

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ В.Н. КАРАЗИНА  
ННІ «ФІЗИКО-ТЕХНІЧНИЙ ФАКУЛЬТЕТ»  
КАФЕДРА ПРИКЛАДНОЇ ФІЗИКИ ТА ФІЗИКИ ПЛАЗМИ

## Пояснювальна записка

до дипломного проекту (роботи)

БАКАЛАВРА

(освітньо-кваліфікаційний рівень)

на тему:

укр. Аналіз кутового розподілу треків у рамках Quality Control Framework  
в ALICE Inner Tracking System

англ. Analysis of Angular Track Distribution in the Quality Control Framework  
of the ALICE Inner Tracking System

Виконав(ла): студентка 4 курсу навчання  
за ОПІ бакалавр  
спеціальності 105 Прикладна фізика та наноматеріали  
освітня програма: Прикладна фізика  
(назва)

Компанієць Р.В.

(прізвище, ім'я, по батькові)

(особистий підпис)

Керівники к.ф.-м.н., ст.н.с. ННФ ХФТІ

Трубніков В.С.

(прізвище, ім'я, по батькові)

(особистий підпис)

к.ф.-м.н., доцент КПФФП ННІ «ФТФ»

Афанасьєва І.О.

(прізвище, ім'я, по батькові)

(особистий підпис)

Рецензент д.ф.-м.н., професор КМФБМНТ ННІ «ФТФ»

Баранник Є.О.

(прізвище, ім'я, по батькові)

(особистий підпис)

Харків – 2025 рік

## **Анотація**

У цій кваліфікаційній роботі розглянуто метод аналізу кутового розподілу треків у системі внутрішнього трекінгу (ITS) детектора ALICE у рамках Quality Control Framework. Розроблено програмний алгоритм на основі CERN ROOT, що дозволяє автоматично виявляти локальні відхилення у розподілах треків та пов'язувати їх із конкретними сенсорними чипами за допомогою геометричної ідентифікації. В результаті роботи сформовано систему діагностики, яка дозволяє здійснювати просторовий моніторинг ефективності чипів ALPIDE та виявляти потенційно неефективні області. Здійснено апробацію методики на експериментальних даних. Отримані результати мають як практичну цінність для підвищення надійності даних у фізичних експериментах, так і потенціал для розширення у напрямку автоматизованих систем контролю якості.

**Ключові слова:** ALICE, ITS, Quality Control, CERN ROOT, трек, ALPIDE, кутовий розподіл, ефективність чипа.

## **Annotation**

This bachelor's thesis presents a method for analyzing the angular distribution of tracks in the ALICE Inner Tracking System (ITS) within the Quality Control Framework. A custom CERN ROOT-based algorithm has been developed to automatically detect local deviations in track distributions and correlate them with specific sensor chips using geometrical identification. As a result, a diagnostic tool was created that enables spatial monitoring of ALPIDE chip performance and the detection of potentially inefficient areas. The method was tested on experimental data. The obtained results provide practical value in ensuring the reliability of data in high-energy physics experiments and show potential for future integration into automated quality control systems.

**Keywords:** ALICE, ITS, Quality Control, CERN ROOT, track, ALPIDE, angular distribution, chip efficiency.

## ЗМІСТ

<b>ЗМІСТ.....</b>	<b>3</b>
<b>ВСТУП.....</b>	<b>4</b>
<b>РОЗДІЛ 1. ОГЛЯД ЕКСПЕРИМЕНТУ ALICE.....</b>	<b>6</b>
1.1. Загальна характеристика LHC і ролі ALICE.....	6
1.2. Завдання експерименту ALICE у фізиці важких іонів.....	7
1.3. Архітектура детектора ALICE.....	9
<b>РОЗДІЛ 2. СИСТЕМА ВНУТРІШНЬОГО ТРЕКІНГУ (ITS).....</b>	<b>12</b>
2.1 Призначення ITS.....	12
2.2. Структура та компоненти ITS.....	13
2.3. Робота ITS у контексті трекінгу частинок.....	15
2.4. Чому важлива стабільність і контроль якості у ITS.....	17
<b>РОЗДІЛ 3. QUALITY CONTROL FRAMEWORK.....</b>	<b>19</b>
3.1. Навіщо потрібен контроль якості в ALICE: роль QC та типові проблеми з чипами.....	19
3.2. Методи оцінки якості в ALICE ITS.....	21
3.3. Місце мого аналізу у загальному QC.....	24
<b>РОЗДІЛ 4. АНАЛІЗ КУТОВОГО РОЗПОДІЛУ ТРЕКІВ.....</b>	<b>26</b>
4.1. Опис алгоритму.....	26
4.2. Математичні та статистичні методи.....	27
4.3. Як саме алгоритм визначає аномалії.....	29
4.4. Приклади результатів.....	30
<b>РОЗДІЛ 5. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ТА ІНСТРУМЕНТИ.....</b>	<b>34</b>
5.1. CERN ROOT та як ми його використовуємо.....	34
5.2. Приклад роботи з файлами.....	35
<b>ВИСНОВКИ.....</b>	<b>38</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>40</b>
<b>ДОДАТОК А.....</b>	<b>42</b>
<b>ДОДАТОК Б.....</b>	<b>47</b>
<b>ДОДАТОК В.....</b>	<b>48</b>
<b>ДОДАТОК Г.....</b>	<b>49</b>

## ВСТУП

Стабільність і точність роботи системи внутрішнього трекінгу (Inner Tracking System, ITS) є критично важливими для реалізації наукових завдань експерименту ALICE, що функціонує на Великому адронному колайдері (LHC) у CERN. Сучасні методи фізики високих енергій вимагають не лише високої просторової роздільної здатності детекторів, а й постійного контролю за їхньою ефективністю. Порушення у роботі окремих сенсорів можуть призводити до значних систематичних помилок під час реконструкції подій, що унеможлиблює точний фізичний аналіз. У цьому контексті розробка інструментів для автоматизованого виявлення неефективних чіпів ITS має актуальне як наукове, так і прикладне значення.

**Метою** даного дослідження є аналіз кутового розподілу щільності треків частинок з метою виявлення локальних відхилень, що можуть свідчити про знижену ефективність окремих чіпів ALPIDE у системі ITS, а також розробка програмного інструменту для автоматизованої ідентифікації таких неефективностей у рамках Quality Control Framework.

**Об'єктом** дослідження є процес реєстрації траєкторій частинок у системі внутрішнього трекінгу ALICE.

**Предметом** дослідження виступає алгоритм аналізу розподілу кутових координат ( $\phi$ ,  $\eta$ ) зареєстрованих треків для виявлення аномалій у роботі сенсорних чіпів.

У роботі використано комплекс методів:

- **Гістографічний аналіз** (використовується для побудови еталонного розподілу координати псевдошвидкості  $\eta = -\ln(\tan\frac{\theta}{2})$  та порівняння окремих зрізів при фіксованих  $\phi$ );
- **Методи нормалізації** (для усунення впливу абсолютної кількості подій і порівнянності форм розподілів);

- **Статистичний аналіз** відхилень (оцінка відносних аномалій порівняно з еталонним розподілом);
- **Просторова локалізація** аномалій (на основі координат  $\phi$  та  $\eta$  обчислюється відповідність до конкретних чіпів за їх геометричним розміщенням);
- **Візуалізація даних** (побудова PDF-звітів із графіками, що відображають відхилення та підозрілі області).

**Наукова цінність** роботи полягає в інтеграції просторової інформації про геометрію ITS із аналізом кутового розподілу треків. На відміну від раніше реалізованих методів, які обмежуються глобальною оцінкою ефективності, запропонований підхід дозволяє локалізувати потенційно проблемні чіпи з точністю до їхнього глобального ідентифікатора та координат.

**Практичне значення** результатів полягає у можливості оперативного виявлення нестабільної роботи окремих сенсорів, що значно спрощує процедуру контролю якості в експерименті ALICE.

**Апробація** отриманих результатів відбулася під час виконання дослідницького проєкту в межах стажування в CERN, а також у рамках наукової конференції ННЦ “Харківський фізико-технічний інститут” “XXIII Конференція з фізики високих енергій та ядерної фізики” (Харків, 08-10 квітня 2025 р.).

## РОЗДІЛ 1. ОГЛЯД ЕКСПЕРИМЕНТУ ALICE

### 1.1. Загальна характеристика LHC і ролі ALICE

Великий адронний колайдер (Large Hadron Collider, LHC) — це найбільший і найпотужніший прискорювач заряджених частинок у світі, який було запущено в експлуатацію Європейською організацією з ядерних досліджень (CERN) у 2008 році. Колайдер розташований на глибині близько 100 метрів під землею, у тунелі довжиною 27 кілометрів, що охоплює територію на кордоні між Францією та Швейцарією. Його основна функція — прискорення та зіткнення протонів або важких іонів зі швидкостями, близькими до швидкості світла, з метою дослідження фундаментальних властивостей матерії [1].

У межах LHC працює кілька великих експериментальних детекторів, серед яких: **ATLAS**, **CMS**, **LHCb** та **ALICE**. Кожен із них орієнтований на певне фізичне завдання. Унікальність експерименту **ALICE** (A Large Ion Collider Experiment) полягає в його спеціалізації на **зіткненнях важких іонів**, зокрема ядер свинцю (Pb–Pb). У таких зіткненнях утворюються надвисокі температури та густини ядерної матерії, що створює умови для переходу звичайної ядерної матерії в особливий стан — **кварк-глюонну плазму (КГП)** [19].

ALICE було спеціально сконструйовано для дослідження цього нового стану матерії, у якому кварки та глюони, зазвичай зв'язані всередині адронів (протонів і нейтронів), переходять у стан деконфайнменту, тобто можуть рухатися відносно вільно на відстанях, що перевищують розміри окремих адронів, утворюючи щільне середовище з сильною взаємодією. Умови, що створюються під час таких зіткнень, імовірно, нагадують ті, що панували у Всесвіті в перші мікросекунди після Великого вибуху. Тому експеримент ALICE виконує не лише прикладову функцію в межах LHC, а й дає змогу відтворити фізичну реальність на ранніх етапах розвитку космосу.

Окрім зіткнень важких іонів, ALICE також проводить дослідження в рамках протон-протонних (pp) та протон-свинцевих (p–Pb) зіткнень, які служать як контрольні дані. Це дозволяє відокремити ефекти, притаманні саме КГП, від тих, що зумовлені більш звичайною протонною динамікою [2].

Таким чином, ALICE не лише доповнює фізичну картину, що формується у межах LHC, а й робить унікальний внесок у дослідження **сильної взаємодії та квантової хромодинаміки (QCD)** в екстремальних умовах. Важливо підкреслити, що завдяки надвисокій точності вимірювальних систем та широкому спектру підсистем, ALICE здатен реєструвати та реконструювати траєкторії десятків тисяч частинок, які утворюються в кожному зіткненні. Саме така складність і масштабність експерименту зумовлюють необхідність постійного контролю якості даних, що, зокрема, є предметом даної роботи [20].

## 1.2. Завдання експерименту ALICE у фізиці важких іонів

Фізика важких іонів досліджує властивості матерії при екстремально високих температурах та густинах, за яких відбувається розпад адронів на елементарні компоненти — кварки та глюони. У звичайних умовах кварки та глюони перебувають у зв'язаному стані завдяки конфайнменту кольорового заряду, зумовленому сильною взаємодією. Однак у надвисокоенергетичних зіткненнях важких іонів створюються умови, за яких відбувається фазовий перехід до стану деконфайнменту — **утворюється кварк-глюонна плазма**, у якій частинки можуть взаємодіяти у колективному середовищі без утворення зв'язаних станів, типових для адронної матерії [3].

Експеримент ALICE було розроблено спеціально для дослідження **фазового переходу між ядерною матерією та КГП**, його динаміки та наслідків. Основне завдання ALICE полягає в реконструкції подій, що виникають у результаті зіткнень іонів свинцю (Pb–Pb) на надвисоких енергіях ( $\sqrt{s_{nn}} = 5.02 \text{ TeV}$ ), та у **вивченні поведінки сильно взаємодіючої матерії** за зазначених умов [21].

Серед ключових фізичних цілей ALICE можна виділити:

- **Дослідження характеристик КГП:** визначення температури, густини енергії, часу життя, в'язкості та інших параметрів цієї фази;
- **Вивчення механізмів утворення частинок:** зокрема, з'ясування того, як кварки та глюони рекомбінуються в адрони при охолодженні плазми;
- **Аналіз колективних ефектів:** таких як анізотропний потік (elliptic flow), який свідчить про гідродинамічну поведінку плазми;
- **Ідентифікація рідкісних частинок:** включаючи D-мезони, J/ψ-мезони та інші важкі кварк-системи, які чутливі до температури й густини середовища;
- **Вивчення гасіння струменів (jet quenching):** ефекту втрати енергії високоенергетичними кварками та глюонами в середовищі КГП, що дозволяє оцінити густину та непрозорість плазми.

Окрім того, ALICE активно аналізує **протон-протонні (pp)** та **протон-свинцеві (p-Pb)** зіткнення, які відіграють важливу роль як контрольні експерименти. Вони дозволяють відокремити ефекти, специфічні для важких іонів, від універсальних механізмів, притаманних будь-яким зіткненням частинок.

Важливим аспектом досліджень є також **вивчення порушення симетрій**, зокрема CP-симетрії, та тестування моделей квантової хромодинаміки (QCD) у режимі сильної взаємодії. Таким чином, експеримент ALICE надає безцінну інформацію для перевірки передбачень Стандартної моделі, а також відкриває потенціал для відкриття нових фізичних явищ поза її межами [4].

Фізика важких іонів у рамках ALICE має не лише фундаментальне значення для розуміння структури матерії, а й утворює міст між сучасними експериментами з високими енергіями та космологією, зокрема, моделями раннього Всесвіту.

### 1.3. Архітектура детектора ALICE

Детектор експеримента ALICE (A Large Ion Collider Experiment) є складною багатокомпонентною системою, спеціально адаптованою для вивчення зіткнень важких іонів на Великому адронному колайдері. Його архітектура передбачає точну реєстрацію та ідентифікацію величезної кількості частинок, що утворюються в кожному акті зіткнення, які можуть досягати десятків тисяч одиниць [5].

ALICE складається з кількох ключових підсистем, кожна з яких виконує унікальну функцію, забезпечуючи повноцінне просторово-часове та енергетичне охоплення події. Загалом, детектор умовно поділяється на центральну частину, розташовані спереду підсистеми та системи тригерингу [22].

#### Центральна частина:

1. **ITS (Inner Tracking System)** — внутрішній трекінг, найближчий до області взаємодії; забезпечує точне визначення первинної та вторинних вершин.
2. **TPC (Time Projection Chamber)** — головна трекінгова система, яка дозволяє реконструювати треки частинок та визначати їх заряд та імпульс.
3. **TRD (Transition Radiation Detector)** — використовується для ідентифікації електронів на основі ефекту переходного випромінювання.
4. **TOF (Time-Of-Flight)** — система вимірювання часу прольоту, яка забезпечує ідентифікацію частинок за допомогою точного визначення їх швидкості.
5. **HMPID (High Momentum Particle IDentification)** — використовується для ідентифікації частинок з високим імпульсом за допомогою черенковського випромінювання.

6. **EMCal (Electromagnetic Calorimeter)** — електромагнітний калориметр, що вимірює енергію фотонів та електронів.
7. **PHOS (Photon Spectrometer)** — високоточний спектрометр фотонів, розташований з одного боку від зони зіткнення.

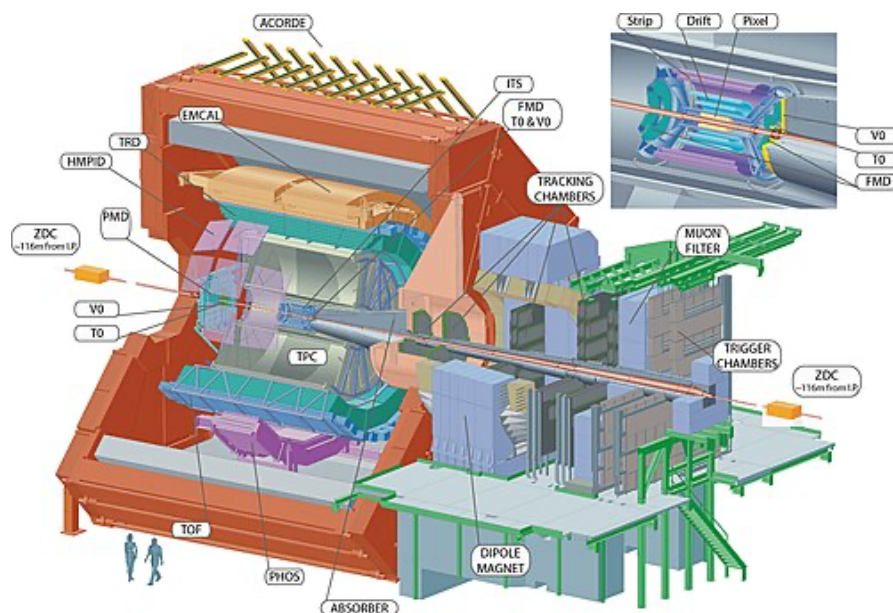


Рис. 1 – Загального розріз детектора ALICE з підписаними підсистемами. Зображення взяте з ресурсу [https://uk.wikipedia.org/wiki/ALICE\\_\(детектор\)](https://uk.wikipedia.org/wiki/ALICE_(детектор)), власна робота користувача *Pcharito*.

Системи, розташовані спереду:

1. **V0, ZDC (Zero Degree Calorimeter), AD та PMD** — забезпечують реєстрацію частинок у передньому напрямку (forward detectors), допомагають у визначенні центральності зіткнень, кратності подій та тригерингу.
2. **Muon Spectrometer** — розташований у передній частині детектора; призначений для ідентифікації та вимірювання імпульсів мюонів, особливо з розпадів важких кварк-систем.

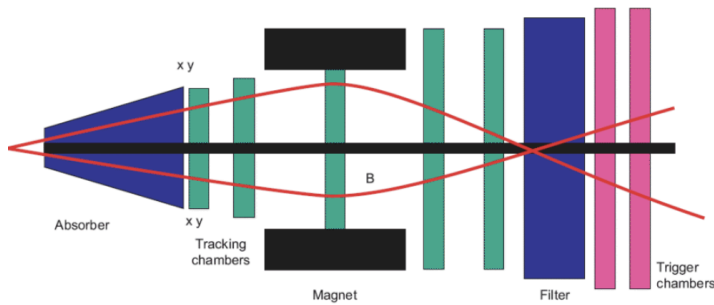


Рис. 2 – Схема мюонного спектрометра ALICE. Зображення взяте з ресурсу [https://alice-collaboration.web.cern.ch/menu\\_proj\\_items/Muon-Spect](https://alice-collaboration.web.cern.ch/menu_proj_items/Muon-Spect).

### Тригерна система:

- **Центральна тригерна система (Central Trigger Processor, CTP)** координує сигнали з різних частин детектора, визначаючи, які події слід зберігати для подальшого аналізу. Це критично важливо, оскільки лише невелика частина зіткнень записується в постійне сховище через обмеження на обсяг даних.

ALICE також оснащено складною інфраструктурою зчитування, калібрування та контролю якості даних у режимі реального часу, що дозволяє забезпечувати високу точність та надійність досліджень. Однією з ключових таких складових є **Quality Control Framework**, в межах якого і реалізовано запропонований у цій роботі алгоритм контролю ефективності [6].

## РОЗДІЛ 2. СИСТЕМА ВНУТРІШНЬОГО ТРЕКІНГУ (ITS)

### 2.1 Призначення ITS

Система внутрішнього трекінгу (Inner Tracking System, ITS) є найближчою до області взаємодії частиною детектора ALICE і виконує критично важливу роль у реконструкції первинної топології подій. Основним її завданням є **точне визначення координат початкового положення треків частинок**, зокрема первинної вершини зіткнення та можливих вторинних вершин, пов'язаних із розпадом короткоживучих частинок [23].

Враховуючи складність фізичних процесів у зіткненнях важких іонів, точність позиціонування на рівні кількох мікронів є вкрай важливою для коректної ідентифікації частинок та подальшої реконструкції траєкторій у центральній частині детектора. ITS також забезпечує:

- **Покращення роздільної здатності реконструкції треків**, особливо у випадках низькоімпульсних частинок, для яких інформація від TPC може бути недостатньою.
- **Визначення вторинних вершин розпаду**, що є ключовим для виявлення важких кварків (наприклад, D-мезонів та B-мезонів), які мають короткий час життя і розпадаються на відстані кількох сотень мікронів від первинної вершини.
- **Уточнення параметрів імпульсу частинок** у комбінації з іншими підсистемами ALICE.
- **Початкову ідентифікацію частинок**, особливо в комбінації з TOF та іншими PID-системами.

ITS працює у безпосередній близькості до пучка, тож на неї покладено й важливу функцію **моніторингу стану прискорювального середовища**, включно з виявленням розсіювання вторинних частинок чи небажаних ефектів калібрування.

Крім того, ITS відіграє суттєву роль у **формуванні тригерної інформації** на ранніх стадіях обробки подій. У поєднанні з іншими

детекторами, вона дозволяє виділяти події з потенційно цікавими характеристиками (наприклад, великою кількістю вторинних вершин чи високою щільністю треків) [7].

Таким чином, ITS є не лише першим етапом реєстрації треків частинок, а й фундаментальним інструментом для забезпечення високої точності, ефективності та достовірності фізичних результатів, що отримуються в межах експерименту ALICE.

## 2.2. Структура та компоненти ITS

Система внутрішнього трекінгу (Inner Tracking System, ITS) детектора ALICE є багатошаровою кремнієвою структурою, розробленою для забезпечення високоточної просторової реконструкції траєкторій заряджених частинок у безпосередній близькості до точки зіткнення. Архітектура ITS оптимізована для досягнення мінімального матеріального бюджету при максимальній точності позиціонування [8].

ITS складається з **семи циліндричних шарів кремнієвих піксельних детекторів**, розташованих концентрично навколо пучкової осі. Ці шари розміщуються в радіальному діапазоні від приблизно 22 мм до 400 мм від осі зіткнення. Вони поділяються на:

- **Внутрішні шари** (Layers 0–2): з максимальною просторовою роздільною здатністю, що забезпечують детальну інформацію про первинну вершину.
- **Середні шари** (Layers 3–4): відповідають за подальше уточнення траєкторій.
- **Зовнішні шари** (Layers 5–6): беруть участь у зв'язку з системою TPC та покращують ефективність трекінгу в умовах високої кратності подій.

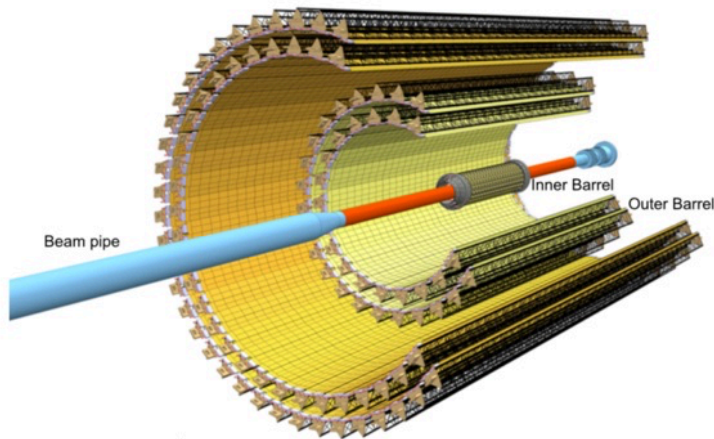


Рис. 3 – Схема ITS. Зображення взяте з ресурсу [https://alice-collaboration.web.cern.ch/menu\\_proj\\_items/its](https://alice-collaboration.web.cern.ch/menu_proj_items/its).

Усі шари побудовані з використанням **ALPIDE-чипів** (ALICE Pixel DEtector), які є монолітними активними піксельними сенсорами (MAPS), виготовленими за допомогою технології CMOS. Кожен чип має:

- площу близько  $15 \times 30 \text{ мм}^2$ ,
- роздільну здатність  $\sim 5 \text{ мкм}$  у площині, перпендикулярній до осі пучка,
- високу стійкість до радіаційного фону,
- інтегровану електроніку для зчитування сигналу.

Всього ITS містить **близько 12.5 мільярдів пікселів**, що забезпечує неперевершену роздільну здатність серед усіх трекерів LHC.

Уся система побудована з **модульних секцій**, які об'єднуються в так звані "**staves**" (сегменти кожного шару), а ті, у свою чергу, формують повне циліндричне покриття. Кожен **stave** містить кілька ALPIDE-чипів, розташованих у певному геометричному порядку. Для кожного чипа визначаються координати у просторі:  $\phi$  (**азимутальний кут**) та  $\eta$  (**псевдошвидкість**), що використовуються в подальшому для аналізу ефективності [9].

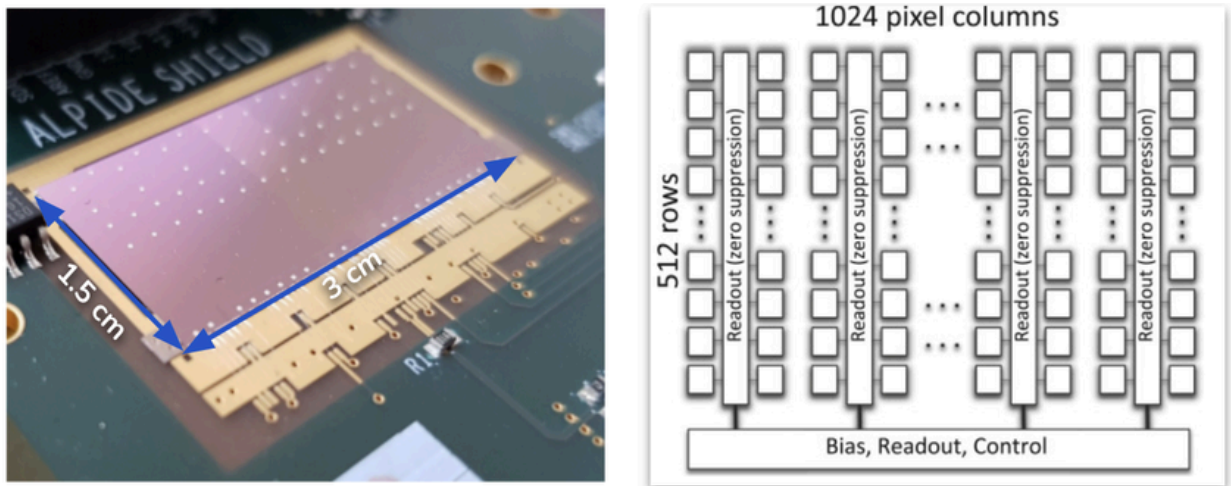


Рис. 4. Зліва – чип ALPIDE. Справа – загальна архітектура чипа. Зображення взяте з ресурсу [https://www.researchgate.net/figure/On-the-left-Photo-of-an-ALPIDE-chip-On-the-right-General-architecture-of-the-ALPIDE\\_fig1\\_367961601](https://www.researchgate.net/figure/On-the-left-Photo-of-an-ALPIDE-chip-On-the-right-General-architecture-of-the-ALPIDE_fig1_367961601).

ITS оснащено **високошвидкісною шиною зчитування**, яка дозволяє досягати пропускної здатності понад 1 Гбіт/с на кожен стев. Це забезпечує швидку обробку даних, що особливо важливо при роботі з високою частотою подій у Pb–Pb зіткненнях.

Оскільки ITS розташована безпосередньо навколо точки зіткнення, важливо мінімізувати **матеріальний бюджет**, аби зменшити ефекти розсіювання частинок. Загальна товщина всіх семи шарів становить **лише ~0.35% довжини радіаційного пробігу ( $X_0$ )**, що є одним із найкращих показників серед трекерів такого масштабу [24].

### 2.3. Робота ITS у контексті трекінгу частинок

Система внутрішнього трекінгу (ITS) є першою підсистемою ALICE, яка взаємодіє з частинками після зіткнення, і тому відіграє ключову роль у трекінгу — тобто у **реконструкції траєкторій заряджених частинок** у просторі на основі сигналів, зареєстрованих у детекторах.

**Основні завдання ITS у процесі трекінгу:**

1. **Визначення первинної вершини зіткнення**

Використовуючи високоточну просторову інформацію, ITS дозволяє з точністю до кількох мікронів встановити місце, де відбулося зіткнення частинок. Цей параметр є критично важливим для подальшої реконструкції подій і коректної ідентифікації частинок.

## **2. Реконструкція треків частинок з низьким імпульсом**

Завдяки своїй високій чутливості та малій відстані до точки взаємодії, ITS дозволяє реєструвати навіть ті частинки, які мають дуже малу кінетичну енергію та не досягають інших підсистем (зокрема TPC). Це суттєво розширює фазовий простір дослідження.

## **3. Ідентифікація вторинних вершин**

У багатьох фізичних процесах (зокрема при розпаді важких кварків) продукуються частинки, які розпадаються на деякій відстані від первинної вершини. ITS дозволяє відстежити такі вторинні вершини, що є основою для ідентифікації, наприклад, D- та B-мезонів.

## **4. Покращення точності трекінгу в поєднанні з іншими детекторами**

Дані ITS інтегруються з інформацією з Time Projection Chamber (TPC), Transition Radiation Detector (TRD) та Time Of Flight (TOF). Це дозволяє створити **комбіновану трекову модель**, яка забезпечує максимальну точність у визначенні імпульсу, ідентифікації частинки та визначенні її напрямку.

## **5. Формування вхідних даних для алгоритмів реконструкції**

ITS надає початкові точки для трек-фітів у глобальних реконструкційних алгоритмах ALICE. Без цієї інформації алгоритми втратили б важливий просторовий контекст, що особливо важливо в умовах високої щільності треків, характерної для зіткнень важких іонів.

## **6. Виявлення неефективних або пошкоджених чіпів**

Через прив'язку кожного зчитування до конкретного ALPIDE-чипа, ITS також дозволяє ідентифікувати ділянки з потенційними аномаліями або зниженням ефективності, що прямо впливає на якість кінцевої реконструкції [10].

## 2.4. Чому важлива стабільність і контроль якості у ITS

Система внутрішнього трекінгу ALICE (ITS), попри свою візуальну компактність, є однією з найскладніших з технологічної точки зору підсистем усього детектора. Вона складається з мільярдів пікселів, розташованих у тисячах чипів на кількох рівнях, кожен з яких має надзвичайно малу товщину, обмежену масу та виняткову просторову точність [25]. У таких умовах **стабільність роботи й регулярний контроль якості (Quality Control, QC)** є критично важливими з таких причин:

### 1. Безперервність трекінгу

Наявність навіть окремих дефектних або неефективних чипів може призвести до локальних “мертвих зон” у трекінговій системі. Це спотворює форму траєкторій частинок, ускладнює або повністю унеможлиблює реконструкцію окремих подій, особливо в регіонах високої щільності треків.

### 2. Системність помилок

На відміну від випадкових шумів або одноразових збоїв, апаратні дефекти мають **регулярну природу**: чип, який працює некоректно, продовжуватиме давати спотворені або відсутні дані в усіх подальших зіткненнях. Якщо не ідентифікувати проблему вчасно, вона вплине на **велику кількість зібраних подій**, що знижує фізичну достовірність аналізу.

### 3. Коректність фізичних висновків

Якість експериментальної фізики базується на достовірності вхідних даних. Якщо певні ділянки ITS систематично дають занижену або спотворену інформацію, це призводить до **зміщення статистики** — наприклад, до заниження ймовірності реєстрації частинок у певних напрямках (кутових зонах). Такий ефект особливо важливий при вивченні **анізотропії, кореляцій та розпадів короткоживучих частинок**, де точність трекінгу є вирішальною.

### 4. Вчасна діагностика й відновлення

Регулярний QC дозволяє виявити відмови на ранніх стадіях. Це важливо не лише для **апаратного ремонту або перезавантаження підсистеми**, а й

для коригування реконструкційних алгоритмів (наприклад, для виключення з аналізу даних із пошкоджених регіонів).

### **5. Роль у трекінгових алгоритмах**

ITS — це стартова точка для глобального трекінгу, і якщо початкова інформація зіпсована, вся траєкторія далі буде хибною. Алгоритми реконструкції треків зазвичай мають вбудовані модулі для **виключення чи компенсації аномальних зон**, однак для цього необхідно мати **актуальну інформацію про стан чипів**.

### **6. Масштаб і складність**

Загальна кількість ALPIDE-чипів у ITS становить понад **24 000 одиниць**, кожен з яких має індивідуальну просторову відповідальність. У таких масштабах **ручна перевірка неможлива**, тому застосовуються автоматизовані алгоритми виявлення неефективностей — зокрема такі, як алгоритм, реалізований у межах цієї роботи [11].

## РОЗДІЛ 3. QUALITY CONTROL FRAMEWORK

### 3.1. Навіщо потрібен контроль якості в ALICE: роль QC та типові проблеми з чипами

У надскладних експериментальних установах, таких як ALICE, де точність вимірювань критично залежить від функціонування кожного окремого компонента, систематичний **контроль якості (Quality Control, QC)** є невід'ємною частиною наукової методології. Особливо це стосується системи внутрішнього трекінгу (ITS), яка є першим і найчутливішим етапом у ланцюгу реконструкції частинок [12].

Контроль якості забезпечує:

- **Достовірність фізичних результатів**, запобігаючи внесенню систематичних похибок, викликаних апаратними проблемами.
- **Ідентифікацію та локалізацію дефектів**, що дозволяє швидко реагувати та вносити відповідні корекції до алгоритмів реконструкції.
- **Оцінку стабільності підсистем у часі**, дозволяючи моніторити деградацію компонентів та проводити превентивне обслуговування.
- **Автоматизовану верифікацію** після монтажу або ремонту апаратного забезпечення.

У масштабах ALICE QC функціонує не як разова процедура, а як **безперервний цикл перевірки та підтвердження надійності систем**, що інтегрується в обробку кожного нового набору експериментальних даних [13].

#### Типові проблеми, які виявляються в рамках QC ITS

##### 1. Нефективність чипів

Це одна з найпоширеніших проблем у ITS. Вона проявляється у вигляді:

- заниженого числа зареєстрованих треків у певному кутовому регіоні;
- повної відсутності відгуку на частинки (так звані "мертві чипи");

- статистично значущого відхилення від середнього еталонного розподілу по  $\eta$ - $\phi$  координатах.

2. Такі аномалії легко виявляються через **кутовий аналіз розподілу треків**, як реалізовано в межах цієї роботи: при нормалізованому порівнянні зрізів розподілу по  $\eta$  при фіксованому  $\phi$ , виявляються області з надмірними або недостатніми значеннями.

### 3. Підвищений шум або фонові сплески

Частина ALPIDE-чипів може давати **аномально високий рівень фонові активності**, навіть за відсутності частинок у відповідному регіоні.

Це може бути спричинено:

- локальними апаратними дефектами;
- помилками у калібруванні;
- флуктуаціями напруги або температури.

4. Такі чипи також виділяються в кутовому аналізі, але на відміну від неефективних, вони мають **надлишкові відхилення** від еталонного розподілу.

### 5. Темпоральна нестабільність

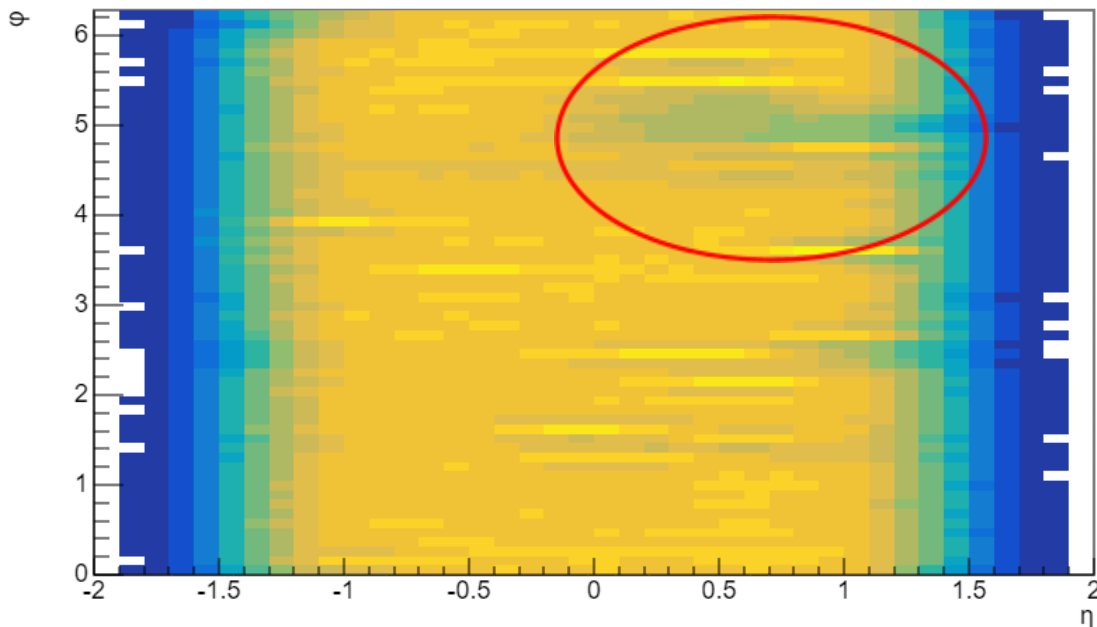
Окремі сенсорні чипи в системі ITS можуть демонструвати нестабільну поведінку, яка проявляється не постійно, а лише в певні моменти часу. Подібні випадки можуть бути зумовлені змінами температурного режиму, коливаннями напруги живлення, радіаційними ефектами або впливом магнітного поля, які не є сталими впродовж усього експерименту. Така поведінка отримала назву **темпоральної нестабільності**.

Для виявлення подібних ефектів важливо аналізувати не лише загальну картину аномалій, а й часовий розвиток відхилень. У рамках запропонованого алгоритму передбачене збереження точного часу (timestamp) кожної події, у якій було виявлено значне відхилення від референсного розподілу. Це дозволяє побудувати часові профілі появи аномалій і визначити, чи є відхилення постійними або зосередженими у конкретні часові інтервали.

### 6. Проблеми зі зчитуванням або синхронізацією

Інколи дефекти не пов'язані безпосередньо з чипом, а з каналом його зчитування, що викликає затримки або втрату частини інформації. Це теж може проявлятися у вигляді фрагментованих аномалій у  $\phi$  або  $\eta$ .

**Quality::GOOD** AngularDistribution (5m39s window)



*For Bachelor Thesis*

Рис. 5 – Приклад двовимірної гистограми кутового розподілу треків, де червоним еліпсом позначений регіон, у якому виявлена кількість записаних треків, значно менша за середню, що може свідчити про потенційну неефективність чипів.

Контроль якості — це не лише технічний обов'язок, а **фундаментальний науковий елемент**, що забезпечує фізичну достовірність і довіру до експерименту ALICE. Розробка, вдосконалення та впровадження алгоритмів QC, зокрема на основі кутового розподілу треків, є невід'ємною частиною сучасної високоточної фізики [14].

### 3.2. Методи оцінки якості в ALICE ITS

У контексті ALICE система внутрішнього трекінгу (ITS) має вирішальне значення для точної реконструкції первинних та вторинних вершин частинок. Через критичну роль ITS у всій інфраструктурі детектора питання

**контролю якості** цієї підсистеми набуває стратегічного значення.

Методи оцінки якості в ALICE ITS побудовані на комбінації **апаратних перевірок, калібрувальних процедур, аналітичного аналізу даних, а також візуалізаційних інструментів**. Основною метою цих підходів є виявлення відхилень у роботі окремих сенсорів, груп сенсорів або цілих підсистем [15].

### **1. Апаратний моніторинг і калібрування**

На початковому етапі контроль якості реалізується через системи **внутрішнього моніторингу електроніки**, що слідкують за температурою, споживанням енергії, напругою на кожному з чіпів.

Паралельно із запуском експерименту запускаються **калібрувальні пуски (calibration runs)**, у яких генеруються події без фактичних зіткнень — це дозволяє оцінити шумовий фон і виявити потенційно нестабільні елементи.

### **2. Реконструкція треків і статистичний аналіз**

Основним інструментом виявлення відхилень є **порівняння очікуваного розподілу треків у просторі  $\eta$ - $\phi$  з реальними даними**. У нормальних умовах розподіл треків має бути **гладким та симетричним** для кожного шару ITS. Всі **локальні флуктуації** за відсутності фізичних причин (наприклад, геометричних артефактів або рідкісних подій) інтерпретуються як потенційні **ознаки неефективності**.

У дипломному проєкті та розробленому алгоритмі застосовано метод **нормалізованого порівняння кожного зрізу по  $\phi$  із середнім по всьому  $\phi$  для фіксованого  $\eta$** .

Кожне суттєве відхилення, що перевищує заданий поріг (наприклад, 10%), фіксується як **аномалія**, і система намагається пов'язати її з конкретними сенсорами.

### **3. Просторовий аналіз та ідентифікація дефектних чіпів**

Завдяки тому, що геометрична інформація про чіпи ITS (зокрема їх координати у  $\eta$ - $\phi$ ) зберігається в окремому файлі конфігурації (ChipGeometry.txt), алгоритм дозволяє не лише виявити факт аномалії, а й

асоціювати її з конкретним чипом, з точністю до шару, стейва та локального ID.

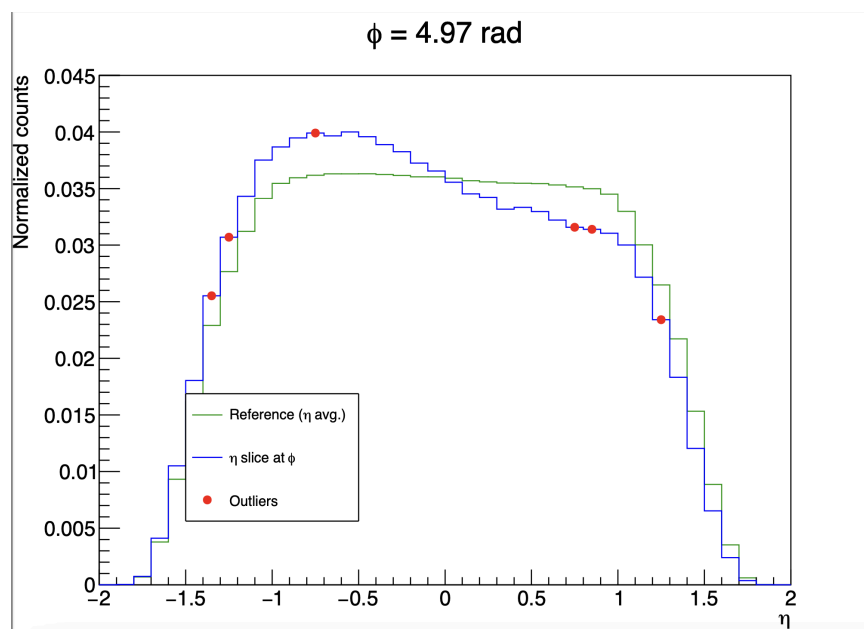
Таким чином, з кожною підозрілою точкою у фазовому просторі пов'язується **фізичний елемент системи**, який потенційно потребує діагностики або заміни. Додатково результати записуються у текстовий та CSV-файли, що уможлиблює **масштабний аналіз** та генерацію звітів.

#### 4. Візуалізація та вивід результатів

У якості візуалізаційного методу використовуються **гістограми у форматі PDF**, які автоматично генеруються для кожного кута  $\phi$  з аномальними відхиленнями:

- **Зелений графік** — еталонний розподіл по  $\eta$ .
- **Синій графік** — поточний розподіл для  $\phi$ .
- **Червоні маркери** — виявлені аномалії (outliers).

Це не лише полегшує **візуальну верифікацію**, а й надає можливість фізикам експерименту швидко приймати рішення щодо валідності даних.



*For Bachelor Thesis*

Рис. 6 – Приклад отриманої проєкції  $\eta$  з візуалізованими аномаліями.

## 5. Інтеграція з внутрішнім ALICE QC Framework

Розроблений підхід відповідає філософії “**data quality tagging**”, що вже застосовується у рамках QC ALICE. На практиці, результати подібного аналізу можуть бути:

- використані для **маскування дефектних сенсорів** під час реконструкції;
- враховані при **виборі подій для фізичного аналізу**;
- інтегровані до бази даних **quality flags**, яка формується автоматично після обробки кожного прогона.

Методи оцінки якості в ALICE ITS базуються на комбінації апаратного, статистичного та просторового аналізу. Запропонований у цій роботі алгоритм не лише вписується в цю систему, а й розширює її за рахунок гнучкої інтеграції з ROOT-інструментарієм та автоматизованого картографування аномалій [16].

### 3.3. Місце мого аналізу у загальному QC

У межах загальної системи забезпечення якості даних (Quality Control, QC) в ALICE, запропонований алгоритм займає проміжне місце між **первинною обробкою детекторних сигналів і підсумковим етапом валідації даних для фізичного аналізу**. Його основна роль полягає в **оперативному виявленні локальних порушень ефективності в роботі сенсорних чипів ITS**.

На відміну від стандартних методів, які переважно оцінюють **інтегральні характеристики** на рівні всього шару або підсистеми, розроблений підхід зосереджений на **детальному аналізі кутового розподілу треків у фазовому просторі ( $\eta$ - $\phi$ )**, що дозволяє виявляти **локалізовані аномалії**, пов’язані з окремими чипами.

Алгоритм може бути інтегрований у регулярний **QC pipeline** як додатковий інструмент для:

- формування **чорного списку неефективних сенсорів**;
- позначення часткових деградацій чипів у часі;
- автоматичного формування **індикаторів якості прогона (run quality flags)** на основі статистичної оцінки.

Таким чином, розроблений метод аналізу дозволяє підвищити ефективність наявних процедур контролю якості в системі ITS завдяки високій просторовій точності у виявленні потенційно дефектних чипів. Окрім цього, алгоритм забезпечує можливість побудови автоматизованих карт неефективностей у координатному просторі ( $\eta$ ,  $\phi$ ), що сприяє більш оперативному й локалізованому виявленню проблем у структурі ALPIDE-чипів. Метод особливо корисний на етапах постобробки даних та ретроспективного аналізу, де потрібна висока деталізація та просторове зіставлення результатів з геометрією детектора.

## РОЗДІЛ 4. АНАЛІЗ КУТОВОГО РОЗПОДІЛУ ТРЕКІВ

### 4.1. Опис алгоритму

У межах даної роботи розроблено ROOT-алгоритм, що реалізує метод виявлення потенційних аномалій у розподілі треків частинок у просторі псевдошвидкості ( $\eta$ ) та азимутального кута ( $\phi$ ) в ALICE ITS. Головна ідея полягає в автоматизованому порівнянні локальних зрізів розподілу з еталонним, середнім розподілом по всіх  $\phi$ .

#### Основні етапи реалізації алгоритму:

##### 1. Завантаження геометрії чипів.

Зчитується файл ChipGeometry.txt, що містить інформацію про положення кожного ALPIDE-чипа у фазовому просторі: його глобальний і локальний ідентифікатори, розташування по шарах та координати  $\eta$ ,  $\phi$ . Дані завантажуються у структуру Chip.

##### 2. Обробка ROOT-файлу.

Відкривається ROOT-файл, який містить гістограми типу TH2D з назвою формату runXXX\_AngularDistribution\_YYY, де відображено кутовий розподіл треків у координатах  $\eta$ - $\phi$  для заданого номера прогона (runNumber).

##### 3. Генерація еталонного розподілу.

Створюється проекція ProjectionX по  $\eta$  на всіх  $\phi$  — це усереднений, нормалізований розподіл, який виступає еталоном для подальших порівнянь.

##### 4. Побудова локальних зрізів.

Для кожного  $\phi$  будується окремий розподіл  $\eta$  (зріз), який також нормалізується.

##### 5. Виявлення аномалій.

Здійснюється покрокове порівняння значень у кожному біні зрізу з еталонним розподілом. Якщо відносне відхилення у біні перевищує встановлений поріг (типово 10%), бін вважається аномальним.

##### 6. Просторове співвіднесення з чипами.

Для кожного аномального біна визначаються чипи, що розташовані в його околі у координатах  $\eta$ – $\phi$ . Це дозволяє встановити ймовірні джерела неефективності.

### 7. Запис результатів.

Ідентифіковані аномальні чипи записуються у .txt та .csv-файли разом із відповідною інформацією: timestamp,  $\phi$ ,  $\eta$ , ідентифікатори чипів, а також координати.

### 8. Візуалізація.

Для кожного випадку з відхиленням будується графік з:

- зеленим еталонним розподілом,
- синім — локальним,
- червоними маркерами — аномальними бінами.

### 9. Глобальна статистика.

У фінальній частині роботи створюється гістограма, яка демонструє, скільки разів кожен чип був ідентифікований як потенційно проблемний у межах одного прогона.

Алгоритм дозволяє не лише швидко ідентифікувати потенційні дефекти, а й пов'язати їх з конкретними фізичними елементами системи. Такий підхід є придатним для масштабування та інтеграції в регулярні QC-процедури ALICE.

## 4.2. Математичні та статистичні методи

Для виявлення локальних аномалій у розподілах треків у фазовому просторі використовується метод відносного відхилення у гістограмах нормалізованої густини ймовірностей. Це дозволяє незалежно від абсолютної кількості подій оцінити відхилення форми розподілу окремих  $\phi$ -зрізів від еталонного розподілу по  $\eta$ .

### 1. Побудова еталонного розподілу

Для кожного прогона будується **еталонна гістограма**  $H_{ref}(\eta)$ , яка отримується шляхом проекції двовимірної гістограми  $H(\eta, \phi)$  на вісь  $\eta$ :

$$H_{ref}(\eta) = \sum_{\varphi} H(\eta, \varphi)$$

**Гістограма нормалізується за інтегралом:**

$$\hat{H}_{ref}(\eta) = \frac{H_{ref}(\eta)}{\int H_{ref}(\eta) d\eta}$$

## 2. Побудова локального зрізу

Аналогічно, для кожного фіксованого значення  $\varphi$  будується **зріз розподілу по  $\eta$** :

$$H_{slice}^{\varphi_i}(\eta) = H(\eta, \varphi_i)$$

І після нормалізації:

$$\hat{H}_{slice}^{\varphi_i}(\eta) = \frac{H_{slice}^{\varphi_i}(\eta)}{\int H_{slice}^{\varphi_i}(\eta) d\eta}$$

## 3. Виявлення аномалій за відносним відхиленням

Для кожного біну  $\eta$  у зрізі обчислюється **відносне відхилення**:

$$\delta(\eta) = \frac{|H_{slice}^{\varphi_i}(\eta) - \hat{H}_{ref}(\eta)|}{\hat{H}_{ref}(\eta)}$$

Бін вважається аномальним, якщо:

$$\delta(\eta) > \varepsilon$$

де  $\varepsilon$  — функція, яку можна задати під час запуску алгоритма (наприклад, 0.1 або 10%).

## 4. Візуалізація результатів

Для кожного  $\varphi$ -зрізу будується графік із:

- **зеленим графіком** — еталонний розподіл,
- **синім графіком** — локальний зріз,
- **червоними точками** — аномальні значення  $\eta$ .

## 5. Просторове співвіднесення

Після виявлення аномального біна, його координати  $(\eta, \varphi)$  використовуються для пошуку фізичних чипів, що потрапляють у цю ділянку фазового простору. Це реалізується шляхом аналізу геометричних даних чипів, що зберігаються у вигляді:

$$Chip = \{ID_{global}, \eta, \varphi, layer, stave, \dots\}$$

Таким чином, математичний апарат дозволяє забезпечити кількісно обґрунтоване виявлення аномалій, зберігаючи при цьому інтуїтивну інтерпретацію результатів, необхідну для фізичного аналізу якості даних у ALICE.

### 4.3. Як саме алгоритм визначає аномалії

Реалізований алгоритм виявлення аномалій у кутовому розподілі треків побудований на поетапній обробці двовимірних гістограм з ROOT-файлів, які містять інформацію про розподіл треків у просторі  $\eta$ – $\varphi$  для кожного прогона. Нижче наведено основні етапи виконання аналізу:

#### 1. Зчитування геометрії чипів

Передусім із текстового файлу зчитується геометрична інформація про всі чипи системи ITS. Для кожного чипа відомі його глобальний ідентифікатор, координати  $\eta$  та  $\varphi$ , шар і номер модуля. Ця інформація дозволяє пізніше співвідносити виявлені аномалії з фізичним розташуванням чипа.

#### 2. Витяг гістограм з ROOT-файлу

Для заданого номера прогона відкривається відповідний ROOT-файл, у якому зберігаються гістограми типу **TH2D**, що представляють двовимірні розподіли треків. Програма автоматично фільтрує лише ті гістограми, що відповідають назві "runX\_AngularDistribution\_timestamp".

#### 3. Побудова еталонного розподілу

Із кожної гістограми будується еталонний розподіл по  $\eta$  (усереднення по всіх  $\varphi$ ), що служить базовою лінією для подальших порівнянь.

#### 4. Обробка кожного $\varphi$ -зрізу

Гістограма по  $\eta$  будується також для кожного окремого зрізу по  $\varphi$ . Обидві гістограми (еталонна й поточна) нормалізуються, після чого виконується обчислення відносного відхилення в кожному біні.

#### 5. Порогова перевірка

Якщо відхилення в певному біні перевищує встановлений поріг (типово — 10%), цей бін позначається як аномальний. Візуально такі біни виділяються на графіку червоними маркерами.

#### 6. Просторове співставлення з чипами

Для кожного аномального біна координати  $\eta$  та  $\varphi$  використовуються для пошуку відповідних чипів у ITS. Це дозволяє локалізувати джерело потенційної неефективності на фізичній мапі детектора.

#### 7. Збереження результатів

Інформація про виявлені аномалії зберігається у вигляді:

- графіків у форматі **PDF**, що ілюструють розподіли з виділеними аномаліями;
- **текстового звіту** (.txt) та **таблиці** (.csv) зі списком чипів, координатами аномалій та часовими мітками.

Таким чином, алгоритм не лише виконує виявлення статистично значущих відхилень у кутовому розподілі треків, але й дозволяє на пряму ідентифікувати чипи, що потенційно функціонують некоректно. Це створює інструмент для точного й ефективного моніторингу якості детектора ITS у режимі онлайн або офлайн-контролю.

#### 4.4. Приклади результатів

Результати аналізу кутового розподілу треків представлені у вигляді PDF-графіків та таблиць, що містять інформацію про виявлені відхилення для конкретних значень координат  $\varphi$  та  $\eta$ . Для кожного зрізу, у якому знайдено аномалії, формується візуалізація зі згладженими кривими та маркерами відхилень. Нижче подано приклад типового виходу алгоритму.

#### Типовий вигляд графіка

На згенерованому графіку зображено два розподіли:

- **Зелена лінія** – еталонний розподіл по  $\eta$ , який отримано шляхом усереднення всіх  $\phi$ -зрізів;
- **Синя лінія** – розподіл  $\eta$  для конкретного значення  $\phi$ ;
- **Червоні маркери** – бін із відхиленням, яке перевищує заданий поріг.

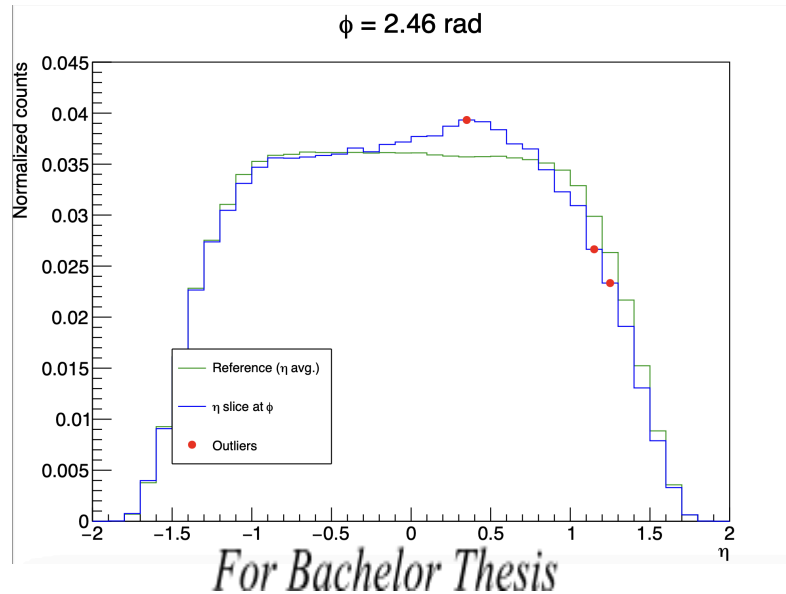


Рис. 7 – Приклад порівняння еталонного та поточного розподілу  $\eta$  для  $\phi = 2.46 \text{ рад}$ .

### Локалізація аномальних чипів

На основі координат відхилень, алгоритм шукає відповідні чипи в геометрії ITS. Якщо відхилення було виявлено, наприклад, при  $\phi = 1.04 \text{ рад}$  і  $\eta = 0.6$ , ідентифікується один або кілька чипів, що фізично знаходяться в цій області.

DetectedChips\_Run559362

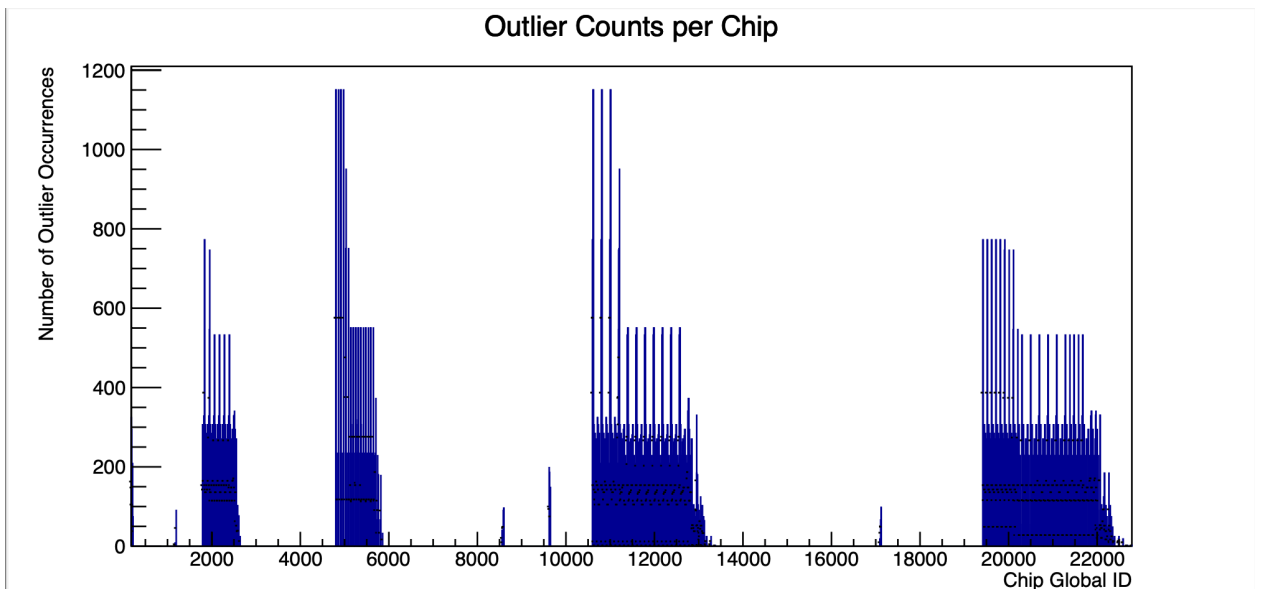
timestamp	phi	eta	chipID	layer	stave	module	chip_phi	chip_eta
1730369023854	4.86947	0.85	183	1	8	-169.011	0.850142	
1730369023854	4.86947	0.85	192	1	9	-146.468	0.858853	
1730369023854	4.86947	-0.85	194	1	9	-146.302	-0.847997	
1730369023854	4.86947	-0.85	203	1	10	-123.499	-0.851178	
1730369023854	4.86947	-0.85	212	1	11	-100.761	-0.853522	

*For Bachelor Thesis*

Рис. 8 – фрагмент таблиці .csv, у якій зазначений список чипів із їхніми координатами та іншими параметрами.

### Підсумкова гістограма кількості відхилень на чип

Наприкінці аналізу будується гістограма, що показує, скільки разів кожен чип був позначений як потенційно неефективний протягом заданого прогона. Це дозволяє швидко виявити найбільш проблемні ділянки в системі



*For Bachelor Thesis*

Рис. 9 – Кількість виявлених аномалій для кожного чипа.

### Інтерпретація

Переважна частина чипів не демонструє відхилень або потрапляє в межі статистичних флуктуацій. Проте в окремих випадках фіксується значне накопичення outlier-ів, що може свідчити про:

- локальні апаратні дефекти;
- деградацію елементів;
- проблеми з калібруванням;
- некоректну обробку сигналу.

Саме така концентрація дозволяє виділити чипи-кандидати на перевірку в рамках Quality Control Framework ALICE.

## РОЗДІЛ 5. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ТА ІНСТРУМЕНТИ

### 5.1. CERN ROOT та як ми його використовуємо

ROOT — це потужний об'єктно-орієнтований фреймворк для обробки та аналізу даних, створений у CERN спеціально для потреб високоефективної фізики елементарних частинок. Його основна мета — забезпечити дослідникам інструменти для зберігання, візуалізації, статистичного аналізу та математичної обробки експериментальних даних, зокрема отриманих з детекторів LHC [17].

ROOT написаний мовою C++ і підтримує макроси, модулі в стилі Python (через PyROOT), візуалізацію в реальному часі, збереження великих обсягів даних у спеціалізованих форматах (наприклад .root) та роботу з гістограмами, графами, деревами та матрицями [18].

#### Застосування ROOT у цій роботі

У межах даного дослідження ROOT використовується для реалізації алгоритму аналізу кутового розподілу треків у системі внутрішнього трекінгу ALICE (ITS). Основні функціональні блоки, реалізовані з використанням ROOT, охоплюють:

- **Читання даних** із .root-файлів, що містять гістограми типу TH2D, які відображають розподіл треків у просторі змінних  $\eta$  (псевдошвидкість) та  $\phi$  (азимутальний кут);
- **Проектування та обробку гістограм**: алгоритм використовує проєкції по осі  $\eta$  для отримання зрізів  $\phi$  та побудови еталонного розподілу;
- **Нормалізацію та порівняння розподілів** для виявлення статистично значущих відхилень;
- **Побудову графіків і візуалізацію результатів**: усі ключові кроки доповнюються візуальними звітами, які зберігаються у форматі PDF;
- **Генерацію вихідних таблиць** у форматах .txt та .csv із переліком виявлених неефективностей та координат відповідних чипів.

## Модулі та класи ROOT, задіяні в алгоритмі

У процесі роботи використовуються такі основні елементи:

- TFile — для зчитування даних із .root-файлів;
- TH2D та TH1D — для збереження та аналізу двовимірних і одновимірних гістограм;
- TCanvas — для побудови графіків;
- TGraph — для візуального виділення аномальних точок;
- TLegend, TStyle, TKey — для покращення читабельності та стилізації графіків.

## Переваги використання ROOT у контексті ALICE

- **Сумісність:** ROOT є основним інструментом для зберігання та обробки даних у рамках колаборації ALICE, що забезпечує повну відповідність структурі даних.
- **Гнучкість:** Можливість поєднувати аналітику, візуалізацію й автоматичну генерацію звітів у межах одного середовища.
- **Продуктивність:** ROOT оптимізовано для роботи з великими обсягами даних, що є критичним у контексті високочастотного детекторного моніторингу.

## 5.2. Приклад роботи з файлами

Однією з основних задач, реалізованих у межах даної роботи, є зчитування, обробка та візуалізація експериментальних даних, що зберігаються у форматі .root. Дані містять двовимірні гістограми, які описують кутовий розподіл треків (у координатах псевдошвидкості  $\eta$  та азимутального кута  $\phi$ ) для різних моментів часу. Для коректного аналізу таких даних було розроблено скрипт на базі ROOT, що реалізує кілька ключових кроків.

## Зчитування ROOT-файлу

За допомогою класу TFile відкривається ROOT-файл fout\_mw.root, що містить гістограми кутових розподілів:

```
TFile* file = TFile::Open("fout_mw.root");
if (!file || file->IsZombie()) {
    std::cerr << "[ERROR] Could not open ROOT file!" << std::endl;
    return;
}
```

Із файлу зчитуються всі об'єкти типу TH2D, які містять двовимірні розподіли у площині  $\eta$ - $\phi$ . Назви гістограм містять номер прогона та часову мітку.

### Обробка гістограм

Для кожної гістограми:

- створюється еталонний розподіл по  $\eta$  шляхом проєкції всієї гістограми;
- обчислюється нормалізація (масштабування на одиницю);
- здійснюється порівняння зі зрізами по  $\phi$  з метою виявлення відхилень.

```
TH1D* reference = hAngular->ProjectionX("ref_eta");
if (reference->Integral() == 0) {
    delete reference;
    continue;
}
reference->Scale(1.0 / reference->Integral());
```

### Виявлення аномалій

Порівняння кожного окремого розподілу  $\eta$  при фіксованому  $\phi$  із еталоном дозволяє виявити відносні відхилення. Якщо відхилення перевищує заданий поріг (наприклад, 10%), такий бін вважається аномальним:

```
double dev = std::abs(singleVal - refVal) / refVal;
if (dev > threshold)
    outliers.emplace_back(i, dev);
```

### Просторова локалізація чипів

Для кожного аномального значення координати ( $\phi$ ,  $\eta$ ) визначаються чипи ITS, які потенційно відповідають за відхилення. Це здійснюється шляхом порівняння геометричних координат чипів, зчитаних із текстового файлу ChipGeometry.txt, із координатами виявлених outlier-ів:

```
auto nearbyChips = FindChipsInRegion(chips, phi, eta);
if (nearbyChips.empty()) continue;
```

Інформація про такі чипи зберігається у форматах .txt та .csv для подальшого аналізу:

```
outFile << timestamp << "," << phi << "," << eta << "," << globalID << "," << layer << "," << stave << "," << chip_phi << "," << chip_eta << "\n";
csvFile << timestamp << "," << phi << "," << eta << "," << globalID << "," << layer << "," << stave << "," << chip_phi << "," << chip_eta << "\n";
```

### Візуалізація результатів

Для кожного  $\phi$  будується графік із трьома шарами:

1. Еталонний розподіл — зелений колір;
2. Розподіл для конкретного  $\phi$  — синій колір;
3. Виявлені аномалії — червоні маркери.

Графіки зберігаються у форматі .pdf для зручного перегляду та презентації:

```
c->SaveAs(Form("%s_phi%d.pdf", histName.c_str(), i));
delete c;
```

## ВИСНОВКИ

У межах даної роботи було реалізовано комплексний підхід до аналізу ефективності сенсорних чипів системи внутрішнього трекінгу (ITS) детектора ALICE шляхом дослідження кутового розподілу треків частинок. На основі аналізу двовимірних гістограм у площині ( $\eta$ ,  $\phi$ ), отриманих із ROOT-файлів, було створено програмний алгоритм, який дозволяє з високою точністю виявляти локальні аномалії у розподілах та просторово ідентифікувати чипи, потенційно пов'язані з відхиленнями. Алгоритм враховує геометрію детектора, здійснює нормалізацію даних для виключення статистичних флуктуацій та реалізує візуалізацію результатів у вигляді PDF-звітів і табличної інформації у форматах .txt та .csv.

Результати, отримані в ході роботи, демонструють не лише можливість оперативного виявлення неефективних або нестабільних чипів, але й забезпечують базу для подальшого вдосконалення процедур контролю якості у реальному часі. Особливу цінність має інтеграція просторового контексту — можливість пов'язати аномалії з конкретними координатами в ITS — що підвищує точність діагностики і дозволяє оперативно реагувати на потенційні збої під час збору даних.

Перспективи подальшої роботи передбачають удосконалення алгоритму в напрямку автоматичної класифікації типів аномалій (наприклад, відокремлення електронного шуму від механічних дефектів), інтеграцію з централізованими системами моніторингу CERN та використання методів машинного навчання для прогнозування відмов чипів. Крім того, можливою є адаптація алгоритму для використання в інших детекторних підсистемах ALICE або навіть в інших експериментах Великого адронного колайдера.

Таким чином, дана робота робить свій внесок у розвиток інструментів моніторингу детектора ALICE, підвищуючи надійність збору даних та точність фізичних вимірювань. У контексті Quality Control Framework така

аналітика є ключовою складовою як для оперативного втручання, так і для довгострокового стратегічного аналізу стабільності роботи детектора.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ALICE Collaboration. "The ALICE experiment at the CERN LHC." *Journal of Instrumentation*, 3:S08002, 2008.
2. CERN. The Large Hadron Collider. [Офіційний сайт CERN]  
<https://home.cern/science/accelerators/large-hadron-collider>
3. ALICE Collaboration. "Technical Design Report for the Upgrade of the ALICE Inner Tracking System." CERN-LHCC-2013-024, 2013.
4. ALICE Collaboration. "Upgrade of the ALICE Experiment: Letter of Intent." CERN-LHCC-2012-012.
5. S. Bjelogrić et al. "A framework for online and offline quality control in the ALICE O2 system." *EPJ Web of Conferences*, 245, 2020.
6. L. Musa. "The ALICE experiment: from heavy-ion physics to cosmic rays." *Nuclear Physics A*, vol. 774, 2006.
7. F. Carena et al. "ALICE Inner Tracking System upgrade: Overview." *Nuclear Instruments and Methods in Physics Research Section A*, 824, 2016.
8. ROOT Data Analysis Framework. [Офіційний сайт] <https://root.cern/>
9. Brun, R. & Rademakers, F. "ROOT — An object oriented data analysis framework." *Nuclear Instruments and Methods in Physics Research Section A*, 389(1–2), 1997.
10. ALICE Collaboration. "Performance of the ALICE experiment at the CERN LHC." *International Journal of Modern Physics A*, Vol. 29, 2014.
11. ALICE Collaboration. "Overview of the ALICE Offline Quality Control Framework." *CERN Note*, ALICE-O2-Note-2021-004.
12. ALICE Collaboration. "O2 Quality Control Documentation." GitHub Repository: <https://github.com/AliceO2Group/QualityControl>
13. ALICE ITS Upgrade Project. [Офіційний сайт проекту]  
<https://alice-its.web.cern.ch/>
14. Ivan Kisel et al. "ALICE High-Level Trigger: Real-Time Tracking and Calibration." *Computer Physics Communications*, Vol. 174, 2006.

15. K. Aamodt et al. “Alignment of the ALICE Inner Tracking System with cosmic-ray tracks.” *JINST*, 5(03):P03003, 2010.
16. ALPIDE chip documentation – ALICE ITS. [CERN documents archive]
17. S. Trogolo. “ALICE ITS performance and upgrade.” *Nuclear and Particle Physics Proceedings*, Vol. 273–275, 2016.
18. M. Krivda et al. “Online monitoring and data quality assurance in ALICE.” *Proceedings of the CHEP Conference*, 2012.
19. ALICE Collaboration. “Data Processing and Analysis in Run 3 and Beyond.” *ALICE Technical Report*, 2021.
20. ALICE Collaboration. “ITS2 Quality Assurance.” ALICE Internal Note, 2022.
21. C. Cavicchioli et al. “O2 Common Software and Framework.” *ALICE O2 Documentation*, CERN, 2021.
22. P. Moreira et al. “The GBT project.” *Topical Workshop on Electronics for Particle Physics*, 2007.
23. F. Grosa. “Quality Assurance of the ITS2 Stave Production.” ALICE Report, CERN, 2021.
24. A. Rossi. “Monitoring detector performance in ALICE using Quality Control.” *ALICE Workshop on Offline Software*, 2020.
25. CERN. “ALICE Offline Software Documentation.”  
<https://aliceo2.web.cern.ch/>

## ДОДАТОК А

### Повний вихідний код аналізу

```

#include <TFile.h>
#include <TH2D.h>
#include <TH1D.h>
#include <TCanvas.h>
#include <TGraph.h>
#include <TLegend.h>
#include <TStyle.h>
#include <TKey.h>
#include <TClass.h>
#include <iostream>
#include <fstream>
#include <sstream>
#include <vector>
#include <tuple>
#include <cmath>
#include <map>
#include <set>

struct Chip {
    int layer, stave, localID, globalID;
    double phi, eta;
};

// Load chip geometry from text file
std::vector<Chip> LoadChipGeometry(const std::string& filename) {
    std::vector<Chip> chips;
    std::ifstream infile(filename);
    if (!infile.is_open()) {
        std::cerr << "[ERROR] Could not open geometry file: " << filename << std::endl;
        return chips;
    }

    int layer, stave, localID, globalID;
    double phi, eta;
    while (infile >> layer >> stave >> localID >> globalID >> phi >> eta) {
        chips.push_back({layer, stave, localID, globalID, phi, eta});
    }

    std::cout << "[INFO] Loaded " << chips.size() << " chips from geometry." << std::endl;
    return chips;
}

// Find chips within a region defined by phi and eta windows
std::vector<Chip> FindChipsInRegion(const std::vector<Chip>& chips, double phi, double eta, double
window_eta = 0.01, double window_phi = 0.01) {
    std::vector<Chip> nearby;
    for (const auto& chip : chips) {
        double chip_phi_rad = chip.phi * M_PI / 180.0;
        double dphi = std::fabs(phi - chip_phi_rad);
        if (dphi > M_PI) dphi = 2 * M_PI - dphi;

        if (dphi <= window_phi && std::fabs(chip.eta - eta) <= window_eta)
            nearby.push_back(chip);
    }
}

```

```

    }
    return nearby;
}

// Detect outliers comparing a given distribution to the reference
std::vector<std::tuple<int, double>> DetectOutliers(TH1D* reference, TH1D* single, double threshold) {
    std::vector<std::tuple<int, double>> outliers;
    int nBins = reference->GetNbinsX();
    for (int i = 1; i <= nBins; ++i) {
        double eta = single->GetXaxis()->GetBinCenter(i);
        if (std::abs(eta) > 1.35) continue;

        double refVal = reference->GetBinContent(i);
        double singleVal = single->GetBinContent(i);
        if (refVal == 0) continue;

        double dev = std::abs(singleVal - refVal) / refVal;
        if (dev > threshold)
            outliers.emplace_back(i, dev);
    }
    return outliers;
}

// Main analysis function
void analyzeALPIDE(int runNumber, double threshold = 0.1, bool doPlot = true) {
    gStyle->SetOptStat(0);
    std::vector<Chip> chips = LoadChipGeometry("ChipGeometry.txt");
    if (chips.empty()) return;

    // Output files
    std::ofstream outFile(Form("DetectedChips_Run%d.txt", runNumber));
    std::ofstream csvFile(Form("DetectedChips_Run%d.csv", runNumber));
    if (!outFile.is_open() || !csvFile.is_open()) {
        std::cerr << "[ERROR] Could not create output files!" << std::endl;
        return;
    }

    outFile << "Run: " << runNumber << "\n";
    outFile << "timestamp,phi,eta,chipID,layer,stave,module,chip_phi,chip_eta\n";
    csvFile << "timestamp,phi,eta,chipID,layer,stave,module,chip_phi,chip_eta\n";

    // Map to store unique chips encountered (key: chip globalID, value: tuple of info)
    std::map<int, std::tuple<long int, double, double, int, int, int, double, double>> uniqueChips;

    // Map to count how many times each chip triggers an outlier
    std::map<int, int> chipOutlierCounts;

    TFile* file = TFile::Open("fout_mw.root");
    if (!file || file->IsZombie()) {
        std::cerr << "[ERROR] Could not open ROOT file!" << std::endl;
        return;
    }

    TIter next(file->GetListOfKeys());
    TKey* key;
    while ((key = (TKey*)next())) {
        TObject* obj = key->ReadObj();

```

```

if (!obj->InheritsFrom("TH2D")) continue;

std::string histName = obj->GetName();
if (histName.find(Form("run%d_AngularDistribution_", runNumber)) == std::string::npos) continue;

size_t pos = histName.find_last_of('_');
if (pos == std::string::npos || pos + 1 >= histName.size()) continue;
std::string timestampStr = histName.substr(pos + 1);
long int timestamp = std::stol(timestampStr);

TH2D* hAngular = dynamic_cast<TH2D*>(obj);
if (!hAngular) continue;

TH1D* reference = hAngular->ProjectionX("ref_eta");
if (reference->Integral() == 0) {
    delete reference;
    continue;
}
reference->Scale(1.0 / reference->Integral());

int nPhiBins = hAngular->GetNbinsY();
for (int i = 1; i <= nPhiBins; ++i) {
    double phi = hAngular->GetYaxis()->GetBinCenter(i);
    TH1D* single = hAngular->ProjectionX(Form("eta_phi_%d", i), i, i);
    if (single->Integral() == 0) {
        delete single;
        continue;
    }
    single->Scale(1.0 / single->Integral());

    auto outliers = DetectOutliers(reference, single, threshold);
    if (outliers.empty()) {
        delete single;
        continue;
    }

    TGraph* g = new TGraph();
    g->SetMarkerStyle(20);
    g->SetMarkerColor(kRed);

    for (auto& o : outliers) {
        int bin = std::get<0>(o);
        double eta = single->GetXaxis()->GetBinCenter(bin);
        double val = single->GetBinContent(bin);

        auto nearbyChips = FindChipsInRegion(chips, phi, eta);
        if (nearbyChips.empty()) continue;

        for (const auto& chip : nearbyChips) {
            // Count every occurrence of chip as an outlier
            chipOutlierCounts[chip.globalID]++;
            // Insert only if not already recorded
            if (uniqueChips.find(chip.globalID) == uniqueChips.end()) {
                uniqueChips[chip.globalID] = std::make_tuple(
                    timestamp, phi, eta,
                    chip.globalID, chip.layer, chip.stave, chip.phi, chip.eta
                );
            }
        }
    }
}

```

```

    }
}

g->SetPoint(g->GetN(), eta, val);
}

if (doPlot) {
    TCanvas* c = new TCanvas(Form("c_%s_%d", histName.c_str(), i), Form("Phi %.2f", phi),
800, 600);

    reference->SetLineColor(kGreen+2);
    single->SetLineColor(kBlue);
    reference->SetTitle(Form("#phi = %.2f rad; #eta; Normalized counts", phi));
    reference->GetYaxis()->SetRangeUser(0, 0.045);
    single->GetYaxis()->SetRangeUser(0, 0.045);

    reference->Draw("HIST");
    single->Draw("HIST SAME");
    g->Draw("P SAME");

    TLegend* leg = new TLegend(0.2, 0.2, 0.4, 0.4);
    leg->AddEntry(reference, "Reference (#eta avg.)", "l");
    leg->AddEntry(single, "#eta slice at #phi", "l");
    leg->AddEntry(g, "Outliers", "p");
    leg->Draw();

    c->SaveAs(Form("%s_phi%d.pdf", histName.c_str(), i));
    delete c;
}

delete g;
delete single;
}

delete reference;
}

// Now write unique chips only once to output files
for (const auto& [chipID, data] : uniqueChips) {
    long int timestamp;
    double phi, eta;
    int globalID, layer, stave;
    double chip_phi, chip_eta;

    std::tie(timestamp, phi, eta, globalID, layer, stave, chip_phi, chip_eta) = data;

    outFile << timestamp << "," << phi << "," << eta << "," << globalID << "," << layer << "," << stave
<< "," << chip_phi << "," << chip_eta << "\n";
    csvFile << timestamp << "," << phi << "," << eta << "," << globalID << "," << layer << "," << stave
<< "," << chip_phi << "," << chip_eta << "\n";
}

outFile.close();
csvFile.close();

// Now create histogram of chip global IDs based only on the CSV (unique chips counted)
// X-axis: chip globalID; Y-axis: how many times each chip was detected as outlier

```

```

if (!chipOutlierCounts.empty()) {
    int minID = chipOutlierCounts.begin()->first;
    int maxID = chipOutlierCounts.rbegin()->first;
    int nBins = maxID - minID + 1;

    TH1D* hChipCounts = new TH1D("hChipCounts", "Outlier Counts per Chip;Chip Global
ID;Number of Outlier Occurrences", nBins, minID - 0.5, maxID + 0.5);

    for (const auto& [chipID, count] : chipOutlierCounts) {
        hChipCounts->Fill(chipID, count);
    }

    TCanvas* c = new TCanvas("c_chipCounts", "Outlier Counts per Chip", 1200, 600);
    hChipCounts->SetFillColor(kAzure - 9);
    hChipCounts->Draw("BAR");
    c->SaveAs(Form("ChipOutlierCounts_Run%d.pdf", runNumber));
    delete c;
    delete hChipCounts;
}

file->Close();
delete file;
}

```

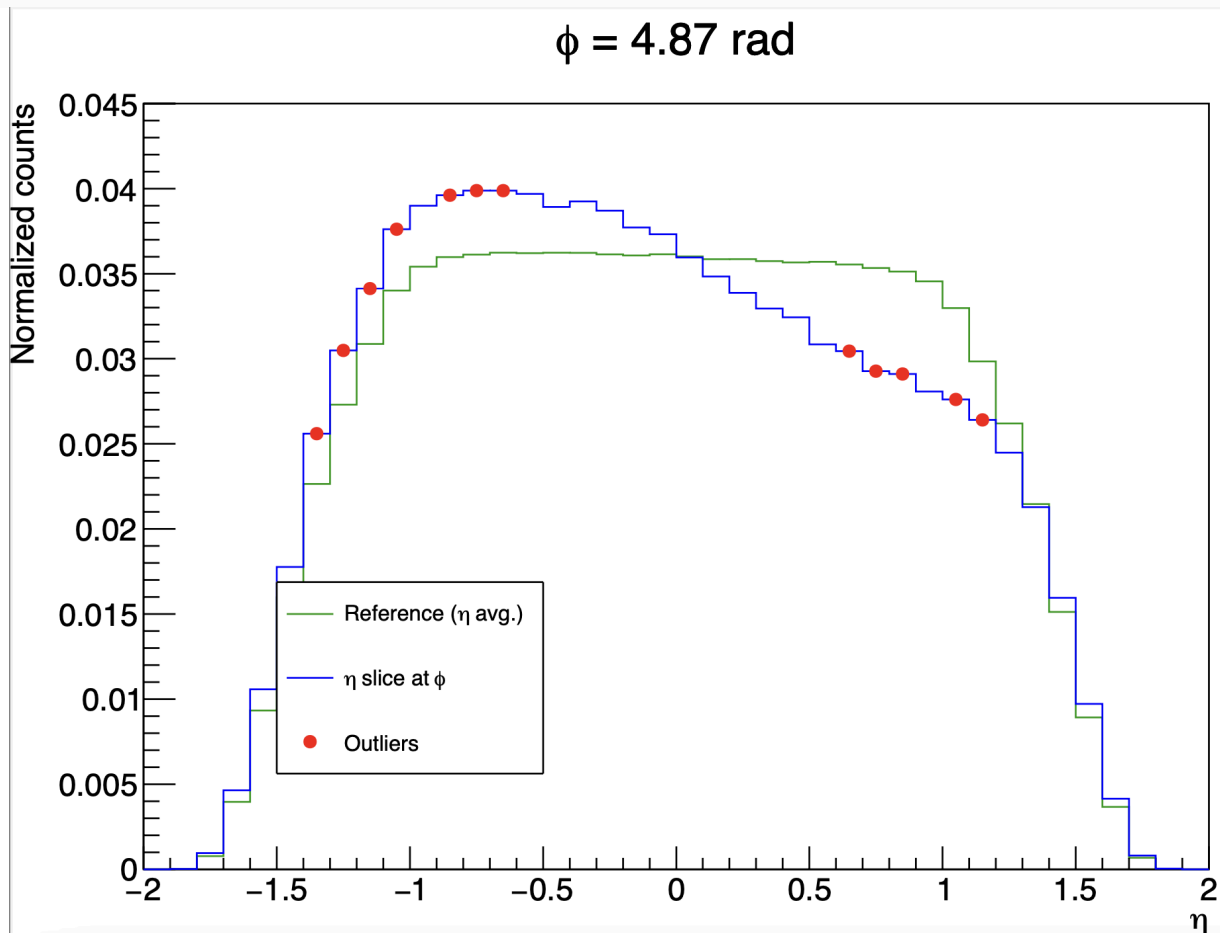
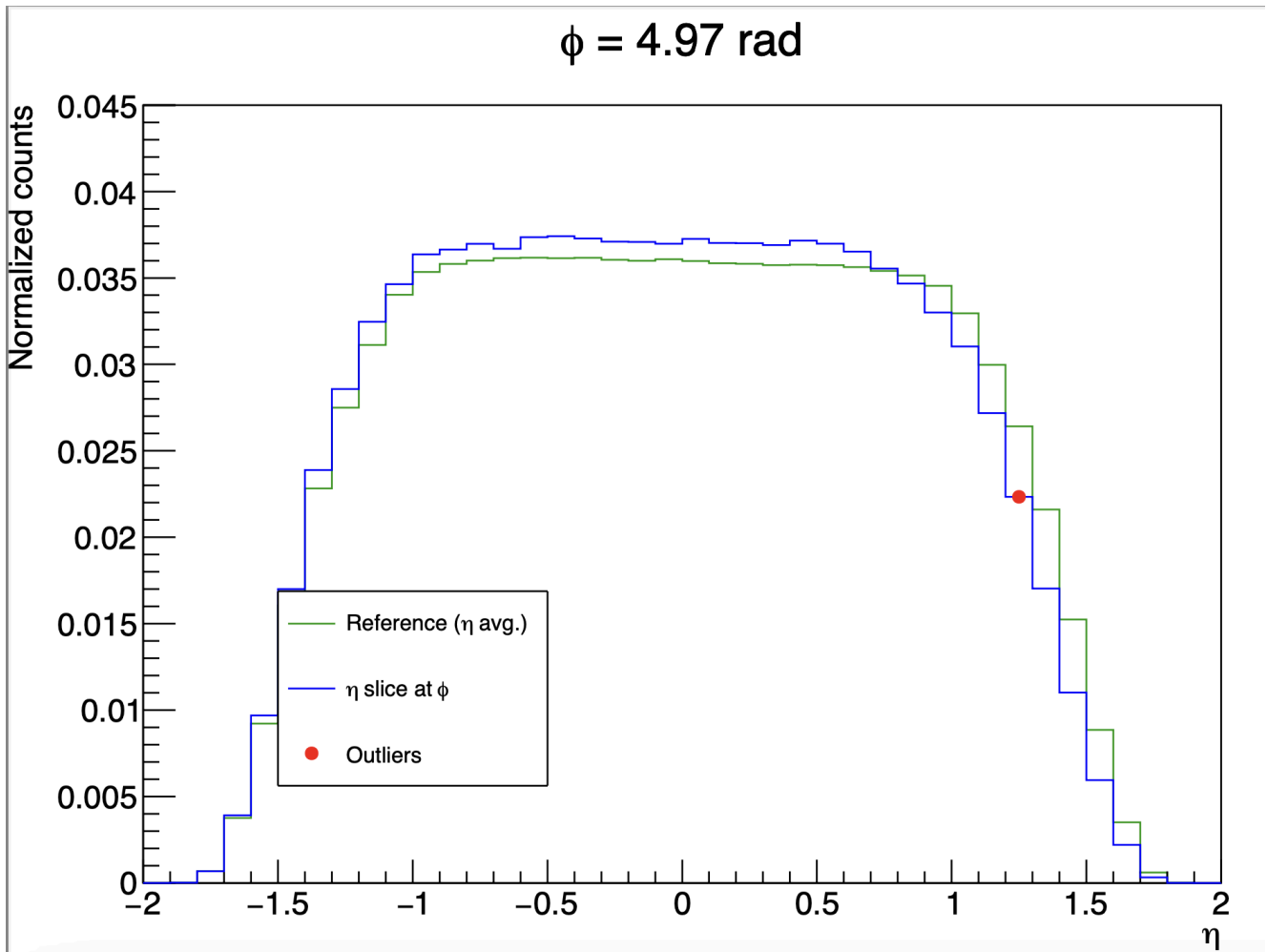
## ДОДАТОК Б

### Приклад вхідних файлів, що використовуються у програмі

ChipGeometry.txt		
0 0	0 0 16.3556 2.32807	run559362_AngularDistribution_1730351358778_phi24.pdf
0 0	1 1 16.6772 2.04846	run559362_AngularDistribution_1730351358778_phi34.pdf
0 0	2 2 16.9898 1.66363	run559362_AngularDistribution_1730351358778_phi35.pdf
0 0	3 3 17.2902 1.06477	run559362_AngularDistribution_1730351358778_phi36.pdf
0 0	4 4 17.5846 0.0123091	run559362_AngularDistribution_1730351358778_phi48.pdf
0 0	5 5 17.8581 -1.0492	run559362_AngularDistribution_1730351358778_phi49.pdf
0 0	6 6 18.1411 -1.65336	run559362_AngularDistribution_1730351384339_phi23.pdf
0 0	7 7 18.4271 -2.03953	run559362_AngularDistribution_1730351384339_phi24.pdf
0 0	8 8 18.7055 -2.3176	run559362_AngularDistribution_1730351384339_phi34.pdf
0 1	0 9 46.1165 2.32982	run559362_AngularDistribution_1730351384339_phi35.pdf
0 1	1 10 46.358 2.0482	run559362_AngularDistribution_1730351384339_phi36.pdf
0 1	2 11 46.5891 1.66202	run559362_AngularDistribution_1730351384339_phi46.pdf
0 1	3 12 46.8117 1.06234	run559362_AngularDistribution_1730351384339_phi47.pdf
0 1	4 13 47.0309 0.0144985	run559362_AngularDistribution_1730351384339_phi48.pdf
0 1	5 14 47.2386 -1.03953	run559362_AngularDistribution_1730351384339_phi49.pdf
0 1	6 15 47.4381 -1.6399	run559362_AngularDistribution_1730351564340_phi23.pdf
0 1	7 16 47.6244 -2.02284	run559362_AngularDistribution_1730351564340_phi24.pdf
0 1	8 17 47.7961 -2.29931	run559362_AngularDistribution_1730351564340_phi34.pdf
0 2	0 18 75.8437 2.33402	run559362_AngularDistribution_1730351564340_phi35.pdf
0 2	1 19 75.9905 2.05058	run559362_AngularDistribution_1730351564340_phi36.pdf
0 2	2 20 76.1173 1.66257	run559362_AngularDistribution_1730351564340_phi46.pdf
0 2	3 21 76.2292 1.06125	run559362_AngularDistribution_1730351564340_phi47.pdf
0 2	4 22 76.338 0.014986	run559362_AngularDistribution_1730351564340_phi48.pdf
0 2	5 23 76.4353 -1.03426	run559362_AngularDistribution_1730351564340_phi49.pdf
0 2	6 24 76.5367 -1.63211	run559362_AngularDistribution_1730351744340_phi23.pdf
0 2	7 25 76.6315 -2.01343	run559362_AngularDistribution_1730351744340_phi24.pdf
0 2	8 26 76.7097 -2.28949	
0 3	0 27 105.749 2.33991	
0 3	1 28 105.76 2.05552	
0 3	2 29 105.761 1.66748	
0 3	3 30 105.744 1.06553	
0 3	4 31 105.719 0.0155633	
0 3	5 32 105.681 -1.03723	
0 3	6 33 105.644 -1.63539	
0 3	7 34 105.604 -2.01631	
0 3	8 35 105.55 -2.29212	
0 4	0 36 135.459 2.34219	
0 4	1 37 135.342 2.05818	
0 4	2 38 135.227 1.67032	
0 4	3 39 135.103 1.06812	
0 4	4 40 134.976 0.0138584	
0 4	5 41 134.839 -1.04402	
0 4	6 42 134.71 -1.64388	
0 4	7 43 134.565 -2.02635	
0 4	8 44 134.404 -2.30419	
0 5	0 45 165.516 2.3431	
0 5	1 46 165.288 2.06146	
0 5	2 47 165.062 1.67531	
0 5	3 48 164.842 1.07427	
0 5	4 49 164.623 0.0139329	
0 5	5 50 164.397 -1.0537	
0 5	6 51 164.17 -1.65761	

## ДОДАТОК В

## Приклади візуалізації кутових розподілів та виявлених аномалій



## ДОДАТОК Г

## Фрагмент результатного CSV-файлу з виявленими аномальними чипами

DetectedChips\_Run559362

timestamp	phi	eta	chipID	layer	stave	module	chip_phi	chip_eta
1730369023854	4.86947	0.85	183	1	8	-169.011	0.850142	
1730369023854	4.86947	0.85	192	1	9	-146.468	0.858853	
1730369023854	4.86947	-0.85	194	1	9	-146.302	-0.847997	
1730369023854	4.86947	-0.85	203	1	10	-123.499	-0.851178	
1730369023854	4.86947	-0.85	212	1	11	-100.761	-0.853522	
1730359304367	5.07891	-0.85	221	1	12	-77.8941	-0.854003	
1730367404410	1.72788	1.25	1170	3	6	99.3918	1.25201	
1730362544381	1.72788	0.65	1185	3	6	99.3723	0.642426	
1730368664499	1.72788	-0.65	1197	3	6	99.3568	-0.642698	
1730369023854	4.86947	0.65	1792	3	12	-174.043	0.654761	
1730369023854	4.86947	0.65	1801	3	12	-178.518	0.652455	
1730369023854	4.86947	-0.65	1808	3	12	-174.083	-0.654733	
1730369023854	4.86947	-0.65	1813	3	12	-178.557	-0.652429	
1730369023854	4.86947	-1.35	1822	3	12	-174.128	-1.34823	
1730369023854	4.86947	-1.35	1827	3	12	-178.605	-1.3446	
1730369023854	3.61283	1.25	1835	3	12	-166.523	1.24773	
1730369023854	3.61283	1.25	1842	3	12	-170.901	1.25308	
1730369023854	4.86947	0.65	1857	3	12	-170.932	0.643537	
1730369023854	4.86947	-0.65	1869	3	12	-170.969	-0.643471	
1730369023854	4.86947	-1.25	1877	3	12	-166.617	-1.24794	
1730369023854	4.86947	-1.25	1884	3	12	-170.997	-1.25325	
1730369023854	4.86947	0.65	1904	3	13	-158.989	0.655453	
1730369023854	4.86947	0.65	1913	3	13	-163.47	0.653495	
1730369023854	4.86947	-0.65	1920	3	13	-159.026	-0.655152	
1730369023854	4.86947	-0.65	1925	3	13	-163.509	-0.653069	
1730369023854	4.86947	-1.35	1934	3	13	-159.056	-1.34917	
1730369023854	4.86947	-1.35	1939	3	13	-163.533	-1.34624	
1730369023854	4.76475	1.25	1947	3	13	-151.451	1.24886	