UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ

Instituto de Física

Study of the search of the decay mode

$K^+ \to \pi^+\gamma\gamma\gamma$

in the NA62 Experiment at CERN

TESIS PARA OBTENER EL GRADO DE

Maestro en Ciencias Física

PRESENTA

Lic. Akbar Emmanuel Díaz Rodarte

Director de Tesis

Dr. Jürgen Engelfried

San Luis Potosí, S.L.P.

Mayo 2026

INSTITUTO DE FÍSICA

Abstract

An upper limit for the branching ratio of the decay mode $K^+ \to \pi^+\gamma\gamma\gamma$ of 10^{-4}, was last reported by Y. Asano 1982. A study for the detection and possible improvement to the branching ratio upper limit employing data from the NA62 experiment at CERN is reported. A blind study was performed, employing only data from the year 2017, and Monte Carlos simulations provided by the collaboration. As result, the feasibility of the possible detection of the decay mode is confirmed as well as the possibility of improving the upper limit by 2 orders of magnitude.

Un límite superior para la fracción de decaimiento del modo $K^+ \to \pi^+\gamma\gamma\gamma$ de 10^{-4} fur reportada por última vez por Y. Asano en 1982. En la presente tesis se reporta el estudio para la detección y posible mejora de la fracción de decaimiento empleando datos del experimento NA62 del CERN. Se realizó un estudio ciego, empleando únicamente datos del año 2017, y simulaciones de Monte Carlo proporcionadas por la colaboración. Como resultado, la factibilidad de la detección del modo de decaimiento es confirmada, así como la posibilidad de mejorar el límite superior por 2 ordenes de magnitud.

This work is dedicated to
A. Rodarte, A. Díaz. A. R. Díaz
I. Alanis, who have allways supported
my career and goals.

Acknowledgements

This project wouldn't have been possible without the direction of **Dr. Jürgen Engelfried**, director of this master thesis. During my participation with the Mexican team of the NA62 Collaboration, Dr. Jürgen and **M.Sc. Nora Estrada** have always been supportive with my academic and personal development, sharing new perspectives, widening my horizons and questioning my believes. They have been a keystone in my participation in the NA62 activities such as taking shifts, presenting my work to the collaboration, and pursuing an internship at the Max Planck Institute for Physics. They have pushed me forward to overcome my insecurities. I can only say that you will always have a space in my memories and life.

I would also like to thank the NA62 Collaboration for the support, and feedback during my activities. In particular the MPP crew who kindly accepted me during my stay in München, and **Dr. Döbrich** for giving me the opportunity to work along her direction. An also important part for the conclusion of this work has come from my Mexican friends and colleges from the institute, **M. Sc. Alan Martinez, M. Sc. Alejandro Briano, M. Sc. Kevin Rodriguez**, among others, who have shared their ideas, time and support. It is also important to mention the attention and technical support of **I.E Luz del Carmen Nuche** and **L.E.S.D. José Limón** to the High Energy Physics Laboratory of the institute. Also, I would like to acknowledge the role of the institute, that provided the facilities, lectures, talks, and personal.

For the side of private funding, I would like to remark the help of the the Nuclemedica Soluciones, San Luis Potosí, Mexico, in particular to **I. Mercado**, whose generosity helped us cover the expenses for the 2025 shifts at the experiment. And the Hanrieder Excellence Program, supported by **Dr. Wolfgang Hanrieder** , that offers fellowships and internships at the Max Planck Institüt Für Physik, for Latin American students enrolled in High Energy Research programs, as was my case.

This work was supported by the 'Fondo Sectorial de Investigación para la Educación SEP/CONACyT', project 242139, and by the project 'Participación de México en la Frontera de Física de Altas Energías en el CERN', CONACyT Frontier Projects, number 2042.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

An important part of the experimental job is to stress the limitations of a given theory, as well as testing new ones. The current Standard Model of Physics (SM) considers the existence of discrete symmetries, namely $\mathbb{C}$, $\mathbb{P}$ and $\mathbb{T}$, and depending on the interaction this symmetries are conserved or violated in a given process. In the specific case of the weak interaction, which is involved in flavor change, $\mathbb{CP}$, and $\mathbb{P}$ are violated, while the electromagnetic interaction preserves all three discrete symmetries and their combinations. In the case of the $K^+ \to \pi^+\gamma\gamma\gamma$ decay, the weak interaction plays a role in the change of flavor of the quarks of the K^+ giving rise to the production of the π^+, while the electromagnetic interaction gives rise to the production of the three photons. If the $\mathbb{C}$ operator is applied to both initial and final states, it can be observed that the production of the three photons produces a violation of $\mathbb{C}$ conservation, which is prohibited for the electromagnetic interaction. In order to encounter a final state with three photons, then it is plausible to propose the existence of an intermediate produced particle which then decays into two photons. This scenario, occurs in the SM for example with the decay $K^+ \to \pi^+\pi^0\gamma$, where the π^0 then decays into two additional photons. Currently, extension theories to the SM trying to solve problems such as what dark mater is or the strong $\mathbb{CP}$ problem propose the existence of particles such as B or B-L bosons, which then decay into the pair $\pi^0\gamma$ [1], or axion like particles which then decay in photon pairs. In a work presented by Y. Asano [2] in 1982, an experimental limit for the Branching Ratio of the decay $K^+ \to \pi^+\gamma\gamma\gamma$ was reported to be 10^{-4} $CL_s = 90\%$, and had as motivation the possible identification of axions decaying into any number of photons.

The content of the following thesis corresponds to the area of high energy physics, focused on the data analysis of the experiment NA62 at CERN. The accumulated data, related to K^+ decays collected over the course of ten years, as well as the good resolution of the sub-detector systems of the experiment makes of NA62 a great candidate for the search of the $K^+ \to \pi^+\gamma\gamma\gamma$ decay. The main objective is to determine whether it is feasible or not the detection of such a final state, and if so, to determine a possible improvement to the previously reported Branching Ratio. For accomplishing this, data analysis software was developed and employed over 2017 data, as well as Monte Carlo simulations. Also, to open the door to future analysis of SM extensions, the search remained model independent, and a blind analysis strategy over data was performed to avoid confirmation biases.

Chapter 2

Brief review on Particle Physics

2.1 Standard model of particles physics

The Standard Model of Physics, SM for short, is the mathematical model employed to explain the fundamental blocks of matter and its interactions. It is constituted by two groups of elementary particles, fermions whom constitute matter and boson which carry the interaction between fermions. Currently, four fundamental interactions are conceived in the model, the electromagnetic, the strong nuclear interaction, the weak nuclear interaction, and gravitation.

Electromagnetic interaction happens between particles with electric charge, that can be either positive or negative, and is a quantized quantity since it is present in multiples of 1/3 of the electron change. The boson responsible for this interaction is the photon, a neutral massless particle of spin 1, parity of -1, and charge conjugation number of -1. Strong interaction happens between particles that have color charge, this is present in three different color charges represented by the $SU(3)_c$ group, the combination of the three colors or color anti-color give as result the charge color of white, and every free system has as net color white. The interaction is mediated by 8 gluons, which are massless particles that also contain color, have spin 1, and parity of -1. This interaction is the responsible for the stability of nuclei. The weak interaction is the responsible for the change of flavor between particles. It is mediated by three massive particles, the W^+ and W^-, and the Z^0 of spin 1. This interaction is the responsible of the beta decay of nuclei, the not electromagnetic scattering of neutrinos, etc. The gravitational interaction is responsible for the attraction of massive particles and is theoretically propagated by the graviton of spin 2.

From the side of the fermions, two sub-sets are formed, leptons and quarks. The main aspect which differentiate both is that leptons do not interact by the strong nuclear interaction, while quarks do interact strongly. both groups are subsequently divided into 3 different families, whose difference relies on the mass of the particles, see table 2.1. It should be also mentioned, that each fermion has its corresponding anti-particle, which has the same mass and family, but has complementary charges.

Type	Family	Flavor	Mass[MeV/c^2]	E.M. charge[e]
Leptons	1	Electron (e^-)	$(.5109 \pm 15x10^{-11})$	-1
		e-neutrino (ν_e)	$< 1.1eV/c^2$	0
	2	Muon (μ^-)	$(105.6 \pm 23x10^{-7})$	-1
		μ-neutrino (ν_μ)	-	0
	3	Tau (τ^-)	(1776.86 ± 0.12)	-1
		τ-neutrino (ν_τ)	-	0
Quarks	1	Up (u)	$(2.16^{+.49}_{-.26})$	2/3
		Down (d)	$(4.67^{+.48}_{-.17})$	$-1/3$
	2	Charm (c)	$(1.27 \pm 0.2) \times 10^3$	2/3
		Strange (s)	$(93.4^{+8.6}_{-0.17})$	$-1/3$
	3	Top (t)	$(172.69 \pm 0.30) \times 10^3$	2/3
		Bottom (b)	$(4.18^{+.03}_{-.02}) \times 10^3$	$-1/3$

Table 2.1: Elementary Fermion Particles, Their Family And Mass

As previously mentioned, the weak interaction is the responsible of changing the family of the fermions, by the emission of $W^\pm$ bosons. This only couples to the left-chiral component of the fermions, always with the same strength regardless of its family or type (weak interaction universality). In the case of leptons, the members of each family may be regarded as elements of tuples.

$$\mathfrak{E}_L = \begin{pmatrix} \nu_e \\ e^- \end{pmatrix}_L \quad \mathfrak{M}_L = \begin{pmatrix} \nu_\mu \\ \mu^- \end{pmatrix}_L \quad \mathfrak{T}_L = \begin{pmatrix} \nu_\tau \\ \tau^- \end{pmatrix}_L$$

In the case of the Quarks, a similar representation of the members of each family is possible. Yet the base in which the weak interaction couples to, expressed with primed literals, is not the same as the mass base, denoted with b, s, d (for bottom, strange and down). The rotation matrix which stablish the change of base is known as the CKM matrix (from Cabibbo, Kobayashi, Maskawa), see references [3] and [4]. The modulus squared of its entries determines the transition from quarks from a certain family to other.

$$\mathfrak{V} = \begin{pmatrix} cos\theta_{12}cos\theta_{13} & sin\theta_{12}cos\theta_{13} & sin\theta_{13}e^{-i\delta} \\ -sin\theta_{12}cos\theta_{23} - cos\theta_{12}sin\theta_{23}sin\theta_{13}e^{i\delta} & cos\theta_{12}cos\theta_{23} - sin\theta_{12}sin\theta_{23}sin\theta_{13}e^{i\delta} & sin\theta_{23}cos\theta_{13} \\ sin\theta_{12}sin\theta_{23} - cos\theta_{12}cos\theta_{23}sin\theta_{13}e^{i\delta} & -cos\theta_{12}sin\theta_{23} - sin\theta_{12}cos\theta_{23}sin\theta_{13}e^{i\delta} & cos\theta_{23}cos\theta_{13} \end{pmatrix}$$

$$\begin{pmatrix} d' \\ s' \\ b' \end{pmatrix}_L = \mathfrak{V} \begin{pmatrix} d \\ s \\ b \end{pmatrix}_L = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} \begin{pmatrix} d \\ s \\ b \end{pmatrix}_L$$

$$\mathfrak{D}_L = \begin{pmatrix} u \\ d' \end{pmatrix}_L \quad \mathfrak{S}_L = \begin{pmatrix} c \\ s' \end{pmatrix}_L \quad \mathfrak{B}_L = \begin{pmatrix} t \\ b' \end{pmatrix}_L$$

2.1.1 Baryons and Mesons

As previously mentioned, since free systems need to have white color charge, quarks group themselves into composed systems, which are called Hadrons. A subgroup of Hadrons are the Mesons, and consists of pairs of systems with pairs valence quarks, with color anti-color charges. The other subdivision of hadrons are the Baryons, which consists of three valence quarks, each with a different charge color.

Meson	**Valence Quarks**	**E.M Charge**$[e]$	**Spin** $[\hbar]$	**Mass**$[MeV/c^2]$
K^-	$\|s\bar{u}\rangle$	-1	0	493.677 ± 0.016
$\bar{K}^0$	$\|s\bar{d}\rangle$	0	0	497.611 ± 0.01
π^-	$\|\bar{u}d\rangle$	-1	0	139.57039 ± 0.00018
π^0	$\frac{1}{\sqrt{2}}(\|u\bar{u}\rangle - \|d\bar{d}\rangle)$	0	0	134.9768 ± 0.0005
η	$\frac{1}{\sqrt{2}}(\|u\bar{u}\rangle + \|d\bar{d}\rangle)$	0	0	547.862 ± 0.017
η'	$\|s\bar{s}\rangle$	0	0	957.78 ± 0.06
π^+	$\|u\bar{d}\rangle$	$+1$	0	139.57039 ± 0.00018
K^0	$\|d\bar{s}\rangle$	0	0	497.611 ± 0.013
K^+	$\|u\bar{s}\rangle$	$+1$	0	493.677 ± 0.016
K^{*-}	$\|s\bar{u}\rangle$	-1	1	1718 ± 18
$\bar{K}^{*0}$	$\|s\bar{d}\rangle$	0	1	1425 ± 50
ρ^-	$\|\bar{u}d\rangle$	-1	1	775.26 ± 0.23
ρ^0	$\frac{1}{\sqrt{2}}(\|u\bar{u}\rangle - \|d\bar{d}\rangle)$	0	1	775.26 ± 0.23
ω	$\frac{1}{\sqrt{2}}(\|u\bar{u}\rangle + \|d\bar{d}\rangle)$	0	1	782.66 ± 0.13
ϕ	$\|s\bar{s}\rangle$	0	1	1019.461 ± 0.016
ρ^+	$\|u\bar{d}\rangle$	$+1$	1	775.26 ± 0.23
K^{*0}	$\|d\bar{s}\rangle$	0	1	1425 ± 50
K^{*+}	$\|u\bar{s}\rangle$	$+1$	1	1718 ± 18

Table 2.2: Light Mesons And Some Of Their Properties

Spin $[\hbar]$	**Baryon**	**Valence Quark**	**Mass** $[MeV/c^2]$
	Proton(p)	$\|uud\rangle$	$938.27208816 \pm 29x10^{-8}$
	Neutron(n)	$\|ddu\rangle$	$939.5654205 \pm 5x10^{-7}$
	Σ^-	$\|dds\rangle$	1197.449 ± 0.03
$\frac{1}{2}$	Λ^0	$\|uds\rangle$	115.683 ± 0.006
$\frac{1}{2}$	Σ^0	$\|uds\rangle$	1192.642 ± 0.024
	Σ^+	$\|uus\rangle$	189.37 ± 0.07
	Ξ^-	$\|dss\rangle$	1321.71 ± 0.07
	Ξ^0	$\|uss\rangle$	1314.86 ± 0.20
	Δ^-	$\|ddd\rangle$	1234.9 ± 1.4
	Δ^0	$\|udd\rangle$	1231.3 ± 0.6
	Δ^+	$\|uud\rangle$	1234.9 ± 1.4
	Δ^{++}	$\|uuu\rangle$	1230.55 ± 0.20
$\frac{3}{2}$	Σ^{*-}	$\|dds\rangle$	1387.2 ± 0.5
$\frac{3}{2}$	Σ^{*0}	$\|uds\rangle$	1383.7 ± 1.0
	Σ^{*+}	$\|uus\rangle$	1382.83 ± 0.34
	Ξ^{*-}	$\|dss\rangle$	1535.0 ± 0.6
	Ξ^{*0}	$\|uss\rangle$	1531.80 ± 0.32
	Ω^-	$\|sss\rangle$	1672.45 ± 0.29

Table 2.3: Light Baryons And Some Of Their Properties
The Δ masses reported come from the Breit-Wigner masses reported by the PDG[5].

2.2 Discrete symmetries

The conservation laws observed from classical physics are the result of symmetries. The conservation of energy, momentum and angular momentum, arises from the invariance of interaction with respect to their canonically conjugated quantities, time, location and orientation in which it takes place. The transformations, for this quantities have in common that an infinitesimal transformation can be applied, so called continues transformation, in order to determine whether the interaction remains invariant, showing an additive behavior. Yet, other type of transformations exists, which are discrete transformations. The conservation laws that arise from this other type of symmetries have a multiplicative behavior, and in general the multi-component wave function will transform as follows:

$$\psi'_A(x') = \mathbf{U_D}{}_A^B \psi_B(x) \; where \; x'^\mu = (\hat{\mathbf{U}}_\mathbf{D} \; x)^\mu$$
$$\Rightarrow \psi'_A(x) = \mathbf{U_D}{}_A^B \psi_B(\hat{\mathbf{U}}_\mathbf{D}^{-1} \; x)$$

Where $\hat{\mathbf{U}}_\mathbf{D}$, is a representation of the transformation acting on the space-time components, while $\mathbf{U_D}$ is a matrix acting on the components of the wave function. The three most important discrete symmetries for particle physics are:

- Parity ($\mathbb{P}$): Also called mirror transformation, consist on carrying out every point in space diametrically opposite to the origin, hence changing the sign of every spatial component.

  $$\mathbb{P} : (t, x, y, z) \to (t, -x, -y, -z)$$

  The interactions which are invariant under parity transformations are the electromagnetic, and strong interactions, while the weak interaction doesn't preserve it.

- Time Reversal ($\mathbb{T}$): As the name implies, this transformation consists on changing the time component to its negative.

  $$\mathbb{T} : (t, x, y, z) \to (-t, x, y, z)$$

  This transformation is difficult to appreciate in macroscopic systems since they evolve in a way that time "flows" in a way in which the entropy of a system increments. Yet in quantum systems, this transformation can be observed in processes in which going from state A to B, is as possible as going from B to A. Like the nuclear reaction:

  $$n + p \to D + \gamma \Longleftrightarrow D + \gamma \to n + p$$

  When both process have the same reaction rate the process is invariant over $\mathbb{T}$, and receives the name of principle of equal balance. The interactions that are known to be invariant over $\mathbb{T}$ are the strong and electromagnetic.

- Charge Conjugation ($\mathbb{C}$): Is the transformation which preserves quantities such as momentum, mass, energy and spin, but reverses the charges such as electric charge, strangeness, iso-spin, etc. In other words, changes the state of a particle into its corresponding anti-particle.

  $$\mathbb{C}\,|e^-\rangle = |e^+\rangle \quad \mathbb{C}\,|K^+\rangle = |K^-\rangle$$

Only those particles that are their own anti-particles are eigen-states of this transformation, such as the photon or the π^0. For Fermion-AntiFermion bound states, applying $\mathbb{C}$ to the state produces a phase of $(-1)^{l+s}$. For the photon, applying the operator produces a change of sign -1 per present photon in the reaction. Up to now, it is stated that the electromagnetic, as well as the strong interaction, preserve charge conjugation, whereas the weak interaction doesn't.

Now, consider the following weak decay of the K^+, where n states for number the produced photons:

$$K^+ \to \pi^+ n\gamma$$

In this decay, the π^+ is produced by the weak interaction of the quark $\bar{s}$, changing into $\bar{u}$ and emitting a W^+ which decays and produces a quark u and a quark $\bar{d}$, which then joins the remaining quark u to form the π^+, the two remaining quarks might annihilates one-another electromagnetically instead of forming a π^0, hence producing the n photons.

By Furry's theorem, which comes from $\mathbb{C}$ conservation in electromagnetic decay processes, it is stated that any odd number of photons in the final state is prohibited [6], since any three vertex contribution has vanishing amplitude. Yet a paper by Basham, C. L [7] has proposed that, in weak interaction scenarios, the existence of 3 photons states have non vanishing contributions to the reaction rate. Nevertheless, in this model, two of the photons remain virtual, hence not detectable, and the contributions of this final states to the total reaction rate is in the order of 10^{-22} to 10^{-20}.

2.3 Transition Rate And Branching Ratio

The transition rate Γ_{if} is defined as the probability per unit of time for a particle in the initial state $|i\rangle$ to transition into a final state $|f\rangle$ as time progresses. Thus, in order to calculate it, the probability for the transition, P_{if} is needed.

$$\Gamma_{if} = \frac{dP_{if}}{dt} \tag{2.1}$$

In order to calculate P_{if}, time-perturbation theory can be used. Specifically, the so called Dyson-Series[8], equation 2.2, can be used. In it, $\hat{V}(t)_I$ refers to the transition potential applied to the particle after time t_o; $|i,t\rangle_I$, $|i,t_o\rangle_I$ both refer to the initial states at times t and t_o; the subscript I indicates that the state or potential is represented in the interaction picture, whereas $\hat{H}_o$ is the unperturbed Hamiltonian.

$$\begin{aligned} |i,t_o\rangle_I &= \exp\left(i\hat{H}_o t/\hbar\right)\ |i,t_o\rangle \\ \hat{V}(t)_I &= \exp\left(i\hat{H}_o t/\hbar\right)\ \hat{V}(t)\exp\left(-i\hat{H}_o t/\hbar\right) \end{aligned}$$

$$\begin{aligned} |i,t\rangle_I = |i,t_o\rangle_I & \\ &+\ (\frac{-i}{\hbar})\int_{t_o}^{t} dt'\hat{V}(t')_I\,|i,t_o\rangle_I \\ &+ (\frac{-i}{\hbar})^2\int_{t_o}^{t} dt'\int_{t_o}^{t'} dt''\hat{V}(t')_I\hat{V}(t'')_I\,|i,t_o\rangle_I\ +\ ... \end{aligned} \tag{2.2}$$

If $|i\rangle$ and $|f\rangle$ are eigen-states of $\hat{H}_o$ then the following equations are true:

$$\begin{aligned} |i,t\rangle &= \exp\left(-iE_i t/\hbar\right)|i\rangle \\ |i,t\rangle_I &= |i\rangle \\ |f,t\rangle &= \exp\left(-iE_f t/\hbar\right)|f\rangle \\ |f,t\rangle_I &= |f\rangle \end{aligned} \tag{2.3}$$

The transition probability can now be obtained by using the equation 2.2 and the set of equalities 2.3, by calculating the square modulus of the probability amplitudes:

$$\begin{aligned} P_{if} &= |{}_I\langle f,t|i,t\rangle_I|^2 \\ &= |\langle f|\left(|i,t_o\rangle_I + (\frac{-i}{\hbar})\int_{t_o}^{t} dt'\hat{V}(t')_I|i,t_o\rangle_I + \ ...\right)|^2 \\ &= |\langle f|i\rangle + (\frac{-i}{\hbar})\int_{t_o}^{t} dt'\langle f|\hat{V}(t')_I|i,t_o\rangle_I + \ ...|^2 \\ &= |\delta_{fi} + (\frac{-i}{\hbar})\int_{t_o}^{t} dt'\langle f|\hat{V}(t')_I|i\rangle + \ ...|^2 \\ &= |\delta_{fi} + (\frac{-i}{\hbar})\int_{t_o}^{t} dt'\langle f|exp(it'E_f/\hbar)\hat{V}(t')exp(-it'E_i/\hbar)|i\rangle + \ ...|^2 \\ &= |\delta_{fi} + (\frac{-i}{\hbar})\int_{t_o}^{t} dt' exp(it'\frac{(E_f-E_i)}{\hbar})\langle f|\hat{V}(t')|i\rangle + \ ...|^2 \end{aligned}$$

Lets consider now a potential $\hat{V}$ such that can be taken out from the integral (like a time independent one), and also lets set t_o as cero; It should also be noticed that the eigen-states $|i\rangle$ and $|f\rangle$ are orthogonal; hence $\delta_{if} = 0$. Then as a first order approximation for the transition probability it could be calculated as follows:

$$\begin{aligned} P_{if} &= \frac{1}{\hbar^2}|\langle f|\hat{V}|i\rangle\int_0^t \exp(it'\frac{(E_f-E_i)}{\hbar})dt'|^2 \\ &= \frac{1}{\hbar^2}|M_{if}\int_0^t \exp\left(it'\Delta\omega_{fi}\right)dt'|^2, \ M_{if} = \langle f|\hat{V}|i\rangle, \ \Delta\omega_{fi} = \frac{(E_f-E_i)}{\hbar} \\ &= \frac{1}{\hbar^2}|M_{if}(\frac{-i}{\Delta\omega_{fi}})(\exp\left(it\Delta\omega_{fi}\right)-1)|^2 \\ &= \frac{1}{\hbar^2}|M_{if}|^2\frac{(\cos^2\left(t\Delta\omega_{fi}\right)-1)^2+\sin^2\left(t\Delta\omega_{fi}\right)}{\Delta\omega_{fi}^2} \\ &= \frac{1}{\hbar^2}|M_{if}|^2(\frac{2-2\cos\left(t\Delta\omega_{fi}\right)}{\Delta\omega_{fi}^2}) \\ &= \frac{1}{\hbar^2}4|M_{if}|^2(\frac{1-\cos\left(t\Delta\omega_{fi}\right)}{2(\Delta\omega_{fi})^2}) \\ &= \frac{t}{\hbar^2}|M_{if}|^2(\frac{\sin^2\left(t\Delta\omega_{fi}/2\right)}{(\Delta\omega_{fi}/2)^2 t}) \end{aligned}$$

The last factor of this expression behaves as $t \to \infty$, and for final states with E_f close to E_i, as a $\delta(\Delta\omega_{fi})$ distribution, with the exception that it is not normalized. Hence, and considering that time is running from 0 to ∞, the following normalization should be taken into consideration.

$$\lim_{t\to\infty}\int_{-\infty}^{\infty}\frac{\sin^2\left(t\Delta\omega_{fi}/2\right)}{(\Delta\omega_{fi}/2)^2 t}dt \ = \pi$$

With this in mind, it can be stated that the transition probability, as well as the Reaction rate, can be estimated for final states with E_f close to E_i as follows:

$$\begin{aligned} P'_{if} &= \frac{2\pi t}{\hbar^2}|M_{if}|^2\delta(\Delta\omega_{fi}) \\ &= \frac{2\pi t}{\hbar}|M_{if}|^2\delta(E_f - E_i) \end{aligned} \tag{2.4}$$

$$\Gamma'_{if} = \frac{dP_{if}}{dt} = \frac{2\pi}{\hbar}|M_{if}|^2\delta(E_f - E_i) \tag{2.5}$$

If we consider that we are dealing with a continuous spectrum of states, then the number of initial states transitioning to the final state can be obtained with the density of states ρ as $\rho(E_i)dE_i$, and thus by integrating the transition rate, it becomes equation 2.6, also known as ***Fermi's second golden rule*** [8],[9].

$$\Gamma_{if} = \int \rho(E_i)\Gamma'_{if}\ dE_i = \frac{2\pi}{\hbar}|M_{if}|^2\rho(E_f) \tag{2.6}$$

In the case of unstable systems, their lifetime, λ, relates to the transition rate as $\lambda = 1/\Gamma$. Where we should consider that unstable systems could decay in many different final states, also called decay modes. If this was the case, the total transition rate for a finite set of final states $\{|f'\rangle\}$ would be:

$$\Gamma \ = \ \sum_{f'} \Gamma_{if'} \tag{2.7}$$

Another important concept to define is the so called **Branching Ratio** (equation 2.8), which is the relative rate of an specific transition, from state $|i\rangle$ to a state $|f\rangle$, with respect to the total transition rate.

$$Br(|i\rangle \to |f\rangle) \ = \ \frac{\Gamma_{if}}{\Gamma} \tag{2.8}$$

This relative transition rate actually determines the probability for the particle to decay in one specific decay mode; if N particles are considered then, it could be expected $N_{decayed}$ decays, from which N_{if} is the number of decays in the specific mode, hence the probability of that decay.

$$Br(|i\rangle \to |f\rangle) \ = \ \frac{N\Gamma_{if}}{N\Gamma} \ = \frac{N_{if}}{N_{decayed}} \tag{2.9}$$

2.4 The Kaon

Close to the year 1943 Leprince-Ringuet and collaborators employed cloud chambers in the French Alps for studying cosmic rays. In the process, they found evidence of particles with mass between 870 and 1110 times the electrons mass. This particle was initially proposed as a meson whose mass was greater than the pions mass. Nevertheless, this idea wasn't universally accepted due to the uncertainties in the measurements, which raised the possibility of having measured the protons mass, thus possibly misidentifying it. It was not until 1947, with the work of George Rochester and Clifford Butler, that the proper identification of these "heavy" particles, the positively charged kaons, was achieved[10]. The kaons are a set of four different mesons (in their base state); all of them are characterized by having a quark strange, hence only decaying weakly because all other interactions preserve particle flavor. On table 2.4 the lifetime of the different Kaons, as well as measurements of the Branching Ratios of some decay modes are displayed. The first four rows of it show the main decay mode (prohibited by $\mathbb{C}$ in the SM) of this study (violet), as well as the most relevant background decay modes (blue).

Kaon type	**Lifetime [ns]**	**Decay mode**	**Branching Ratio**
K^+	12.379 ± 0.021	$\pi^+\gamma\gamma\gamma$	$< 1.0 \times 10^{-4}$, 90% CLs
		$\pi^+\pi^0(\gamma)$	$(20.67 \pm 0.08) \times 10^{-2}$
		$\pi^+\pi^0\pi^0$	$(1.760\pm) \times 10^{-2}$
		$\pi^+\gamma\gamma$	$(9.61 \pm 0.15 \pm 0.07) \times 10^{-7}$
		$\mu^+\nu_\mu$	$(63.56 \pm 0.11) \times 10^{-2}$
		$\pi^+\pi^-\pi^+$	$(5.583 \pm 0.024) \times 10^{-2}$
K_S^0	0.08954 ± 0.0004	$\pi^+\pi^-$	$(69.20 \pm 0.05) \times 10^{-2}$
		$\pi^0\pi^0$	$(30.69 \pm 0.05) \times 10^{-2}$
		$\pi^+\pi^-\gamma$	$(0.179 \pm 0.0004) \times 10^{-2}$
K_L^0	50.99 ± 0.21	$\pi^\pm e^\mp \nu_e$	$(40.55 \pm 0.11) \times 10^{-2}$
		$\pi^\pm \mu^\mp \nu_\mu$	$(27.04 \pm 0.07) \times 10^{-2}$
		$\pi^0\pi^0\pi^0$	$(19.52 \pm 0.12) \times 10^{-2}$
		$\pi^+\pi^-\pi^0$	$(12.54 \pm 0.05) \times 10^{-2}$

Table 2.4: Main Decay Modes And Lifetimes Of Kaons.

.The decay products for the K^- are the charged conjugate of the K^+. More measurements of the kaons properties as well as other particles information can be found in [5].

In the production of neutral kaons, K^0 and $\bar{K^0}$, they are always found as a superposition of both particles, known as K_s^0(K short) and K_L^0 (K long). Their names are related to their respective lifetimes one bigger than the other, as can be appreciated in table 2.4. The superposition of both particles is shown in equation 2.10, where the mixing parameter ϵ is a complex scalar whose consequence is a violation of $\mathbb{CP}$ symmetry.

$$\begin{aligned} |K_1^0\rangle &= \frac{1}{\sqrt{2}}(|K^0\rangle - |\bar{K^0}\rangle) \\ |K_2^0\rangle &= \frac{1}{\sqrt{2}}(|K^0\rangle + |\bar{K^0}\rangle) \\ |K_L^0\rangle &= \frac{1}{\sqrt{1+|\epsilon|^2}}(\epsilon\,|K_1^0\rangle + |\bar{K_2^0}\rangle) \\ |K_S^0\rangle &= \frac{1}{\sqrt{1+|\epsilon|^2}}(|K_1^0\rangle + \epsilon\,|\bar{K_2^0}\rangle) \end{aligned} \tag{2.10}$$

Chapter 3

The NA62 Experiment

The NA62 experiment is located in the north area of the SPS (Super Proton Synchrotron), in the facilities of CERN. The experiment collaboration is constituted by 30 institutes from 15 different countries, with 278 participants. It's an experiment capable of producing decay-in-flight measurements of Kaons, and was designed with the main objective of measuring the branching ratio of the decay mode $K^+ \to \pi^+ \nu\bar{\nu}$, within a 10% precision of the theoretical prediction of $(0.84 \pm 0.10) \times 10^{-11}$. For achieving its purpose, it's necessary to collect on the order of 100 events, which implies, assuming an acceptance of 10% in the whole experiment, an exposition to 10^{13} decays, from which about 10^{12} events coming from different decay modes would be neglected. For doing so, angular and momentum resolutions of the experiment must be in the order of 0.0060 *mrad* and 1% respectively. Data taking began in the year 2016 and is expected to end in 2026. In 2025, with the collected data until 2022, a branching ratio of $(13.0^{+3.3}_{-3.0} \times 10^{-11})$ with a significance above 5σ was reported, so far becoming the smallest measured branching ratio with this significance [11].

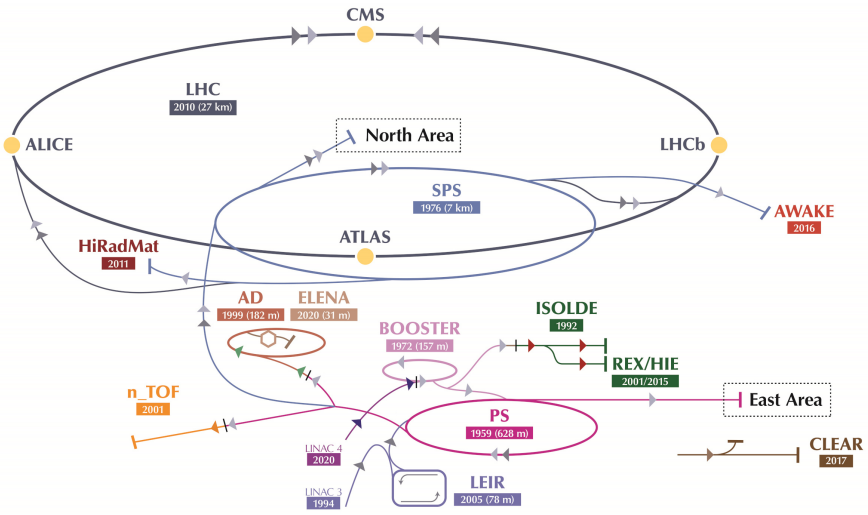

Figure 3.1: Experiments And Accelerators Layout At CERN.

El Conseil Européen pour la Recherche Nucléaire (CERN) known in English as The European Organization for Nuclear Research, was established in 1954 in Meryn at the border between France and Switzerland. Since its foundation, it has been home of many leading edge experiments in High Energy Physics, producing fundamental results such as the discovery of Higgs Boson in 2012 [12].

3.1 Experiment Beams

The experiment has two beams; the primary beam consists of protons with a nominal momentum of 400 GeV/c coming from the SPS accelerator. It collides with a fixed target made of beryllium. This target is denoted as T10 and is the origin of the reference system of the experiment. Its dimensions are 400 mm large and 2 mm of diameter, and is located in the tunnel which connects the SPS with the underground facilities of the experiment. The Z axis of the reference system is defined with the straight line joining T10 with the center of the LKr calorimeter. After the collision of the protons with the beryllium nucleus, secondary particles are produced such as pions and kaons. From this, a secondary beam is produced (K12) with a central momentum of 75 GeV/c; this value was chosen to maximize the amount of produced kaons with respect to other particles (70% pions, 23% protons and 6% kaons).

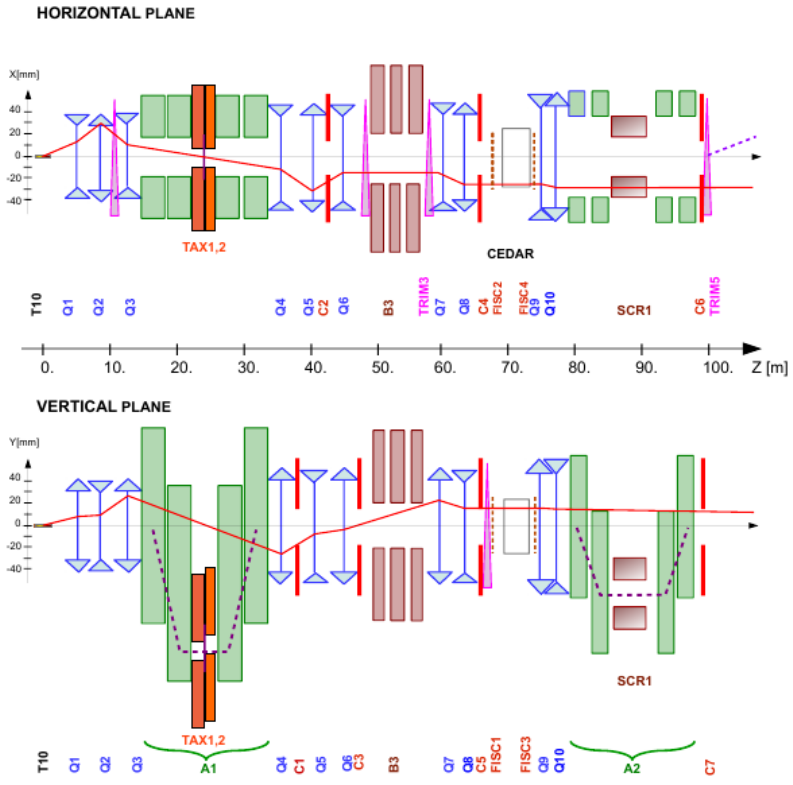

Figure 3.2: Scheme Of The Optics And Layout Of The High Intensity K^+ Beam. [13]

After T10, a copper collimator is found, it is 950 mm large, and has a variety of apertures with different diameters, from which the 15 mm diameter one is generally employed. With it, a high intensity beam is formed, and passes through a triad of high intensity quadrupoles (Q1,Q2,Q3) which collect a width acceptance in the solid angle (± 2.7 $mrad$ horizontally and ± 1.5 $mrad$ vertically). Afterwards there's

an achromat (A1) that selects with a 1% rms the 75 GeV/c momentum beam. It consists of four magnetic dipoles that deflect vertically the particles. The first pair produce a downwards parallel displacement of 110 mm, while the other pair restores the beam to its original axis. In between both pairs of magnetic dipoles, a set of graduated holes are mounted in beam dump mobile units, TAX1 and TAX2. The particles with the desired momentum pass through the specified whole while the rest are absorbed in the units, this way producing the momentum selection of the beam. Between both TAX, a radiator consisting of 5 mm plates of tungsten is found; its purpose is to slow down produced Bremsstrahlung positrons that are further discarded while minimizing the loss of produced hadrons. Then another triad of magnetic quadrupoles is found (Q4,Q5,Q6) which focuses the beam in the vertical plane while limiting its width in the horizontal plane. The drag region between these quadrupoles is occupied by the collimators (C1,C2), they redefine the horizontal and vertical acceptance of the transmitted beam. Subsequently, C3 collimator readjusts the beam in the vertical plane, while absorbing the slowed down positrons. Afterwards the beam passes through a 40 mm diameter hole within iron plates inserted in between the poles of three magnetic dipoles of 2 m large (B3), which help reduce the amount of muons in the beam. The deviation produced by B3 is then corrected by two dipoles (TRIM22,TRIM3). Then there is a differential Cherenkov Counter (CEDAR), it is equipped with an array of 8 photo-detectors (KTAG) which has as purpose the identification of the kaons. After CEDAR, a pair of magnetic quadrupoles are placed (Q7,Q8) with a pair of collimator (C4,C5) which absorb the particles in the tail of the beam. Also the filament scintillator counters (FISC1,FISC2,FISC3,FISC4) are employed to measure and adjust the divergence of the beam to zero. Then the magnetic quadrupoles (Q9,Q10) prepare the beam for the following and measurement of the momentum stage GTK. This stage consisted of 3 stations (GTK1,GTK2,GTK3) until the experiment update that occurred during 2019-2021. These stations consist of silicon pixel detector and are installed in the beams trajectory. Between GTK1 and GTK3 the achromat (A2) is located, and it consists of a magnetic collimator made of iron (SCR1) and four magnetic dipoles in C shape, these last deflect the beam vertically. After the GTK3 collimators (C6,C7) intercept background outside the beams acceptance. At last, TRIM5 deflects the beam horizontally in the X axis at an angle of 1.2 $mrad$, this deviations is then corrected by the magnetic spectrometer MNP33 in -3.6 $mrad$, so the beam passes through the center aperture of the LKr.

3.2 Detectors

For identifying the particles, and their trajectories, it is necessary to make use of different detectors such as calorimeters, spectrometers, Cherenkov detectors, among others. In the case of NA62, its detectors were designed and positioned with the objective of measuring the decay $K^+ \to \pi^+ \nu\bar{\nu}$ with the highest precision as possible, yet it isn't limited only to this decay mode. The arrangement of the detectors is shown in figure 3.3, the intention of the following sub-chapters is to give a description of the detectors employed in the experiment up to the year 2020. After this year, more detectors were incorporated into the experiment.

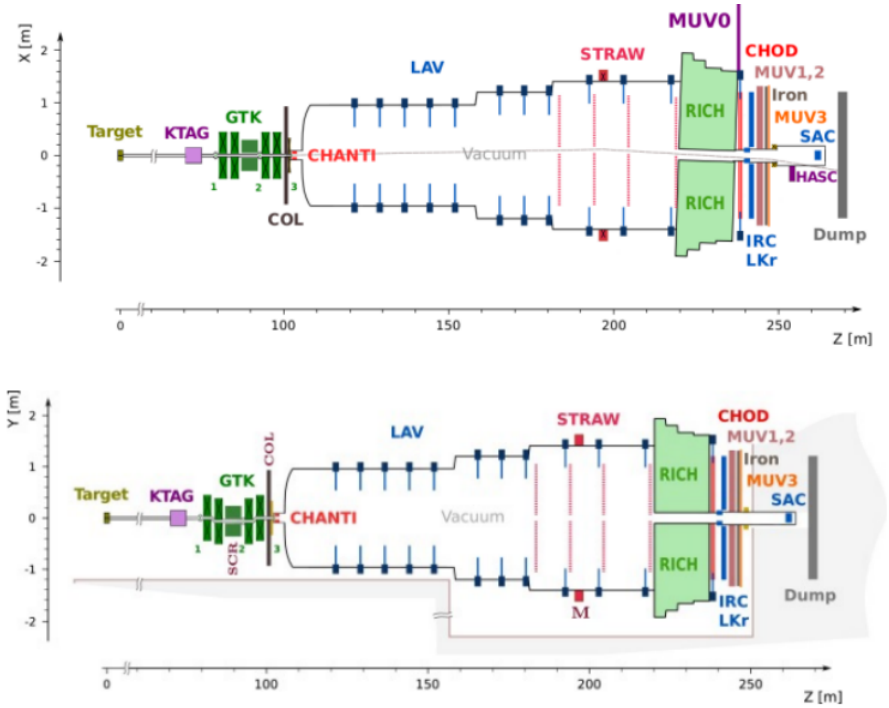

Figure 3.3: Scheme Of The Experiment Layout.

The upper part of the figure shows the XZ plane of the experiment, while the lower part shows the YZ plane. It's important to notice that for the X and Y axis the scale is in the order of some meters, while the Z axis in in the order of hundreds of meters [13].

3.2.1 Kaon Tagger (KTAG)

The function of the KTAG is identifying the produced kaons, it achieves this with a time resolution of 70 ps, and an efficiency above the 98% when there's a light coincidence of 5 sectors. For doing so, it uses the emission of Cherenkov light produced inside the CEDAR. The CEDAR is a Cherenkov differential counter, with achromat focusing rings. It's filled with 0.94 m^3 of nitrogen at 1.75 bar at room temperature. The gas pressure was selected so only light coming from the desired particle passes through a ring-shaped diaphragm with fixed central radius. The light is afterwards focused to 8 quartz windows and is subsequently focused into 8

spherical mirrors. This light is then reflected to 8 boxes, denoted as sectors, filled with nitrogen, each consists at its entrance of a light guide formed with a 64 conic sectors matrix, which is aligned with 48 photo-multipliers (the 16 remaining cones in the edge aren't intrumentalized). KTAGs parts can be appreciated in figure 3.4.

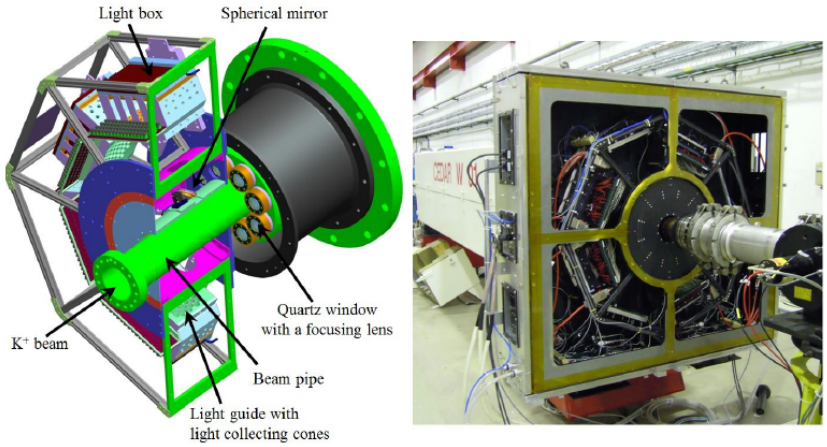

Figure 3.4: Scheme of KTAGs Composition And Parts.

At the left side a digital sketch of the KTAG and CEDAR are shown, while in the right side a photo taken in 2012 of both detectors is shown. Notice that in the photo, 4 of the 8 sectors are mounted, and can be seen in the left side of the detector [13].

3.2.2 Beam Spectrometer (GTK)

The GTK is a set of three silicon detectors which measure the time, direction, and momentum of the particles in the beam. The spectrometer is inside the vacuum pipe where the beam particles fly; it measures the momentum of the particles by measuring the vertical displacement of them after passing through four magnetic dipoles, see figure 3.5. This detector is capable of measuring the momentum of the beam particles with a precision of 0.2%, and it's also capable of measuring their direction dX/dZ and dY/dZ at the exit of the achromat with a precision of 16 μrad. The time resolution of the detector is 150 ps.

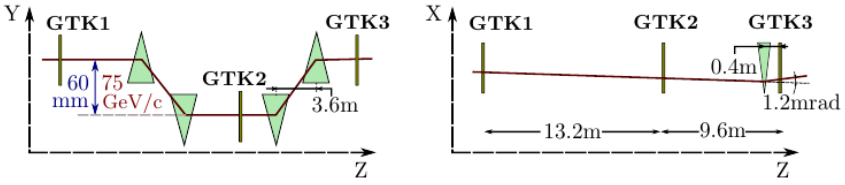

Figure 3.5: Scheme With The Position Of The GTK Stations.

At the left side, the YZ plane of the GTK is shown, the magnetic dipoles are represented with the green triangles. At the right side, the XZ plane view is shown.[13]

The three stations are formed by silicon plates of 500 μm width, which form a matrix array of 18000 pixels (200×90) in an area of 62.8×27 mm^2. The lecture of

the matrix is done by five application-specific integrated circuits ASIC. One of the stations can be seen in figure 3.6.

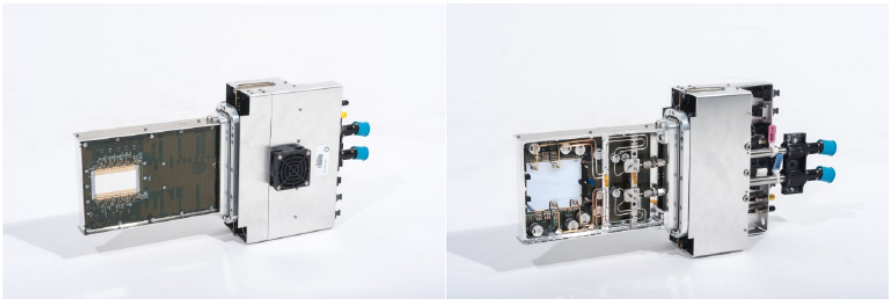

Figure 3.6: Photograph Of One GTK Station.

At the left side, the silicon matrix which is pierced by the beam can be appreciated (white rectangle). At the right side, the readout electronics and cooling system are displayed [13].

3.2.3 Charged Anti-coincidence Detector (CHANTI)

The function of the CHANTI is to reject the background events, produced by inelastic interactions of the beam with the last station of the GTK (GTK3), or by a muon halo produced prior to GTK3. It's composed of 6 stations of squared-shaped hodoscopes, with a cross section of $300 \times 300\ mm^2$ with a centered cavity of $95 \times 65\ mm^20$ where the beam passes through. The first station is located at 28 mm upstream from GTK3, the distance between stations is approximately double for successive stations. Therefore, achieving a hermetical angular coverture of 49 $mrad$ to 1.3 rad. The hodoscopes were formed with 48 scintillation bars based on polystyrene, which form the readout planes X and Y. The produced light is conducted with optic fibers to their corresponding electronics, which will then process the signals.

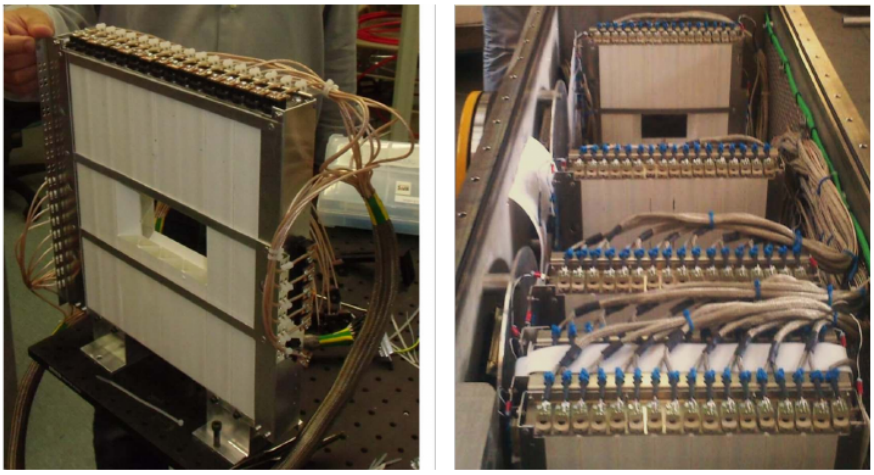

Figure 3.7: Photograph of CHANTI Stations.
[13]

3.2.4 Straw Spectrometer (STRAW)

The TRAW spectrometer measures the trajectories and momentum of the charged particles. It has an extension of 35 m along the beam line. It has four straw chambers, and in between them an aperture magnetic dipole (MNP33) which provides a magnetic field of $0.9T$. Each chamber consists of two modules, one of them contains the view for the X,Y planes $(0°, 90°)$ respectively, while the other module contains the views of the U,V plains $(-45°, 45°)$. The active region of the chambers is a circle of 2.1 m of diameter centered in the Z axis, once overlapped the views, a 6 cm apothem octagon is obtained. Each of the views has a gap of 12 cm without straws close to the center, creating thus a hole that has as purpose to let the beam pass through, see figure 3.8. Due to that it has an angle of 1.2 $mrad$ and $-.6$ $mrad$ of aperture in the horizontal plane after and before the magnet, it is not centered in the Z axis. Yet, it has compensations along the X direction for each chamber. Each chamber contains 1792 straws with a 9.82 mm diameter and a large of 2160 mm, and each of the views is conformed with 448 straws. Every straw is filled with a mixture of 70/30 Ar and CO_2 respectively, the container consist of a layer of 50 nm copper, 36 μm PET, and an inner gold layer of 20 nm. The spatial resolution of the chambers is ± 200 μm. For trajectories crossing the four chambers, the time resolution corresponds to 5.1 ns.

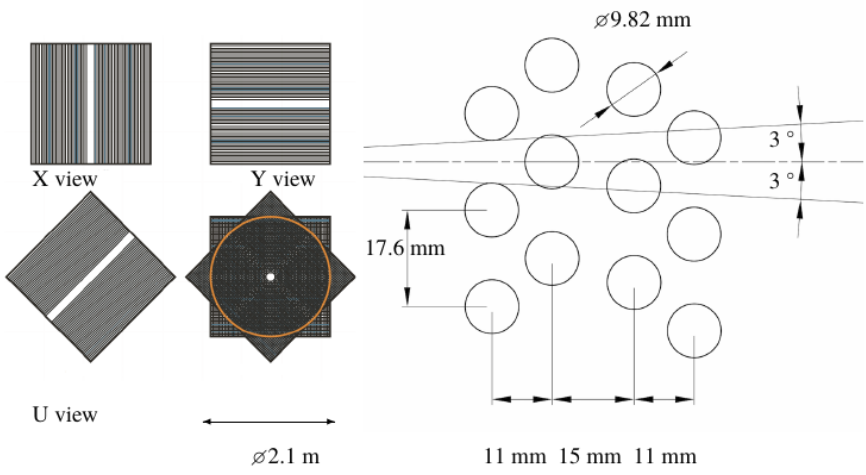

Figure 3.8: Disposition Of Views For The STRAW Chambers, And Position Arrangement Of The Straws.

At left the disposition of the views (X,Y,U,V) of the chambers is displayed with their respecting holes without straws. At right the space and position of the straws in each of the views is shown, this space helps resolve the ambiguity of the left or right way of each trajectory [13].

The MNP33 is a magnetic dipole, whose cross-section dimensions are 4.4 m width, 4 m high and 1.3 m large, and weights 105 tons. The formed magnetic field points mostly to the negative Y axis, and the rest of the components have a smaller magnitude by a factor of 10^{-3}. This field gives a transversal momentum kick to the particles of 270 MeV/c, deflecting the trajectory of the charged particles. With this, measuring the momentum of the particles is plausible, by measuring the de-

flection angle formed with the trajectories after and before the magnetic field. The momentum resolution is consistent with equation 3.1, where the momentum units are GeV/c. The obtained angular resolution is 60 μrad for a 10 GeV/c track and 20 μrad for a 50 GeV/c track.

$$\frac{\sigma(p)}{p} = 0.30\% \oplus 0.005\% p \tag{3.1}$$

3.2.5 Photon Veto System

The main objective of the experiment consists on measuring the branching ratio of $K^+ \to \pi^+ \nu\bar{\nu}$, which has as main background the decay mode $K^+ \to \pi^+\pi^0$. The π^0 has a lifetime in the order of 10^{-17} s, and it mainly decays into a pair of photons. By applying kinematic cuts, it is possible to reject in the order of 10^4 events coming from this background source, while an order of 10^8 events can be rejected with the photon veto system. It consists of four subsystems Large-Angle Veto System (LAV), the Liquid Krypton calorimeter (LKr), Intermediate-Ring Calorimeter (IRC), and the Small-Angle calorimeter (SAC). The major part of the veto is done by the LKr, the LAV handles the detection of photons emitted in angles greater than does that can be detected by the LKr, while the SAC and IRC handle the detection of highly energetic photons emitted in sufficiently narrow angles to cross along the beam pipe.

3.2.6 Large-Angle Veto System (LAV)

The LAV consists of 12 ring stations formed with blocks of lead-glass Schott FS57, which consists of approximately 75% lead oxide. In each block a Hamamatsu R2238 photo-multiplayer of 76 mm is coupled with a cylindrical optic guide of 4 cm large made of FS57. In figure 3.9 one of the blocks can be seen, while in figure 3.10 two LAV stations can be seen. The angular cover of the LAV is 8.5 $mrad$ up to 50 $mrad$ with respect to the Z axis. The LAV inefficiency is in the order of 10^{-4} for photons with energy above the 200 MeV, which are the 95% of the produced photons within the acceptance of the detector.

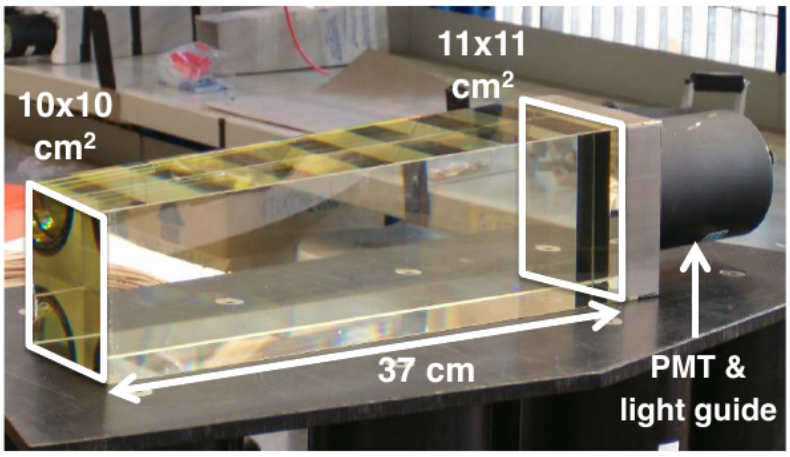

Figure 3.9: FS57 Lead-glass Block.

In the image is shown the blocks that constitute the LAV rings. This blocks were recycled from OPAL in 2007, their precise dimensions vary from station to station [13].

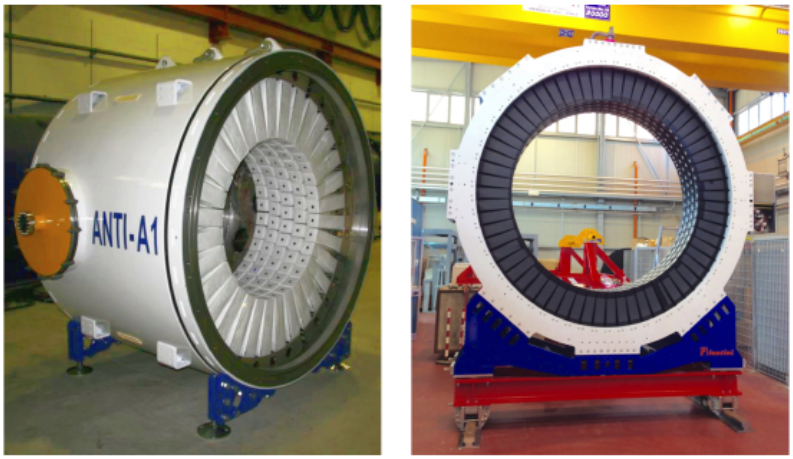

Figure 3.10: LAV Stations 1 and 12.

At right the first LAV station can be seen, while at left the 12-th station is shown. Stations 1 to 11 operate within the fiducial volume, while the last station is located 3 m upstream from the LKr [13].

3.2.7 Liquid Krypton Calorimeter (LKr)

The LKr is a calorimeter whose function is not only the detection of photons, but also the identification of particles. It is composed of 9000 l of krypton at $120K$ contained in a cryostat. It is extended from the beam pipe ($\approx$ / cm) up to a radius of 128 cm, and has a depth of 127 cm. The sensitive area of the detector consists of 13248 longitudinal cells with a cross section of 2×2 cm^2. These cells are formed with electrodes in a zig-zag shape, made from copper-beryllium aligned along the Z axis. The signals produced by the charged particles crossing it are sent to amplifiers which are inside the cryostat. This detector has an angular cover with respect to the Z axis of 1 $mrad$ up to 8.5 $mrad$. Its inefficiency for photon detection is in the order of 10^{-3} for photons with energy above 1 GeV, and it's in the order of 10^{-5} for those with an energy above 10 GeV.

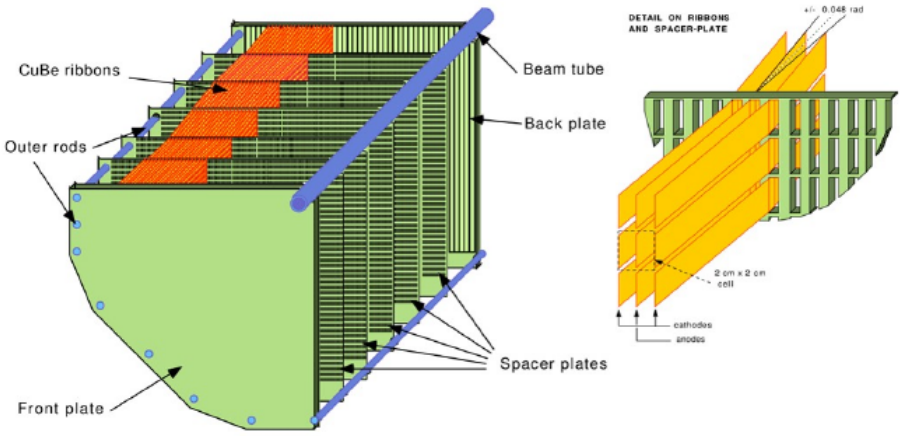

Figure 3.11: Scheme Of Quarters And Cells Of The LKr. [13].

3.2.8 Small-Angle Calorimeter (SAC)

The SAC is constituted by 140 plates, 70 of them are made of lead while the other 70 are made of a scintillation plastic. Its dimensions are $205 \times 205\ mm^2$ of cross-section, with a depth of 1.5 mm. Each of the plates has 484 holes of 9.5 mm of diameter, spaced 1.5 mm from one an other in a rectangular grid. The plates are piled in such a way that after a lead plate there's light diffuser plate made of Tyvek, followed by a scintillation plate. The produced light is conducted with optic fibers, placed inside the holes, in a U-shaped way, thus allowing the readout of both extremes of the detector. The fiber optics are then grouped in four sets which are then connected to four photo-multipliers. Due to the fact that the scintillation plates have no transversal segmentation, the four readout channels coming from the photo-multipliers are optically connected, and hence the SAC ends up having just one readout channel. The position of the detector in front of the beam pipe allows it to detect photons between the 0 $mrad$ up to 1 $mrad$, and counts with an inefficiency of 10^{-4} for photons with energy above the 5 GeV.

3.2.9 Intermediate-Ring Calorimeter (IRC)

The IRC is a Shashlyk calorimeter made of lead, as its absorbing material, and scintillators, shaped in a cylinder surrounding the beam pipe upstream from the LKr. Its external diameter is 290 mm, while its central hole has 120 mm of diameter. This last has a shift of 12 mm in the positive X direction, this way taking into account the deflection of the beam due to the magnetic field of the spectrometer. The detector is divided in two longitudinal modules of 89 mm and 154 mm depth, both separated 40 mm from one an other, as can be observed in figure 3.12. The modules have 25 and 45 ring layers made of 1.5 mm of lead/antimony 97/3 and 1.5 mm of Saint Gobain BC-400 scintillation material covered with reflective paint. Each ring has 570 holes of 1.5 mm. The scintillation discs are divided in four quadrants optically isolated. The light signals generated in the material are taken through fiber optics to four photo-multipliers which give readout for each of the four quadrants. Its inefficiency is in the order of 10^{-4} for photons with energy greater than 5 GeV.

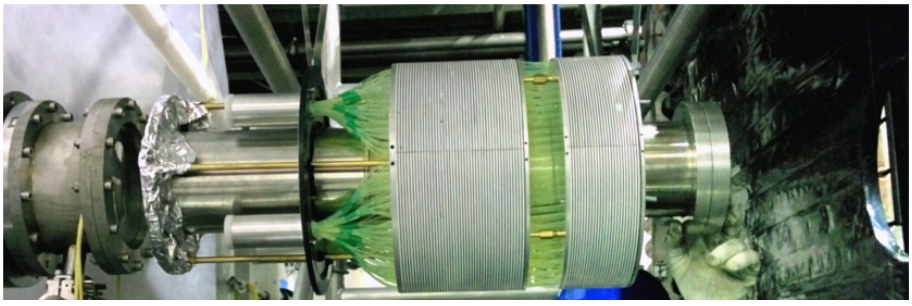

Figure 3.12: Intermediate-Ring Calorimeter.

In the image both modules conforming the IRC can be observed, as well as the green fiber optics whom conduct the light produced in the scintillation material. In this case, the light will go from left to right passing through the center of the detector [13].

3.2.10 Ring Image Cherenkov Counter (RICH)

The purpose of the RICH is the identification of charged particles. It consists of a cylindrical radiator of 17.5 m long, which is sectioned in four sectors of variable diameter, having the sector from which the light enters a diameter of 4.2 m, while its last section has a diameter of 3.2 m. The radiator is filled with neon at 990 $mbar$ of pressure and it is in charge of producing the Cherenkov light. At the final extreme of the tank can be found a mosaic of 20 spherical mirrors whose function is to transform the produced light cones into rings which fall into a photo-multipliers array, see figure 3.13 for a visual reference. The mirrors have a focal distance of 17 ± 0.20 m and have a reflectivity around 90% for wavelengths in $(195, 650)$ nm. The orientation of each mirror can be adjusted by a pair of piezoelectric actuators. The photo-multipliers array is made with 1952 Hamamatsu R7400 U-03 units, with a step between photo-sensors of 18 mm. The active region of the detector is 1.1 m from the beam axis in the radial direction in the extreme corresponding to the beam's entrance, while in the other extreme is 1.4 m. The time resolution of this detector is 100 ps, and in particular for positrons is 74 ps.

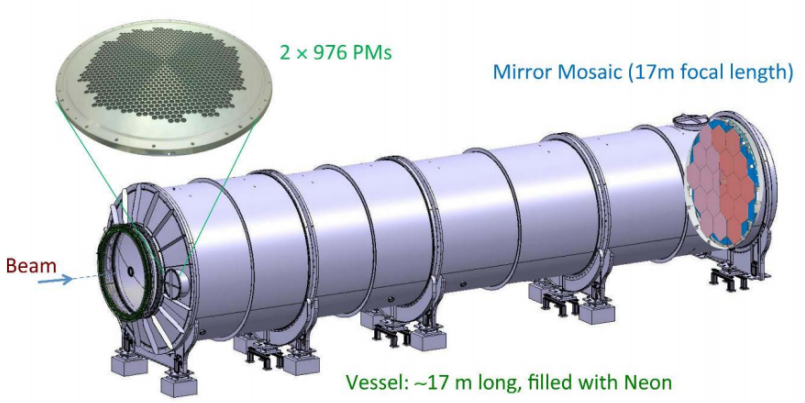

Figure 3.13: Schematic View Of The RICH.

The beam passes in this image through the detector from left to right. At the right side the mounted mirrors can be observed, while in the other extreme the placement of the photo-multipliers can be appreciated [13].

3.2.11 Charged Particles Hodoscopes (CHODS)

The experiment has a scintillation system for the detection of charged particles, which consists of a pair of hodoscopes, one coming from a previous experiment NA48, the other one was specifically created for NA62. The first one is placed right before LAV12, while the last one is placed right after the LAV12. This system takes care of the annular region between the RICH and the LKr, this region has an inner radius of 145 mm and an outer radius of 1070 mm. Its main function is to generate a signal for trigger L0, and give a time signal to charged particles. The NA48 CHOD is constituted by 94 vertical planes and 64 horizontal ones of BC408 scintillation plastic optically connected to PHOTONIS XP2262B photo-multipliers. It consists of 128 counters assembled in four quadrants of 16 tiles in each plane. The NA62 CHOD, also called New CHOD, consists of 152 scintillation plastic tiles with

a depth of 30 mm. The gathered light os transmitted using Kuraray Y11S fiber optics, which are connected to pairs of SensL SiPMs of 3×3 $mm^2$2. The produced signals are then amplified and processed by specialized electronics.

3.2.12 Hadron Calorimeters (MUV1,MUV2)

The MUV1 consists of 24 layers of SE35 steel of 26.8 mm depth, spaced 12 mm, and having 22 of them (the inner layers) 2700×2600 mm^2 of area. Among them, 23 layers of scintillation bars of 9 mm depth and 60 mm width are found, except for four (the closest to the beam) which have a width of 54 mm. The orientations of the bars are horizontal and vertical, and are disposed in an alternating way, ending with 11 vertical layers, and 12 horizontal ones. The majority of the bars have a length of 2620 mm and cover the whole transversal area of the detector. Each bar has two conduits coupled to optic fibers which are connected to photo-multipliers. Each photo-multiplier gets signals from fibers in the same transversal position, ending up with 176 readout channels for the detector.

The MUV2 is a calorimeter similar to MUV1, it consists of 24 steel plates of 25 mm depth. Among them are placed scintillation layers of ELJEN NE 110 plastic. Each scintillation layer is conformed with 44 bars of 1300 mm large, 119 mm width, and 4.5 mm depth. As MUV1, the bars with the same alignment are connected to the same photo-multiplier, giving in total 88 readout channels. Its resolution was measured with data in 2015 and is given by equation 3.2, where the employedd units are GeV.

$$\frac{\sigma(E)}{E} = 0.115 \oplus \frac{0.38}{\sqrt{E}} \oplus \frac{1.37}{E} \tag{3.2}$$

3.2.13 Fast Muon Veto (MUV3)

The main function of this detector is muon identification. It is situated after the hadron calorimeter right after a steel plate of 80 cm depth. It presents a cross-section of 2640×2640 mm^2, and is made with 148 scintillation tiles of 50 mm depth, having 140 of the dimensions of 220×220 mm^2, the rest, closer to the beam pipe have smaller dimensions. The front and lateral faces of the tiles are covered with Mylar, the remaining face is joined to a light hermetic box, which conduces light to two photo-multipliers. 280 are EMI19814B model while 16 are Philips XP 2262. The overall efficiency exceeds the 99.5% for muons with a momentum above the 15 GeV/c. The time resolution was measured with 2015 data, and it's between the 0.4 ns and 0.6 ns.

3.2.14 Peripheral Muon Veto (MUV0)

The MUV0 is a hodoscope designed for the detection of π^- with momentum below the 10 GeV/c. It is found at the edge of the RICH, in its final extreme. Its purpose is to detect pions which are deflected by the spectrometers magnet but don't cross the RICH. It consists of two layers of 48 scintillation tiles with dimensions of $200 \times 200 \times 10mm^3$. They are located and grouped in "Super Tiles" consisting of 9 tiles, see figure 3.15. The emitted light is conducted through optic fibers to Hamamatsu R7400 photo-multipliers.

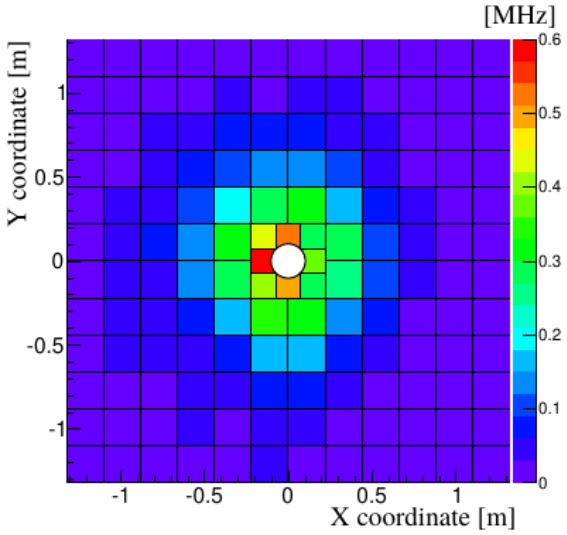

Figure 3.14: Expected Frequencies For MUV3's Tiles.

In this image the expected signals for the MUV3 tiles with a nominal beam intensity is shown. The red tiles correspond to a frequencies of 3.2 MHz and the main contribution comes from pions decays.

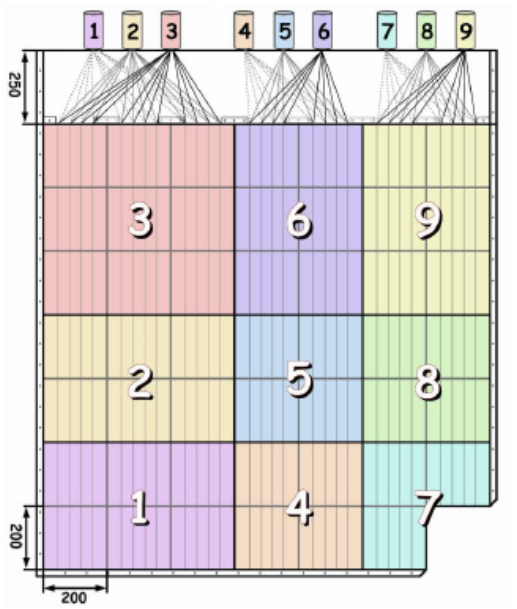

Figure 3.15: MUV0 "Super Tiles".

It is possible to observe in this image the disposition of the tiles forming the 9 Super Tiles of MUV0.

3.2.15 Hadronic Sampling Calorimeter (HASC)

The HASC is a detector whose purpose is to detect pions with a momentum above the 50 GeV/c that propagate through the central holes of the STRAW. It is localized after the MUV3 and the magnetic dipole BEND, and covers a region from $-0.48\ m < X < 0.18\ m$, $|Y| < 0.15\ m$. It consists of 9 identical modules, which have as active elements 60 plates of 16 mm lead, intercalated with scintillation plates of 4 mm depth. Both types of plates have transversal dimensions of $100 \times 100\ mm^2$. Each module is organized into 10 longitudinal readout sections. The produced light is conducted through optic fibers to SiPM sensors of $3 \times 3\ mm^2$ coupled to amplifier boards.

3.3 Trigger And Data Adquisition System (TDAQ)

Due to the intense flux of the experiment, a trigger and data acquisition system that minimizes the standby time between measurements while maximizing the reliability of data collection was necessary. For this purpose, the TDAQ was designed, consisting of 3 trigger systems. L0 which corresponds to hardware triggers, L1 and L2 which are software triggers. The estimated number of decays in the experiment is 10 MHz, L0 produces a downscale to 1 MHz, while with the other two triggers, a 10 KHz downscale is achieved.

3.3.1 L0 Trigger

This trigger system employs the signals produced by a group of fast detectors to conduct an initial filtering of events. The initial implementation consists of the following detectors:

- CHOD: It provides the positive primitives for any charged trajectory; They are based on the multiplicity of hits and their reference time.
- RICH: It provides the positive primitives for any charged trajectory that overcomes the Cherenkov threshold, based on the multiplicity of hits.
- LAV: It provides primitives for muon or photon vetoing; They are based on hits of adjacent blocks.
- MUV3: It provides primitives for muons, either positive ones or for vetoing logic; They are based on the multiplicity of tiles.
- Calorimeters (LKr,MUV1,MUV2): They provide primitives for positive identification of pions; They are based on the energy deposition and the multiplicity of the formed clusters in the LKr.

The obtained primitives are produced asynchronously, in a variable time not greater than 100 μs. Each of them consists in a 64 $bits$ block that contains a time below 25 ns and an identification that specifies the conditions achieved in-time. One created these primitives, L0TP (L0 Trigger Processor) finds in-time coincidences, with a time window of 25 ns or 100 ps for fine time measurements.

3.3.2 High Level Triggers L1 And L2

The L1 and L2 triggers employ software to produce a downscale of 10 MHz and 10 KHz respectively. The L1 trigger is based on algorithms that take information from the detectors in an individual way, while the L2 trigger takes partially reconstructed events and employs correlations between detectors.

3.4 NA62 Framework

For the data analysis as well as creating Monte Carlo simulations, the NA62 collaboration has produced a programming framework based on Geant4[14], ROOT[15] and C++17[16]. This framework consist in the following four modules:

- **NA62Tools:** It contains the code employed for the rest of components; in particular, the persistency libraries and the CDB (condition data base) access code.
- **NA62MC:** It is the framework based on Geant4 for the simulation of the experiment.
- **NA62Reconstruction:** It is a reconstruction package based on ROOT, which is modularized in libraries for each subdetector.
- **NA62Analysis:** It is a framework based on ROOT for the modularized development of simple or complex analysis. It contains tools and the necessary examples for a width range of analysis including calibration and corrections code.

For technical details, as well as examples on how to use the NA62 framework please have a look at its web page [17].

Chapter 4

Analysis Strategy

4.1 Global Overview Of The Analysis

As stated in the introduction, the main goal of the analysis is to determine whether it is feasible or not to detect the decay mode $K^+ \to \pi^+\gamma\gamma\gamma$, which is forbidden by $\mathbb{C}$ conservation in electromagnetic processes in the SM. For achieving this task, an analysis strategy had to be implemented not only to identify the decay mode by using information about its decay products; but also to distinguish it from all the other decay modes of the K^+.

Briefly saying, the analysis program should be able to identify the π^+, and its track (trajectory) and four-momenta. It should also identify the K^+ from which the π^+ comes from, as well as its track and four-momenta. Both trajectories should form an intersection vertex inside the beam line. The program should also identify the produced photons, and their four-momenta. With this data, it is possible to calculate the invariant mass of the reconstructed K^+. This mass hence becomes the main variable for the identification of the desired decay.

4.1.1 Invariant Mass Of Reconstructed Particles

For the identification of decay modes it is common to make use of invariant quantities which are conserved before and after the decay, and that are also invariant in any inertial frame, such as the laboratory or the center of mass of the K^+. Let us consider the conservation of the K^+ four-momenta, P_{K^+}, which is distributed among the four-momenta of the decay products: P_{π^+} for the pion and $P_{\gamma i}$ for the three photons, $i = 1, 2, 3$.

$$P_{K^+} = P_{\pi^+} + \sum_i P_{\gamma i} \tag{4.1}$$

From special relativity, the product of four-vectors in Lorentz Space is given by equation 4.2, where A and B are any kind of four-vectors and c stands for the speed of light in vacuum. From 4.2, it follows that the invariant mass of particles, is related to its four momentum by equation 4.3, E being the energy of the particle, and $\vec{p}$ its

momentum.

$$\begin{aligned} A &= (A_1, A_2, A_3, A_4) = (A_1, \vec{A}) \\ B &= (B_1, B_2, B_3, B_4) = (B_1, \vec{B}) \\ AB &= \frac{A_1 B_1}{c^2} - \vec{A} \cdot \vec{B} \end{aligned} \tag{4.2}$$

$$(Mc)^2 = P^2 = \frac{E^2}{c^2} - \vec{p} \cdot \vec{p} \tag{4.3}$$

Hence by combining equations 4.1 and 4.3 we can relate the four-momenta of the decay products with the invariant mass of the decayed particle, equation 4.4. Where M_{K^+} is the K^+ mass.

$$(c \;\; M_{K^+})^2 = (P_{\pi^+} + \sum_i P_{\gamma i})^2 \tag{4.4}$$

This indicates that by measuring the four-momenta of a set of particles, it is possible to determine whether they come from a K^+ or not. This method isn't unique for K^+ mesons and could be applied to any kind of decay process. However, in experiments, the finite accuracy of detectors, and statistical fluctuations, produce a distribution with a certain width around the nominal mass of the given particle, see figure 4.8 as an example; Therefore, in practice, determining whether the reconstructed mass corresponds to the searched decay relies on checking if it lies in an specified mass interval.

4.1.2 Use of Monte Carlo Simulations

The use of Monte Carlo simulations is essential for evaluating the performance of a given analysis program. For instance, as stated in section 3.4, the collaboration has developed tools for Monte Carlo simulations. In them, the response of the different detector systems and the particles crossing them are simulated. The produced simulation samples follow the experiment parameters from representative data collection periods called "Golden Runs", and are grouped by their corresponding run, decay mode, reconstruction version and special conditions (such as artificially introduced biases). It is worth mentioning that Monte Carlo simulation can be of type No Overlay or Overlay. The first only contains decays of kaons in an specific decay mode, while in Overlay samples, among the specific kaon decay mode other particles such as protons, pions as well as the 6 main decay modes of the kaon are added.

Often, decay modes with similar characteristics to the searched mode could be misidentified as desired events. In such cases, using Monte Carlo simulations becomes of great importance to determine the misidentification mechanism, since the initial and end positions, production mechanism, kinematic variables, among other parameters of the simulated particles are known. Furthermore, the analysis program can be run over simulation samples of isolated decay modes, hence allowing the direct identification of background contributions for a given identification program.

Acceptance

Given a sample of size N of events of an specific process, as could be a decay. The geometrical position of the employed detectors, their resolutions and the selection

conditions applied in the analysis program will diminish the amount of final observed events N_{obs}. The ratio of the remaining events and the total events is known as acceptance, see equation 4.5. If the acceptances for a given decay mode is known, then an estimation of the actual total number of events can be determined.
In principle the acceptance for simulations should be close to the actual experimental data acceptance, as long as the simulations replicate properly the experiment conditions. This is of great importance since usually for experimental data, the number of total events isn't known or easily obtained, and hence calculating the acceptance from it isn't feasible. In contrast, the total amount of events in simulations is well established and known, hence calculating the acceptance is most of the time a straight forward procedure.

$$A = \frac{N_{obs}^{exp}}{N_{exp}} = \frac{N_{obs}^{MC}}{N_{MC}} \tag{4.5}$$

4.1.3 Normalization And Background Estimation.

Equation 2.9 relates the total number of decays in a given time N, with the amount of decays in an specific decay mode, N_f. Then it follows immediately that for a sufficiently large N, the number of decays for each decay mode is related by equation 4.6, where $Br(f)$ stands for the branching ratio of each final state f.

$$N = \sum_f N_f = \sum_f N Br(f) \tag{4.6}$$

In the experiment, after applying selection criteria, a much smaller set of decays will pass the selection. The amount of decays for each mode remaining after the selection is determined by its respective acceptance A_f. Then equation 4.6 can by modified to express the amount of remaining events, S, as well as the remaining events for each mode S_f.

$$S = \sum_f N \cdot Br(f) \cdot A_f = \sum_f S_f \tag{4.7}$$

With no loss of generality, N can be calculated if the number of decays of an specific decay mode, N_n, is known with the help of equation 2.9; If selection conditions are applied equation 4.5 would also have to be employed, hence leading to equation 4.15. It is customary to called this mode the normalization channel. In principle any decay mode could be used as a normalization channel, yet employing a decay mode with similar characteristics as the searched one is preferred since first order systematic uncertainties are diminished due to the quotient. Also, selecting normalization channels with large branching ratios is helpful since it usually implies a greater amount of events passing the selection, hence reducing the overall statistic uncertainty and also improving the chance of having $S_n \neq 0$.

$$\begin{aligned} S &= \sum_f \frac{S_n \cdot Br(f) \cdot A_f}{Br(n) \cdot A_n} \\ &= \sum_f \frac{N_n \cdot Br(f) \cdot A_f}{Br(n)} \end{aligned} \tag{4.8}$$

Notice that for a given data set, an identification algorithm for the decay channel can be produced, with conditions such that background events from other channels diminish at a point in which they can be depreciated. The resulting filtered data can be consider to be composed exclusively from the decay channel, hence allowing the estimation of the number of decays from the original data set with equation 4.9.

$$N = \frac{S_n}{A_n \cdot Br(n)} \tag{4.9}$$

Consider the situation in which there is an identification algorithm for a certain decay mode. For a given experimental data set, the number of events corresponding to the searched decay and the number of background events must be known. In order to achieve this, a sample of simulated events in which different decay modes are distinguishable can be processed with the same algorithm. The output will replicate the signal-to-background proportion observed in the experimental data. Hence, an estimation of experimental results could be obtained by scaling the simulation results in such a way that the number of simulated decays, N', matches that of the experimental data set N. This procedure is called normalization.

As previously mentioned, the collaboration has prepared simulation samples for the different decay modes of the K^+. In order to normalized this samples with respect to the experimental data, first they should be scaled with respect to the simulated sample of the normalization channel. As notation, any quantity from now on obtained from Monte Carlo simulations will have an apostrophe next to it. Given the sample corresponding to the ith-decay mode, it will contain the number N'_i of decays, from equation 2.9, it can be stated that the sample comes from a set of $\bar{N}'$ K^+ decays; For the normalization channel, its corresponding number of K^+ decays will be N'. With this in mind, the scaling factor for each decay mode sample, β'_i, is given by equation 4.11.

$$\begin{aligned} N' &= \beta'_i \bar{N}' \\ \frac{N'_n}{Br(n)} &= \beta'_i \frac{N'_i}{Br(i)} \end{aligned} \tag{4.10}$$

$$\Rightarrow \beta'_i = \frac{N'_n}{N'_i}\left(\frac{Br(i)}{Br(n)}\right) \tag{4.11}$$

The obtained scaling factor can also be applied to the number of remaining events, $\bar{S}'_i$, after applying the identification algorithm. To demonstrate this, both members of equation 4.10 can be multiplied by $Br(i)$ and A_i. Thus, getting the following set of equations.

$$\begin{aligned} \frac{N'_n}{Br(n)} \cdot Br(i) \cdot A_i &= \beta'_i \frac{N'_i}{Br(i)} \cdot Br(i) \cdot A_i \\ N' \cdot Br(i) \cdot A_i &= \beta'_i \cdot N'_i \cdot A_i \end{aligned}$$

$$S'_i = \beta'_i \bar{S}'_i \tag{4.12}$$

Once calculated each S'_i, the next is normalizing the simulation samples with respect to the experimental data set. In a similar manner to equation 4.10, a quantity β

can be established. And with the help of equation 4.9 applied to both experimental and simulated data, and considering that acceptances are the same for experimental and simulated events, β can be calculated with equation 4.14.

$$\begin{aligned} N &= \beta N' \\ \frac{S_n}{Br(n)\cdot A_n} &= \beta \frac{S'_n}{Br(n)\cdot A_n} \\ S_n &= \beta \cdot S'_n \end{aligned} \tag{4.13}$$

$$\Rightarrow \beta = \frac{S_n}{S'_n} \tag{4.14}$$

With equations 4.7 and 4.13 it can be shown that by multiplying each scaled S'_i by β gives an estimation for S.

$$\begin{aligned} S &= \sum_f N \cdot Br(f) \cdot A_f = \sum_f (\beta N') \cdot Br(f) \cdot A_f \\ &= \sum_f \beta (N' \cdot Br(f) \cdot A_f) = \sum_f \beta \cdot S'_f \end{aligned} \tag{4.15}$$

Equations 4.14 and 4.15 are employed for normalizing the events coming from a Monte Carlo simulation with respect to an experimental data sample, which is useful to determine the amount of possible background in a given signal. Then, by employing the CL_s method briefly described in appendix A, an upper limit to the number of events of the searched signal, S_o, can be computed. With this value and with the help of equations 4.9, as well as the acceptances of the searched mode, A_o, and a normalization mode, A_N, the Branching Ratio of it, $Br_{(N)}$, and the number of events passing the selection for the normalization mode, S_N. An upper limit to the Branching Ratio of the searched mode can be computed using equation 4.16.

$$\begin{aligned} Br(o) &< \frac{S_o}{A_o \times N} = \frac{S_o}{A_o \times (\frac{S_N}{A_N \times Br(N)})} \\ Br(o) &< Br(N) \times \frac{S_o}{S_N} \times \frac{A_N}{A_o} \end{aligned} \tag{4.16}$$

4.1.4 Employed Data and Monte Carlo Simulation Samples

In the following subsection, the experimental and simulation data sets, as well as some important aspects of them are presented.

The employed experimental data consist on the data collected during 2017, this data is subdivided in four groups labeled as 2017A, 2017B, 2017C and 2017D. Each of them consist of different amounts of bursts, each burst consisting on one, two or three spills, which are six second lapses of active beam coming from the SPS. On table 4.1 the dates in which the data was taken, as well as the reconstruction conditions and number of burst are shown. A table of employed simulation samples can be found in table 4.2. In particular, the $K^+ \to \pi^+\gamma\gamma\gamma$ samples was produced assuming the matrix element is equal to one in equation 2.6.

A preselection filter was applied to these data sets before being employed. The corresponding filter is the PNN filter, which requires at list one track with momentum below 65 GeV/c, passing through the four STRAW chambers, with a reconstruction χ^2 below 30, associated geometrically to the CHOD, presenting an in-time signal in it, being also in-time (5 ns) with the KTAG. At last, the respective event should correspond to either the PNN trigger line or the Control trigger line, see [18] for trigger line details.

The corresponding normalization channels employed in this work correspond to $K^+ \to \pi^+\gamma\gamma$, and $K^+ \to \pi^+\pi^0\pi^0$. Both having their $\pi^0 \to \gamma\gamma$, which has a branching ratio of $(98.823 \pm 0.034)\%$ according to [5].

Data Group	**Dates**	**Number of Bursts**	**Reconstruction Version**
2017A	21.09.2017 – 23.10.2017	40k	v3.8.9
2017B	23.07.2017 – 18.09.2017	28k	v3.8.9
2017C	07.07.2017 – 22.07.2017	151k	v3.8.9
2017D	15.06.2017 – 03.07.2017	82k	v3.8.9

Table 4.1: Table Of Employed Experimental Data Sets.

Detailed information and characteristics of the employed data sets can be found on [17].

Decay Mode	**Branching Ratio**	**Sample Size**	
$K^+ \to \pi^+\pi^0(\gamma)\ ;\ \pi^0_{\gamma\gamma}$	$(20.43 \pm 0.08) \times 10^{-2}$	No Overlay: Overlay:	1.58×10^8 1.58×10^8
$K^+ \to \pi^+\pi^0(\gamma)\ ;\ \pi^0_D$	$(0.24 \pm 0.007) \times 10^{-2}$	No Overlay: Overlay:	7.40×10^7 7.39×10^7
$K^+ \to \pi^+\pi^0\pi^0$	$(1.760 \pm .023) \times 10^{-2}$	No Overlay: Overlay:	2.30×10^8 2.30×10^8
$K^+ \to \pi^+\gamma\gamma$	$(9.61 \pm 0.15 \pm 0.07) \times 10^{-7}$	No Overlay: Overlay:	8.12×10^6 8.11×10^6
$K^+ \to \pi^+\gamma\gamma\gamma$	$< 10^{-4}\ , CLs = 90\%$	No Overlay: Overlay:	8.48×10^5 8.47×10^5
$K^+ \to \pi^0 e^+ \nu_e(\gamma)$	$(5.07 \pm 0.04) \times 10^{-2}$	No Overlay: Overlay:	2.38×10^8 6.31×10^8

Table 4.2: Table Of Employed Monte Carlos Simulation Data Sets.
The sample size corresponds to the amount of K^+ decays whose decay position lies in the fiducial volume of the experiment $Z \in (110, 170)m$. The reconstruction version of the samples as well as the Monte Carlo production version can be found in [17].

4.2 Selection Criteria For The Decay Products

The corresponding analysis program was written with the framework version v3.11.4, and its code can be found in Appendix B. In the following sections the analysis algorithms for both normalization and searched decay channels are presented.

4.2.1 Identification Of The π^+ Track

With the help of the STRAWs, the tracks of charged particles crossing them can be obtained. In order to determine the momentum of the reconstructed track, it is needed that the charged particle crosses STRAW stations at different sides from the MNP33 magnet. Also, the deflection direction determines the sign of the electric charged of the crossing particle. The reconstruction algorithm, created by the collaboration also provides information about the reconstruction error of the track, denoted by χ^2 value; the amount of STRAW chambers employed for the reconstruction; the distance between the track and the nominal beam position, CDA; geometric and activity associations with other detector systems such as the LKr, MUV3, RICH, etc; reconstruction time value with respect to the trigger time.

The electric charge of the π^+ is $+e$, hence its track must have the same charge. Since many other charged particles share the same electric charge, the RICH is employed for the particle identification. For this reason, it is required that the tracks have a geometrical and activity associations with the RICH, and also that the produced ring most likely comes from a π^+. If this condition is satisfied, then the track time is defined with the RICH ring time. Association with the CHOD is needed since it is desired that the π^+ is the source of the trigger, while association with the LKr is important for the latter identification of the photons. Since in the decay there is no present neutrino, a geometrical association of the track with the MUV3 is needed, yet without associated activity in this detector. Since the π^+ is the only electrically charged particle of the decay products, it is expected to have only one track in the event. Hence a veto to events with tracks with time differences between the $\pm 21\ ns$ is applied, see figure 4.2. The numerical values of the corresponding selection criteria of the track can be found on table 4.3. All tracks satisfying these criteria are considered as π^+, its corresponding four-momenta is computed with equation 4.3 employing the π^+ mass and the measured momentum by the STRAWs as input.

Criteria	**Value**
Electric charge	$+e$
Number of STRAW chambers employed for reconstruction	4
CDA with respect to the nominal beam position	$< 25\ mm$
Tracks reconstruction χ^2	< 20
Time difference with other reconstructed tracks	$> 21\ ns$
Time difference with respect to the trigger time	$\in (-7, 3.5)\ ns$
Track momentum	$\in (1.5, 75)\ GeV/c^2$

Table 4.3: Criteria Values For Selected π^+ Tracks.

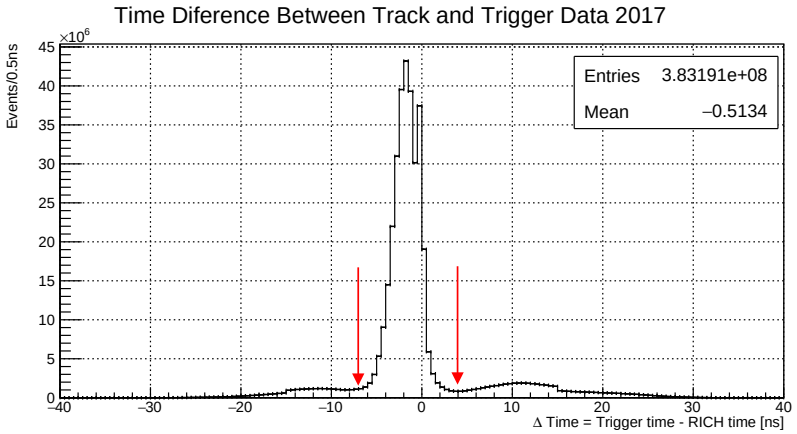

Figure 4.1: Time Difference Between Tracks and Trigger Times.

The observed distribution is asymmetric, since the CHOD has a 4 ns delay due to internal properties, and wide as result of the different time resolutions between the CHOD (source of the trigger) and the RICH. The employed time window is intended to select tracks within the central part of the distribution, shown with the red arrows.

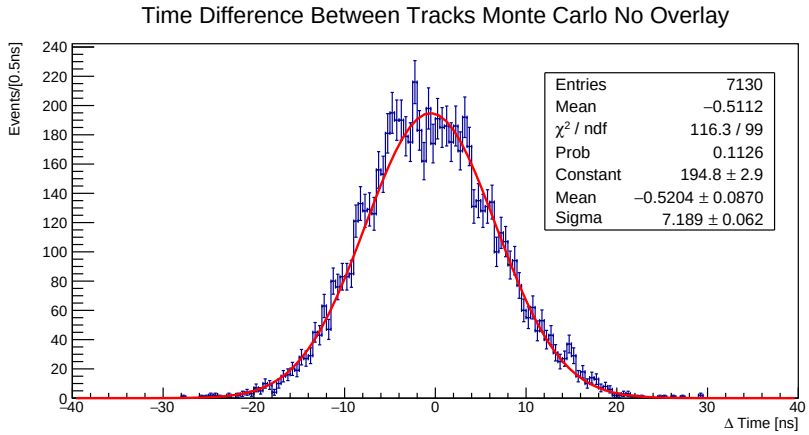

Figure 4.2: Time Difference Between Tracks For $K^+ \to \pi^+\gamma\gamma\gamma$.

The time difference comes from events with more than one reconstructed track. Since the No Overlay simulations consists of pure $K^+ \to \pi^+\gamma\gamma\gamma$ events, the presence of a second track indicate an unexpected process in the reconstruction. In red, a gaussian fit is displayed. The rejection time window for events with more than 1 track consist of a three sigma cut.

4.2.2 Identification Of The K^+ Track and π^+ Matching

Once identified the π^+ track, its turn for the K^+ one. For this, two detectors systems are employed, the KTAG and the GTK. First, the MatchingRG algorithm is employed to create a list of possible GTK tracks and STRAW tracks matches. This list is ordered from the most probable match (if it exists) to the least probable. The matches, are data structures which contain the formed vertex between GTK and STRAWs tracks, the GTK candidate time, the CDA between tracks, among other information, see [17] MatchingRG section for details. In this case, the most probable match is employed to reconstruct the K^+ decay vertex. This reconstructed vertex is needed to be positioned in the fiducial volume of the experiment, between 110 m and 170 m along the Z direction of the experiment, else they are discarded. Also, events without a match are discarded. Afterwards, the time of the GTK candidate is employed to match KTAGs candidates. The absolute time difference between them, should be below 0.4 ns, see figure 4.3. Also, the number of active sectors in the CEDAR for the time-matched candidates should be above 4 active sectors, see figure 4.4 for details. If at least on KTAG candidate satisfied both criteria, then the GTK track is considered to come from a K^+. The corresponding four momentum of the K^+ is obtained with equation 4.3 by assigning the GTK track the K^+ mass. Events without any matched K^+ are rejected.

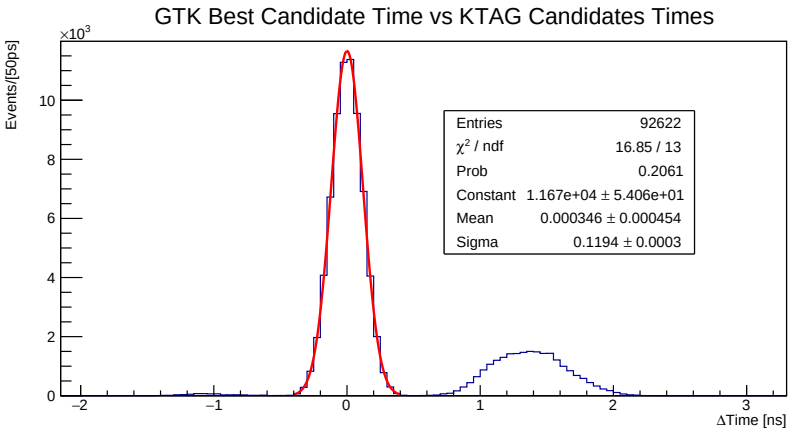

Figure 4.3: Time Difference of GTK Best Match And KTAG Candidates.

The above histogram was obtained from Monte Carlo simulations of $K^+ \to \pi^+\gamma\gamma\gamma$. To obtain a useful matching time window between CEDAR candidates and the GTK track, a gaussian function was fitted between -0.4 ns and 0.4 ns. The end window resulted in ≈ 4 sigmas of the gaussian.

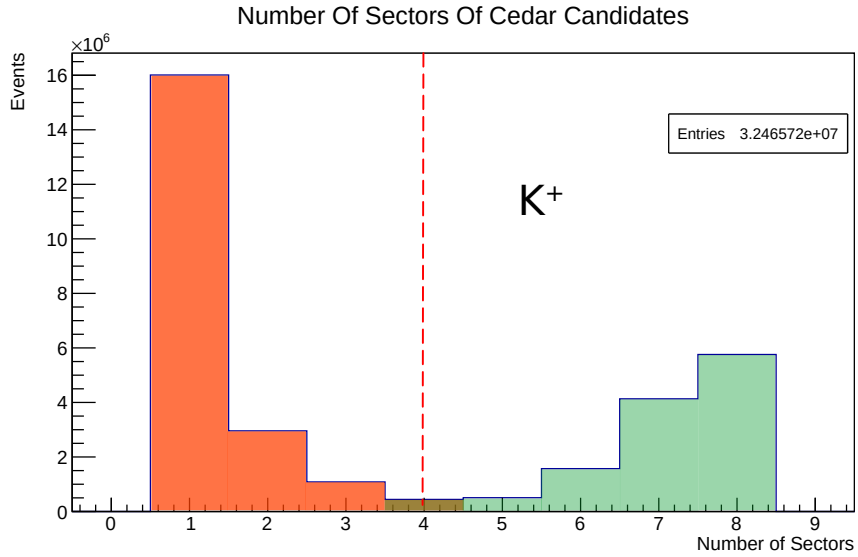

Figure 4.4: CEDAR Active Sectors.

The figure above was obtained from 2024 data. In it, two populations can be distinguished, one peaking at 8 active sectors arising from in-time Kaons, while the other-one is produced by the combination of late hits in the PMTs and the reconstruction software. The tail of the distribution of both populations intersects at 4 active sectors.

4.2.3 Identification And Selection Criteria Of γs

Since the proper identification of the photons as well as their four-momenta is of great relevance for calculating the reconstructed mass of the K^+, as well for properly identifying the normalization modes and the searched mode, the photon detection system to be used should be well suited for this purpose. The energy resolution of the LKr, places it as the photon detection system to be used. Hence, a rejection to events with activity in the LAV stations, the IRC and the SAC, since all produced photons are desired to cross the LKr.

Particles crossing the LKr will end up depositing their energy. During this process a shower of created particles will arise inside the detector. The original as well as the produced particles ionize the liquid krypton, hence producing currents in the cells of the detector. The current in each cell can be associated to a certain energy deposited in the cell. Many of this active cells can be associated from one-another by their respective relative distance, relative time, and energy. This collection of associated active cells, is called an energy cluster. On software the reconstruction of clusters is produced by the "EnergyClusterBuilder" pre-analizer provided by the collaboration. The coordinates of the cluster, it time with respect to the trigger, its energy, and association to the reconstructed tracks by the STRAWs, are provided. On table 4.4 the criteria for identifying clusters produced by photons, and associated to the tracks is presented. Since the normalization channel $K^+ \to \pi^+\gamma\gamma$ has at least two photons, and $K^+ \to \pi^+\pi^0\pi^0$ has four at most, hence events must have two to four neutral clusters, otherwise they are rejected.

In order to determine the four-momenta of the photons, the energy of the clus-

ters is employed. To determine the direction of the $\vec{p}$ component, the decay vertex of the K^+ determined with the MatchingRG algorithm as well as the position of the clusters is used.

Criterion	**Description**
Neutrality of cluster	No STRAW track spatially associated to the cluster.
Association with the π^+	Time difference between π^+ track in $(-3, 5)$ ns.
Energy interval of cluster	Energy of the clusters in $(2, 55)$ GeV.
Maximum size of cluster	Maximum RMS of the cluster in $(10, 20)$ mm.

Table 4.4: Criteria Employed For Photon Clusters.

Identifying the origin of the photons is key to properly determine whether the events correspond to the normalization channels or $K^+ \to \pi^+\gamma\gamma\gamma$. Specifically, the normalization channels have photons produced by the decay of the present π^0 while the search decay doesn't. To distinguish both types of photons, first the sum of pairs of photons four-momenta is computed, $P_{\gamma i,j} = P_{\gamma i} + P_{\gamma j}$. Then, with the help of equation 4.3 the mass $M_{\gamma i,j}$ is computed. Finally, a pair of photons is said to come from a π^0 if inequality 4.17 is satisfied. The employed upper limit to this mass difference was obtained from Monte Carlo simulations, see figure 4.5

$$|M_{\gamma i,j} - M_{\pi^0}| < \ 15 \ \frac{MeV}{c^2} \tag{4.17}$$

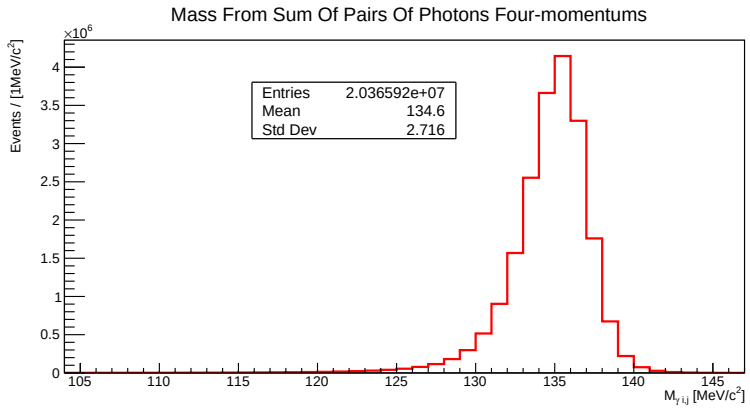

Figure 4.5: Computed Mass For $K^+ \to \pi^+\pi^0(\gamma)$, Monte Carlo No Overlay.

The mass distribution was obtained from a sample of $K^+ \to \pi^+\pi^0(\gamma)$, simulations without overlay. The reported mass of the π^0 in the PDG corresponds at the moment to 134.9768 ± 0.0005 MeV/c^2. Only events were two neutral clusters in the LKr were used. The asymmetry of the distribution is produced by events that also contain an Internal-Bremsstrahlung photon that doesn't cross the LKr.

4.3 Selection Criteria For Normalization Modes

At this point, the four momentum of the decay products as well as the K^+ passing through the GTK have been calculated. The difference in momentum, ΔP,between the reconstructed four-momenta of the K^+ and the one measured with the GTK can be calculated and used as a criterion for the quality of the reconstruction. Also, the magnitude of the transversal momentum of the reconstructed candidate with respect to the GTK, $P_\perp$, candidate can be also employed. Additional conditions to the employed neutral clusters were also utilized to improve the selection of both normalization channels. It was required a separation between the cluster produced by the track and the neutral clusters bigger than the RMS/size of the clusters. Also, the produced neutral clusters should be away from the LKr dead cells. The purpose of both conditions is having a better quality in the photons energy measurement, hence improving the final reconstruction of the four-momenta. Finally to separate events from $K^+ \to \pi^+\pi^0(\gamma)$ and $K^+ \to \pi^+\pi^0\pi^0$, the number of neutral clusters is employed, as well as the number of identified π^0. It should be mentioned that in the case of $K^+ \to \pi^+\pi^0(\gamma)$, only events without the internal-Bremsstrahlung photons were selected. The selection values for the quantities previously described can be found on table 4.5.

Criterion	$K^+ \to \pi^+\pi^0$	$K^+ \to \pi^+\pi^0\pi^0$
$\Delta P \ < 20\ GeV/c$	✓	✓
$P_\perp^2 < 400\ (GeV/c)^2$	✓	✓
Number of $\gamma\ = 2$ and Number of $\pi^0\ = 1$	✓	×
Number of $\gamma\ = 4$ and Number of $\pi^0\ = 2$	×	✓

Table 4.5: Criteria Employed For Selecting Normalization Channel Events.

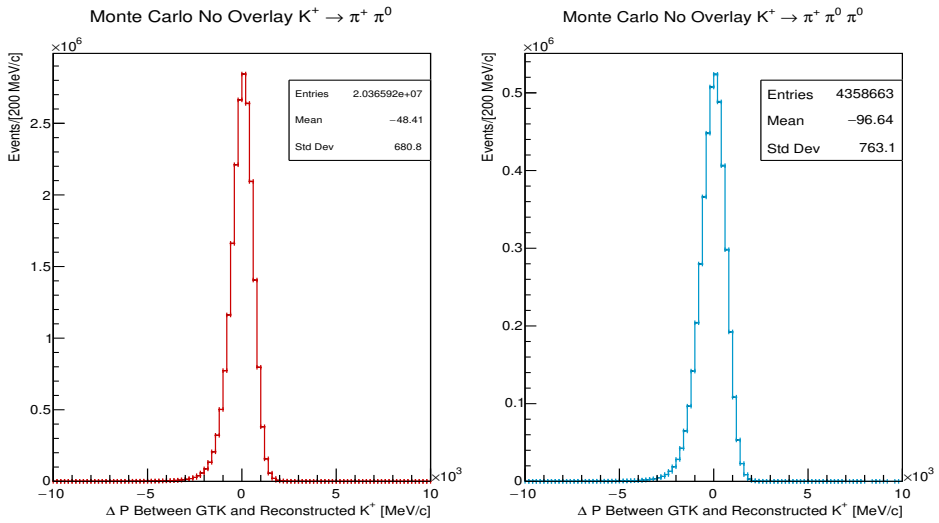

Figure 4.6: ΔP Between Reconstructed K^+ and GTK K^+, Monte Carlo No Overlay.

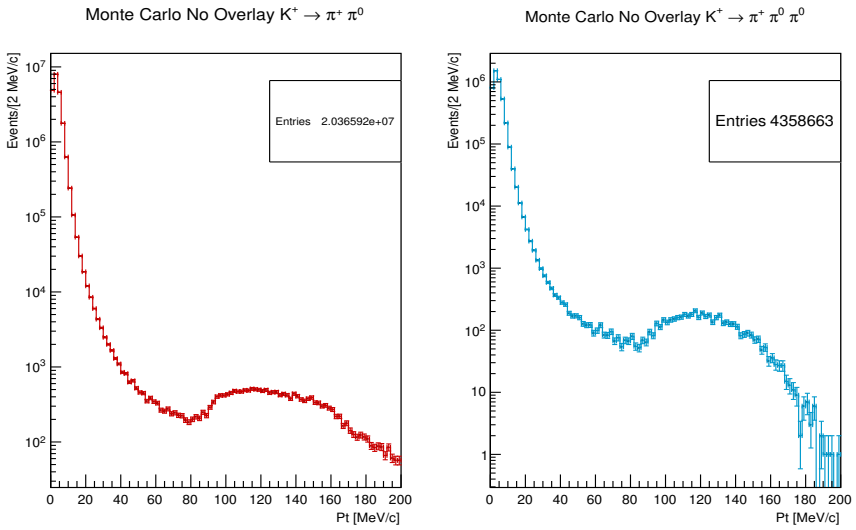

Figure 4.7: $P_\perp$ Of Reconstructed K^+ And GTK K^+, Monte Carlo No Overlay.

4.4 Selection Criteria for $K^+ \to \pi^+\gamma\gamma\gamma$

For the identification of $K^+ \to \pi^+\gamma\gamma\gamma$, a comparison between the reconstructed K^+ four-momenta wrt. the measured from the GTK is performed in the same way as with the normalization mode, this means computing ΔP and $P_\perp$ and applying the same selection criteria as with the normalization modes. Since among the decay products no pair of photons was produced from the decay of a π^0, the number of counted π^0 should be zero, while the number of neutral clusters in the LKr should be three. In order to avoid confirmation biases, a blinded mass region as well as two control regions were defined as follows:

- **Blinded mass region:** Invariant mass of reconstructed K^+ between $(476, 510) MeV/c^2$; accessible only for simulation data sets.
- **Control Region I and II:** Invariant mass of reconstructed K^+ below $476 MeV/c^2$ for I, and above $510 MeV/c^2$ for II; accessible for both simulation and experimental data sets for comparison purposes.

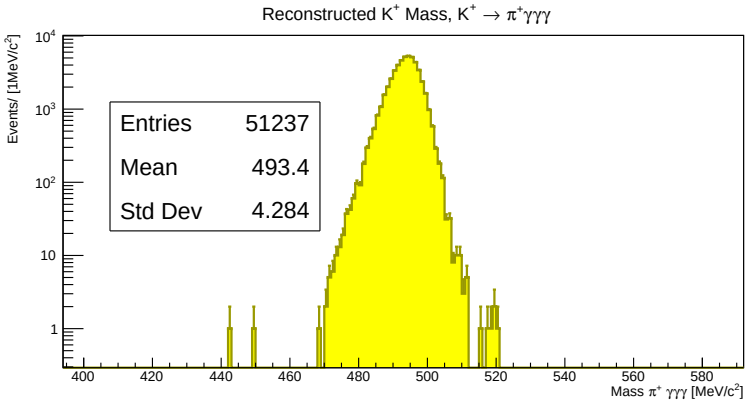

Figure 4.8: Mass Distribution Of The Reconstructed $K^+ \to \pi^+\gamma\gamma\gamma$.

This mass distribution was obtained from a sample of Monte Carlo No Overlay simulations of the decay process $K^+ \to \pi^+\gamma\gamma\gamma$. Conditions of ΔP and $P_\perp$, as well as for the number of neutral clusters (3) were applied in the selection process.

Even if the number of photons characterizes the decay mode $K^+ \to \pi^+\gamma\gamma\gamma$, other decay modes might produce background events even for the criteria selection previously stated. For example, $K^+ \to \pi^+\gamma\gamma$ has three photons in its final state, hence the reconstructed K^+ mass will actually be in the signal region. It could also happen that $K^+ \to \pi^+\pi^0\pi^0$, looses one of its photons which has so little four-momenta that the reconstructed mass also approaches the K^+ mass. Other effects such as photons landing close enough from one-another in the LKr to form just one neutral cluster, or pair produces e^+ e^-, creating non-track associated neutral clusters might also be the source for background events. In order to get rid of this undesired events, additional selection criteria were implemented in the selection of $K^+ \to \pi^+\gamma\gamma\gamma$ events, each of them will be discussed in the following subsections.

4.4.1 Missing Mass Cut

In order to address the background produced by $K^+ \to \pi^+\pi^0$, a way to identify the π^0 is needed. The requirement for three neutral clusters in the event implies problems in the counting of neutral clusters. It can be related to the absence of association of STRAW tracks with the present clusters in the LKr. An example of this can be the pair production of electron-positron by one of the π^0 photons in the third STRAW station, with a sufficient aperture angle, both the e^+ and e^- might produce two different clusters without any reconstructed track associated. Hence, for this situations, employing a method for identifying the π^0 without the LKr is preferable. Lets recall equation 4.1, by subtracting the four-momenta of the π^+ to the four-momenta of the K^+, only the sum of the four-momenta of the photons remain, see equation 4.18.

$$P_{K^+} - P_{\pi^+} = \sum_{i=1}^{3} P_{\gamma i} \tag{4.18}$$

Then by employing equation 4.3 with equation 4.18, a reconstructed mass can be computed with the GTK and the STRAWs. This is called the Missing Mass, M_{miss}, since it comes from the four-momenta of particles that didn't produce nor have associated a track, see equation 4.19. If the K^+ decays into two bodies, the computed missing mass will have a peak at the mass of the particle that wasn't employed in the subtraction, particularly for $K^+ \to \pi^+\pi^0$ around the π^0 mass. While for decays with more than two decay products, it shouldn't present a resonance behavior but a smooth distribution, see figure4.9.

$$(M_{miss}\ c)^2 = \ (P_{K^+} - P_{\pi^+})^2 = \begin{cases} (M_{\pi^0}\ c)^2 & , K^+ \to \pi^+\pi^0 \\ (M_{\gamma\gamma\gamma}\ c)^2 & , K^+ \to \pi^+\gamma\gamma\gamma \end{cases} \tag{4.19}$$

It should be noticed that missing mass of $K^+ \to \pi^+\gamma\gamma\gamma$ depends on the kinematics of the decay products, and in some cases the computation can give $M_{\gamma\gamma\gamma} = M_{\pi^0}$. This means that applying a veto to events whose missing mass lies close to M_{π^0} will also reduce the number of selected events of $K^+ \to \pi^+\gamma\gamma\gamma$, hence affecting its acceptance. Nevertheless, rejecting events whose squared missing mass lies on the interval $[12000, 24000](MeV/c^2)^2$, the squared mass of the π^0, has a higher impact reducing the number of $K^+ \to \pi^+\pi^0$ background events than reducing the number of $K^+ \to \pi^+\gamma\gamma\gamma$, as can be observed in figure 4.9.

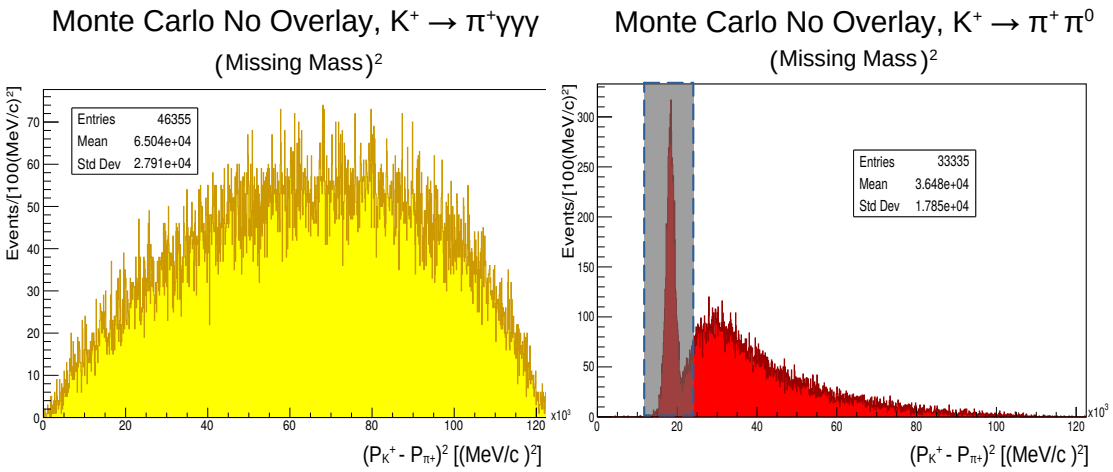

Figure 4.9: Missing Mass Of 3 Neutral Cluster Events.

In both images, the missing mass of events having three neutral clusters can be observed. At left, the missing mass obtained from a sample of $K^+ \to \pi^+\gamma\gamma\gamma$. It can be seen that some events are present in the rejecting region, but the do not constitute the majority of events. The picture on the right comes from a sample of $K^+ \to \pi^+\pi^0(\gamma)$, clearly a resonant behavior can be observed around 18225 $(MeV/c^2)^2$, the squared mass of the π^0 and rejection zone, signalized in gray. The events outside peak come from the decays with an Inner-Bremsstrahlung photon.

4.4.2 Radiative Photon Identification

In the previous subchapter, the background events from $K^+ \to \pi^+\pi^0$ were treated. As observed in figure 4.9, events with a radiative photon produced wont be affected by de rejection criteria established. In this cases, the final state consists of three photons and at least a pair of them should come from the decay of the π^0. Nonetheless, even with the counting of the π^0 with the inequality 4.17, some events might pass the rejection due to the imperfections of the measurement of the photons four-momenta. To overcome this situation, equation 4.20 can be employed. If is is assumed that the three photons actually come from $K^+ \to \pi^+\pi^0(\gamma)$, then the π^0 mass can be calculated by subtracting the radiative photon and the π^+ four-momenta from the one of the K^+.

$$(P_{K^+} - P_{\pi^+} - P_{\gamma i})^2 = \begin{cases} (P_{\pi^0})^2 = (M_{\pi^0}\ c)^2 & , K^+ \to \pi^+\pi^0(\gamma) \\ (P_{\gamma j} + P_{\gamma k})^2 = (M_{\gamma\gamma}\ c)^2 & , K^+ \to \pi^+\gamma\gamma\gamma \end{cases} \tag{4.20}$$

The problem is that, the identification of the radiative photon by it self can be complicated. Nevertheless, the calculation can be performed for each of the measured photon four-momenta. Ass shown in equation 4.20, if the event corresponds to $K^+ \to \pi^+\gamma\gamma\gamma$, the calculated mass will present a distribution without any peak, while for $K^+ \to \pi^+\gamma\gamma$, the distribution will have a peak around the π^0 mass as can be observed in figure 4.10. As a criteria, events with a $(M_{\gamma\gamma})^2$ inside the interval $[14000, 22000](MeV/c^2)^2$ are rejected.

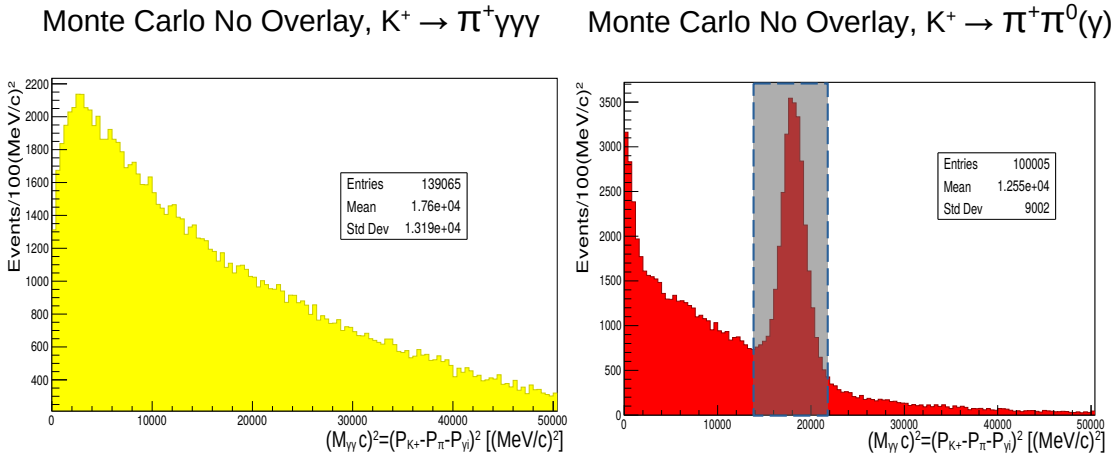

Figure 4.10: Distribution Of $(M_{\gamma\gamma}\ c)^2$ For Events With Three Neutral Clusters.

The image at the left side shows the distribution of masses obtained from a simulation sample of $K^+ \to \pi^+\gamma\gamma\gamma$. At the right side, the same distribution is shown for events of $K^+ \to \pi^+\pi^0(\gamma)$, the peak around $18225(MeV/c)^2$ correspond to events with a radiative photon.

4.4.3 Asymmetry of Neutral Clusters

Consider the decay mode $K^+ \to \pi^+\pi^0\pi^0$, in where both the π^0s decay in to two photons. In order for this process to survive the the selection criteria of three neutral clusters, two possible scenarios have to happen. The first of them is in which one of the four produced photons doesn't reach the LKr. But this events aren't likely to survive the selection for the following two reasons. First, at list a pair of photons from one of the π^0s while land on the LKr, and hence the reconstructed mass of the π^0 while be obtained, and hence the event can be vetoed. Second, even if the reconstructed mass of the π^0 doesn't lie on the counting region established in the inequality 4.17, the reconstructed K^+ mass is likely no to lie on the signal region. The second, and most dangerous situation arises not when one of the photons gets lost, but when the four of them reach the LKr, and yet produce just three neutral clusters. This is the case in which two photons land on the LKr close enough from one-another so the reconstruction algorithm for the clusters outputs just one cluster candidate. The result of this situation is that the reconstructed mass of the K^+ will lie in the signal region since no energy is lost, and the measured four-momenta of this cluster emulates the four-momenta of the photons of $K^+ \to \pi^+\gamma\gamma\gamma$. Also, inequality 4.17 can't be employed for counting π^0s properly, since the four-momenta of the individual photons can't be recovered from the produced LKr directly, hence making impossible to reconstruct the π^0 mass from them.

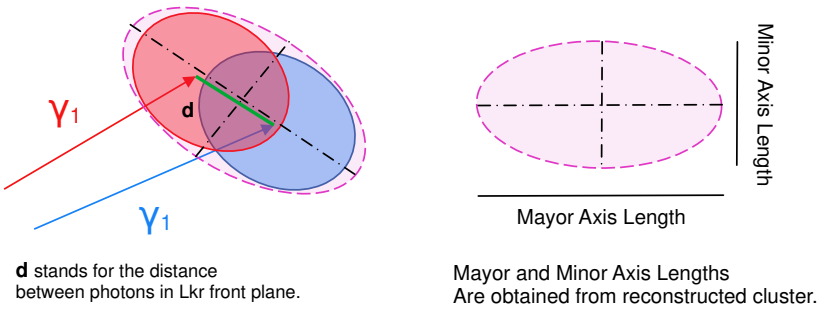

Figure 4.11: Asymmetry Of LKr Clusters.

The image presents the conceptual idea behind the asymmetry of clusters, produced by merged photons. The axis of the ellipses are obtained with the RMS of the cluster, which is calculated using the energy and central position of the LKr hits. The major and minor axis don't necessarily lie along the XY axis, but the framework of the experiment offers routines that identify them.

To address this last situation, a geometrical approach can be used. Clusters have a ellipse like shape projected on the front face of the LKr. For clusters produced by single photons, the mayor and minor axis are mostly the same size, so the final shape can be thought to approximate to a circle. In cases in which both photons land sufficiently away from one another, the shape of the produced cluster will present a greater asymmetry between both axis, as is pictured in figure 4.11. As a definition, the ratio between the minor axis (computed with the minimum RMS distance to the center), and the mayor axis (computed with the maximum RMS distance to the center) will be denoted as the cluster asymmetry.

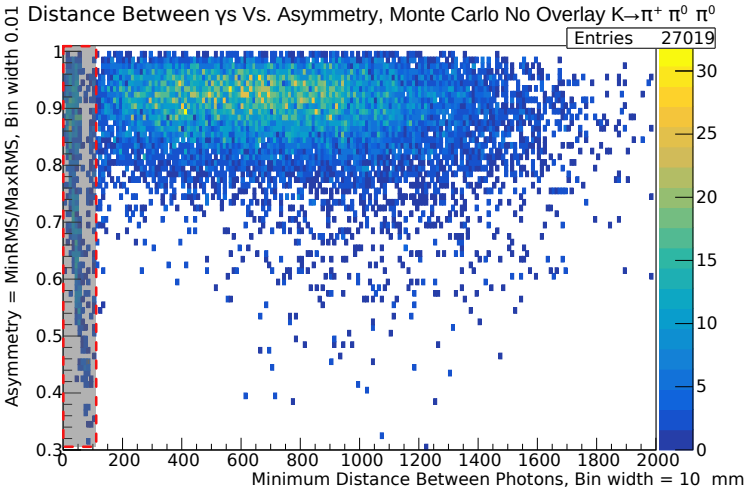

Figure 4.12: Asymmetry Of Neutral Clusters With Respect Of The Distance Between Photons.

The picture shows how the asymmetry of the clusters changes wrt. the distance between photons in the front plane of the LKr. The picture was obtained from a simulation sample of $K^+ \to \pi^+\pi^0\pi^0$. The distance was obtained by getting the minimum distance between the simulated photons. Two populations can be distinguished, the first one, signalized by a gray box corresponds to clusters produced by merged photons whose distance from one-another is below the 100 mm. The other population correspond to clusters produced by single photons.

As can be seen in figure 4.12, events which have photons that land on the LKr at distances below the 100 mm, have a remarkable asymmetry from 0.3 up to 1.. As a rejection criteria for the events, if at least one cluster has an asymmetry below 0.85 the event is rejected. Yet clusters composed by single photons might have asymmetries similar to those who were produced by two photons, this events are treated in the following subchapter.

4.4.4 Merged Photons Energy

For those cases in which the cluster asymmetry is similar to the single photon clusters, an approach arising from the kinematics of the decay can be employed. Imagine the case in which two photons, each produced by the decay of different π^0s cross the LKr sufficiently close to produce just one cluster, while the other two photons produce separated clusters, see figure 4.13. The energy of the cluster formed by the two photons, E_{c3}, is equal to the sum of the individual energies of the two photons, $E_{\gamma 1,2} + E_{\gamma 2,2}$. The energy of the other clusters E_{c1} and E_{c2}, correspond to the energy of the photons, $E_\gamma 1, 1$ and $E_{\gamma 2,1}$. The energy of the photons $E_{\gamma 1,2}$ and $E_{\gamma 1,1}$, are correlated since both come from the same π^0, and the same happens for the remaining pair. This indicates that by knowing the four-momenta of the isolated photon, gives back the four-momenta of one of the merged photons.

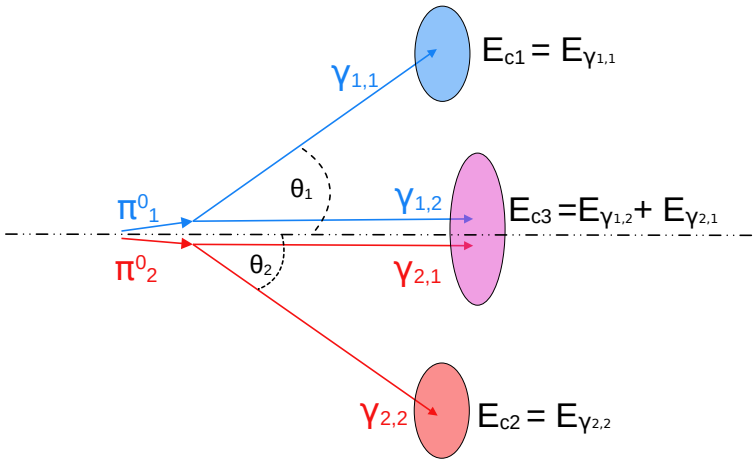

Figure 4.13: Merged Photons Energy Sketch.

Let the subindex i denote if the photons come from the π_1^0 or the π_2^0. Then with the help of equation 4.1 one can get the following relations:

$$\begin{aligned}
(M_{\pi^0}c)^2 &= (P_{\gamma i,1} + P_{\gamma i,2})^2 \\
&= (P_{\gamma i,1})^2 + (P_{\gamma i,1})^2 + 2P^{\mu}_{\gamma i,1} P_{\gamma i,2\ \mu} \\
&= 2P^{(\mu)}_{\gamma i,1} P_{\gamma i,2\ (\mu)} \\
&= 2[\frac{E_{\gamma i,1}E_{\gamma i,2}}{c^2} - \vec{p}_{\gamma i,1} \cdot \vec{p}_{\gamma i,2}] \\
&= 2[\frac{E_{\gamma i,1}E_{\gamma i,2}}{c^2} - \frac{E_{\gamma i,1}E_{\gamma i,2}}{c^2}\cos(\theta_i)] \\
&= 2[\frac{E_{\gamma i,1}E_{\gamma i,2}}{c^2}][1 - \cos(\theta_i)]
\end{aligned} \tag{4.21}$$

$$\Longrightarrow E_{\gamma i,2} = \frac{(M_{\pi^0}c^2)^2}{2E_{i,1}[1 - \cos(\theta_i)]} \tag{4.22}$$

From the previous equation $E_{\gamma 1,2} + E_{\gamma 2,2}$ can be computed. Also, by considering θ_i close enough to the angle formed from the K^+ decay vertex and the center of the isolated clusters and the merged photon clusters, and by substituting the energy of the isolated photons with the energy of their respective clusters, equation 4.23 is obtained.

$$E_{c3} = (\frac{(M_{\pi^0}c^2)^2}{2})[\frac{1}{E_{c2}\cos(\theta_2)} + \frac{1}{E_{c1}\cos(\theta_1)}] \tag{4.23}$$

For instance, if any event come from a $K^+ \rightarrow \pi^+\pi^0\pi^0$, or as far as both π^0s decayed close to one-another, equation 4.23 will be satisfied. Hence, by testing if the clusters in the event satisfy it, can be employed as a rejection criterion. From figure 4.14, it can be appreciated that the energy difference between the clusters energy and the expected energy calculated from equation 4.23 is almost centered around $0MeV$, as a rejection criterion, a value of 1.5 GeV for the absolute difference between expected energy and the cluster energy is employed.

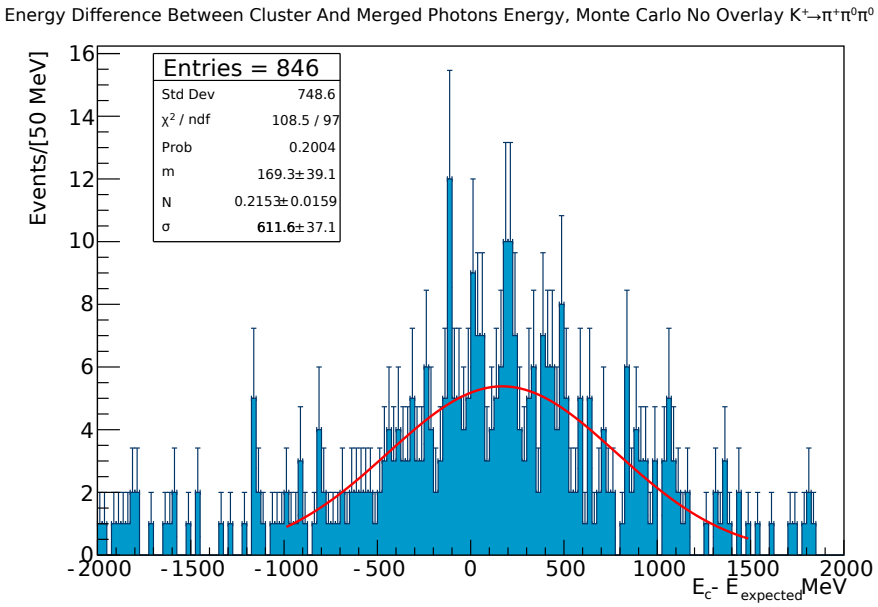

Figure 4.14: Distribution Of Energy Difference Between Cluster Energy And Expected Energy For Merged Photons.

The distribution was obtained from a sample of $K^+ \rightarrow \pi^+\pi^0\pi^0$ of a no overlay Monte Carlo simulation with three neutral clusters in the event. In red, a gaussian distribution was fitted, showing a mean value of $168MeV$ and a standard deviation of $611MeV$.

4.4.5 Cut In Missing Mass And $M_{\gamma\gamma}$

An other way to eliminate the remaining $K^+ \to \pi^+\pi^0\pi^0$ background is by employing a rejection criterion which utilize a mix of different kinematic variables such as the Missing Mass or the mass from the four-momenta of two photons. In case that the events of $K^+ \to \pi^+\pi^0\pi^0$ mostly occupy a region apart from the events of $K^+ \to \pi^+\gamma\gamma\gamma$, then a rejection space can be defined using both variables. From figure 4.15 it can be appreciated that the majority of events of $K^+ \to \pi^+\pi^0\pi^0$ are away from the events of $K^+ \to \pi^+\gamma\gamma\gamma$. The first one mostly contained in the region $(Missing\ Mass)^2 > 72000(MeV/c^2)^2$ and $M_{\gamma\gamma} > 20000(MeV/c^2)^2$, hence events in it are rejected.

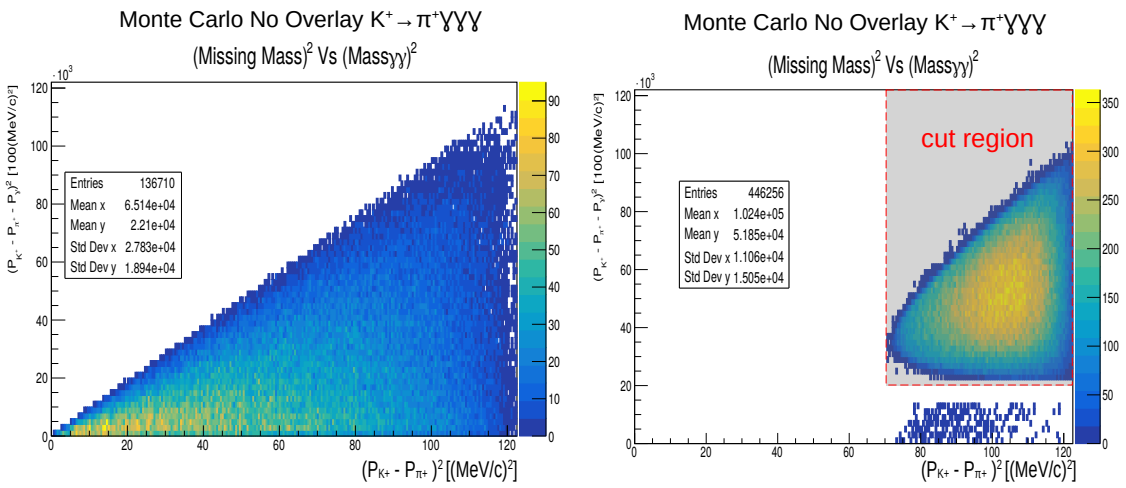

Figure 4.15: Distribution Of (Missing Mass)^2 And Mass$_{\gamma\gamma}$.

The current pair of distributions were obtained from Monte Carlo no overlay simulations of $K^+ \to \pi^+\pi^0\pi^0$ and $K^+ \to \pi^+\gamma\gamma\gamma$ with only three neutral clusters in the event.

4.4.6 Summary Of Selection Criteria

Criterion	**Description**
Number of γ $= 3$ and number of π^0 $= 0$	The photons in the final state shouldn't come from a π^0.
$\Delta P < 20\ GeV/c$	Difference in momentum between reconstructed K^+ and GTk candidate.
$P_\perp^2 < 400\ (GeV/c)^2$	Transversal momentum between reconstructed K^+ and GTk candidate.
$M_{miss}^2 \notin\ [12, 24] \times 10^3\ (\frac{MeV}{c^2})^2$	Missing mass squared away from π^0 mass squared.
$M_{\gamma\gamma}^2\ \notin [14, 22] \times 10^3\ (\frac{MeV}{c^2})^2$	Reconstructed squared mass of photon pairs away from π^0 mass squared.
Cluster asymmetry > 0.85	Ratio of Minimum RMS over Maximum RMS of a cluster is defined as its asymmetry.
$\lvert Ec - E_{expected}\rvert > 1.5\ GeV$	Absolute energy difference between clusters and $K^+ \to \pi^+\pi^0\pi^0$ merged photons cluster expected value, see equation 4.23 for definition.
$(\frac{M_{miss}^2}{10^3}, \frac{M_{\gamma\gamma}^2}{10^3}) \notin [72, \infty) \times [20, \infty)$	The event mustn't lie in the same squared missing mass×squared mass of photon pairs of $K^+ \to \pi^+\pi^0\pi^0$.

Table 4.6: Summary Of Selection Criteria Of $K^+ \to \pi^+\gamma\gamma\gamma$.

Chapter 5

Results

5.1 Normalization Modes

For the normalization mode, the most important quantities are the acceptance of $K^+ \to \pi^+\pi^0$ and $K^+ \to \pi^+\pi^0\pi^0$, as well as the number of events for each normalization modes, since they are employed for normalizing the estimated background in the signal region (with the help of equation 4.15), as well as for calculating an upper limit for the branching ratio of $K^+ \to \pi^+\gamma\gamma\gamma$, with equation 4.16. The total number of K^+ decays calculated with equation 4.6 for both normalization modes can be employed as a crosscheck for the normalization, since both calculated quantities should be the same or at least of the same order of magnitude. The previously enunciated quantities for both Monte Carlo Overlay and No Overlay simulations can be found in table 5.1.

The results of table 5.1 are heavily dependent on how well the simulation actually describes the behavior of particles as they cross the experiment. For the No Overlay samples of the simulation, we can see the comparison with experimental data for both $K^+ \to \pi^+\pi^0\pi^0$ and $K^+ \to \pi^+\pi^0(\gamma)$ in the figures 5.1 and 5.3. For both results, it is appreciable that the majority of events are close to the mass region of the K^+. For $K^+ \to \pi^+\pi^0$, the similarity between simulation and experiment deviates less than a 10% in the region of 480 MeV/c^2 to 500 MeV/c^2, and then drops drastically showing unfilled areas. This unfilled areas are explained with the help of the Overlay simulation, figure 5.2 doesn't show the unfilled areas, indicating that the missing events for the No Overlay simulation might be explained by the lack of overlap between decays, affecting the selection of the mode. For $K^+ \to \pi^+\pi^0\pi^0$, a similar behavior is observed, the mass region from 482 MeV/c^2 to 500 MeV/c^2 presents a deviation of the simulation with respect to the experiment of less than 10%, and then drops drastically. In the Overlay simulation scenario, shown in figures 5.2 and 5.4, it can be appreciated a better correspondence between simulation and data. For both $K^+ \to \pi^+\pi^0\pi^0$ and $K^+ \to \pi^+\pi^0$ the similarity close to the K^+ mass deviates below a 20%, which might give the impression that is worse than the No Overlay result. Yet, the ratio behaves different, showing more simulation events to the left of the K^+ mass and more to the right, instead of just dropping down as in the case of No Overlay.

In the table 5.1, a discrepancy of the number of K^+ decays between normalization modes for the Overlay and No Overlay results can also be appreciated. For the No Overlay case the ratio for the number of decays is 1.20 ± 0.02, while for the Overlay case is 1.12 ± 0.01. Even if this indicates a possible systematic effect present in the selection of a normalization channel, in principle this doesn't affects the upper limit for the branching ratio, since what should be taken into account is indeed the relative acceptances between normalization mode and $K^+ \to \pi^+\gamma\gamma\gamma$. Another noticeable difference between No Overlay results and Overlay is a factor 2 present in the quotient of acceptances. This indicates the reduction of selection in events probably produced by the overlap of different decay modes incrementing the number of clusters or satisfying the veto conditions. Nonetheless, this factor 2 is present for each of the decay modes, and effects all acceptances equally, so the relative acceptance in equation 4.16 isn't affected.

Mode	**Simulation**	**Acceptance[%]**	**N. Events[10^5]**	**N. K^+ Decays[10^9]**
$K^+ \to \pi^+\pi^0$	No Overlay	$12.713 \pm .003$	1192.8 ± 0.4	$4.59 \pm .02$
	Overlay	$7.305 \pm .002$		$7.99 \pm .03$
$K^+ \to \pi^+\pi^0\pi^0$	No Overlay	$0.9350 \pm .0006$	6.14 ± 0.01	$3.82 \pm .05$
	Overlay	$0.5004 \pm .0005$		$7.13 \pm .09$

Table 5.1: Results For Normalization Modes.

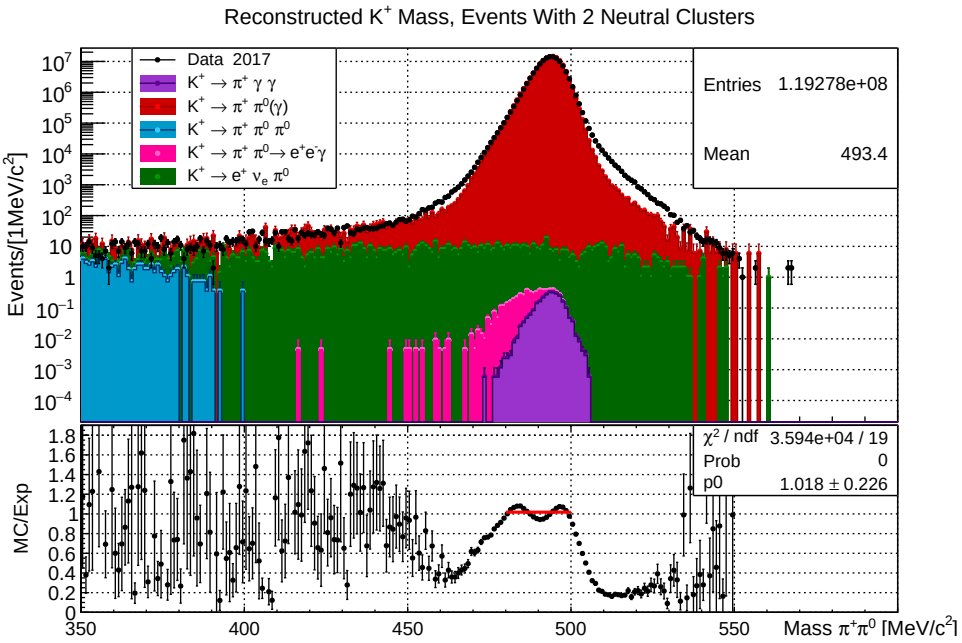

Figure 5.1: Reconstructed Mass Of π^+ π^0 For Different Monte Carlo No Overlay Decay Modes.

As observed, a peak corresponding to $K^+ \to \pi^+\pi^0$ is present around the K^+ mass. The decay $K^+ \to \pi^+\gamma\gamma$ produces also a peak several orders of magnitude smaller than $K^+ \to \pi^+\pi^0$.

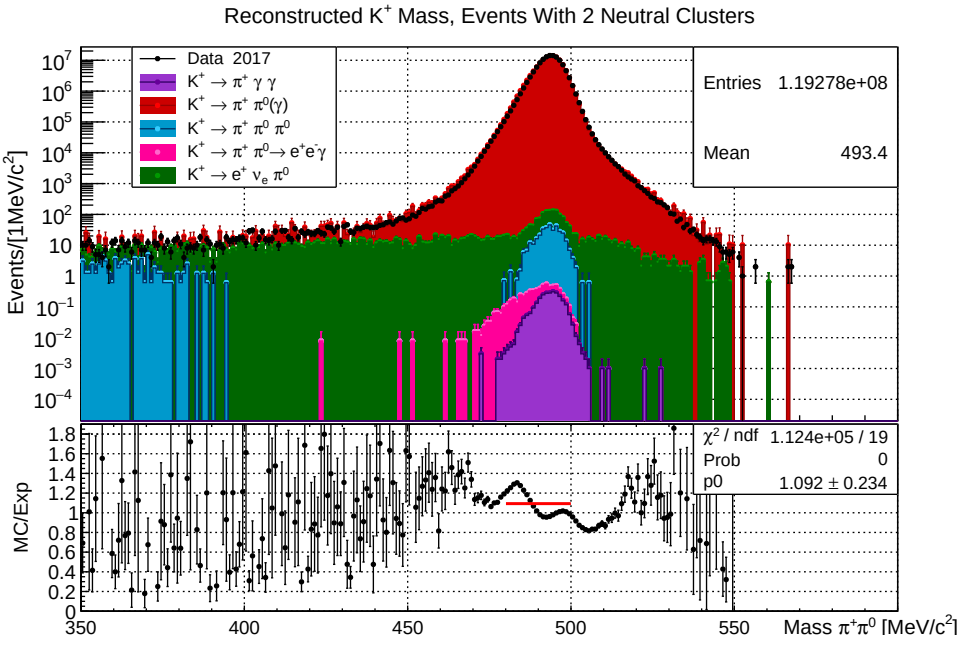

Figure 5.2: Reconstructed Mass Of π^+ π^0 For Different Monte Carlo Overlay Decay Modes.

As displayed, the use of Overlay sample also reflects the same peaks as the No Overlay scenario, yet a small sample of $K^+ \to \pi^+\pi^0\pi^0$ also forms a peak around the K^+ mass.

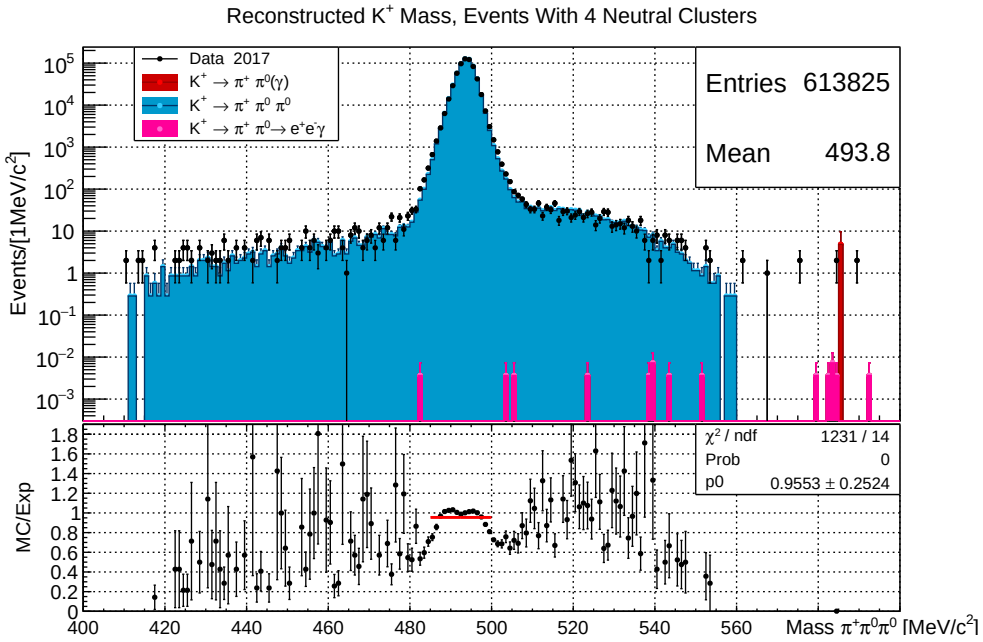

Figure 5.3: Reconstructed Mass Of π^+ $\pi^0\pi^0$ For Different Monte Carlo No Overlay Decay Modes.

In contrast with $K^+ \to \pi^+\pi^0$, the distribution here is almost uniquely produced by the decay $K^+ \to \pi^+\pi^0\pi^0$, with no other decay mode peaking around the K^+ mass.

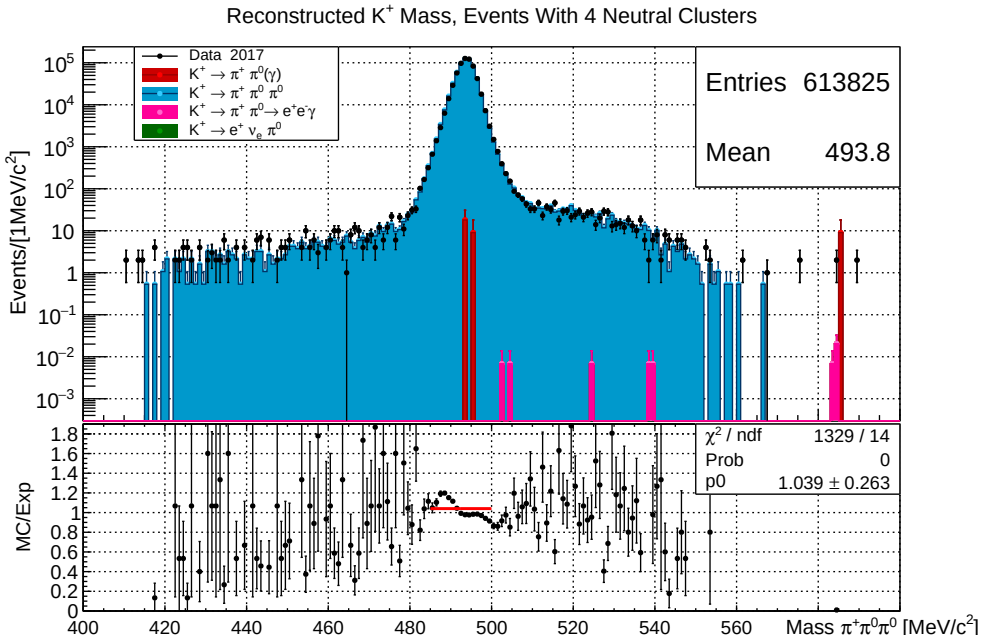

Figure 5.4: Reconstructed Mass Of π^+ $\pi^0\pi^0$ For Different Monte Carlo Overlay Decay Modes.

5.2 Mode $K^+ \to \pi^+\gamma\gamma\gamma$

It can be found in this section the histograms for the reconstructed K^+ mass for events with three neutral clusters, for the cases in which only the ΔP and $P_\perp$ selection criteria were applied, figures 5.6 and 5.5. Also, the histograms for the K^+ mass in which all the selection criteria, listed in table 4.6, are displayed in figures 5.7 and 5.8. In the table 5.2, the estimated events with the Monte Carlo Simulations in both control regions and the encountered events in data is shown, as well as, estimated background in the K^+ mass region. The estimations of the new upper limit for the branching ratio of $K^+ \to \pi^+\gamma\gamma\gamma$ can be found in table 5.3.

The figures 5.5 and 5.6 are helpful to show the similarity between simulation and experimental data for the reconstructed K^+ mass. For both Overlay and no Overlay simulations, it can be observed that the similarity of simulation and experiment is withing 10%. The Control Region I, is dominated by $K^+ \to \pi^+\pi^0\pi^0$ background, and this is to be expected, since most of this events correspond to the case in which one of the four produces photons doesn't reach the LKr, hence loosing a part of the total four-momentum of the reconstructed K^+. For the Control Region II, shows a similarity close to the unity for the mass region of 510 MeV/c^2 to 600 MeV/c^2, but presents high statistical fluctuations as can be noticed in the lower part of both figures. This region is dominated by $K^+ \to \pi^+\pi^0(\gamma)$ background, for both decay modes of the π^0. The reason for the existence of events which apparently breaks the four-momentum conservation giving higher mass is that all the events in this region are nothing else but accidents. To clarify what an accident means in this context, consider the production of a e^+, e^- pair as a photons passes through some material, as could be the STRAWS, the RICH mirror, etc. If the pair doesn't produce a track in the STRAWS, and manages to reach the LKr, the end result is one or two energy clusters that are neutral by the given definition in section 4. This clusters not only aren't produced by photons, but also the four-momentum associated to this clusters doesn't correspond to their production particles, since the direction and magnitude of the $\vec{P}$ component is calculated assuming cero invariant mass, and a production vertex corresponding to the K^+ decay vertex and not the pair production vertex.

It is also worth noticing that comparing the reconstructed four-momentum of the K^+ with the one measured with the GTk, is not sufficient to distinguish $K^+ \to \pi^+\gamma\gamma\gamma$ form the background, since the estimated events for both $K^+ \to \pi^+\pi^0(\gamma)$ and $K^+ \to \pi^+\pi^0\pi^0$ in the masked region reach the order of 10^4 events. In contrast, once applied all the selection criteria, it can be observed from figures 5.7 and 5.8 that the end result is an almost background free region. In this case, the background new remanent background becomes $K^+ \to \pi^+\gamma\gamma$, and $K^+ \to \pi^+\pi^0(\gamma)$, this last one being diminished many orders of magnitude.

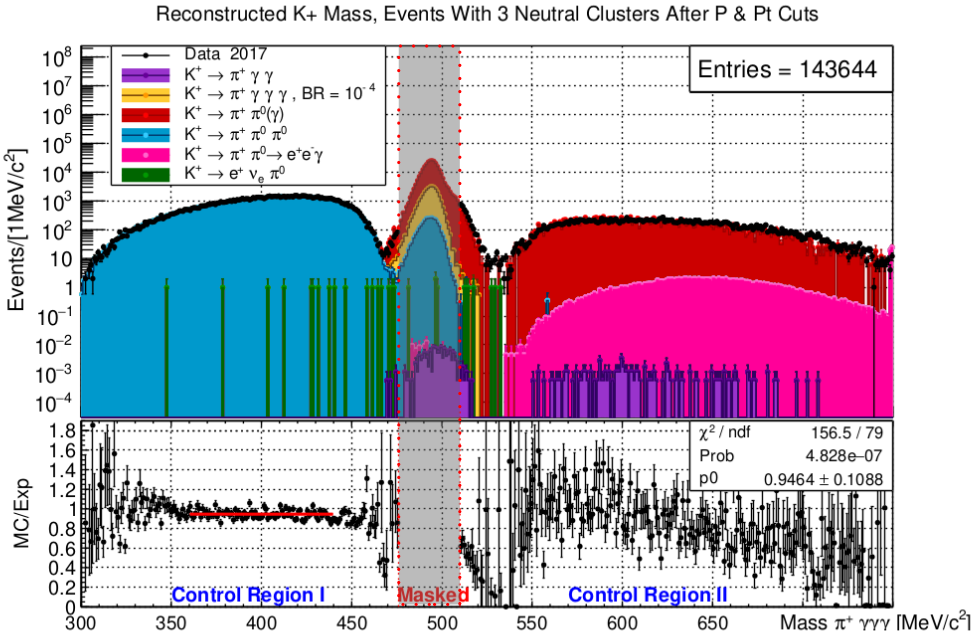

Figure 5.5: Reconstructed Mass Of π^+ $\gamma\gamma\gamma$ For Different Monte Carlo No Overlay Decay Modes, After ΔP And $P_\perp$ Cuts But Previous To π^0 Cut.

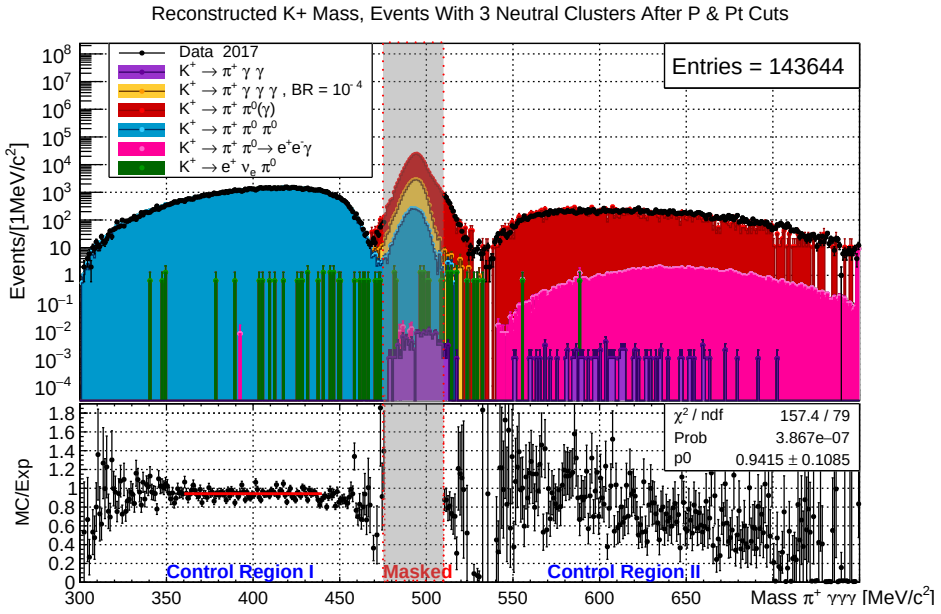

Figure 5.6: Reconstructed Mass Of π^+ $\gamma\gamma\gamma$ For Different Monte Carlo Overlay Decay Modes, After ΔP And $P_\perp$ Cuts But Previous To π^0 Cut.

For both Overlay and No Overlay figures, the shown distribution for $K^+ \to \pi^+\gamma\gamma\gamma$ was calculated assuming a branching ratio of 10^{-4}, as reported by [2].

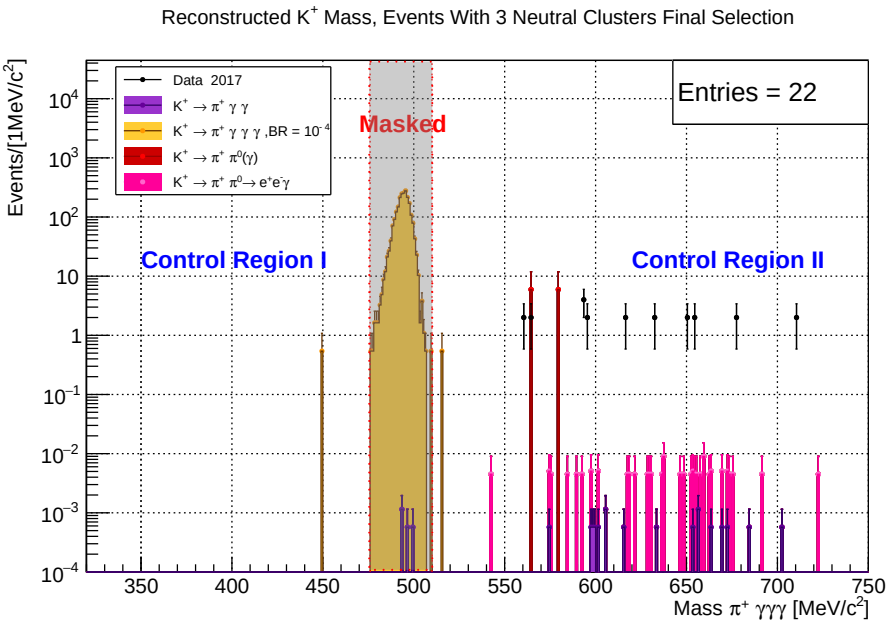

Figure 5.7: Reconstructed Mass Of π^+ $\gamma\gamma\gamma$ For Different Monte Carlo No Overlay Decay Modes After All Cuts.

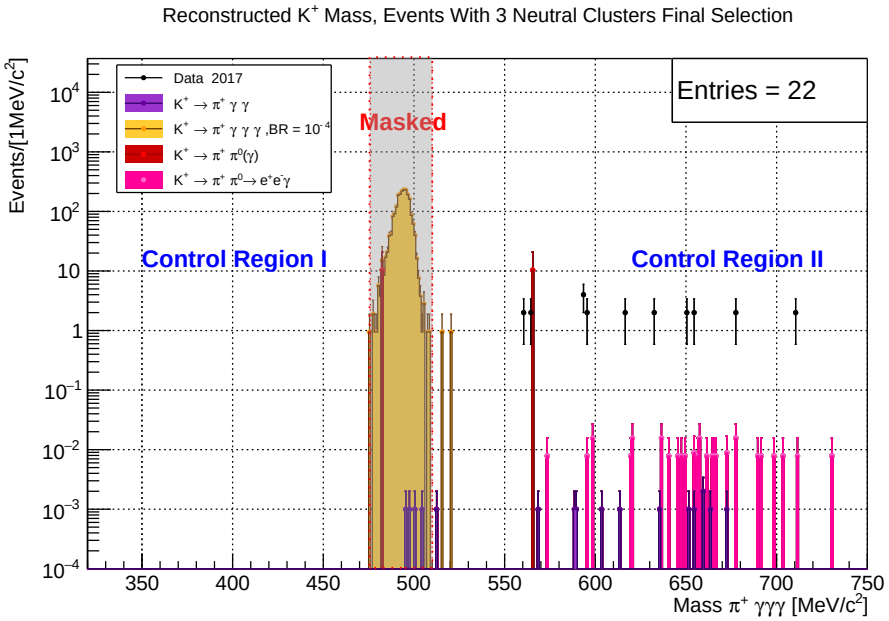

Figure 5.8: Reconstructed Mass Of π^+ $\gamma\gamma\gamma$ For Different Monte Carlo Overlay Decay Modes After All Cuts.

The displayed $K^+ \to \pi^+\gamma\gamma\gamma$ distribution was calculated, for both Overlay and No Overlay simulations, assuming a branching ratio of 10^{-4} as reported by [2]

From table 5.2, it can be observed that for both Control Regions, the observed events and the expected events from the simulations are statistically compatible. The pull for Control Region II is 1.25 for No Overlay simulations and 1.2 for Overlay simulations. In control region I, it can be said with a CLs of 90% that 2.3 events were expected at most, and this is consistent with the 0 observed events. In the masked region a discrepancy between Overlay and No Overlay expected values is observed, but it should be taken into account that the 10 events in the Overlay simulation might be actually smaller. Previous to the normalization, this used to be a single event, which might have been produced as a statistical fluctuation. To gain information of the distribution of this background, more simulation events are needed, or a biased simulation with the same characteristics of this events must be produced and properly weighted. Yet, this surpasses the objective and time for this thesis.

Mode	**Control Region I**		**Control Region II**	
	No Overlay	Overlay	No Overlay	Overlay
$K^+ \to \pi^+\pi^0(\gamma)$	0	0	11.9 ± 8.4	10 ± 10
$K^+ \to \pi^+\pi^0_D$	0	0	0.15 ± 0.03	0.23 ± 0.04
$K^+ \to \pi^+\gamma\gamma$	0	0	0.01 ± 0.002	0.013 ± 0.003
Total	0	0	12 ± 8	10 ± 10
Datos Exp.	0		22 ± 4	

Mode	**Masked Region**	
	No Overlay	Overlay
$K^+ \to \pi^+\pi^0(\gamma)$	0	10.36 ± 10.36
$K^+ \to \pi^+\pi^0_D$	0	0
$K^+ \to \pi^+\gamma\gamma$	$(0.2 \pm 0.1) \times 10^{-2}$	0.013 ± 0.003
Total	$(0.2 \pm 0.1) \times 10^{-2}$	10 ± 10
Datos Exp.	To be revealed	

Table 5.2: Expected And Observed Results For The Control And Masked Regions For $K^+ \to \pi^+\gamma\gamma\gamma$.

Tipo de Simulation	**Acceptance $K^+ \to \pi^+\gamma\gamma\gamma$**	**Normalization**	**N. Obs**	**CLs**	**Br Upper Limit [10^{-7}]**
No Overlay	$(0.479 \pm .007)\%$	$K^+ \to \pi^+\pi^0$	0	2.3	1.04
			1	3.9	1.77
			2	5.3	9.76
		$K^+ \to \pi^+\pi^0\pi^0$	0	2.3	1.26
			1	3.9	5.36
			2	5.3	9.74
Overlay	$(0.259 \pm .005)\%$	$K^+ \to \pi^+\pi^0$	0	2.3	1.11
			10	11.1	5.36
			20	20.2	9.76
		$K^+ \to \pi^+\pi^0\pi^0$	0	2.3	1.24
			10	11.1	6.00
			20	20.2	10.9

Table 5.3: Estimated Values For The Branching Ratio Upper Limit.

Chapter 6

Conclusions And Future Prospects

As stated in the introduction, the main objective of this thesis was to determine whether it is feasible or not to detect the decay mode $K^+ \to \pi^+\gamma\gamma\gamma$ with data from NA62. The result from figures 5.5 and 5.6 show that by employing the current limit for the branching ratio of $K^+ \to \pi^+\gamma\gamma\gamma$, the number of expected events is a couple or more orders of magnitude bigger than the expected background events in the masked region, making it easily recognizable. Yet, in the Standard Model scenario or in one in which this final state is sufficiently rare, expecting a number of events produced only by background sources, with orders of magnitude shown by table 5.2 would be more conservative. In this last scenario, by taking the results from the No Overlay simulation, observing a signal in the order of tenths would be explained by the background sources employed for this study. If the Overlay results are employed, then it would be sufficient to have more than 60 events to state that the signal can't be fully explained by the studied background sources. Nevertheless, a better understanding of the background distribution of the $K^+ \to \pi^+\pi^0(\gamma)$ background in the overlay case is needed to have a better estimate the impact of this background, since the reported result can be caused by statistical fluctuations and might not be representative of real expected value. From table 5.3, it can be observed that in the most favorable scenario, the expected improvement to the Branching Ratio Upper Limit, is of three orders of magnitude, while for the less favorable result, the improvement is only of two orders. Hence, results show that the objectives of this thesis were achieved.

In the context of measuring $\mathbb{C}$ prohibited processes of the electromagnetic interaction, for the lepton sector, [19] has obtained limits for prohibited branching ratios of ortho and para-positronium of 3.7×10^{-6} and 2.7×10^{-7} with a CL_s of 90%. The J-PET collaboration aims for improvements of $\mathbb{C}$ violation measurements among others also employing positronium as source[20].

In the matter of the future prospects of this analysis, it should be considered, that up to now, 2017 data group is close to be a tenth of the total data. The inclusion of other data groups is, by the time of publication of this work, still under process but an even greater improvement of the upper limit seems to be achievable, and with it the publication of the result. The key points to further investigate are, the distribution of $K^+ \to \pi^+\pi^0(\gamma)$ background for Overlay samples, the proper selection of triggers and change of relative acceptances for other data groups.

Appendix A

CLs Method

The CL_s method is a conservative approach for computing a limit for the confidence on the signal-only hypothesis. It is by itself not a confidence in the usual frequentist approach, but instead the ratio of to confidences. One for the background-only hypothesis, CL_b, and the other for the signal+background hypothesis, CL_{s+b}. Then a confidence α is related to CL_s by the following equation:

$$1 - \alpha = CL_s = \frac{CL_{s+b}}{CL_b} \tag{A.1}$$

Consider the single counting experiment, were N_{obs} events are observed. In such a case, we would like to contrast the hypothesis of observing signal + background events, $s+b$, vs the hypothesis of seeing only background, b. Both cases will follow a Poisson distribution, the first one with expected value $s+b$, while the latter having b as expectation value. Then the following likelihood ratio, for N events, can be stated:

$$X = \frac{\exp\left(-(s+b)\right)(s+b)^N}{N!} / \frac{\exp\left(-b\right)b^N}{N!} \tag{A.2}$$

Then for N_{obs}, a value X_{obs} is computed. Then CL_b is defined as $P(X_b \leq X_{obs})$, and can be computed with the help of Monte Carlo integration, by sampling $Poisson(N_b; b)$ with a fixed value of s. In a similar manner, CL_{s+b} is defined as $P(X_{s+b} \leq X_{obs})$, and can be computed also with Monte Carlo integration by sampling from $Poisson(N_{s+b}\ ; s+b)$. In case that the background estimation has uncertainties, then instead of sampling from a Poisson distribution, is preferable to sample from the convolution of a Normal distribution, with standard deviation σ and the corresponding Poisson distribution. Then for a given value of confidence α, the task is to find s_o such that the CL_s satisfies it.

The computed value of s_o can be thought as an upper limit such that the probability of having not observed a value $s > s_o$ is CL_s as is discussed in [21] and [22]. Since it is a usual method employ in different analysis in the NA62 collaboration, proper code for computing s_o was developed, and used in this work, and can be consulted in [23].

Appendix B

Analysis Code

```
/*
  * Name:    KpiNGamma.hh
  * Autor:   Jurgen Engelfried, Akbar Diaz
  * Date:    September, 2025
  * Description: Header file for KpiNGamma.cpp
*/

#ifndef KAONFLUX_HH
#define KAONFLUX_HH

#include <stdlib.h>
#include <vector>
#include <array>
#include <TCanvas.h>
#include <iostream>
#include "Analyzer.hh"
#include "EventHeader.hh"
#include "SpectrometerRICHAssociationAlgorithm.hh"
#include "TriggerConditions.hh"
#include "L0PrimitiveHandler.hh"
#include "BeamParameters.hh"
#include "MatchingGTKtoStrawVertex.hh"
//#include "L0TPSpecialTrigger.hh"

struct CedarKaonWindow{
  TH2D* Histogram;
  Double_t Width;
};

class KaonFlux : public NA62Analysis::Analyzer{
public:
  explicit KaonFlux(NA62Analysis::Core::BaseAnalysis *ba);
  ~KaonFlux();
  void InitHist();
  void InitOutput();
  void ProcessSpecialTriggerUser(int iEvent, unsigned int triggerType);
  void ProcessEOBEvent();
  void Process(int iEvent);
  void PostProcess();
  void StartOfBurstUser();
  void EndOfBurstUser();
  void StartOfRunUser();
  void EndOfRunUser();
  void EndOfJobUser();
  void DrawPlot();

protected:

private:
  //Debuging flags
  Bool_t fSkipL0DataCheck;
  Bool_t fSkipL1DataCheck;
  Bool_t fSkipBoseEinstein;
  Bool_t fSkipOutTime;
```

```
55    Bool_t fSaveTrees;
56    //Trigger Related Attributes
57    Int_t fEventID;
58    Int_t fTriggerDownscale;
59    L0PrimitiveHandler* fPrimitiveHandler;
60    NA62Analysis::Core::TriggerConditions *fTriggerConditions;
61    TriggerConditions::l0_alt_ids fTriggerIds;
62    std::vector<TString> fL0LinesMC;
63
64    //General Atributes
65    TString fInputFileName;
66    std::vector<Int_t> fOVK;
67    std::vector<Int_t> fOVPi;
68    std::vector<Int_t> fTrK;
69    std::vector<Int_t> fTrPi;
70
71    //for debugging
72    Int_t fC;
73    Int_t PrevBT;
74
75    //output Trees atributes
76    Int_t fRunID;
77    Int_t fTriggerTime;
78    Int_t fNSectors;
79    Int_t fBurstTimeScaler;
80    Int_t fBurstTime;
81    Int_t fEventTime;
82    Double_t fKaonTimes;
83    Double_t fArgonio1;
84    Double_t fArgonio2;
85    Double_t fT10;
86    Double_t fGtkIntensity;
87    Double_t fGtkIntensityE;
88    Double_t fCedarIntensity;
89    Double_t fCedarIntensityE;
90
91    // Run range for histograms vs RunNumber
92    const Int_t fMinRunID = 6278;   // start of sample 2016A
93    const Int_t fMaxRunID = 14603;  // end of sample 2024F
94
95    //Out of time, Beam Composition attributes:
96    CedarKaonWindow fCedarKaonsInit;
97    //General Sampling window attributes
98    Int_t fRUN;
99    Int_t fInitialDT;
100   //Postive DT sampling windows
101   std::vector<CedarKaonWindow> fCedarKaonsP;
102   std::vector<TH2D*> fCedarSectorsP;
103   //Negative DT samplint windows
104   std::vector<CedarKaonWindow> fCedarKaonsN;
105   std::vector<TH2D*> fCedarSectorsN;
106
107   //Bose-Einstein Atributes
108   std::unique_ptr<MatchingGTKtoStrawVertex> fMatchGTK;
109 };
110 #endif
```

```
1  /*
2    * Name:    KpiNGamma.cpp
3    * Autor:   Jurgen Engelfried, Akbar Diaz
4    * Date:    September, 2025
5    * Description: The following analyzer uses the NA62 Analysis
6      Frame Work. As input, filtered data with .root extension should
7      be provided. The output consists of histograms of the K+ invariant
8      mass for events with 2 (K2pi), 3 (Kpiggg) or 4 (K3pi0) nuetral
9      clusters present in the event. Complementary histograms related to
10     kinematic variables of the events are also present.
11 */
12
13 #include "KpiNGamma.hh"
14 #include <iostream>
15 #include <TChain.h>
16 #include "TGraphErrors.h"
```

```
17 #include "functions.hh"
18 #include "Event.hh"
19 #include "Persistency.hh"
20 #include "GeometricAcceptance.hh"
21 #include "DownstreamTrack.hh"
22 #include "SpectrometerRICHAssociation.hh"
23 #include "LAVMatching.hh"
24 #include "SAVMatching.hh"
25 #include "EnergyCluster.hh"
26 #include "BeamParameters.hh"
27 #include "SpectrometerTrackVertex.hh"
28 #include "RICHParameters.hh"
29 #include "PhotonVetoEfficiency.hh"
30 #include "L0PrimitiveHandler.hh"
31
32 using namespace NA62Analysis;
33 using namespace std;
34
35 struct GammaPair {
36   Int_t I;
37   Int_t J;
38   TLorentzVector FourMomentum;
39   Double_t Mass = FourMomentum.M();
40   Double_t Mass2 = FourMomentum.M2();
41 };
42
43 KpiNGamma::KpiNGamma(Core::BaseAnalysis *ba):Analyzer(ba,"KpiNGamma"){
44   RequestAllMCTrees();
45   RequestAllRecoTrees();
46
47
48   RequestTree("LAV", new TRecoLAVEvent, "Reco");
49   RequestTree("IRC", new TRecoIRCEvent, "Reco");
50   RequestTree("SAC", new TRecoSACEvent, "Reco");
51   RequestTree("RICH", new TRecoRICHEvent, "Reco");
52   RequestTree("Spectrometer", new TRecoSpectrometerEvent, "Reco");
53   RequestTree("GigaTracker",new TRecoGigaTrackerEvent, "Reco");
54   RequestTree("Cedar",new TRecoCedarEvent, "Reco");
55
56   RequestL0Data();
57   fPrimitiveHandler = L0PrimitiveHandler::GetInstance();
58   fTriggerConditions = TriggerConditions::GetInstance();
59   fPrimitiveHandler->DeclareL0Emulators(fParent);
60
61   fMatchingRG = new MatchingRG(ba,this,"MatchingRG");
62   fMatchingRG->InitForProcess("");
63   fMatchingRG->InitForFinalSelection("");
64 }
65
66 void KpiNGamma::InitOutput(){
67 }
68
69 void KpiNGamma::InitHist(){
70   fInputFileName = "bla";
71   BookHisto(new TH1D("OriginalEvents",
72                      "Number of Events from MC"
73                      ,200,0.,200000.));
74
75   //Trigger Histograms:
76   BookHisto(new TH1D("NEventsPhysicsTrigger",
77                      "Is Physics trigger; yes = 1, no = 0",
78                      2,-0.5,1.5));
79   BookHisto(new TH1D("CTRLChecker",
80                      "Is Control Trigger; yes = 1, no = 0"
81                      ,2,-0.5,1.5));
82
83   //Event Info Hitogramas
84   BookHisto(new TH1D("DownstreamTracks",
85                      "Downstream Tracks in Event",
86                      10, -0.5,9.5));
87   BookHisto(new TH1D("NDecayedMC",
88                      "Number of decayed Kaons in MC",
89                      10, -0.5,9.5));
```

```
90    BookHisto(new TH1D("ClustersInEvent",
91                       "Lkr Clusters Formed in Event",
92                       10, -0.5,9.5));
93    BookHisto(new TH1D("TriggerBits",
94                       "Trigger bits; bit; NEvents",
95                        19,-0.5,18.5));
96    BookHisto(new TH1D("TriggerBitsWeighted",
97                        "Weighted Trigger bits; bit; NEvents",
98                        19,-0.5,18.5));
99    BookHisto(new TH1D("TimeDiffTriggerTrack",
100   "Time Diference RICH vs Trigger (PionHypo cut);
101    TimeDifference [ns];N Events/0.5ns"
102                       ,160,-40.,40. ));
103   BookHisto(new TH1D("EventsAfterCuts",
104                       "Remainning events after cuts",
105                       10,-0.5,9.5));
106
107   //Track Cuts Histograms
108   BookHisto(new TH1D("TrackChi2",
109                      "Chi2 of Tracks",
110                      25,0.,25.));
111   BookHisto(new TH1D("TrackCDA",
112                      "Track CDA to Beam; Distance [mm];Events/1mm",
113                      30,0.,30.));
114   BookHisto(new TH1D("KTAGGTKATime",
115             "GTK Best Candidate Time vs KTAG Time Difference;
116             Time [ns]; Events/[50ps]",
117             200,-5.,5.));
118   BookHisto(new TH1D("TimeDiffNClusterVsTrack",
119             "Time difference Track vs Lkr Clusters;
120             Time [ns]; Events/[50ps]",
121             400,-10.,10.));
122   BookHisto(new TH1D("CloseTrackTimes",
123             "Time Difference Between Tracks;
124             Time [ns];Events/[0.5ns]",
125             160,-40.,40.));
126
127   //Lkr Cluster Cuts Histograms
128   BookHisto(new TH2D("ClustersEnergyVsMaxRMS",
129             "Energy vs MaxRMS form LKr Clusters;Energy [MeV];
130             Max RMS [mm]",
131             80,0.,80000.,80,0.,40.));
132   BookHisto(new TH1D("TrackLAVHitsDt0",
133             "Track vs LAVs Hits absolute time difference before cut;
134             |TrackTime-HitTime| [ns]",
135             250,-25.,25.));
136
137   //2 Neutral Cluster Histograms
138   BookHisto(new TH1D("2Clusters/EventsAfterCuts",
139            "Events remaining after aplaying cuts; Cut number;
140             Remaining Events"
141              ,9,-0.5,8.5));
142   BookHisto(new TH1D("2Clusters/MassKaonCand0",
143             "K^{+} Mass no cuts, 2 Clusters ;
144             Mass #pi^{+} #gamma #gamma [MeV/C^{2}];
145             Events/1MeV/C^{2}",
146             600,150.,750.));
147   BookHisto(new TH1D("2Clusters/MassKaonCand1",
148             "K^{+} Mass P cut, 2 Clusters ;
149              Mass #pi^{+} #gamma #gamma [MeV/C^{2}];
150              Events/1MeV/C^{2}",
151              600,150.,750.));
152   BookHisto(new TH1D("2Clusters/MassKaonCand2",
153             "K^{+} Mass Pt cut, 2 Clusters ;
154             Mass #pi^{+} #gamma #gamma [MeV/C^{2}];
155             Events/1MeV/C^{2}",
156             600,150.,750.));
157   BookHisto(new TH1D("2Clusters/MassKaonCand99",
158             "K^{+} Mass #pi^{0} cut, 2 Clusters ;
159             Mass #pi^{+} #pi^{0} [MeV/C^{2}];
160             Events/1MeV/C^{2}",
161             600,150.,750.));
162
```

```
163     BookHisto(new TH1D("2Clusters/Mass2Gamma0",
164               "2#gammas Mass no cuts, 2 Clusters ;
165               Mass #gamma #gamma [MeV/C^{2}];
166               Events/1MeV/C^{2}",
167               200,0.,200.));
168     BookHisto(new TH1D("2Clusters/Mass2Gamma1",
169               "2#gammas Mass P cut, 2 Clusters ;
170               Mass #gamma #gamma [MeV/C^{2}];
171               Events/1MeV/C^{2}",
172               200,0.,200.));
173     BookHisto(new TH1D("2Clusters/Mass2Gamma2",
174               "2#gammas Mass Pt cut, 2 Clusters ;
175               Mass #gamma #gamma [MeV/C^{2}];
176               Events/1MeV/C^{2}",
177               200,0.,200.));
178     BookHisto(new TH1D("2Clusters/Mass2Gamma99",
179               "2#gammas Mass #pi^{0} cut, 2 Clusters ;
180               Mass #gamma #gamma [MeV/C^{2}];
181               Events/1MeV/C^{2}",
182               200,0.,200.));
183
184     BookHisto(new TH1D("2Clusters/GammasEnergy0",
185               "Energy from #gammas no cuts, 2 Clusters ;
186               Energy [MeV];Events/500MeV ",
187               150,0.,75000));
188     BookHisto(new TH1D("2Clusters/GammasEnergy1",
189               "Energy from #gammas P cut, 2 Clusters ;
190               Energy [MeV];Events/500MeV ",
191               150,0.,75000));
192     BookHisto(new TH1D("2Clusters/GammasEnergy2",
193               "Energy from #gammas Pt cut, 2 Clusters ;
194               Energy [MeV];Events/500MeV ",
195               150,0.,75000));
196     BookHisto(new TH1D("2Clusters/GammasEnergy99",
197               "Energy from #gammas #pi^{0} cut, 2 Clusters ;
198               Energy [MeV];Events/500MeV ",
199               150,0.,75000));
200
201     BookHisto(new TH1D("2Clusters/NumberKineParts0",
202               "Number of KineParts 2 Clusters ;
203               NKineparts()",
204               15,-0.5,14.5));
205     BookHisto(new TH1D("2Clusters/NumberKineParts1",
206               "Number of KineParts P cut 2 Clusters ;
207               NKineparts()",
208               15,-0.5,14.5));
209     BookHisto(new TH1D("2Clusters/NumberKineParts2",
210               "Number of KineParts Pt cut 2 Clusters ;
211               NKineparts()",
212               15,-0.5,14.5));
213     BookHisto(new TH1D("2Clusters/NumberKineParts99",
214               "Number of KineParts #pi^{0} cut 2 Clusters ;
215               NKineparts()",
216               15,-0.5,14.5));
217
218     BookHisto(new TH1D("2Clusters/GTKKaonCandMomentumDiff0",
219               "Momentum Difference KaonCand to Beam 2 Clusters;
220               P [MeV/C]",
221               100,-10000.,10000.));
222     BookHisto(new TH1D("2Clusters/GTKKaonCandMomentumDiff1",
223               "Momentum Difference KaonCand to Beam P cut 2 Clusters;
224               P [MeV/C]",100,-10000.,10000.));
225
226     BookHisto(new TH1D("2Clusters/PtToBeamKaonCand0",
227               "Pt to Beam KaonCand, 2 Clusters;
228               Pt [MeV/C]",
229               100,0.,200.));
230     BookHisto(new TH1D("2Clusters/PtToBeamKaonCand1",
231               "Pt to Beam KaonCand P, cut 2 Clusters;
232               Pt [MeV/C]",
233               100,0.,200.));
234     BookHisto(new TH1D("2Clusters/PtToBeamKaonCand2",
235               "Pt to Beam KaonCand Pt, cut 2 Clusters;
```

```
236             Pt [MeV/C]",
237             100,0.,200.));
238
239   //3 Neutral Cluster Histograms
240   BookHisto(new TH1D("3Clusters/EventsAfterCuts",
241             "Events remaining after aplaying cuts;
242             Cut number;
243             Remaining Events",
244             9,-0.5,8.5));
245   BookHisto(new TH1D("3Clusters/MassKaonCand0",
246             "K^{+} Mass no cuts, 3 Clusters;
247             Mass #pi^{+} #gamma #gamma #gamma [MeV/C^{2}];
248             Events/1MeV/C^{2}",
249             600,150.,750.));
250   BookHisto(new TH1D("3Clusters/MassKaonCand1",
251             "K^{+} Mass After P cut, 3 Clusters;
252             Mass #pi^{+} #gamma #gamma #gamma [MeV/C^{2}];
253             Events/1MeV/C^{2}",
254             600,150.,750.));
255   BookHisto(new TH1D("3Clusters/MassKaonCand2",
256             "K^{+} Mass After Pt cut, 3 Clusters;
257             Mass #pi^{+} #gamma #gamma #gamma [MeV/C^{2}];
258             Events/1MeV/C^{2}",
259             600,150.,750.));
260   BookHisto(new TH1D("3Clusters/MassKaonCand3",
261             "K^{+} Mass After (P_{K+}-P_{#pi+})^{2} cut, 3 Clusters;
262             Mass #pi^{+} #gamma #gamma #gamma [MeV/C^{2}];
263             Events/1MeV/C^{2}",
264             600,150.,750.));
265   BookHisto(new TH1D("3Clusters/MassKaonCand4",
266             "K^{+} Mass After (P_{K+}-P_{#pi+}-P_{#gamma})^{2} cut,
267             3 Clusters;
268             Mass #pi^{+} #gamma #gamma #gamma [MeV/C^{2}];
269             Events/1MeV/C^{2}",
270             600,150.,750.));
271   BookHisto(new TH1D("3Clusters/MassKaonCand5",
272             "K^{+} Mass After Merged Cluster cut, 3 Clusters;
273             Mass #pi^{+} #gamma #gamma #gamma [MeV/C^{2}];
274             Events/1MeV/C^{2}",
275             600,150.,750.));
276   BookHisto(new TH1D("3Clusters/MassKaonCand6",
277             "K^{+} Mass After Asymmetry cut, 3 Clusters;
278             Mass #pi^{+} #gamma #gamma #gamma [MeV/C^{2}];
279             Events/1MeV/C^{2}",
280             600,150.,750.));
281   BookHisto(new TH1D("3Clusters/MassKaonCand7",
282             "K^{+} Mass After Missing Mass and 2#gamma Mass cut,
283             3 Clusters;
284             Mass #pi^{+} #gamma #gamma #gamma [MeV/C^{2}];
285             Events/1MeV/C^{2}",
286             600,150.,750.));
287   BookHisto(new TH1D("3Clusters/MassKaonCand99",
288             "K^{+} Mass After #gamma from #pi^{0} cut, 3 Clusters;
289             Mass #pi^{+}#gamma#gamma#gamma [MeV/C^{2}];
290             Events/1MeV/C^{2}",
291             600,150.,750.));
292   BookHisto(new TH1D("3Clusters/MassKaonCand100",
293             "K^{+} Mass After K3#pi^{0} cut, 3 Clusters ;
294             Mass #pi^{+} #gamma #gamma #gamma [MeV/C^{2}];
295             Events/1MeV/C^{2}",
296             600,150.,750.));
297
298   BookHisto(new TH1D("3Clusters/GammasEnergy0",
299             "Energy from #gammas no cuts, 3 Clusters ;
300             Energy [MeV];Events/500MeV ",
301             150,0.,75000));
302   BookHisto(new TH1D("3Clusters/GammasEnergy2",
303             "Energy from #gammas After P&Pt cuts, 3 Clusters ;
304             Energy [MeV];Events/500MeV ",
305             150,0.,75000));
306   BookHisto(new TH1D("3Clusters/GammasEnergy99",
307             "Energy from #gammas After #pi^{0} Mass cut,3 Clusters ;
308             Energy [MeV];Events/500MeV ",
```

```
309             150,0.,75000));
310
311     BookHisto(new TH1D("3Clusters/Mass2Gamma0",
312             "2#gammas Mass no cuts, 3 Clusters ;
313             Mass #gamma #gamma [MeV/C^{2}];
314             Events/1MeV/C^{2}",
315             200,0.,200.));
316     BookHisto(new TH1D("3Clusters/Mass2Gamma2",
317             "2#gammas Mass After P&Pt cuts, 3 Clusters ;
318             Mass #gamma #gamma [MeV/C^{2}];
319             Events/1MeV/C^{2}",
320             200,0.,200.));
321     BookHisto(new TH1D("3Clusters/Mass2Gamma99",
322             "2#gammas Mass After #pi^{0} Mass cut, 3 Clusters ;
323             Mass #gamma #gamma [MeV/C^{2}];
324             Events/1MeV/C^{2}",
325             200,0.,200.));
326
327     BookHisto(new TH1D("3Clusters/NumberKineParts0",
328             "Number of KineParts no cuts,3 Clusters ;
329             NKineparts()",
330             15,-0.5,14.5));
331     BookHisto(new TH1D("3Clusters/NumberKineParts1",
332             "Number of KineParts  After Pt cut,3 Clusters ;
333             NKineparts()",
334             15,-0.5,14.5));
335     BookHisto(new TH1D("3Clusters/NumberKineParts2",
336             "Number of KineParts  After P&Pt cuts,3 Clusters ;
337             NKineparts()",
338             15,-0.5,14.5));
339     BookHisto(new TH1D("3Clusters/NumberKineParts99",
340             "Number of KineParts After #pi0 Mass cut,3 Clusters ;
341             NKineparts()",
342             15,-0.5,14.5));
343
344     BookHisto(new TH1D("3Clusters/PtToBeamKaonCand",
345             "Pt to Beam KaonCand 3Clusters;
346             Pt [MeV/C]",
347             100,0.,200.));
348     BookHisto(new TH1D("3Clusters/GTKKaonCandMomentumDiff",
349             "Momentum Difference KaonCand to Beam 3 Clusters;
350             P [MeV/C]",
351             100,-10000.,10000.));
352
353     BookHisto(new TH1D("3Clusters/RemainingFMomentumMass",
354             "Missing Mass; P_{K^{+}}-P_{#pi^{+}} Mass[MeV/c^{2}];
355             Events",
356             750,0.,750.));
357     BookHisto(new TH1D("3Clusters/K2piGCandMass",
358             "Two Gammas Fourmomentum Mass;
359             Mass(P_{#gamma1#gamma2}) =
360             Mass(P_{K^{+}}-P_{#pi^{+}}-P_{#gamma}) [MeV/c^{2}]"
361             ,800,-400.,400.));
362
363     /*BG variables*/
364     BookHisto(new TH1D("3Clusters/BGVariables/NKineparts",
365             "KineParts K^+ mass final",
366             13,-6.5,6.5));
367     BookHisto(new TH2D("3Clusters/BGVariables/ClustersEnergyVsMaxRMS",
368             "Energy vs MaxRMS from LKr Clusters K^+ mass final;
369             Energy [MeV];
370             Max RMS [mm]",
371             80,0.,80000.,80,0.,40.));
372     BookHisto(new TH1D("3Clusters/BGVariables/Mass2Gamma",
373             "2#gammas Mass K^+ Mass final, 3 Clusters ;
374             Mass #gamma #gamma [MeV/C^{2}]; Events/1MeV/C^{2}",
375             200,0.,200.));
376     BookHisto(new TH1D("3Clusters/BGVariables/DDeadCell",
377             "Clusters Distance To Dead Cells;
378             Distance [mm]: N Events",
379             250,0,2500.));
380     BookHisto(new TH1D("3Clusters/BGVariables/MinDisClus",
381             "Minimum Distance Between Clusters;
```

```
            Distance [mm];
            Events/[10 mm]",
            2500,0.,25000.));
  BookHisto(new TH1D("3Clusters/BGVariables/K2piGCandMass",
            "(Mass#gamma#gamma)^{2};
            |P_{K^{+}} - P_{#pi^{+}} - P_{#gamma}|^{2}
            [100(MeV/c^{2})^{2}]",
            306,0.,122400.));
  BookHisto(new TH1D("3Clusters/BGVariables/MissingMass",
            "(Mass#gamma#gamma#gamma)^{2} ;
            (P_{K+}} - P_{#pi^{+}})^{2} [(MeV/c^{2})^{2}]",
            1225,0.,122500.));
  BookHisto(new TH2D("3Clusters/BGVariables/MissingMassVsK2piGCandMass",
            "(Mass#gamma#gamma#gamma)^{2}
            Vs (Mass#gamma#gamma)^{2};
            (P_{K+}} - P_{#pi^{+}})^{2}[(MeV/c^{2})^{2}];
            |P_{K^{+}} - P_{#pi^{+}} - P_{#gamma}|^{2}
            [100(MeV/c^{2})^{2}]"
            ,1225,0.,122500.,306,0.,122400.));

  BookHisto(new TH2D("3Clusters/IndirectGammaMass",
            "K^{+} vs Mass #gamma;
            #pi^{+}#gamma#gamma#gamma;
            Mass #gamma",
            150,400.,550.,374,-10000,27400));

  //K3pi0 histograms
  BookHisto(new TH1D("4Clusters/EventsAfterCuts",
            "Events remaining after aplaying cuts;
            Cut number;
            Remaining Events",
            9,-0.5,8.5));
  BookHisto(new TH1D("4Clusters/MassKaonCand0",
            "K^{+} Mass no cuts, 4 Clusters ;
            Mass #pi^{+} 4#gammas [MeV/C^{2}];
            Events/1MeV/C^{2}",
            600,150.,750.));
  BookHisto(new TH1D("4Clusters/MassKaonCand1",
            "K^{+} Mass After P cut, 4 Clusters ;
            Mass #pi^{+} 4#gammas [MeV/C^{2}];
            Events/1MeV/C^{2}",
            600,150.,750.));
  BookHisto(new TH1D("4Clusters/MassKaonCand2",
            "K^{+} Mass After Pt cut, 4 Clusters ;
            Mass #pi^{+} 4#gammas [MeV/C^{2}];
            Events/1MeV/C^{2}",
            600,150.,750.));
  BookHisto(new TH1D("4Clusters/MassKaonCand99",
            "K^{+} Mass After #pi^{0} cut, 4 Clusters ;
            Mass #pi^{+}#pi^{0}#pi^{0} [MeV/C^{2}];
            Events/1MeV/C^{2}",
            600,150.,750.));

  BookHisto(new TH1D("4Clusters/GammasEnergy0",
            "Energy from #gammas no cuts, 4 Clusters ;
            Energy [MeV];Events/500MeV ",
            150,0.,75000));
  BookHisto(new TH1D("4Clusters/GammasEnergy2",
            "Energy from #gammas After P&Pt cuts, 4 Clusters ;
            Energy [MeV];Events/500MeV ",
            150,0.,75000));
  BookHisto(new TH1D("4Clusters/GammasEnergy99",
            "Energy from #gammas After #pi^{0} Mass cut,4 Clusters ;
            Energy [MeV];Events/500MeV ",
            150,0.,75000));

  BookHisto(new TH1D("4Clusters/Mass2Gamma0",
            "2#gammas Mass no cuts, 4 Clusters ;
            Mass #gamma #gamma [MeV/C^{2}];
            Events/1MeV/C^{2}",
            200,0.,200.));
  BookHisto(new TH1D("4Clusters/Mass2Gamma2",
            "2#gammas Mass After P&Pt cuts, 4 Clusters ;
```

```
455             Mass #gamma #gamma [MeV/C^{2}];
456             Events/1MeV/C^{2}",
457             200,0.,200.));
458   BookHisto(new TH1D("4Clusters/Mass2Gamma99",
459             "2#gammas Mass After #pi^{0} Mass cut, 4 Clusters ;
460             Mass #gamma #gamma [MeV/C^{2}];
461             Events/1MeV/C^{2}",
462             200,0.,200.));
463   BookHisto(new TH1D("4Clusters/NumberKineParts0",
464             "Number of KineParts no cuts, 4 Clusters ;
465             NKineparts()",
466             15,-0.5,14.5));
467   BookHisto(new TH1D("4Clusters/NumberKineParts2",
468             "Number of KineParts  After P&Pt cuts, 4 Clusters ;
469             NKineparts()",
470             15,-0.5,14.5));
471   BookHisto(new TH1D("4Clusters/NumberKineParts99",
472             "Number of KineParts After #pi^{0} Mass cut, 4 Clusters ;
473             NKineparts()",
474             15,-0.5,14.5));
475   BookHisto(new TH1D("4Clusters/PtToBeamKaonCand",
476             "Pt to Beam KaonCand 4 Clusters;
477             Pt [MeV/C]",
478             100,0.,200.));
479   BookHisto(new TH1D("4Clusters/GTKKaonCandMomentumDiff",
480             "Momentum Difference KaonCand to Beam 4 Clusters;
481             P [MeV/C]",
482             100,-10000.,10000.));
483   BookHisto(new TH1D("4Clusters/Pi0Pairs",
484             "Number of Pi0 pairs, 4 Clusters;
485             Number of Pairs;
486             Number of Events",
487             4,-0.5,3.5));
488
489 }
490
491 void KpiNGamma::DefineMCSimple(){}
492 void KpiNGamma::StartOfRunUser(){}
493 void KpiNGamma::StartOfBurstUser(){}
494 void KpiNGamma::ProcessSpecialTriggerUser
495     (int iEvent, unsigned int triggerType){}
496
497 void KpiNGamma::KinePartsTypePloter
498   (Event* MCEVENT, std::string HistoName){
499   for(Int_t t= 0; t < MCEVENT->GetNKineParts(); t++){
500     switch(MCEVENT->GetKinePart(t)->GetPDGcode()){
501     case 22:/*foton*/      FillHisto(HistoName.c_str(),0);break;
502     case 11:/*electron*/   FillHisto(HistoName.c_str(),1);break;
503     case -11:/*positron*/  FillHisto(HistoName.c_str(),-1);break;
504     case 12:/*e neutrino*/   FillHisto(HistoName.c_str(),2);break;
505     case -12:/*e aneutrino*/FillHisto(HistoName.c_str(),-2);break;
506     case 13:/*Muon*/       FillHisto(HistoName.c_str(),3);break;
507     case -13:/*antimuon*/    FillHisto(HistoName.c_str(),-3);break;
508     case 14:/*mu neutrino*/  FillHisto(HistoName.c_str(),4);break;
509     case -14:/*mu anneutrino*/FillHisto(HistoName.c_str(),-4);break;
510     case 211:/*Pi+*/         FillHisto(HistoName.c_str(),5);break;
511     case 321:/*Kaon*/        FillHisto(HistoName.c_str(),6);break;
512     default: FillHisto(HistoName.c_str(),999);break;
513     }
514   }
515 }
516
517 void KpiNGamma::Process(int iEvent){
518
519   if (GetWithMC()) {
520     if (!fInputFileName.EqualTo(GetCurrentFile()->GetName())) {
521       fInputFileName = GetCurrentFile()->GetName();
522       std::vector<Int_t> OriginalEvents = GetMCInfo()->GetNEvents();
523       for (Int_t i = 0; i<OriginalEvents.size(); i++){
524   FillHisto("OriginalEvents",OriginalEvents[i],OriginalEvents[i]);
525       }
526     }
527   }
```

```
528
529   TRecoLAVEvent* LAVEvent = GetEvent<TRecoLAVEvent>();
530   TRecoIRCEvent* IRCEvent = GetEvent<TRecoIRCEvent>();
531   TRecoSACEvent* SACEvent = GetEvent<TRecoSACEvent>();
532   TRecoRICHEvent* RICHEvent = GetEvent<TRecoRICHEvent>();
533   TRecoLKrEvent* LKrEvent = GetEvent<TRecoLKrEvent>();
534   TRecoSpectrometerEvent* STRAWEvent =
535                           GetEvent<TRecoSpectrometerEvent>();
536   TRecoGigaTrackerEvent* GTKEvent =
537                           GetEvent<TRecoGigaTrackerEvent>();
538   TRecoCedarEvent* CEDAREvent =
539                           GetEvent<TRecoCedarEvent>();
540
541   Event* MCEvent = GetMCEvent();
542
543   Int_t RunID = GetEventHeader()->GetRunID();
544   Int_t EventID = GetEventHeader()->GetEventNumber();
545
546   L0TPData* L0Data = UserMethods::GetL0Data();
547   UInt_t TriggerFlags = L0Data->GetTriggerFlags();
548   Double_t TriggerTime =
549   L0Data->GetReferenceFineTime()*
550   NA62ConditionsService::GetInstance()->GetTdcCalib();
551   UInt_t TriggerType  = L0Data->GetDataType();
552   Bool_t PhysicsTrigger = false;
553   if (TriggerType & 0x1)PhysicsTrigger = true;
554   Bool_t ControlTrigger = false;
555   ControlTrigger = fTriggerConditions->IsControlTrigger(GetL0Data());
556
557   //Trigger Emulation
558   Bool_t Trigger[17]={false,false,false,false,
559   false,false,false,false,
560   false,false,false,false,
561   false,false,false,false,
562   false};
563   Double_t Downscaling[17]={1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1};
564   for (int icond=0;
565        icond < fTriggerConditions->GetNumberOfL0Conditions() ;
566        icond++){
567     Int_t tBit = fTriggerConditions->GetTriggerBit(RunID,icond);
568     if (GetWithMC() && tBit>-1) Trigger[tBit] =
569     fPrimitiveHandler->CheckEmulatedPrimitives(
570     fTriggerConditions->GetL0ConditionName(icond),
571     L0Data->GetReferenceFineTime());
572     if (tBit>-1) Downscaling[tBit] =
573     fTriggerConditions->GetL0TriggerDownscaling(RunID,icond);
574   }
575   ///CONTROL TRIGGER EMULATION FOR MC
576   /*
577   if (GetWithMC()) Trigger[16] =
578   fPrimitiveHandler->CheckEmulatedPrimitives("Q1",
579   L0Data->GetReferenceFineTime());
580   Downscaling[16] =
581   fTriggerConditions->GetControlTriggerDownscaling(RunID);
582   */
583   ///CONTROL TRIGGER for Data
584   if (!GetWithMC()) {
585     Int_t mask = 1;
586     for (Int_t tBit=0;tBit<16;tBit++) {
587       if (PhysicsTrigger) Trigger[tBit] = TriggerFlags & mask;
588       mask <<= 1;
589     }
590     Trigger[16] = fTriggerConditions->IsControlTrigger(GetL0Data());
591   }
592   for (Int_t tBit=0;tBit<17;tBit++) {
593     if (Trigger[tBit]) {
594       FillHisto("TriggerBits",tBit);
595       FillHisto("TriggerBitsWeighted",tBit,1./Downscaling[tBit]);
596     }
597   }
598
599   if(!GetWithMC() && !ControlTrigger) {
600     FillHisto("CTRLChecker", 0);
```

```
601       return;
602     }
603     FillHisto("CTRLChecker", 1);
604     if(!GetWithMC() && !PhysicsTrigger) {
605       FillHisto("NEventsPhysicsTrigger", 0.);
606     }
607     else{
608       FillHisto("NEventsPhysicsTrigger", 1.);
609     }
610
611
612     OutputState DownstreamState;
613     std::vector<DownstreamTrack> DownTracks =
614     *(std::vector<DownstreamTrack>*)
615     GetOutput("DownstreamTrackBuilder.Output",DownstreamState);
616     if(DownstreamState != kOValid) return;
617     FillHisto("DownstreamTracks",DownTracks.size());
618
619     const Double_t MinDecayZ = 110000.;
620     const Double_t MaxDecayZ = 170000.;
621
622     /*MC*/
623     Bool_t NoKdecayInVolume = false;
624     Int_t NKineParts=-1;//Number of kineparts in MC event
625     Int_t MCPi0Counter=-1;//Number of Pi0 in the MC event
626     Int_t MCGammaCounter=-1;//Number of Photons present in the event
627     //Index of the MC Gammas in the KineParts
628     std::vector <Int_t> MCGammaIndex;
629     //Index of the MC Electrons in the KineParts
630     std::vector <Int_t> MCElectronIndex;
631     //Index of the MC Muons and Neutrinos in the KineParts
632     std::vector <Int_t> MCMuonIndex;
633     std::vector <Int_t> MCMNeutrinoIndex;
634     std::vector <Int_t> MCENeutrinoIndex;
635     Int_t MCKaonIndex=-1;//Kaon Index of the MC event
636     //Int_t MCPionIndex=-1;//Indice del Pion en los KineParts
637     //Pares de Gammas del MC
638     std::vector<GammaPair> MCNotPi0GammaPairs;
639     std::vector<GammaPair> MCPi0GammaPairs;
640
641     enum MCMode {k2pi,k2pig,kpigg,kpiggg,k3pi0,k3pi0g,none};
642     MCMode CurrentMode = none;
643
644     if(GetWithMC()){
645       NKineParts= MCEvent->GetEventBoundary(0)->GetNKineParts();
646       if(NKineParts < 3) NoKdecayInVolume = true;
647       //MC Pi0 and photon counters
648       MCPi0Counter = 0;
649       for(Int_t IGene =
650               MCEvent->GetEventBoundary(0)->GetFirstGenePartIndex();
651       IGene <= MCEvent->GetEventBoundary(0)->GetLastGenePartIndex();
652       IGene++){
653         if(MCEvent->GetGenePart(IGene)->GetPDGcode()==111){
654     MCPi0Counter++;
655         }
656       }
657
658       for(Int_t IKine =
659       MCEvent->GetEventBoundary(0)->GetFirstKinePartIndex();
660       IKine <= MCEvent->GetEventBoundary(0)->GetLastKinePartIndex();
661       IKine++){
662         Int_t KineCode = MCEvent->GetKinePart(IKine)->GetPDGcode();
663         switch(KineCode){
664         //Gammas
665         case 22:            MCGammaIndex.push_back(IKine);       break;
666         //Electron or positron
667         case 11: case -11: MCElectronIndex.push_back(IKine);    break;
668         //Electron neutrino
669         case 12: case -12: MCENeutrinoIndex.push_back(IKine);   break;
670         //Pi+
671        //case 211:           MCPionIndex = IKine;               break;
672         //K+
673         case 321:           MCKaonIndex = IKine;                break;
```

```
674         //Muons
675         case 13: case -13: MCMuonIndex.push_back(IKine);          break;
676         //Muon Neutrino
677         case 14: case -14: MCMNeutrinoIndex.push_back(IKine);     break;
678         default:                                                  break;
679         }
680       }//End for(Int_t IKine = 0; IKine < NKineParts; IKine++)
681       MCGammaCounter = MCGammaIndex.size();
682
683       switch(MCPi0Counter){
684       case 0:
685         if(MCGammaCounter == 2) CurrentMode = kpigg;
686         if(MCGammaCounter == 3) CurrentMode = kpiggg;
687         break;
688       case 1:
689         if(MCGammaCounter == 2) CurrentMode = k2pi;
690         else if(MCGammaCounter == 3) CurrentMode = k2pig;
691         break;
692       case 2:
693         if(MCGammaCounter == 4) CurrentMode = k3pi0;
694         else if(MCGammaCounter == 5) CurrentMode = k3pi0g;
695         break;
696       default:break;
697       }//End switch(MCPi0Counter)
698
699       /*Creates a vector of GammaPairs 4Momentum and the index*/
700       if(MCGammaCounter > 1){
701         for(Int_t i =0;
702         i < MCGammaCounter && -1 != MCGammaIndex[i];
703         i++){
704     for(Int_t j = i+1;
705       j < MCGammaCounter && -1 != MCGammaIndex[j];
706       j++){
707       GammaPair GP;
708       GP.I = MCGammaIndex[i];
709       GP.J = MCGammaIndex[j];
710       GP.FourMomentum =
711         MCEvent->GetKinePart(MCGammaIndex[i])->GetInitial4Momentum()+
712           MCEvent->GetKinePart(MCGammaIndex[j])->GetInitial4Momentum();
713       if(fabs(GP.FourMomentum.M() - MP0) < 0.1){
714         MCPi0GammaPairs.push_back(GP);
715         }
716       else{
717         MCNotPi0GammaPairs.push_back(GP);
718         }
719     }//End for(Int_t j=0;j<MCGammaCounter && -1!=MCGammaIndex[j];j++)
720         }//End for(Int_t i =0;i<MCGammaCounter&&-1!=MCGammaIndex[i];i++)
721       }//End (MCGammaCounter > 0)
722       if(MCEvent->GetKinePart(MCKaonIndex)->GetEndPos().Z() < MinDecayZ
723       || MCEvent->GetKinePart(MCKaonIndex)->GetEndPos().Z() > MaxDecayZ)
724           NoKdecayInVolume = true;
725     }//End if(GetWithMC())
726
727
728     OutputState ClusterState;
729     std::vector<EnergyCluster> Clusters = *(std::vector<EnergyCluster>*)
730     GetOutput("EnergyClusterBuilder.Output",ClusterState);
731     if(ClusterState != kOValid) return;
732     FillHisto("ClustersInEvent",Clusters.size());
733
734     if(!NoKdecayInVolume && GetWithMC()) FillHisto("NDecayedMC",1);
735     if(Clusters.size() < 3) return;
736
737     for(UInt_t ITrack = 0 ; ITrack < DownTracks.size(); ITrack++){
738       if(!DownTracks[ITrack].RICHAssociationSuccessful()) continue;
739       if(DownTracks[ITrack].GetCharge() != 1) continue;
740       if(DownTracks[ITrack].GetRICHMostLikelyHypothesis()
741       != NA62::kRICHHypothesisPion) continue;
742       if(
743       !DownTracks[ITrack].RICHRingTimeExists(NA62::kRICHHypothesisPion))
744       continue;
745       if(!DownTracks[ITrack].LKrAssociationExists()) continue;
746       if(!DownTracks[ITrack].CHODTimeExists()) continue;
```

```
747     // if(!DownTracks[ITrack].RICHTimeExists()) continue;
748     //If Track has a succesful association, then a RICH time must exist
749     if(!GeometricAcceptance::GetInstance()->InAcceptance(
750     &DownTracks[ITrack],NA62::kMUV3)) continue;
751     if(DownTracks[ITrack].MUV3AssociationExists()) continue;
752     if(DownTracks[ITrack].GetNChambers()!=4)continue;
753
754     //Track time is defined by RICH, and it must be
755     //in time with the trigger
756     Double_t TrackTime  = DownTracks[ITrack].GetRICHRingTime(
757     NA62::kRICHHypothesisPion);
758     FillHisto("TimeDiffTriggerTrack",TriggerTime-TrackTime);
759     if(GetWithMC() && fabs(TriggerTime-TrackTime)>2.) continue;
760     if(!GetWithMC() && (TriggerTime-TrackTime < -7.
761     || TriggerTime-TrackTime > 3.5 )) continue;
762
763     //Track Criteria Cuts
764     const Double_t Chi2Limit = 20.;
765     FillHisto("TrackChi2", DownTracks[ITrack].GetChi2());
766     if(DownTracks[ITrack].GetChi2()>Chi2Limit) continue;
767
768     const Double_t MinMomentum = 1500.;
769     const Double_t MaxMomentum = 75000.;
770     if(DownTracks[ITrack].GetMomentum()<MinMomentum
771     || DownTracks[ITrack].GetMomentum()>MaxMomentum) continue;
772
773     const Double_t CDALimit = 25.;
774     FillHisto("TrackCDA",DownTracks[ITrack].GetBeamAxisCDA());
775     if(DownTracks[ITrack].GetBeamAxisCDA()>CDALimit) continue;
776     FillHisto("EventsAfterCuts",0);
777
778     //Events must Neither have 2 or more in time tracks,
779     //nor have 2 or more tracks coming from the same beam spot.
780     Bool_t CloseTrack = false;
781     const Double_t TTLimit = 21.;
782     for(UInt_t JTrack=0; JTrack<DownTracks.size();JTrack++){
783       if(JTrack == ITrack) continue;
784       FillHisto("CloseTrackTimes",
785       TrackTime-DownTracks[JTrack].GetTrackTime());
786       if(fabs(TrackTime-DownTracks[JTrack].GetTrackTime())<TTLimit)
787       CloseTrack=true;
788     }//End for(UInt_t JTrack=0;JTrack<DownTracks.size();JTrack++)
789     if(CloseTrack) continue;
790     FillHisto("EventsAfterCuts",1);
791
792     //Veto to LAV or SAV gammas
793     OutputState LAVState;
794     LAVMatching* LAVMatch= *(LAVMatching**)
795     GetOutput("PhotonVetoHandler.LAVMatching",LAVState);
796     LAVMatch->SetReferenceTime(TrackTime);
797     TClonesArray &HitsArray = (*(LAVEvent->GetHits()));
798     for(Int_t i=0; i<LAVEvent->GetNHits();i++){
799       TRecoLAVHit *Hit = static_cast<TRecoLAVHit*>(HitsArray[i]);
800       FillHisto("TrackLAVHitsDt0",Hit->GetTime()-TrackTime);
801     }
802     if(LAVMatch->LAVHasTimeMatching(LAVEvent,0.)) continue;
803     FillHisto("EventsAfterCuts",2);
804
805     OutputState SAVState;
806     //Wrapper para el SAV
807     SAVMatching* SAVMatch= *(SAVMatching**)
808     GetOutput("PhotonVetoHandler.SAVMatching",SAVState);
809     SAVMatch->SetReferenceTime(TrackTime);
810     if(SAVMatch->SAVHasTimeMatching(IRCEvent,SACEvent)) continue;
811     FillHisto("EventsAfterCuts",3);
812
813     //Matching for the spectrometer track and the GTk track candidate
814     TRecoSpectrometerCandidate *STRAWCand =
815     static_cast<TRecoSpectrometerCandidate*>
816     (STRAWEvent->GetCandidate(ITrack));
817     std::vector<TVector3>MatchedGTKMomenta;
818     std::vector<TVector3>MatchedVertices;
819     std::vector<Double_t>MatchedGTKTimes;
```

```
820       std::vector<Int_t>      MatchedGTKIDs;
821       Double_t KTAGTime = TrackTime;
822       Double_t BestTime;
823       TVector3 BestVertex;
824       fMatchingRG->Process(GTKEvent,
825                            STRAWCand,
826                            KTAGTime,
827                            KTAGTime,
828                            TrackTime,
829                            1,"");
830       fMatchingRG->FinalSelection(KTAGTime,TrackTime,1,"");
831       MatchedGTKIDs=fMatchingRG->GetMatchedGTKIDs();
832       if(MatchedGTKIDs.at(0)==-1) continue;
833       MatchedGTKMomenta = fMatchingRG->GetGTKMomentaAtVertices();
834       MatchedGTKTimes =   fMatchingRG->GetGTKTimes();
835       MatchedVertices =   fMatchingRG->GetVertices();
836       BestTime = MatchedGTKTimes.at(0);
837       BestVertex = MatchedVertices.at(0);
838       FillHisto("EventsAfterCuts",4);
839
840       if(BestVertex.Z()<MinDecayZ || BestVertex.Z()>MaxDecayZ) continue;
841       FillHisto("EventsAfterCuts",5);
842
843       const Double_t KTAGGTKTimeWindow = 0.4;
844       Bool_t IsKaon = false;
845       for(Int_t ICandidate = 0;
846       ICandidate < CEDAREvent->GetNCandidates();
847       ICandidate++){
848         TRecoCedarCandidate *CedarCandidate =
849         (TRecoCedarCandidate*)CEDAREvent->GetCandidate(ICandidate);
850         FillHisto("KTAGGTKATime",CedarCandidate->GetTime()-BestTime);
851         if(fabs(CedarCandidate->GetTime()-BestTime)<KTAGGTKTimeWindow
852        && CedarCandidate->GetNSectors()>4)
853         IsKaon = true;
854       }//Final for(ICandidates)
855       if(!IsKaon) continue;
856       FillHisto("EventsAfterCuts",6);
857
858       TLorentzVector PionCand;
859       PionCand.SetVectM(DownTracks[ITrack].GetMomentumBeforeMagnet(),MPI);
860       TLorentzVector GTKBeamKaon;
861       GTKBeamKaon.SetVectM(MatchedGTKMomenta.at(0),MKCH);
862
863       Double_t DClusterVsBestTrackCluster;
864       std::vector<UInt_t> NeutralCluster;
865       std::vector<UInt_t> GoodNeutralCluster;
866       UInt_t NeutralClusters = 0;
867       Double_t MinDis = 99999999999.;
868       Double_t AbsDistance = 99999999999.;
869       //Double_t MinDisCut  = 155.;//
870       for(UInt_t ICluster = 0; ICluster < Clusters.size(); ICluster++){
871         //Neutral Cluster Criteria:
872         if(Clusters[ICluster].SpectrometerAssociationExists())
873           continue;
874         FillHisto("TimeDiffNClusterVsTrack",
875         Clusters[ICluster].GetTime()-TrackTime);
876         if(Clusters[ICluster].GetTime()-TrackTime < -3.
877         || Clusters[ICluster].GetTime()-TrackTime > 5.)
878           continue;
879         if(Clusters[ICluster].GetEnergy()>55e3)continue;
880         if(Clusters[ICluster].GetEnergy()<2e3)continue;
881         FillHisto("ClustersEnergyVsMaxRMS",
882         Clusters[ICluster].GetEnergy(),
883         Clusters[ICluster].GetLKrCandidate()->GetMaxSize());
884         if(Clusters[ICluster].GetLKrCandidate()->GetMaxSize() > 20.)
885           continue;
886         if(Clusters[ICluster].GetLKrCandidate()->GetMaxSize() < 10.)
887           continue;
888         NeutralCluster.push_back(ICluster);
889
890         //Good Neutral Cluster Criteria:
891         DClusterVsBestTrackCluster =
892           TMath::Power((DownTracks[ITrack].GetLKrClusterX()
```

```
                        -Clusters[ICluster].GetX()),2.)
          +TMath::Power((DownTracks[ITrack].GetLKrClusterY()
                        -Clusters[ICluster].GetY()),2.);
        if(DClusterVsBestTrackCluster < 250.*250.) continue;
        if(Clusters[ICluster].GetDDeadCell() < 20.) continue;
        GoodNeutralCluster.push_back(ICluster);
      }//Final for(ICluster)
      NeutralClusters = NeutralCluster.size();

      if(NeutralClusters < 2 || NeutralClusters > 4) continue;
      FillHisto("EventsAfterCuts",7);

      std::vector<TLorentzVector> Gammas;
      for(UInt_t i = 0; i< NeutralCluster.size(); i++){
        Gammas.push_back(
        Clusters[NeutralCluster[i]].PhotonMomentum(BestVertex));
      }

      std::vector <GammaPair> GammaPairs;
      for(UInt_t i= 0;i < NeutralCluster.size()-1; i++ ){
        for(UInt_t j=i+1;j<NeutralCluster.size(); j++){
    GammaPair GP;
    GP.FourMomentum = Gammas[i]+Gammas[j];
    GP.I = i;
    GP.J = j;
    GammaPairs.push_back(GP);
        }
      }

      Int_t Pi0Pairs = 0;
      Double_t Pi0MassWidth = 15.;
      std::vector<GammaPair> Pi0GammaPairs;
      for(const GammaPair & GP : GammaPairs){
        if(fabs( GP.FourMomentum.M() - MP0) < Pi0MassWidth){
    Pi0GammaPairs.push_back(GP);
        }
      }
      Pi0Pairs = Pi0GammaPairs.size();

      TLorentzVector KaonCand;
      Double_t MassKaonCand;
      KaonCand = PionCand;
      for(UInt_t i = 0 ; i< Gammas.size() ;i++)
      {KaonCand += Gammas[i];}
      MassKaonCand = KaonCand.M();

      const Double_t DeltaPGTKKaon = 2000.;
      const Double_t Pt2Cut = 20.*20.;

      //2 Clusters
      if( NeutralClusters == 2
      && GoodNeutralCluster.size() == NeutralClusters ){
        //No Cuts
        FillHisto("2Clusters/EventsAfterCuts",0);
        FillHisto("2Clusters/MassKaonCand0",MassKaonCand);
        FillHisto("2Clusters/Mass2Gamma0",GammaPairs[0].FourMomentum.M());
        for(UInt_t i =0; i < Gammas.size() ; i++){
    FillHisto("2Clusters/GammasEnergy0",Gammas[i].E());}
        FillHisto("2Clusters/NumberKineParts0",NKineParts);
        FillHisto("2Clusters/GTKKaonCandMomentumDiff0",
        KaonCand.P()-GTKBeamKaon.P());
        FillHisto("2Clusters/PtToBeamKaonCand0",
        KaonCand.Perp(GTKBeamKaon.Vect()));

        if(fabs(KaonCand.P()-GTKBeamKaon.P())>DeltaPGTKKaon)continue;
        FillHisto("2Clusters/EventsAfterCuts",1);
        FillHisto("2Clusters/MassKaonCand1",MassKaonCand);
        FillHisto("2Clusters/Mass2Gamma1",GammaPairs[0].FourMomentum.M());
        for(UInt_t i =0; i < Gammas.size(); i++){
    FillHisto("2Clusters/GammasEnergy1",Gammas[i].E());}
        FillHisto("2Clusters/NumberKineParts1",NKineParts);
        FillHisto("2Clusters/GTKKaonCandMomentumDiff1",
        KaonCand.P()-GTKBeamKaon.P());
```

```
966         FillHisto("2Clusters/PtToBeamKaonCand1",
967         KaonCand.Perp(GTKBeamKaon.Vect()));
968
969         if(KaonCand.Perp2(GTKBeamKaon.Vect()) > Pt2Cut)continue;
970         FillHisto("2Clusters/EventsAfterCuts",2);
971         FillHisto("2Clusters/MassKaonCand2",MassKaonCand);
972         FillHisto("2Clusters/Mass2Gamma2",GammaPairs[0].FourMomentum.M());
973         for(UInt_t i =0; i < Gammas.size() ; i++){
974     FillHisto("2Clusters/GammasEnergy2",Gammas[i].E());}
975         FillHisto("2Clusters/NumberKineParts2",NKineParts);
976         FillHisto("2Clusters/PtToBeamKaonCand2",
977         KaonCand.Perp(GTKBeamKaon.Vect()));
978
979         if(Pi0Pairs != 1) continue;
980         FillHisto("2Clusters/EventsAfterCuts",8);
981         TLorentzVector Pi0Cand;
982         Pi0Cand.SetVectM(GammaPairs[0].FourMomentum.Vect(),MP0);
983         KaonCand = PionCand + Pi0Cand;
984         FillHisto("2Clusters/MassKaonCand99",MassKaonCand);
985
986         FillHisto("2Clusters/Mass2Gamma99",
987         GammaPairs[0].FourMomentum.M());
988         for(UInt_t i =0; i < Gammas.size(); i++){
989     FillHisto("2Clusters/GammasEnergy99",Gammas[i].E());}
990         FillHisto("2Clusters/NumberKineParts99",NKineParts);
991       }//End if(GoodNeutralCluster.size() == 2&&NeutralClusters == 2)
992
993       //Rutina para KPiGGG
994       else if(NeutralClusters == 3){
995         if(!GetWithMC()
996         && (MassKaonCand > 476. && MassKaonCand < 510.))
997           continue;
998         FillHisto("3Clusters/EventsAfterCuts",0);
999         FillHisto("3Clusters/MassKaonCand0", MassKaonCand);
1000        FillHisto("3Clusters/NumberKineParts0",NKineParts);
1001        FillHisto("3Clusters/GTKKaonCandMomentumDiff",
1002        KaonCand.P()-GTKBeamKaon.P());
1003        FillHisto("3Clusters/PtToBeamKaonCand",
1004        KaonCand.Perp(GTKBeamKaon.Vect()));
1005        for( UInt_t i=0 ; i<Gammas.size() ; i++ ){
1006        FillHisto("3Clusters/GammasEnergy0", Gammas[i].E());}
1007        for(const GammaPair& GP : GammaPairs){
1008        FillHisto("3Clusters/Mass2Gamma0",GP.FourMomentum.M());}
1009
1010        if(fabs(KaonCand.P()-GTKBeamKaon.P())>DeltaPGTKKaon)continue;
1011        FillHisto("3Clusters/EventsAfterCuts",1);
1012        FillHisto("3Clusters/NumberKineParts1",NKineParts);
1013        FillHisto("3Clusters/MassKaonCand1",MassKaonCand);
1014
1015        if(KaonCand.Perp2(GTKBeamKaon.Vect())>Pt2Cut)continue;
1016        FillHisto("3Clusters/EventsAfterCuts",2);
1017        FillHisto("3Clusters/MassKaonCand2", MassKaonCand);
1018        FillHisto("3Clusters/NumberKineParts2",NKineParts);
1019        for(UInt_t i = 0; i<Gammas.size();i++){
1020        FillHisto("3Clusters/GammasEnergy2", Gammas[i].E());}
1021        for(const GammaPair& GP : GammaPairs){
1022        FillHisto("3Clusters/Mass2Gamma2",GP.FourMomentum.M());}
1023
1024        //Missing Mass Cut (to avoid K->K->#pi+ #pi0-> e+e-#gamma
1025        //or #pi0->#gamma#gamma->e+e- convertion)
1026        TLorentzVector RemainingFMomentum = GTKBeamKaon-PionCand;
1027        if(RemainingFMomentum.M2() > 12e3
1028        && RemainingFMomentum.M2() < 24e3)
1029          continue;
1030        FillHisto("3Clusters/EventsAfterCuts",3);
1031        FillHisto("3Clusters/MassKaonCand3",MassKaonCand);
1032
1033        //K->#pi+ #pi0 #gamma cut
1034        TLorentzVector K2piGCand;
1035        Bool_t IsK2PiGamma = false;
1036        for(UInt_t i = 0; i<Gammas.size() ; i++){
1037    K2piGCand = RemainingFMomentum - Gammas[i];
1038    if(14e3 < K2piGCand.M2() && K2piGCand.M2()<22e3 )
```

```
1039      IsK2PiGamma = true;
1040        }//End for(UInt_t i = 0; i<NeutralClusters ; i++)
1041        if(IsK2PiGamma) continue;
1042        FillHisto("3Clusters/EventsAfterCuts",4);
1043        FillHisto("3Clusters/MassKaonCand4",MassKaonCand);
1044
1045        //Tagging of K3pi0 merged clusters:
1046        const Double_t MassConstant = MP0*MP0/2.;
1047        Double_t Thetaik = 0;
1048        Double_t Thetajk = 0;
1049        Double_t TaggingEnergy=0;
1050        Thetaik = Gammas[2].Angle(Gammas[0].Vect());
1051        Thetajk = Gammas[2].Angle(Gammas[1].Vect());
1052        TaggingEnergy = MassConstant*(1./(Gammas[0].E()*(1.-cos(Thetaik)))
1053        +1./(Gammas[1].E()*(1.-cos(Thetajk))));
1054        if(fabs(Gammas[2].E()-TaggingEnergy) < 1e3)
1055          continue;
1056        Thetaik = Gammas[1].Angle(Gammas[0].Vect());
1057        Thetajk = Gammas[1].Angle(Gammas[2].Vect());
1058        TaggingEnergy = MassConstant*(1./(Gammas[0].E()*(1.-cos(Thetaik)))
1059        +1./(Gammas[2].E()*(1.-cos(Thetajk))));
1060        if(fabs(Gammas[1].E()-TaggingEnergy) < 1e3)
1061          continue;
1062        Thetaik = Gammas[0].Angle(Gammas[1].Vect());
1063        Thetajk = Gammas[0].Angle(Gammas[2].Vect());
1064        TaggingEnergy = MassConstant*(1./(Gammas[1].E()*(1.-cos(Thetaik)))
1065        +1./(Gammas[2].E()*(1.-cos(Thetajk))));
1066        if(fabs(Gammas[0].E()-TaggingEnergy) < 1e3)
1067          continue;
1068        FillHisto("3Clusters/EventsAfterCuts",5);
1069        FillHisto("3Clusters/MassKaonCand5",MassKaonCand);
1070
1071        //Asymetry cut
1072        Bool_t AsymmetricCluster = false;
1073        Double_t AsymmetryCut = 0.85;
1074        for(const UInt_t &i: NeutralCluster){
1075    if(Clusters[i].GetLKrCandidate()->GetMinSize()
1076      /Clusters[i].GetLKrCandidate()->GetMaxSize()
1077      < AsymmetryCut) AsymmetricCluster = true;
1078        }
1079
1080        if(AsymmetricCluster) continue;
1081        FillHisto("3Clusters/EventsAfterCuts",6);
1082        FillHisto("3Clusters/MassKaonCand6", MassKaonCand);
1083
1084        Bool_t K3picut = false;
1085        for(UInt_t i = 0; i<Gammas.size() ; i++){
1086      K2piGCand = RemainingFMomentum - Gammas[i];
1087      FillHisto("3Clusters/BGVariables/MissingMassVsK2piGCandMass",
1088        RemainingFMomentum.M2(),K2piGCand.M2());
1089      if(RemainingFMomentum.M2() > 72e3 && K2piGCand.M2() > 20e3)
1090            K3picut = true;
1091        }
1092
1093        if(K3picut) continue;
1094        FillHisto("3Clusters/EventsAfterCuts",7);
1095        FillHisto("3Clusters/MassKaonCand7",MassKaonCand);
1096        for(UInt_t i = 0; i<Gammas.size() ; i++){
1097      K2piGCand = RemainingFMomentum - Gammas[i];
1098      FillHisto("3Clusters/BGVariables/MissingMassVsK2piGCandMass99",
1099        RemainingFMomentum.M2(),K2piGCand.M2());
1100      if(RemainingFMomentum.M2() > 72e3 && K2piGCand.M2() > 20e3)
1101        K3picut = true;
1102        }
1103
1104        if(Pi0Pairs != 0) continue;
1105        FillHisto("3Clusters/EventsAfterCuts",8);
1106        FillHisto("3Clusters/MassKaonCand99", MassKaonCand);
1107        FillHisto("3Clusters/NumberKineParts99",NKineParts);
1108        FillHisto("3Clusters/RemainingFMomentumMass",
1109        RemainingFMomentum.M());
1110        for(const GammaPair& GP : GammaPairs){
1111        FillHisto("3Clusters/Mass2Gamma99",GP.FourMomentum.M());}
```

```
1112        for(UInt_t i = 0; i<Gammas.size() ; i++){
1113    FillHisto("3Clusters/GammasEnergy99", Gammas[i].E());
1114    K2piGCand = RemainingFMomentum - Gammas[i];
1115    FillHisto("3Clusters/K2piGCandMass",K2piGCand.M());
1116        }
1117
1118        /*Important Variables to instpect in K^+ mass region*/
1119        if(MassKaonCand>476 && MassKaonCand<510 && GetWithMC()){
1120    FillHisto("3Clusters/BGVariables/MissingMass",
1121      RemainingFMomentum.M2());
1122    for(UInt_t i = 0; i<Gammas.size() ; i++){
1123      K2piGCand = RemainingFMomentum - Gammas[i];
1124      FillHisto("3Clusters/BGVariables/K2piGCandMass",
1125        K2piGCand.M2());
1126    }
1127    for(const GammaPair& GP : GammaPairs){
1128      FillHisto("3Clusters/BGVariables/Mass2Gamma",
1129      GP.FourMomentum.M());}
1130    if(GetWithMC())
1131          FillHisto("3Clusters/BGVariables/NKineparts",
1132          NKineParts);
1133    for(const UInt_t & i : NeutralCluster){
1134      FillHisto("3Clusters/BGVariables/ClustersEnergyVsMaxRMS",
1135        Clusters[i].GetEnergy(),
1136        Clusters[i].GetLKrCandidate()->GetMaxSize());
1137      MinDis=99999999999;
1138      AbsDistance = 99999999999;
1139      for(const UInt_t & j : NeutralCluster){
1140        if(j == i) continue;
1141        AbsDistance =
1142                  sqrt(pow(Clusters[i].GetX()-Clusters[j].GetX(),2.)
1143              +pow(Clusters[i].GetY()-Clusters[j].GetY(),2.));
1144        if(AbsDistance < MinDis) MinDis = AbsDistance;
1145      }//End for(JCluster)
1146      FillHisto("3Clusters/BGVariables/MinDisClus",MinDis);
1147      FillHisto("3Clusters/BGVariables/DDeadCell",
1148        Clusters[i].GetDDeadCell());
1149    }//End for(const Int_t & i : NeutralCluster)
1150        }//End if(MassKaonCand>476&&MassKaonCand<510&&GetWithMC())
1151
1152        /*No extra hits in LKr, cut*/
1153        TLorentzVector IndirectGammaMass;
1154        Bool_t IsK3pi0 = false;
1155        IndirectGammaMass = RemainingFMomentum - Gammas[0]-Gammas[1];
1156        FillHisto("3Clusters/IndirectGammaMass",
1157        MassKaonCand,IndirectGammaMass.M2());
1158        if(fabs(IndirectGammaMass.M2()-MP0*MP0)<2e3) IsK3pi0 = true;
1159        IndirectGammaMass = RemainingFMomentum - Gammas[0]-Gammas[2];
1160        FillHisto("3Clusters/IndirectGammaMass",
1161        MassKaonCand,IndirectGammaMass.M2());
1162        if(fabs(IndirectGammaMass.M2()-MP0*MP0)<2e3) IsK3pi0 = true;
1163        IndirectGammaMass = RemainingFMomentum - Gammas[1]-Gammas[2];
1164        FillHisto("3Clusters/IndirectGammaMass",
1165        MassKaonCand,IndirectGammaMass.M2());
1166        if(fabs(IndirectGammaMass.M2()-MP0*MP0)<2e3) IsK3pi0 = true;
1167
1168        if(IsK3pi0) continue;
1169        FillHisto("3Clusters/MassKaonCand100",MassKaonCand);
1170        FillHisto("3Clusters/EventsAfterCuts",10);
1171
1172      }//End else if(NeutralClusters == 3)
1173
1174      /*4 Neutral Clusters (K3pi0)*/
1175      else if( NeutralClusters == 4 &&
1176      GoodNeutralCluster.size() == NeutralClusters ){
1177        FillHisto("4Clusters/EventsAfterCuts",0);
1178        FillHisto("4Clusters/MassKaonCand0", MassKaonCand);
1179        FillHisto("4Clusters/NumberKineParts0",NKineParts);
1180        FillHisto("4Clusters/GTKKaonCandMomentumDiff",
1181        KaonCand.P()-GTKBeamKaon.P());
1182        FillHisto("4Clusters/PtToBeamKaonCand",
1183        KaonCand.Perp(GTKBeamKaon.Vect()));
1184        for(UInt_t i =0; i<Gammas.size();i++){
```

```
1185        FillHisto("4Clusters/GammasEnergy0", Gammas[i].E());}
1186        for(const GammaPair& GP : GammaPairs){
1187        FillHisto("4Clusters/Mass2Gamma0",GP.FourMomentum.M());}
1188
1189        if(fabs(KaonCand.P()-GTKBeamKaon.P())>DeltaPGTKKaon)continue;
1190        FillHisto("4Clusters/EventsAfterCuts",1);
1191        FillHisto("4Clusters/MassKaonCand1", MassKaonCand);
1192
1193        if(KaonCand.Perp2(GTKBeamKaon.Vect())>Pt2Cut)continue;
1194        FillHisto("4Clusters/EventsAfterCuts",2);
1195        FillHisto("4Clusters/MassKaonCand2", MassKaonCand);
1196        FillHisto("4Clusters/NumberKineParts2",NKineParts);
1197        FillHisto("4Clusters/Pi0Pairs",Pi0Pairs);
1198        for(UInt_t i=0; i < Gammas.size() ; i++){
1199        FillHisto("4Clusters/GammasEnergy2", Gammas[i].E());}
1200        for(const GammaPair& GP : GammaPairs){
1201        FillHisto("4Clusters/Mass2Gamma2",GP.FourMomentum.M());}
1202
1203        if(Pi0Pairs !=2 ) continue;
1204        TLorentzVector Pi0Cand1;
1205        Pi0Cand1.SetVectM(Pi0GammaPairs.at(0).FourMomentum.Vect(),MP0);
1206        TLorentzVector Pi0Cand2;
1207        Pi0Cand2.SetVectM(Pi0GammaPairs.at(1).FourMomentum.Vect(),MP0);
1208        KaonCand = Pi0Cand1+Pi0Cand2+PionCand;
1209        MassKaonCand = KaonCand.M();
1210        FillHisto("4Clusters/EventsAfterCuts",8);
1211        FillHisto("4Clusters/MassKaonCand99", MassKaonCand);
1212        FillHisto("4Clusters/NumberKineParts99",NKineParts);
1213        for(UInt_t i=0; i < Gammas.size(); i++){
1214        FillHisto("4Clusters/GammasEnergy99", Gammas[i].E()); }
1215        for(const GammaPair& GP : GammaPairs){
1216        FillHisto("4Clusters/Mass2Gamma99",GP.FourMomentum.M());}
1217
1218     }//End else if( NeutralClusters == 4
1219     //&&  GoodNeutralCluster.size() == NeutralClusters )
1220
1221   }//End for(UInt_t ITrack = 0;ITrack<DownTracks.size();ITrack++)
1222 }//End KpiNGamma::Process
1223
1224 void KpiNGamma::PostProcess(){}
1225 void KpiNGamma::EndOfBurstUser(){}
1226 void KpiNGamma::EndOfRunUser(){}
1227 void KpiNGamma::EndOfJobUser(){
1228   SaveAllPlots();
1229 }
1230 void KpiNGamma::DrawPlot(){}
1231
1232 //Class destructor
1233 KpiNGamma::~KpiNGamma(){
1234   delete fMatchingRG;
1235 }
```

Appendix C

Normalization and Background Counting Macro

```
/*
  * Name:    ExpectedEvents.cpp
  * Autor:   Jurgen Engelfried, Akbar Diaz
  * Date:    September, 2025
  * Description: The following ROOT CERN Macro employs
    *as inputs the data file with root extention
    *and the MC simulation files, obtained with KpiNGamma analyzer,
    *relative path to the files should be provided to the Macro
    *in the section "Files Path"
    *An an output, a PDF file is created with all relevan histograms
    *.txt files are also provided containing the scaling factors,
    *number of background event, etc.
*/

#include "ROOT/TypeTraits.hxx"
#include "Rtypes.h"
#include "RtypesCore.h"
#include "TBox.h"
#include "TH1.h"
#include "TF1.h"
#include "TLine.h"
#include "TPaveText.h"
#include "TROOT.h"
#include "TFile.h"
#include "TPaveStats.h"
#include "TColor.h"
#include "THStack.h"
#include "TH1D.h"
#include "TCanvas.h"
#include "TStyle.h"
#include "TText.h"
#include "TVirtualPad.h"
#include "TLegend.h"
#include <cmath>
#include <ios>
#include <iostream>
#include <math.h>
#include <ostream>
#include <string>
#include <fstream>

Double_t Polynomial(Double_t *x, Double_t *par){
  //par[5] is the value from which the polinomial will be adjusted.
   return par[0]+par[1]*(x[0]-par[5])+par[2]*pow(x[0]-par[5],2.)
  +par[3]*pow(x[0]-par[5],3.)+par[4]*pow(x[0]-par[5],4.);
}

void Similarity(TCanvas* Canvas_,THStack* NUMERATOR_,
TH1D* DENOMINATOR_,TLegend *LEGEND_, Double_t xmin,
Double_t xmax, Double_t xo,Int_t PolDegree) {
  TPad* UperPad = new TPad("UperPad","",0.,.3,1.,1.);
```

```
52  TPad* LowerPad = new TPad("LowerPad","",0.,0.05,1,.4);
53  LowerPad->SetFillStyle(4000);
54  Canvas_->Clear();
55  UperPad->Draw();
56  LowerPad->Draw();
57  Canvas_->Update();
58  Canvas_->Modified();
59
60  UperPad->cd();
61  DENOMINATOR_->Draw("E1");
62  NUMERATOR_->Draw("SAME HIST  E1");
63  DENOMINATOR_->Draw("SAME E1");
64  DENOMINATOR_->SetXTitle("");
65  DENOMINATOR_->SetLabelSize(0.0,"X");
66  DENOMINATOR_->SetTitleSize(0.05,"Y");
67  DENOMINATOR_->SetTitleOffset(0.8,"Y");
68  DENOMINATOR_->SetLabelSize(0.048,"Y");
69  DENOMINATOR_->GetYaxis()->CenterTitle();
70  LEGEND_->Draw("SAME");
71  UperPad->SetLogy();
72  UperPad->Modified();
73  TPaveStats *psu =
74    (TPaveStats*)DENOMINATOR_->GetListOfFunctions()->FindObject("stats");
75  psu->SetX1NDC(0.7); psu->SetX2NDC(0.9);
76  psu->SetY1NDC(0.60); psu->SetY2NDC(0.9);
77  UperPad->SetGrid();
78  UperPad->Update();
79
80  LowerPad->cd();
81  TH1D* Quotient;
82  Quotient=((TH1D*)NUMERATOR_->GetStack()->Last()->Clone());
83  Quotient->Divide(DENOMINATOR_);
84  Quotient->SetLineColor(kBlack);
85  Quotient->SetMarkerColor(kBlack);
86  Quotient->Draw("E1");
87  LowerPad->SetTickx(1);
88  Quotient->SetMinimum(0);
89  Quotient->SetMaximum(1.9);
90  Quotient->SetTitle("");
91  Quotient->SetYTitle("MC/Exp");
92  Quotient->SetTitleSize(.09,"Y");
93  Quotient->GetYaxis()->CenterTitle(true);
94  Quotient->SetTitleOffset(0.4,"Y");
95  Quotient->SetLabelSize(0.09,"Y");
96  Quotient->SetLabelSize(0.085,"X");
97
98  //Quotient->SetTitleOffset(0.75,"X");
99  Quotient->SetTitleSize(.0,"X");
100 Quotient->SetLabelOffset(0.03,"X");
101 Quotient->SetTickLength(0.01,"Y");
102
103 //Polynomial fit
104 TF1 *fit = new TF1("fit",Polynomial,xmin,xmax,4);
105 if(PolDegree < 0 || PolDegree > 4){
106 //If given a degree out of range, a constant is fitted.
107   for(Int_t i = 1 ; i<5 ; i++){
108     fit->FixParameter(i,0.);
109   }
110 } else{
111   for(Int_t i = PolDegree+1;i<5;i++){
112     fit->FixParameter(i,0);
113   }
114 }
115 fit->FixParameter(5,xo); //Sets the fit around xo
116 Quotient->Fit(fit,"R Q L","",xmin,xmax);
117 fit->Draw("SAME");
118 LowerPad->Update();
119 TPaveStats *psl =
120   (TPaveStats*)Quotient->GetListOfFunctions()->FindObject("stats");
121 psl->SetX1NDC(0.7); psl->SetX2NDC(0.9);
122 psl->SetY1NDC(0.60); psl->SetY2NDC(0.9);
123 psl->SetOptStat(0);
124 psl->SetOptFit(1111);
```

```
125
126   TPaveText* XAxisTitle = new TPaveText();
127   XAxisTitle->SetX1NDC(0.78); XAxisTitle->SetX2NDC(.95);
128   XAxisTitle->SetY1NDC(-0.055); XAxisTitle->SetY2NDC(0.06);
129   XAxisTitle->AddText(Quotient->GetXaxis()->GetTitle());
130   XAxisTitle->SetTextSize(Quotient->GetXaxis()->GetLabelSize()*1.1);
131   XAxisTitle->SetTextFont(Quotient->GetXaxis()->GetLabelFont());
132   XAxisTitle->Draw("SAME");
133   XAxisTitle->SetFillColor(Canvas_->GetFillColor());
134   gPad->Update();
135
136   LowerPad->SetGrid();
137   LowerPad->Modified();
138
139   Canvas_->SetGrid(1,1);
140   Canvas_->Modified();
141   Canvas_->Update();
142
143   /*
144   delete fit;
145   delete psl;
146   delete psu;
147   delete LowerPad;
148   delete UperPad;
149   */
150 }
151
152 void GetEventAndErrors(Double_t &N, Double_t &ErrorN,
153 TH1D* Histogram, Double_t _MaskInf, Double_t _MaskSup) {
154     for(Int_t B = 1;
155     Histogram->GetBinWidth(B)*(B-1)
156     + Histogram->GetBinLowEdge(1) < _MaskSup
157     && B < Histogram->GetNbinsX(); B++ ){//Loop Over bins
158       if(Histogram->GetBinLowEdge(B)<_MaskInf) continue;
159       if(Histogram->GetBinContent(B) == 0) continue;
160       N +=Histogram->GetBinContent(B);
161       ErrorN += pow(Histogram->GetBinError(B),2.);
162     }
163
164   ErrorN = sqrt(ErrorN);
165 }
166
167 void ExpectedEvents(){
168   enum {Data,Kpigg,Kpiggg,K2pi,K3pi0,K2pid,Kpie3,Pi0gg};
169
170   /*Branching Ratios and their Errors*/
171   Double_t BRatio[8];
172   BRatio[Data]=1;
173   BRatio[K2pi] = 20.427e-2;
174   BRatio[K2pid] = .24267e-2;
175   BRatio[K3pi0] = 1.760e-2;
176   BRatio[Kpigg] = 1.01e-6;
177   BRatio[Kpiggg] = 1e-4;
178   BRatio[Pi0gg] = 98.823e-2;
179   BRatio[Kpie3] = 5.07e-2;
180   Double_t ErrorBRatio[8];
181   ErrorBRatio[Data]= 0;
182   BRatio[K2pid] = .007235e-2;
183   ErrorBRatio[K2pi]= 0.07937e-2;
184   ErrorBRatio[K3pi0] = 0.023e-2;
185   ErrorBRatio[Kpiggg] = 0;
186   ErrorBRatio[Kpigg] = 0.06e-6;
187   ErrorBRatio[Pi0gg]= 0.034e-2;
188   ErrorBRatio[Kpie3]=0.04e-2;
189
190   /*Files Paths*/
191   std::string MCDir = "../Results/MC/";
192   std::string ExpDir = "../Results/Exp/";
193   std::string File[7];//5
194
195   File[Data]  = "2017KPINGAMMAV413102025";
196   File[Kpiggg]= "Kpiggg/OVKPIGGG13102025";
197   File[K3pi0] =   "K3pi0/OVK3PI013102025";
```

```
198   File[K2pi]  =      "K2pi/OVK2PI13102025";
199   File[K2pid] =    "K2pi/OVK2PID131022025";
200   File[Kpigg] =    "Kpigg/OVKPIGG13102025";
201   File[Kpie3] =      "Kpie3/OVKE313102025";
202
203   std::string FileName = "ResultsOV";
204   std::string DataGroup = " 2017";
205   std::string MCTipe = " OV";
206   std::string FilterName = " PNN Filter";
207
208   /*Autopass Values*/
209   Double_t AutopassK2pi = 1.;
210   Double_t AutopassK3pi0 = 1.;
211
212   /*Downscale Values*/
213   Double_t FilterDownscale = 1; //Set to one if no downscale
214
215   /*Masked Region*/
216   Double_t MaskInf = 476.;
217   Double_t MaskSup = 510.;
218
219   /*Some useful strings*/
220   std::string StyledModeName[7];
221   StyledModeName[Data] = " Data "+DataGroup;
222   StyledModeName[K2pi]=" K^{+} #rightarrow #pi^{+} #pi^{0}(#gamma) ";
223   StyledModeName[Kpiggg]=" K^{+} #rightarrow #pi^{+} #gamma #gamma #gamma ";
224   StyledModeName[Kpigg]=" K^{+} #rightarrow #pi^{+} #gamma #gamma ";
225   StyledModeName[K3pi0]=" K^{+} #rightarrow #pi^{+} #pi^{0} #pi^{0} ";
226   StyledModeName[K2pid]=
227     " K^{+} #rightarrow #pi^{+} #pi^{0}#rightarrow e^{+}e^{-}#gamma ";
228   StyledModeName[Kpie3]=" K^{+} #rightarrow e^{+} #nu_{e} #pi^{0} ";
229   std::string ModeName[7];
230   ModeName[K2pi]=" K2pi";
231   ModeName[Kpiggg]=" Kpi3g ";
232   ModeName[Kpigg]=" Kpi2g ";
233   ModeName[K3pi0]=" K3pi0 ";
234   ModeName[K2pid]=" K2piD";
235   ModeName[Kpie3]=" Kpie3";
236   ModeName[Data] =" Data ";
237
238   for(Int_t i = 0; i < 7; i++){
239     if(i == Data){ File[i] = ExpDir+File[i]+".root";}
240     else{        File[i] = MCDir+File[i]+".root";}
241   }
242
243   /*Color scheme*/
244   Int_t Color[7];
245   Color[Data]=kGray;
246   Color[Kpiggg]= kOrange-2;
247   Color[Kpigg] = kViolet-5;
248   Color[K2pi] = kRed+1;
249   Color[K3pi0] = kAzure+9;
250   Color[Kpie3]=kGreen+3;
251   Color[K2pid] = kPink+7;
252
253   Int_t MarkerColor[7];
254   MarkerColor[Kpiggg] = kOrange+3;
255   MarkerColor[Kpigg] = kViolet +3;
256   MarkerColor[K2pi] = kRed+2;
257   MarkerColor[K3pi0] = kAzure+3;
258   MarkerColor[Kpie3]=kGreen+3;
259   MarkerColor[K2pid] = kPink+7;
260
261   /*Graphs Section*/
262   TFile* HistCol =
263     new TFile((FileName+".root").c_str(),"RECREATE","ExpectedSignals",0);
264   TCanvas* Canvas= new TCanvas("Canvas","Canvas",1800,1200);
265   std::string PDF = FileName+".pdf";
266   HistCol->cd();
267   Canvas->cd();
268   Canvas->Print((PDF+"[").c_str());
269   TPaveStats *psl;//Pointer to stat box
270   /*Downstream Tracks*/
```

```
271   TH1D* DownstreamTracks[7];
272   for(Int_t i = 0; i< 7; i++){
273     Canvas->Clear();
274     DownstreamTracks[i] =
275         (TH1D*)(TFile::Open(
276         (File[i]).c_str())->Get("KpiNGamma/DownstreamTracks")->
277         Clone(ModeName[i].c_str()));
278     DownstreamTracks[i]->Sumw2();
279     DownstreamTracks[i]->SetLineColor(kBlack);
280     if(i == Data)
281         DownstreamTracks[i]->SetTitle(
282         ("Downstream Tracks " + ModeName[i]+DataGroup+FilterName).c_str());
283     DownstreamTracks[i]->SetTitle(
284         ("Downstream Tracks " + ModeName[i]).c_str());
285     DownstreamTracks[i]->SetFillColor(Color[i]);
286     DownstreamTracks[i]->Draw("HIST E1");
287     gStyle->SetOptStat(1111111);
288     Canvas->Update();
289     gStyle->SetStatX(0.9);
290     gStyle->SetStatY(0.9);
291     Canvas->Modified();
292     // Canvas->Print(PDF.c_str());
293   }
294
295   /*Total Simulated Events*/
296   Double_t TotalEvents[7]={};
297   TH1D *EventsSimulated[7];
298   Double_t MCKaonSampleSize[7]={};
299   Double_t EMCKaonSampleSize[7]={};
300   for(Int_t i = 0; i<7 ; i++){
301     if(i ==Data )continue;
302     EventsSimulated[i] =
303     (TH1D*)(TFile::Open((File[i]).c_str())->
304     Get("KpiNGamma/NDecayedMC")->Clone(ModeName[i].c_str()));
305     EventsSimulated[i]->Sumw2();
306     EventsSimulated[i]->SetTitle(
307         ("Original Events Simulated for "+StyledModeName[i]).c_str());
308     EventsSimulated[i]->SetLineColor(kBlack);
309     EventsSimulated[i]->SetFillColor(Color[i]);
310     EventsSimulated[i]->Draw("HIST E");
311     TotalEvents[i] = (Double_t)EventsSimulated[i]->Integral();
312     gStyle->SetOptStat(1111111);
313     Canvas->Update();
314     gStyle->SetStatX(0.9);
315     gStyle->SetStatY(0.9);
316     Canvas->Modified();
317     //Canvas->Print(PDF.c_str());
318
319     MCKaonSampleSize[i] = TotalEvents[i]/BRatio[i];
320     if(i == K2pi){
321       MCKaonSampleSize[i] /= AutopassK2pi;
322     }
323     if(i == K3pi0){
324       MCKaonSampleSize[i] /= AutopassK3pi0;
325     }
326     EMCKaonSampleSize[i] = MCKaonSampleSize[i]*
327     sqrt(1/(TotalEvents[i])+pow(ErrorBRatio[i]/BRatio[i],2.));
328   }
329
330   /*Two Clusters Kaon Mass Routine*/
331   Double_t MassI,MassF;
332   MassI = 350.;
333   MassF = 650.;
334   /*First all MC modes are scaled with respecto to K2pi*/
335   Double_t ScaleFactorMC2K2pi[7]={};
336   Double_t EScaleFactorMC2K2pi[7]={};
337   for(Int_t i = 0; i<7;i++){
338     ScaleFactorMC2K2pi[i]=1.;
339     if(i == Data || i == K2pi) continue;
340     ScaleFactorMC2K2pi[i] = MCKaonSampleSize[K2pi]/MCKaonSampleSize[i];
341     EScaleFactorMC2K2pi[i] = ScaleFactorMC2K2pi[i] *
342     sqrt(pow(EMCKaonSampleSize[i]/MCKaonSampleSize[i],2.)
343     +pow(EMCKaonSampleSize[K2pi]/MCKaonSampleSize[K2pi],2.));
```

```
344   }
345
346   TH1D *MassKCand2C[7];
347   Int_t NMCmodes2C =7;
348   for(Int_t i = 0 ; i < NMCmodes2C;i++ ){
349     MassKCand2C[i]= (TH1D*)(
350     TFile::Open((File[i]).c_str())->
351     Get("KpiNGamma/2Clusters/MassKaonCand99")->Clone(
352     ("2C"+ModeName[i]).c_str()));
353     MassKCand2C[i]->Sumw2();
354   }
355   HistCol->cd();
356   Canvas->cd();
357
358   TH1D *MassKCand2CScaled[7];
359   Double_t EMassKCand2CScaled[7]={0,0,0,0,0,0,0};
360   Double_t TotalMCEvents2C = 0;
361   Double_t ETotalMCEvents2C = 0 ;
362
363   for(Int_t i = 0; i< NMCmodes2C ; i++){//loop over Histograms
364     if(i == Data) continue;
365     MassKCand2CScaled[i] =
366     (TH1D*)MassKCand2C[i]->Clone(("2CScaled "+ModeName[i]).c_str());
367     // MassKCand2CScaled[i]->Sumw2();
368
369     for(Int_t B = 1; B+1 < MassKCand2C[i]->GetNbinsX(); B++ ){//Loop Over bins
370       if(MassKCand2CScaled[i]->GetBinContent(B) == 0) continue;
371       MassKCand2CScaled[i]->SetBinContent(
372       B,MassKCand2CScaled[i]->GetBinContent(B)*ScaleFactorMC2K2pi[i]);
373       MassKCand2CScaled[i]->SetBinError(B,
374           MassKCand2CScaled[i]->GetBinContent(B)
375                       *sqrt(1./MassKCand2C[i]->GetBinContent(B)
376                       +pow(EScaleFactorMC2K2pi[i]/ScaleFactorMC2K2pi[i],2.)));
377       EMassKCand2CScaled[i] += pow(MassKCand2CScaled[i]->GetBinError(B), 2.);
378     }
379     EMassKCand2CScaled[i] = sqrt(EMassKCand2CScaled[i]);
380     std::cout<<ModeName[i]<<" : "<<EMassKCand2CScaled[i]<<std::endl;
381   }
382
383   for(Int_t i = 0 ; i < NMCmodes2C ; i++){
384     if(i == Data) continue;
385     TotalMCEvents2C += MassKCand2CScaled[i]->Integral();
386     ETotalMCEvents2C += pow(EMassKCand2CScaled[i],2.);
387   }
388   ETotalMCEvents2C = sqrt(ETotalMCEvents2C);
389   std::cout<<TotalMCEvents2C<<" \\pm "<<ETotalMCEvents2C<<std::endl;
390
391   /*Normalizing MC according to Data*/
392   Double_t K2piNormalization = 0 ;
393   Double_t EK2piNormalization = 0;
394   K2piNormalization = ((Double_t) MassKCand2C[Data]->Integral())/TotalMCEvents2C;
395   EK2piNormalization = K2piNormalization *
396   sqrt(pow(ETotalMCEvents2C/TotalMCEvents2C,2.)
397   +1./((Double_t) MassKCand2C[Data]->Integral()));
398   std::cout<<K2piNormalization<<" \\pm "<<EK2piNormalization<<std::endl;
399
400   TH1D* MassKCand2CNormalized[7];
401   for(Int_t i = 0 ; i< NMCmodes2C; i++){
402     if(i == Data) continue;
403     MassKCand2CNormalized[i] = (TH1D*)MassKCand2CScaled[i]->Clone(
404     ("2CNormalized "+ModeName[i]).c_str());
405     for(Int_t B = 1 ; B+1 < MassKCand2CNormalized[i]->GetNbinsX();B++){
406       if(MassKCand2CNormalized[i]->GetBinContent(B) == 0) continue;
407       MassKCand2CNormalized[i]->SetBinError(B,
408       MassKCand2CNormalized[i]->GetBinContent(B)
409       *K2piNormalization*
410       sqrt(pow(EK2piNormalization/K2piNormalization,2.)
411       +pow(MassKCand2CNormalized[i]->GetBinError(B)
412       /MassKCand2CNormalized[i]->GetBinContent(B),2.)));
413       MassKCand2CNormalized[i]->SetBinContent(B,
414       MassKCand2CNormalized[i]->GetBinContent(B)*K2piNormalization);
415       }
416   }
```

```
417
418   /*Printing Routine*/
419   MassI=350.;
420   MassF=600.;
421   for(Int_t i = 0 ; i < NMCmodes2C; i++){//Original MC Events
422     MassKCand2C[i]->GetXaxis()->SetTitle("Mass #pi^{+}#pi^{0} [MeV/c^{2}]");
423     MassKCand2C[i]->GetYaxis()->SetTitle("Events/[1MeV/c^{2}]");
424     MassKCand2C[i]->GetXaxis()->SetRangeUser(MassI,MassF);
425     MassKCand2C[i]->SetMarkerStyle(7);
426     if(i != Data){
427       MassKCand2C[i]->SetTitle(
428       ("K^{+} Mass, after #pi^{0} mass cut,"+StyledModeName[i]
429       +" MC "+MCTipe).c_str());
430       MassKCand2C[i]->SetLineColor(MarkerColor[i]);
431       MassKCand2C[i]->SetMarkerColor(Color[i]-1);
432       MassKCand2C[i]->SetFillColor(Color[i]);
433       MassKCand2C[i]->Draw("HIST E1");
434     }
435     else {
436       //MassKCand2C[i]->SetTitle(
437       ("Reconstructed K^{+} Mass, Events With 2 Neutral Clusters, "
438       +DataGroup+" Data").c_str());
439       MassKCand2C[i]->SetTitle(
440       ("Reconstructed K^{+} Mass, Events With 2 Neutral Clusters"));
441       MassKCand2C[i]->GetXaxis()->SetTitle("Mass #pi^{+}#pi^{0} [MeV/c^{2}] ");
442       MassKCand2C[i]->SetLineColor(kBlack);
443       MassKCand2C[i]->Draw("E1");
444     }
445
446     gStyle->SetOptStat(1111111);
447     Canvas->Update();
448     gStyle->SetStatX(0.9);
449     gStyle->SetStatY(0.9);
450     Canvas->SetLogy();
451     Canvas->Modified();
452     //Canvas->Print(PDF.c_str());
453   }
454
455   for(Int_t i = 0 ; i < NMCmodes2C; i++){//MC Events scaled to Normalization (K2pi)
456     if(i == Data) continue;
457     MassKCand2CScaled[i]->GetXaxis()->SetRangeUser(MassI,MassF);
458     MassKCand2CScaled[i]->GetXaxis()->SetTitle("Mass #pi^{+}#pi^{0} [MeV/c^{2}]");
459     MassKCand2CScaled[i]->GetYaxis()->SetTitle("Events/[1MeV/c^{2}]");
460     MassKCand2CScaled[i]->SetTitle(
461     ("K^{+} Mass, after #pi^{0} mass cut, Scaled to K2#pi,"+StyledModeName[i]+
462     " MC "+MCTipe).c_str());
463     MassKCand2CScaled[i]->SetLineColor(MarkerColor[i]);
464     MassKCand2CScaled[i]->SetFillColor(Color[i]);
465     MassKCand2CScaled[i]->SetMarkerColor(Color[i]-1);
466     MassKCand2CScaled[i]->SetMarkerStyle(7);
467     MassKCand2CScaled[i]->Draw("HIST E1");
468
469     gStyle->SetOptStat(1111111);
470     Canvas->Update();
471     gStyle->SetStatX(0.9);
472     gStyle->SetStatY(0.9);
473     Canvas->SetLogy();
474     Canvas->Modified();
475     Canvas->Print(PDF.c_str());
476   }
477   for(Int_t i = 0 ; i < NMCmodes2C; i++){//MC Events scaled to Data
478     if(i == Data) continue;
479     MassKCand2CNormalized[i]->GetXaxis()->SetTitle(
480     "Mass #pi^{+}#pi^{0} [MeV/c^{2}] ");
481     MassKCand2CNormalized[i]->GetYaxis()->SetTitle("Events/[1MeV/c^{2}]");
482     MassKCand2CNormalized[i]->GetXaxis()->SetRangeUser(MassI,MassF);
483     MassKCand2CNormalized[i]->SetTitle(("K^{+} Mass,
484     after #pi^{0} mass cut, Normalized to Data,"+StyledModeName[i]+
485     " MC "+MCTipe).c_str());
486     MassKCand2CNormalized[i]->SetLineColor(MarkerColor[i]);
487     MassKCand2CNormalized[i]->SetFillColor(Color[i]);
488     MassKCand2CNormalized[i]->SetMarkerColor(Color[i]-1);
489     MassKCand2CNormalized[i]->SetMarkerStyle(7);
```

```
490     MassKCand2CNormalized[i]->Draw("HIST E1");
491
492     gStyle->SetOptStat(1111111);
493     Canvas->Update();
494     gStyle->SetStatX(0.9);
495     gStyle->SetStatY(0.9);
496     Canvas->SetLogy();
497     Canvas->Modified();
498     Canvas->Print(PDF.c_str());
499   }
500
501   TLegend *Legend2C = new TLegend();
502   THStack *Stack2C = new THStack();
503   for(Int_t i= 0; i<Pi0gg; i++){
504     if(i == Data)continue;
505     MassKCand2CNormalized[i]->GetXaxis()->SetRangeUser(MassI,MassF);
506   }
507   for(Int_t i = 0; i<Pi0gg; i++ ){
508     if(i == Kpiggg) continue;
509     if(MassKCand2C[i]->GetEntries() == 0) continue;
510     Legend2C->AddEntry(MassKCand2C[i],StyledModeName[i].c_str(),"lpf");
511     if(i == Data) continue;
512   }
513
514   Stack2C->Add(MassKCand2CNormalized[Kpigg],"HIST E1");
515   Stack2C->Add(MassKCand2CNormalized[K2pid],"HIST E1");
516   Stack2C->Add(MassKCand2CNormalized[K3pi0],"HIST E1");
517   Stack2C->Add(MassKCand2CNormalized[Kpie3],"HIST E1");
518   Stack2C->Add(MassKCand2CNormalized[K2pi],"HIST E1");
519
520   Canvas->Clear();
521   Canvas->SetLogy();
522   MassKCand2C[Data]->SetMinimum(2e-5);
523   MassKCand2C[Data]->GetXaxis()->SetRangeUser(MassI,MassF);
524   MassKCand2C[Data]->Draw();
525   gPad->Update();
526   TPaveStats *statbox1 = (TPaveStats*)MassKCand2C[Data]->FindObject("stats");
527   statbox1->SetOptStat(110);
528
529   Stack2C->Draw("SAME");
530   // Stack2C->GetXaxis()->SetRangeUser(MassI,MassF);
531   MassKCand2C[Data]->Draw("SAME E1");
532   Legend2C->SetX1(0.15);
533   Legend2C->SetX2(0.35);
534   Legend2C->SetY1(.6);
535   Legend2C->SetY2(.9);
536   Legend2C->Draw("SAME");
537
538   Stack2C->GetXaxis()->SetRangeUser(MassI,MassF);
539   Canvas->Modified();
540   Canvas->Update();
541   Canvas->Print(PDF.c_str());
542   /*
543   Canvas->Clear();
544   TH1D* S = (TH1D*)Stack2C->GetStack()->Last()->Clone();
545   S->Divide(MassKCand2C[Data]);
546   S->Draw();
547   std::cout<<std::endl<<std::endl<<"AQUI"<<std::endl<<std::endl;
548   Canvas->Modified();
549   Canvas->Update();
550   Canvas->Print(PDF.c_str());
551   */
552   Similarity(Canvas,Stack2C,MassKCand2C[Data],Legend2C,480.,500.,493.667,0);
553   Canvas->Print(PDF.c_str());
554
555   Double_t AcceptanceK2pi=
556   ((Double_t)MassKCand2C[K2pi]->Integral())/TotalEvents[K2pi]*AutopassK2pi;
557   Double_t EAcceptanceK2pi=
558   AcceptanceK2pi*sqrt(1/(Double_t)MassKCand2C[K2pi]->Integral()+
559   1./TotalEvents[K2pi]);
560   std::cout<<"Acceptance K2pi: "<<AcceptanceK2pi<<
561   " \\pm "<<EAcceptanceK2pi<<std::endl;
562
```

```
563   /*Number of Kaons using K2pi as normalization*/
564   Double_t NKaonsK2pi = 0;
565   Double_t ENKaonsK2pi = 0;
566   Double_t K2piFrac = MassKCand2C[K2pi]->Integral()/TotalMCEvents2C;
567   Double_t EK2piFrac = K2piFrac *
568   sqrt(pow(ETotalMCEvents2C/TotalMCEvents2C,2.)
569   +1./MassKCand2C[K2pi]->Integral());
570   Double_t NumberK2pi = MassKCand2C[Data]->Integral()*K2piFrac;
571   Double_t ENumberK2pi =
572   NumberK2pi*sqrt(1./MassKCand2C[Data]->Integral()+pow(EK2piFrac/K2piFrac,2.));
573   NKaonsK2pi =
574   MassKCand2C[Data]->Integral()*K2piFrac
575   /(BRatio[K2pi]*AcceptanceK2pi*FilterDownscale);
576   ENKaonsK2pi = NKaonsK2pi * sqrt(1./MassKCand2C[Data]->Integral()+
577           pow(EK2piFrac/K2piFrac,2.)+
578           pow(ErrorBRatio[K2pi]/BRatio[K2pi],2.)+
579           pow(EAcceptanceK2pi/AcceptanceK2pi,2.));
580
581   /*4Clusters section*/
582   /*All MC files are scaled now respecto to K3PI0*/
583   Double_t ScaleFactorMC2K3pi0[7]={};
584   Double_t EScaleFactorMC2K3pi0[7]={};
585   Int_t NMCmodes4C = 7;
586   for(Int_t i = 0; i<NMCmodes4C;i++){
587     ScaleFactorMC2K3pi0[i]=1.;
588     if(i == Data || i == K3pi0) continue;
589     ScaleFactorMC2K3pi0[i] =
590     MCKaonSampleSize[K3pi0]/MCKaonSampleSize[i];
591     EScaleFactorMC2K3pi0[i] =
592     ScaleFactorMC2K3pi0[i] *
593     sqrt(pow(EMCKaonSampleSize[i]/MCKaonSampleSize[i],2.)
594     +pow(EMCKaonSampleSize[K3pi0]/MCKaonSampleSize[K3pi0],2.));
595     // std::cout<<ModeName[i]+
596     " K3pi0 as normalization"<<ScaleFactorMC2K3pi0[i]<<std::endl;
597   }
598
599   /*Getting Histograms from MC and data files*/
600   TH1D *MassKCand4C[7];
601   for(Int_t i = 0 ; i < NMCmodes2C;i++ ){
602     MassKCand4C[i]=
603     (TH1D*)(TFile::Open((File[i]).c_str())->
604     Get("KpiNGamma/4Clusters/MassKaonCand99")->Clone(("4C"+ModeName[i]).c_str()));
605     MassKCand4C[i]->Sumw2();
606   }
607   HistCol->cd();
608   Canvas->cd();
609
610   TH1D *MassKCand4CScaled[7];
611   Double_t EMassKCand4CScaled[7]={};
612   Double_t TotalMCEvents4C = 0;
613   Double_t ETotalMCEvents4C = 0 ;
614
615   /*Scaling histograms according to K3pi0 as normalization channel*/
616   for(Int_t i = 0; i< NMCmodes4C ; i++){//loop over Histograms
617     if(i == Data) continue;
618     MassKCand4CScaled[i] =
619     (TH1D*)MassKCand4C[i]->Clone(("4CScaled "+ModeName[i]).c_str());
620     MassKCand4CScaled[i]->Sumw2();
621
622     for(Int_t B = 1; B+1 < MassKCand4C[i]->GetNbinsX(); B++ ){
623     //Loop Over bins
624       if(MassKCand4CScaled[i]->GetBinContent(B) == 0) continue;
625       MassKCand4CScaled[i]->SetBinContent(
626       B,MassKCand4CScaled[i]->GetBinContent(B)*ScaleFactorMC2K3pi0[i]);
627       MassKCand4CScaled[i]->SetBinError(B,
628      MassKCand4CScaled[i]->GetBinContent(B)*
629        sqrt(1./MassKCand4C[i]->GetBinContent(B)
630       +pow(EScaleFactorMC2K3pi0[i]/ScaleFactorMC2K3pi0[i],2.)));
631       EMassKCand4CScaled[i] += pow(MassKCand4CScaled[i]->GetBinError(B), 2.);
632     }
633     EMassKCand4CScaled[i] = sqrt(EMassKCand4CScaled[i]);
634   }
635
```

```
636   for(Int_t i = 0 ; i < NMCmodes4C ; i++){
637     if(i == Data) continue;
638     TotalMCEvents4C += MassKCand4CScaled[i]->Integral();
639     ETotalMCEvents4C += pow(EMassKCand4CScaled[i],2.);
640   }
641   ETotalMCEvents4C = sqrt(ETotalMCEvents4C);
642
643   /*Normalizing MC with Data*/
644   Double_t K3pi0Normalization = 0 ;
645   Double_t EK3pi0Normalization = 0;
646   K3pi0Normalization =
647   ((Double_t) MassKCand4C[Data]->Integral())/TotalMCEvents4C;
648   EK3pi0Normalization =
649   K3pi0Normalization * sqrt(pow(ETotalMCEvents4C/TotalMCEvents4C,2.)
650   +1./((Double_t) MassKCand4C[Data]->Integral()));
651   TH1D* MassKCand4CNormalized[7];
652   for(Int_t i = 0 ; i< NMCmodes4C; i++){
653     if(i == Data) continue;
654     MassKCand4CNormalized[i] =
655     (TH1D*)MassKCand4CScaled[i]->Clone(("4CNormalized "+ModeName[i]).c_str());
656     for(Int_t B = 1 ; B+1 < MassKCand4CNormalized[i]->GetNbinsX();B++){
657       if(MassKCand4CNormalized[i]->GetBinContent(B) == 0) continue;
658       MassKCand4CNormalized[i]->SetBinError(B,
659       MassKCand4CNormalized[i]->GetBinContent(B)*K3pi0Normalization*
660   sqrt(pow(EK3pi0Normalization/K3pi0Normalization,2.)
661     +pow(MassKCand4CNormalized[i]->GetBinError(B)
662     /MassKCand4CNormalized[i]->GetBinContent(B),2.)));
663       MassKCand4CNormalized[i]->SetBinContent(B,
664       MassKCand4CNormalized[i]->GetBinContent(B)*K3pi0Normalization);
665       }
666   }
667
668   for(Int_t i = 0 ; i < NMCmodes4C; i++){//Original MC Events
669     MassKCand4C[i]->GetXaxis()->SetTitle(
670     "Mass #pi^{+}#pi^{0}#pi^{0} [MeV/c^{2}]");
671     MassKCand4C[i]->GetYaxis()->SetTitle("Events/[1MeV/c^{2}]");
672     MassKCand4C[i]->GetXaxis()->SetRangeUser(MassI,MassF);
673     MassKCand4C[i]->SetMarkerStyle(7);
674     if(i != Data){
675       MassKCand4C[i]->SetTitle(
676       ("K^{+} Mass, after #pi^{0} mass cut,"+StyledModeName[i]
677       +" MC "+MCTipe).c_str());
678       MassKCand4C[i]->SetLineColor(MarkerColor[i]);
679       MassKCand4C[i]->SetMarkerColor(Color[i]-1);
680       MassKCand4C[i]->SetFillColor(Color[i]);
681       MassKCand4C[i]->Draw("HIST E1");
682     }
683     else {//"Reconstructed K^{+} Mass, Events With 2 Neutral Clusters"
684       //MassKCand4C[i]->SetTitle(
685       ("K^{+} Mass"+DataGroup+" Data").c_str());
686       MassKCand4C[i]->SetTitle(
687       ("Reconstructed K^{+} Mass, Events With 4 Neutral Clusters"));
688       MassKCand4C[i]->SetLineColor(kBlack);
689       MassKCand4C[i]->Draw("E1");
690     }
691
692     gStyle->SetOptStat(1111111);
693     Canvas->Update();
694     gStyle->SetStatX(0.9);
695     gStyle->SetStatY(0.9);
696     Canvas->SetLogy();
697     Canvas->Modified();
698     //Canvas->Print(PDF.c_str());
699   }
700
701   for(Int_t i = 0 ; i < NMCmodes4C; i++){
702   //MC Events scaled to Normalization (K3pi0)
703     if(i == Data) continue;
704     MassKCand4CScaled[i]->GetXaxis()->SetTitle(
705     "Mass #pi^{+} #pi^{0} #pi^{0} [MeV/c^{2}]");
706     MassKCand4CScaled[i]->GetYaxis()->SetTitle("Events/[1MeV/c^{2}]");
707     MassKCand4CScaled[i]->GetXaxis()->SetRangeUser(MassI,MassF);
708     MassKCand4CScaled[i]->SetTitle(
```

```
709       ("K^{+} Mass, after #pi^{0} mass cut, Scaled to K3#pi^{0},"
710       +StyledModeName[i]+" MC "+MCTipe).c_str());
711       MassKCand4CScaled[i]->SetLineColor(MarkerColor[i]);
712       MassKCand4CScaled[i]->SetFillColor(Color[i]);
713       MassKCand4CScaled[i]->SetMarkerColor(Color[i]-1);
714       MassKCand4CScaled[i]->SetMarkerStyle(7);
715       MassKCand4CScaled[i]->Draw("HIST E1");
716
717       gStyle->SetOptStat(1111111);
718       Canvas->Update();
719       gStyle->SetStatX(0.9);
720       gStyle->SetStatY(0.9);
721       Canvas->SetLogy();
722       Canvas->Modified();
723       Canvas->Print(PDF.c_str());
724     }
725
726     for(Int_t i = 0 ; i < NMCmodes4C; i++){
727     //MC Events scaled to Data
728       if(i == Data) continue;
729       MassKCand4CNormalized[i]->GetXaxis()->SetTitle(
730       "Mass #pi^{+}#pi^{0}#pi^{0} [MeV/c^{2}]");
731       MassKCand4CNormalized[i]->GetYaxis()->SetTitle(
732       "Events/[1MeV/c^{2}]");
733       MassKCand4CNormalized[i]->GetXaxis()->SetRangeUser(MassI,MassF);
734       MassKCand4CNormalized[i]->SetTitle(
735       ("K^{+} Candidate Mass, after #pi^{0} mass cut, Normalized to Data,"
736       +StyledModeName[i]+" MC "+MCTipe).c_str());
737       MassKCand4CNormalized[i]->SetLineColor(MarkerColor[i]);
738       MassKCand4CNormalized[i]->SetFillColor(Color[i]);
739       MassKCand4CNormalized[i]->SetMarkerColor(Color[i]-1);
740       MassKCand4CNormalized[i]->SetMarkerStyle(7);
741       MassKCand4CNormalized[i]->Draw("HIST E1");
742
743       gStyle->SetOptStat(1111111);
744       Canvas->Update();
745       gStyle->SetStatX(0.9);
746       gStyle->SetStatY(0.9);
747       Canvas->SetLogy();
748       Canvas->Modified();
749       Canvas->Print(PDF.c_str());
750     }
751
752     TLegend *Legend4C = new TLegend();
753     THStack *Stack4C = new THStack();
754     MassI = 400.;
755     MassF = 600.;
756     for(Int_t i = 0; i<Pi0gg; i++ ){
757       if(i == Kpiggg) continue;
758       if(MassKCand4C[i]->GetEntries() == 0) continue;
759       Legend4C->AddEntry(MassKCand4C[i],StyledModeName[i].c_str(),"lpf");
760       if(i == Data) continue;
761     }
762
763     Stack4C->Add(MassKCand4CNormalized[K2pid],"HIST E1");
764     Stack4C->Add(MassKCand4CNormalized[K2pi],"HIST E1");
765     Stack4C->Add(MassKCand4CNormalized[K3pi0],"HIST E1");
766
767     Canvas->Clear();
768     MassKCand4C[Data]->SetMinimum(3e-4);
769     MassKCand4C[Data]->Draw();
770     gPad->Update();
771     TPaveStats *statbox2 =
772     (TPaveStats*)MassKCand4C[Data]->FindObject("stats");
773     statbox2->SetOptStat(110);
774     //gStyle->SetOptStat(1111111);
775     Stack4C->Draw("SAME");
776     MassKCand4C[Data]->Draw("SAME E1");
777     Legend4C->Draw("SAME");
778     //gStyle->SetStatX(0.9);
779     //gStyle->SetStatY(0.9);
780     Canvas->SetLogy();
781     for(Int_t i= 0; i<Pi0gg; i++){
```

```
782       MassKCand4C[i]->GetXaxis()->SetRangeUser(MassI,MassF);
783       if(i == Data)continue;
784       MassKCand4CNormalized[i]->GetXaxis()->SetRangeUser(MassI,MassF);
785     }
786     Stack4C->GetXaxis()->SetRangeUser(MassI,MassF);
787     Legend4C->SetX1NDC(0.15);
788     Legend4C->SetX2NDC(0.35);
789     Legend4C->SetY1NDC(.7);
790     Legend4C->SetY2NDC(.9);
791     //gStyle->SetOptStat("mer");
792     Canvas->Modified();
793     Canvas->Update();
794     Canvas->Print(PDF.c_str());
795
796     Similarity(Canvas,Stack4C,MassKCand4C[Data],Legend4C,485.,500.,493.667,0);
797     Canvas->Print(PDF.c_str());
798
799     /*Acceptance for k3pi0 computation*/
800     Double_t AcceptanceK3pi0 = (
801     (Double_t)MassKCand4C[K3pi0]->Integral())
802     /TotalEvents[K3pi0]*AutopassK3pi0;
803     Double_t EAcceptanceK3pi0=
804     AcceptanceK3pi0*sqrt(1/(Double_t)MassKCand4C[K3pi0]->Integral()
805     +1./TotalEvents[K3pi0]);
806     // std::cout<<"Acceptance K3pi0: "<<
807     AcceptanceK3pi0<<" \\pm "<<EAcceptanceK3pi0<<std::endl;
808
809     /*Number of Kaons using K3pi0 as normalization*/
810     Double_t NKaonsK3pi0 = 0;
811     Double_t ENKaonsK3pi0 = 0;
812     Double_t K3pi0frac = MassKCand4C[K3pi0]->Integral()/TotalMCEvents4C;
813     Double_t EK3pi0frac =
814     K3pi0frac * sqrt(pow(ETotalMCEvents4C/TotalMCEvents4C,2.)
815     +1./MassKCand4C[K3pi0]->Integral());
816     Double_t NumberK3pi0 =
817     MassKCand4C[Data]->Integral()*K3pi0frac;
818     Double_t ENumberK3pi0 =
819     NumberK3pi0 * sqrt(1./MassKCand4C[Data]->Integral()
820     + pow(EK3pi0frac/K3pi0frac,2.));
821     NKaonsK3pi0 =
822     MassKCand4C[Data]->Integral()*K3pi0frac
823     /(BRatio[K3pi0]*AcceptanceK3pi0*FilterDownscale*pow(BRatio[Pi0gg],2));
824     ENKaonsK3pi0 = NKaonsK3pi0 * sqrt(1./MassKCand4C[Data]->Integral()+
825               pow(EK3pi0frac/K3pi0frac,2.)+
826               pow(ErrorBRatio[K3pi0]/BRatio[K3pi0],2.)+
827               pow(EAcceptanceK3pi0/AcceptanceK3pi0,2.)
828                                         );
829
830     /*Ratio between Number of Kaons with different Normalization Channels*/
831     Double_t RatioNKaons = 1.;
832     double_t ERatioNKaons = 0.;
833     RatioNKaons = NKaonsK2pi/NKaonsK3pi0;
834     ERatioNKaons = RatioNKaons*
835     sqrt(pow(ENKaonsK3pi0/NKaonsK3pi0,2.)+pow(ENKaonsK2pi/NKaonsK2pi,2.));
836
837     /*3 Clusters section*/
838     TH1D *MassKCand3C[7];
839     Int_t NMCmodes3C =7;
840     for(Int_t i = 0 ; i < NMCmodes2C;i++ ){
841       MassKCand3C[i]= (TH1D*)
842       (TFile::Open((File[i]).c_str())->
843       Get("KpiNGamma/3Clusters/MassKaonCand99")->
844       Clone(("3C"+ModeName[i]).c_str()));
845       MassKCand3C[i]->Sumw2();
846     }
847
848     HistCol->cd();
849     Canvas->cd();
850
851     TH1D *MassKCand3CScaled[7];
852     Double_t EMassKCand3CScaled[7]={};
853     Double_t TotalMCEvents3C = 0;
854     Double_t ETotalMCEvents3C = 0 ;
```

```
855
856   /*Scaling histograms according to K2pi0 as normalization channel*/
857   for(Int_t i = 0; i< NMCmodes3C ; i++){//loop over Histograms
858     if(i == Data) continue;
859     MassKCand3CScaled[i] =
860     (TH1D*)MassKCand3C[i]->Clone(("3CScaled "+ModeName[i]).c_str());
861     //MassKCand3CScaled[i]->Sumw2();
862     std::cout<<ModeName[i]<<" : "<<ScaleFactorMC2K2pi[i]
863     <<" \\pm "<<EScaleFactorMC2K2pi[i]<<std::endl;
864     for(Int_t B = 1; B+1 < MassKCand3C[i]->GetNbinsX(); B++ ){
865     //Loop Over bins
866       if(MassKCand3CScaled[i]->GetBinContent(B) == 0) continue;
867       if(i != K3pi0){
868   MassKCand3CScaled[i]->SetBinError(
869     B,MassKCand3CScaled[i]->GetBinContent(B)
870     *ScaleFactorMC2K2pi[i]*sqrt(1./MassKCand3C[i]->GetBinContent(B)
871     +pow(EScaleFactorMC2K2pi[i]/ScaleFactorMC2K2pi[i],2.)));
872   MassKCand3CScaled[i]->SetBinContent(
873     B,MassKCand3CScaled[i]->GetBinContent(B)*ScaleFactorMC2K2pi[i]);
874       }
875       EMassKCand3CScaled[i] +=
876       pow(MassKCand3CScaled[i]->GetBinError(B), 2.);
877     }
878     EMassKCand3CScaled[i] = sqrt(EMassKCand3CScaled[i]);
879     std::cout<<ModeName[i]
880     <<" : "<<MassKCand3CScaled[i]->Integral()
881     <<" \\pm "<<EMassKCand3CScaled[i]<<std::endl;
882   }
883
884   /*Normalization with respec to Data*/
885   std::cout<<"bin errors: "<<std::endl;
886   TH1D* MassKCand3CNormalized[7];
887   for(Int_t i = 0 ; i< NMCmodes3C; i++){
888     if(i == Data) continue;
889     MassKCand3CNormalized[i] =
890     (TH1D*)MassKCand3CScaled[i]->Clone(
891     ("3CNormalized "+ModeName[i]).c_str());
892     for(Int_t B = 1 ;
893     B+1 < MassKCand3CNormalized[i]->GetNbinsX();B++){
894       if(MassKCand3CNormalized[i]->GetBinContent(B) == 0) continue;
895       if(i != K3pi0){
896   MassKCand3CNormalized[i]->SetBinError(
897     B, MassKCand3CNormalized[i]->GetBinContent(B)*K2piNormalization
898     *sqrt(pow(EK2piNormalization/K2piNormalization,2.)
899     +pow(MassKCand3CNormalized[i]->GetBinError(B)
900     /MassKCand3CNormalized[i]->GetBinContent(B),2.)));
901   MassKCand3CNormalized[i]->SetBinContent(
902     B,MassKCand3CNormalized[i]->GetBinContent(B)*K2piNormalization);
903       }
904       else{
905   MassKCand3CNormalized[i]->SetBinError(
906     B, MassKCand3CNormalized[i]->GetBinContent(B)*K3pi0Normalization*
907     sqrt(pow(EK3pi0Normalization/K3pi0Normalization,2.)
908     +pow(MassKCand3CNormalized[i]->GetBinError(B)
909     /MassKCand3CNormalized[i]->GetBinContent(B),2.)));
910   MassKCand3CNormalized[i]->SetBinContent(
911     B,MassKCand3CNormalized[i]->GetBinContent(B)*K3pi0Normalization);
912       }
913       if(i == K3pi0)
914         std::cout<<MassKCand3CNormalized[i]->GetBinError(B)<<std::endl;
915     }
916   }
917
918   /*Acceptance of Kpiggg*/
919   Double_t AcceptanceKpiggg ;
920   AcceptanceKpiggg = MassKCand3C[Kpiggg]->Integral()/TotalEvents[Kpiggg];
921   Double_t EAcceptanceKpiggg;
922   EAcceptanceKpiggg = AcceptanceKpiggg*
923   sqrt(1./MassKCand3C[Kpiggg]->Integral()+1./TotalEvents[Kpiggg]);
924
925   /*Printing Routine*/
926   MassI = 320.;
927   MassF = 750.;
```

```
928   for(Int_t i = 0 ; i < NMCmodes3C; i++){//Original MC Events
929     MassKCand3C[i]->GetXaxis()->SetTitle(
930     "Mass #pi^{+} #gamma#gamma#gamma [MeV/c^{2}]");
931     MassKCand3C[i]->GetYaxis()->SetTitle("Events/[1MeV/c^{2}]");
932     MassKCand3C[i]->GetXaxis()->SetRangeUser(MassI,MassF);
933     MassKCand3C[i]->SetMarkerStyle(7);
934     if(i != Data){
935       MassKCand3C[i]->SetTitle(
936       ("K^{+} Mass, after #pi^{0} mass cut,"
937       +StyledModeName[i]+" MC "+MCTipe).c_str());
938       MassKCand3C[i]->SetLineColor(MarkerColor[i]);
939       MassKCand3C[i]->SetMarkerColor(Color[i]-1);
940       MassKCand3C[i]->SetFillColor(Color[i]);
941       MassKCand3C[i]->Draw("HIST E1");
942     }
943     else {
944       //"Reconstructed K^{+} Mass, Events With 4 Neutral Clusters"
945       //MassKCand3C[i]->SetTitle(("K^{+} Mass"+DataGroup+" Data").c_str());
946       MassKCand3C[i]->SetTitle(
947       "Reconstructed K^{+} Mass, Events With 3 Neutral Clusters Final Selection");
948       MassKCand3C[i]->SetLineColor(kBlack);
949       MassKCand3C[i]->Draw("E1");
950     }
951
952     gStyle->SetOptStat(1111111);
953     Canvas->Update();
954     gStyle->SetStatX(0.9);
955     gStyle->SetStatY(0.9);
956     Canvas->SetLogy();
957     Canvas->Modified();
958     //Canvas->Print(PDF.c_str());
959   }
960
961   for(Int_t i = 0 ; i < NMCmodes3C; i++){
962   //MC Events scaled to Normalization (K2pi)
963     if(i == Data) continue;
964     MassKCand3CScaled[i]->GetXaxis()->
965     SetTitle("Mass #pi^{+} #gamma#gamma#gamma [MeV/c^{2}]");
966     MassKCand3CScaled[i]->GetYaxis()->
967     SetTitle("Events/[1MeV/c^{2}]");
968     MassKCand3CScaled[i]->GetXaxis()->
969     SetRangeUser(MassI,MassF);
970     MassKCand3CScaled[i]->
971     SetTitle(("K^{+} Mass, after #pi^{0} mass cut, Scaled to K2#pi,"
972     +StyledModeName[i]+" MC "+MCTipe).c_str());
973     if(i == K3pi0) MassKCand3CScaled[i]->
974     SetTitle(("K^{+} Mass, after #pi^{0} mass cut, Scaled to K3#pi^{0},"
975     +StyledModeName[i]+" MC "+MCTipe).c_str());
976     MassKCand3CScaled[i]->SetLineColor(MarkerColor[i]);
977     MassKCand3CScaled[i]->SetFillColor(Color[i]);
978     MassKCand3CScaled[i]->SetMarkerColor(Color[i]-1);
979     MassKCand3CScaled[i]->SetMarkerStyle(7);
980     MassKCand3CScaled[i]->Draw("HIST E1");
981
982     gStyle->SetOptStat(1111111);
983     Canvas->Update();
984     gStyle->SetStatX(0.9);
985     gStyle->SetStatY(0.9);
986     Canvas->SetLogy();
987     Canvas->Modified();
988     Canvas->Print(PDF.c_str());
989   }
990
991
992   for(Int_t i = 0 ; i < NMCmodes3C; i++){//MC Events scaled to Data
993     if(i == Data) continue;
994     MassKCand3CNormalized[i]->GetXaxis()->
995     SetTitle("Mass #pi^{+} #gamma#gamma#gamma [MeV/c^{2}]");
996     MassKCand3CNormalized[i]->GetYaxis()->
997     SetTitle("Events/[1MeV/c^{2}]");
998     MassKCand3CNormalized[i]->GetXaxis()->
999     SetRangeUser(MassI,MassF);
1000    MassKCand3CNormalized[i]->SetTitle(
```

```
1001      ("K^{+} Mass, after #pi^{0} mass cut, Normalized to Data,"
1002      +StyledModeName[i]+" MC "+MCTipe).c_str());
1003      MassKCand3CNormalized[i]->SetLineColor(MarkerColor[i]);
1004      MassKCand3CNormalized[i]->SetFillColor(Color[i]);
1005      MassKCand3CNormalized[i]->SetMarkerColor(Color[i]-1);
1006      MassKCand3CNormalized[i]->SetMarkerStyle(7);
1007      MassKCand3CNormalized[i]->Draw("HIST E1");
1008
1009      gStyle->SetOptStat(1111111);
1010      Canvas->Update();
1011      gStyle->SetStatX(0.9);
1012      gStyle->SetStatY(0.9);
1013      Canvas->SetLogy();
1014      Canvas->Modified();
1015      Canvas->Print(PDF.c_str());
1016    }
1017
1018    TLegend *Legend3C = new TLegend();
1019    THStack *Stack3C = new THStack();
1020    for(Int_t i = 0; i<Pi0gg; i++ ){
1021      if(MassKCand3C[i]->GetEntries() == 0) continue;
1022      //if(i == Kpiggg)continue;
1023      if(i != Kpiggg) Legend3C->
1024      AddEntry(MassKCand3C[i],StyledModeName[i].c_str(),"lpf");
1025      else Legend3C->AddEntry(MassKCand3C[i],
1026      (StyledModeName[i]+",BR = 10^{-4}").c_str(),"lpf");
1027    }
1028    Stack3C->Add(MassKCand3CNormalized[Kpigg],"HIST E1");
1029    Stack3C->Add(MassKCand3CNormalized[K2pid],"HIST E1");
1030    Stack3C->Add(MassKCand3CNormalized[K2pi],"HIST E1");
1031    Stack3C->Add(MassKCand3CNormalized[K3pi0],"HIST E1");
1032    Stack3C->Add(MassKCand3CNormalized[Kpiggg],"HIST E1");
1033
1034    Canvas->Clear();
1035    Canvas->SetLogy(1);
1036    Stack3C->Draw();
1037    MassKCand3C[Data]->SetMinimum(1e-4);
1038    MassKCand3C[Data]->SetMaximum(
1039    Stack3C->GetHistogram()->GetMaximum()*100);
1040    Canvas->Clear();
1041    MassKCand3C[Data]->Draw();
1042    gStyle->SetOptStat("e");
1043    Stack3C->Draw("SAME");
1044    MassKCand3C[Data]->Draw("SAME E1");
1045    Legend3C->SetX1(0.13); Legend3C->SetX2(0.35);
1046    Legend3C->SetY1(.69); Legend3C->SetY2(.89);
1047    Legend3C->Draw("SAME");
1048    psl =
1049    (TPaveStats*)MassKCand3C[Data]->
1050    GetListOfFunctions()->FindObject("stats");
1051    psl->SetX1NDC(0.70); psl->SetX2NDC(0.90);
1052    psl->SetY1NDC(0.80); psl->SetY2NDC(0.90);
1053    for(Int_t i= 0; i<Pi0gg; i++){
1054      MassKCand3C[i]->GetXaxis()->SetRangeUser(MassI,MassF);
1055      if(i == Data)continue;
1056      MassKCand3CScaled[i]->GetXaxis()->SetRangeUser(MassI,MassF);
1057    }
1058    Stack3C->GetXaxis()->SetRangeUser(MassI,MassF);
1059    TBox* MaskedBox =
1060    new TBox(MaskInf,0.,MaskSup,Stack3C->
1061    GetHistogram()->GetMaximum()*100);
1062    MaskedBox->SetFillColorAlpha(kGray,0.25);
1063    TLine* LineInf =
1064    new TLine(MaskInf,0.,MaskInf,Stack3C->GetHistogram()
1065    ->GetMaximum()*100);
1066    TLine* LineSup =
1067    new TLine(MaskSup,0.,MaskSup,Stack3C->GetHistogram()
1068    ->GetMaximum()*100);
1069    LineInf->SetLineColor(kRed);
1070    LineInf->SetLineStyle(3);
1071    LineInf->SetLineWidth(1);
1072    LineSup->SetLineColor(kRed);
1073    LineSup->SetLineStyle(3);
```

```
1074    LineSup->SetLineWidth(1);
1075    TText* ControlRegion1 =
1076    new TText(350.,Stack3C->GetMaximum()/20,"Control Region I");
1077    ControlRegion1->SetTextSize(0.04);
1078    TText* ControlRegion2 =
1079    new TText(620.,Stack3C->GetMaximum()/20,"Control Region II");
1080    ControlRegion2->SetTextSize(0.04);
1081    TText* MaskedRegion =
1082    new TText(470.,Stack3C->GetMaximum()*10,"Masked");
1083    MaskedRegion->SetTextSize(0.04);
1084    MaskedRegion->SetTextColor(kRed);
1085    ControlRegion1->SetTextColor(kBlue);
1086    ControlRegion2->SetTextColor(kBlue);
1087    //
1088    MaskedRegion->Draw("SAME");
1089    ControlRegion2->Draw("SAME");
1090    ControlRegion1->Draw("SAME");
1091    LineInf->Draw("SAME");
1092    LineSup->Draw("SAME");
1093    // MaskedBox->Draw("SAME");
1094
1095    Canvas->Modified();
1096    Canvas->Update();
1097    Canvas->Print(PDF.c_str());
1098
1099    Canvas->Clear();
1100    Canvas->SetLogy(0);
1101    MassKCand3C[Data]->SetMinimum(0);
1102    MassKCand3C[Data]->
1103    SetMaximum(Stack3C->GetHistogram()->GetMaximum());
1104    //MassKCand3C[Data]->
1105    SetMaximum(MassKCand3C[Data]->Integral()*2);
1106    MassKCand3C[Data]->Draw();
1107    gStyle->SetOptStat("e");
1108    Stack3C->Draw("SAME");
1109    MassKCand3C[Data]->Draw("SAME E1");
1110    Legend3C->Draw("SAME");
1111    LineInf->SetVertical(1);
1112    LineSup->SetVertical(1);
1113    LineInf->SetY2(Stack3C->
1114    GetHistogram()->GetMaximum());
1115    LineSup->SetY2(Stack3C->
1116    GetHistogram()->GetMaximum());
1117    MaskedBox->SetY2(Stack3C->
1118    GetHistogram()->GetMaximum());
1119    LineInf->Draw("SAME");
1120    LineSup->Draw("SAME");
1121    // MaskedBox->Draw("SAME");
1122    ControlRegion1->SetY(Stack3C->GetMaximum()/2);
1123    ControlRegion2->SetY(Stack3C->GetMaximum()/2);
1124    MaskedRegion->SetY(Stack3C->GetMaximum()*1.25);
1125    ControlRegion1->Draw("SAME");
1126    ControlRegion2->Draw("SAME");
1127    MaskedRegion->Draw("SAME");
1128    Canvas->Modified(); Canvas->Update();
1129    Canvas->Print(PDF.c_str());
1130
1131     /*Total Background and signal calculation*/
1132    Double_t TotalBackground = 0;
1133    Double_t ETotalBackground = 0;
1134
1135    for(Int_t i = 0 ; i < NMCmodes3C ; i++){
1136      if(Data == i || Kpiggg == i) continue;
1137      //TotalBackground += MassKCand3CNormalized[i]->Integral();
1138      for(Int_t B = 1;
1139      MassKCand3CNormalized[i]->GetBinWidth(B)*(B-1)
1140      + MassKCand3CNormalized[i]->GetBinLowEdge(1) < MaskSup
1141      && B<MassKCand3CNormalized[i]->
1142      GetNbinsX(); B++ ){//Loop Over bins
1143        if(MassKCand3CNormalized[i]->
1144        GetBinLowEdge(B)<MaskInf) continue;
1145        if(MassKCand3CNormalized[i]->
1146        GetBinContent(B) == 0) continue;
```

```
1147        TotalBackground +=
1148        MassKCand3CNormalized[i]->GetBinContent(B);
1149        ETotalBackground +=
1150        pow(MassKCand3CNormalized[i]->GetBinError(B),2.);
1151      }
1152    }
1153    ETotalBackground = sqrt(ETotalBackground);
1154
1155    Double_t Masked[NMCmodes3C];
1156    Double_t MaskedErrors[NMCmodes3C];
1157    for(Int_t i = 0; i< NMCmodes3C;i++){
1158      Masked[i] = 0; MaskedErrors[i]=0;
1159      if(Data == i || Kpiggg == i) continue;
1160      if(MassKCand3CNormalized[i]->Integral()==0)continue;
1161      GetEventAndErrors(Masked[i],
1162      MaskedErrors[i],MassKCand3CNormalized[i],MaskInf,MaskSup);
1163    }
1164
1165    //Control Region I
1166    //Control Region II:
1167    Double_t ControlII[NMCmodes3C];
1168    Double_t ControlIIErrors[NMCmodes3C];
1169    for(Int_t i = 0; i< NMCmodes3C;i++){
1170      ControlII[i] = 0; ControlIIErrors[i]=0;
1171      if(Data == i || Kpiggg == i) continue;
1172      if(MassKCand3CNormalized[i]->Integral()==0)continue;
1173      GetEventAndErrors(ControlII[i],ControlIIErrors[i],
1174      MassKCand3CNormalized[i],MaskSup,751);
1175    }
1176    Double_t TotalControlII=0.;
1177    Double_t TotalControlIIError=0.;
1178    for(Int_t i = 0; i< NMCmodes3C ; i++){
1179      TotalControlII += ControlII[i];
1180      TotalControlIIError += pow(ControlIIErrors[i],2.);
1181    }
1182    TotalControlIIError = sqrt(TotalControlIIError);
1183
1184    /*Before Pi0Mass cut part*/
1185    TH1D *MassKCand3CComp[7];
1186    Int_t NMCmodes3CComp =7;
1187    for(Int_t i = 0 ; i < NMCmodes2C;i++ ){
1188      MassKCand3CComp[i]= (TH1D*)
1189      (TFile::Open((File[i]).c_str())->
1190      Get("KpiNGamma/3Clusters/MassKaonCand2")->
1191      Clone(("3CComp"+ModeName[i]).c_str()));
1192      MassKCand3CComp[i]->Sumw2();
1193    }
1194
1195    HistCol->cd();
1196    Canvas->cd();
1197
1198    TH1D *MassKCand3CScaledComp[7];
1199    Double_t EMassKCand3CScaledComp[7]={};
1200    Double_t TotalMCEvents3CComp = 0;
1201    Double_t ETotalMCEvents3CComp = 0 ;
1202
1203    /*Scaling histograms according to K2pi0 as normalization channel*/
1204    for(Int_t i = 0; i< NMCmodes3C ; i++){//loop over Histograms
1205      if(i == Data) continue;
1206      MassKCand3CScaledComp[i] =
1207      (TH1D*)MassKCand3CComp[i]->Clone(("3CCompScaled "+ModeName[i]).c_str());
1208      for(Int_t B = 1; B+1 < MassKCand3C[i]->GetNbinsX(); B++ ){
1209      //Loop Over bins
1210        if(MassKCand3CScaledComp[i]->GetBinContent(B) == 0) continue;
1211        if(i != K3pi0){
1212    MassKCand3CScaledComp[i]->SetBinError(B,
1213      MassKCand3CScaledComp[i]->GetBinContent(B)
1214      *ScaleFactorMC2K2pi[i]
1215      *sqrt(1./MassKCand3CComp[i]->GetBinContent(B)
1216      +pow(EScaleFactorMC2K2pi[i]/ScaleFactorMC2K2pi[i],2.)));
1217    MassKCand3CScaledComp[i]->SetBinContent(B,
1218      MassKCand3CScaledComp[i]->GetBinContent(B)*ScaleFactorMC2K2pi[i]);
1219        }
```

```
1220        EMassKCand3CScaledComp[i] +=
1221        pow(MassKCand3CScaledComp[i]->GetBinError(B), 2.);
1222      }
1223      EMassKCand3CScaledComp[i] =
1224      sqrt(EMassKCand3CScaledComp[i]);
1225    }
1226
1227    /*Normalization with respec to Data*/
1228    TH1D* MassKCand3CNormalizedComp[7];
1229    for(Int_t i = 0 ; i< NMCmodes3C; i++){
1230      if(i == Data) continue;
1231      MassKCand3CNormalizedComp[i] =
1232      (TH1D*)MassKCand3CScaledComp[i]->
1233      Clone(("3CCompNormalized "+ModeName[i]).c_str());
1234      for(Int_t B = 1 ;
1235      B+1 < MassKCand3CNormalizedComp[i]->GetNbinsX();B++){
1236        if(MassKCand3CNormalizedComp[i]->GetBinContent(B) == 0) continue;
1237        if(i != K3pi0){
1238    MassKCand3CNormalizedComp[i]->
1239      SetBinError(B, MassKCand3CNormalizedComp[i]->
1240      GetBinContent(B)*K2piNormalization*
1241      sqrt(pow(EK2piNormalization/K2piNormalization,2.)
1242      +pow(MassKCand3CNormalizedComp[i]->GetBinError(B)
1243      /MassKCand3CNormalizedComp[i]->GetBinContent(B),2.)));
1244    MassKCand3CNormalizedComp[i]->
1245      SetBinContent(B,MassKCand3CNormalizedComp[i]->
1246      GetBinContent(B)*K2piNormalization);
1247        }
1248        else{
1249    MassKCand3CNormalizedComp[i]->
1250      SetBinError(B, MassKCand3CNormalizedComp[i]->
1251      GetBinContent(B)*K3pi0Normalization*
1252      sqrt(pow(EK3pi0Normalization/K3pi0Normalization,2.)
1253      +pow(MassKCand3CNormalizedComp[i]->
1254      GetBinError(B)/MassKCand3CNormalizedComp[i]->GetBinContent(B),2.)));
1255    MassKCand3CNormalizedComp[i]->
1256      SetBinContent(B,MassKCand3CNormalizedComp[i]->
1257      GetBinContent(B)*K3pi0Normalization);
1258        }
1259      }
1260    }
1261
1262    /*Printing Routine*/
1263    MassI = 300.;
1264    MassF = 750.;
1265    for(Int_t i = 0 ; i < NMCmodes3C; i++){//Original MC Events
1266      MassKCand3CComp[i]->GetXaxis()->
1267      SetTitle("Mass #pi^{+} #gamma#gamma#gamma [MeV/c^{2}]");
1268      MassKCand3CComp[i]->GetYaxis()->SetTitle("Events/[1MeV/c^{2}]");
1269      MassKCand3CComp[i]->GetXaxis()->SetRangeUser(MassI,MassF);
1270      MassKCand3CComp[i]->SetMarkerStyle(7);
1271      if(i != Data){
1272        MassKCand3CComp[i]->SetTitle(
1273        ("K^{+} Mass, before #pi^{0} mass cut,"
1274        +StyledModeName[i]+" MC "+MCTipe).c_str());
1275        MassKCand3CComp[i]->SetLineColor(MarkerColor[i]);
1276        MassKCand3CComp[i]->SetMarkerColor(Color[i]-1);
1277        MassKCand3CComp[i]->SetFillColor(Color[i]);
1278        MassKCand3CComp[i]->Draw("HIST E1");
1279      }
1280      else {
1281        //MassKCand3CComp[i]->SetTitle(
1282        ("K^{+} Mass"+DataGroup+" Data").c_str());
1283        MassKCand3CComp[i]->SetTitle(
1284        ("Reconstructed K^{+} Mass, Events With 3 Neutral Clusters Final Selection"));
1285        MassKCand3CComp[i]->SetLineColor(kBlack);
1286        MassKCand3CComp[i]->Draw("E1");
1287      }
1288
1289      gStyle->SetOptStat(1111111);
1290      Canvas->Update();
1291      gStyle->SetStatX(0.9);
1292      gStyle->SetStatY(0.9);
```

```
1293      Canvas->SetLogy();
1294      Canvas->Modified();
1295      //Canvas->Print(PDF.c_str());
1296    }
1297
1298    for(Int_t i = 0 ; i < NMCmodes3C; i++){//MC Events scaled to Normalization
1299      if(i == Data) continue;
1300      MassKCand3CScaledComp[i]->
1301      GetXaxis()->SetTitle("Mass #pi^{+} #gamma#gamma#gamma [MeV/c^{2}]");
1302      MassKCand3CScaledComp[i]->
1303      GetYaxis()->SetTitle("Events/[1MeV/c^{2}]");
1304      MassKCand3CScaledComp[i]->
1305      GetXaxis()->SetRangeUser(MassI,MassF);
1306      MassKCand3CScaledComp[i]->
1307      SetTitle(("K^{+} Mass, before #pi^{0} mass cut, Scaled to K2#pi,"
1308      +StyledModeName[i]+" MC "+MCTipe).c_str());
1309      if(i == K3pi0) MassKCand3CScaledComp[i]->SetTitle(
1310      ("K^{+} Mass, before #pi^{0} mass cut, Scaled to K3#pi^{0},"
1311      +StyledModeName[i]+" MC "+MCTipe).c_str());
1312      MassKCand3CScaledComp[i]->SetLineColor(MarkerColor[i]);
1313      MassKCand3CScaledComp[i]->SetFillColor(Color[i]);
1314      MassKCand3CScaledComp[i]->SetMarkerColor(Color[i]-1);
1315      MassKCand3CScaledComp[i]->SetMarkerStyle(7);
1316      MassKCand3CScaledComp[i]->Draw("HIST E1");
1317
1318      gStyle->SetOptStat(1111111);
1319      Canvas->Update();
1320      gStyle->SetStatX(0.9);
1321      gStyle->SetStatY(0.9);
1322      Canvas->SetLogy();
1323      Canvas->Modified();
1324      //Canvas->Print(PDF.c_str());
1325    }
1326
1327    for(Int_t i = 0 ; i < NMCmodes3C; i++){//MC Events scaled to Data
1328      if(i == Data) continue;
1329      MassKCand3CNormalizedComp[i]->GetXaxis()->
1330      SetTitle("Mass #pi^{+} #gamma#gamma#gamma [MeV/c^{2}]");
1331      MassKCand3CNormalizedComp[i]->GetYaxis()->
1332      SetTitle("Events/[1MeV/c^{2}]");
1333      MassKCand3CNormalizedComp[i]->GetXaxis()->
1334      SetRangeUser(MassI,MassF);
1335      MassKCand3CNormalizedComp[i]->SetTitle(
1336      ("K^{+} Mass, before #pi^{0} mass cut, Normalized to Data,"
1337      +StyledModeName[i]+" MC "+MCTipe).c_str());
1338      MassKCand3CNormalizedComp[i]->SetLineColor(MarkerColor[i]);
1339      MassKCand3CNormalizedComp[i]->SetFillColor(Color[i]);
1340      MassKCand3CNormalizedComp[i]->SetMarkerColor(Color[i]-1);
1341      MassKCand3CNormalizedComp[i]->SetMarkerStyle(7);
1342      MassKCand3CNormalizedComp[i]->Draw("HIST E1");
1343
1344      gStyle->SetOptStat("mer");
1345      Canvas->Update();
1346      gStyle->SetStatX(0.9);
1347      gStyle->SetStatY(0.9);
1348      Canvas->SetLogy();
1349      Canvas->Modified();
1350      Canvas->Print(PDF.c_str());
1351    }
1352
1353    TLegend *Legend3CComp = new TLegend();
1354    THStack *Stack3CComp = new THStack();
1355    for(Int_t i = 0; i<Pi0gg; i++ ){
1356      if(MassKCand3CComp[i]->GetEntries() == 0) continue;
1357      if(i != Kpiggg) Legend3CComp->
1358      AddEntry(MassKCand3CComp[i],StyledModeName[i].c_str(),"lpf");
1359      else Legend3CComp->
1360      AddEntry(MassKCand3CComp[i],
1361      (StyledModeName[i]+", BR = 10^{-4}").c_str(),"lpf");
1362    }
1363
1364    Stack3CComp->Add(MassKCand3CNormalizedComp[Kpigg],"HIST E1");
1365    Stack3CComp->Add(MassKCand3CNormalizedComp[Kpie3],"HIST E1");
```

```
1366    Stack3CComp->Add(MassKCand3CNormalizedComp[K2pid],"HIST E1");
1367    Stack3CComp->Add(MassKCand3CNormalizedComp[K3pi0],"HIST E1");
1368    Stack3CComp->Add(MassKCand3CNormalizedComp[Kpiggg],"HIST E1");
1369    Stack3CComp->Add(MassKCand3CNormalizedComp[K2pi],"HIST E1");
1370
1371    MassKCand3CComp[Data]->SetMaximum(Stack3CComp->GetMaximum()*1e4);
1372    MassKCand3CComp[Data]->SetMinimum(3e-5);
1373    MassKCand3CComp[Data]->
1374    SetTitle("Reconstructed K+ Mass,
1375    Events With 3 Neutral Clusters After P & Pt Cuts");
1376    Canvas->Clear();
1377    Canvas->SetLogy(1);
1378    MassKCand3CComp[Data]->Draw();
1379    psl =
1380    (TPaveStats*)MassKCand3CComp[Data]->
1381    GetListOfFunctions()->FindObject("stats");
1382    psl->SetX1NDC(0.70); psl->SetX2NDC(0.90);
1383    psl->SetY1NDC(0.80); psl->SetY2NDC(0.90);
1384    Stack3CComp->Draw("SAME");
1385    MassKCand3CComp[Data]->Draw("SAME");
1386    Legend3CComp->SetX1(0.13); Legend3CComp->SetX2(0.4);
1387    Legend3CComp->SetY1(.6); Legend3CComp->SetY2(.9);
1388    Legend3CComp->Draw("SAME");
1389
1390    for(Int_t i= 0; i<Pi0gg; i++){
1391      MassKCand3CComp[i]->GetXaxis()->SetRangeUser(MassI,MassF);
1392      if(i == Data)continue;
1393      MassKCand3CNormalizedComp[i]->GetXaxis()->SetRangeUser(MassI,MassF);
1394    }
1395    Stack3CComp->GetXaxis()->SetRangeUser(MassI,MassF);
1396
1397    LineInf->SetY2(Stack3CComp->GetHistogram()->GetMaximum()*0.7e4);
1398    LineSup->SetY2(Stack3CComp->GetHistogram()->GetMaximum()*0.7e4);
1399    MaskedBox->SetY2(Stack3CComp->GetHistogram()->GetMaximum()*0.7e4);
1400    LineInf->Draw("SAME");
1401    LineSup->Draw("SAME");
1402    // MaskedBox->Draw("SAME");
1403    ControlRegion1->SetY(Stack3CComp->GetMaximum());
1404    ControlRegion2->SetY(Stack3CComp->GetMaximum());
1405    MaskedRegion->SetY(Stack3CComp->GetMaximum()*1e1);
1406    ControlRegion1->Draw("SAME");
1407    ControlRegion2->Draw("SAME");
1408    MaskedRegion->Draw("SAME");
1409    gStyle->SetOptStat("e");
1410    Canvas->Modified();
1411    Canvas->Update();
1412    Canvas->Print(PDF.c_str());
1413
1414    Similarity(Canvas,
1415    Stack3CComp,MassKCand3CComp[Data],Legend3CComp,360.,440.,420,0);
1416    psl->SetX1NDC(0.70); psl->SetX2NDC(0.90);
1417    psl->SetY1NDC(0.80); psl->SetY2NDC(0.90);
1418    //gStyle->SetOptStat("e");
1419    LineInf->SetY2(2.);
1420    LineSup->SetY2(2.);
1421    MaskedBox->SetY2(4.9);
1422
1423
1424    ControlRegion1->SetTextSize(0.09);
1425    ControlRegion2->SetTextSize(0.09);
1426    MaskedRegion->SetTextSize(0.09);
1427    MaskedRegion->SetX(475.); MaskedRegion->SetY(0.05);
1428    ControlRegion1->SetX(350.); ControlRegion1->SetY(.05);
1429    ControlRegion2->SetX(570.); ControlRegion2->SetY(.05);
1430
1431    LineInf->Draw("SAME");
1432    LineSup->Draw("SAME");
1433    //MaskedBox->Draw("SAME");
1434    ControlRegion1->Draw("SAME");
1435    ControlRegion2->Draw("SAME");
1436    MaskedRegion->Draw("SAME");
1437    Canvas->Print(PDF.c_str());
1438
```

```
1439    Canvas->Print((PDF+"]").c_str());
1440
1441    /*Output File*/
1442    std::ofstream OutputFile;
1443    OutputFile.open(FileName+".txt",std::ios::trunc);
1444    if(!OutputFile.is_open()){
1445      std::cout<<"Couldn't create nor open "+FileName<<std::endl;
1446      return ;
1447    }
1448    else{
1449      OutputFile<<
1450      "----------MC modes simulated and Autopass values----------"<<
1451      std::endl;
1452      for(Int_t i = 0; i< Pi0gg ;i++){
1453        if(Data == i) continue;
1454        OutputFile<<ModeName[i]+" Simulated Events: "<<
1455        TotalEvents[i]<<" Br: "<<BRatio[i]<<" \\pm "<<
1456        ErrorBRatio[i]<<std::endl;
1457      }
1458      OutputFile<<std::endl<<std::endl;
1459      OutputFile<<"--------------------Normalization factors--------------------"<<
1460      std::endl;
1461      OutputFile<<"Normalization factor K2pi: "<<
1462      K2piNormalization<<" \\pm "<<EK2piNormalization <<std::endl;
1463      OutputFile<<"Normalization factor K3pi0: "<<
1464      K3pi0Normalization<<" \\pm "<<EK3pi0Normalization<<std::endl;
1465      OutputFile<<std::endl<<std::endl;
1466      OutputFile<<"--------------------Calculated Acceptances--------------------"<<
1467      std::endl;
1468      OutputFile<<"Acceptance of  K2pi: "<<
1469      AcceptanceK2pi<<" \\pm "<<EAcceptanceK2pi<<std::endl;
1470      OutputFile<<"Acceptance of  K3pi0: "<<
1471      AcceptanceK3pi0<<" \\pm "<<EAcceptanceK3pi0<<std::endl;
1472      OutputFile<<"Acceptance of Kpiggg: "<<
1473      AcceptanceKpiggg<<" \\pm "<<EAcceptanceKpiggg<<std::endl;
1474      OutputFile<<std::endl<<std::endl;
1475      OutputFile<<"--------------------Normalization Events--------------------"<<
1476      std::endl;
1477      OutputFile<<"Number of K2pi: "<<NumberK2pi<<" \\pm "<<ENumberK2pi<<std::endl;
1478      OutputFile<<"Number of K3pi0: "<<NumberK3pi0<<" \\pm "<<ENumberK3pi0<<std::endl;
1479      OutputFile<<std::endl<<std::endl;
1480      OutputFile<<"----------Estimated Number of Decayed Kaons----------"<<
1481      std::endl;
1482      OutputFile<<"Number obtained with K2pi as Normalization:  "<<NKaonsK2pi<<
1483      " \\pm "<<ENKaonsK2pi<<std::endl;
1484      OutputFile<<"Number obtained with K3pi0 as Normalization: "<<NKaonsK3pi0<<
1485      " \\pm "<<ENKaonsK3pi0<<std::endl;
1486      OutputFile<<"Ratio between stimated number of Kaons [K2pi/K3pi0]: "<<
1487      RatioNKaons<<" \\pm "<<ERatioNKaons<<std::endl;
1488      OutputFile<<std::endl<<std::endl;
1489      OutputFile<<"--------------------Control Region II----------"<<
1490      std::endl;
1491      OutputFile<<"K2pi              : "<<ControlII[K2pi]<<
1492      "\\pm"<<ControlIIErrors[K2pi]<<std::endl;
1493      OutputFile<<"K3pi0             : "<<ControlII[K3pi0]<<
1494      "\\pm"<<ControlIIErrors[K3pi0]<<std::endl;
1495      OutputFile<<"K2piD             : "<<ControlII[K2pid]<<
1496      "\\pm"<<ControlIIErrors[K2pid]<<std::endl;
1497      OutputFile<<"Kpigg             : "<<ControlII[Kpigg]<<
1498      "\\pm"<<ControlIIErrors[Kpigg]<<std::endl;
1499      OutputFile<<"Total             : "<<TotalControlII<<
1500      " \\pm "<<TotalControlIIError<<std::endl;
1501      OutputFile<<std::endl<<std::endl;
1502
1503      OutputFile<<"----------Estimated Background for Kpiggg----------"<<
1504      std::endl;
1505      OutputFile<<"K2pi              : "<<Masked[K2pi]<<
1506      "\\pm"<<MaskedErrors[K2pi]<<std::endl;
1507      OutputFile<<"K3pi0             : "<<Masked[K3pi0]<<
1508      "\\pm"<<MaskedErrors[K3pi0]<<std::endl;
1509      OutputFile<<"K2piD             : "<<Masked[K2pid]<<
1510      "\\pm"<<MaskedErrors[K2pid]<<std::endl;
1511      OutputFile<<"Kpigg             : "<<Masked[Kpigg]<<
```

```
1512      "\\pm"<<MaskedErrors[Kpigg]<<std::endl;
1513      OutputFile<<"Total background: "<<TotalBackground<<
1514      " \\pm "<<ETotalBackground<<std::endl;
1515      OutputFile<<std::endl<<std::endl;
1516      OutputFile.close();
1517    }
1518    delete Legend2C;
1519    delete Legend4C;
1520    delete Legend3C;
1521    delete Legend3CComp;
1522    delete Stack2C;
1523    delete Stack4C;
1524    delete Stack3C;
1525    delete Stack3CComp;
1526    delete LineInf;
1527    delete LineSup;
1528    delete MaskedBox;
1529    delete MaskedRegion;
1530    delete ControlRegion1;
1531    delete ControlRegion2;
1532    delete Canvas;
1533    delete HistCol;
1534  }
```

Bibliography

[1] Philip Ilten et al. "Serendipity in dark photon searches". In: *Journal of High Energy Physics* 2018.6 (2018), p. 004. DOI: 10.1007/JHEP06(2018)004. arXiv: 1801.04847 [hep-ph].

[2] Y. Asano, E. Kikutani, S. Kurokawa, et al. "A new experimental limit for the decay K+ -¿ pi+ gamma gamma". In: *Physics Letters B* 113.2 (1982), pp. 195–198. DOI: 10.1016/0370-2693(82)90423-3.

[3] M. Kobayashi and T. Maskawa. "CP-Violation in the Renormalizable Theory of Weak Interaction". In: *Progress of Theoretical Physics* (1972). DOI: 10.1143/PTP.49.652.

[4] Nicola Cabbibo. "UNITARY SYMMETRY AND LEPTONIC DECAYS". In: *Physical Review Letters* (1963). DOI: 10.1103/PhysRevLett.10.531.

[5] S. Navas et al. "Review of particle physics". In: *Phys. Rev. D* 110.3 (2024), p. 030001. DOI: 10.1103/PhysRevD.110.030001.

[6] W. H. Furry. "A Symmetry Theorem in the Positron Theory". In: *Physical Review* 51.2 (1937), pp. 125–129. DOI: 10.1103/PhysRev.51.125.

[7] C. L. Basham and P. K. Kabir. "Possible three-photon couplings". In: *Physical Review D* 15.11 (1977), pp. 3388–3393. DOI: 10.1103/PhysRevD.15.3388.

[8] Schwabl F. *Quantum Mechanics*. Springer Berlin, Heidelberg, 2007. DOI: https://doi.org/10.1007/978-3-540-71933-5.

[9] Bogdan Povh et al. *Particles and Nuclei*. Springer Berlin, Heidelberg, 1993. DOI: https://doi.org/10.1007/978-3-662-46321-5.

[10] G. Fontaine, B. Degrange, and P. Fleury. "Tracking Louis Leprince-Ringuet'sto cosmic-ray physics". In: *Physics Today* (2013). DOI: 10.1063/PT.3.1989.

[11] E. Cortina Gil et al. NA62 Collaboration. "Observation of the $K^+ \to \pi^+ \nu\bar{\nu}$ decay and measurement of its branching ratio". In: *J. High Energ. Phys.* (2025). DOI: https://doi.org/10.1007/JHEP02(2025)191.

[12] Fabienne Landua. "The CERN accelerator complex layout in 2022. Complexe des accélérateurs du CERN en janvier 2022". In: (2022). General Photo. URL: https://cds.cern.ch/record/2813716.

[13] E. Cortina Gil et al. NA62 Collaboration. "The Beam and detector of the NA62 experiment at CERN". In: *JINST* 12.05 (2017), P05025. DOI: 10.1088/1748-0221/12/05/P05025. arXiv: 1703.08501 [physics.ins-det].

[14] S. Agostinelli et al. “Geant4—a simulation toolkit”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 506.3 (2003), pp. 250–303. ISSN: 0168-9002. DOI: https://doi.org/10.1016/S0168-9002(03)01368-8. URL: https://www.sciencedirect.com/science/article/pii/S0168900203013688.

[15] Rene Brun and Fons Rademakers. *ROOT - An Object Oriented Data Analysis Framework*. See also "ROOT" [software], Release v6.28/04, 08/05/2023. 1996. DOI: 10.5281/zenodo.848818.

[16] International Organization for Standardization. *ISO/IEC 14882:2017: Programming Languages – C++*. C++17 Standard. ISO/IEC, 2017. URL: https://isocpp.org/std/the-standard.

[17] *NA62Framework*. 2020. URL: https://na62-sw.web.cern.ch.

[18] E. Cortina Gil et al. NA62 Collaboration. “Performance of the NA62 trigger system”. In: *JHEP* 03 (2023), p. 122. DOI: 10.1007/JHEP03(2023)122. arXiv: 2208.00897 [hep-ex].

[19] P. A. Vetter and S. J. Freedman. “Branching-ratio measurements of multiphoton decays of positronium”. In: *Physical Review A* 66.5 (Nov. 2002), p. 052505. DOI: 10.1103/PhysRevA.66.052505. URL: https://doi.org/10.1103/PhysRevA.66.052505.

[20] M. Silarski and the J-PET Collaboration. “Status and Prospects of Discrete Symmetries Tests in Positronium Decays with the J-PET Detector”. In: *arXiv preprint arXiv:1811.10081* (2018). DOI: 10.48550/arXiv.1811.10081. arXiv: 1811.10081 [nucl-ex]. URL: https://arxiv.org/abs/1811.10081.

[21] T. Junk. “Confidence level computation for combining searches with small statistics”. In: *Journal of Physics G: Nuclear and Particle Physics* 28 (2002), pp. 2693–2704. DOI: 10.1088/0954-3899/28/10/313.

[22] *Confidence Limits: Proceedings of the Workshop*. Tech. rep. CERN-2000-005. Geneva: CERN, 2000. URL: https://cds.cern.ch/record/411537/files/CERN-2000-005.pdf.

[23] Evgueni Goudzovski. *Implementation of the CLs method for a single counting experiment*. Tech. rep. NA62 at CERN, 2019. URL: https://na62.web.cern.ch/restricted/NotesDoc/NA62-19-04.pdf.