samples.

Figure 4.8 shows the distribution of the observable t, assuming either no global asymmetry (left) or a huge global asymmetry of 20% (right). In each plot I show the distribution assuming no *CPV* (blue distribution) and modelling non-zero $\Delta a_{K^*(892)^0}$ or $\Delta\phi_{K^*(892)^0}$ (red distribution). The statistical power is not significantly affected.

I estimate the mean and the standard deviation of the t distribution by performing a fit with a Gaussian. The fit results and uncertainty are reported in the table 4.2 and 4.3. The differences between these values remain under the fit uncertainty.

	RS		WS	
	$\Delta a_R = 0.04$	$\Delta \phi_R = 1.5°$	$\Delta a_R = 0.04$	$\Delta \phi_R = 1.5°$
$\langle N_{D^0} \rangle = \langle N_{\overline{D}^0} \rangle$	25.024 ± 0.077	20.998 ± 0.070	12.328 ± 0.045	14.378 ± 0.048
$\frac{\langle N_{D^0} \rangle - \langle N_{\overline{D}^0} \rangle}{\langle N_{D^0} \rangle + \langle N_{\overline{D}^0} \rangle} = 0.2$	24.930 ± 0.081	20.892 ± 0.073	12.377 ± 0.045	14.340 ± 0.048

Table 4.2: Comparison between the mean of the t distribution with and without flat global asymmetry. $R = K^*(892)^0$.

	RS		WS	
	$\Delta a_R = 0.04$	$\Delta \phi_R = 1.5°$	$\Delta a_R = 0.04$	$\Delta \phi_R = 1.5°$
$\langle N_{D^0} \rangle = \langle N_{\overline{D}^0} \rangle$	5.420 ± 0.059	5.037 ± 0.053	3.137 ± 0.032	3.311 ± 0.034
$\frac{\langle N_{D^0} \rangle - \langle N_{\overline{D}^0} \rangle}{\langle N_{D^0} \rangle + \langle N_{\overline{D}^0} \rangle} = 0.2$	5.562 ± 0.052	5.018 ± 0.051	3.159 ± 0.032	3.342 ± 0.034

Table 4.3: Comparison between the standard deviation of the t distribution with and without flat global asymmetry. $R = K^*(892)^0$.

4.3.3 Sensitivity to *CP* asymmetry in other resonances

While these statistics are designed to be specifically sensitive to *CP* asymmetries around the $K^*(892)^0$ resonance, a *CP* asymmetry in other resonances may still affect the observable t, biasing it away from 0. I have performed tests of this effect with dedicated simulations, by plotting the distribution of the observables t (whose definition is kept fixed), when *CP* asymmetries are introduced in modes other than the one we are targeting.

Figures 4.9, 4.10, 4.11, 4.12, 4.13, and 4.14 show the distribution of t in these cases. The observables t do not show sensitivity to other resonances *CPV* of the same magnitude of the one for the $K^*(892)^0$ resonance, except for the RS sample of $K^*(1410)^0$. I evaluated to separately measure the latter and perform an explicit correction, while dealing with other possible resonance asymmetries as systematic uncertainties within the general context of the uncertainty on the amplitude model.

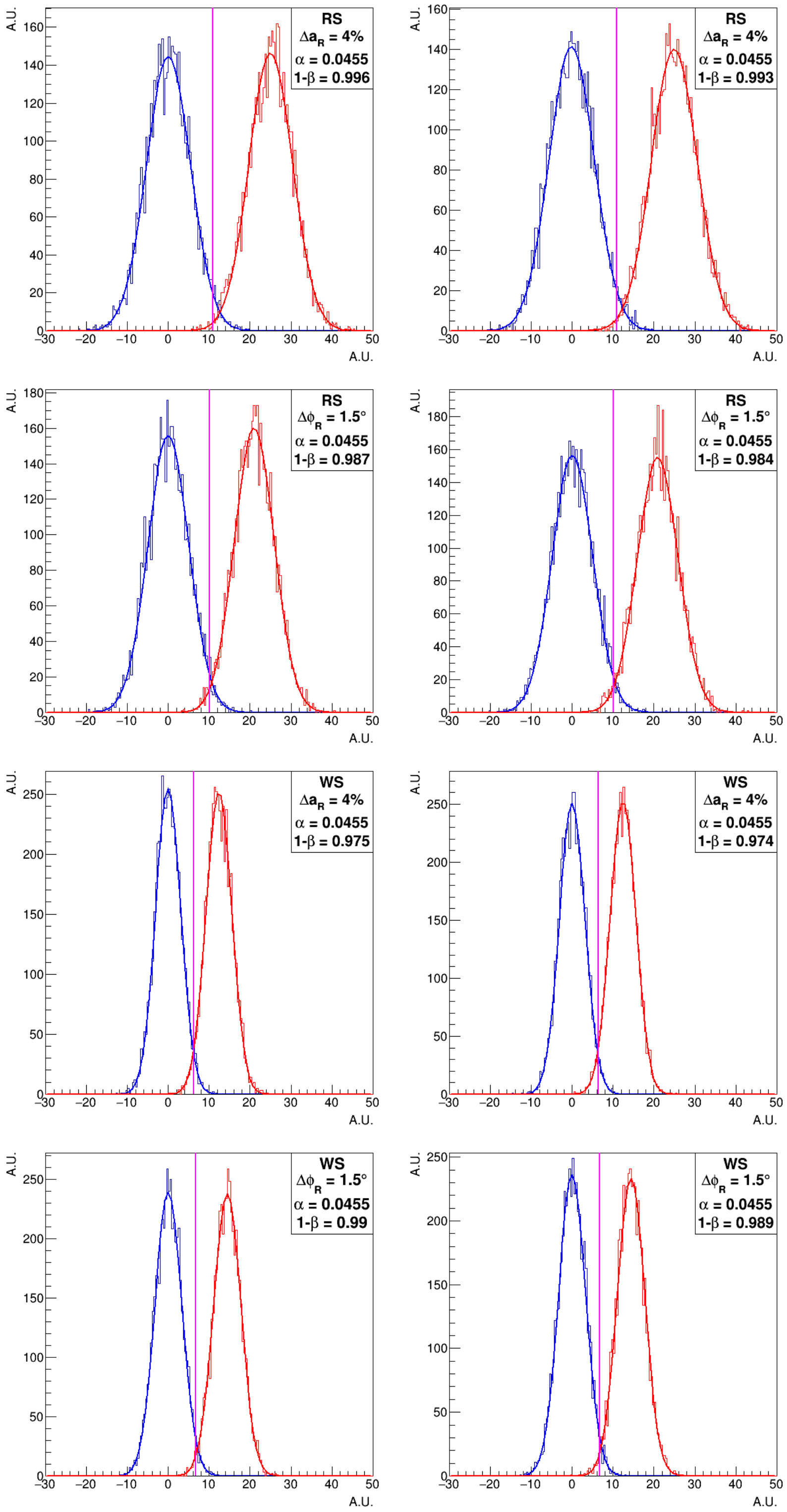


Figure 4.8: Comparison between the separation of t distributions assuming no global asymmetry between the D^0 and $\overline{D}^0$ samples (left) and a global asymmetry of 20% (right). $R = K^*(892)^0$.

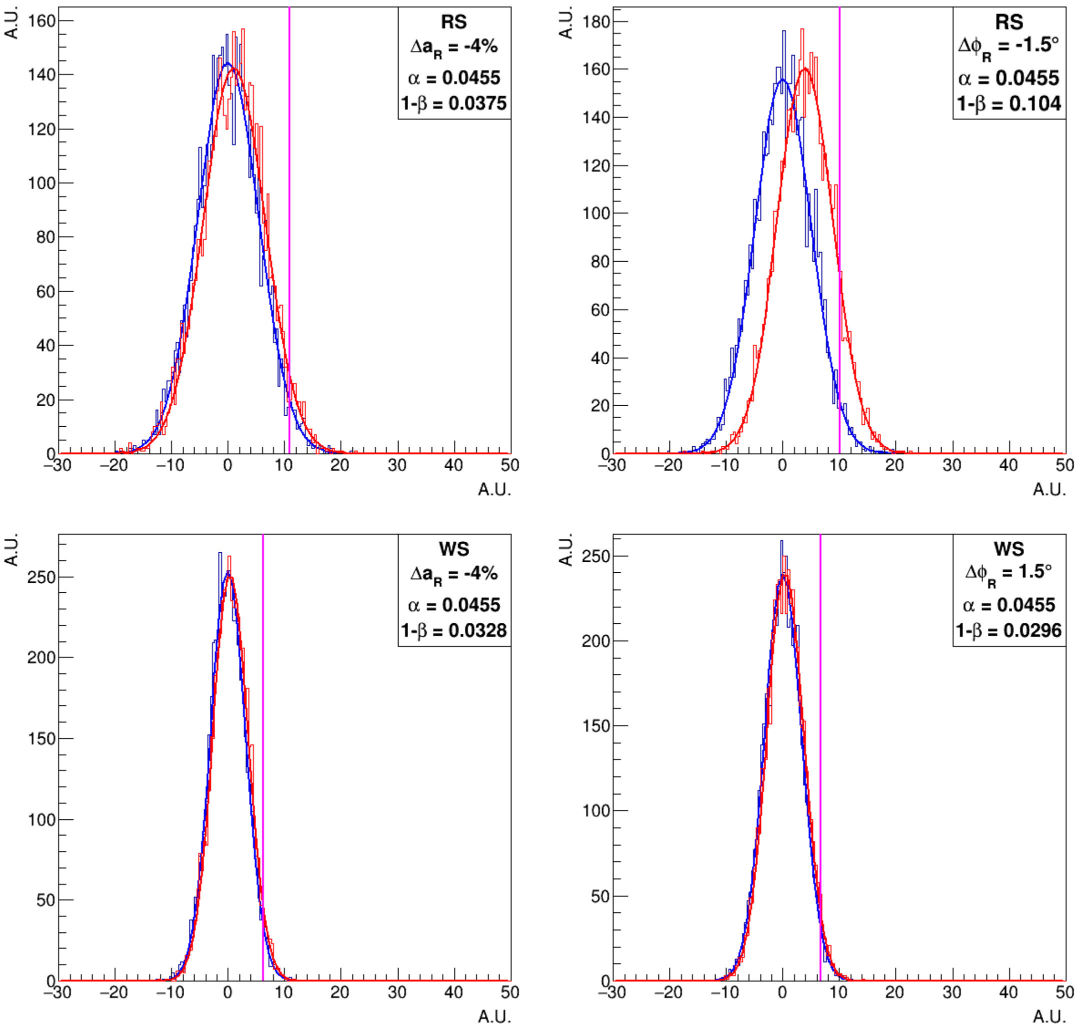


Figure 4.9: Distribution of t_a^{RS} (up left) t_ϕ^{RS} (up right) t_a^{WS} (down left) t_ϕ^{WS} (down right) in *CP*-symmetry hypothesis (blue) and with when *CP* asymmetries are introduced in $K^*(892)^+$ resonance (red).

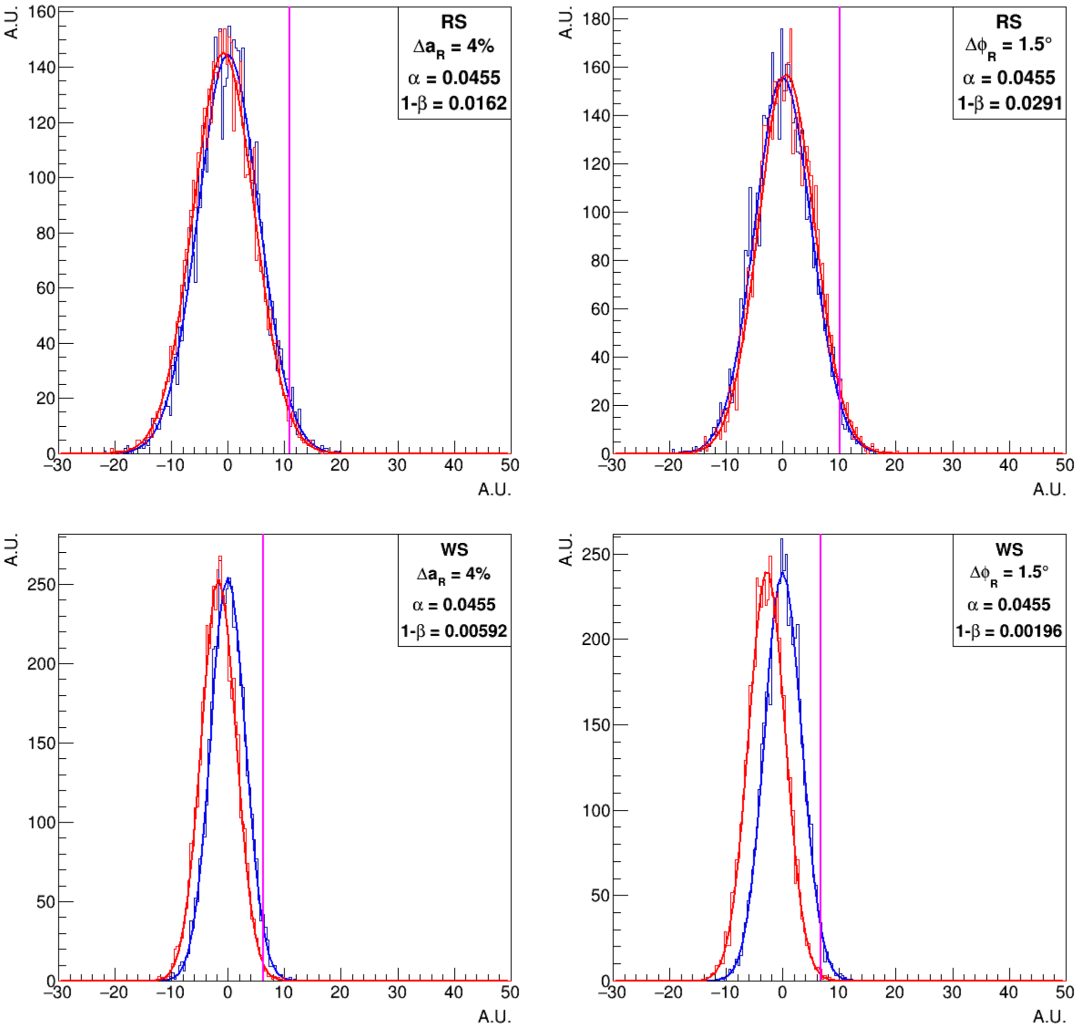


Figure 4.10: Distribution of t_a^{RS} (up left) t_ϕ^{RS} (up right) t_a^{WS} (down left) t_ϕ^{WS} (down right) in *CP*-symmetry hypothesis (blue) and with when *CP* asymmetries are introduced in $R = K^*(1410)^+$ resonance (red).

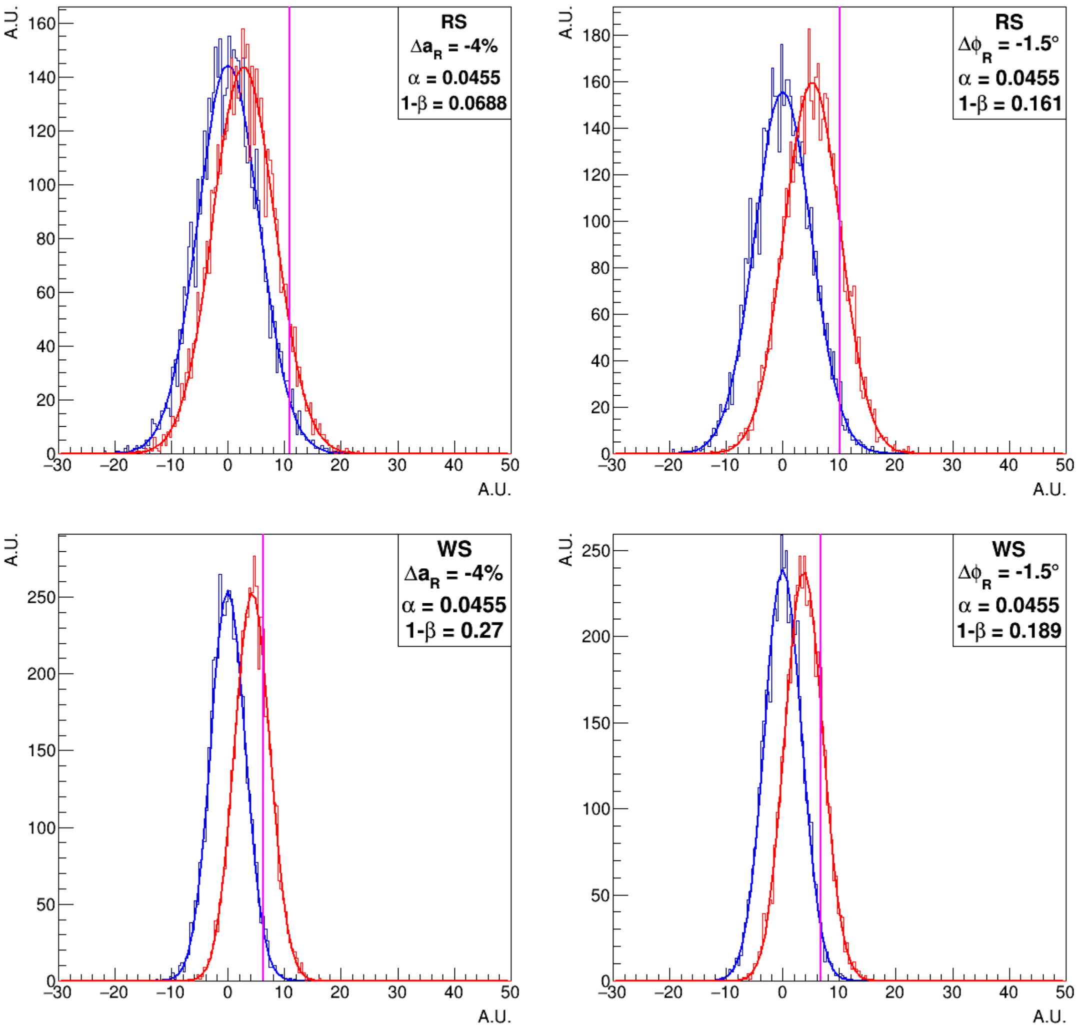


Figure 4.11: Distribution of t_a^{RS} (up left) t_ϕ^{RS} (up right) t_a^{WS} (down left) t_ϕ^{WS} (down right) in *CP*-symmetry hypothesis (blue) and with when *CP* asymmetries are introduced in $R = K_0^*(1430)^+$ resonance (red).

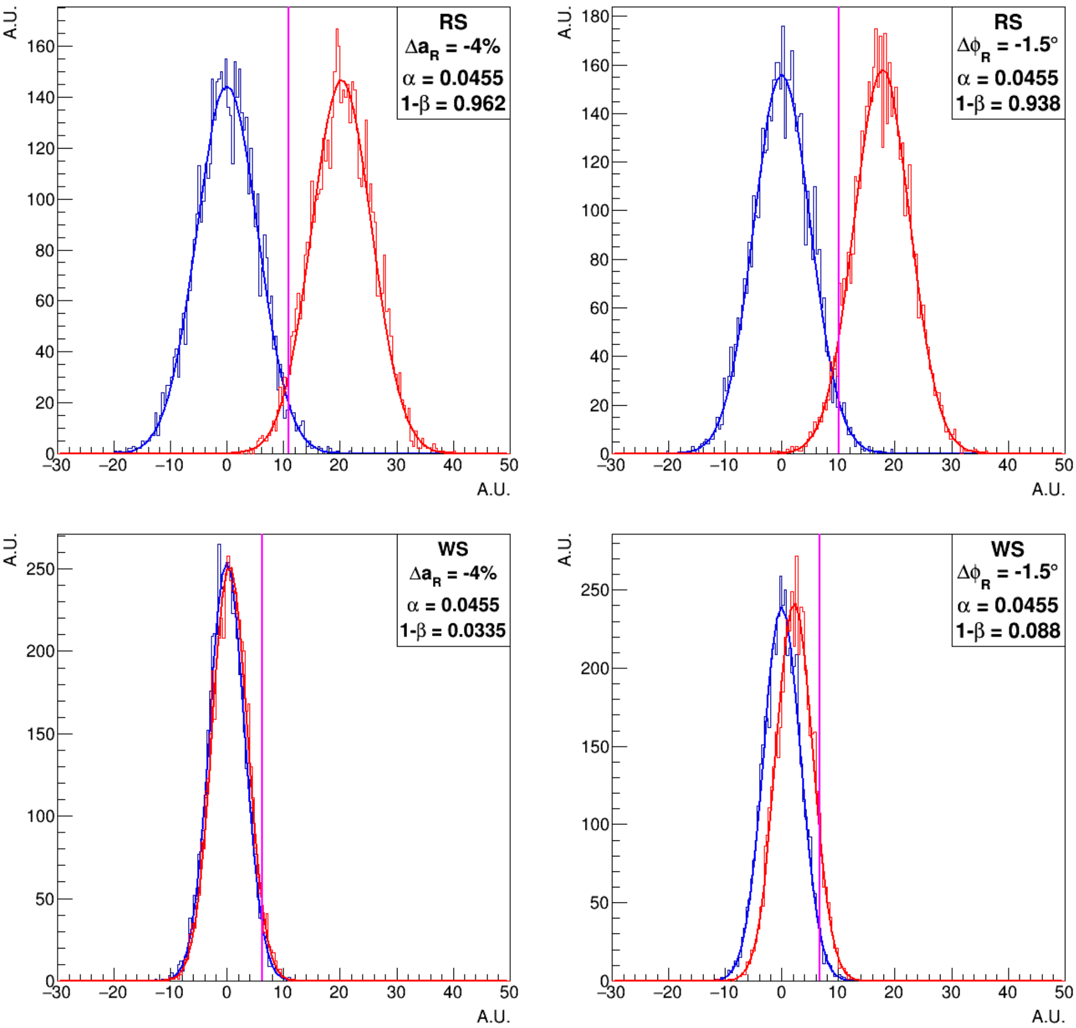


Figure 4.12: Distribution of t_a^{RS} (up left) t_ϕ^{RS} (up right) t_a^{WS} (down left) t_ϕ^{WS} (down right) in *CP*-symmetry hypothesis (blue) and with when *CP* asymmetries are introduced in $R = K^*(1410)^0$ resonance (red).

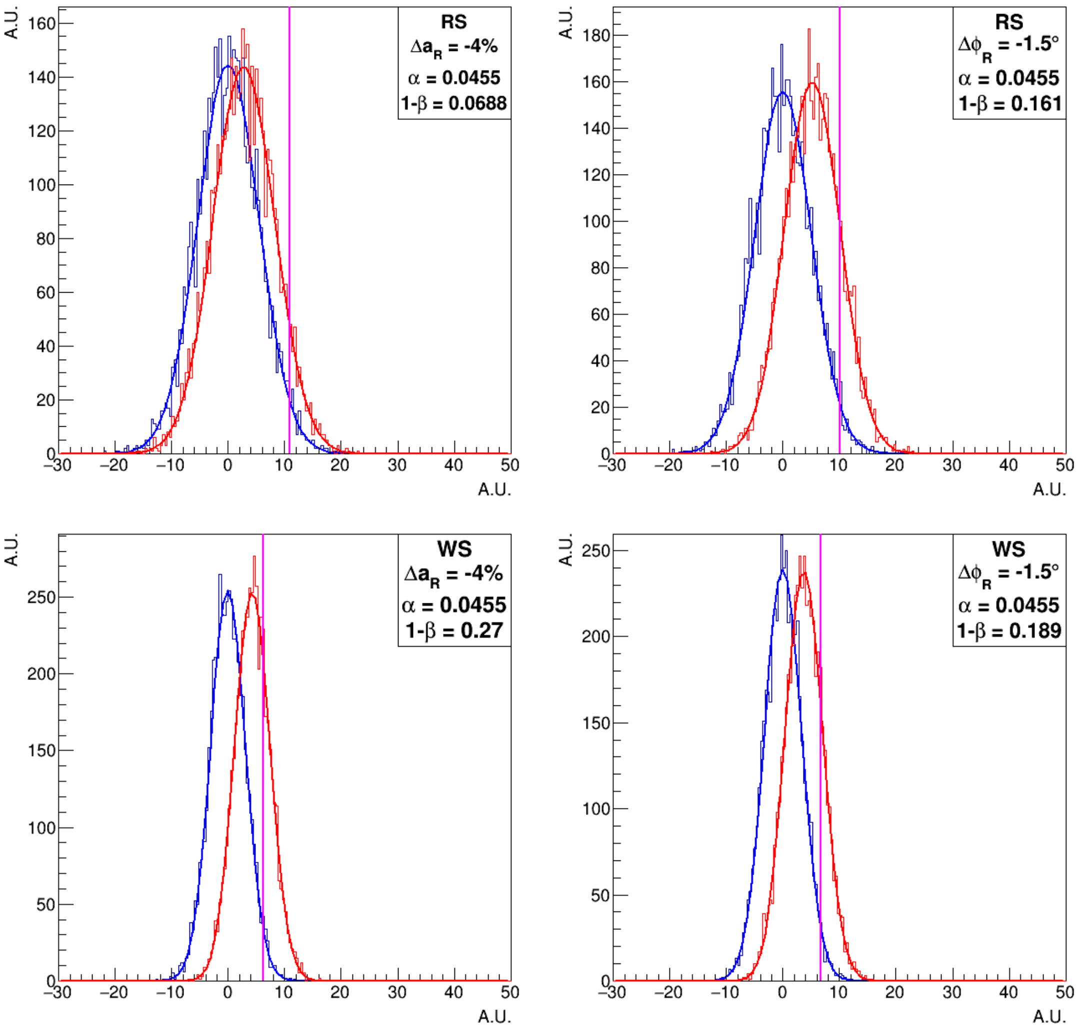


Figure 4.13: Distribution of t_a^{RS} (up left) t_ϕ^{RS} (up right) t_a^{WS} (down left) t_ϕ^{WS} (down right) in *CP*-symmetry hypothesis (blue) and with when *CP* asymmetries are introduced in $R = K_0^*(1430)^0$ resonance (red).

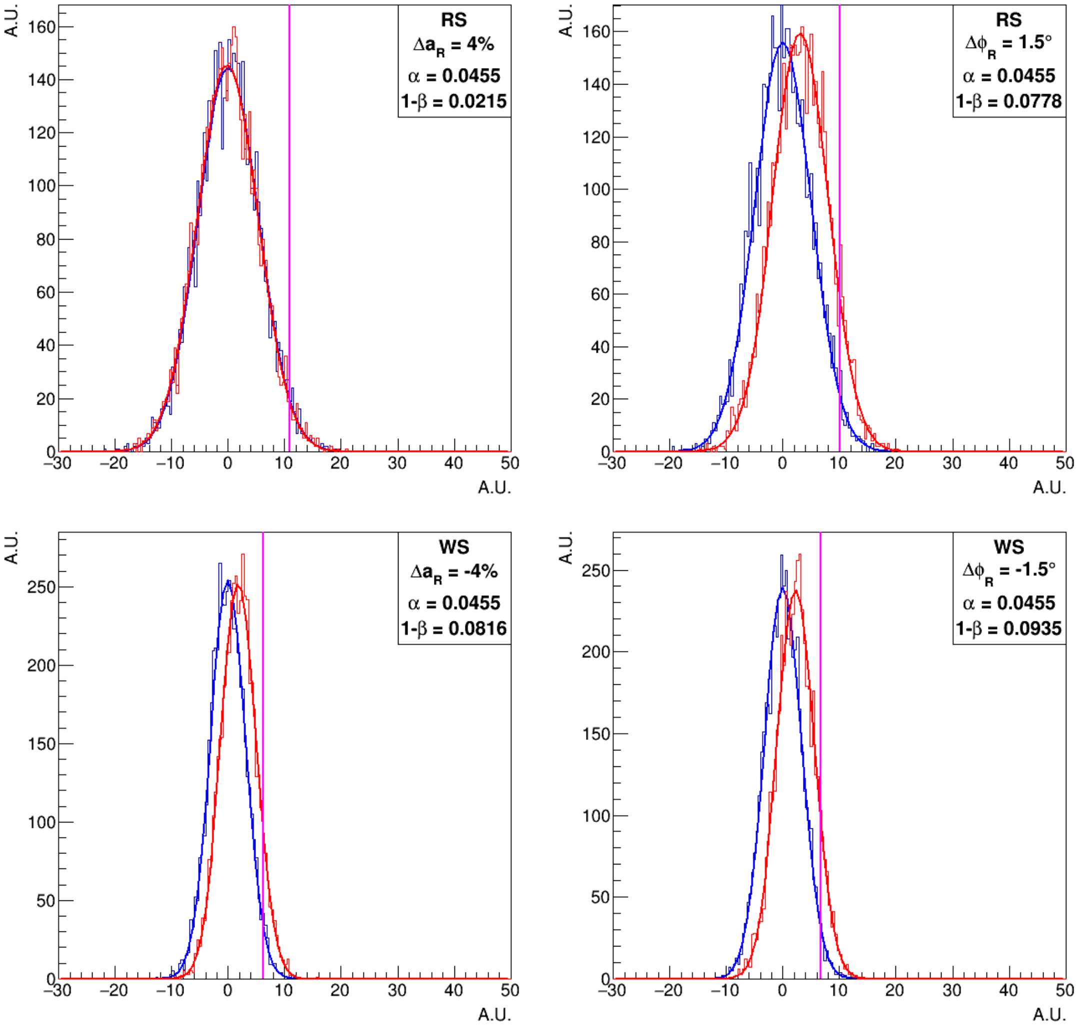


Figure 4.14: Distribution of t_a^{RS} (up left) t_ϕ^{RS} (up right) t_a^{WS} (down left) t_ϕ^{WS} (down right) in *CP*-symmetry hypothesis (blue) and with when *CP* asymmetries are introduced in $R = K_2^*(1430)^0$ resonance (red).

4.4 Data and selection

This analysis uses the LHCb dataset recorded during 2016, 2017, and 2018 (Run 2) corresponding to a total integrated luminosity of $\sim 5.6\,\text{fb}^{-1}$.

4.4.1 Trigger selection

The first decay selection is performed by the trigger. As mentioned before, the LHCb trigger is organised in multiple levels: L0, HLT1, and HLT2. A trigger line is defined as the sequence of algorithms that returns the decision to accept or reject an event according to a particular event topology. The trigger line responsible for accepting an event is stored, this information can be used to restrict the analysis to events collected under well-defined conditions. Given a trigger line and a track (or a combination of tracks), events are classified as Trigger On Signal (TOS) or Trigger Independent-of-Signal (TIS) events [53]. TOS events are triggered on the signal decay chain independently of the presence of other tracks. This condition is fulfilled if the information used to reconstruct the signal tracks is sufficient to satisfy the selection criteria of the respective trigger line. TIS events are triggered independently of the presence of the signal. A candidate is considered to be TIS with respect to a trigger selection if removing it from the event would still cause the trigger selection to accept the event, *i.e.* if the other particles in the event are sufficient to satisfy the trigger selection.

Prior to discussing the details of trigger selection, it is useful to define some relevant quantities used in the present analysis.

Primary Vertex (PV) It is the vectorial position of the reconstructed primary pp interaction. In the cases where it is important to stress the vectorial nature, it will be written $\overrightarrow{PV}$.

Decay Vertex (DV) It is the vectorial position of the reconstructed point where a certain particle X is decayed. In the cases where it is important to stress the vectorial nature, it will be written $\overrightarrow{DV}(X)$. In this analysis, if X is not specified it is referred to the DV of the D^0 particle.

Preudorapidity (η) It is defined as $\eta = -\log(\tan\theta/2)$. It is another way to parametrise the θ angle with respect to the z-axis. It is widely used in particle physics because, in the limit of ultra-relativistic particles, the pseudorapidity approximates the rapidity, hence it is Lorentz invariant for boosts in the z direction.

θ_{DIRA} It is the direction angle, *i.e.* the angle between the momentum of the particle and the displacement vector, defined by the PV and the DV of the particle. For a fully reconstructed particle its total momentum tends to be aligned to the displacement vector, resulting into a θ_{DIRA} close to zero.

Impact Parameter (IP) Distance of closest approach to a particle trajectory to a given point. D^0 daughters have in general large impact parameters with respect to the PV because of the displaced decay vertex of the D^0 meson. Instead, the IP of the D^0 meson with respect to the PV tends to be close to zero within the experimental uncertainty, in the assumption of "prompt" decays (*i.e.* coming from the PV).

Impact Parameter Chi square ($\boldsymbol{\chi^2_{IP}}$) Difference between the χ^2 of the primary vertex fit, obtained with and without considering the particle in the fit. If the particle does not come from the PV, the χ^2_{IP} will be generally larger than the one obtained with a prompt particle.

Flight Distance (FD) Distance travelled by a particle from the production point to the DV.

Flight Distance Chi square ($\boldsymbol{\chi^2_{FD}}$) It is the fit χ^2 to distance between PV and DV of the particle, *i.e.* a measurement of the significance of the displacement vector to be different from zero.

Particle identification (PID$_X$) or Delta-log-likelihood (DLL$_{X\pi}$) Difference between the logarithm of the likelihoods in the X and the π hypothesis. An high value of PID$_X$ means an high probability for the track to be a X particle. The likelihood is associated to a track by combining information from several sub-detectors.

Track ghost probability ($\boldsymbol{\mathcal{P}_{ghost}}$) It is the probability for a track to be a "ghost"track, *i.e.* a misidentified track. This quantity is calculated with a Neural Network [54], that combines information from different variables which describe the track reconstruction and global event properties in order to separate ghost tracks, which are spurious combination of hits, from real tracks.

Decay Tree Fitter (DTF) Algorithm used to refit all the candidates offline. The algorithm takes a complete decay chain, parameterises it in terms of vertex positions, decay lengths and momentum parameters. Then, it fits these parameters simultaneously, taking into account constraints such as measured parameters and 4-momentum conservation at each vertex [55]. In this thesis, when the subscript DTF will be used, it will imply that the constraint on the D^0 and on the soft pion to come from the primary vertex was required.

$\boldsymbol{\Delta m}$ In this analysis it is the difference between the D^* and the D^0 masses obtained using the DTF algorithm. The Δm distribution has a starting point for the pion mass value (139.57 MeV/c^2). In the difference, part of the uncertainties on the D^0 mass cancels out, allowing Δm to have a much better mass resolution than D^* mass.

Vertex fit χ^2/ndf It is the χ^2 value of the vertex fit, normalised to the degrees of freedom. The fitter takes as input a vector of particles, performs the fit and updates the mother particle [56].

L0 requirements

To avoid introducing hard-to-simulate effects in the decay kinematics, I required `L0Global` to be TIS on D^* signal. This is a conservative decision that select the 85% of the initial sample. I considered to also keep events selected by `L0Hadron` TOS on the $K\pi$ pair, but, since its is optimised to select rare events, its threshold on transverse energy ($E_T \geq 3.7\,\text{GeV}$) is high with respect to HLT2 requirement on the transverse momentum of D^0 ($p_\text{T} > 1\,\text{GeV}/c$).

HLT1 requirements

At the HLT1 level, it is required that at least one between the $K^\pm$ and the $\pi^\pm$ daughters of the D^0 fired the `Hlt1TrackMVA` line or, alternatively, that the combination of the two particles fired the `Hlt1TwoTrackMVA` line. The requirements made by these trigger lines aim at selecting respectively one or two detached high-momentum good-quality long tracks, in order to identify D^0 decays independently from which L0 trigger line fired.

The `Hlt1TrackMVA` trigger line selects a single long track that satisfies the following requirements:

- A least 9 hits in the VELO.
- Track fit $\chi^2/\text{ndf} < 2.5$.
- $\mathcal{P}_{ghost} < 0.4(0.2)$ depending on data acquisition time.
- $p > 3(5)\,\text{GeV}/c$ depending on data acquisition time.
- The tracks is required to be displaced with respect to the PV.

The latter statement is ensured though the multivariate cut

$$\begin{gathered}\Big\{p_\text{T} > 25 \wedge \chi^2_\text{IP} > 7.4\Big\}\vee \\ \Big\{[1 < p_\text{T} < 25]\wedge\Big[\ln\chi^2_\text{IP} > \ln 7.4 + \frac{1}{(p_\text{T}-1.0)^2} + \alpha\Big(1-\frac{p_\text{T}}{25}\Big)\Big]\Big\}\end{gathered} \tag{4.16}$$

where p_T is the transverse momentum expressed in GeV/c, χ^2_IP is the significance of the impact parameter (IP) with respect to the primary vertex in the event for which the IP significance is the minimum, and α varies during the Run 2 data taking. The boundaries of the selected region are illustrated in Fig. 4.15 for different values of α.

The `Hlt1TwoTrackMVA` trigger line is designed to select a combination of two good-quality long tracks. Each track must satisfy the following requirements:

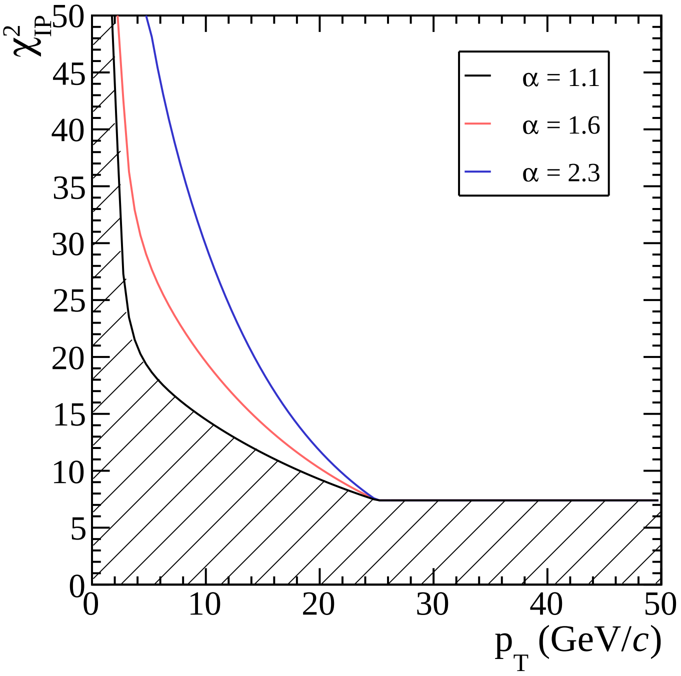


Figure 4.15: Boundaries of the selected region in the track $\chi^2_{\text{IP}} - p_{\text{T}}$ plane for the Hlt1TrackMVA line. The shaded area represents the excluded region with the $\alpha = 1.1$ requirement.

- $p_{\text{T}} > 0.5(0.6)\,\text{GeV}/c$ depending on data acquisition time.
- $p > 3(5)\,\text{GeV}/c$ depending on data acquisition time.
- $\mathcal{P}_{ghost} < 0.4(0.2)$ depending on data acquisition time.
- Track fit $\chi^2/\text{ndf} < 2.5$.
- $\chi^2_{\text{IP}} > 4$.

The combination of the two tracks must satisfy the following requirements:

- $p_{\text{T}}(\text{trk}_1{+}\text{trk}_2) > 2\,\text{GeV}/c$.
- Vertex(trk$_1$, trk$_2$) fit $\chi^2 < 10$.
- Vertex(trk$_1$, trk$_2$) $\eta \in [2, 5]$.
- $m_{corr}(\text{trk}_1{+}\text{trk}_2) > 1\,\text{GeV}/c$.
- $\theta_{DIRA} > 0$.
- Output of the classifier $> 0.95(0.97)$ depending on data acquisition time.

The classifier is a BDT that use as input the χ^2 of the two-track vertex, the distance between the primary vertex (PV) and the two-track vertex, the sum of the p_{T} of the two-track, the number of tracks with $\chi^2_{\text{IP}} > 16$.

HLT2 requirements

The analysis relies on the following Turbo trigger lines:

- `Hlt2CharmHadDstp2D0Pip_D02KS0KmPip_KS0DDTurbo`
- `Hlt2CharmHadDstp2D0Pip_D02KS0KmPip_KS0LLTurbo`
- `Hlt2CharmHadDstp2D0Pip_D02KS0KpPim_KS0DDTurbo`
- `Hlt2CharmHadDstp2D0Pip_D02KS0KpPim_KS0LLTurbo`

K^0_S candidates are reconstructed in the $K^0_S \to \pi^+\pi^-$ decay mode. The different trigger lines select separately the Right-Sign (RS) and the Wrong-Sign (WS) samples defined in Section 4.1 with the pions produced in the K^0_S decay reconstructed from long tracks or Downstream tracks. Respectively they select:

- Right-Sign sample with K^0_S reconstructed from Downstream tracks.
- Right-Sign sample with K^0_S reconstructed from long tracks.
- Wrong-Sign sample with K^0_S reconstructed from Downstream tracks.
- Wrong-Sign sample with K^0_S reconstructed from long tracks.

Multiple files are produced according to the trigger line, the polarity of the magnet, and the data acquisition year.

Table 4.4 reports the HLT2 selection requirements.

4.4.2 Offline selection

Initially, some base offline cuts are applied in order to reduce the main physics backgrounds and equalise HLT2 selections among data taking periods. To suppress background from $D^0 \to K^-\pi^+\pi^+\pi^-$ decay, I apply a cut on χ^2_{FD} of the K^0_S. Figure 4.16 shows that pions pair under the cut do not produce a peak around the K^0_S mass. I also select decays with K^0_S mass within 2 σ from the peak. Table 4.5 summarises the base offline cuts.

Additional cuts are then applied to maximise the score function

$$\mathcal{R} = \frac{S}{\sqrt{S+B}}, \tag{4.17}$$

of the Δm distribution, where S and B are respectively the number of reconstructed signal and background candidates in the region $\Delta m \in [144.625, 146.125]\,\mathrm{MeV}/c^2$, a symmetrical range around the modal value of the distribution in order to reach the 95% of the signal. I optimised the score function comparing its value with different rectangular cuts over kinematics and track quality variables of the different particles. Due to the large number of variables, I factorised the whole space of variables in sub-spaces, that can be considered as independent with a good level of approximation.

Candidate	Variable	HLT2 Requirement	Units
$\pi^{\pm}$ from K^0_{S}	Track χ^2/ndf	$< 3^L$	-
		$< 4^D$	-
	χ^2_{IP}	$> 36^L$	-
	p	$> 3^D$	GeV/c
	p_{T}	$> 175^D$	MeV/c
K^0_{S}	$\lvert m(\pi^+\pi^-) - m^{PDG}_{K^0_S}\rvert$	$< 35^L$	MeV/c^2
		$< 64^D$	MeV/c^2
	Vertex-fit χ^2/ndf	< 30	-
	Decay time wrt primary vertex	$> 2^L$	ps
		$> 0.5^D$	ps
	z position of the vertex	$[-0.1,0.5]^L$	m
		$[0.4, 2.275]^D$	m
$h^{\pm}$ from D^0	Track fit χ^2/ndf	< 3	-
	$\mathcal{P}_{ghost}$	< 0.4	-
	p	> 1	GeV/c
	p_{T}	> 200	MeV/c
	IP χ^2	> 4	-
	PID$_K$	< 5 π	-
		> 5 K	-
D^0	Mass	[1.765, 1.965]	GeV/c^2
	Vertex fit χ^2/ndf	< 20	-
	p_{T}	> 1.8	GeV/c^2
	θ_{DIRA}	< 34.6	mrad
	Decay time	> 0.1	ps
	Flight-distance χ^2	> 20	-
$\pi^{\pm}_{soft}$	Track fit χ^2/ndf	< 3	-
	$\mathcal{P}_{ghost}$	$< 0.4^{'16}, 0.25^{'17,'18}$	-
	p	> 1	GeV/c
	p_{T}	$> 100^{'16}, 200^{'17,'18}$	MeV/c
$D^{*\pm}$	Vertex fit χ^2/ndf	< 10	-
	Δm	[135, 165]	MeV/c^2

Table 4.4: HLT2 selection requirements. $h^{\pm}$ stands for the charged particles from the D^0 decay, the K and π. The superscripts L or D specify the type of track to which the requirement is applied. The superscripts '16, '17 and '18 stand for the year in which a certain threshold is chosen. If it is not specified by any kind of superscript, the requirement is on both types of tracks and the threshold is the same over all the data taking period.

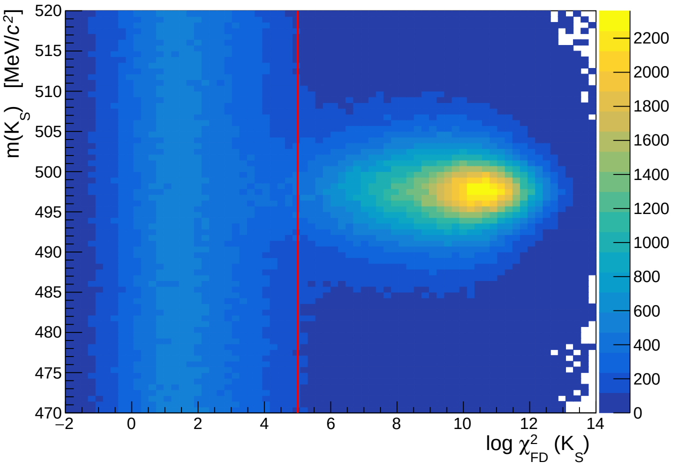


Figure 4.16: 2D distribution of the m(K^0_S) and $\log \chi^2_{FD}(K^0_S)$. The vertical red line indicates the cut $\log \chi^2_{FD}(K^0_S) > 5$.

Candidate	Variable	Requirement	Units
D^0	Lifetime wrt best PV	$[0.3, 8]\tau$	-
	χ^2_{IP}	< 9	-
$h^\pm$ from D^0	PID_K	$< -5\ \pi^\pm$	-
	p	> 5	GeV/c
	p_{T}	> 800	MeV/c
	η	$[2, 4.2]$	-
K^0_S	mass	$[485, 510]$	MeV/c^2
	$\log(\chi^2_{FD})$ wrt origin vertex	> 5	-
$\pi^\pm$ from K^0_S	η	$[2, 4.2]$	-
$\pi^\pm_{soft}$	p_{T}	> 200	MeV/c
	$\mathcal{P}_{ghost}$	< 0.25	-

Table 4.5: Base offline cuts.

The optimisation is performed on each sub-space without cutting on the variable of others sub-spaces, then I checked the accuracy of such an approximation comparing different rectangular cuts over the variables that mostly affect the score function.

To optimise the cuts on variables related to D^0 and D^0 daughters I optimised the score function of the D^0-mass distributions. In this case the number of reconstructed signal and background candidates are counted in the region $m(D^0) \in [1854.03, 1875.63]\,\text{MeV}/c^2$. Table 4.6 lists the variables related to D^0 and D^0 daughters, the range of their distributions, the range for the rectangular cuts (for each variables 10 bins are scanned plus the case without cut), the optimal cut, and the sub-space for the optimisation.

Candidate	Variable	Distribution range	Cut (n.bin 10) range	Result	Units	Sub-space
K^0_S	p	$[3, 180]$	$[5, 100]$	-	GeV/c	1
	p_T	$[0, 8000]$	$[0, 500]$	> 200	MeV/c	1
	Vertex fit χ^2/ndf	$[0, 30^*]$	$[0, 30^*]$	-	-	1
$\pi^\pm$ of K^0_S	Track χ^2/ndf	$[0, 3^*]$	$[0, 3^*]$	-	-	1
	$\cos\theta_{\pi\pi}$	$[0.97, 1]$	$[0.97, 1]$	-	-	1
$h^\pm$ of D^0	p	$[5^*, 140]$	$[5^*, 100]$ $\pi^\mp$	-	GeV/c	2
		$[5^*, 140]$	$[5^*, 100]$ $K^\pm$	-	GeV/c	2
	p_T	$[800^*, 9000]$	$[800^*, 4000]$ $\pi^\mp$	-	MeV/c	2
		$[800^*, 9000]$	$[800^*, 4000]$ $K^\pm$	-	MeV/c	2
	χ^2_{IP}	$[4^*, 10000]$	$[4^*, 500]$ $\pi^\mp$	-	-	2
		$[4^*, 10000]$	$[4^*, 500]$ $K^\pm$	-	-	2
	$\cos\theta_{K\pi}$	$[0.98, 1]$	$[0.99, 1]$	-	-	2
	PID_K	$[-100, 5^*]$	$[-15, 5^*]$ $\pi^\mp$	-	-	3
		$[5^*, 100]$	$[5^*, 20]$ $K^\pm$	-	-	3
	Track χ^2/ndf	$[0, 3^*]$	$[0, 3^*]$ $\pi^\mp$	-	-	3
		$[0, 3^*]$	$[0, 3^*]$ $K^\pm$	-	-	3
	$\mathcal{P}_{GHOST}$	$[0, 0.4^*]$	$[0, 0.4^*]$ $\pi^\mp$	-	-	3
		$[0, 0.4^*]$	$[0, 0.4^*]$ $K^\pm$	-	-	3
D^0	p	$[15, 350]$	$[20, 200]$	-	GeV/c	4
	p_T	$[1.8^*, 16]$	$[1.8^*, 3]$	> 2.52	GeV/c	4
	Vertex fit χ^2/ndf	$[0, 20^*]$	$[0, 20^*]$	-	-	4
	FD χ^2	$[0, 400]$	$[0, 100]$	-	-	4
	FD	$[0, 1000]$	$[0, 200]$	-	mm	4
	χ^2_{IP}	-	$[0, 9^{**}]$	-	-	4

Table 4.6: Result of the D^0-mass distributions optimisation process. $h^\pm$ stands for the charged particles from the D^0 decay, the K and π. *Trigger Cut. **Baseline cut.

I applied the two optimised cuts $p_T(K^0_S) > 200\,\text{MeV}/c$ and $p_T(D^0) > 2520\,\text{MeV}/c$ and a cut around the D^0 mass peak: $m(D^0) \in [1832.43, 1897.23]\,\text{MeV}/c^2$. Then I optimised the score function of Δm distribution scanning the cuts on variables related to the D^* and the π_{soft}. Table 4.7 lists the variables related to D^* and π_{soft}, the range of their distributions, the range for the rectangular cuts (for each variables 10 bins are scanned plus the case without cut), the optimal cut.

The previous optimisation steps evidenced 4 cuts having a significant impact on the score function of the D^0-mass and Δm distributions. In order to check the accuracy of the space factorisation I re-ran the optimisation procedure over these variables. The results, shown in Table 4.8, are very similar to the results obtained when optimising the observables relative to the D^0 candidates and to its decays products independently from those relative to the D^* candidates. The values of

Candidate	Variable	Distribution range	Cut (n.bin 10) range	Result	Units
D^*	$DTF\ \chi^2$	$[0, 5000]$	$[10, 50]$	-	-
	DTF Vertex fit χ^2/ndf	$[0, 500]$	$[0, 10]$	-	-
π_{soft}	Track χ^2/ndf	$[0, 3^*]$	$[0, 3^*]$	-	-
	PID_e	$[-15, 15]$	$[-5, 10]$	< 5.5	-
	$\chi^2_{\rm IP}$	$[0, 120]$	$[0, 100]$	< 40	-
	PID_K	$[-40, 30]$	$[-30, 20]$	-	-
	$\mathcal{P}_{Ghost}$	$[0, 0.25^*]$	$[0, 0.25^*]$	-	-
	$p_{\rm T}$	$[100, 1100]$	$[200, 1000]$	-	MeV/c

Table 4.7: Result of the Δm distribution optimisation. *Trigger requirement.

$\mathcal{R}$, computed in the two configurations are indeed very similar: from $\mathcal{R} = 60.83$ to $\mathcal{R} = 60.96$ after the final optimisation.

Candidate	Variable	Cut (n.bin 10) range	Result	Units
K^0_S	$p_{\rm T}$	$[100, 600]$	> 350	MeV/c
D^0	$p_{\rm T}$	$[2, 3]$	> 2.5	GeV/c
π_{soft}	PID_e	$[-5, 6]$	< 4.9	-
	$\chi^2_{\rm IP}$	$[0, 100]$	< 40	-

Table 4.8: Result of the final optimisation.

Table 4.9 reports the complete set of off-line requirements.

A single D^0 candidates may be associated to more than one soft pion, resulting in multiple candidates within the same event, and multiple D^0 can be produced in the same event. I randomly chose just one decay candidate for each event number. Since the RS $D^0 \to K^0_S K^- \pi^+$ decay could be classified as a WS $\overline{D}^0 \to K^0_S K^- \pi^+$ decay associating the D^0 to a π^-_{soft}, I searched for multiple candidates in the full data sample. I stored the number of occurrences of each event scanning all the files produced by the different trigger lines (RS-WS and LL-DD). Then for each event number I extracted one occurrence, and fill new files with these entries. The fraction of multiple candidates was found to be $\sim 10\%$.

Figure 4.17 shows the Δm distribution of the selected data for RS and WS samples. Figure 4.18 shows the Dalitz plot of the data keeping only the candidates in the Δm peak region $(144.82, 146.04)\,\mathrm{MeV}/c^2$. Figure 4.19 shows the projection of Dalitz plots separately for the regions $m^2(K^0_S\pi^+) \in (1.2, 1.85)\,\mathrm{GeV}^2/c^4$ (up) and $m^2(K^0_S\pi^+) \in (0.45, 0.65)\,\mathrm{GeV}^2/c^4$ (down), and for the RS (left) and WS (right) samples.

Candidate	Variable	Requirements	Units
$\pi^\pm$ of K^0_S	η	$[2, 4.2]$	-
K^0_S	$mass$	[485,510]	MeV/c^2
	$\log(\chi^2_{FD})$ wrt origin vertex	> 5	-
	p_T	> 350	MeV/c
$h^\pm$ of D^0	p	> 5	GeV/c
	p_T	> 800	MeV/c
	$DLL_{K\pi}$	$< -5\ \pi^\pm$	-
	η	$[2, 4.2]$	-
D^0	Vertex-fit χ^2 /ndf	< 4.6	-
	p_T	> 2.5	GeV/c^2
	Lifetime wrt best PV	$[0.3, 8]\tau$	-
	χ^2_{IP}	< 9	-
π_{soft}	$\mathcal{P}_{ghost}$	< 0.25	-
	p_T	> 200	MeV/c
	PID_e	< 4.9	-
	χ^2_{IP}	< 40	-
$D^{*\pm}$	DTF Vertex fit χ^2/ndf	< 3	-

Table 4.9: Summary of the off-line selections. $h^\pm$ stands for the charged particles from the D^0 decay, the K and π.

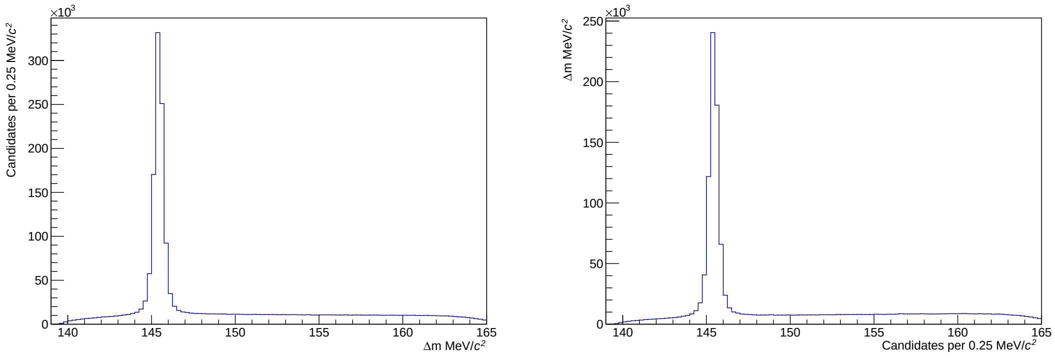


Figure 4.17: Δm distribution of the selected RS (left) and WS (right) data.

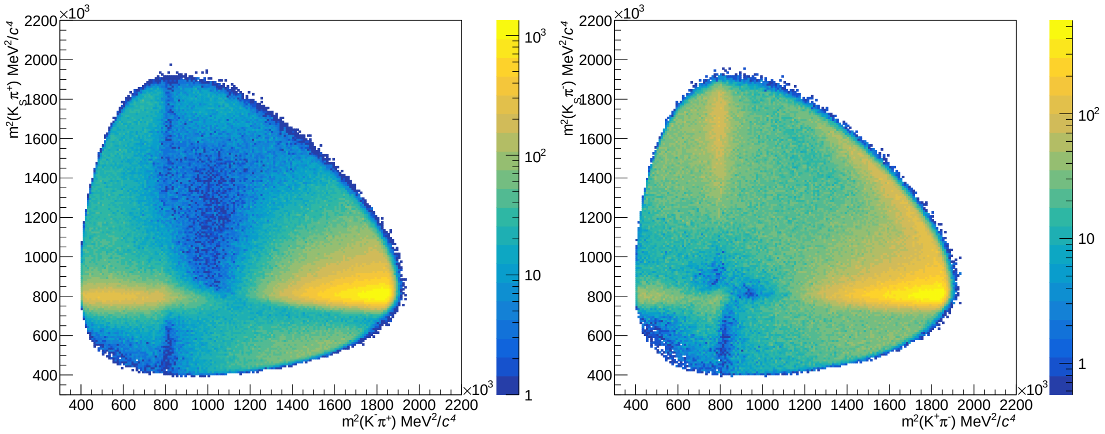


Figure 4.18: Dalitz plots of the selected RS (left) and WS (right) data.

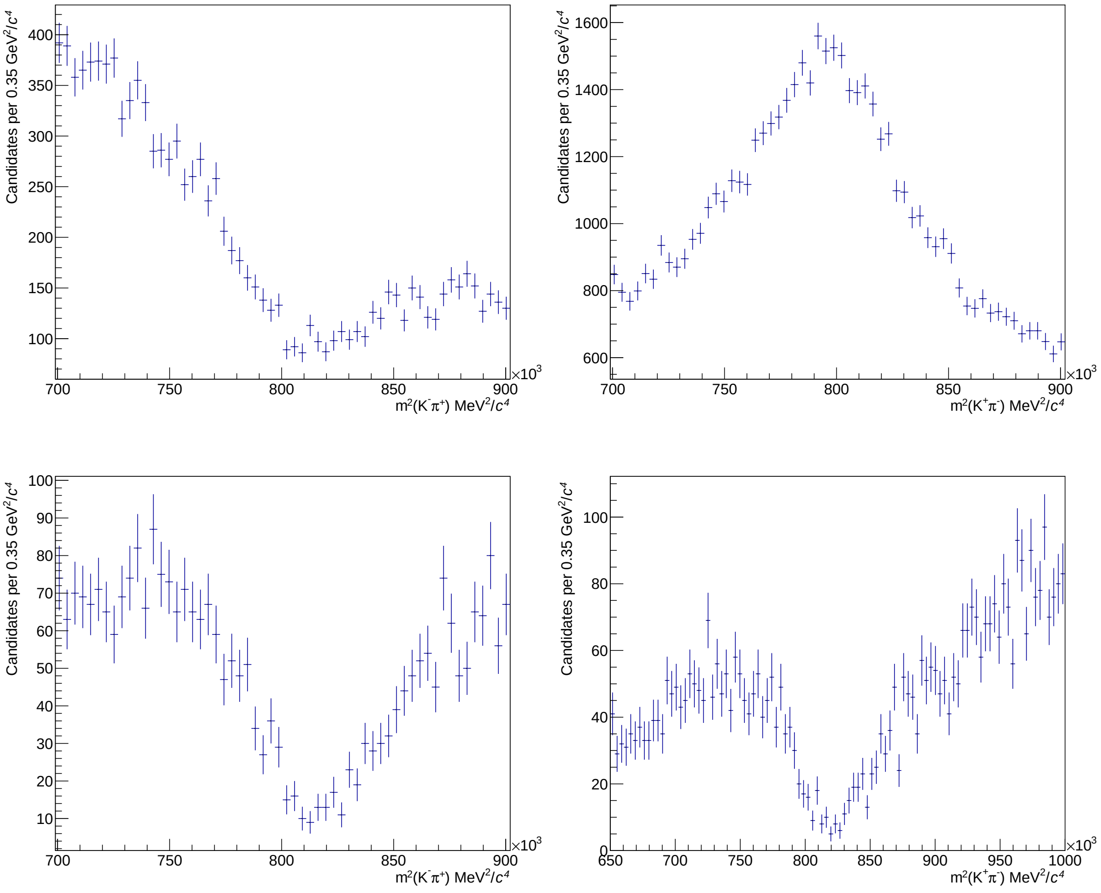


Figure 4.19: Projection of Dalitz plots. $m^2(K^0_{\rm S}\pi^+) \in (1.2, 1.85)\,{\rm GeV}^2/c^4$ (up) and $m^2(K^0_{\rm S}\pi^+) \in (0.45, 0.65)\,{\rm GeV}^2/c^4$ (down). Separated in RS (left) and WS (right) samples.

4.5 Statistical uncertainties

In Section 4.3.1 I obtained an estimate of the resolution expected on $\Delta a_{K^*(892)^0}$ and $\Delta\phi_{K^*(892)^0}$, by generating random samples from the amplitude model distribution. I now perform a direct determination of the uncertainty on the actual data sample. Since the measurement procedure is not based on a fit, but rather the simple measurement of a statistic, I determine the statistical uncertainty via a bootstrap procedure.

From the Dalitz plot of the data sample I obtained the average distribution $f(x)$ of the data. Then I randomly extracted two samples (S^+ and S^-) distributed as $f(x)$, and with size extracted from a Poisson distribution with mean equal to the half of the data sample. This amounts to the very reasonable assumption that the observed *CP* asymmetry will be small, and allows to determine the statistical uncertainty without unblinding the result.

By repeating the extraction and measurements steps 5k times I obtain the distributions of the t observables. The σ of the distributions, divided by the p_1 value reported in Table 4.1, gives the statistical uncertainty on $\Delta a_{K^*(892)^0}$ and $\Delta\phi_{K^*(892)^0}$.

Table 4.10 reports the statistical uncertainties measured in this way, and compares them with the estimate from the model and the results actually obtained in the Run 1 analysis.

		σ_{H_0} model	σ_{H_0} data	σ_Δ model	σ_Δ data	σ_Δ Run 1
RS	a	5.49	6.71	0.0090	0.011	0.031
	ϕ	4.93	6.58	0.35°	0.47°	1.6°
WS	a	3.12	3.55	0.011	0.012	0.024
	ϕ	3.28	4.75	0.34°	0.50°	1.8°

Table 4.10: Statistical uncertainty on $\Delta a_{K^*(892)^0}$ and $\Delta\phi_{K^*(892)^0}$, measured with the model and with the data, and the statistical uncertainty in Run 1 analysis.

The resolution predicted by the analytical model of the Dalitz distribution is very much in line with the observed result (with a slight deterioration presumably due to the presence of background in the data, that is not included in the model). The resolutions scales up as expected by the increase of data sample, thus confirming that the custom measuring methodology developed for this measurement does not lose power in comparison with the full Dalitz fit utilised in the Run 1 measurement.

4.6 Systematic uncertainties

In this section I highlights the main systematic effects affecting this measurement, and possible strategies for precisely assessing the associated uncertainties.

CPV from Other resonances

In section 4.3.3 I verified that the observable t has little sensitivity to the possible presence of *CPV* in other resonances, except for $K^*(1410)^0$ in the RS sample. This is large enough to require a subtraction. A strategy is to perform a determination of the asymmetry of $K^*(1410)^0$ with a method strictly analogous to what I have described for the $K^*(892)^0$, and then subtract its effect. Defining an additional set of observables

$$t_{K^*(1410)^0} = \frac{1}{N_{S^+}} \sum_{x \in S^+} \frac{g_{K^*(1410)^0}(x)}{f_a(x)} - \frac{1}{N_{S^-}} \sum_{x \in S^-} \frac{g_{K^*(1410)^0}(x)}{f_a(x)} \tag{4.18}$$

where S^+ and S^- are the data samples of $D^0 \to K^0_S K \pi$ and $\overline{D}^0 \to K^0_S K \pi$ decays limited to the region highlighted in Figure 4.20 where is located the $K^*(1410)^0$ distribution peak, and

$$g_{K^*(1410)^0}(x) = f_m^+(x) - f_m^-(x)$$

assuming that *CP* is violated on $K^*(1410)^0$ resonance. Measure the value of $t_{K^*(1410)^0}$ corresponds to measure the value of $\Delta a_{K^*(1410)^0}$ and $\Delta \phi_{K^*(1410)^0}$, allowing to correct the measurement of t. Other resonances with smaller impact on the observable will be dealt with by varying their asymmetry within reasonable ranges and using the results as systematic uncertainties.

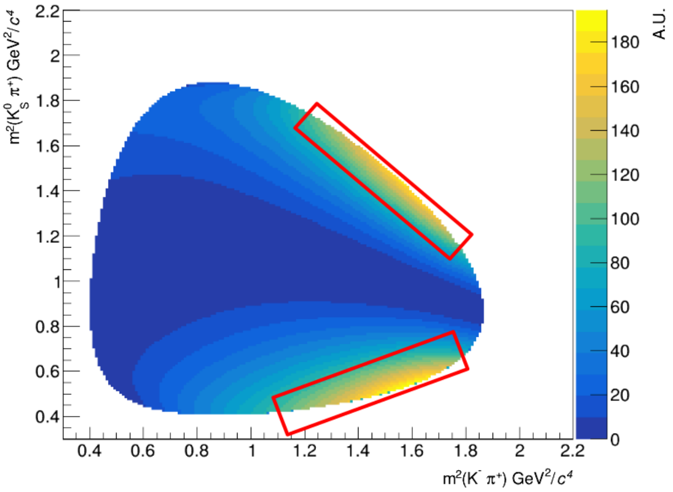


Figure 4.20: Dalitz plot of the $D^0 \to K^0_S K^*_0(1410)^0$ decay. The red box indicates he regions used to evaluate $t_{K^*(1410)^0}$.

Dalitz distribution uncertainty

The shape of the Dalitz distribution is affected by some uncertainty, from the modelling of the physics involved, the experimental uncertainty on the parameter values, and the efficiency function. All these effects have a direct influence on the definition of the t observable; however, any deviation from the assumptions from the reality would only have the effect of decreasing the optimality of the choice, but would not affect the validity of the *CPV* test.

To the second order, an uncertainty can be introduced in the calibration function that relates the observed value of t to the *CPV* parameters of the resonance amplitude, and this is an effect that needs assessing.

There may be other possible weak dependencies of the result on the assumed shape of the model, but I expect them to be negligible in comparison to the effects mentioned here.

This systematic uncertainty can be assessed by varying the modelling function parameterisation assumptions, and their parameters within uncertainties determined from the results of the Run 1 analysis.

Asymmetries in detection efficiency

The method that I introduced is intrinsically insensitive to an overall charge asymmetry in the sample, that can be induced by detection or selection effects. However, those asymmetries are not necessarily uniform over the Dalitz plot, and this can potentially create deviations of t from 0 even in the absence of any physical *CP* asymmetry. It is therefore important to assess their effects.

To this purpose, I used a control sample of fake D^0 candidates due to combinatorial background. This class of events will suffer the same detection and selection asymmetries of the signal, but is obviously free from any asymmetry resembling the *CPV*-induced patterns I am looking for. Note that real K^{*0} particles can and should be present, but they will not interfere with other states. An overall charge asymmetry is definitely possible, but I have already seen that it does not affect the observables.

As a combinatoric sample, I chose the sideband region $m(D^0) \in [1907.5, 1940.0]\,\text{MeV}/c^2$ of the $m(D^0)$ distribution (Figure 4.21). I avoided the region $m(D^0) \in [1800, 1840]\,\text{MeV}/c^2$ and $m(D^0) > 1940\,\text{MeV}/c^2$ because these regions contain some partially or wrongly reconstructed decays, that would not be purely combinatoric in nature. The sizes of the sideband samples in the region of interest are $3.5 \cdot 10^4$ and $2.5 \cdot 10^4$, respectively for the RS and WS channel. For comparison the sizes of the data samples in the signal region are $3.4 \cdot 10^4$ and $6.5 \cdot 10^4$ for the RS and WS channel respectively.

I produced a t distribution from these data with a bootstrap procedure: I randomly extracted 1'000 D^0 and $\overline{D}^0$ samples with the same distribution of the sideband data sample without use the D^0 and $\overline{D}^0$ tag, and I measured the value of t of these samples obtaining the distribution in no *CPV* hypothesis. Then I measured the value of the observables t of the sideband data sample taking in account the D^0 and $\overline{D}^0$ tag.

The values of the observables t obtained with this test are compatible with zero, as seen in figure 4.23. This shows that the systematic effects due to asymmetry disuniformities is below the level of the statistical uncertanty, so it will not be a dominant effect in the result. It will still be needed to determine its value quantitatively; this can be done generating data sets with the simulated Dalitz distribution, and tagging the entries of the sets according to the asymmetry of the sideband D^0 and $\overline{D}^0$ distribution in the corresponding point of the Dalitz phase space.

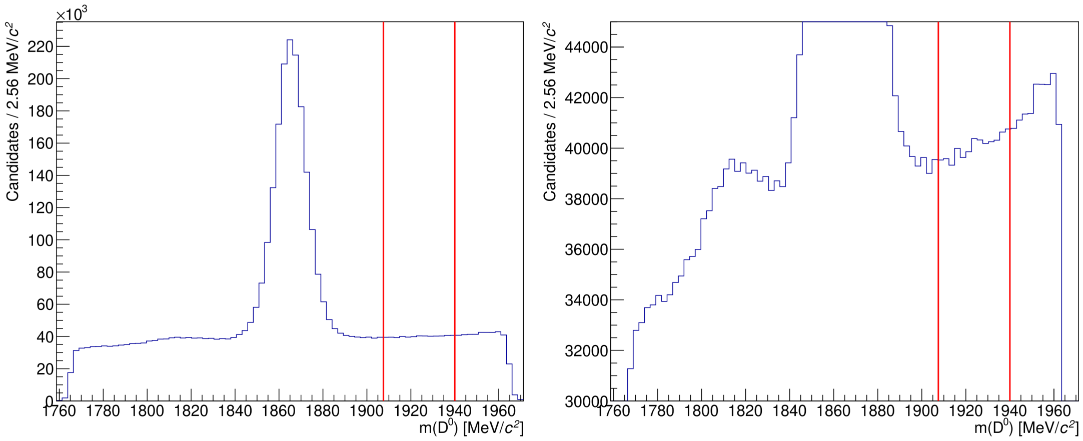


Figure 4.21: $m(D^0)$ distribution with the selected region for the test on the sideband.

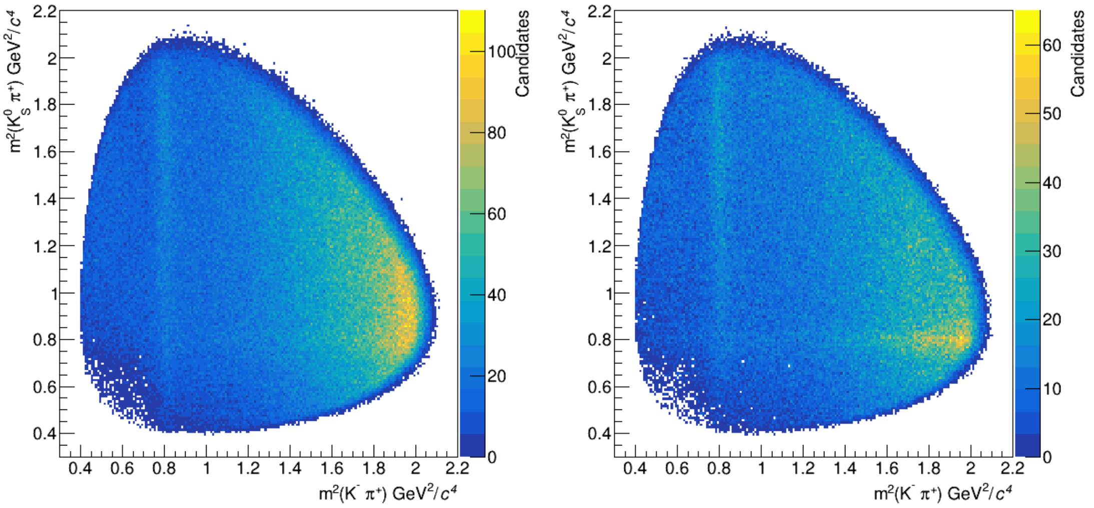


Figure 4.22: Dalitz plot of the $m(D^0)$ sideband of the RS (left) and the WS (right) samples.

4.7 Future perspectives

In this Chapter I introduced a novel analysis method to extract *CP*-violating parameters from $K^*(892)^0$ resonance with the maximum attainable resolution also in

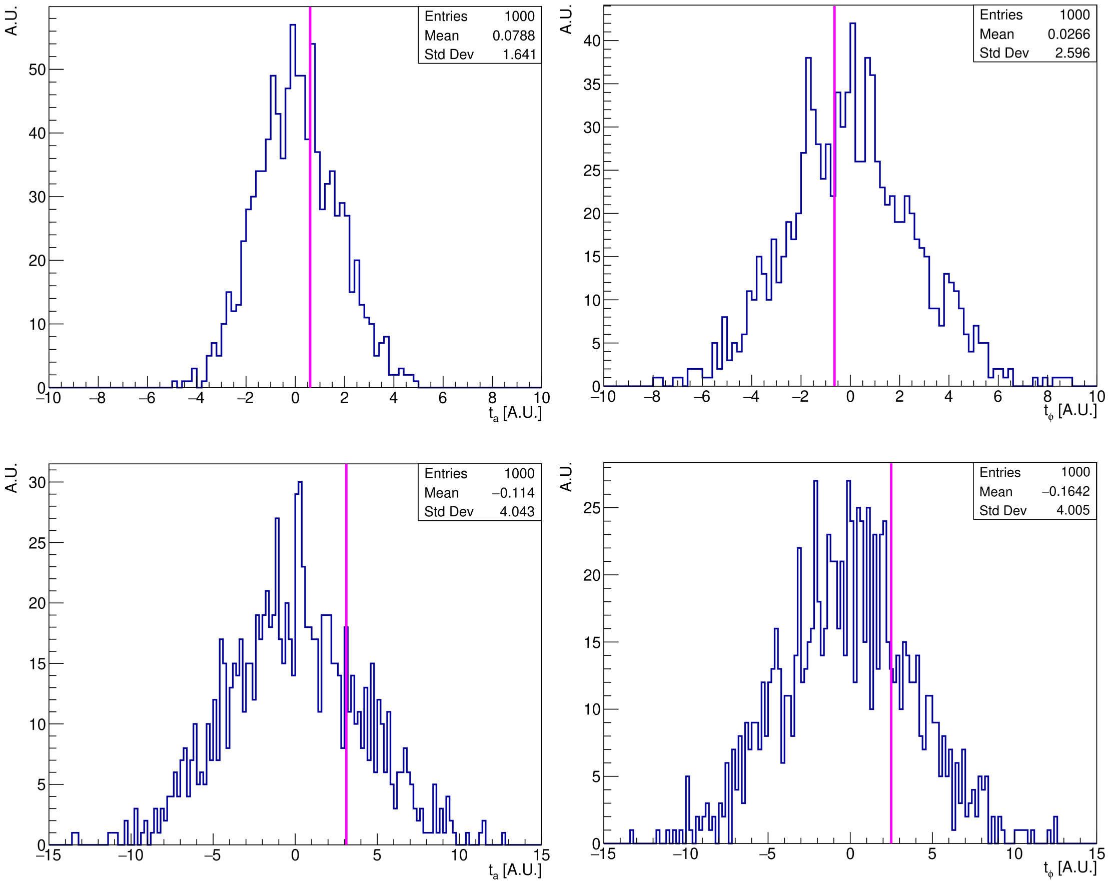


Figure 4.23: Distribution of observable t assuming no *CPV* and the value (magenta line) of the observable t measured for the sideband of the RS (up) and WS (down) samples.

the presence of significant interference effects in the $D^0 \to K^0_S \overline{K}^{*0}$ and $D^0 \to K^0_S K^{*0}$ decays. The studies on the statistical uncertainties show that the resolution scales up as expected by the increase of data sample, thus confirming that the custom measuring methodology developed for this measurement does not lose power in comparison with the full Dalitz fit. The systematic uncertainties require to be precisely studied and quantified. However the observables t show a low sensitivity to *CPV* on resonances different from $K^*(892)^0$, therefore I expect that the systematic uncertainties, mostly related to the Run 1 model uncertainties, will be lower than the statistical uncertainties. This analysis is still blind, and actually it is under a review internal to LHCb. At the end of the review the the result will be unblinded and published. This will be the most precise result available, with an effective A_{CP} resolution of $\mathcal{O}(1\%)$ (Table 4.10). However, according to current predictions [35, 36], it is unlikely that this resolution will be sufficient to observe a *CP* violation signal in this channel. For that goal, it will be necessary to collect significant more data. The only experiment likely to obtain that much additional data in the near future is LHCb itself, in its upgraded configuration.

During the last years LHCb received substantial upgrades in order to reach 50 fb^{-1}

in Run 3 and Run 4, and a further upgrade is planned to integrate $300\,\text{fb}^{-1}$ of data in Run 5. The data used in this analysis corresponds to an integrate luminosity of $5.6\ \text{fb}^{-1}$. Then, assuming to collect data with the same efficiency, the statistical uncertainty will be reduced of a factor 7. This will lead to a resolution that will definitely allow observation of the expected level of *CP* violation in this channel, and allow to check current models and improve our predictions for further *CP* asymmetries. Conversely, a non-observation would in itself imply a failure of the current picture of *CP* violation in charm and will open new scenarios. This analysis is therefore expected to be an important item in the physics program of LHCb for the future, starting already from the current upgrade, but even more in view of the Upgrade II.

However, all this rests on the said assumption that LHCb will be able to keep collection these events, with at least the current efficiency, at much higher luminosities. This is far from being granted. Reconstructing and triggering efficiently event involving the lower-momentum c-hadrons already in the environment of the Run 3 that is about to start, is an unprecedented challenge. It should not be forgotten that the design choices of the trigger for the Run 3 upgrade have been tuned on b-hadron decays, that, while also challenging, have a significantly higher p_{T} distribution, that makes them less sensitive to possible increases of the p_{T} threshold at the higher luminosities that are in front of us.

In the light of the above considerations, it should be apparent that an effort at preserving the trigger efficiency of LHCb data acquisition (DAQ) for low-momentum track is of great importance. It is also an urgent one to undertake, because it is unrealistic to think that this could be achievable without a physical upgrade of the DAQ system to make it capable of reconstructing a higher volume of data in real time, and this requires a significant lead time to allow for the design and deployment of newer computing technologies.

Therefore, I will now turn to describe the other part of my work, that makes the second section of my Thesis.

Chapter 5

Data processing at LHCb in Run 3 and beyond

5.1 The LHCb Upgrade

With the intent to collect $50\,\mathrm{fb}^{-1}$ in Run 3 and Run 4, during the Long Shutdown 2 of the LHC collider (2019 – 2022), the LHCb experiment received substantial upgrades concerning both detector and online systems [57]. All upgrades toke into account the new experimental environment, with a center-of-mass energy of $\sqrt{s} = 14\,\mathrm{TeV}$ and an important increase of luminosity, set to $\mathcal{L} = 2 \cdot 10^{33}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$.

This results in a much higher track multiplicity then Run 2, and in an average number of primary pp interactions per bunch crossing equal to $\nu = 7.6$ that will require new detectors with greater granularity to maintain a good track reconstruction performance. The higher track multiplicity and readout rate required significant changes to the tracking sub-detectors. The VELO moved from silicon strip sensor to silicon pixel, maintaining the retractile structure. The TT has been replaced by the UT. The IT and OT has been replaced by the Scintillating Fibre Tracker (SciFi).

During Run 1 and Run 2, only information from the calorimeters and the muon system are available at the full crossing rate. Then, the trigger selections at the first level were based on simple quantities as the deposit of transverse energy (E_T) or tracks with high transverse momentum (p_T). While this provides high efficiencies on dimuon events, it typically removes half of the fully hadronic signal decays. Indeed, the E_T threshold required to reduce the rate of triggered events to an acceptable level is already a substantial fraction of the B meson mass. As shown in Figure 5.1 the trigger yield therefore saturates for hadronic channels with increasing luminosity [58].

In order to trigger on information that is more discriminating than E_T, the readout rate of all sub-detector was increased to 40 MHz from the former frequency of 1.1 MHz. This allows to access data of the whole detector since the very first trigger stage. The trigger system moved to a fully software solution, with the full event reconstruction performed in real-time, allowing to trigger directly on advanced tracks parameters. This leads to important improvements in annual signal yields, but will also enormously increase the computational demands on EFF.

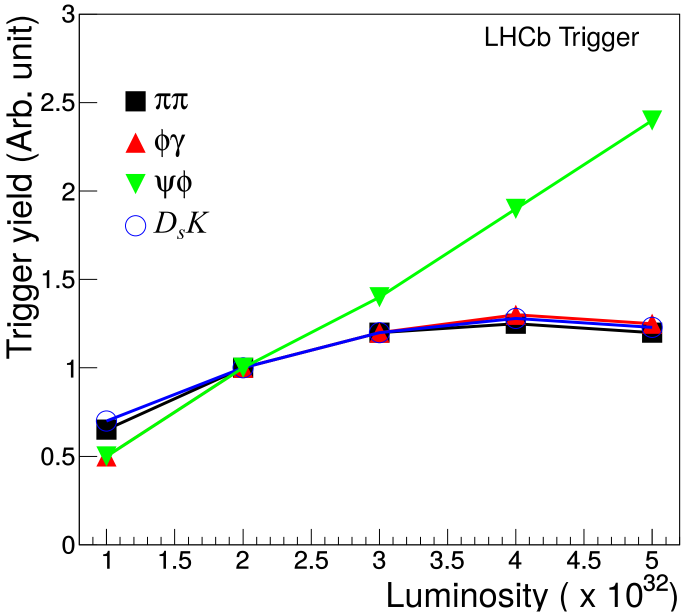


Figure 5.1: The trigger yield for different decays of B mesons. Each point is normalised to the trigger yield at nominal Run 1 luminosity ($2 \cdot 10^{32}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$). Several modes saturate before the nominal Run 3 luminosity of $2 \cdot 10^{33}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$. From Ref. [58].

5.1.1 The silicon pixel VELO

The Run 2 VELO has been replaced by a new detector based on silicon pixel technology [59]. The upgraded VELO consists of 26 tracking layers both in the forward and in the backward regions with respect to the nominal interaction point, as shown in Figure 5.2 (left). Each layer is made of two modules, one on the left hand side of the detector, the other on the right hand side, with the ability of distancing them from the beam axis such as for the former VELO detector. A module contains channels for coolant flow, in addition to readout and control chips. Four silicon sensor tiles are installed on each module, two on the front side, and two on the back side (right hand side of Figure 5.2). The active area of each sensor is $42.46 \times 14.08\,\mathrm{mm}^2$.

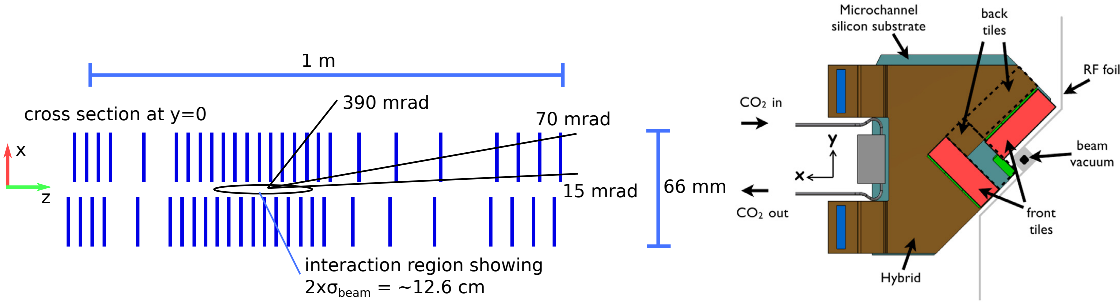


Figure 5.2: Layout of the upgraded VELO. Top view (left). Details of a module (right).

The pixel sizes are $55 \times 55\,\mu\mathrm{m}^2$ and the entire VELO detector includes about 41M pixels. The raw hit resolution varies from 9 µm to 15 µm, depending on the angle of the particle. The inner radius of sensitive area from beam axis is reduced from current $r = 8.2\,\text{mm}$ to less of $r = 5.1\,\text{mm}$, to improve impact parameter resolution. The single hit resolution is expected to be about $12 - 15\,\mu\text{m}$ for both x and y coordinates.

5.1.2 Upstream Tracker

The Upstream Tracker (UT) is the replacement of the TT [60]. It is located upstream the dipole magnet, centred around $Z = 2485\,\text{mm}$. The new detector consists of four planes of silicon micro-strips arranged in a x-u-v-x configuration as the TT. The planes are constructed with vertical strips, called staves. Each stave is the width of a full silicon sensor, approximately 10 cm. The sensors and readout chips are mounted on custom hybrids which in turn are mounted on thermo-mechanical support structures. The staves are about 1.6 m long and mounted vertically. The signals from the sensors are taken out to the top and bottom.

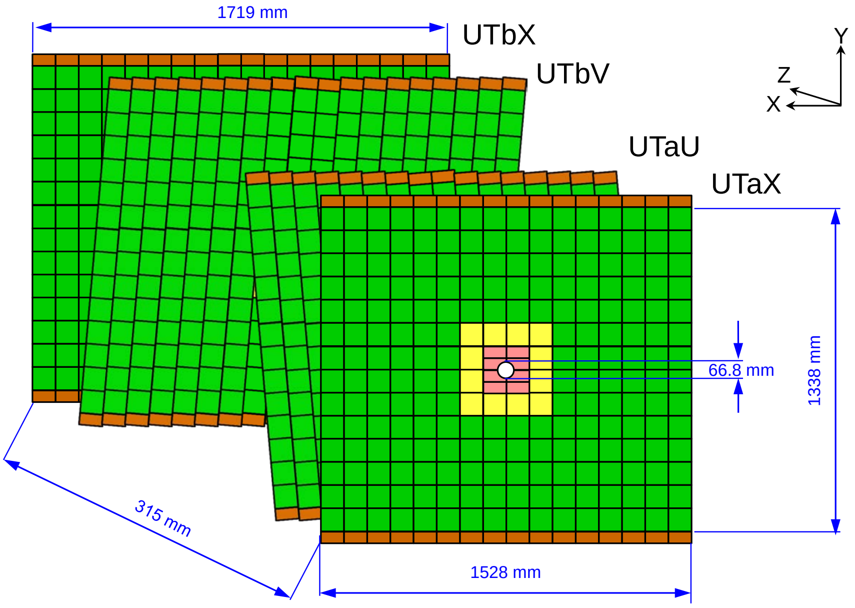


Figure 5.3: Layout of the UT. Sensors with different pitches and lengths.

In contrast to the TT, UT planes use thinner sensors (250 µm *vs.* 500 µm) with finer segmentation in the central region (95 µm *vs.* 183 µm), and provide a larger acceptance coverage. Pitches and lengths of sensors vary depending on their position. Around the beam pipe, sensors with 95 µm pitch and 5 cm long are used, while in central areas we have sensors with 95 µm pitch and 10 cm long. Finally, more externally sensors with 190 µm pitch and 10 cm long are used. Figure 5.3 shows the UT layout and highlights the three types of sensors with different colours. Angular coverage of UT detector is of 314(248) mrad in the bending (non bending) plane.

5.1.3 Scintillating Fibre Tracker

The Scintillating Fibre Tracker (SciFi) replaces both the IT and the OT [60]. It consists of three stations placed after the dipole magnet at the same nominal positions of the OT stations (Fig. 5.4). Each station includes 4 tracking layers arranged in a x-u-v-x configuration. Each layer is made of 12 modules 5 m high and 0.52 m wide. There is a 3 mm gap between modules; the inefficiency due to geometrical gaps and single dead channels is expected to be 1%. The two central modules have cut-outs to allow the beam-pipe to pass through the detector, and they contain six fibres layers. The remaining modules have five fibres layers because of lower radiation exposure. The fibres will be read out by Silicon Photomultipliers (SiPMs) placed at the top and bottom of the detector.

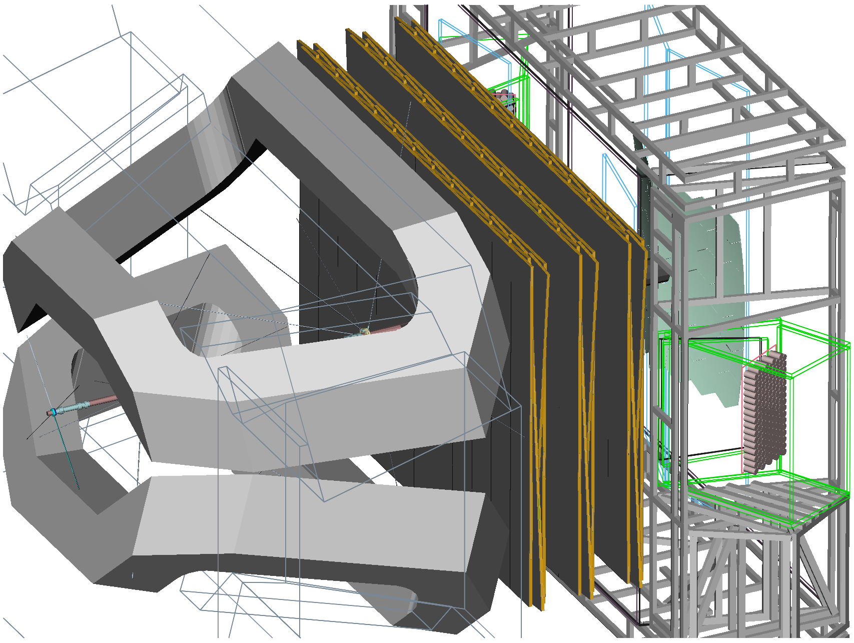


Figure 5.4: 3D model of the SciFi represented between the dipole magnet on the left and RICH2 on the right.

Scintillating fibres are long 2.5 m and have circular cross-section of 250 μm in diameter. A fibre consists of a polymer core, with the addition of an organic fluorescent dye for about $\sim 1\%$ of the fibre weight. Photons are produced by excitation of the polymer core, and are propagated through the fibre by total internal reflection to the SiPM. The decay time of the scintillation light is ≈ 3 ns; the propagation time of light along the fibre is 6 ns/ m. The simulated hit detection efficiency at the end of the lifetime of the detector is above 97.4%.

5.2 The LHCb Upgrade DAQ and trigger system

With the goal of increase the trigger efficiency of hadronic channels, limited by the E_T and p_T low discriminating power, the entire data acquisition and trigger system was redesigned to collect and reconstruct event at full LHC bunch crossing rate [61]. With the inclusion of information from the tracking sub-detectors, the trigger system can now implement selections based on precise measurement of the momentum and of the impact parameter of the reconstructed tracks. As shown in Figure 5.5, the trigger system moved to a two stage full software solution. Due to the high computing power required to reconstruct event at the average rate of 30 MHz, HLT1 is implemented on GPUs [62].

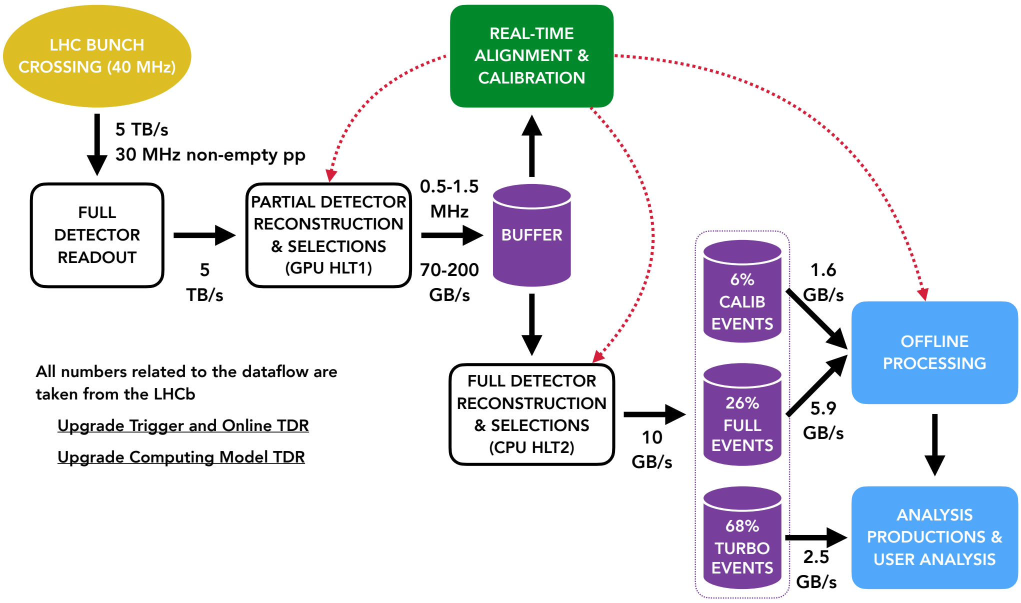

LHC BUNCH CROSSING (40 MHz)
5 TB/s
30 MHz non-empty pp
FULL DETECTOR READOUT
5 TB/s
PARTIAL DETECTOR RECONSTRUCTION & SELECTIONS (GPU HLT1)
0.5-1.5 MHz
70-200 GB/s
BUFFER
REAL-TIME ALIGNMENT & CALIBRATION
FULL DETECTOR RECONSTRUCTION & SELECTIONS (CPU HLT2)
10 GB/s
6% CALIB EVENTS
26% FULL EVENTS
68% TURBO EVENTS
1.6 GB/s
5.9 GB/s
2.5 GB/s
OFFLINE PROCESSING
ANALYSIS PRODUCTIONS & USER ANALYSIS
All numbers related to the dataflow are taken from the LHCb
Upgrade Trigger and Online TDR
Upgrade Computing Model TDR

Figure 5.5: LHCb Upgrade dataflow.

In order to collect data from all sub-detector channels, all the data fragments of a single event are collected and assembled in the same place, resulting in the event building process. Therefore all the different event fragments must be sent over some interconnection network in an all-to-one communication [63]. A dedicated server farm called Event-Builder (EB) perform this process. The EB has 173 nodes connected to the detector front-end, to the other nodes of the EB, and to the HLT2 farm.

Figure 5.6 shows the devices mounted on a EB node. ∼ 10 k optical fibres connect the detector front-end to the Readout Boards. The Readout Boards, also referred to as TELL40, are custom built PCIe board equipped with an Intel Arria 10 FPGA, one of the largest FPGA available during board development. This chip has a PCIe Gen3 interface that allows to write data to the host memory via Direct Memory Access (DMA) at a rate of up to 100 Gbit/s [64]. A logical units, the Readout Unit (RU), sends event fragments to others EB nodes through the EB network. The

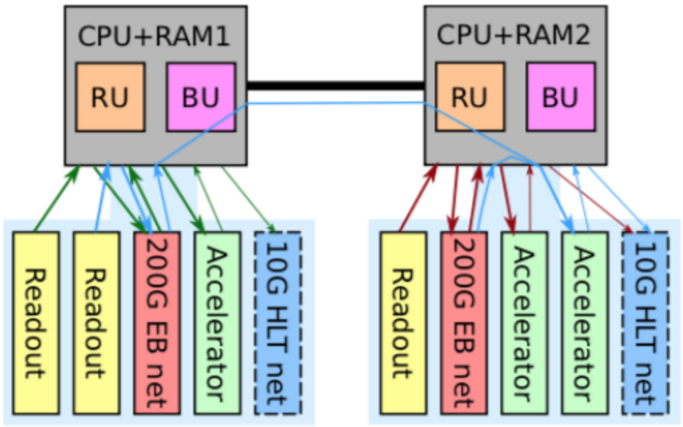


Figure 5.6: Devices installed on a EB node.

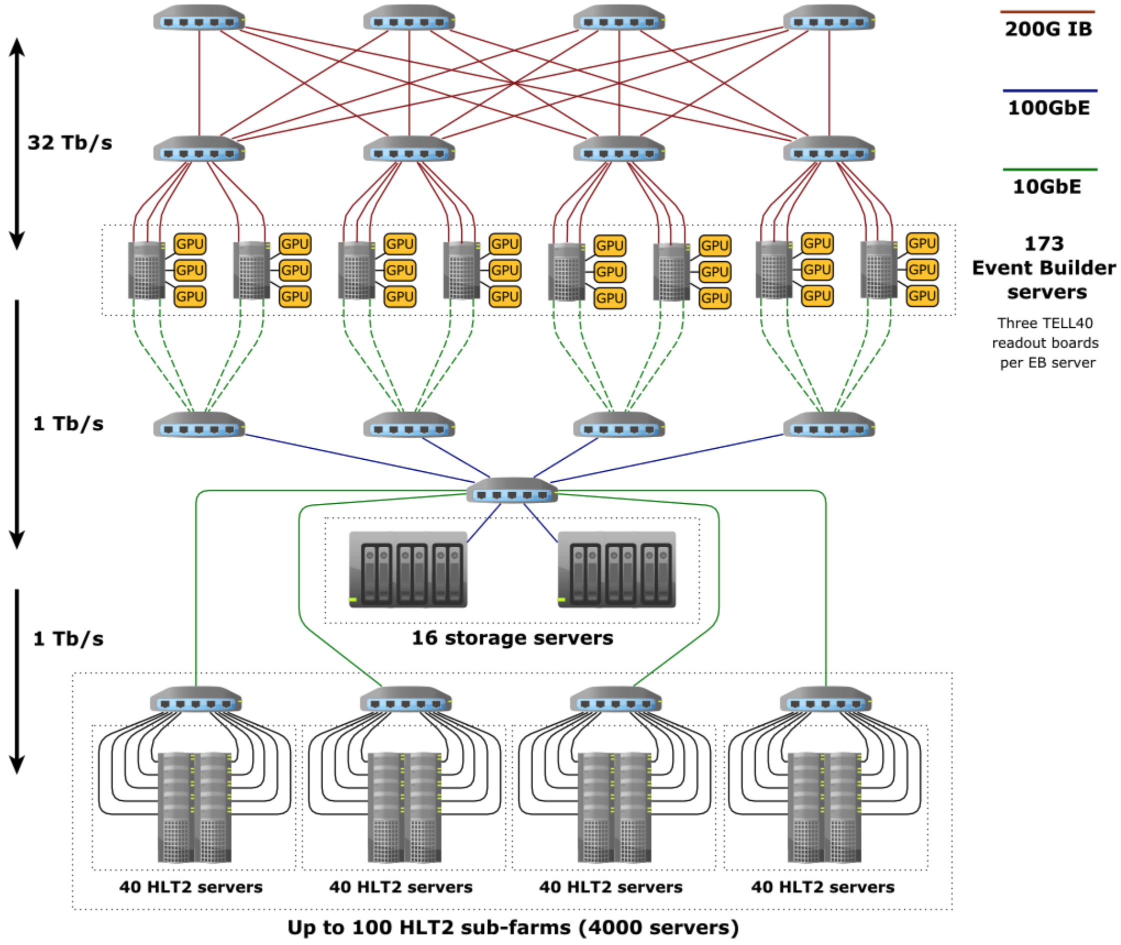


Figure 5.7: LHCb Upgrade DAQ networks.

Builder Unit (BU) collects fragment of the same event building the event. Each EB node can host up to three accelerator boards for data processing. HLT1 runs on GPUs mounted on these slots. Event selected by HLT1 are sent to the EFF through the HLT network.

Reducing the data rate by a factor of $30 - 60$ and beeing installed directly in the EB, HLT1 allows to have a much smaller and cheaper network between the EB and the EFF [65]. The data bandwidth of raw data is 40 Tbit/s, and it require high bandwidth network interface cards (NICs): the EB network relies on 200 Gbit/s InifiniBand NICs. After HLT1, data can be sent to the EFF through 10 Gbit/s Ethernet NICs, interfaces installed by default on servers motherboards. Figure 5.7 shows the topology of the networks.

The EFF is dedicated to run HLT2. The EFF consist of a mixture of servers of different generations and with different numbers of physical cores, due to the asynchronous nature of HLT2 processing, load-balancing between these servers is an implementation detail. Quantifying the computing resources available in terms of a Run 2 EFF machines, the farm has 1450 equivalent servers, with a further 1200 equivalent servers that will be bought.

5.3 Challenges for future Runs

Under the High-Luminosity Large Hadron Collider (HL-LHC) project, during Long Shutdown 3, the LHC will go under several upgrade in order to increase luminosity by a factor of 10 beyond the LHC's design value. LHCb Upgrade will continue data taking during the following period (Run 4), after which it will become too time-consuming to accumulate significantly larger samples, and when the radiation dose will have reached the design values for several critical sub-systems. However there are strong arguments to continue flavour physics studies at the LHC with a dedicated experiment [28]. An Expression Of Interest proposing Upgrade II was submitted in February 2017 [29].

LHCb will be upgraded in order to collect events at the luminosity of $\mathcal{L} = 1.5 \cdot 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$, and integrate $300\,\mathrm{fb}^{-1}$.

The data bandwidth and computing power required by LHCb Upgrade II will be at least 10 times greater that the actual ones. With the the slowing down of Moore's law, LHCb collaboration is looking at heterogeneous computing solutions as a way to manage the increasing data flows and complexity. LHCb Upgrade is at the frontier of these developments having adopted the GPU-based solution for the first stage of trigger. However, further computing enhancement is needed for Upgrade II. The LHCb Real Time Analysis project (RTA), that develops and maintains the real-time processing of LHCb's data, has established the RTA-accelerators Work Package (called WP6). The WP6 includes all the R&D activities related to high performance computing accelerator platforms for the future upgrade. It involves projects aiming to run real-time analysis on divers computing platforms.

One device under development is a highly-parallelized custom tracking processor based on the “Artificial Retina” architecture. The “Artificial Retina” architecture takes advantage of FPGA parallel computational capabilities, by distributing the processing of each event over an array of FPGA cards, interconnected by a high-bandwidth ($\sim 15\,\mathrm{Tbit/s}$) optical network. This is expected to allow operation in real-time at the full LHC collision rate, with no need for time-multiplexing or extra buffering due to its low latency ($< 1\,\mu\mathrm{s}$). A system with this level of performance can be integrated into the DAQ chain of the experiment. Operating in a transparent way during data readout, it provides tracks to the trigger system as a virtual sub-detector, reducing the HLT computational load to a manageable level.

In the next Chapters I will describe in details the “Artificial Retina” architecture and the work that I performed in order to implement this device.

Chapter 6

Real-time data processing with FPGAs

6.1 The Field Programmable Gate Array

Today's complex digital systems are not implemented on interconnected integrated circuits, since the high number of components leads to large, expensive, low efficiency, and unreliable devices. Often, these system are implement on custom integrated circuit for a specific application, called Application Specific Integrated Circuit (ASIC). When high flexibility is required, in development phase but also during operating period, the use of field-programmable devices like Field Programmable Gate Arrays (FPGAs) allows to greatly reduce cost and development time. This is particularly true for powerful electronic systems that need to be produced only in limited quantities, where mass production savings can not be achieved - examples are radars, medical CT scanners, advanced navigation and communication systems. The issues for advanced research equipment are clearly similar.

FPGAs contain an array of programmable logic blocks, and a hierarchy of interconnections, that can be configured and "wired together" according to the firmware downloaded onto the devices. The digital-logic function are described in a hardware description language (HDL) like VHDL and Verilog. Then the actual firmware is generated by compiling the design for a specific target device. The drawbacks of these programmable devices are a somewhat lower speed compared to ASIC, and a higher cost when employed in large productions (which is not our case). Some functions, like memory blocks, Digital Signal Processors (DSPs), Serializer/Deserializer (SerDes) for high speed communications, and PCIe interfaces are required by most design and are speed critical. Many FPGAs implement them as Hard Intellectual Property (HIP) blocks, located in designated areas of the chip. These blocks are optimised to perform predefined tasks, with a limited amount of flexibility: for instance, a memory block only works as a RAM, but the user can choose some parameters like the words width and depth.

Even with the basic concepts being the same, different manufacturers use different names and organisations for the internal components of the FPGAs. In this thesis

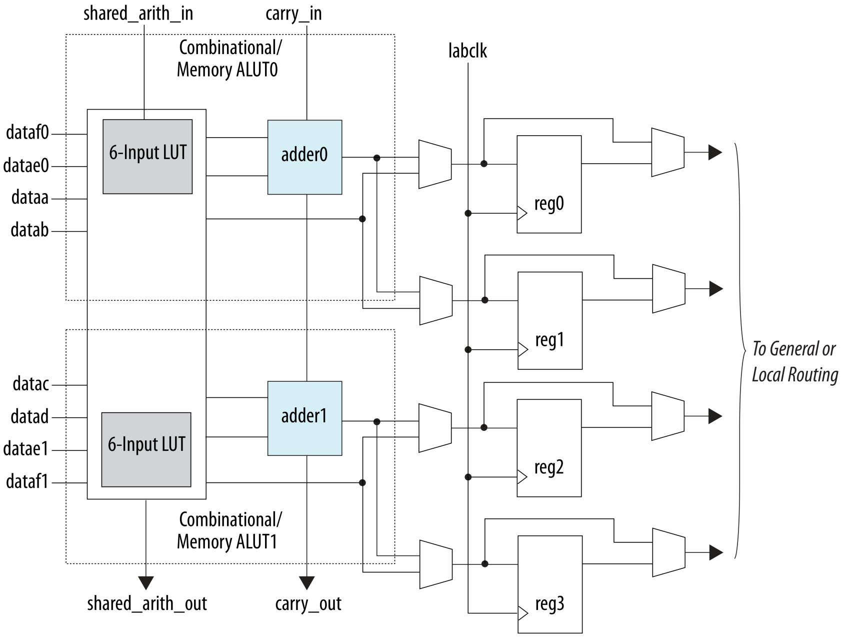


Figure 6.1: ALM High-Level Block Diagram for Intel's Arria 10 Devices.

work, I will be using FPGAs, and the relative language, from the Intel (previously Altera) manufacturer that are the most commonly used in the LHCb experiment.

In Intel's language, the programmable logic blocks inside FPGAs are called Adaptive Logic Modules (ALMs). These modules can be configured to implement logic functions, arithmetic functions, and register functions. Figure 6.1 shows the ALM of a Arria 10 FPGA. An ALM contains Lookup Tables (LUTs), full adders, D-type flip-flops, and multiplexers. Wiring these components, an ALM can perform Boolean algebra with only combinational logic, *i.e.* that the output is a pure function of the present input only with an asynchronous circuit, and synchronising the output to a clock storing it in the flip-flops. The smallest FPGA device used in the present work contains $\sim 170\,\mathrm{k}$ ALMs, the biggest $\sim 930\,\mathrm{k}$.

Giving appropriate consideration to the peculiarities of a FPGA when producing a design allows to achieve optimal performances. All ALMs are capable of working in parallel, thus a single device with enough logic block can carry out many different tasks at the same time, or implement multiple instances of the same logic function (entity). In this way is also possible to divide a complex task into a series of sequential steps performed by different entities in parallel. By an appropriate balancing of the processing time, each entity can be made to process data continuously. Since data flows along the entity chain, this is called 'pipeline architecture'. A pipeline allow to increase the amount of data processed in a time frame (throughput). If the processing time of an entity is not deterministic, a common solution to avoid data overrun is implement a back-pressure mechanism. When a entity is not ready to receive new data, it raises a 'hold' signal, pausing the output of the previous entity. If the previous entity, having completed its task becomes unable to accept further inputs, it will send a hold signal to its own inputs, causing the 'hold' to propagate back along the processing chain. Addition of memory buffers between entities allows to avoid slowing down the process, by making input data available for every entity at all times.

6.2 The "Artificial Retina"

Proposed in 2000, the "Artificial Retina" is a fast parallel track reconstruction architecture, conceived for application to High Energy Physics (HEP) experiments [66], inspired by the neural mechanisms of receptive fields, used in mammalian visual pathways to recognise lines and edges [67, 68]. As visual brain areas contain neurons, each combining several inputs to be sensitive to specific properties of an image (such as shape, orientation, colours), the "Artificial Retina" implement a set of cells, each sensitive to hits belonging to a reference track. As the response of the neurons is not binary, the response of a cell will be larger in proportion to how close the hits are to the reference track. Tracks corresponding to the cells with a higher activation level will be the track that were most likely present in the event. Important features suggested by the natural neural system, in addition to the continuous response, are the fully data-flow organisation, and the high degree of parallelization. These features are crucial to lowering the latency between the visual stimulus and the action to exploit the available distributed computing power to the fullest extent. This brings together a further feature known to exist in the natural vision: the propagation of the stimuli with an overall increase of the bandwidth. In traditional reconstruction systems the bandwidth is progressively reduced during processing, while in the "Artificial Retina" the bandwidth needs to increase significantly at some point, because multiple copies of the same data are allowed to be produced and reach different cells, shrinking down only at a later stage where few tracks are reconstructed from many hits. The FPGA, with its large internal bandwidth and high degree of parallelisation, is the natural device where to implement a system mimicking all these features.

6.2.1 Mathematical aspects

To explain the mathematical aspects of the "Artificial Retina", in this section, I will consider straight tracks traversing an array of n parallel detector layers. The same principles can be applied to a more complex environment. In this sample, if we consider only the transverse view, a track is defined by two parameters (u, v), like the coordinates of intersection of the track with the first and the last layer of the detector. The track parameters space is divided into a grid of cells. The centre of the cell has coordinates (u_i, v_j), that correspond to the parameters of the reference track. The pair (u_i, v_j) define also a set $t_l(u_i, v_j)$ of intersection between the reference track and the detector layers, where l is the layer number. $t_l(u_i, v_j)$ is called receptor for layer l of cell (i, j). For each event, the algorithm computes the activation level:

$$R_{ij} = \sum_{l=1}^{n} \sum_{x_l \in \mathcal{H}_l} \exp\left(\frac{-d(x_l, t_l(u_i, v_j))^2}{2\sigma^2}\right) \tag{6.1}$$

where $\mathcal{H}_l$ is the set of all hits recorded on layer l, $d(x_l, t_l(u_i, v_j)$ is the Euclidean distance between the hit x_l and the receptor $t_l(u_i, v_j)$, and σ is a parameter adjusted to optimise the sharpness of the response of the receptors.

In Eq. 6.1 the distance of each hit from the receptor is weighted with a Gaussian function. Thus the cell returns a response that continuously varies depending on the "distance" of the track from the reference one imitating the continuous neuron response to exciting stimuli. This is a key feature that distinguishes the "Artificial Retina" architecture from previous real-time tracking systems based on patterns stored in databases. Others system like Associative Memories-based systems provide a binary response ("yes" or "no") from the comparison with stored patterns. Therefore to reconstruct tracks with good resolution are necessary a high number of patterns or a successive fitting stage that resolves all the combinations of hits within the pattern [69, 70]. The "Artificial Retina" identifies tracks as local maxima in the cells space, via a local cluster-finding algorithm. Then it interpolates the activation level of the neighbour cells. The interpolation allows to reconstruct tracks at the native resolution, while keeping the cell granularity reasonably small.

Figure 6.2 summarise all the steps performed by the "Artificial Retina" for the track reconstruction.

6.2.2 Architecture

The mathematical aspects of the "Artificial Retina" are similar to the "Hough transform" [71, 72], a method invented for machine analysis of bubble chamber photographs, and now used in computer vision. However the distinctive element of the "Artificial Retina" is the implementation architecture, that allows to reconstruct tracks with throughput and latency performances never attained before.

The "Artificial Retina" architecture has two main components: the Engines and the Distribution Network. The Engines implement the weight and sum mechanism already described. Each Engine corresponds to a cell of the track parameter space, and all Engines work in a full parallel way. To overcome FPGA size limitations without increasing latency, cells can be spread over several chips that work in parallel on the same event. This require a system that allows to delivery hits from the same event to all the chips of the "Artificial Retina". Additionally hits provided by a readout units usually belong to a limited region of the sub-detector. Therefore the system needs also to collects hits from multiple sources. The Distribution Network fulfils this task. This component implements an intelligent delivery system (Switch), with embedded information that allows to deliver to each Engine only hits close to its receptors. In this way, each Engine processes a smaller number of hits, allowing the system to reach higher throughput. The Switch is modular and can be spread

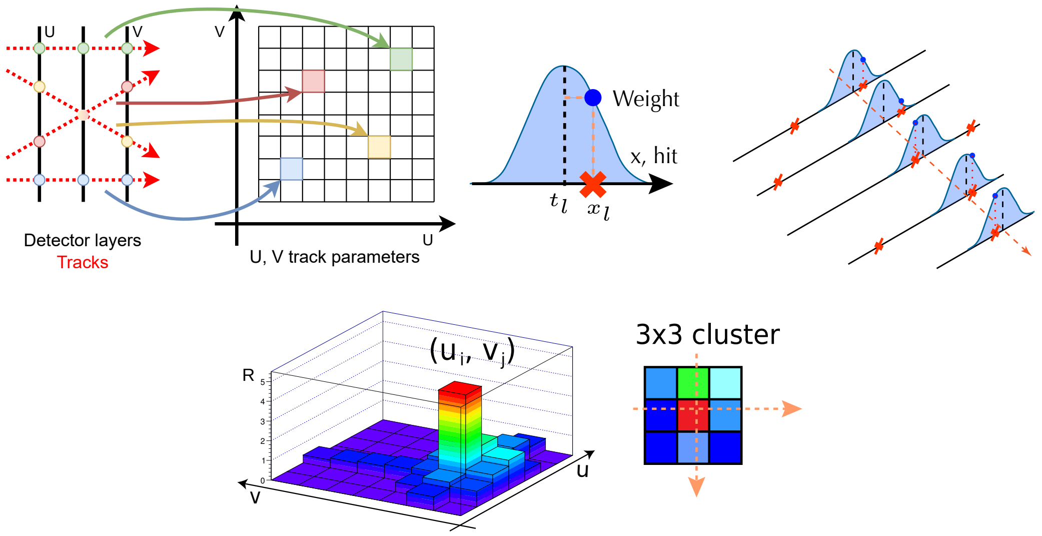


Figure 6.2: Processing steps performed by the "Artificial Retina" for the track reconstruction. From left to right, reference tracks and their interceptions with the detector layers are mapped into a matrix of cell. The cells calculate and sum up the Gaussian weight of the hits x_l. The weight of hits far away from the receptor is null. Tracks are reconstructed finding local maxima in the cells space. Tracks parameters are calculated interpolating the activation level of the 3×3 clusters around the maxima.

over multiple interconnected chips. Modern FPGAs have numerous high-bandwidth transceivers (XCVRs), that can be used to implement optical serial links between boards.

The Distribution Network and Engines are implementable within the same array of FPGAs, in separate and independent locations of the chip. The Tracking Boards, hosting the "Artificial Retina" FPGAs, might be paired to the Readout Boards of desired sub-detector, reading the hits. The Distribution Network routes hits towards appropriate Engines alternating switching and optical communication stages. Then Engines perform the necessary calculations to achieve track recognition.

The Switch

The Switch interconnects all the inputs to the system (the Readout Boards) to each Engines. It is built from a network of nodes. The modular structure allows to scale easily, to store the information only at the nodes where it is required, and to distribute the Switch over multiple devices. The Dispatcher is the basic block. It has two inputs and two outputs, with a LUT that store the routing rules. The Dispatcher delivers hits to any output (even both) according to the routing rules. Combining a sufficient number of Dispatchers it is possible to build a Switch with the desired number of inputs and outputs. To implement a Switch with $N = 2^n$

inputs/outputs, we need M Dispatchers connected together, where

$$\begin{cases} M(0) = 0 \\ M(n) = 2M(n-1) + 2^{n-1} \end{cases} .$$

The basic components of the Dispatcher are the Splitter and the Merger. Figure 6.3 shows the interconnections between these components and how connect Dispatcher to build a Switch with multiples inputs/outputs. The Splitter has one input and two outputs. It searches the hit coordinate inside its routing LUT, and it sends the hit to one or both outputs. The information stored in the LUT are computed offline together with receptor. In a Merger, the data coming from two lines are gather together. This does not correspond to a reduction of the maximum bandwidth allowed, since the Switch is designed to have at least the same number of data lines of the input. However the Split can copy the hits, increasing the data bandwidth. Therefore the Switch, and in particular the Merger, can be a bottleneck for the system. To avoid this, these components, and more in general all "Artificial Retina" entities, must be carefully designed, avoiding as much idle cycles as possible. During my thesis work I wrote a design of the Splitter and the Merger that can elaborate a hit every clock cycles. I will describe these designs and their performances in Section 7.3.

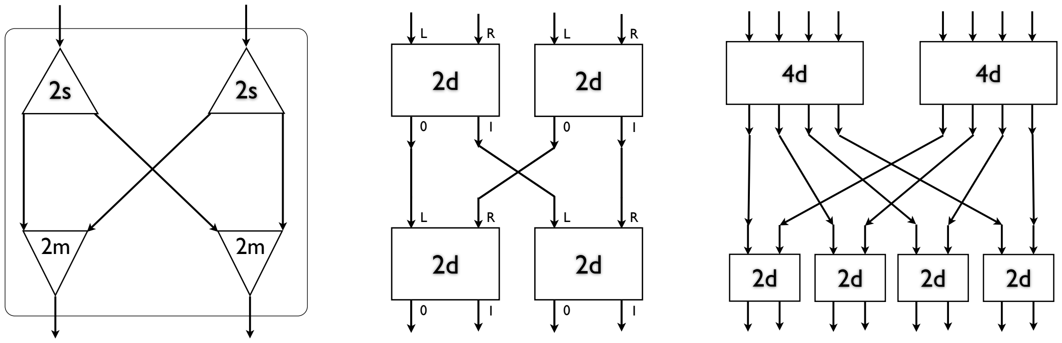


Figure 6.3: Interconnection between Splitter (2s) and Merger (2m) to build a Dispatcher (left). Interconnection between dispatchers (2d) to build a Switch with 4 input/output (center). Interconnection between 2 sub-Switches with 4 inputs/outputs (4d) and 4 dispatchers to build a Switch with 8 inputs/outputs (right). Repeating the scheme it is possible to build a Switch with the desired number of inputs and outputs.

The optical communication

The modular design of the Switch allows to segment it and implement different sections on different interconnected FPGAs. Reading data from a distributed system like the DAQ of an experiment, a custom board that concentrates all the chips and interconnections is not feasible nor desirable. Furthermore, a similar board would not be flexible, and producing a small number of custom boards is uneconomical.

The development of high-bandwidth communication channels is pushed forward by the huge recent growth of the telecommunication sector. Modern FPGAs have numerous XCVRs, that can be used to implement fast serial communication between boards. Optical serial links allow to exchange data with great flexibility and large bandwidths, connecting also distant boards. The commercial Stratix 10 board used in this thesis carries 16 XCVRs with a bandwidth of 26 Gbit/s each. This results in a total external communication bandwidth that is 4 times larger than the bandwidth of the PCIe connection used for reading input data. This factor should be enough to accommodate the mentioned bandwidth expansion required to implement the retina system.

To demonstrate that the envisioned Distributions Network is actually feasible, I have implemented and tested a prototype network connecting a certain number of boards in a full-mesh topology. In this test, smaller, cheaper boards are used with respect to those needed for the final system, to maximise the size and number of connections of the test network with a limited expenditure.

The Engine and Max Finder

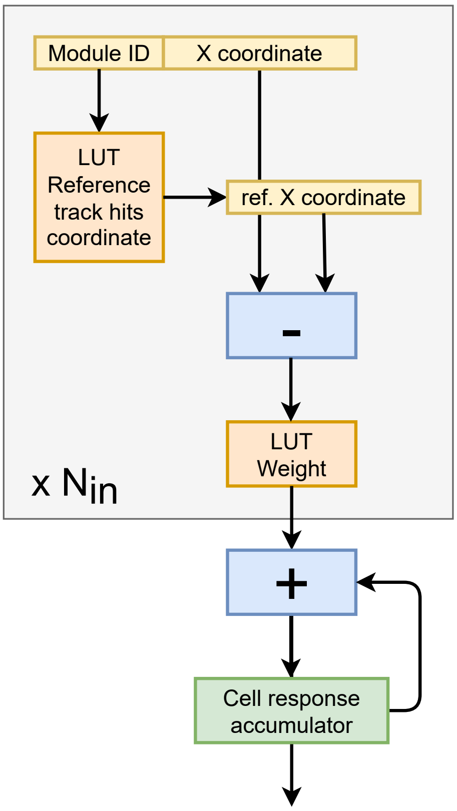

Module ID
X coordinate
LUT Reference track hits coordinate
ref. X coordinate
-
LUT Weight
x N_{in}
+
Cell response accumulator

Figure 6.4: Structure of the Engine. Block inside the grey box are replicated to match the number of Engine inputs.

The Engine is the entity that computes the activation level of the track parameters space cell according to Eq. 6.1. Figure 6.4 shows the internal structure of a Engine. It gets as input the hit coordinate and the ID of the detector module (or layer). A LUT stores the receptor coordinate for each module, allowing to calculate the distance between hit and receptor. Calculating the Gaussian weight on the fly require complex operations, so it is computed offline and stored in a second LUT. The weight of hits farther than 2σ from the receptor are set to zero. This is an optimisation parameter of the "Artificial Retina" for reducing Engines input bandwidth. The weights are accumulated in a register. Normally an Engine can process a hit per clock cycle, however, we can implement multiple instances of the input chain allowing to process an equal number of hits per clock cycle and increase the Engine throughput.

This design allow to implement the "Artificial Retina" for axial detector with unidimensional hits. During this work I wrote an Engine for 2D detector.

A special word called End Event (EE) separates the hits of a event from the hits of a different one. When the EE is delivered to the Engine, the accumulation process is complete and the search for local maxima begins. To each Engine is paired a Max Finder. It reads the activation level of the Engine together with the ones of the neighbouring Engines. Then it verifies if the central activation level is a maximum. Given the excitation level R_{kl} of the Engines, it computes the track parameters $\overline{u}$ and $\overline{v}$ as:

$$\overline{u} = u_0 + \delta u \frac{\sum_{kl} k R_{kl}}{\sum_{kl} R_{kl}}$$

$$\overline{v} = v_0 + \delta v \frac{\sum_{kl} l R_{kl}}{\sum_{kl} R_{kl}}$$

with $k = i-1, i, i+1$ and $l = j-1, j, j+1$, where u_0 and v_0 are the track parameters of the cells grid origin, δu and δv are the pitch of the cells grid, i and j are the index of the local maximum Engine. A threshold level on the activation level is applied to suppress false positive maxima.

6.3 State of the art

In 2015 part of LHCb Group in Pisa started the "RETINA Project". This is a 3-years initiative supported by INFN-CSN5 and devoted to R&D for a track processor based on "Artificial Retina" architecture. Within this project, concluded before the start of this thesis work, I developed a prototype for a 6-layers axial detector [73–75], taking the LHCb IT and SciFi as reference. Figure 6.5 shows the design of the RETINA prototype with Switches and Engines distributed on two FPGAs interconnected through optical serial lines. This design represents a system that takes inputs from three different sources, and performs tracking on four independent devices. It demonstrated that the logic functionality of the architecture worked.

The RETINA prototype was implemented on a board with 2 Stratix V FPGAs. The clock frequencies were 400 MHz for the Distribution Network and 280 MHz for the Engines. The system event processing rate is proportional to the tracker occupancy defined as the number of real tracks divided by the number of cells. Figure 6.6

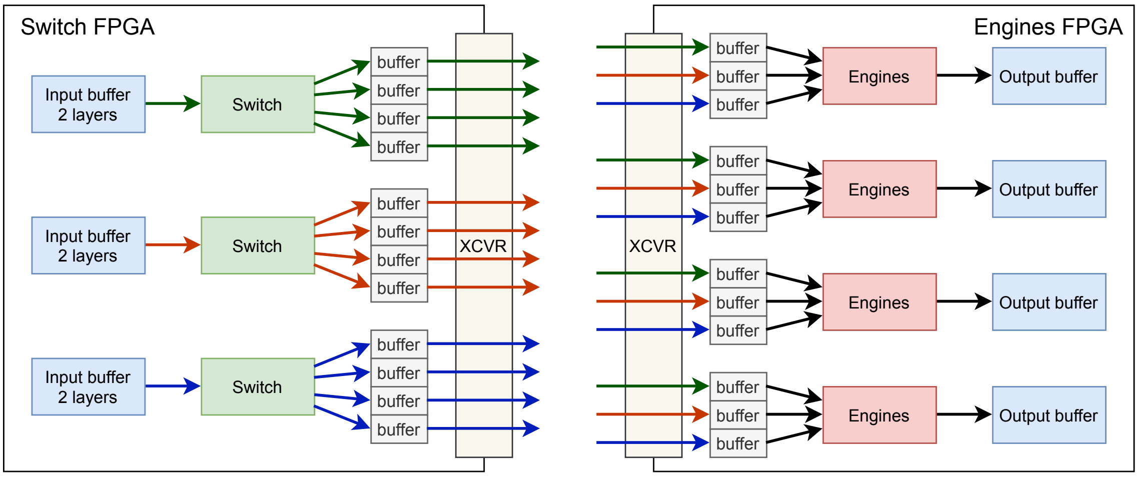


Figure 6.5: Design of the RETINA prototype with Switch and Engines distributed on two FPGAs interconnected through optical serial lines. Lines are rearranged in order to send to each Engines block data from all sources.

shows the event rate of the RETINA prototype as a function of the occupancy. This prototype demonstrated that the event rate of 30 MHz was achievable when the occupancy is lower than ∼ 0.5%. As shown in Figure 6.7 the system latency is ∼ 400 ns, with the largest contribution coming from the optical interconnection.

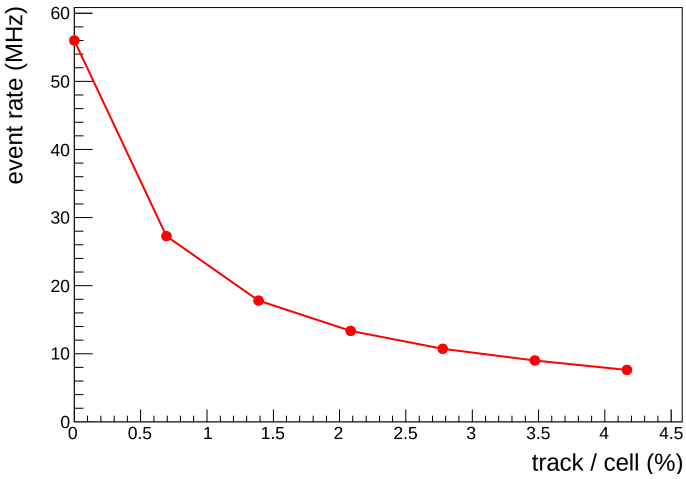


Figure 6.6: Event rate of the RETINA prototype as a function of the occupancy of the system.

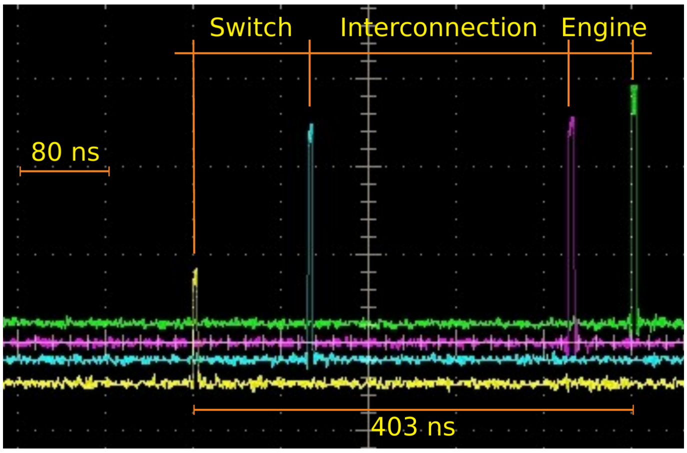


Figure 6.7: Latency measurement for the RETINA prototype using the optical fibres interconnection.

Chapter 7

“Artificial Retina” implementation

The RETINA prototype was an important milestone towards a complete tracking system based on the “Artificial Retina” architecture. Even if it demonstrated that the logic functionality of the architecture worked, the different component of the “Artificial Retina” needs to be implemented in their final version, in order to integrate the system in the reconstruction chain of a real detector like LHCb.

7.1 System integration in LHCb DAQ

The “Artificial Retina” system, by nature, works on unbuilt data, *i.e.* previous that the data of the same event recorded by different sub-detectors is packetised. This approach has different advantages. Before the event building data of a specific sub-detector is located on a small number of nodes, and distributed to a large computing farm after the building. Working on unbuilt data means that just the number of cards needed for the detector of interest are needed. When dealing with many small packet (a single LHCb event require less than 200 kB), the read from disk operations and the event unpacking requires a significant fraction of the processing time. Working on data stream instead of packet, the “Artificial Retina” is not affected by this effects. However, the integration of a device in the readout chain requires specific solutions related to the DAQ system architecture. The most promising way to integrate the “Artificial Retina” system in LHCb DAQ chain is placing the FPGA boards inside the EB nodes. As explained in Section 5.2, each EB node hosts three Readout Boards connected to the detector front-end, the NICs for the EB and EFF networks, and up to three accelerator boards, where only one is used for the HLT1 GPU. The plan is to mount the Tracking Boards, hosting the “Artificial Retina” FPGA, in the unused server slots.

Figure 7.1 shows the data flux from the Readout Board towards the EFF network for a standard EB node (left) and a node with the “Artificial Retina” Tracking Board (right). For clarity, the Figure shows only one card per kind, the flux is the same also with multiple boards. The Readout Board write data of a portion of the detector on a dedicated location inside the host RAM. In this location the RAM stores information of all the events. A process exchanges unbuilt data thought the

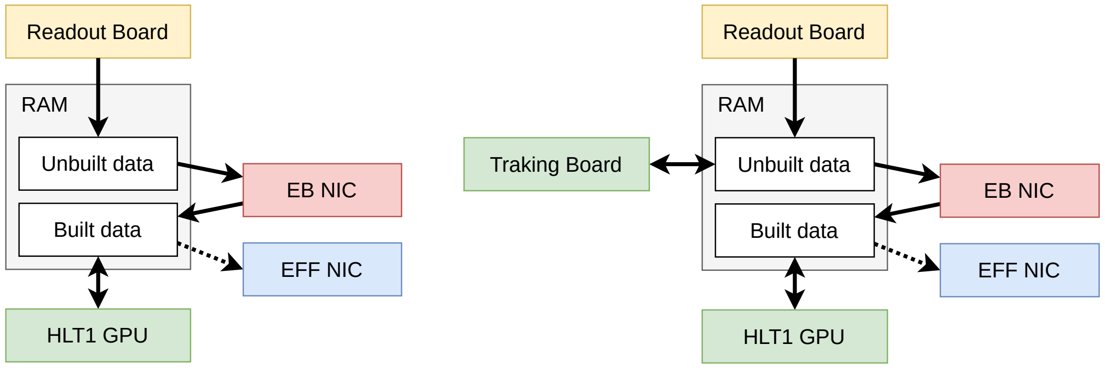


Figure 7.1: Data flux from the Readout Board towards the EFF network for a standard EB node (left) and a node with the “Artificial Retina” Tracking Board (right). The dotted lines indicate that not all the events are sent to the EFF network.

EB network, then it builds the events and writes them on a different location of the host RAM. A single EB node store all the information recorder for a subset of events. HLT1, executed on GPU, reconstructs the events and flags events that satisfy trigger selections. Flagged events are sent to the EFF network to be processed by HLT2. The Tracking Board take place before the data exchange thought the EB network. The board driver sends the unbuilt data to the Tracking Board and writes in the unbuilt RAM location information about the reconstructed tracks. Then it tells to the EB process that the data is ready to be exchanged. In following steps, the tracks data is treated just like other raw data coming from the detector. In this way, the perturbation on the EB is minimal.

This solution has some constraint due to transmission bandwidth between the Readout Boards and the Tracking Boards. Each Readout Boards will transfer data at 100 Gbit/s saturating the bandwidth of the PCIe Gen3 interface. If also the Tracking Boards use PCIe Gen3 interface it is not possible to delivery all the data to the “Artificial Retina”. I found three possible solution to this issue: connect up to two Readout Boards of an EB node to the sub-detector of interest, and the remaining board to a different sub-detector, this do not interfere with the event building process; mount on EB nodes connected to desired sub-detector only two Readout Boards, but this increases the number of nodes and consequently the cost of the system; use Tracking Boards with PCIe Gen4 interface, that doubles the bandwidth respect to Gen3 interface. The EB nodes are compatible with PCIe Gen4, and also some FPGA are compliant with this interface.

7.2 Tracking Boards

The Tracking Boards must fulfil several requirements. The FPGA mounted on the board must have a high number of ALMs to implement all the “Artificial Retina” component. It must have several XCVRs to build the Distribution Network spread over different boards. And, as seen in the previous Section, the Tracking Boards must have a PCIe interface in order to integrate the "Artificial Retina" system in the LHCb DAQ.

Design of a custom board would allow to fulfil all the requirements. However, custom hardware requires long development times and a team with very specific knowledge, and even with the right know-how, a small scale production is expensive. For this reason a desired feature of the final system is that can be implemented using only devices available on the market. The choice fell on the Nallatech 520N[1].

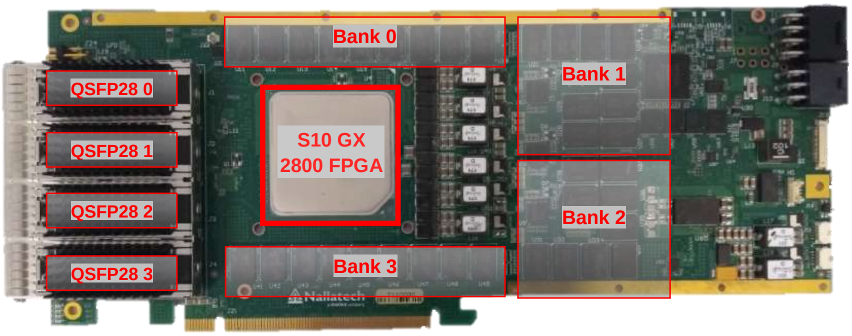


Figure 7.2: Physical Layout of the Nallatech 520N board.

Figure 7.2 shows the physical Layout of the Nallatech 520N board. This board has a PCIe full-height form factor. It is equipped with one of the biggest Intel Stratix 10 FPGA. This FPGA has 933k ALMs, 229 Mbit of internal memory, 11.5k DSPs, and XCVRs supporting a bandwidth up to 26.6 Gbit/s and PCIe Gen3. Its maximum clock speed is 900 MHz. The board expose 16 XCVRs through 4 Quad Small Form-factor Pluggables (QSFPs), and a PCIe x16 interface. It is equipped with 4 banks of 8 GB DDR4 SDRAMs. These features fulfil all the requirements not only from a computational point of view, but also from a communication point of view. Even if the final system will be implemented in this board, during development two additional boards were used: the Dini DN0237, and the Bittware a5pl.

The Dini DN0237, shown in Figure 7.3, is a prototyping board equipped with two Intel Stratix V FPGAs, each with 359k ALMs. It expose 96 XCVRs with a maximum speed of 12.5 Gbit/s, providing a total I/O bandwidth toward the external world in excess of 1.2 Tbit/s. This is an unusually large bandwidth to be found on a single board, as most applications do not require it. The maximum clock speed of the FPGAs of this board is 650 MHz.

The Bittware a5pl is a PCIe Low Profile board. It is equipped with an Intel Arria V GZ FPGA. This FPGA has 170k ALMs, XCVRs supporting a bandwidth up to 12.5 Gbit/s and PCIe Gen3. The maximum clock speed of this FPGA is 650 MHz. The board expose 8 XCVRs through 2 QSFPs, and a PCIe x8 interface. Figure 7.4 shows the physical Layout of the Bittware a5pl board.

[1] https://www.bittware.com/fpga/520n/

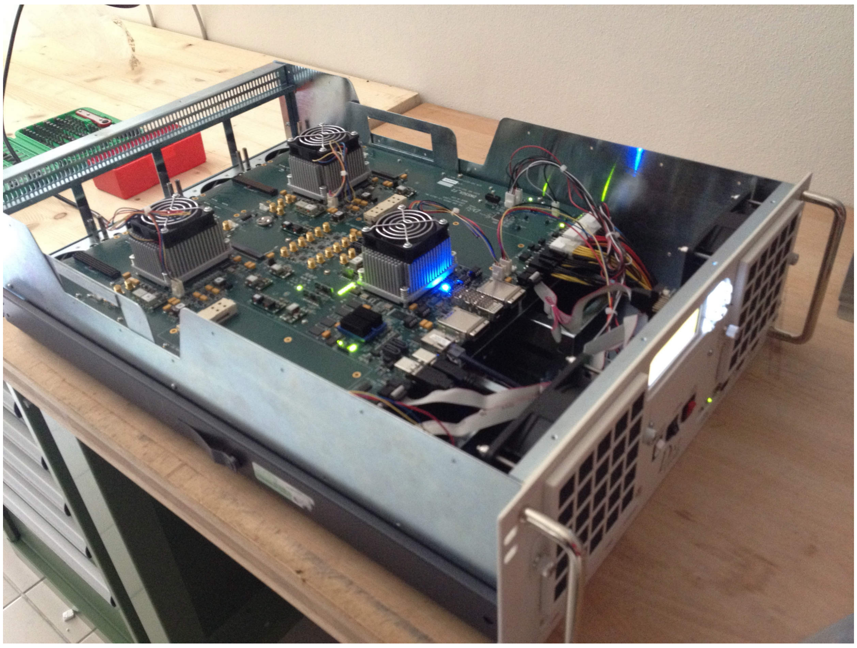


Figure 7.3: The Dini DN0237 prototyping board.

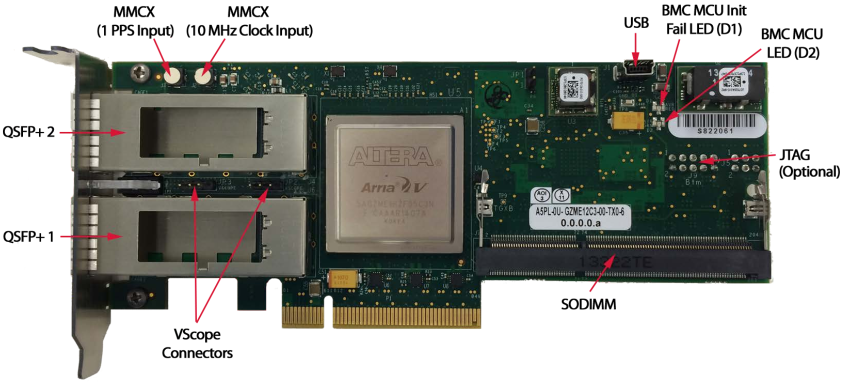


Figure 7.4: Physical Layout of the Bittware a5pl board.

7.3 Fast Dispatcher implementation

The Dispatcher implemented in the RETINA prototype was written as a finite-state machine (FSM) with a high-level tool that describes logic block as a graphical state diagram. The graphical tool does not allow to exploit full hardware flexibility, therefore the design of this component is not optimised to sustain a high throughput. In fact the Dispatcher's Merger was unable to process more than one hit every three clock cycles, creating a bottleneck. Modifying the old component would not have lead to the desired speed boost, so I decided to rewrite the Merger and the Splitter directly in VHDL.

Merger

Figure 7.5 shows the structure of the Fast Merger. The main components are the "R0" "R1" and "State" registers, the "MUX" multiplexer, and the FSM that controls the Merger according to the inputs.

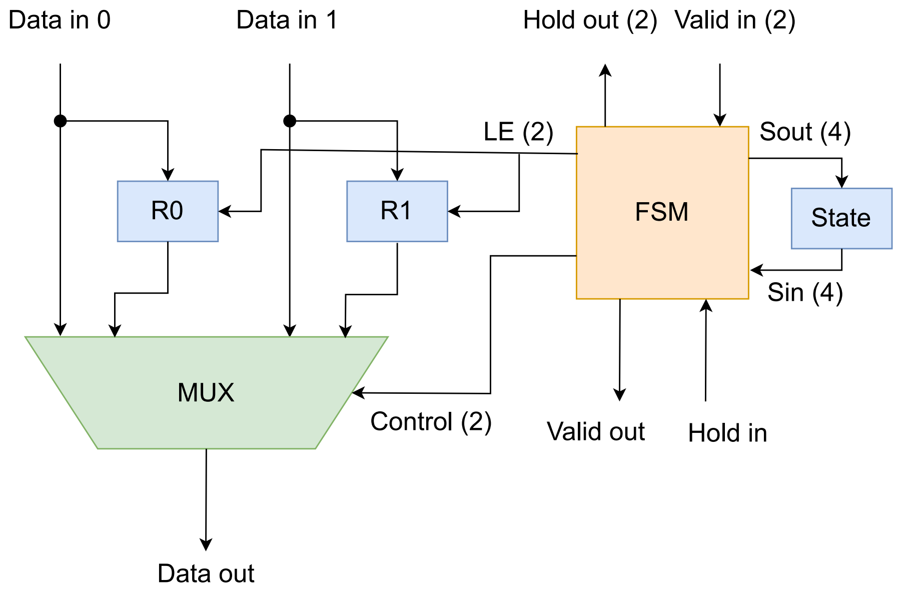


Figure 7.5: Structure of the Fast Merger. The number inside the bracket indicates the width of non-data signals, if omitted the width is 1.

The "R0" and "R1" registers store the incoming data when it cannot be sent out. This happens if the "Hold in" signal is asserted and then the next component is not ready to accept data, if the FSM chooses to send data from the other line/register, or if the incoming data is an End Event (EE). The EE is a special word that separates the hits of a event from the hits of a different one. When an EE is delivered to an input line, that line must be stopped, otherwise hits of different events could be mixed. A single EE is sent as output when an EE is received also to the other input line. The EE word contains the ID of the event, to allow checking that the hits belong to the same event. The "State" register indicates if the "R0" and "R1" registers are empty, contain data, or EEs.

The FSM generates signals that regulate the behaviour of the others components. It takes in input the "State" register, the "Hold in" signal, the "Valid in" signal for the two lines, and if the incoming data are or not EEs, for a total of 9 bit of information (512 combinations). Not all combinations are valid, as an example "Valid in" of a line can not be asserted if the register of that line contains data, in case of forbidden combinations the Merger raise an error flag. The outputs are the latch enables (LEs) for the "R0" and "R1" registers, the "Hold out" signals for the two lines, the "Valid out" signal, the "Control" signal that select the input of the multiplexer, the data that will be written in the "State" register, and the error flag.

The FSM works with combinatorial logic: the output signals are a combination of "and", "or", and "not" operators between the input signals. I wrote the truth table that describe the Merger.

Splitter

I also redesigned the Splitter in order to match the performance of the Fast Merger. Since it has one data input and two outputs, its structure is simpler than the one of the Merger (Fig. 7.6), with one register for the data and a 2 bit "state" register. The FSM, implemented with combinatorial logic, has 6 input signals and 8 outputs. Even if the data lane is one, there are two "Valid in" signals. The assertion of one "Valid in" indicates that the incoming data must be sent to the corresponding output. Both signals can be asserted at the same time; in that case data will be duplicated and sent to both the data out lines. Placed before the Splitter, the routing LUT generates the "Valid in" signals, allowing to route hits to the desired Engines. However the routing LUT is optional, and these signals can be generated in a different way.

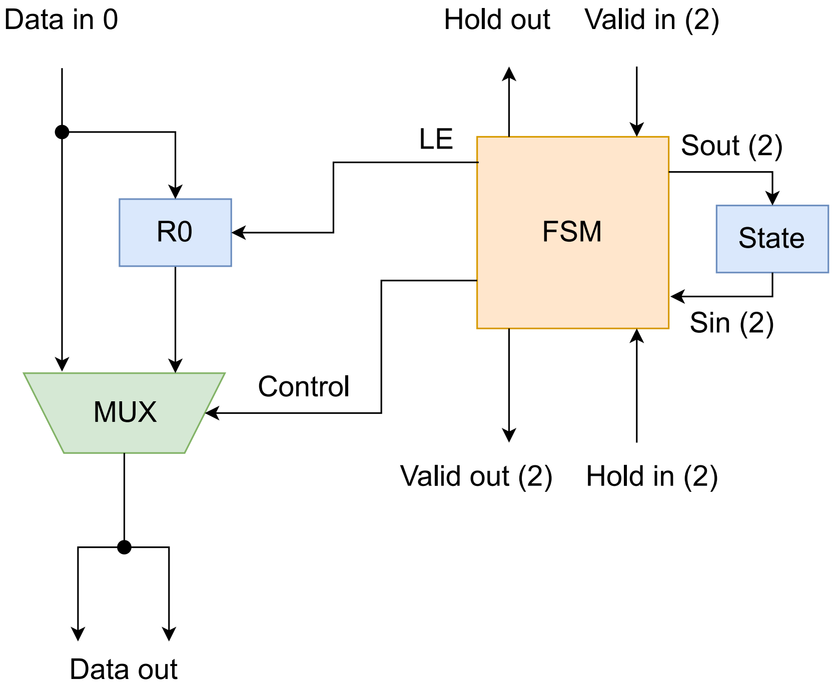


Figure 7.6: Structure of the Fast Splitter. The number inside the bracket indicates the width of non-data signals, if omitted the width is 1.

Throughput measurement

Implementing the FSM of the Merger and Splitter with combinatorial logic, and using registers to store incoming data in case the "hold" signal is raised allow to remove idle states and to process a hit every clock cycle. The old Merger was unable to process more than one hit every three clock cycles, therefore the system speedup should be of a factor 3. In practice, due to the more complex design, the new components cannot work at the same clock frequency as the old ones. The maximum frequency achieved by the old Switch is 400 MHz, while the new design works with a clock of 300 MHz. Since the system throughput is linear with clock frequency, the new system is expected to have a 2.25 times higher throughput. However, when removing a bottleneck, others component may limit the throughput of the system, so the reliable way to quantify the speedup is measure the event rate on a real device.

I updated the firmware of the RETINA prototype with the new Switch, and tested it on the same Stratix V test board. This design represents a system that takes inputs from three different sources, and performs tracking on four independent devices (Fig. 6.5). Three different Switches route the hits from a source to the appropriate engines.

Figure 7.7 shows the comparison of the event rates of the two configuration. Table 7.1 shows the measured event rates, with a sizeable speedup between 1.2 and 2.44 depending on the system occupancy. Due to the new Switch the system throughput is larger than 30 MHz up to an occupancy of 1.5%, thus enabling the reconstruction of tracks in LHCb detectors.

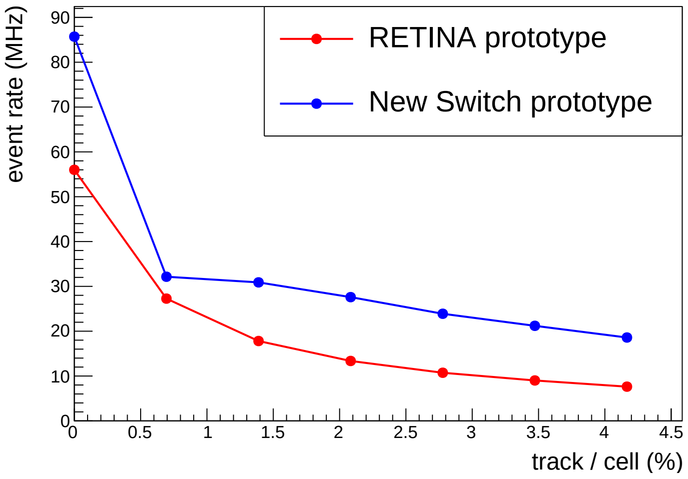


Figure 7.7: Comparison of event rate between the RETINA prototype and the prototype with the new Switch.

7.4 Development of the Distribution Network

The dedicated Distribution Network is a crucial element of the system, which allows to collect data from several DAQ nodes, overcoming the size limits of a single FPGA device, while attaining the high-throughput and low-latency goals of the project. This is a technologically challenging aspect of the system that needs to be thoroughly tested.

	Event rate (MHz)	
Occupancy (%)	RETINA prototype	New Switch prototype
0.00	56.00	85.71
0.69	27.27	32.15
1.39	17.81	30.89
2.08	13.36	27.60
2.78	10.73	23.89
3.47	9.01	21.20
4.17	7.63	18.59

Table 7.1: Comparison of event rate between the RETINA prototype and the prototype with the new Switch.

The RETINA prototype included serial communication between two FPGAs physically mounted on the same board. The present design demonstrates the crucial point that the insertion of serial links does not slow down the system, and that the latency remains within tight limits ($< 1\,\mu s$).

7.4.1 Tolerance to inputs time skew

The setup described above was not a completely faithful reproduction of a distributed system, since implementing all inputs on a single device makes them synchronous, and the clock of the two FPGAs is generated by the same board. The question of the reliability of the sustained operation of this design, in conditions where the timings of all lines are actually independent, was therefore an open one and was explicitly raised during the review of the project organised by the LHCb collaboration. To fully answer this question, I designed a different test setup, specifically aimed at answering this question by verifying the tolerance to asynchronous inputs and the stability of the system, when distributed over several physically distinct boards with independent time references.

The "Artificial Retina" uses the EE word to separate the events. When a component with multiple input lines receives an EE on one line, it stops that line until receiving the corresponding EE also on all other inputs. This effectively re-aligns the inputs at the event boundaries. The hold signal could propagate backward until it reaches the input of the system, but the input rate is fixed by the experiment and can not be paused. To take under control the propagation of the hold signal, several memory buffers are placed between the main components of the "Artificial Retina" design. It is crucial to simulate realistic time skews between the system input, in order to verify that the EE realignment mechanism works properly, and that the buffers are deep enough to prevent the back propagation of the hold signal up to the system input.

I added to the setup the option of emulating a time skew between the different input buffer. I set the input sources to send event at a fixed rate of 30 MHz, with a input delayed by 10.24 µs respect to the others. This is a very conservative time skew, being the upper limit set by the working group that developed the VELO readout. The Engines are the first components that require data from different input buffer, so I expect that the buffers between the XCVRs and the Engines, limited to the ones that receive data from the not delayed sources, are at risk of overflow.

As it turns out, the system behaves as expected, and the buffers get filled by no more than 1000 words. A buffer requires less than 0.5‰ of available memory. I estimated the buffer size of the full-size system. Also in this case the buffer that can become full are the ones after the XCVRs. The worst case occurs when the optical links bandwidth is saturated and the time skew between the sources hits its maximum. In that case, the buffer needs to hold:

$$\frac{25.8\,\mathrm{Gbit/s}\cdot 10.24\,\mu\mathrm{s}}{32\,\mathrm{bit/words}} = 8256\,\mathrm{words}.$$

A buffer of 16k words requires 32 memory blocks. The boards of the full-size system has $\sim$ 12k memory blocks. Since one buffer is required for each of the 16 board XCVRs, the whole buffer system requires around the 4.4% of available resources, a perfectly adequate amount of resources. The conclusion is that the system, as designed, will work correctly even in the most extreme skew conditions expected in the LHCb DAQ

7.4.2 Design of the Distributed Network

The complexity of the network requires to build a distributed system increases exponentially with the number of boards. A full-mesh network, where each node is connected to all the others nodes, require

$$c = \frac{n(n-1)}{2}$$

duplex connections, where n is the number of nodes. The LHCb sub-detector requiring less Readout Boards is the muon system (24 boards), while the SciFi requires 144 boards and the VELO 52. The “Artificial Retina” for one of these sub-detector requires a huge number of connections.

In order to demonstrate that the network can actually be managed, I designed, implemented and tested a prototype network. While current FPGAs have numerous fast serial lines , most commercially available PCIe FPGA boards can only implement a full-mesh network with up to $\sim$ 16 nodes. I also designed the Distribution Network of the VELO “Artificial Retina”, I will describe it in Section 8.3.

Realistic size full-mesh network

The first network I implemented was a full-mesh network between 8 ’logic FPGAs’. For this network I used the same board of the RETINA prototype with 2 FPGAs, implementing in each FPGA several link blocks, grouped in logic FPGAs that works independently. A full-mesh network of 8 nodes requires 28 duplex links. The aim of this test is to demonstrate the feasibility of physical connections between a reasonable numbers of nodes. Figure 7.8 shows the structure of a link block, a logic FPGA, and physical FPGA. A pseudo-random binary sequence (PRBS) generator creates a random bit stream that is sent out by the transmission side of a XCVR. Data coming from the block of an another logic FPGA is received by the XCVR and sent to a PRBS checker. Since the binary sequence is only pseudo-random, the checker can locally generate exactly the same sequence and compare it with the incoming stream. In this way the PRBS checker can detect transmission errors. The logic FPGAs are connected in a full-mesh network.

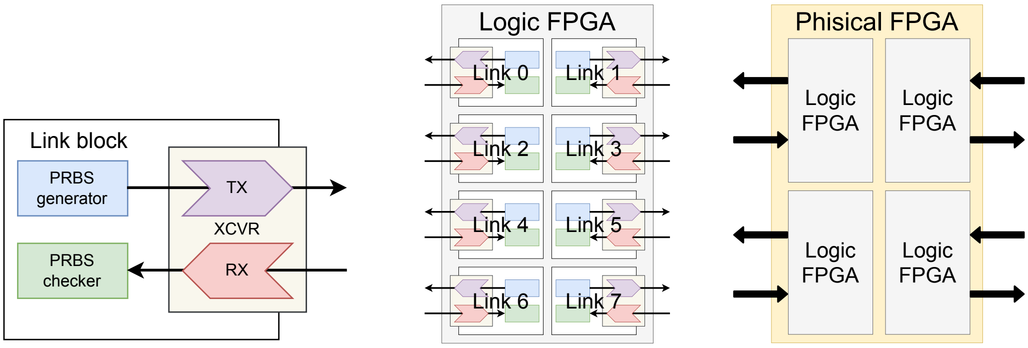


Figure 7.8: Structure of a link block, a logic FPGA, and physical FPGA of the 8 nodes full-mesh network test.

Typically, the XCVRs of a FPGA board are accessible for external communication through QSFP transceivers (Fig. 7.9 left) rather than as individual serial lines. One QSFP connects 4 XCVRs to a single multi-fibres cable, and a FPGA board usually does not have more than 4 QSFPs. To allow connecting each board to as many other nodes as possible, I used breakout cassettes (Fig. 7.9 right) to split the fibres of a same multi-fibres cable and to connect them to different nodes with single fibre cables. In the following description, the cables and breakout cassettes ensemble will often be referred to as the "Patch Panel".

Figure 7.10 shows the results of a one-day long test. No communication errors are detected except on one link. The bit error ratio (BER) of said link is $< 10^{-15}$, which is equivalent to less than one wrong bit per day.

However, the aim of this test was not just to verify the stability of the optical communication through the Patch Panel, but to demonstrate that a large Patch Panel is feasible. The wiring of the Patch Panel worked at the first attempt without fail on a single link. So this does not point out ant major issue about the Patch Panel; however the cabling was pretty chaotic, as can be seen in Figure 7.11. Even using shorter cables, keeping floating cables in front of the breakout cassettes will always produce a similar results. In order to simplify the Patch Panel cabling, I looked for commercial solutions to reduce the number of connectors. A good solution was found in the use of fan-out cables. As shown in Figure 7.12 (left), fan-out cables offer a multi-fibres connector a terminal, and multiple single-fibre connectors on the other

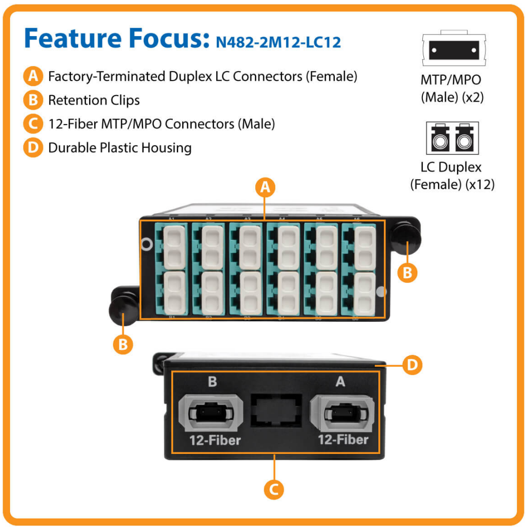
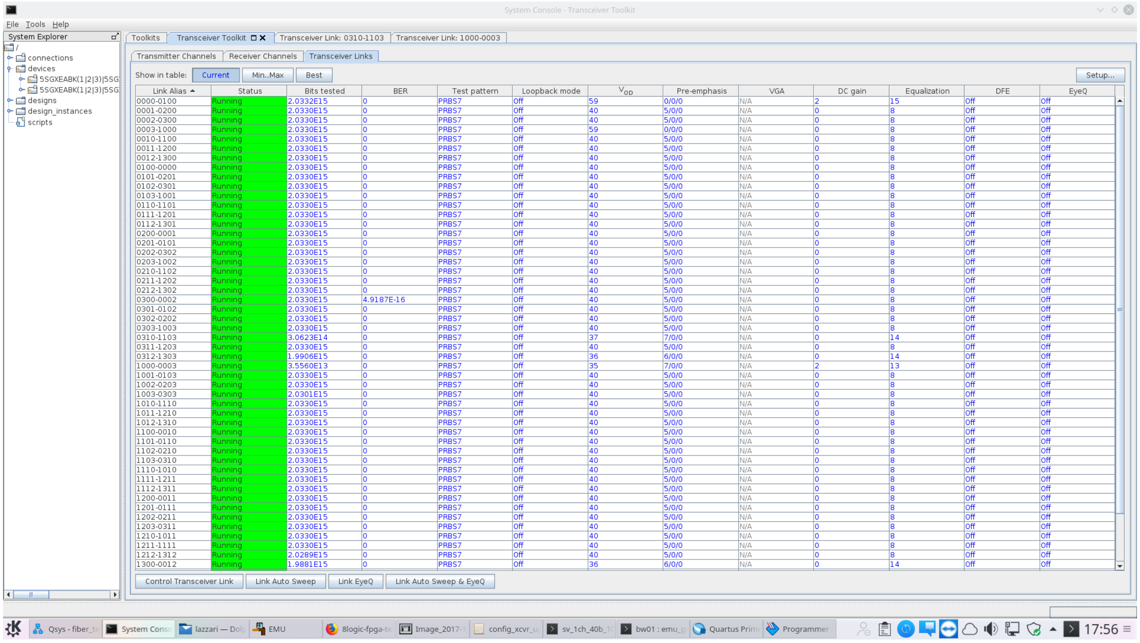

Feature Focus: N482-2M12-LC12
A Factory-Terminated Duplex LC Connectors (Female)
B Retention Clips
C 12-Fiber MTP/MPO Connectors (Male)
D Durable Plastic Housing
MTP/MPO (Male) (x2)
LC Duplex (Female) (x12)
A
B
B
D
B
A
12-Fiber
12-Fiber
C

Figure 7.9: A QSFP transceiver (left), and a breakout cassette (right). A breakout cassette connects each fibre of a multi-fibres cable to a dedicated port.



Link Alias ▲	Status	Bits tested	BER	Test pattern	Loopback mode	V_{OD}	Pre-emphasis	VGA	DC gain	Equalization	DFE	EyeQ
0000-0100	Running	2.0332E15	0	PRBS7	Off	59	0/0/0	N/A	2	15	Off	Off
0001-0200	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0002-0300	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0003-1000	Running	2.0330E15	0	PRBS7	Off	59	0/0/0	N/A	0	8	Off	Off
0010-1100	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0011-1200	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0012-1300	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0100-0000	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0101-0201	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0102-0301	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0103-1001	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0110-1101	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0111-1201	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0112-1301	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0200-0001	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0201-0101	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0202-0302	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0203-1002	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0210-1102	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0211-1202	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0212-1302	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0300-0002	Running	2.0330E15	4.9187E-16	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0301-0102	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0302-0202	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0303-1003	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0310-1103	Running	3.0623E14	0	PRBS7	Off	37	7/0/0	N/A	0	14	Off	Off
0311-1203	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
0312-1303	Running	1.9906E15	0	PRBS7	Off	36	6/0/0	N/A	0	14	Off	Off
1000-0003	Running	3.5560E13	0	PRBS7	Off	35	7/0/0	N/A	2	13	Off	Off
1001-0103	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
1002-0203	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
1003-0303	Running	2.0301E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
1010-1110	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
1011-1210	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
1012-1310	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
1100-0010	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
1101-0110	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
1102-0210	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
1103-0310	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
1110-1010	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
1111-1211	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
1112-1311	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
1200-0011	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
1201-0111	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
1202-0211	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
1203-0311	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
1210-1011	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
1211-1111	Running	2.0330E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
1212-1312	Running	2.0289E15	0	PRBS7	Off	40	5/0/0	N/A	0	8	Off	Off
1300-0012	Running	1.9881E15	0	PRBS7	Off	36	6/0/0	N/A	0	14	Off	Off

Figure 7.10: Results of the full-mesh network test. No communication errors are detected expect on one link.

side. These cables allow also to connect directly a QSFP to the breakout cassettes of different nodes, avoiding floating cables as shown in Figure 7.12 (right). Fan-out cables allow also to halve the number of required breakout cassettes, reducing the costs of the Patch Panel and the space occupied on server rack. Obviously, a fan-out cable can not be connected to another fan-out cable since both single-fibre connectors are male. I planned the connection between multiple boards verifying that network topologies can be implemented in a flexible way even using these cables.

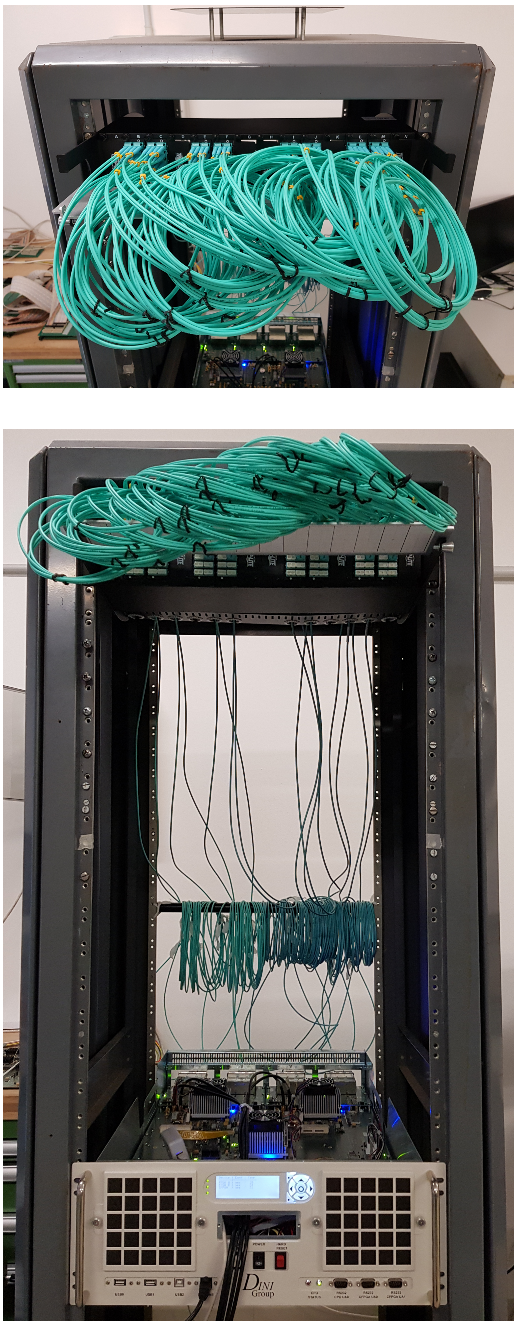


Figure 7.11: Cabling for a 8-nodes full-mesh network.

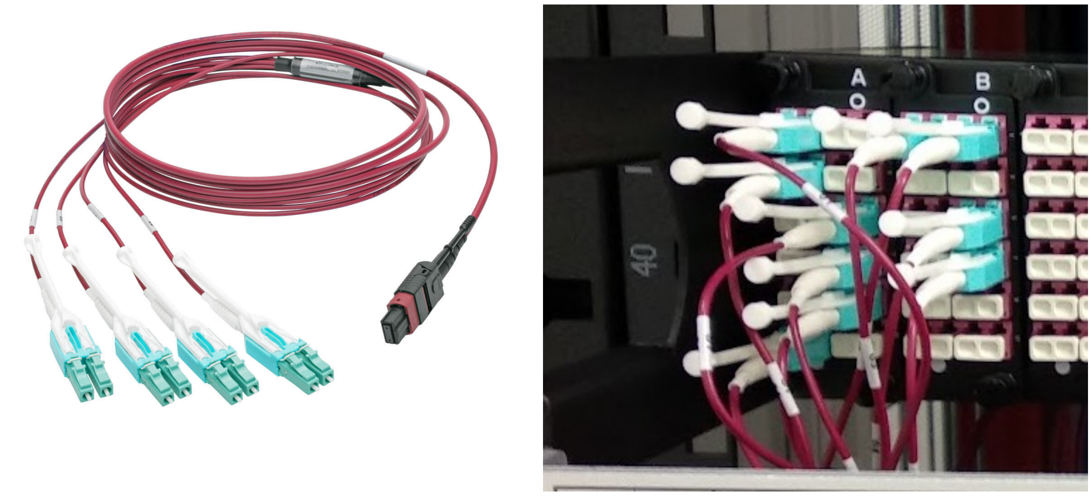


Figure 7.12: A fan-out cable (left). A 5-nodes full-mesh network implemented with breakout cassettes and fan-out cables (right).

Full-mesh network over multiple boards

To verify the communication between multiple boards, I performed a test with 5 Arria V cards in a full-mesh configuration. Every board sends pseudo-random data to the other 4 FPGAs through 10 Gbit/s optical links, for a total of 20 simplex communication channels. The design structure implemented on each board is the same of the 8 logic FPGAs network (Fig. 7.8), but this time the logic is actually implemented in physically separated boards.

All the link blocks are instantiated, but only 4 for board were connected. I tested the system for 23 consecutive days at maximum speed, without detecting transmission errors on all but one link. This allowed me to evaluate the bit error ratio (BER), an important indicator of the quality of a communication channel. This quantity is defined as:

$$\mathrm{BER} = \frac{\#\, transmission\, errors}{\#\, transmitted\, bits}. \tag{7.1}$$

The upper limit to the BER for links where no transmission errors are detected is $1.45 \cdot 10^{-16}$ with a confidence level of 95%. Appendix B reports how the upper limit to the BER was estimated. The BER of the faulty link turns out to be $\sim 10^{-13}$. By reducing the transmission speed, the errors on this link disappeared. This behaviour is attributable to imperfect synchronisation between signals inside the FPGA. The software used to compile the firmware allows to tune this synchronisation, but since this was only a preliminary test with PRBS generators, this type of fine tuning was postponed to the final design.

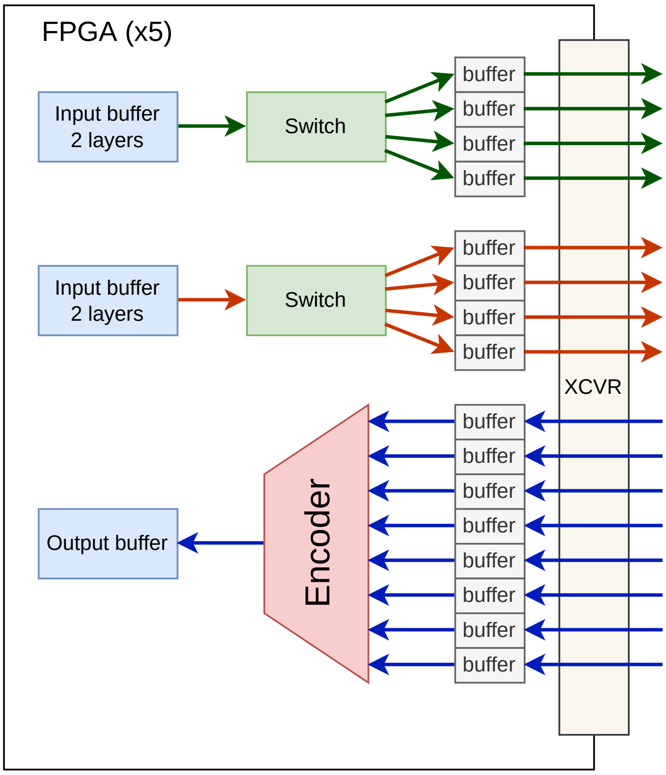


Figure 7.13: Structure of the design for implementing a realistic Distribution Network. This design was implemented on 5 FPGAs connected in a full-mesh network.

Realistic Distribution Network over multiple boards

I implemented a realistic Distribution Network using simulated hits, written into the FPGAs memories, from different events, instead of pseudo-random sequences. The design used for this test (Fig.7.13) is similar to the design of the "Artificial Retina" prototype (Fig.6.5). Two independent switches read data from input buffers and route them to appropriate external lines. Data coming from external lines are stored in buffers. For this test, I replaced the Engines by a special component, called Encoder. The Encoder has the same requirement of the Engine on EEs alignment, raising a flag if EEs with different identifier are delivered, but it merges the input lines in a single wider line without performing the pattern-recognition functions of the Engine. I had chosen to use the Encoder instead the Engine because, to monitor the correct behaviour of the Distribution Network and the absence of transmission errors, I read the output data from the host computer, and compared it with the simulation. Since the Engines process the data, some transmission errors could be not detectable.

This design was implemented on 5 boards connected in full-mesh as in the previous test. Having implemented 8 lines using only 5 FPGAs, 4 XCVRs per FPGA are closed in serial loopback. That means that all the XCVRs are used, but the data stream leaving the transmitter side of a XCVR is used as input for the receiver side of the same. This is the same protocol used by the RETINA prototype. Data reading on different FPGAs is not synchronised, for a better correspondence to the actual environment in which the system will be placed. During a 5 weeks test I did not detected any error.

Full-speed communication

The network tests described in previous sections were mostly aimed at testing the logic aspects of communications, and were performed at speeds of 10 Gbit/s that are not sufficient for the performance required by the demonstrator. We need to prove the feasibility of operating the “Artificial Retina” system at the full speed of the final application in the LHCb DAQ. The bandwidth of 26 Gbit/s provided by newer FPGAs, like the Intel Stratix 10, is compilant to the requirements of the demonstrator. However the Distribution Network design used in previous tests could not be ported directly to these newer chips since it uses SerialLite II protocol. This is an old protocol that, even if tested at 10 Gbit/s, officially can operate only with bandwidths lower than 6.4 Gbit/s per lane. For this reason Intel has not implemented it on the newer devices and a different protocol is required.

I compared different communication protocols to find a substitute for the SerialLite II. I opted for the Intel SuperLite II V4. This protocol fulfils every requirements of the system since it can operate at any speed, included 26 Gbit/s, it includes flow-control function for implementing back-pressure, it allows to connect each XCVR to a different endpoint. This protocol is fully free and available in source code, so I was able to move some internal components to adapt it to the “Artificial Retina” utilisation case and, if needed, implement it on others FPGA families, without relying on Intel porting plans. Finally, this protocol has a better efficiency than the SerialLite II, *i.e.* the ratio between effective data bandwidth and the raw bandwidth: since each protocol adds a header and some kind of encoding not all the link bandwidth is available for data transfer. The SerialLite II efficiency is less than 80%, the efficiency of SuperLite II V4 is 96.3%.

Intel provides an example design that implements this protocol[2]. After interfacing it to the Stratix 10 boards, I verified its behaviour on a single board with XCVRs closed in loopback through optical fibres. The communication was established, and I had not detected errors running the links at full speed. Also, the flow-control function worked properly. Then I ported the firmware of the realistic Distribution Network (Fig.7.13) on the Stratix 10 board, replacing the XCVRs block that implements the SerialLite II protocol with the newer one. By specification the SuperLite II V4 can bond an arbitrary number of physical lines to increase the total bandwidth of a channel. Since the “Artificial Retina” architecture requires to connect a board to as many nodes as possible, this feature is not required. However the example design bonded 4 lines. During the porting I analysed the source code of the example design and I edited it in order to implement a SuperLite II V4 module for each line. This

[2] https://community.intel.com/t5/FPGA-Wiki/High-Speed-Transceiver-Demo-Designs-Stratix-10-GX-Series/ta-p/735749

required to change the size of internal signals and the number of XCVRs instantiated, and to move the components that generates XCVRs clock outside the SuperLite module: to transfer data at 26 Gbit/s the XCVRs require a clock of 12.9 GHz. Due to the high frequency (ALM clock is lower than 1 GHz), XCVRs use a dedicated clock network with clock generators placed near the XCVRs. During the porting, I respected the requirements placed by the Stratix 10 XCVRs clock network.

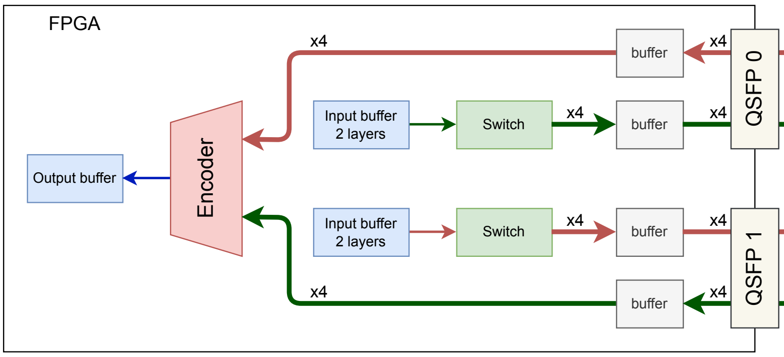


Figure 7.14: Structure of the design of the Distribution Network limited to one FPGA. In this case the XCVRs of a QSFP are connected through optical fibres to the XCVRs of the other QSFP.

Figure 7.14 shows the structure of the design. In a first test, I connected the XCVRs of a QSFP to the XCVRs of the other QSFP through optical fibres. This test allows to check if the design was implemented correctly, and to verify if the connection is stable. I ran the test with the XCVRs set to 12.9 Gbit/s.

I loaded the same design on a second board and I connected them together with two multi-fibres cables as shown in Figure 7.15. I ran the test with the XCVRs set to 12.9 Gbit/s for 90 hours, and setting them to 25.8 Gbit/s for 70 hours.

Finally I implemented a network of 3 Stratix 10 boards. With 3 boards a full-mesh network require only 2 full-duplex links per node, when the design has 8 links. To increase the number of interconnections, I implemented two full-mesh network, one for each QSFP. Figure 7.16 shows the topology of one of these networks. Each FPGA is connected to the other, 2 links per QSFP are closed in loopback through optical fibres. I used the breakout cassettes of the Patch Panel to implement all the links. I ran the test for 135 hours with XCVRs set to 25.8 Gbit/s, without detecting any errors.

The extensive tests that I performed demonstrate that the proposed protocol and overall design arrangement are effective solutions to the problem of inter-board communication in the "Artificial Retina" system. The conclusion is that, at this point of the development work, there is no reason to think that a full-size "Artificial Retina" system cannot be made to operate correctly and consistently with the specifications for extended period of time, to process data at the LHC crossing frequency of 30 MHz.

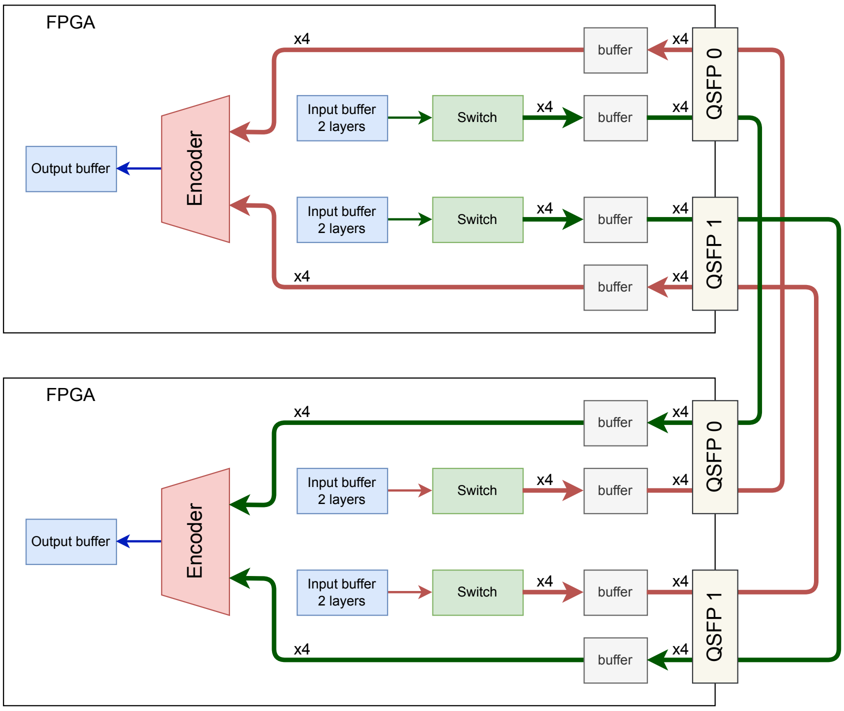


Figure 7.15: Structure of the design for implementing a realistic Distribution Network. This design was implemented on 5 FPGAs connected in a full-mesh network.

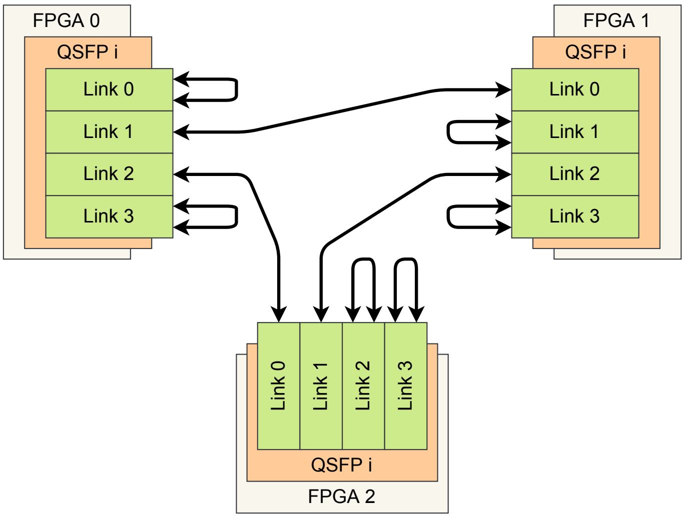


Figure 7.16: Topology of the network of three Stratix 10 board. Some links are connected in loopback through optical fibres. Two network was implemented, one for each QSFP.

Chapter 8

Building a working demonstrator for Run 3

8.1 Benefits from real-time pre-build tracking in LHCb

In the previous Chapter, I demonstrated that there are no reasons to think that a full-size "Artificial Retina" system cannot be integrated in the LHCb DAQ, performing the tracking in real-time potentially of the whole detector. Even tracking only some sub-detectors, LHCb can benefit from this system and improve its physics program.

In the Expression Of Interest proposing LHCb Upgrade II, the "Artificial Retina" is proposed to reconstruct tracks downstream of the magnet at the earliest trigger level [29]. This capability is not part of the baseline Run 3 trigger scheme on account of the significant CPU time required to execute the search [61]. Reconstructing these tracks will allow to tag the presence of long-lived s-quark final states (K^0_S, $\mathit{\Lambda}$) decaying outside the VELO. This will yield a large acceptance boost for these particles, that play a crucial role as final states of many important bottom and charm decays for *CPV* measurements - first and foremost, the golden candidate decays for charm *CPV* study, like the $D^0 \to K^0_S K^0_S$ decay introduced in Section 2.3.3, and the $D^0 \to K^0 \overline{K}^{*0}$ and $D^0 \to \overline{K}^0 K^{*0}$, that are the goals of the analysis presented in the first section of thesis. But many other important decays will benefit as well - amongst them: $B^0 \to K^0_S K^0_S$,$B^0 \to K^0_S K^0_S K^0_S$, $B^0 \to \eta K^0_S$, $D^+_s \to K^0_S \pi^+$, $D^+ \to K^0_S K^+$, $\mathit{\Lambda}^0_b \to 3\mathit{\Lambda}$, and $K^0_S \to \mu\mu$.

In an wider perspective, boosting the processing power available in the early trigger stages is a crucial enabler for all LHCb physics targets that require the collection of high-statistics samples of data at high luminosities.

Downstream tracks in the early trigger would also open new avenues for searches of exotic long-lived particles at LHCb. With such tracks reconstructed, LHCb could cover a phase-space region currently unreachable by other experiments in the search for Heavy Neutral Leptons with mass $< 10\,\mathrm{GeV}/c^2$ and $c\tau < 10\,\mathrm{m}$.

The "Artificial Retina" can be applied also to the muon system: combining the information of this sub-detector with few layers of the T-stations, muon information can be provided to the trigger level without momentum of impact-parameter cuts. Reconstruction of lower-p_T muons increases efficiency for multi-body muon final states. Amongst them, $B^0 \to K^{*0}\mu\mu$, is of fundamental importance to rule out or confirm the recent experimental hints [76] of violation of lepton universality. Looser p_T and impact parameter thresholds during the initial selection of interesting track candidates in the VELO in the HLT farm result into larger reconstruction efficiency of golden channels as $B^0_s \to \mu\mu$, $B^0 \to \mu\mu$, and $D^0 \to \mu\mu$. The Run 3 request of a significant impact parameter, that can be removed with the "Artificial Retina", completely excludes prompt muon signals, like $Z \to \mu\mu$, $Z \to 4\mu$, $J/\psi \to \mu\mu$, and $\Upsilon(nS) \to \mu\mu$. They have several valuable applications. Heavy quarkonia are crucial calibration channels for the tracking system, enabling all precision measurements based on tracking. Moreover, heavy quarkonia production is a long-standing puzzle in QCD, and LHCb is the only experiment with a potential to continue to investigate it at higher precisions.

A detailed, specific Technical Design Report for the use of the "Artificial Retina" system within the DAQ system of the LHCb Upgrade II is currently in preparation. Given the size and impact of this project, an important ingredient in this process is the construction and operation of a limited-size, but complete demonstrator of the "Artificial Retina" architecture working on a portion of the detector in the real data taking environment of LHCb. This will be the subject of the remainder of this Chapter.

8.2 "Artificial Retina" VELO demonstrator

The demonstrator represents a realistic implementation of a full "Artificial Retina" system, reconstructing a significant portion of a sub-detector in a parasitical test during Run 3. The SciFi is readout by ~ 140 boards, therefore it requires too many Tracking Boards to be a good candidate for the demonstrator.

The VELO, with its 52 Readout Boards, is a relatively compact sub-detector. At the same time it is the first detector reconstructed by HLT, mandatory for all the subsequent steps. Currently its reconstruction requires almost a half of the available HLT1 computing resources, making it a very significant benchmark test for a new processor.

Studies of the physics performance attainable with a "Artificial Retina"-based VELO tracker are already available [77–79]. According to these studies, the HLT1 algorithm performance is negligibly affected when based on tracks produced by the "Artificial Retina". The "Artificial Retina" produce a slightly higher number of ghost tracks, *i.e.* misidentified tracks, compared to the standard HLT1 reconstruction; however the effect is less than 5% and remains under control. In most cases, the efficiency of the "Artificial Retina" reconstruction is within less than a percent from the HLT1 reconstruction. Figure 8.1 shows the efficiency as a function of the z coordinate of the track origin vertex simulation 1000 events containing a $B^0_s \to \phi\phi$

decay, and Table 8.1 reports the measured efficiency for long tracks.

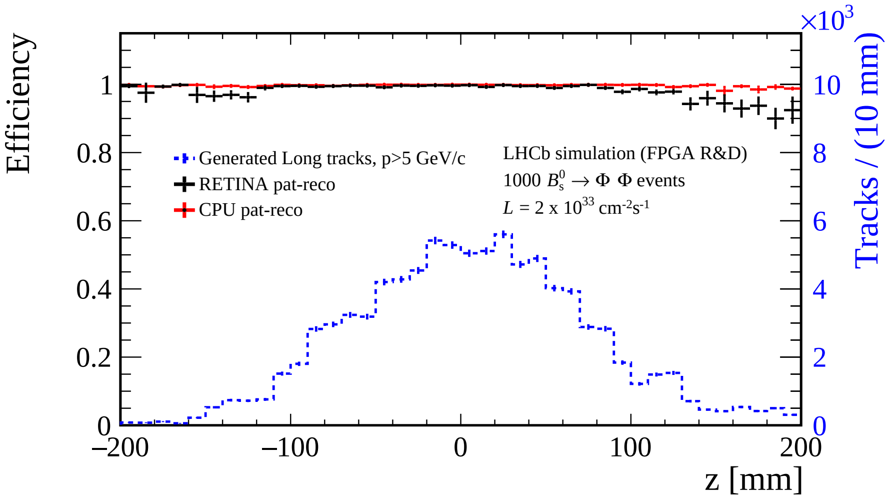


Figure 8.1: Comparison between VELO tracking efficiencies obtained with HLT1 and the "Artificial Retina" reconstruction algorithm. The efficiency is shown as a function of the z coordinate of the track origin vertex. From Ref. [77–79].

Track type	Eff. HLT1 rec. (%)	Eff. "Artificial Retina" rec. all z	 fiducial z-region
Long tracks	99.84 ± 0.02	99.27 ± 0.06	99.45 ± 0.05
Long tracks from b	99.61 ± 0.13	99.24 ± 0.21	99.41 ± 0.18
Long tracks from c	99.89 ± 0.12	98.50 ± 0.53	98.62 ± 0.53

Table 8.1: Summary of efficiencies of the VELO tracking algorithm for different type of tracks using both the HLT1 and the "Artificial Retina" reconstruction algorithm. Numbers obtained on 1000 simulated $B_s^0 \to \phi\phi$ decays. The efficiency (defined as the number of reconstructed tracks divided by the number of reconstructible tracks, *i.e.* tracks with at least 3 hits in the VELO) is calculated using long tracks with $2 < \eta < 5$, $p > 5\,\mathrm{GeV}/c$ and with more than 5 hits (Monte Carlo truth) in the VELO detector. Tracks belong to the fiducial region if the z coordinate of the origin vertex is located between $-200\,\mathrm{mm}$ and $200\,\mathrm{mm}$. From Ref. [77–79].

The studies pointed out that of the 26 VELO layers, only the 19 placed in the forward region with respect to the nominal interaction point are required for tracking. The remaining layers are mainly useful to optimise the primary vertex precision. Two Readout Board acquires data from a VELO layer: one for each module, therefore 38 Tracking Boards are required to read data from the detector. The "Artificial Retina" system considered by the performance studies required a nominal number of 100k

Engines, excluding the regions outside VELO geometrical acceptance, the effective number of Engines is 62k.

8.3 Implementation of VELO Distribution Network

Even if the working principle of the "Artificial Retina" Distribution Network is general, its size and detailed topology depends on the detector, the number of boards, and the number of available links on each board. I studied how to structure the Distribution Network for the VELO

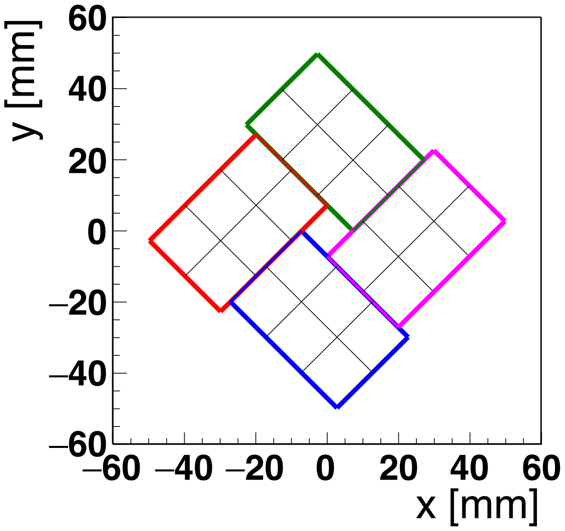


Figure 8.2: Transverse view of the VELO. Each layer is naturally divided in 4 zones.

Since the magnetic field in the VELO region is negligible, the studies parametrise a track by two quantities: the x and y coordinates of the track at a given z coordinate; a track is then represented as the intersection point with a given transverse plane. Thus, the VELO geometry suggests to treat each layer as divided in 4 quadrants (Fig. 8.2). The tracks with hits on a specific quadrant are concentrated in a well defined area of the track parameters space. I performed a logical division of the track parameters space in 4 quadrants according to the natural VELO structure. The FPGAs that implements the engines of the the same space quadrant are called a group, and they are paired to the Readout Boards that read the corresponding physical quadrant of the various layers. If a track intersects a layer quadrant, it will likely intersect the same quadrant on an another layer. Thus the FPGAs belonging to the same group need to exchange a larger numbers of hits, requiring a highly interconnected network. FPGAs of different groups may also need to share hits, but they will need less bandwidth.

I drew the Distribution Network according to the Dragonfly topology [80, 81]. With this topology I can ensure low latency, high bandwidth, and a number of link

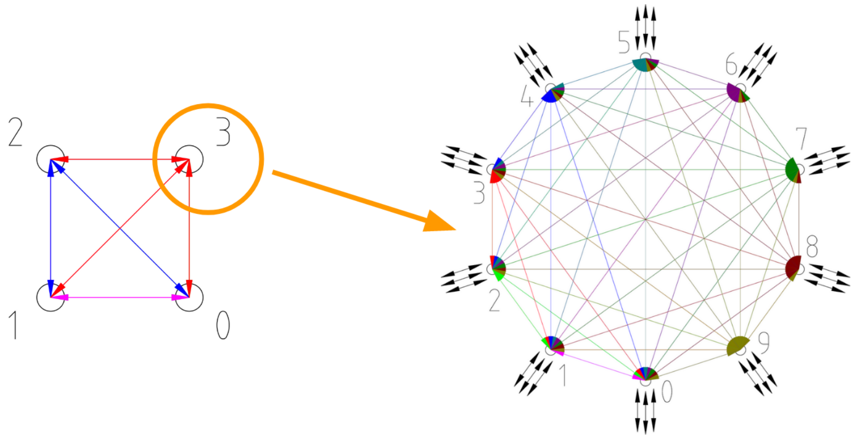


Figure 8.3: Dragonfly topology implemented as a full-mesh network of 4 full-mesh subnetworks based on 10 FPGAs each.

per FPGA compatible with the Stratix 10 boards specification. Since 38, the number of Tracking Boards indicated by the studies, is not divisible by 4, I rounded it up to a 40 boards system. Each group will be composed of 10 boards, interconnected in a full-mesh network. Then the i-th FPGA of a group will be connected to the i-th FPGA of the other three groups creating a full-mesh of full-meshes. (Figure 8.3). This configuration requires 12 links for each FPGA. The chosen Stratix 10 boards are equipped with 16 XCVRs.

Figure 8.4 shows the structure of the VELO Distribution Network with the Switches and the links between FPGAs. A set of Switches with one input and 4 outputs arrange the hits coming from the Readout Boards by quadrant. The hits relevant to the Engines implemented in the same group remain inside the FPGA, while others are sent to the other groups through the inter-group optical links. A second set of Switches with 4 inputs and 10 outputs and the intra-group links route the hits to the right FPGA within the group. The final set of Switches gather the hits and deliver them to the Engines implemented in the FPGA. Hits exchange between not-directly connected FPGAs require an extra hop, increasing the communication latency, but still remaining below a µs.

8.4 LHCb testbed initiative

During data tacking the access to EB will be strongly limited. Therefore, the demonstrator cannot be directly installed in the EB. To allow performing the present and other R&D studies for the future upgrade, LHCb established a *coprocessor testbed*, to run parasitical tests of new processing solutions in realistic DAQ conditions during Run 3. This is the target environment of our demonstrator.

Physically located in the main building of LHCb site, the testbed hosts several projects under development within RTA-WP6. Different projects have different requirements, Figure 8.5 shows the conceptual scheme of interconnections between a

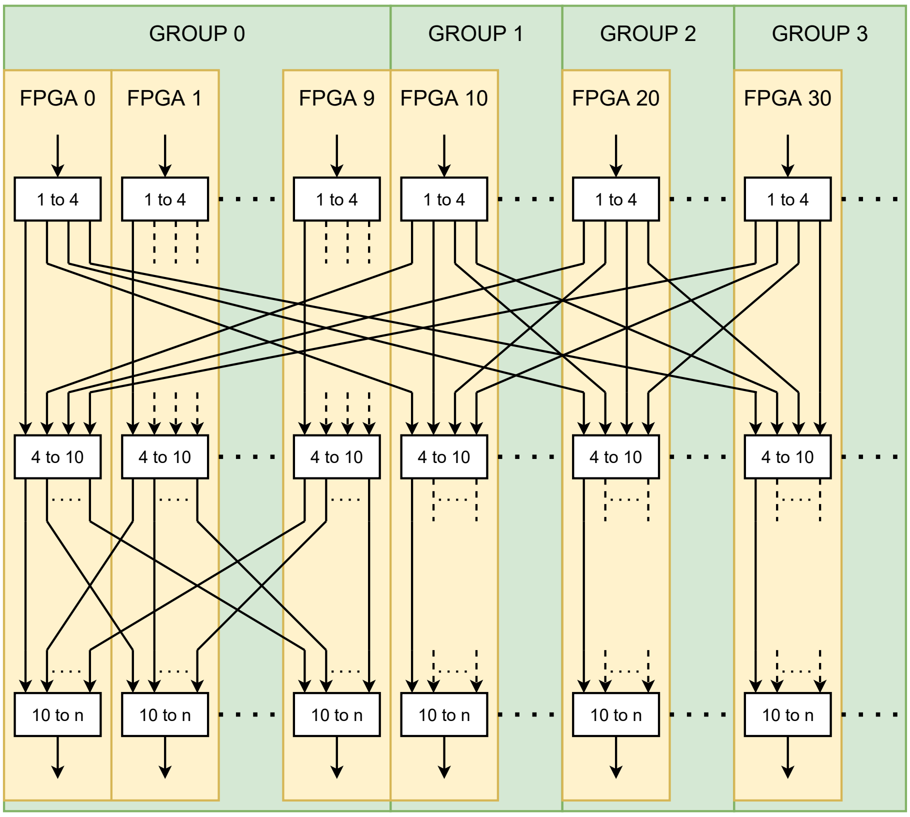


Figure 8.4: Structure of of the VELO Distribution Network.

EB server and a testbed server, the testbed will be able to accommodate projects that work “pre-build” (like the “Artificial Retina”) and projects that work “post-build”.

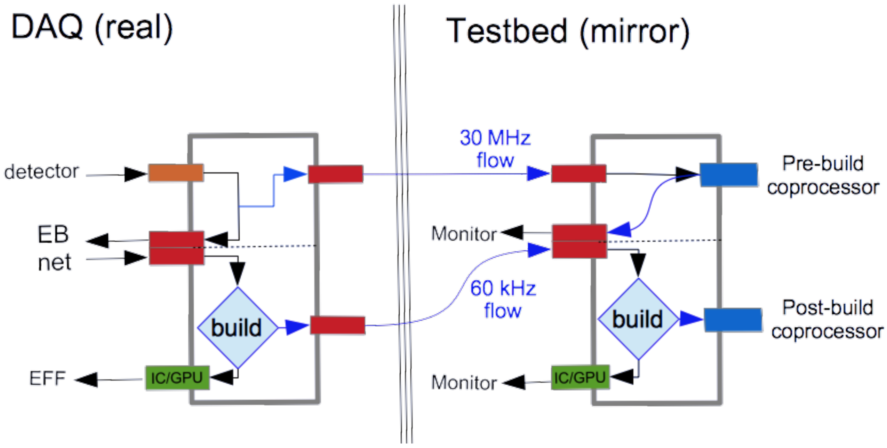


Figure 8.5: Conceptual scheme of parasitic testbed.

The INFN funded the realisation of the “Artificial Retina” demonstrator, as a 2-year project to be completed by the end of 2022. It will process data from a

VELO quadrant, considered a significant portion of this detector. Therefore will be implemented a Distribution Network group, installing 10 Tracking Boards. The boards have already been ordered, but not all of them delivered, due to the well-know current difficulty in obtaining certain electronic parts on the open market.

The Patch Panel will be made with 5 breakout cassettes and 15 fan-out cables, already available and used to built the Full-mesh network over multiple boards (Sec. 7.4.2).

8.5 Implementation of VELO Engine

The RETINA prototype performs tracking for an axial-detector. Since the VELO is a pixel detector, the demonstrator engine must handle 2D hits. The working principle of the "Artificial Retina" for 1D and 2D detectors is the same. As explained in Section 6.2.1 and Section 6.2.2, the Engine performs the weighted sum of the Euclidean distances between the hits and the cell receptors. For a given cell and layer the 2D distance is

$$d(h,t) = \sqrt{(h_x - t_x)^2 + (h_y - t_y)^2} \tag{8.1}$$

where h_x and h_y are the hit coordinates of h, and t_x and t_y are the receptor coordinates. The distance calculation in FPGA require at least two DSPs to compute the exponentiation plus the resources needed by the square root. Moreover, the resources required must be multiplied by the number of Engines and the number of input implemented for each Engine. The final system will instantiate around 2500 Engines per boards, implementing Engine with 3 inputs (like for the RETINA prototype), 15k DSPs are required. The Stratix 10 board, that mounts one of the biggest FPGA, has a 11.5k DSPs. Therefore I evaluated other solutions to keep under control the utilisation of hardware resources.

FPGAs are particularly well suited to perform operations with integer or fixed-point variables. Hits coordinates on the detector are discrete variables. This allows to evaluate advanced mathematical function with a reasonable amount of resources. The "Artificial Retina" adopts a truncated Gaussian function to weight the hits distance, for 1D hits it is

$$w(h_x) = \begin{cases} 0 & \text{if } d_s \leq |h_x - t_x| \\ \exp\left(\dfrac{-(h_x - t_x)^2}{2\sigma^2}\right) & \text{if } |h_x - t_x| < d_s \end{cases}, \tag{8.2}$$

where σ is a parameter adjusted to optimise the sharpness of the response of the receptors, and d_s is a cutoff parameter for reducing Engines input bandwidth. Evaluating the weight function on the fly would require a high amount of resources, but given that the hit coordinates can only take a limited number of discrete values, this function was evaluated beforehand for each possible value of $d < d_s$ (64 values), and the results loaded in a LUT. The Engine computes $d(h_x) = |h_x - t_x|$ then it retrieves $w(h_x)$ reading the $d(h_x)$-th word of the LUT. In this way $w(h)$ is quickly evaluated, using only a memory block of 512 bits. Figure 8.6 shows the structure of the Engine, with the LUT storing the precalculated values of the weight function.

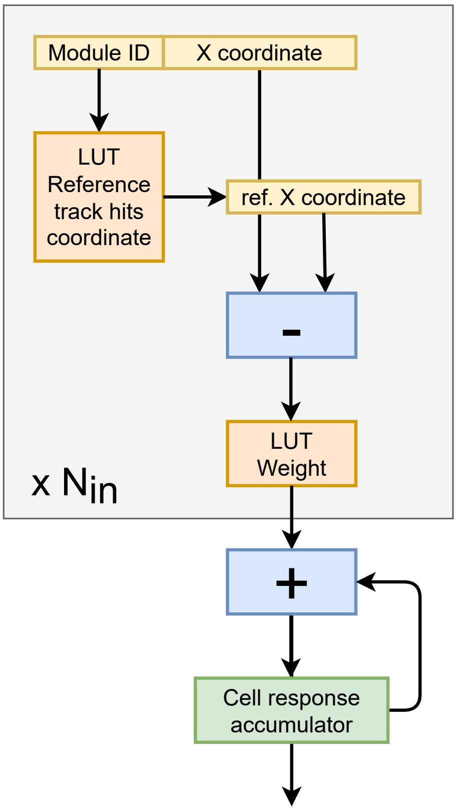

Module ID
X coordinate
LUT Reference track hits coordinate
ref. X coordinate
-
LUT Weight
x Nin
+
Cell response accumulator

Figure 8.6: Structure of the Engine. Block inside the grey box are replicated to match the number of Engine inputs.

This approach, very effective in dealing with 1D hits, however, does not scale well to 2D hits. I evaluated the resources required to evaluate the weight of 2D hits by the same approach:

$$w(h_x, h_y) = \begin{cases} 0 & \text{if } d_s \leq d(h,t) \\ \exp\left(\dfrac{-[(h_x - t_x)^2 + (h_y - t_y)^2]}{2\sigma^2}\right) & \text{if } d(h,t) < d_s \end{cases} . \tag{8.3}$$

In this case the LUT must store the value of $w(h_x, h_y)$ for each possible pair (h_x, h_y). Consequently, the each LUT requires 32.8 kbit of memory, for a total of 246 Mbit- versus a total availability of 229 Mbit in the whole chip.

A better approach can be devised by rewriting Equation 8.3 as follows:

$$w(h_x, h_y) = \begin{cases} 0 & \text{if } d_s \leq d(h,t) \\ \exp\left(\dfrac{-(h_x - t_x)^2}{2\sigma^2}\right) \cdot \exp\left(\dfrac{-(h_y - t_y)^2}{2\sigma^2}\right) & \text{if } d(h,t) < d_s \end{cases} . \tag{8.4}$$

In this way the 2D weight is obtained as the product of two 1D weight. I have therefore coded an Engine that separately computes two 1D weight with two LUT, and then uses a DSP to multiply them and obtain the 2D weight. Figure 8.7 shows the structure of this 2D Engine.

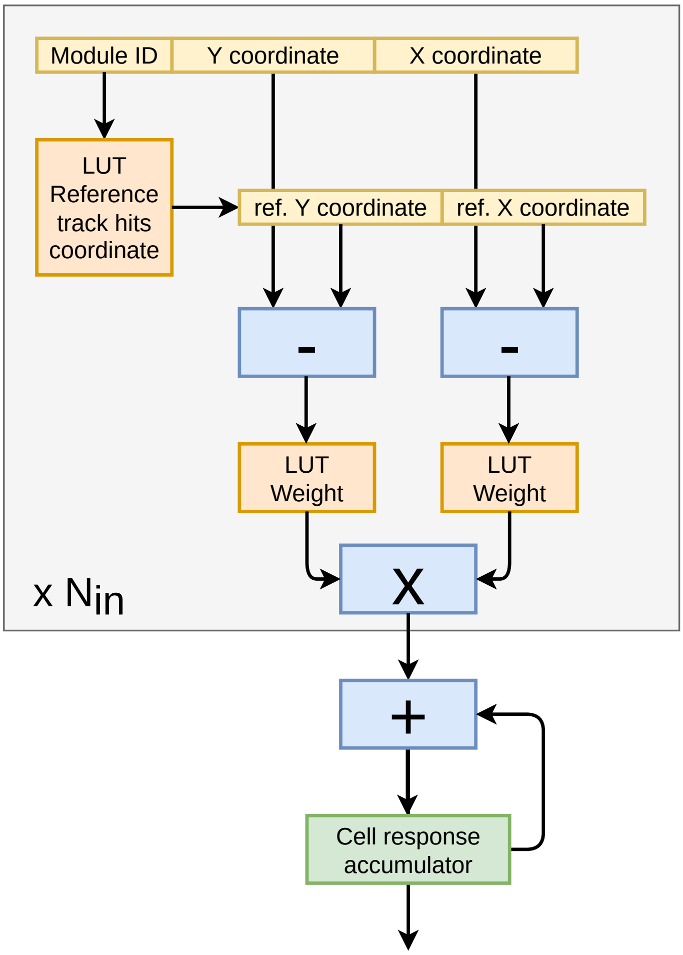


Figure 8.7: Structure of the 2D Engine. Block inside the grey box are replicated to match the number of Engine inputs.

This allow to process a hit on every clock cycle in each input line, exactly as the 1D Engine. This Engine requires 1024 bit of memory for the weight LUT and 1 DSP. The FPGA resources needed to implement 2500 3-input Engines of this type are 7.68 Mbit and 7.5k DSPs; therefore 2500 Engines can be instantiated on a Stratix 10 board.

In Equation 8.4 the Euclidean distance is still required in the inequality with the cutoff d_s. This inequality produce a circular cut around the weight peak. Since the uncut weight

$$w_{uncut}(h_x, h_y) = \exp\left(\frac{-(h_x - t_x)^2}{2\sigma^2}\right) \cdot \exp\left(\frac{-(h_y - t_y)^2}{2\sigma^2}\right) \tag{8.5}$$

is a monotonically decreasing function in $d(h,t)$, I could implement the cut on the distance via an appropriate cut on $w_{uncut}(h_x,h_y)$ value:

$$w(h_x,h_y) = \begin{cases} 0 & \text{if } w_{cut} \leq w_{uncut}(h_x,h_y) \\ w_{uncut}(h_x,h_y) & \text{if } w_{uncut}(h_x,h_y) < w_{cut} \end{cases}, \tag{8.6}$$

with the value of w_{cut} chosen to reproduce the same cut performed by d_s. Figure 8.8 shows the weight function according to Equation 8.4 (left) and Equation 8.6 (centre). The cut on $w_{uncut}(h_x,h_y)$ correctly reproduces the cut on the distance. Figure 8.8 (right) shows little border effects, this is generated by the integer rounding and a choice of w_{cut} value oriented to optimise the in hardware implementation. These differences are negligible for the "Artificial Retina" purposes.

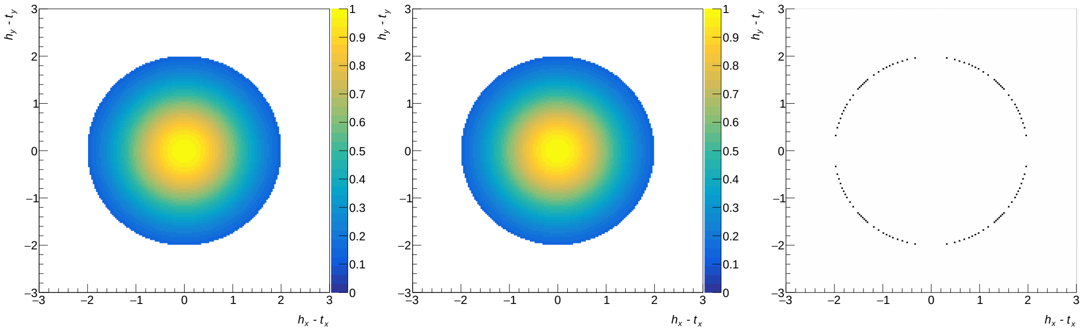


Figure 8.8: Weight function calculated according to Equation 8.4 (left), to Equation 8.6 (centre), and the difference between the two (right). Axis in units of the cells grid pitch.

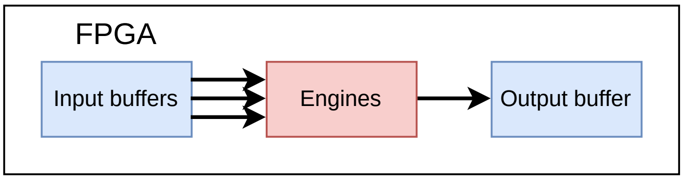


Figure 8.9: Structure of the 2D Engine test design.

8.5.1 Throughput measurement

I prepared a simple test design (Fig.8.9) with three input buffers (one for each Engine input) and a block of 2D Engines. Then I verified the 2D Engine design with the Questa Advanced Simulator[1]. I provided to the Engines hits generated by the VELO "Artificial Retina" C++ simulation and compared the output of the Engines with the tracks reconstructed by the simulation. The two results are identical bit a bit.

[1] https://eda.sw.siemens.com/en-US/ic/questa/simulation/advanced-simulator/

The critical performance parameter is however the throughput. The 2D Engine can process a hit per clock cycle per each input, like the 1D Engine (Sec. 6.2.2), so it should in principle sustain events at the same rate of 1D Engines. To verify that, I provided to the Engines events of the same size of the ones used with 1D Engines and measured the event rate on Questa simulation. Figure 8.10 and Table 8.2 shows the measured event rate. The measurements are performed with a clock frequency of 300 MHz, the same used for the throughput test in Section 7.3. These event rates are not directly comparable with the rates reported in Table 7.1, because in that case the design includes also the switch. However the event rate of the system is determined by the slowest component; if the values reported in Table 8.2 do not show drop respect to the ones in Table 7.1, the 2D engine is faster than the switch or as fast as the 1D engine. The 2D Engines does not show drops in throughput, and the event rate is greater than 30 MHz for occupancies lower than 1.5%.

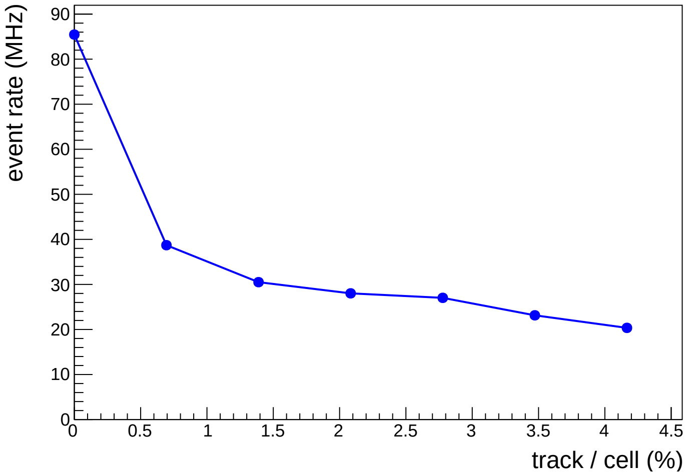


Figure 8.10: Event rate of 2D Engines.

Occupancy (%)	Event rate (MHz)
0.00	85.46
0.69	38.71
1.39	30.50
2.08	28.02
2.78	27.02
3.47	23.14
4.17	20.36

Table 8.2: Event rate of 2D Engines.

Counting how many hits are used by each engine to perform tracking is possible to estimate if they are fast enough to process events in real-time. Figure 8.11 shows the average number of hits within the engine distance search. On average the busiest engine processes 6 hits. Since it should be able to process 3 hits per clock cycle, at a clock frequency of 300 MHz, it should process up to 30 hits keeping an event rate higher than 30 MHz. In practise some factor could slow down the engine, but there is enough leeway to say that the engine is fast enough to perform VELO tracking.

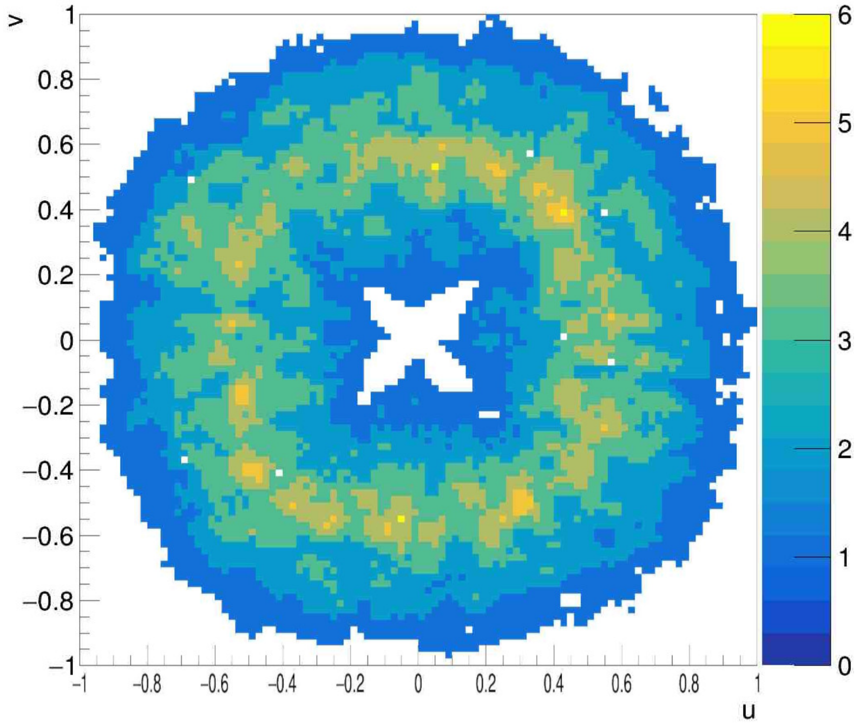


Figure 8.11: Average number of hits within the engine distance search from VELO “Artificial Retina” simulation. Each bin is a cell in the track parameters space.

The results described in this chapter and the previous, taken together, demonstrate the feasibility of building the “Artificial Retina” demonstrator according to the specifications required to ensure reliable operation at the LHC crossing rate, and provide all the necessary firmware components. The final step will be to assemble the needed number of boards (not yet available) and integrate them in the testbed setup environment.

However, the raw data produced by the VELO detector are not in the format of hit coordinates that “Artificial Retina” expects, and therefore require to be preprocessed before use. In the next Chapter, I will describe the system that performs this preprocessing. This is the first building block of the “Artificial Retina” system that has been fully completed, and it is available for use not just by the testbed demonstrator, but also by the HLT tracking of the VELO that is about to take physics data in the imminent Run 3 of LHCb.

Chapter 9

VELO clustering on FPGA

The “Artificial Retina”, like any other tracking algorithm, requires the coordinates of the hits on the detector to perform tracking. The VELO does not produce exactly this information, but the coordinates of active pixel within an event. It is a subtle difference that produce an huge impact on the tracking performance: a particle that hits a detector layer could activate multiple adjacent pixels, if the tracking algorithm threats each pixel as a different hit, the number of hits combination that it must resolve increases exponentially. Moreover multiple tracks will be reconstructed from different pixels activated by a single track. The tracking algorithm could discard these clone tracks, but this step requires extra processing time. For this reason before tracking contiguous active pixels are grouped in a single cluster. Find cluster on a 2D detector at LHC rates is a not trivial job, and there are not many publication about it [82].

HLT1 was supposed to perform the clustering of the VELO, using $\sim 17\%$ of its computing resources. Working before the trigger the “Artificial Retina” needs to perform this task on its own. Implementing this task on FPGA reduces the amount of hardware resources available for tracking. However, since clusters group contiguous pixels, a cluster is completely contained within a VELO sensor, and clustering do not require to share information between different boards. Therefore, if the free FPGA resources are enough, this task could be implemented also in the Readout Boards. This solution is interesting because it allows to dedicate all the Tracking Boards resource to the tracking, but it also allows to provides directly clusters to HLT, relieving the trigger farm workload.

I conceived and developed a FPGA-based clustering algorithm, capable of performing this task in real time at 30 MHz event rate using a modest amount of hardware resources. This algorithm includes specific features that are tailored for the VELO detector. However, the structure and its building blocks are general and can be easily applied to any silicon pixel detector.

VELO data are read as aggregated groups of 4×2 pixels, named SuperPixels (SPs). A SP carry the information of which of the eight pixels was active, its coordinates, and also if any neighbouring SP is active: a SP is flagged as isolated if none of its eight SP neighbours has any active pixel. This information is exploited by both HLT and FPGA algorithms for optimising the performance of the cluster reconstruction process [83, 84], allowing the implementation of a much faster algorithm for isolated SPs, that account for about 53% of the total number of SPs.

The sizes of clusters created by individual charged particles crossing VELO layers are typically rather small ($1-4$ pixels in 96% of cases), with larger clusters being the product of merged hits or secondary emissions (δ-rays, etc.). Figure 9.1 shows the distribution of cluster sizes as predicted by the LHCb Monte Carlo Simulation. This cluster property suggests to optimise the FPGA clustering algorithm for reconstructing small cluster.

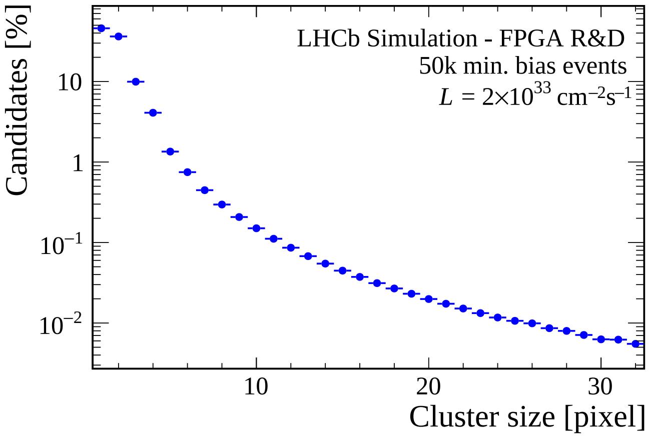


Figure 9.1: Distribution of cluster size in units of number of pixels.

For a VELO sensor, on average, there are $\sim$ 7 clusters per event. A sensor contains $\sim$ 197k pixels, therefore the average detector occupancy is low. Even considering that the occupancy is not uniform, for the regions closest to the beam pipe, it is expected to be around 0.125% [59]. This suggests to not implement a single sensor-sized matrix in favour of smaller matrices that map dynamically different sensor areas. That approach allows to reduce hardware resources usage.

9.1 The clustering algorithm

The structure of the VELO output, grouped in SP, leads to a natural distinction between clusters that are completely contained within a single SP, and clusters spanning two or more SPs. The two cases need very different treatment, and is therefore convenient to explicitly divide the processing in separate sections, for best efficiency.

Isolated SPs can only take a limited numbers of active pixel configuration: 2^8, and they completely contains the corresponding cluster. The cluster centre of mass is calculated for each configuration and stored in a LUT. The isolated SP is directly resolved into clusters retrieving the LUT entry corresponding to the SP pixel configuration. A 4×2 SP can contain up to two clusters. The LUT contains information about, and resolves both clusters. This LUT-based reconstruction allows an extremely fast processing of isolated SPs, with a very limited amount of logic resources. This is particularly relevant since isolated SPs are about 53% of the total for each bunch crossing.

The algorithm for non-isolated SPs requires, instead, the parallel processing of an ensemble of SPs. Cluster are resolved by the interaction of the cells of a pixel matrix. Like in the "Artificial Retina" Max Finder, each cell reads the state of the neighbours cells determining the position of the cluster. Pixel matrices can contain up to nine contiguous SPs, organized in three rows and three columns, for a total size of 6×12 pixels. Differently from "Artificial Retina" matrices that maps a fixed region of track parameters space, the position of a matrix in the sensor is not fixed a priori, but the first arriving SP fills the center of an empty matrix and determines the physical location of the matrix inside the VELO, as well as the set of coordinates of the other SPs that can fill it. Matrices are arranged in a chain. The SPs move along the chain. If a SP belongs to a matrix it fills it, otherwise it moves forward checking the next available matrix of the chain or filling the centre of an empty one. An explanatory graphical illustration of such mechanism is shown in Figure 9.2.

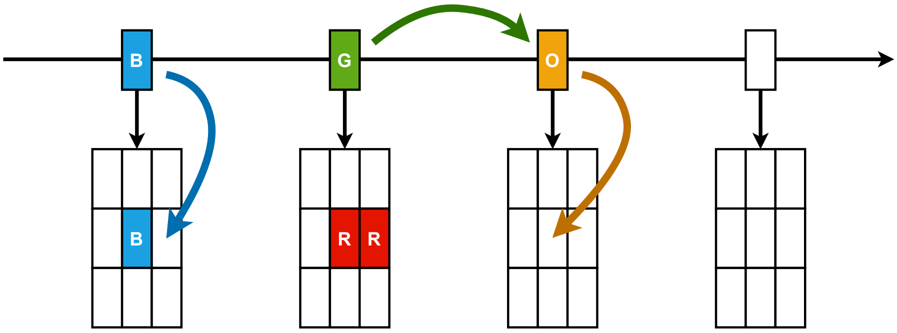


Figure 9.2: Sketch of the matrix filling mechanism with non-isolated SPs. SPs with same color (label) are neighbours with active pixels. The blue SP (B) fills the first matrix in the line that is already populated with one of its neighbours. The green SP (G) does not belong to any of the already populated matrices, so it moves forward. The orange SP (O) has reached an non-initialised matrix, so it fills the centre.

After the end of the event is reached and all SPs have been loaded in the matrices, the cluster finding can start: each pixel of each matrix, in parallel, checks if the neighbouring pixels match some pre-determined patterns (Fig. 9.3).

If one of the patterns is matched, the cluster candidate is recognised in the 3×3 grid (green pixels in figure). The "L" shaped sequence of inactive pixels is needed to ensure separation of the cluster from surrounding active pixels. The left configuration, with one active pixel surrounded by not active pixels on two sides, is the most natural one, but does not recognise some kind of clusters. The right one, with two active pixels in diagonal, allows to recover the cluster rejected from the former configuration. The combination of these two patterns allows to recognise cluster with an high-efficiency. Since the majority of clusters (> 96%) are made of up to four pixels the 3×3 grid is

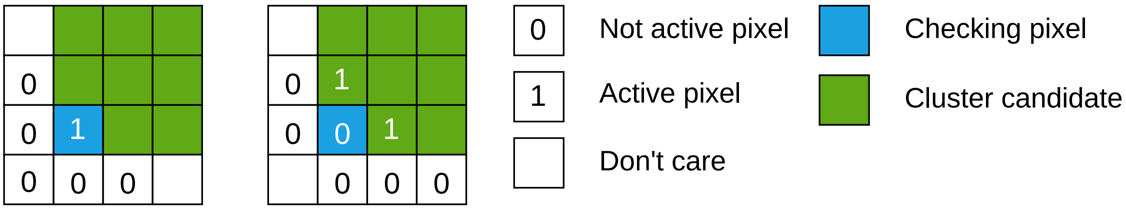


Figure 9.3: Patterns that generate a cluster.

sufficient to contain and resolve exactly the majority of cluster.

As for isolated SPs, the centre of mass of the 3×3 cluster candidate is determined by a LUT. The absolute position of the cluster candidate is obtained as a sum of three vectors of coordinates: the position of the matrix with respect to the detector, the position checking-pixel with respect to the matrix, and the position of the reconstructed cluster with respect to the checking pixel.

The algorithm has several parameters that can be tuned to optimise its performance in terms of speed, efficiency and quality of the reconstruction: the size of the matrix, the shape of the patterns, and the size of the cluster candidate. I tuned all these parameters to balance the physics performance of the algorithm and the amount of hardware resources. An interesting parameter is the number of matrices. In a FPGA-based implementation, their number cannot be dynamically adjusted to cope with events with a higher number of (non-isolated) SPs, and an appropriate fixed value must be chosen. Overflow should of course be avoided, but there is also a desire not to make this number too large, in order to avoid consuming too many of the precious FPGA logic resources. Studying the distribution of the number of SPs with neighbours, obtained by the official LHCb Run 3 simulation, I chose to instantiate 20 matrices for each sensor, as less of 0.1% of events exceed this value. In those cases, some SPs will travel through the entire chain without filling any matrices. This obviously rises the question of what to do with those overflows. Instead of throwing them away, I decided to extract a partial information, by processing them as if they were isolated. This has the consequence of occasionally splitting some cluster in two closely spaced hits, but without losing any of them; and this can be argued to be only mildly damaging to track reconstruction quality.

9.2 Physics performances

Before this reconstruction methodology can be proposed for actual physics data taking, it is mandatory to carefully check that the approximations introduced to fit within FPGAs limitations do not significantly affect the physics results.

To this purpose, I performed extensive studies of the physics performance obtained by using FPGA clusters as an input to the tracking algorithms, and carried out several comparisons with the standard CPU-based algorithms, that enjoy a much greater freedom from hardware-imposed limitations.

The key differences between the two implementations that can affect physics performances are: the matching pattern mechanism, the cluster candidate dimensions (limited to 3 × 3 pixels), and the matrix filling scheme. These differences can lead to partial cluster reconstruction, cluster splitting within the same matrix or clustering splitting between different matrices.

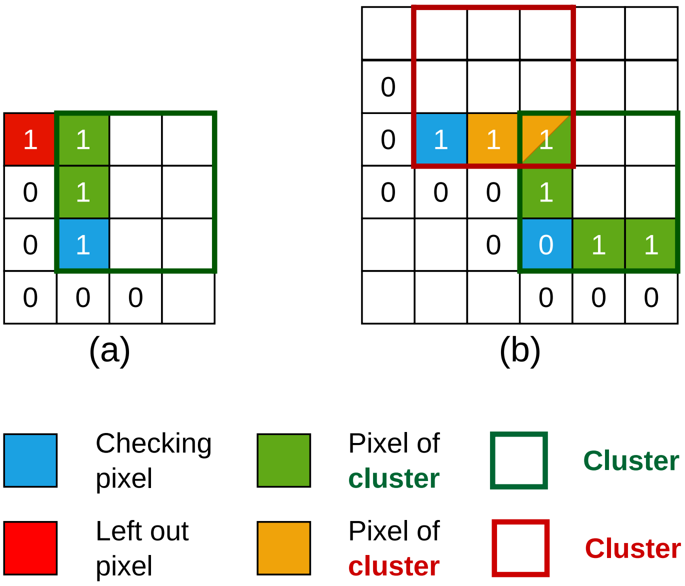


Figure 9.4: Example cases of undesirable behaviour of the FPGA algorithm. (a) partial cluster reconstruction: the pixel shown in red is left out of the cluster (b) cluster splitting: the pixels within the red and green boundaries are reconstructed separately (with a common pixel).

Figure 9.4(a) shows an example of partial cluster reconstruction, in which a pixel is left out from the cluster candidate. This causes a shift of the reconstructed particle hit position. The subsequent reconstruction may lead to a degradation of the track quality or a loss in efficiency if the associated track is not reconstructed at all.

Figure 9.4(b) shows an example of cluster splitting where the algorithm finds two clusters, with a pixel in common, from a contiguous group of active pixels. In this case, the peculiar cluster shape leads to two “checking pixels” recognising the state of neighbour pixels as corresponding to a valid pattern. Subsequently the tracking algorithm might reconstruct more than one track from a single one.

This matrix-based cluster reconstruction can also lead to cluster splitting between different matrices. When a cluster is not fully contained within a single matrix, a subset of its SPs may end up filling a different matrix, causing also in this case the reconstruction of multiple clusters from a single contiguous group of pixels. Figure 9.5 shows an example of clustering splitting between different matrices. The cluster in the example spans three different SPs. Assuming that the first arriving SP is the lower one, it fill the centre of a matrix. The central SP fill the same matrix. The upper one can not be contained in this matrix and it fill a centre of a second matrix. Since each matrix works knowing only the states of pixels inside it, two clusters are reconstructed.

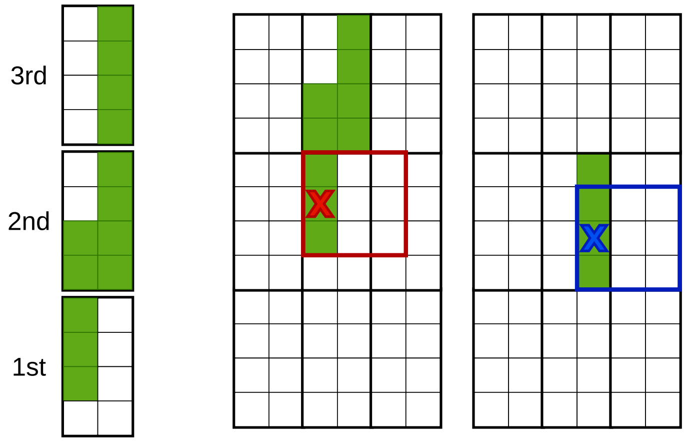


Figure 9.5: Peculiar clustering behaviours of the FPGA algorithm. Example of clustering splitting between different matrices.

To quantify the effects of these behaviours I wrote a bit-level C++ simulation of the FPGA clustering algorithm, that can be run within the official LHCb Simulation, and used it to perform the studies described in the next section.

9.2.1 The LHCb and clustering simulations

To better explain the studies that follow, it is useful to briefly summarise the structure of the LHCb simulation.

In LHCb the task of modelling the behaviour of the detector for the different type of events occurring in the experiment is carried out by two separate applications, called Gauss and Boole [85]. Gauss generates the initial particles and simulates their transport through the LHCb detector [86, 87], whilst Boole reproduces the different sub-detectors responses and their digitisation converting the data in the same format provided by the experiment electronics and the DAQ system. After digitisation, real data and Monte Carlo data follow the same path through trigger, reconstruction and analysis code. In LHCb the production of particles coming out of the primary *pp* collision of the LHC beams is handled by default with Pythia [88], a general purpose event generator, whilst the decay and time evolution of the produced particles is delegated to EvtGen package [89]. Lastly, in Gauss the simulation of the physics processes undergone by the particles travelling through the detector, is delegated to the Geant4 toolkit [90, 91]. Allen and Moore are respectively the HLT1 and HLT2 applications [61, 62]. They are responsible for filtering an input event of 30 MHz of visible collisions down to an output rate of around 100 kHz. For the purpose of data taking, Allen is executed on GPUs, but its code can also be compiled and executed on CPUs, for convenience in carrying out offline performances studies.

All the LHCb applications are customisable by choosing and configuring the set of algorithms to execute in a given sequence. I added to the LHCb applications a set of new algorithms, reproducing the FPGA procedure of cluster reconstruction, and the code needed to readout and use the results they produce. Running these algorithms together with the standards sequences, allowed me to directly compare the reconstruction performance of Allen and Moore using CPU-based and FPGA-based clusters, using the standard LHCb monitor tools.

The core algorithm of the clustering simulation is `VPRetinaClusterCreator`. It reproduces all the logical operations performed by the FPGA algorithm, producing RetinaClusters (RCs) from SPs. The RCs are represented as 32-bit words, packing the coordinates of the cluster plus some additional information. Great care was taken to ensure a perfect correspondence between the RCs produced by the C++ simulation and the real FPGA operation. An exact correspondence is needed not only to perform reliable performances studies, but also for use as a debugging tool for the FPGA clustering firmware. Indeed, during tests, the clusters produced by the FPGA are compared bit a bit to the simulation clusters, highlighting coding error in the simulation or the FPGA design. I added this algorithm to the default sequence of Boole. Thus, when the LHCb collaboration produce new Monte Carlo (MC) samples, they will contain also the RCs and they can be used by Allen and Moore. Alternatively, I wrote the Low Level Accelerator Application (`LLAApp`) that adds RCs to old MC samples. It is useful to perform tests with MC used by the collaboration as performances benchmarks, indeed these samples were produced before the introduction of `VPRetinaClusterCreator` in the Moore sequence and did not contain RCs.

Allen and Moore have then been provided with an algorithm (`VPRetinaClusterDecoder`) to decode the RCs and produce high-level cluster objects in the format usable by other algorithms.

Another algorithm, `VPRetinaFullClustering` allows linking reconstructed tracks to MC tracks for efficiency and resolution tests. Similarly to `VPRetinaClusterCreator`, it reproduces all the logical operations performed by the FPGA algorithm, but produces directly high-level objects with additional information respect to the objects created by `VPRetinaClusterDecoder`. The `VeloClusterTrackingSIMD` is the baseline VELO tracking algorithm. I added a template for decoding RetinaClusters and perform tracking with them.

Using the software tools described above, I performed comparisons between CPU-FPGA algorithms on a sample of 50k generic inelastic events (minimum-bias), at the conditions expected for Run 3 data tacking: centre of mass energy $\sqrt{s} = 14$ TeV and luminosity $\mathcal{L} = 2 \times 10^{33}$ cm^{-2} s^{-1}.

9.2.2 Cluster

Clustering efficiency

The clustering efficiency is defined as

$$\epsilon \equiv \frac{N_{\text{linked MC hits}}}{N_{\text{MC hits}}},$$

where $N_{\text{linked MC hits}}$ is the number of MC hits with a linked reconstructed cluster, and $N_{\text{MC hits}}$ is the total number of MC hits. MC hits carry a set of information about the interaction of a particle with the detector material, like the position of the intersection and the amount of energy released. A LHCb simulation algorithm determines whether a cluster contains information related to a MC hit, linking the two objects.

When selecting only MC hits of VELO-reconstructible tracks, *i.e.* tracks with at least 3 MC hits on VELO, the clustering efficiency turns out as 99.92%. When considering all the MC hits, including tracks with to few hits to be reconstructed by HLT, the FPGA clustering efficiency is still quite high, with a value of 99.82%. Figure 9.6 shows the clustering efficiency, for both the FPGA and CPU clustering algorithms, as a function of the radius and module number of the hit and of η, ϕ, p and p_{T} of the corresponding track. FPGA clustering efficiencies is shown considering all tracks (including those from non-reconstructible VELO tracks) and selecting only clusters from VELO reconstructible tracks.

In order to better highlight possible efficiency dependencies on the kinematic variables, I also plotted clustering inefficiency as a function of the same variables (Fig. 9.7). While not precisely uniform, this inefficiency on cluster from VELO-reconstructible tracks does not show any "hot spots" in particular places, varying around 0.08% over most of the space; the largest effects are observed at low momenta, where the maximum inefficiency reaches 0.17%. An exception to that is a peak of large inefficiency at pseudorapidity close to zero; where, however, the distribution of MC tracks has very few entries. The origin of this effect is easily identified: here tracks graze VELO sensors at a very low angles, producing very spread out clusters. For this reason, these hits are unlikely to be accurately measured whatever the clustering algorithm. Moreover, the LHCb tracking volume covers the pseudorapidity range $2 < \eta < 5$, leaving out this region with higher inefficiencies. The only contribution to tracking of clusters in this region is in the reconstruction of the primary vertex position. I will show in Section 9.2.3 that the primary vertex reconstruction quality is not affected.

Cluster residuals

I studied the quality of the reconstructed hit positions using cluster residuals, defined as the distance between the reconstructed cluster centroid and the position of the particle hit associated with it. Figure 9.8 shows a comparison between CPU and FPGA cluster residual distributions.

Differences between CPU and FPGA distributions are only visible in logarithmic scale. When considering all types of clusters, including those from non-reconstructible tracks, the FPGA distribution shows higher tails starting at two orders of magnitude below the peak. Those tails can be tracked down to large clusters, that are reconstructed with lower precision by FPGA clustering. However, when selecting only clusters from VELO-reconstructible tracks, the difference becomes negligible, and is actually in favor of FPGA clustering.

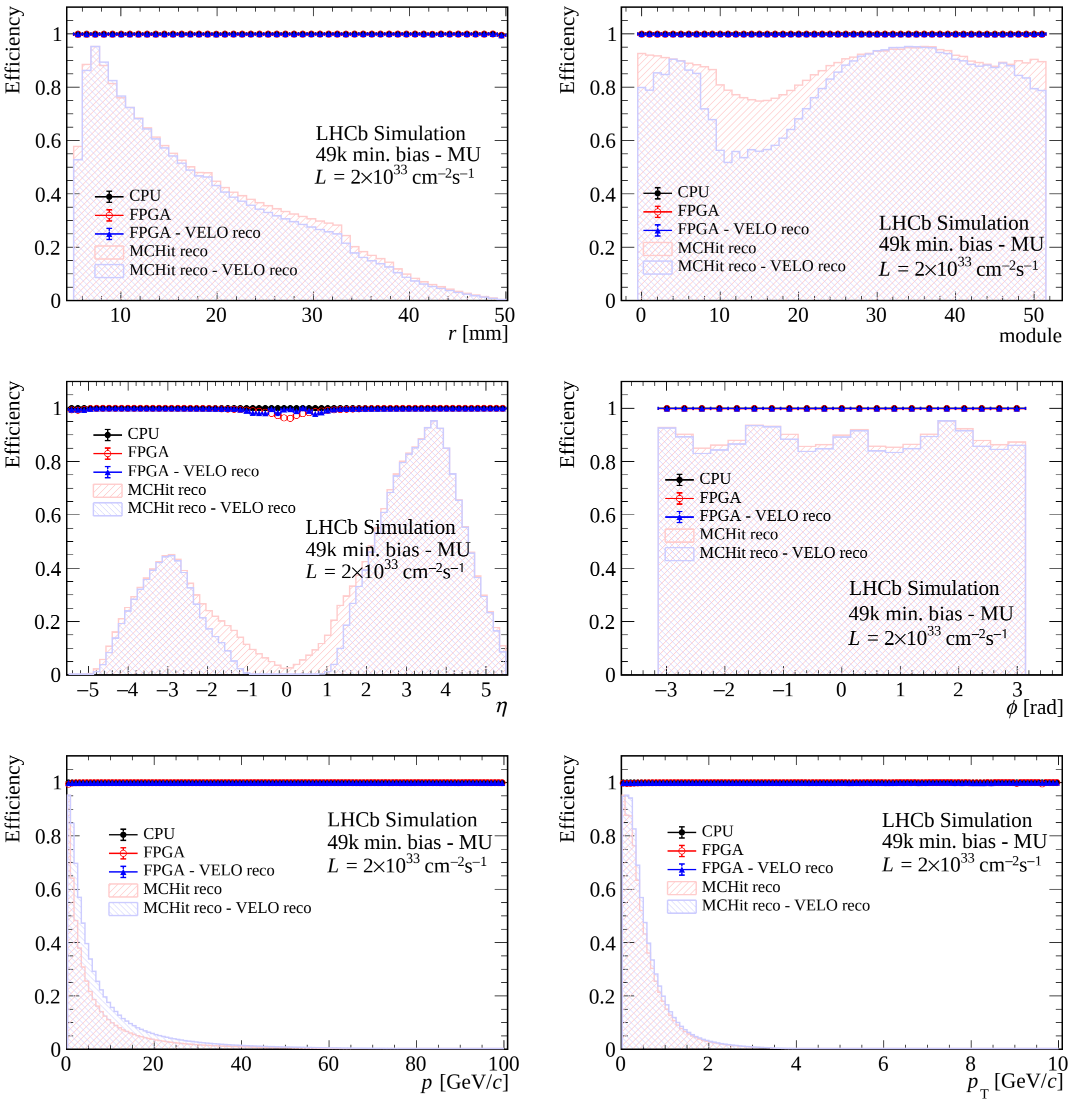


Figure 9.6: Comparison of the clustering efficiency of all clusters and clusters from VELO reconstructible tracks when using the FPGA and CPU-based clusterings, as a function of various variables. The red and blue histograms show the distribution of MC hits from all types of tracks and selecting only VELO reconstructible tracks, respectively.

Finally I plotted the cluster residual distributions of all CPU clusters and the of CPU clusters not reconstructed by the FPGA (Fig. 9.9). The reconstruction quality of clusters missed by the FPGA is clearly worse, showing high residuals.

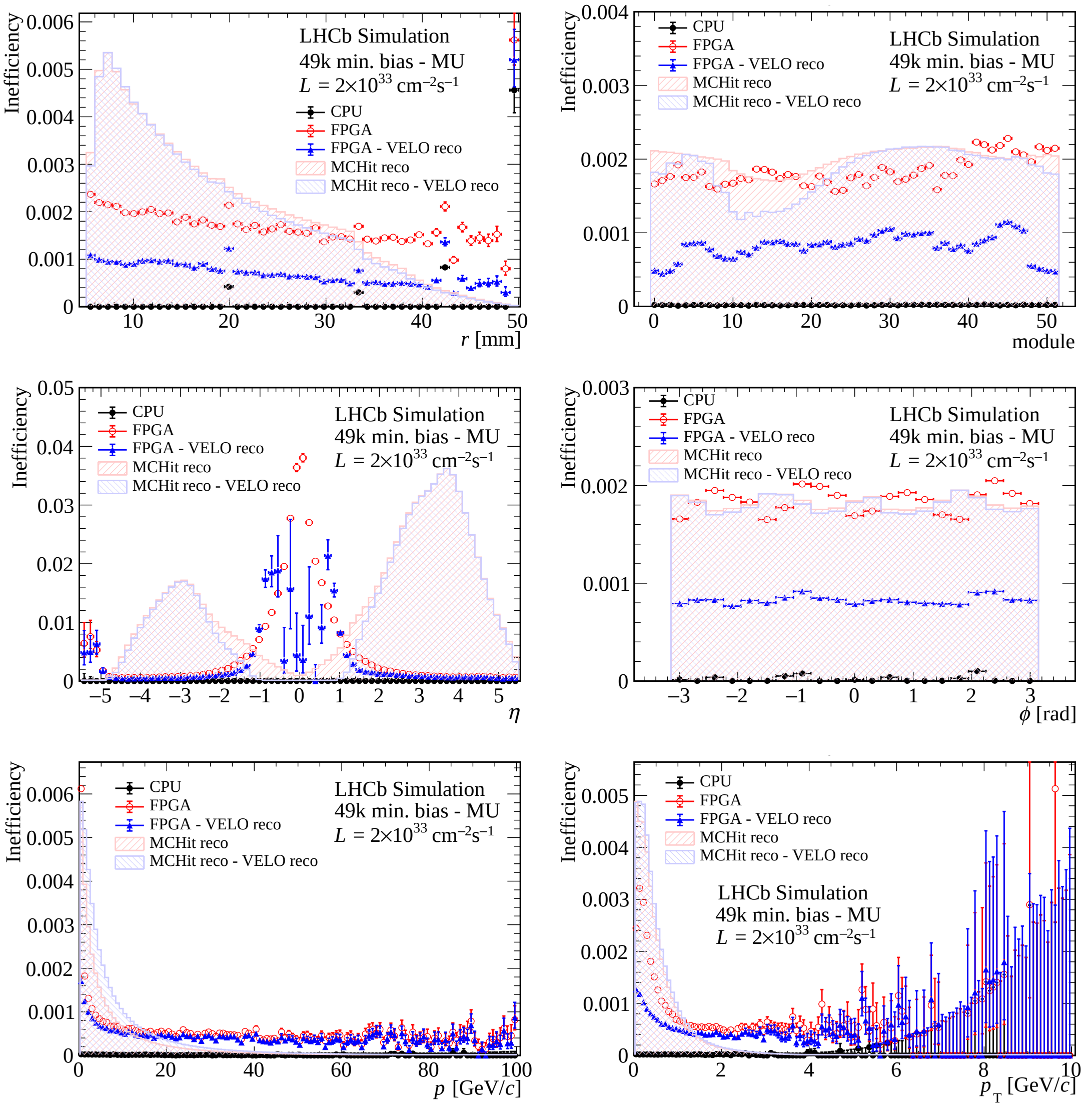


Figure 9.7: Comparison of the clustering inefficiency of all clusters and clusters from VELO reconstructible tracks when using the FPGA and CPU-based clusterings, as a function of various variables. The red and blue histograms show the distribution of MC hits from all types of tracks and selecting only VELO reconstructible tracks, respectively.

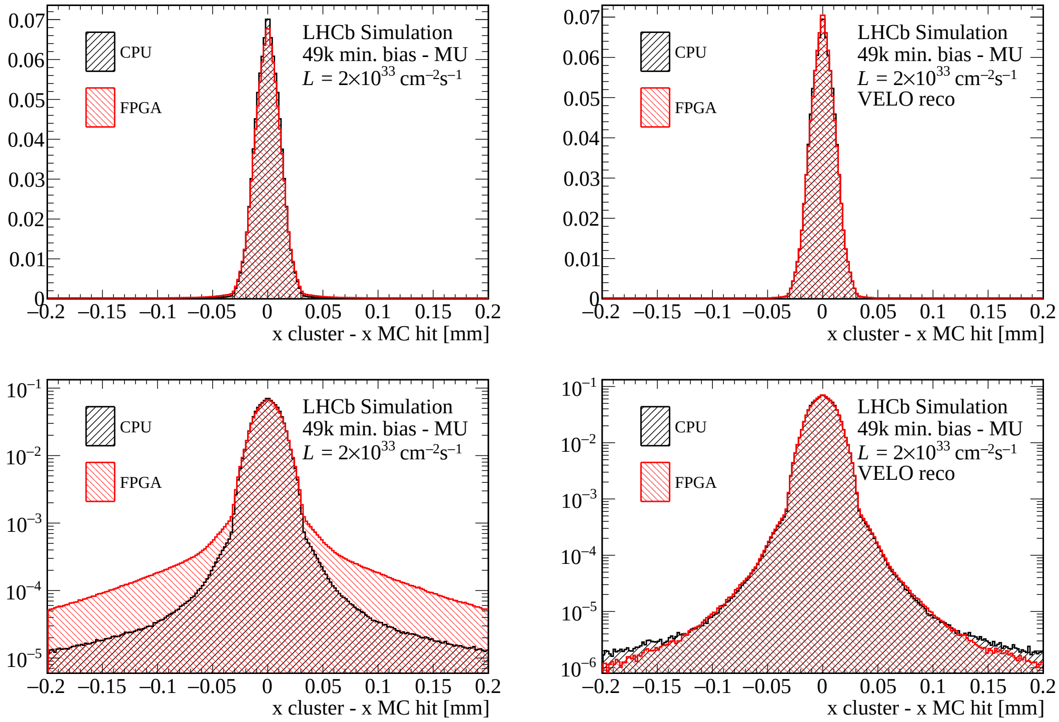


Figure 9.8: Comparison of the normalised distributions of the cluster residuals (left) for all clusters, including those produced by non-reconstructible tracks and (right) for clusters produced by VELO reconstructible tracks. Top (bottom) plots are displayed in linear (logarithmic) scale.

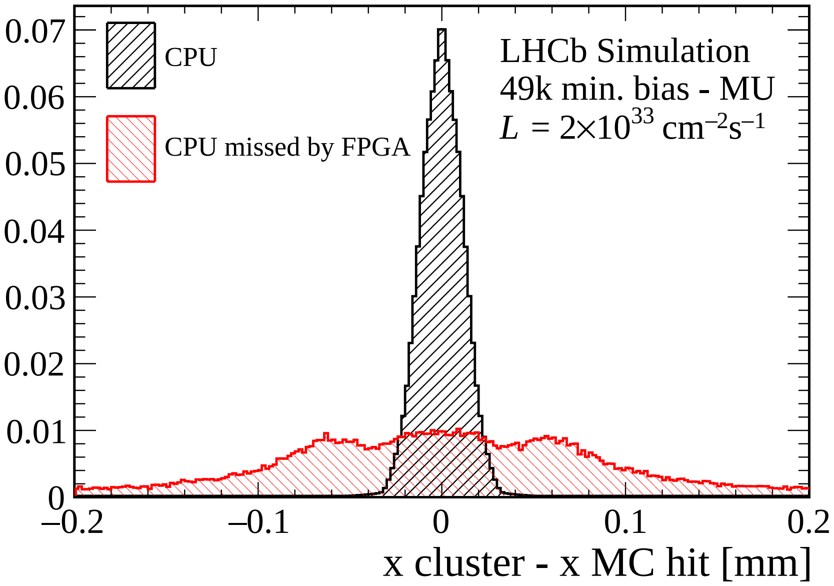


Figure 9.9: Cluster residual distributions comparing all CPU clusters with CPU clusters not reconstructed by FPGA.

9.2.3 Tracking

Tracking efficiency

In LHCb tracks are categorised according to the detectors in which they released hits. For most LHCb analyses, including my own analysis described in the first part of this thesis, the most relevant are the *long* tracks, that run through all tracking sub-detectors. More precisely, long tracks must have at least 3 hits on VELO and 1 x and 1 stereo hit in each SciFi station. Others interesting tracks for the VELO detector are the VELO tracks with at least 3 hits on this detector.

The tracking efficiency for a specific kind of tracks is defined as

$$\epsilon \equiv \frac{N_{\text{matched MC track}}}{N_{\text{MC track}}},$$

where $N_{\text{matched MC track}}$ is the number of MC tacks that are matched to a reconstructed track, and $N_{\text{MC track}}$ is the total number of MC tracks of that kind. A reconstructed track matches a MC track if it is reconstructed using at least 70% of hits of that MC track.

Table 9.1 shows a comparison between HLT1 tracking efficiency using CPU and FPGA clusters. It shows also the rates of clone and ghost tracks. Any additional reconstructed track matching the same MC track is a clone tracks, whereas a ghost track is a reconstructed track not associated to any MC track.

Track type	Quantity	CPU cluster	FPGA cluster
VELO tracks	efficiency	98.254% ± 0.007%	98.254% ± 0.007%
	clone	1.231% ± 0.006%	1.234% ± 0.006%
Long tracks	efficiency	99.252% ± 0.006%	99.252% ± 0.006%
	clone	0.806% ± 0.006%	0.806% ± 0.006%
	ghost	0.848% ± 0.003%	0.928% ± 0.003%

Table 9.1: Track reconstruction efficiency, clone and ghost track rates, comparing CPU and FPGA clustering algorithms on a 50k minimum bias MC sample.

Figure 9.10 and Figure 9.11 shows the tracking efficiency as a function of p, p_{T}, ϕ, η of the track, and as a function of the number of PVs in the event for VELO and long tracks respectively. In conclusion, the tracking efficiencies obtained with the FPGA clustering are practically indistinguishable from those that use the full-fledged CPU clustering algorithm.

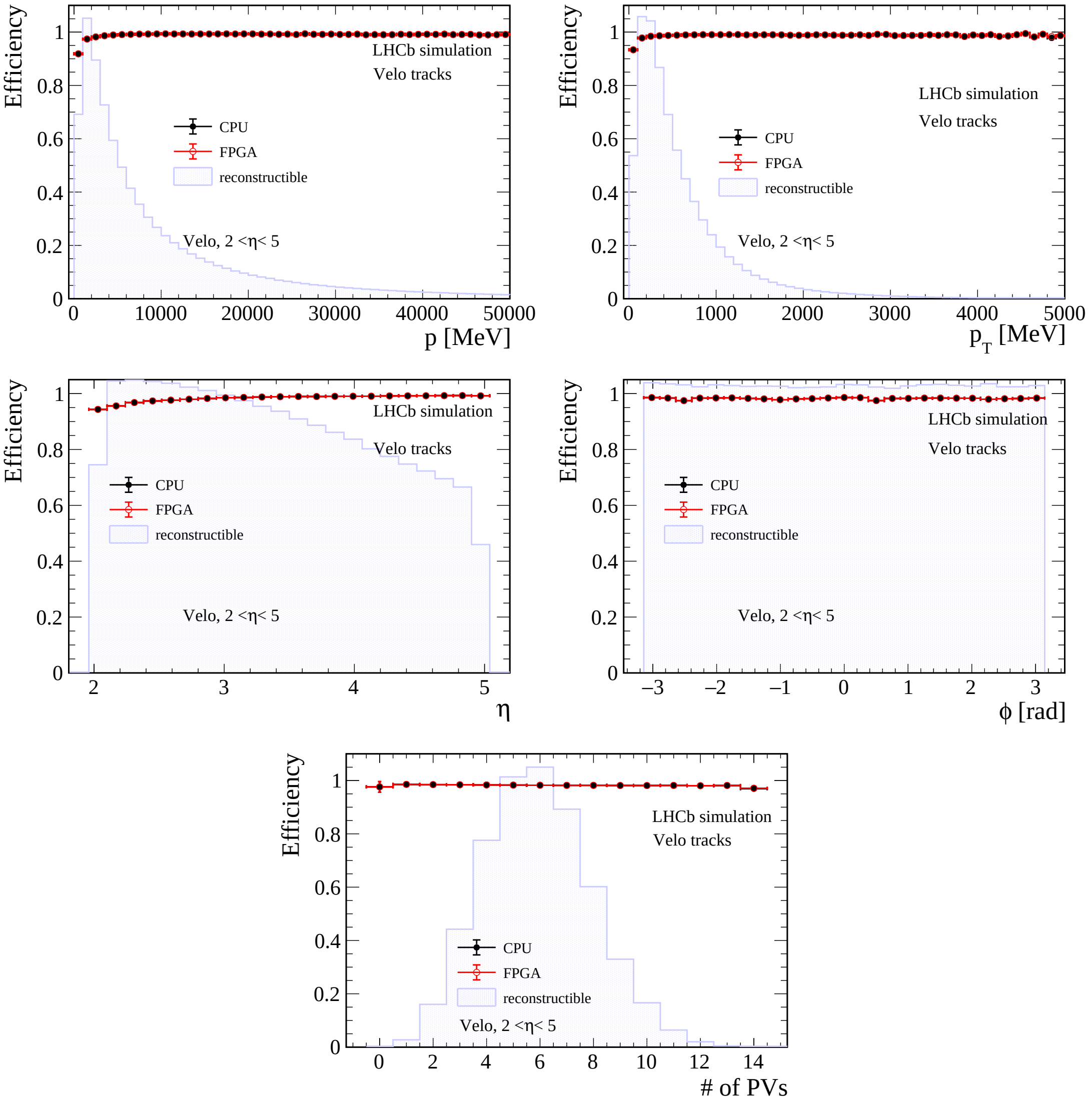


Figure 9.10: Comparison of the reconstruction efficiency of all reconstructible VELO tracks when using the FPGA and CPU clusters, as a function of various kinematic variables.

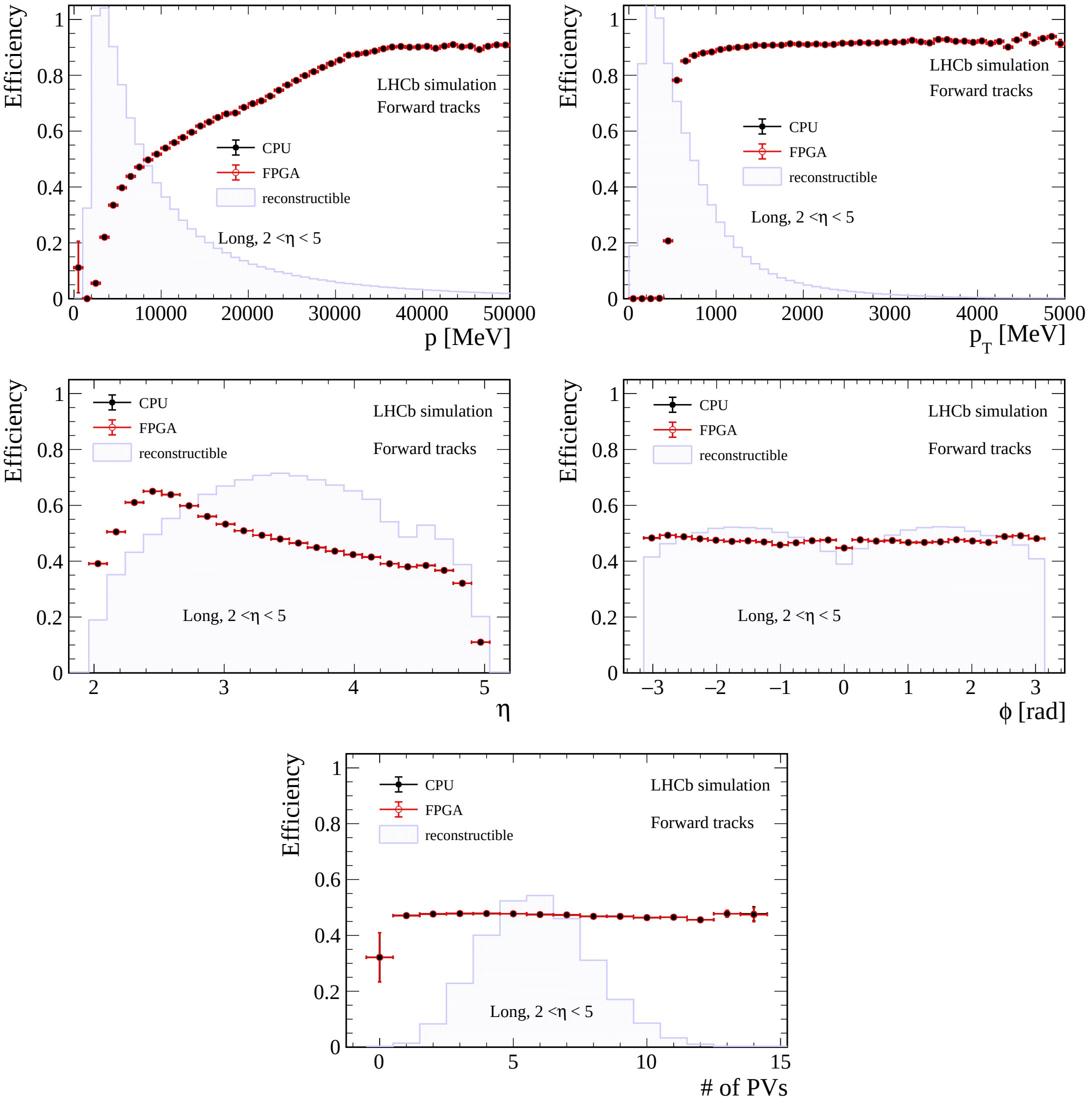


Figure 9.11: Comparison of the reconstruction efficiency of all reconstructible long tracks when using the FPGA and CPU clusters, as a function of various kinematic variables.

The ghost rate of long tracks, defined as the fraction of reconstructed ghost tracks over all reconstructed tracks, is displayed in Figure 9.12 as a function of the p, p_{T}, η of the tracks and as a function of the number of PVs. Also the ghost rate turns out to be indistinguishable between the two clustering methods.

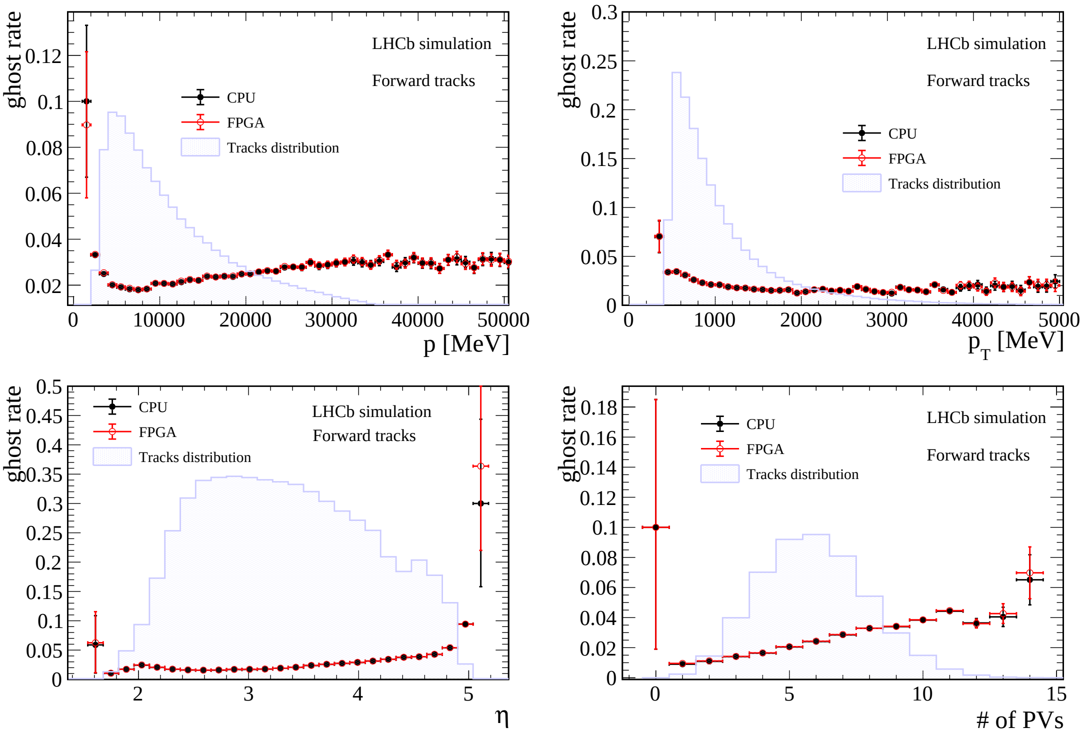


Figure 9.12: Comparison of the ghost probability of long tracks reconstructed using the FPGA and CPU clusters, as a function of various kinematic variables.

Tracking resolution

I also studied the quality of the reconstructed tracks. In particular, the resolution on the IP with respect to the PV is checked using HLT1 VELO Kalman-fitted tracks. The resolution on IP_x and IP_y observables are analysed separately. The IP_x observable is defined as the x component of the vector linking the PV to the intersection of the track with the plane transverse to the z axis and passing through the PV,

$$\mathrm{IP}_x \equiv (x - x_{\mathrm{PV}}) - (z - z_{\mathrm{PV}})\frac{p_x}{p_z},$$

where all variables are referred to reconstructed quantities and (x, y, z) is the point of closest approach of the track to the PV (IP_y is the analogue of IP_x with the substitution $x \to y$). The resolutions of IP_x and IP_y, estimated using the sigma of a Gaussian function fitted to their distributions in the range $[-300, 300]$ µm, are displayed in bins of $1/p_{\mathrm{T}}$ and η in Figure 9.13. Also in this case, the performances of the CPU and FPGA clusterings are almost indistinguishable.

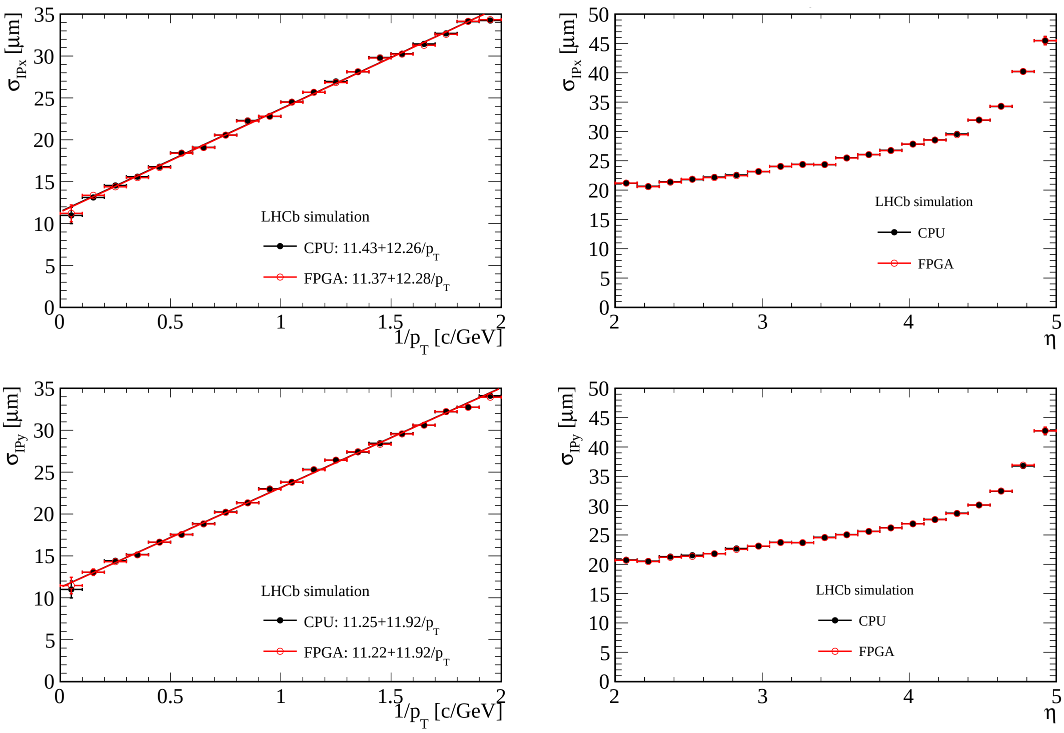


Figure 9.13: Comparison of the resolution of IP$_x$ (top) and IP$_y$ (bottom) of HLT1 Kalman-fitted VELO tracks when using the FPGA and CPU clusters, as a function of the inverse of the true transverse momentum (left) and as a function of the true pseudorapidity (right) of the track.

Figure 9.14 shows the relative resolution on the momentum magnitude for long tracks in the range $2 < \eta < 5$ as a function of the true momentum and pseudorapidity. The resolution is defined as the sigma of a Gaussian function fitted to the dp/p distribution in the range $[-10\%, 10\%]$ ($[-5\%, 5\%]$) for the p (η) observable. Also in this case the performances of the CPU and FPGA clusterings are almost indistinguishable.

Primary vertex reconstruction

The tracking performance has a direct impact on the reconstruction of the primary vertexs (PVs). As I mentioned before, vertex reconstruction makes use of some track categories that are not well covered by the analysis of the previous sections, and must be considered separately.

The PV reconstruction efficiency is defined as

$$\epsilon \equiv \frac{N_{\text{MC-matched}}}{N_{\text{MC-reconstructible}}},$$

where $N_{\text{MC-matched}}$ is the number of reconstructed PVs that are matched to a MC PV, and $N_{\text{MC-reconstructible}}$ is the number of MC PVs with at least four reconstructed VELO

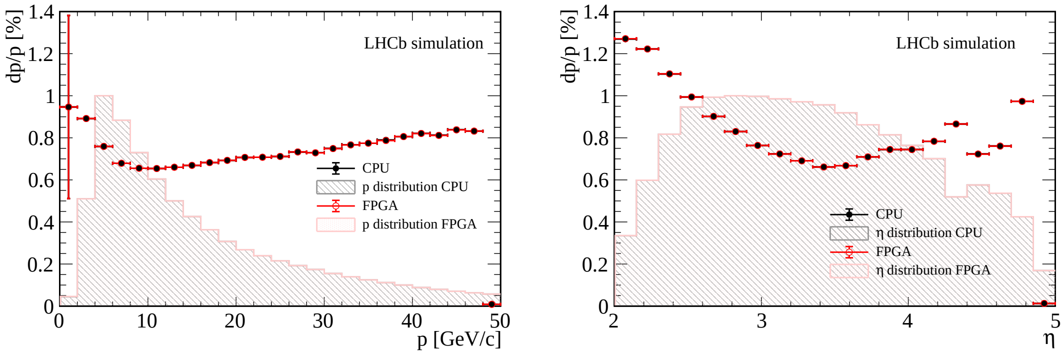


Figure 9.14: Comparison of the resolution on the momentum of long tracks when using the FPGA and CPU clusters, as a function of the true momentum (left) and as a function of the true pseudorapidity (right) of the tracks.

tracks. The MC matching is performed by distance, requiring that the reconstructed PV lies at a distance along the z axis less than 2 mm or $5\sigma(z_{\mathrm{PV}})$, whichever is less, from the MC PV, where $\sigma(z_{\mathrm{PV}})$ is the uncertainty of the reconstructed position of the PV along the z axis.

This efficiency is compared for the two clustering methods, both as a function of the number of reconstructed tracks associated to the corresponding MC PV and of the z coordinate of the MC PV. The results are nearly indistinguishable (Fig. 9.15). The reconstruction efficiency as a function of z_{PV} is rather flat in both cases, but both clustering methods exhibit a small drop in efficiency for values of z_{PV} between -60 and 0 mm.

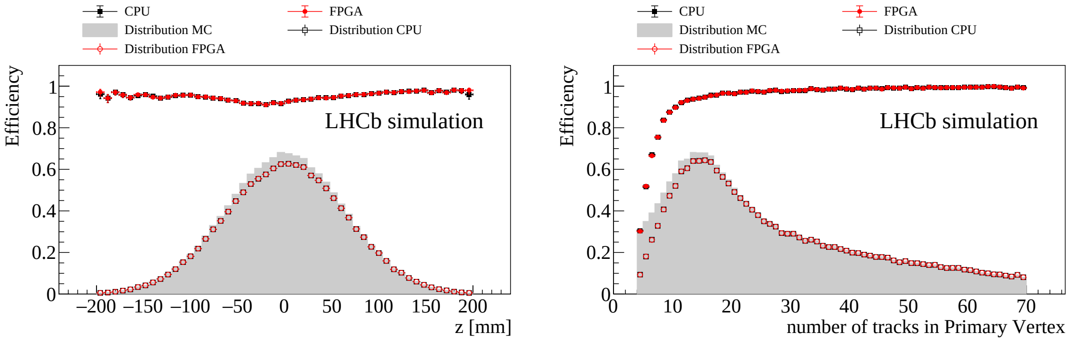


Figure 9.15: Comparison of the reconstruction efficiency of the PVs with the FPGA and CPU clusters, as a function of the true z position of the PV vertex (left) and of the number of reconstructed tracks of the PV (right).

The resolution and bias of the PV reconstruction is quantified along each coordinate axis, based on the distribution of the residuals of the reconstructed position of the PV minus the true one ($\Delta x \equiv x_{\mathrm{reconstructed}} - x_{\mathrm{MC}}$, *etc.*). To prevent the estimates from being confounded by the presence of few outliers far in the tails of these high-statistics distributions, I adopted a more robust procedure to measure resolutions than the raw root mean square (RMS). First, the RMS of the bulk of the distribution is estimated from the values of the 25th and 75th percentiles of the distribution. Second, a Gaussian fit is performed with a range limited to ±4 RMS around zero, and the sigma of this Gaussian is taken as a measure of the resolution. Figure 9.16 shows the results as a function of z_{PV} and of the number of reconstructed tracks of the PV. The performances of the CPU and FPGA-based clusterings turn out to be barely distinguishable. The resolutions along all the axes show a small knee in the region $-60 < z_{\mathrm{PV}} < 0\,\mathrm{mm}$ that is characterised by lower reconstruction efficiency (see Fig. 9.15).

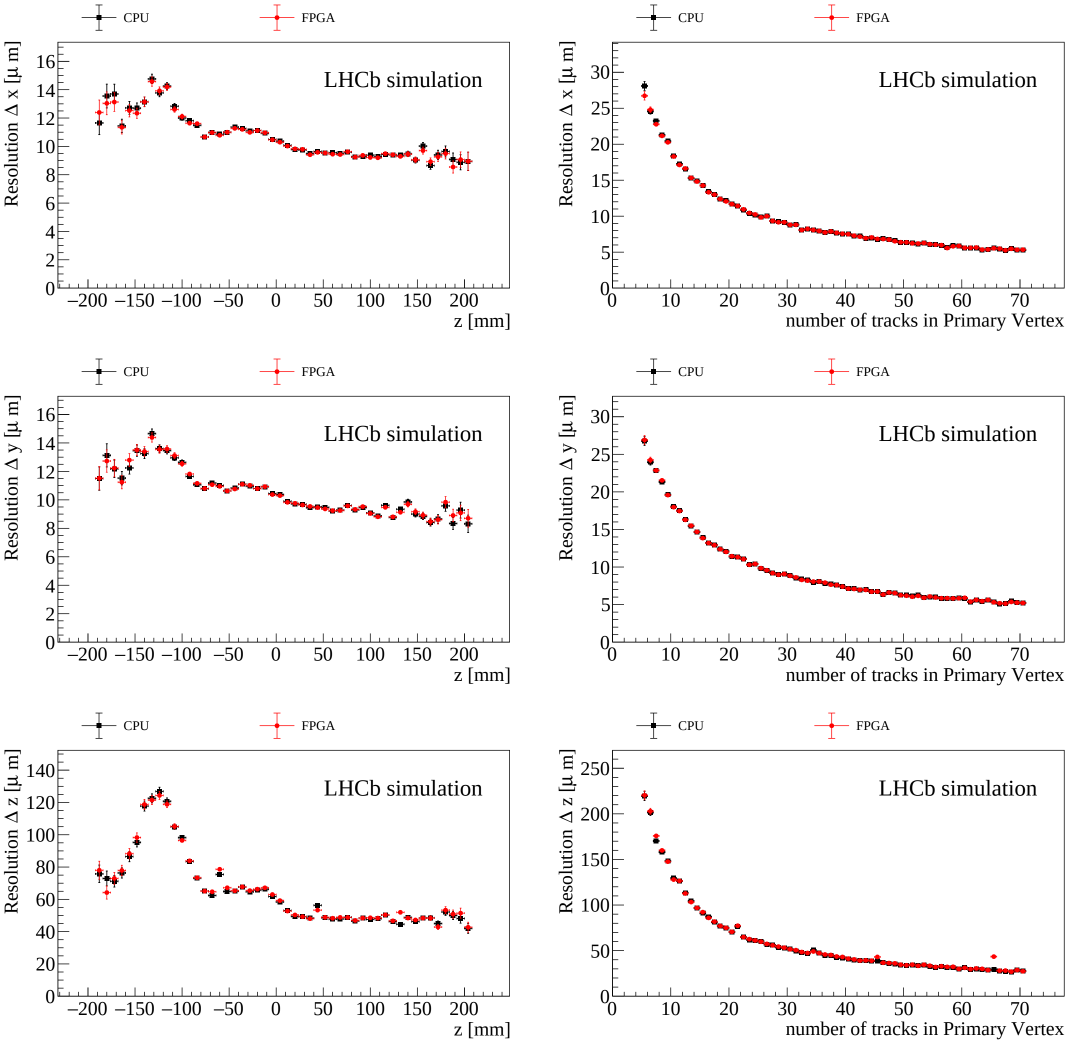


Figure 9.16: Comparison of the resolution in reconstructing the PVs with the FPGA and CPU clusters, as a function of the true z position of the PV vertex (left) and of the number of reconstructed tracks (right), for the x coordinate (top), the y coordinate (centre) and the z coordinate (bottom) of the PV.

9.2.4 Robustness to VELO occupancy

It should not be forgotten that the FPGA-based clustering particularly aims at a high-luminosity environment, when CPU- and GPU- based reconstruction will be too slow and expensive to perform.

It is therefore important to investigate the performances of this algorithm at detector occupancies that can be typical of future runs of LHCb. Being the clustering a local algorithm, the test was performed based on the official Run 3 simulation, looking at the clustering inefficiency as a function of local MC Hits density, and the tracking performances as a function of the total number of SPs per event, since at high luminosity the number of tracks, and then of SPs, increases.

To evaluate the clustering inefficiency as a function of the occupancy, for each VELO sensor, I filled a 2D-histogram with the positions of the MC Hits. The 2D-histogram is divided into $0.5\,\mathrm{mm} \times 0.5\,\mathrm{mm}$ bins. I define the local occupancy as the ratio of the number of counts in each bin to the area of the bin, divided by the total number of events reconstructed.

I filled two 1D-histograms in occupancy bins with the number of MC Hits and the number of MC Hits not linked to a cluster. The inefficiency is calculated as the ratio between the two 1D-histograms. Figure 9.17 shows a comparison between CPU and FPGA clustering inefficiencies as a function of the VELO occupancy. The FPGA inefficiency is plotted, both considering all MC Hits and selecting only MC Hits from VELO reconstructible tracks. It is important to note that clustering inefficiency does not show any significant tendency to increase when the local VELO occupancy is increased.

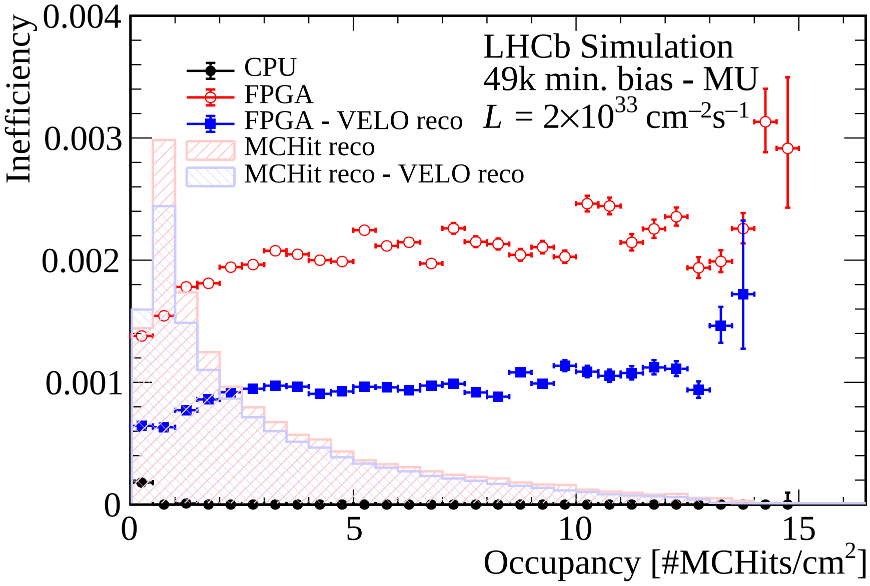


Figure 9.17: Clustering inefficiency as a function of the local VELO occupancy.

Tracking performances are instead studied as a function of the total number of SPs per event in the VELO detector. Figure 9.18 shows the distribution of the number of SPs per event, split into 4 equally populated quantiles. The rightmost quantile is further divided into two regions to investigate the effect of events with a high number of SPs on the reconstruction quality. Moving from the leftmost quantile to the tail on the right the number of SPs spans almost a factor of 5.

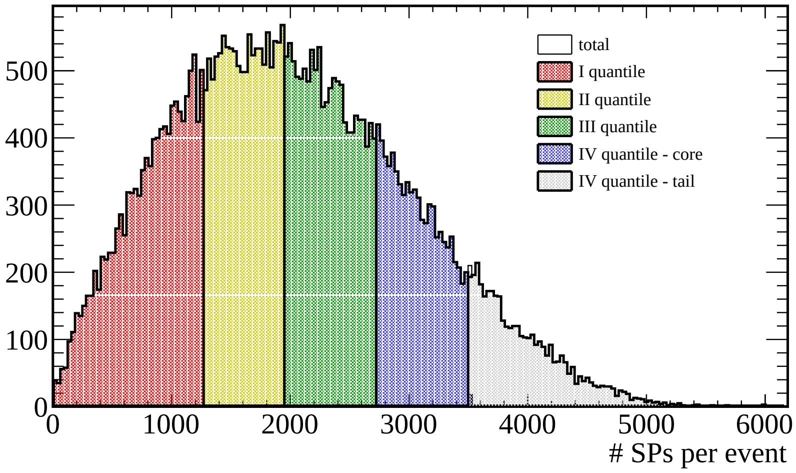


Figure 9.18: Distribution of the number of VELO SPs per event.

Figure 9.19 shows the ghost rate as a function of the number of SPs. Each point is centred on the average fraction of each of the five highlighted regions of Figure 9.18. CPU and FPGA ghost rates follow the same trend, with an absolute difference between the two staying below 0.1%.

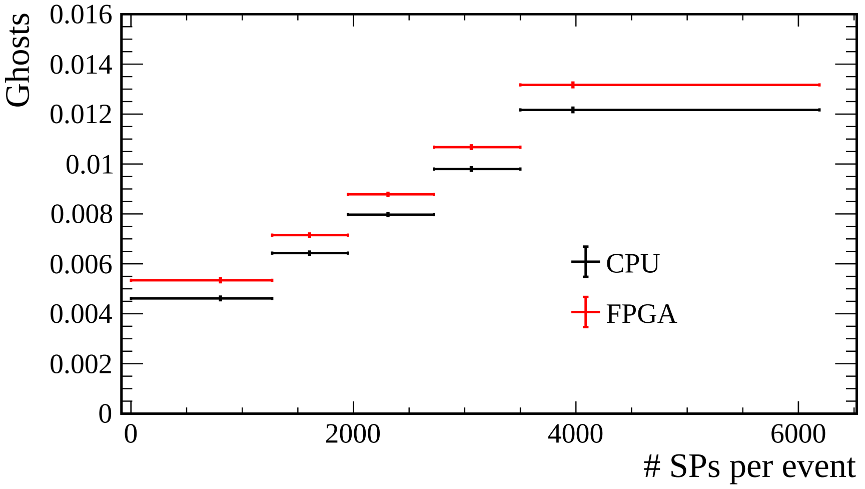


Figure 9.19: Ghost rate as a function of the number of SPs per event.

Figures 9.20 and 9.21 show tracking reconstruction efficiencies and clone rates for VELO (left) and long (right) tracks. All the represented quantities show permille-level changes spanning over the number of SPs per event distribution in Figure 9.18 (left). The FPGA clustering behaviour follows the CPU trend, within permille level differences. The small differences observed between FPGA and CPU clustering algorithms do not show any tendency to enlarge when the number of SPs is increased, neither for ghost rate, nor for clone rate or tracking efficiency.

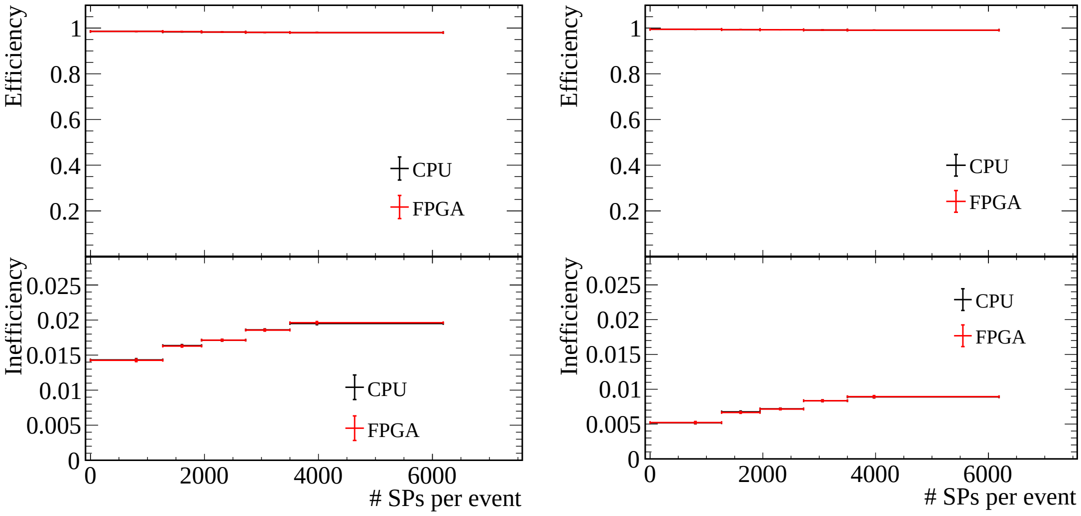


Figure 9.20: Efficiency (top) and inefficiency (bottom) for VELO tracks (left) and for long tracks (right) as a function of the number of SPs per event.

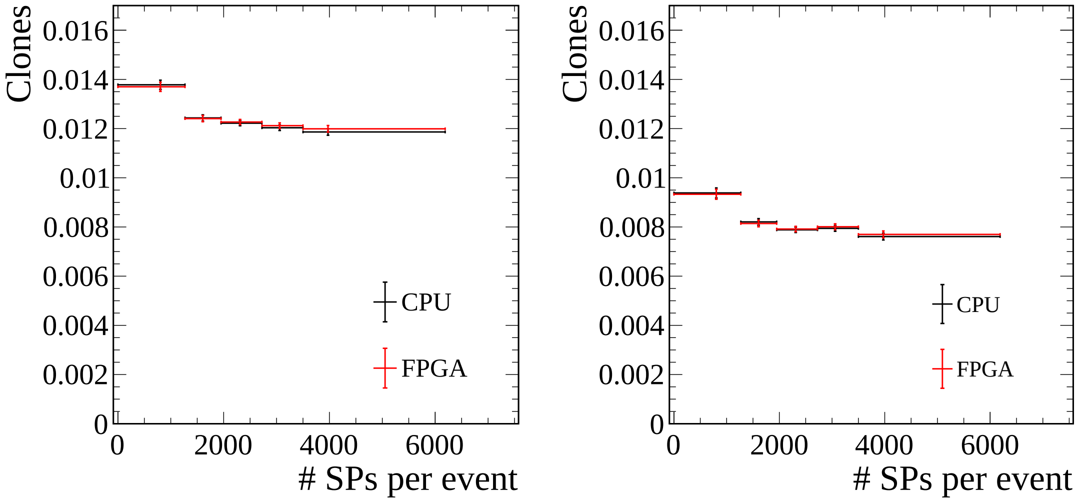


Figure 9.21: Clone rate for VELO tracks (left) and for long tracks (right) as a function of the number of SPs per event.

9.3 The in hardware implementation

The clustering design receives data with the packaging used in LHCb Readout Boards, so, in addition to the entities that clusterise isolated and non-isolated SPs, it includes some entities that manage the data stream. Figure 9.22 shows the main components of the clustering design and their connections. Starting from the input side (left side of the figure), a decoding stage splits data into separate streams, a couple of switches sends data to the appropriate cluster processing blocks. Reconstructed clusters are then encoded back to the appropriate output format.

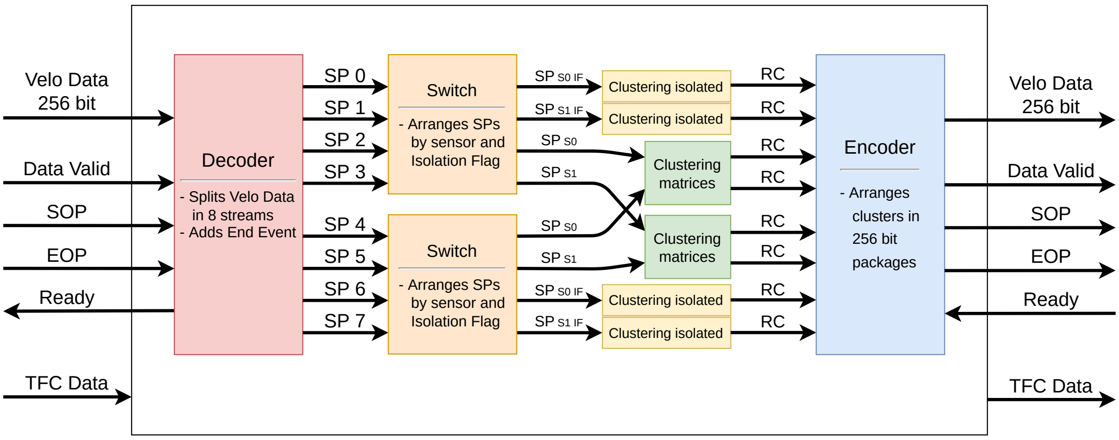


Figure 9.22: Structure of the clustering design.

9.3.1 Data format

SPs are stored in a 32-bit based data format (Fig. 9.23). Eight bits contains the states of the 4×2 pixels grid. The SP position inside the sensor. Given the SP geometry, 6 bit are needed to specify the SP row whereas 9 bit are required for the column. A data stream contains SPs coming from a couple of sensors [92], a bit identifies the sensor. One bit carries the isolation flag.

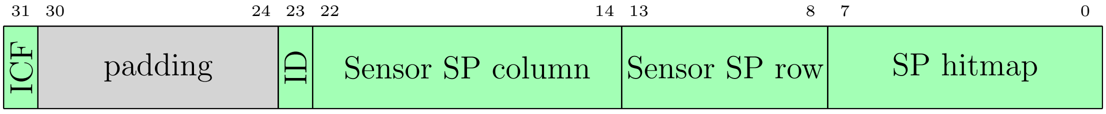


Figure 9.23: SuperPixel (SP) data format.

SPs are arranged in 256-bit words containing eight 32-bit words, a corresponding valid signal is also sent. Start Of Package (SOP) and End Of Package (EOP) signals are also provided. The SOP comes together with the first 256-bit word of the event, while the EOP with the last word of the event; if an event is empty or is made of only one word, SOP and EOP signals are simultaneously sent. The Ready signal is used like a ”not hold” signal to implement the back-pressure mechanism.

At the output side, there are the corresponding signals. Data words have the same size and carries clusters instead of SPs. Clusters are stored in a 32-bit based data format (Fig. 9.24), as well. 18 bit are required to specify the position of the pixel in which the cluster centroid falls: 8 bit for the row and 10 for the column. For clusters with more than one active pixel, the centroid position is calculated with a resolution finer than the pixel pitch. Four bits are used to specify the fractional parts of the cluster coordinates. Hence each cluster position is measured with a resolution of one-forth of a pixel. Similarly to SP data, one bit is used to identify the sensor. Eight additional bits are allocated to encode the cluster topology identifier and the reconstruction quality flags. The topology identifier encodes the full cluster topology that can be used for monitoring purposes. Reconstruction quality flags allows to distinguish between clusters reconstructed within matrices, clusters from isolated SPs, and clusters from SPs overflowing the maximum number of instantiated matrices and treated as isolated. The cluster word also contains self-contained and edge flags for clusters reconstructed through matrices: the former states whether a cluster is fully contained in the 3 × 3 pixel grid, the latter specifies if the pixel grid shares part of its edges with the matrix ones. These two flags are useful to identify partially reconstructed clusters and split clusters discussed in Section 9.2.

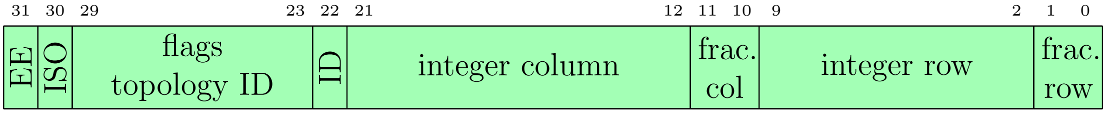


Figure 9.24: RetinaCluster (RC) data format.

9.3.2 Input side

Decoder

Input data come as 256-bit packets, containing 8 SPs each. In order to handle this stream, I wrote the decoder. The main function of the decoder is to convert the packets stream, with the SOP-EOP logic, in the same format used in the "Artificial Retina" system, with the End Event (EE) word for events separation. The clustering design is derived from the "Artificial Retina", so it share several entities and approaches.

Practically the decoder splits the 256-bit stream into 8 32-bit streams, allowing to elaborate each SP separately. At each SOP signal, before sending the corresponding SPs, the decoder sends to each line the EE word. The four least significant bits of an event counter are also included as data identifier in each EE word in order to track data flow and ensure synchronisation.

Switch

Since a cluster can not cross sensors borders, SPs that belong to different sensors are processed separately. Also isolated and non-isolated SPs are processed separately.

A switch arranges by sensor and by isolation flag, feeding corresponding cluster reconstructing blocks, accordingly.

As shown in Figure 9.22, I implemented two switching units. Each of them acts as a 4 to 4 switch, assuring that every input data can go to any of the four output streams, regardless of the origin input stream. The switch is directly derived from the "Artificial Retina" design, with the same Splitters and Mergers described in Section 7.3. However the clustering Splitter do not contain a LUT for storing the routing table, but it determines the output line according to the sensor ID or ICF bits. Figure 9.25 shows the switch structure.

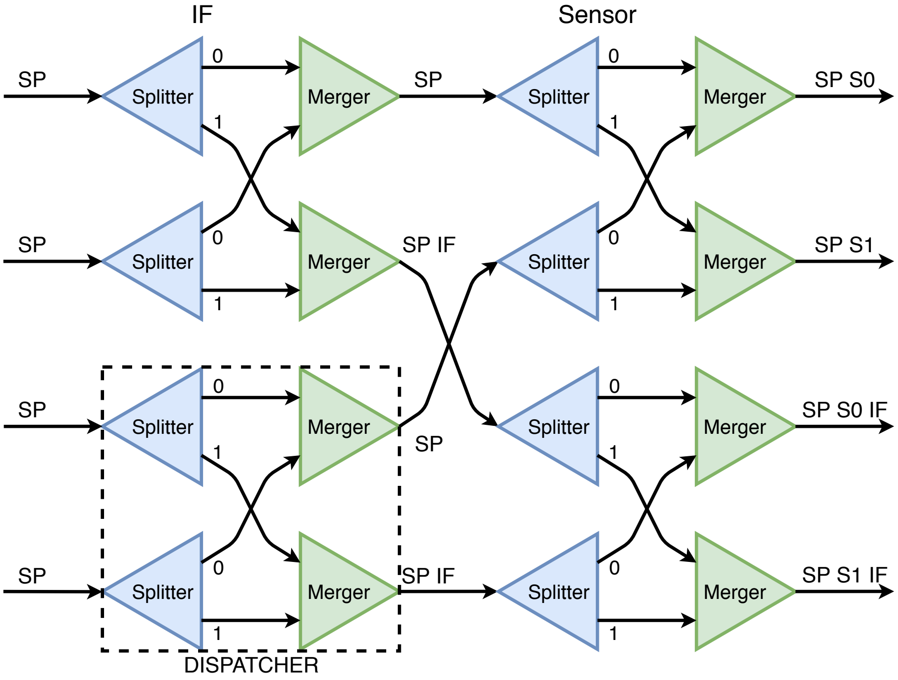


Figure 9.25: Structure of a 4 to 4 switching unit.

9.3.3 Clustering

After the switch, two of the eight lines contain only isolated SPs from sensor 0 (of the pair in the same stream), two lines contain only isolated SPs from sensor 1, two lines contain only non-isolated SPs from sensor 0, and two lines contain only non-isolated SPs from sensor 1. Each line with isolated cluster is reconstructed by a dedicated "isolated clustering block", the two line with non-isolated SPs from the same sensors are reconstructed by a single clustering matrices chain.

Isolated clustering

All isolated SPs, identified by the switch, are sent to the corresponding clustering block. Considering that a SP is composed by only eight pixels then the best way (in terms of throughput and resources usage) to clusterise isolated ones is to use a LUT, where the centre of mass for each possible configuration is stored. Figure 9.26 shows how isolated SPs are resolved by means of a LUT.

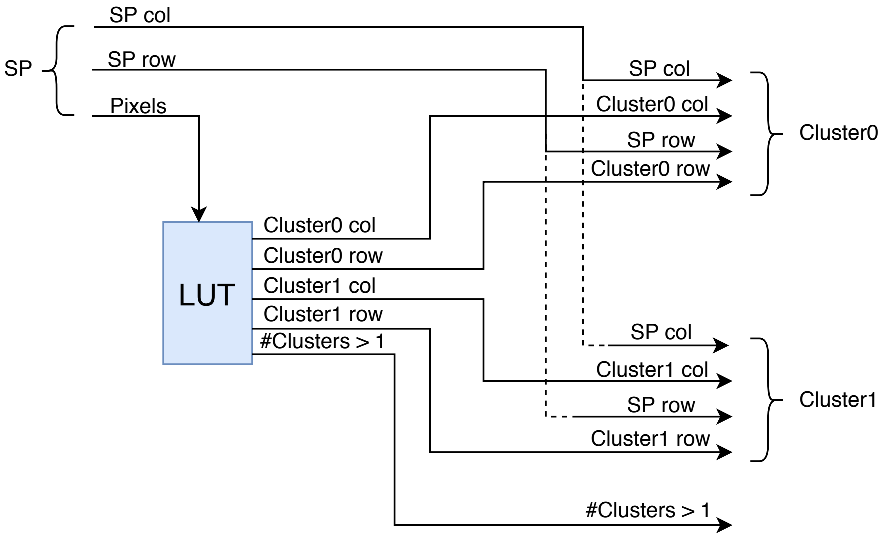


Figure 9.26: Cluster reconstruction of isolated SPs by means of a LUT.

The LUT returns the cluster centroid from the pixel hitmap, extracted from the SP word. The cluster word is then built combining the LUT output with the original SP row and column. Within an isolated SP up to two clusters can coexist, in which case a bit is raised, the two outputs are combined using a merger and then stored into a FiFo.

Non-isolated clustering

Not isolated cluster reconstruction is performed by a chain of 20 matrices. Each matrix receives data from two independent input lines, increasing the data processing rate with respect to a single line. Each input line is combined with an hold signal, that is propagated backwards through the whole chain, controlling the input data flow. Figure 9.27 shows how the SP distribution is performed.

Each matrix is filled starting from its centre. As the first SP populates a matrix, the set of neighbour SPs coordinates is calculated. The subsequent SP feeding into the same matrix compare its coordinates to the set ones. If a SP does not match any possible slot of the matrix, it checks the subsequent matrix of the chain, filling the central position in case of empty matrix, and so on until all the available matrices of the chain are filled. If all the 20 matrices are already busy, the SP is resolved by a LUT, the same as for the isolated SPs.

Each matrix has two input lines; when two SPs arrive on the same clock cycle at the input of a non-empty matrix, they are processed simultaneously, without generating any access conflict. However, if the matrix is empty, the SP on the first

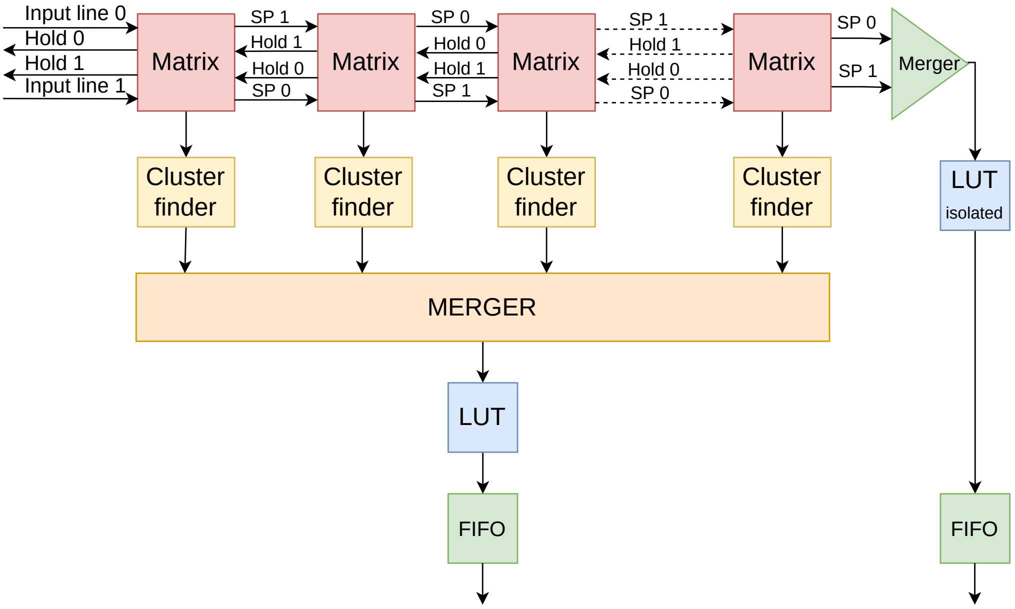


Figure 9.27: SPs distribution in a matrix chain. Clusters are identified through the cluster finder block, resolved by a common LUT and stored into a FiFo. Overflowing SPs are treated separately.

line is processed, and the one on the second line is put in hold, since it might or might not belong to the matrix. To avoid unbalancing the loads on the two lines due to the priority given to the first line, the lines are switched before feeding the next matrix.

When two EE signals have been received on both input lines of a matrix, the matrix content is sent out to the cluster finder block. Then the matrices are reset and they are ready to receive the next event. An error is raised if two different EE signals are detected.

Figure 9.28 shows the structure of the cluster finder. Each pixel in the matrix, through parallel pixel checker instances, checks if it belongs to one of the cluster patterns specified by the algorithm. If a pattern match is found, the corresponding pixel flag is set. An encoder reads the pixel flag content and passes the raised flag address in sequence. These address are used as selector in a multiplexer to extract the 3×3 cluster candidate from the matrix, and as input to a decoder to reset the corresponding flag through the pixel flush signal. It is also recorded in a FiFo together with the matrix coordinated and the cluster candidate as information required for cluster reconstruction.

Reconstructed clusters are then read from the matrix FiFos and gathered in a single line by a tree of merger. The merger used for this task are the same of the switch. The centroid of the cluster candidate is computed using a LUT and the reconstructed cluster word is saved into a FiFo. The actual cluster word is obtained combining the matrix coordinates in the detector, the checked pixel position in the matrix and the LUT output. In this way, a single LUT reconstructs the clusters of all the matrices, reducing the amount of required resources.

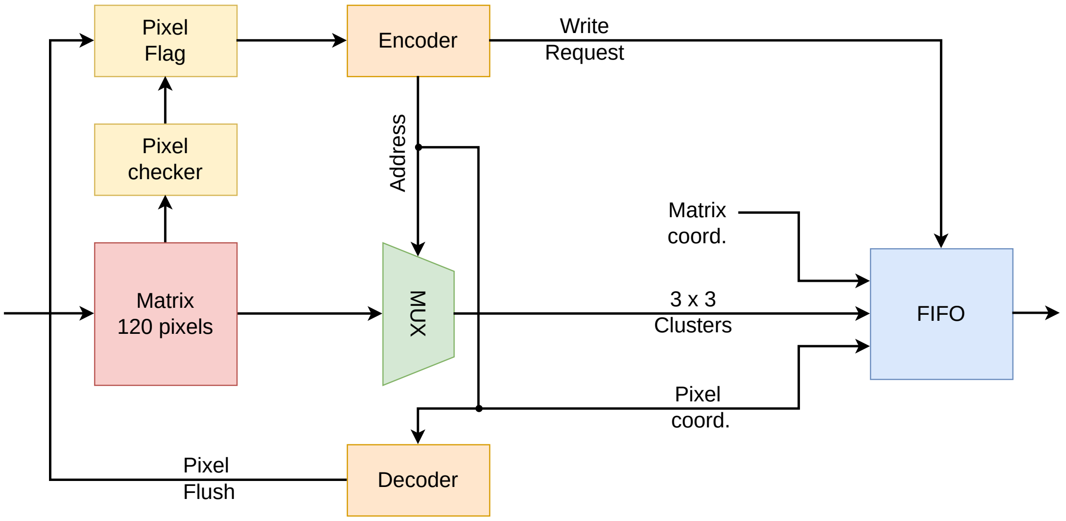


Figure 9.28: Cluster finder block diagram and its data flow.

9.3.4 Encoder

The encoder has the purpose to convert eight 32-bit words to one 256-bit word, and to convert back the EE logic to the SOP-EOP logic. It takes as input eight lines coming from different clustering blocks: 4 isolated clustering, 2 clustering matrices, 2 clustering matrices overflow.

The encoder must satisfy some requirements that make this task not trivial. It must process events at a rate compatible with the LHCb DAQ ($> 30\,\mathrm{MHz}$). Clusters from different events must remain separated. The finite PCIe and EB network bandwidth require to limit the number of words, so the encoder needs to be designed to produce as few words as possible in spite of the unbalanced input, with some lines typically carring more clusters than others. Merging all input lines to a single one, and then populating a 256-bit word for every eight 32-bit words ensures to produce the minimum number of words, but would make it impossible to achieve the target throughput. Populating the 256-bit word with the concatenation of the eight 32-bit lines content ensures maximum throughput, but produces a number of words equal to the number of clusters on the busiest line.

As a compromise between the extremes, I designed the encoder as a tree of basic encoder blocks composed recursively. Each basic clock puts together two input data lines ($N + N$ bit) into a single output ($2N$ bit), where the lines width are adjusted while moving from one recursive layer to the next one. If two words are simultaneously received on the input lines they are sent directly to the output, while if only one word is received it is stored in a register and matched with the next input word. The second word can arrive from the same line, so the output word is populated also when the input is unbalanced. If at the end of the event an odd number of words is received, the output word is zero-padded to the $2N$ output width. The EE word ensures event separation. As soon as EE words are received on both lines, they are compared and sent out, if they match. An error signal is generated otherwise. At the end of the tree, the EE is replaced with SOP-EOP signals.

Figure 9.29 shows the structure of the basic encoder clock. The main components are the "R0" "R1" "R3" and "State" registers, the "MUX0" "MUX1" "MUX3" multiplexers, and the FSM that controls the encoder according to the inputs. The FSM generates signals that regulate the behaviour of the others components. It takes in input the "State" register, the "Hold in" signal, the "Valid in" signal for the two lines, and if the incoming data are or not EEs, for a total of 10 bit of information, and 1024 combinations. It is the most complex FSM of the clustering and "Artificial Retina" designs.

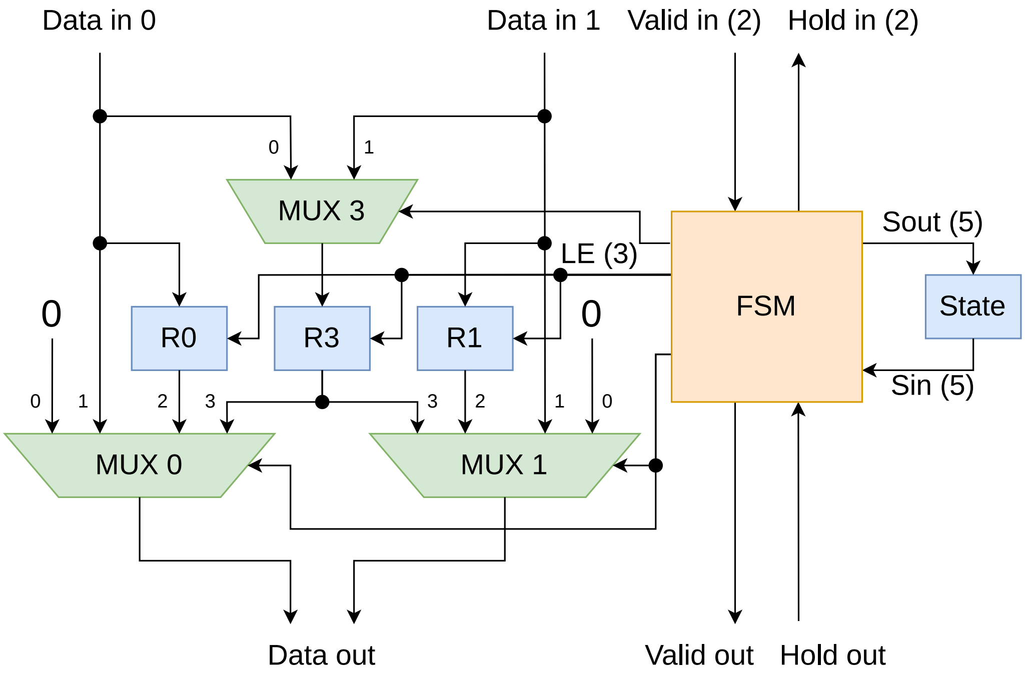


Figure 9.29: Structure of the basic encoder block. The number inside the bracket indicates the width of non-data signals, if omitted the width is 1.

9.3.5 FPGA resources and throughput

After the developing stage, I compiled and tested the design on the Stratix V prototyping board. In order to run clustering as a real-time process, the firmware has to sustain a 30 MHz event processing rate, due to the LHC average bunch crossing rate. The processing rate is determined by the slowest component in the firmware, that is the clustering of not isolated SPs, through the matrix chain. The clustering firmware throughput is inversely proportional to the number of SPs in the event and it can process events with up to 32 SPs on average per event, per sensor pair, at a 350 MHz clock rate. This condition is met for the whole VELO detector, with the highest occupancy being 26 SPs per event, near the nominal interaction point.

I measured an average event processing rate of 38.9 MHz while reconstructing clusters using SPs from the most populated VELO module at 350 MHz clock. This implies that the algorithm can run in real time within the VELO readout chain. I repeated the measurement using high-track-multiplicity $B_s \to \phi\phi$ events. The measured 30.9 MHz throughput ensures that, even with a series of high-occupancy events, the processing rate is still above the average LHC bunch crossing rate.

During testing stages, the firmware is fed with SP data coming from RAM memories that are read in loop. Data are produced using the LHCb Simulation and written to RAMs. The output clusters have been collected and compared to the output of the C++ simulation of the algorithm to ensure the quality of the reconstruction.

9.4 Adoption for Run 3 physics data taking

The FPGA mounted on the prototyping board is comparable to the one used for the LHCb Readout Boards in terms of amount of programmable logic, memory, and speed. Compiling the clustering design for the Readout Board Arria 10 chip, the firmware requires roughly 32% of logic and 10% of memory of the FPGA to process an entire VELO module. The small amount of logic and memory resources needed makes it possible to integrate this design inside the VELO readout firmware. This allows to perform clustering on FPGA since Run 3, without the need for extra cards and cost.

Finding cluster on a 2-dimensional detector requires a significant fraction of event reconstruction time. Despite the optimisations, HLT1 VELO clustering algorithm requires $\sim$ 17% of the total computational time. Offloading clustering to FPGA cards relieves the HLT1 GPU farm workload. Decoding RC in HLT1 instead of performing the entire clustering, reconstruction shows a gain in the event rate throughput of about 11%, consistently on several GPU cards.

Both SPs and RCs are encoded in 32-bit words. Since cluster from non-isolated SPs are spread over multiple SPs, the number of clusters is lower than the number of SPs. There are 24% less RCs than SPs. Therefore the integration of clustering within the VELO firmware helps data distribution between EB nodes reducing the bandwidth required. However the bandwidth reduction cannot be simply quantified comparing the number of SPs and RCs. Indeed the EB nodes exchange the 256-bit words used as input to the decoder (in the case of SPs) or the 256-bit words produced by the encoder (in the case of RCs). In case the number of SPs or RCs in an event is not multiple of 8, the last 256-bit word is zero-padded. Moreover, to process events at a rate compatible with the LHCb DAQ, the encoder was designed in a way that some words are not fully populated with clusters. Thus I compared the number of

256-bit words containing RCs with the number of 256-bit words containing the SPs of the same events. The clustering firmware reduces the words number by 14%. In conclusion this is a precise measure of the VELO bandwidth reduction in the EB network. It is also a index of the encoder efficiency.

Considering all these benefits and the physics performances equivalent to the ones of GPU algorithm, the LHCb collaboration choose to adopt the FPGA clustering as default solution for Run 3 data tacking.

Chapter 10

Conclusion

In 2019, charm *CPV* was observed in $D^0 \to \pi^+\pi^-$ and $D^0 \to K^+K^-$ decays. To understand if it is really compatible with a SM origin, it is of paramount importance to detect and measure *CPV* effects in additional charm decay modes. Theoretical studies pointed out some other charm decays where *CP* asymmetry may be detectable ($\simeq 10^{-3}$) most notably D^0 decays into two neutral kaons, like $D^0 \to K^0\overline{K}^{*0}$ and $D^0 \to \overline{K}^0K^{*0}$ decays. Thus they are promising channel to expand our knowledge on *CPV* in the charm sector in the next future.

This thesis describes a *CPV* analysis of the largest available sample of $D^0 \to K^0_S K^-\pi^+$ and $D^0 \to K^0_S K^+\pi^-$ decays, collected by LHCb in the past Run 2. The analysis, currently under internal review, introduces a novel methodology to measure the difference between the complex coefficient of the $K^*(892)^0$ resonance, without requiring a full Dalitz amplitude analysis. This will be the most precise result available, with a statistical resolution on the amplitude difference between $K^*(892)^0$ complex coefficient of 1.1% for the $\overline{K}^{*0}$ resonance and 1.2% for the K^{*0} resonance. The statistical resolution on the phase difference between $K^*(892)^0$ complex coefficient will be 0.47° for the $\overline{K}^{*0}$ resonance and 0.50° for the K^{*0} resonance. According to current predictions, it is unlikely that these resolution will be sufficient to observe a *CP* violation signal with current data, and a large part of my work has been invested in developing a new data processing technology, to enable the collection of the necessary huge data samples. LHCb has already a plan of future runs, with the potential to collect enough data to enable observation of *CP* violation in this channel, if at least the current trigger efficiency can be preserved at higher luminosities.

I made a major contribution to the development of a highly-parallelized custom tracking processor based on the “Artificial Retina” architecture, initially developed within the “RETINA Project” by INFN-CSN5 that concluded in 2018.

In this thesis I implemented the optimised version of the “Artificial Retina” core firmware, increasing the system throughput by a factor $1.2 - 2.44$ (depending on the occupancy) over the preliminary prototype. This allowed to prove that the “Artificial Retina” can sustain the target event rate of 30 MHz (the average collision rate at LHC) up to tracker occupancies of 1.5%, that are deemed sufficient for any LHCb sub-detector. I also designed and tested the Distribution Network of the system, implementing a optical network with asynchronous inputs and exchanging data between multiple boards at the maximum bandwidth achievable by modern FPGAs.

Finally, I have brought to a very advanced stage a demonstrator of the device, operating on a significant portion of a sub-detector, the VELO.

This demonstrator is being commissioned in the LHCb testbed facility, with a plan of running it in a parasitical test during Run 3. As part of this effort, I conceived and developed a FPGA-based clustering firmware, to feed hits coordinates to the VELO demonstrator. The physics performances of FPGA clustering are nearly indistinguishable from those of the HLT algorithm, and given the compactness of this firmware module, it has been possible to fit it in the spare resources available in the VELO Readout Boards, providing clusters not only to the demonstrator but also to HLT.

This leads to two benefits: a reduction of data size coming from the VELO of about 14%, and an improvement in the HLT1 event rate of about 11%, that directly translates into a corresponding improvement of the size of all physics data samples that LHCb will collect from now on. These improvements, albeit modest, have been achieved using a tiny amount of hardware resources and should be a strong signal of the effectiveness of this new methodology, and of the potential of the future physics program at LHCb.

Appendix A

Amplitude model lineshapes

The matrix elements $\mathcal{M}_R$ of the amplitude model depend on dynamical functions that describe the resonance R. In this appendix I report the lineshapes used in the amplitude model.

Relativistic Breit-Wigner

This is used if there is no particular motivation for an alternative shape. It is defined as:

$$T_R = \frac{1}{(m_R^2 - m_{AB}^2) - i m_R \Gamma_R(m_{AB})} = \frac{q_0}{m_R^2 \Gamma_R(m_{AB}) \rho(m_{AB})} \sin \delta_R(m_{AB}) e^{i\delta_R(m_{AB})} \tag{A.1}$$

where the running width is defined as:

$$\Gamma_R(m_{AB}) = \Gamma_R [B_L^R(q, q_0, d_R)]^2 \frac{m_R}{m_{AB}} \left(\frac{q}{q_0}\right)^{2L+1} \tag{A.2}$$

the scattering phase δ_R is given by:

$$\tan \delta_R(m_{AB}) = \frac{m_R \Gamma_R(m_{AB})}{m_R^2 - m_{AB}^2} \tag{A.3}$$

and the phase-space factor is given by:

$$\rho(m_{AB}) = \frac{q}{m_{AB}} \tag{A.4}$$

Flatté

This is a coupled-channel description used for the $a_0(980)^\pm$ resonance:

$$T_{a_0(980)^\pm} = \frac{1}{(m_{a_0(980)^\pm}^2 - m_{K\bar{K}}^2) - i[\rho_{K\bar{K}} g_{K\bar{K}}^2 + \rho_{\eta\pi} g_{\eta\pi}^2]} \tag{A.5}$$

where:

$$\rho_{AB} = \frac{1}{m^2_{K\bar{K}}}\sqrt{[m^2_{K\bar{K}} - (m_A + m_B)^2][m^2_{K\bar{K}} - (m_A - m_B)^2]}. \tag{A.6}$$

The coupling constants $g_{K\bar{K}}$ and $g_{\eta\pi}$ are set to the values measured by the Crystal Barrel collaboration [93]: $g_{\eta\pi} = 324 \pm 15\,\text{MeV}$, $\frac{g^2_{K\bar{K}}}{g^2_{\eta\pi}} = 1.03 \pm 0.14$.

Gounaris-Sakurai

This is a modified Breit-Wigner shape including finite-width corrections which is used for the $\rho^\pm \to K^0_\text{S} K^\pm$ resonances:

$$T_R = \frac{1 + d(m_R)\frac{\Gamma_R}{m_R}}{(m^2_R - m^2_{K\bar{K}}) + f(m^2_{K\bar{K}}, m^2_R, \Gamma_R) - i m_R \Gamma_R(m_{K\bar{K}})} \tag{A.7}$$

where:

$$d(m_R) = \frac{3m^2_K}{\pi q^2_0}\ln\left(\frac{m_R + 2q_0}{2m_K}\right) + \frac{m_R}{2\pi q_0} - \frac{m^2_K m_R}{\pi q^3_0} \tag{A.8}$$

where m_K is taken as the mean of $m_{K^0_\text{S}}$ and $m_{K^\pm}$.

$$f(m^2_{K\bar{K}}, m^2_R, \Gamma_R) = \Gamma_R \frac{m^2_R}{q^3_0}\{q^2_0\left[h(m^2_{K\bar{K}}) - h(m^2_R)\right] + q^2_0 h'(m^2_R)(m^2_R - m^2_{K\bar{K}})\} \tag{A.9}$$

where $h'(m^2_R) \equiv \frac{dh(m^2_R)}{d(m^2_R)}$ is calculated in the limit that $m_K = m_{K^\pm} = m_{K^0_\text{S}}$, and:

$$h(m^2) = \frac{2q(m^2)}{\pi m}\ln\left(\frac{m + 2q(m^2)}{2m_K}\right) \tag{A.10}$$

Generalised LASS (GLASS)

This shape is used to describe the $K\pi$ S-wave:

$$T_R \sim \frac{F\sin(\delta_F + \phi_F)e^{i(\delta_F+\phi_F)} + R\sin(\delta_R)e^{i(\delta_R+\phi_R)}e^{2i(\delta_F+\phi_F)}}{\rho(m_{AB})} \tag{A.11}$$

with δ_r defined in equation A.3 and

$$\tan\delta_F = \frac{2aq}{2 + arq^2} \tag{A.12}$$

where the a is the scattering length, and r the effective range. Together with F, ϕ_F, and ϕ_R they are free parameters in the fit performed in Run 1 analysis, table A.1 reports their values. R is constant and equal to 1.

Parameter	$K_0^*(1430)^0$	$K_0^*(1430)^\pm$	
F	$0.15 \pm 0.03 \pm 0.14$	1.785 (fixed)	
a	$4.2 \pm 0.3 \pm 2.8$	$4.7 \pm 0.4 \pm 1.0$	$(\,\mathrm{GeV}/c)^{-1}$
ϕ_F	$-2.5 \pm 0.2 \pm 1.0$	$0.28 \pm 0.05 \pm 0.19$	rad
ϕ_S	$-1.1 \pm 0.6 \pm 1.3$	$2.8 \pm 0.2 \pm 0.5$	rad
r	$-3.0 \pm 0.4 \pm 1.7$	$-5.3 \pm 0.4 \pm 1.9$	$(\,\mathrm{GeV}/c)^{-1}$

Table A.1: Value of GLASS parameters found in Run 1 analysis.

Appendix B

Extraction of BER upper limit

In digital transmission data stream can be altered due to noise, interference, distortion or bit synchronisation errors. The bit error ratio (BER):

$$\text{BER} = \frac{\#\, transmission\, errors}{\#\, transmitted\, bits}, \tag{B.1}$$

is an estimator of the bit error probability, *i.e.* the probability that a bit is altered during transmission.

For links where no transmission errors were detected, BER estimated with the maximum likelihood method is 0, however a small bit error probability could produce no transmission errors with some probability; it is therefore appropriate to estimate an upper limit.

Since a single bit can be transmitted rightly or wrongly, this is a Bernoulli process. However, when the numbers of trial is high (I performed the measures with at least $4.58 \cdot 10^{15}$ transmitted bits), the binomial distribution converges to the Poisson distribution:

$$\mathcal{B}(n,\, p) \sim \mathcal{P}(np)$$

where n is the number of trials, and p is the bit error probability. This allow to estimate the upper limit with simpler formulas.

For a Poisson process, given a time interval where on average occur λ events, the probability to observe k events is:

$$P(k|\lambda) = \frac{\lambda^k}{k!} e^{-\lambda}. \tag{B.2}$$

So I can calculate the maximum value of λ compatible with $k = 0$ withing a chosen Confidence Level CL:

$$1 - CL = P(k = 0|\lambda_{CL}) = e^{-\lambda_{CL}} \tag{B.3}$$

$$\lambda_{CL} = -\ln(1 - CL). \tag{B.4}$$

$CL = 95\% \rightarrow \lambda_{95} \approx 3$, and $\text{BER} < \lambda_{95}/n = 6.55 \cdot 10^{-16}$.

In the 23 days test $n = 2.07 \cdot 10^{16}$ and BER $< 1.45 \cdot 10^{-16}$ ($CL = 95\%$).

Also if the number of observed errors is not zero, but still small, the upper limit to the BER could be estimated. In this case we need to consider in the Equation B.3 also the probability of observing more than zero events. The equation became:

$$1 - CL = P(k \leq k_o|\lambda_{CL}) = \sum_{k=0}^{k_o} P(k|\lambda_{CL}) = e^{-\lambda_{CL}} \sum_{k=0}^{k_o} \frac{{\lambda_{CL}}^k}{k!} \tag{B.5}$$

where k_o is the observed number of events. The value of λ_{CL} can also be calculated as the extreme of integration of a χ^2 distribution with $2(k_o + 1)$ degree of freedom for which:

$$CL = \int_0^{2\lambda_{CL}} \chi^2_{2(k_o+1)}(x)dx. \tag{B.6}$$

Literature provides tables with the integral result.

Bibliography

[1] LHCb collaboration, R. Aaij *et al.*, *Observation of CP violation in charm decays*, Phys. Rev. Lett. **122** (2019) 211803, `arXiv:1903.08726`.

[2] S. Weinberg, *A Model of Leptons*, Phys. Rev. Lett. **19** (1967) 1264.

[3] J. C. W. Abdus Salam, *Weak and electromagnetic interactions*, Il Nuovo Cimento **11** (1959) 568.

[4] S. L. Glashow, *Partial-symmetries of weak interactions*, Nuclear Physics **22** (1961) 579.

[5] A. Salam and J. C. Ward, *Electromagnetic and weak interactions*, Physics Letters **13** (1964) 168.

[6] C. A. Baker *et al.*, *Improved Experimental Limit on the Electric Dipole Moment of the Neutron*, Phys. Rev. Lett. **97** (2006) 131801.

[7] V. Baluni, CP-*nonconserving effects in quantum chromodynamics*, Phys. Rev. D **19** (1979) 2227.

[8] A. P. Serebrov and et al. *New measurements of the neutron electric dipole moment*, JETP Letters **99** (2014) 4.

[9] N. Cabibbo, *Unitary symmetry and leptonic decays*, Phys. Rev. Lett. **10** (1963) 531.

[10] M. Kobayashi and T. Maskawa, *CP-violation in the renormalizable theory of weak interaction*, Prog. Theor. Phys. **49** (1973) 652.

[11] L. Wolfenstein, *Parametrization of the Kobayashi-Maskawa Matrix*, Phys. Rev. Lett. **51** (1983) 1945.

[12] *Commutator of the quark mass matrices in the standard electroweak model and a measure of maximal* CP *nonconservation*, .

[13] Particle Data Group, P. A. Zyla *et al.*, *Review of particle physics*, Prog. Theor. Exp. Phys. **2020** (2020) 083C01.

[14] CKMfitter group, J. Charles *et al.*, *Current status of the standard model CKM fit and constraints on $\Delta F = 2$ new physics*, Phys. Rev. **D91** (2015) 073007, `arXiv:1501.05013`, updated results and plots available at `http://ckmfitter.in2p3.fr/`.

[15] V. Weisskopf and E. Wigner, *Over the natural line width in the radiation of the harmonius oscillator*, Z. Phys. **65** (1930) 18.

[16] M. Golden and B. Grinstein, *Enhanced CP violations in hadronic charm decays*, Physics Letters B **222** (1989) 501.

[17] Y. Grossman, A. L. Kagan, and Y. Nir, *New physics and CP violation in singly Cabibbo suppressed D decays*, Phys. Rev. D **75** (2007) 036008.

[18] F. Buccella *et al.*, *Nonleptonic weak decays of charmed mesons*, Phys. Rev. D **51** (1995) 3478.

[19] A. Khodjamirian and A. A. Petrov, *Direct CP asymmetry in $D \to \pi^-\pi^+$ and $D \to K^-K^+$ in QCD-based approach*, Physics Letters B **774** (2017) 235.

[20] BaBar collaboration, D. Boutigny *et al.*, *The BABAR physics book: Physics at an asymmetric B factory*, 1998.

[21] LHCb collaboration, R. Aaij *et al.*, *Measurements of prompt charm production cross-sections in pp collisions at $\sqrt{s}$ =13 TeV*, JHEP **03** (2016) 159, Erratum ibid. **09** (2016) 013, Erratum ibid. **05** (2017) 074, `arXiv:1510.01707`.

[22] LHCb collaboration, R. Aaij *et al.*, *Observation of D^0–$\overline{D}^0$ oscillations*, Phys. Rev. Lett. **110** (2013) 101802, `arXiv:1211.1230`.

[23] BABAR Collaboration, B. Aubert *et al.*, *Evidence for $D^0-\overline{D}^0$ Mixing*, Phys. Rev. Lett. **98** (2007) 211802.

[24] Belle Collaboration, M. Starič *et al.*, *Evidence for $D^0-\overline{D}^0$ Mixing*, Phys. Rev. Lett. **98** (2007) 211803.

[25] CDF Collaboration, T. Aaltonen *et al.*, *Evidence for $D^0 - \overline{D}^0$ Mixing Using the CDF II Detector*, Phys. Rev. Lett. **100** (2008) 121802.

[26] Heavy Flavor Averaging Group, Y. Amhis *et al.*, *Averages of b-hadron, c-hadron, and τ-lepton properties as of 2018*, Eur. Phys. J. **C81** (2021) 226, `arXiv:1909.12524`, updated results and plots available at `https://hflav.web.cern.ch`.

[27] LHCb collaboration, R. Aaij *et al.*, *Observation of the mass difference between neutral charm-meson eigenstates*, Phys. Rev. Lett. **127** (2021) 111801, `arXiv:2106.03744`.

[28] LHCb collaboration, *Physics case for an LHCb Upgrade II — Opportunities in flavour physics, and beyond, in the HL-LHC era*, `arXiv:1808.08865`.

[29] LHCb collaboration, *Expression of Interest for a Phase-II LHCb Upgrade: Opportunities in flavour physics, and beyond, in the HL-LHC era*, CERN-LHCC-2017-003, 2017.

[30] H.-n. Li, C.-D. Lu, and F.-S. Yu, *Branching ratios and direct CP asymmetries in $D \to PP$ decays*, Phys. Rev. D **86** (2012) 036012.

[31] F. Buccella, A. Paul, and P. Santorelli, *$SU(3)_F$ breaking through final state interactions and CP asymmetries in $D \to PP$ decays*, Phys. Rev. D **99** (2019) 113001.

[32] H.-Y. Cheng and C.-W. Chiang, *Revisiting CP violation in $D \to PP$ and VP decays*, Phys. Rev. D **100** (2019) 093002.

[33] U. Nierste and S. Schacht, *CP violation in $D^0 \to K_S K_S$*, Phys. Rev. D **92** (2015) 054036.

[34] LHCb collaboration, R. Aaij *et al.*, *Measurement of CP asymmetry in $D^0 \to K^0_S K^0_S$ decays*, Phys. Rev. **D104** (2021) L031102, `arXiv:2105.01565`.

[35] U. Nierste and S. Schacht, *Neutral $D \to KK^*$ Decays as Discovery Channels for Charm CP Violation*, Phys. Rev. Lett. **119** (2017) 251801.

[36] H.-Y. Cheng and C.-W. Chiang, *CP violation in quasi-two-body $D \to VP$ decays and three-body D decays mediated by vector resonances*, `arXiv:2104.13548`.

[37] Particle Data Group, C. Patrignani *et al.*, *Review of particle physics*, Chin. Phys. **C40** (2016) 100001.

[38] S. Müller, U. Nierste, and S. Schacht, *Topological amplitudes in D decays to two pseudoscalars: A global analysis with linear $SU(3)_F$ breaking*, Phys. Rev. D **92** (2015) 014004.

[39] LHCb collaboration, R. Aaij *et al.*, *Studies of the resonance structure in $D^0 \to K^0_S K^{\pm}\pi^{\mp}$ decays*, Phys. Rev. **D93** (2016) 052018, `arXiv:1509.06628`.

[40] B. Bhattacharya and J. L. Rosner, *Flavor-SU(3) tests from $D^0 \to K^0 K^- \pi^+$ and $D^0 \to \overline{K}^0 K^+ \pi^-$ Dalitz plots*, Physics Letters B **714** (2012) 276.

[41] CLEO Collaboration, J. Insler *et al.*, *Studies of the decays $D^0 \to K^0_S K^- \pi^+$ and $D^0 \to K^0_S K^+ \pi^-$*, Phys. Rev. D **85** (2012) 092016.

[42] O. S. Brüning *et al.*, *LHC Design Report*, CERN Yellow Reports: Monographs, CERN, Geneva, 2004.

[43] LHCb collaboration, A. A. Alves Jr. *et al.*, *The LHCb detector at the LHC*, JINST **3** (2008) S08005.

[44] LHCb collaboration, R. Aaij *et al.*, *LHCb detector performance*, Int. J. Mod. Phys. **A30** (2015) 1530022, `arXiv:1412.6352`.

[45] LHCb collaboration, *LHCb VELO (VErtex LOcator): Technical Design Report*, CERN-LHCC-2001-011, 2001.

[46] M. Tobin, *The LHCb Silicon Tracker*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **831** (2016) 174, Proceedings of the 10th International "Hiroshima" Symposium on the Development and Application of Semiconductor Tracking Detectors.

[47] LHCb collaboration, *LHCb outer tracker: Technical Design Report*, CERN-LHCC-2001-024, 2001.

[48] M. Adinolfi *et al.*, *Performance of the LHCb RICH detector at the LHC*, Eur. Phys. J. **C73** (2013) 2431, `arXiv:1211.6759`.

[49] LHCb collaboration, *LHCb calorimeters: Technical Design Report*, CERN-LHCC-2000-036, 2000.

[50] A. A. Alves Jr. *et al.*, *Performance of the LHCb muon system*, JINST **8** (2013) P02022, `arXiv:1211.1346`.

[51] R. Aaij *et al.*, *Performance of the LHCb trigger and full real-time reconstruction in Run 2 of the LHC*, JINST **14** (2019) P04013, `arXiv:1812.10790`.

[52] G. Dujany and B. Storaci, *Real-time alignment and calibration of the LHCb Detector in Run II*, J. Phys. Conf. Ser. **664** (2015) 082010.

[53] S. Tolk, J. Albrecht, F. Dettori, and A. Pellegrino, *Data driven trigger efficiency determination at LHCb*, LHCb-PUB-2014-039, 2014.

[54] T. Likhomanenko *et al.*, *LHCb topological trigger reoptimization*, J. Phys. Conf. Ser. **664** (2015) 082025.

[55] `https://twiki.cern.ch/twiki/bin/view/LHCb/DecayTreeFitter`.

[56] `https://twiki.cern.ch/twiki/bin/view/LHCb/VertexFitters`.

[57] LHCb collaboration, *Framework TDR for the LHCb Upgrade: Technical Design Report*, CERN-LHCC-2012-007, 2012.

[58] LHCb collaboration, *Letter of Intent for the LHCb Upgrade*, CERN-LHCC-2011-001, 2011.

[59] LHCb collaboration, *LHCb VELO Upgrade Technical Design Report*, CERN-LHCC-2013-021, 2013.

[60] LHCb collaboration, *LHCb Tracker Upgrade Technical Design Report*, CERN-LHCC-2014-001, 2014.

[61] LHCb collaboration, *LHCb Trigger and Online Upgrade Technical Design Report*, CERN-LHCC-2014-016, 2014.

[62] LHCb collaboration, *LHCb Upgrade GPU High Level Trigger Technical Design Report*, CERN-LHCC-2020-006, 2020.

[63] Pisani, Flavio *et al.*, *Network simulation of a 40 MHz event building system for the LHCb experiment*, EPJ Web Conf. **245** (2020) 01012.

[64] P. Durante *et al.*, *100 Gbps PCI-Express readout for the LHCb upgrade*, Journal of Instrumentation **10** (2015) C04018.

[65] R. Aaij *et al.*, *A Comparison of CPU and GPU Implementations for the LHCb Experiment Run 3 Trigger*, Computing and Software for Big Science **6** (2022) .

[66] L. Ristori, *An artificial retina for fast track finding*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **453** (2000) 425, Proc. 7th Int. Conf on Instrumentation for colliding Beam Physics.

[67] D. H. Hubel and T. N. Wiesel, *Receptive fields of single neurones in the cat's striate cortex*, The Journal of physiology **148** (1959) 574–591.

[68] D. H. Hubel and T. N. Wiesel, *Receptive fields, binocular interaction and functional architecture in the cat's visual cortex*, The Journal of physiology **160** (1962) 106–154.

[69] W. Ashmanskas *et al.*, *The CDF online Silicon Vertex Tracker*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **485** (2002) 178.

[70] M. Shochet *et al.*, *Fast TracKer (FTK) Technical Design Report*, CERN-LHCC-2013-007, ATLAS-TDR-021, 2013.

[71] P. Hough, *Machine analysis of Bubble Chamber Pictures*, Proc. Int. Conf. High Energy Accelerators and Instrumentation **C590914** (1959).

[72] P. Hough, *Method and mean for recognizing complex patterns*, US Patent **3069654** (1962).

[73] R. Cenci *et al.*, *First Results of an "Artificial Retina" Processor Prototype*, EPJ Web Conf. **127** (2016) 00005.

[74] R. Cenci *et al.*, *Development of a High-Throughput Tracking Processor on FPGA Boards*, PoS **TWEPP-17** (2018) 136.

[75] R. Cenci *et al.*, *Performance of a high-throughput tracking processor implemented on Stratix-V FPGA*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **936** (2019) 344, Frontier Detectors for Frontier Physics: 14th Pisa Meeting on Advanced Detectors.

[76] LHCb collaboration, R. Aaij *et al.*, *Test of lepton universality with* $B^0 \to K^{*0}\ell^+\ell^-$ *decays*, JHEP **08** (2017) 055, `arXiv:1705.05802`.

[77] G. Punzi *et al.*, *Real-time reconstruction of pixel vertex detectors with FPGAs*, PoS **Vertex2019** (2020) 047.

[78] G. Tuci and G. Punzi, *Reconstruction of track candidates at the LHC crossing rate using FPGAs*, EPJ Web Conf. **245** (2020) 10001.

[79] G. Tuci, *Searching for confirmation of charm CP violation in* K^0_S *final states at LHCb*, PhD Thesis, Università di Pisa, 2020.

[80] J. Kim, W. J. Dally, S. Scott, and D. Abts, *Technology-Driven, Highly-Scalable Dragonfly Topology*, IEEE (2008) 77.

[81] J. J. Wilke, S. Rumley, and M. Y. Teh, *Design space exploration of the Dragonfly topology.*, ExaComm, 2017.

[82] A. Annovi and M. Beretta, *A fast general-purpose clustering algorithm based on FPGAs for high-throughput data processing*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **617** (2010) 254, 11th Pisa Meeting on Advanced Detectors.

[83] D. H. Cámpora Pérez, *Optimization of high-throughput real-time processes in physics reconstruction*, PhD Thesis, Universidad de Sevilla, 2019.

[84] T. Bird *et al.*, *VP Simulation and Track Reconstruction*, LHCb-PUB-2013-018. CERN-LHCb-PUB-2013-018, CERN, Geneva, 2013.

[85] LHCb collaboration, *LHCb computing: Technical Design Report*, CERN-LHCC-2005-019, 2005.

[86] M. Clemencic *et al.*, *The LHCb simulation application, Gauss: Design, evolution and experience*, J. Phys. Conf. Ser. **331** (2011) 032023.

[87] I. Belyaev *et al.*, *Handling of the generation of primary events in Gauss, the LHCb simulation framework*, J. Phys. Conf. Ser. **331** (2011) 032047.

[88] T. Sjöstrand, S. Mrenna, and P. Skands, *PYTHIA 6.4 physics and manual*, Journal of High Energy Physics **2006** (2006) 026.

[89] D. J. Lange, *The EvtGen particle decay simulation package*, Nucl. Instrum. Meth. **A462** (2001) 152.

[90] Geant4 collaboration, S. Agostinelli *et al.*, *Geant4: A simulation toolkit*, Nucl. Instrum. Meth. **A506** (2003) 250.

[91] Geant4 collaboration, J. Allison *et al.*, *Geant4 developments and applications*, IEEE Trans. Nucl. Sci. **53** (2006) 270.

[92] K. Hennessy *et al.*, *Readout Firmware of the Vertex Locator for LHCb Run 3 and Beyond*, IEEE Transactions on Nuclear Science **68** (2021) 2472.

[93] A. Abele *et al.*, *$\overline{p}p$ annihilation at rest into $K_L K^{\pm}\pi^{\mp}$*, Phys. Rev. D **57** (1998) 3860.