

# OOP Project Report

Jacopo Altieri

Febbraio 2023

# Indice

<b>1</b>	<b>Operazioni preliminari</b>	<b>2</b>
<b>2</b>	<b>Claro</b>	<b>3</b>
<b>3</b>	<b>SiPM</b>	<b>5</b>

# Capitolo 1

## Operazioni preliminari

Prima di eseguire i codici, scaricare per ciascun progetto i file *\*\_class.py*, *\*\_main.py*, *requirements.txt* in una cartella.

Eseguire poi il comando:

```
pip install -r requirements.txt
```

In modo da avere tutti i pacchetti necessari per il corretto funzionamento del codice.

# Capitolo 2

## Claro

Il codice per il progetto Claro si divide in due parti:

- `claro_class.py` : Contiene le definizioni delle classi e dei loro metodi
- `claro_main.py` : Implementa le classi ed i metodi e produce diversi output in base al tipo di file in ingresso. Tutti i parametri che l'utente vuole cambiare vanno modificati da questo file.

Per utilizzare questo codice, tramite terminale muoversi fino alla cartella contenente i file class e main. Eseguire poi il comando:

```
$ python .\claro_main.py <input_file/input_directory>
```

Dove il primo argomento è il percorso (assoluto o relativo) di un singolo file Claro, di una lista di file o di una directory.

Se si sta analizzando un singolo file, il programma legge i dati, esegue un fit lineare nella zona di transizione ed un fit di erf su tutti i dati. Printa poi i risultati a schermo assieme ad un plot con i dati ed i relativi fit, come il seguente.

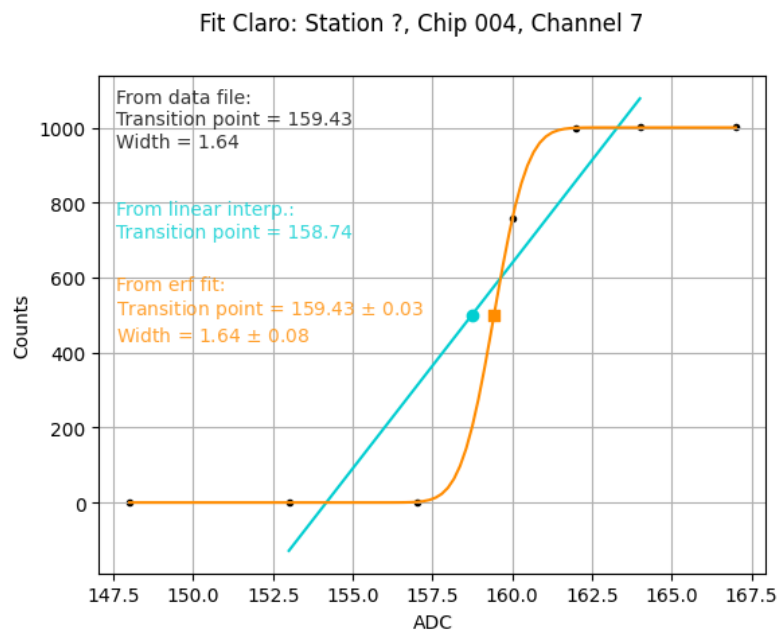


Figura 2.1: Plot relativo alle analisi di un singolo Claro file

Se l'input è invece una directory, il programma crea prima una lista di tutti i file Claro presenti, per poi creare le liste di file buoni e "cattivi" (ovvero tutti quelli con dati mancanti o corrotti). La lista di file buoni viene poi analizzata, prendendo per ciascun file le informazioni relative al chip, i dati ed infine i valori ottenuti tramite il fit della erf. I risultati di tale analisi sono stampati in un file .csv assieme ai metadati del chip, con l'opzione di scartare i file per cui il fit non converge (tipicamente i file in cui ci sono pochi punti nella zona di transizione.)

Vengono infine creati gli istogrammi dei punti di transizione letti dal file e di quelli misurati, oltre all'istogramma delle discrepanze.

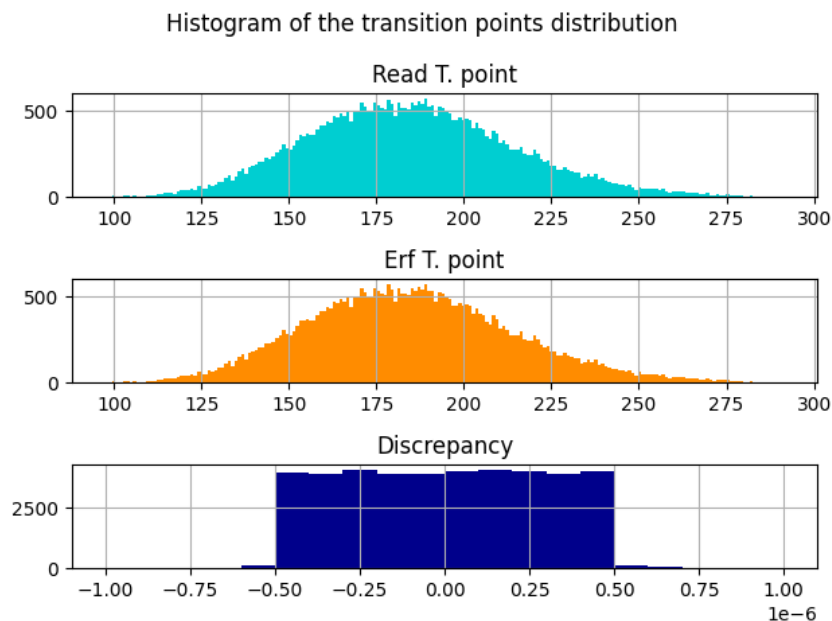


Figura 2.2: Istogrammi dei punti di transizione.

# Capitolo 3

## SiPM

Il codice per l'analisi dei SiPM si divide in due parti:

- `SiPM_class.py` : Contiene le definizioni delle classi e dei loro metodi
- `SiPM_main.py` : Implementa le classi ed i metodi e produce diversi output in base al tipo di file in ingresso. Tutti i parametri che l'utente vuole cambiare vanno modificati da questo file.

Per utilizzare questo codice, muoversi da terminale fino alla cartella contenente i file `class` e `main`. Eseguire poi il comando:

```
$ python .\SiPM_main.py <input_file/input_directory>
```

Dove il primo argomento è il percorso (assoluto o relativo) di un singolo file ARDU o di una directory.

Se si sta analizzando un singolo file, il programma legge i dati ed, in base alla direzione, esegue uno dei seguenti fit:

- Forward: Sulla parte lineare della curva fitta una retta con coefficiente angolare  $m$ . il valore  $1000/m$  rappresenta la Resistenza di quenching di tale SiPM. Questo processo viene eseguito per ogni SiPM, ottenendo un file `.csv` con tutte le resistenze (30) ed un PDF con i plot, come mostrato in figura:

Forward IV curve: SiPM 0.0

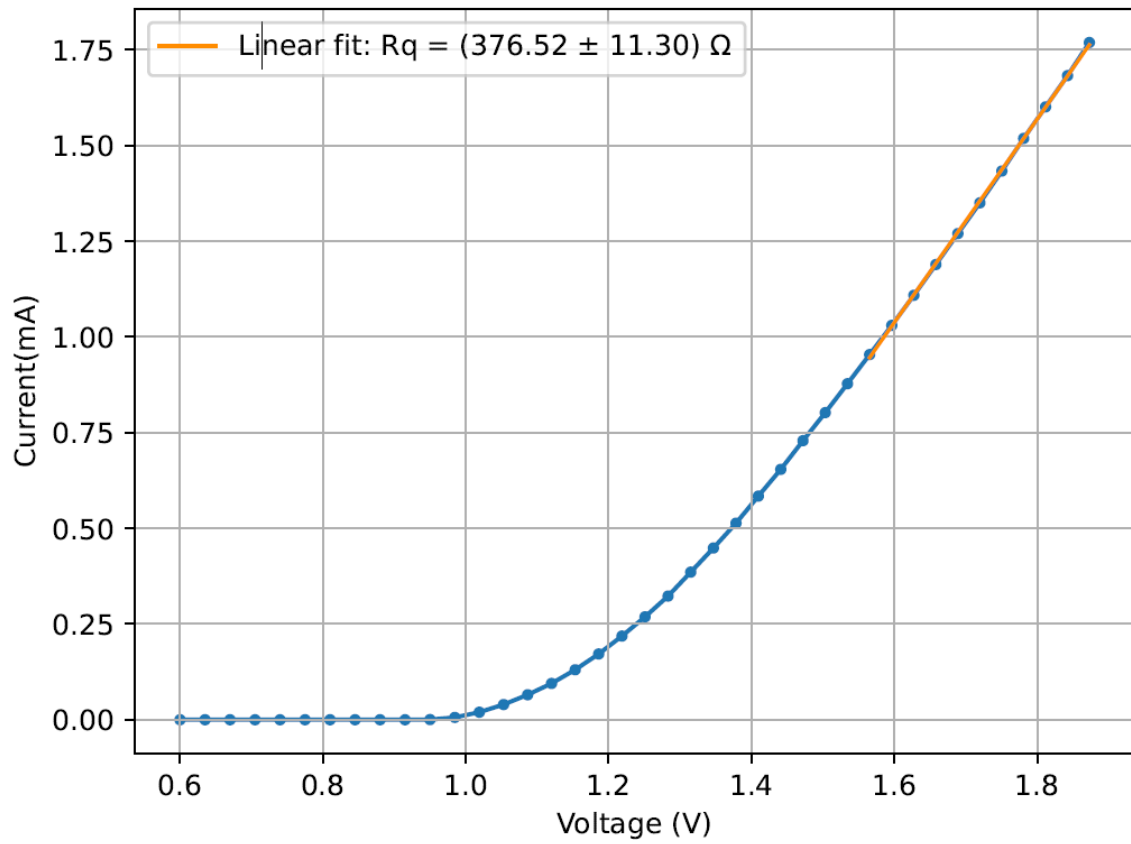


Figura 3.1: Plot della IV curve in polarizzazione diretta.

- Reverse: Calcola la derivata normalizzata  $\frac{1}{I} \frac{dI}{dV}$  ed esegue un fit polinomiale di grado 5 su di essa. Sul massimo del polinomio fitta poi una gaussiana, la cui media rappresenta il valore del Voltaggio di Breakdown. Anche in questo caso si ottengono in output un file .csv ed un PDF.

## Reverse IV curve: SiPM 0.0

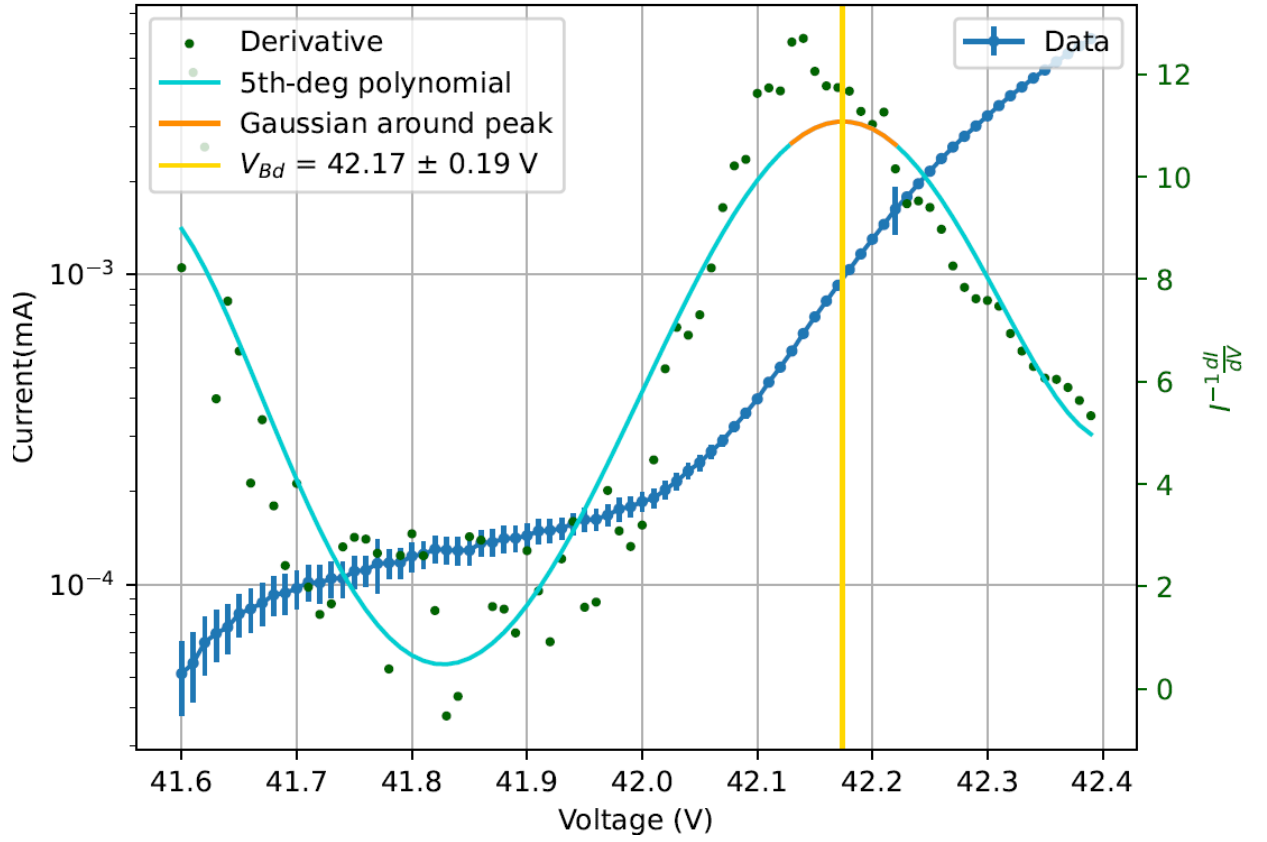


Figura 3.2: Plot della IV curve in polarizzazione inversa.

Se si sta analizzando una cartella, viene creata una nuova directory chiamata "results" in cui sono inserite le analisi di ciascun file (forward e reverse), mantenendo anche la separazione fra temperature e date diverse. Oltre agli output delle analisi singole sono anche prodotti gli istogrammi delle distribuzioni delle resistenze di quenching e delle tensioni di breakdown, nonché quelli del confronto fra le due misure di Aprile e fra le due misure ad azoto liquido.



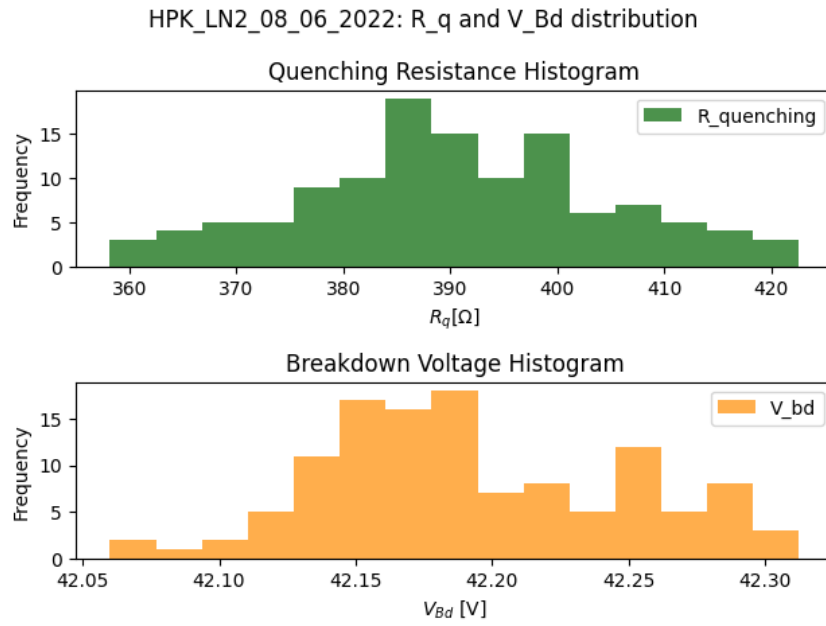


Figura 3.3: Istogrammi delle Resistenze di quenching e delle tensioni di Breakdown per un set di misure.

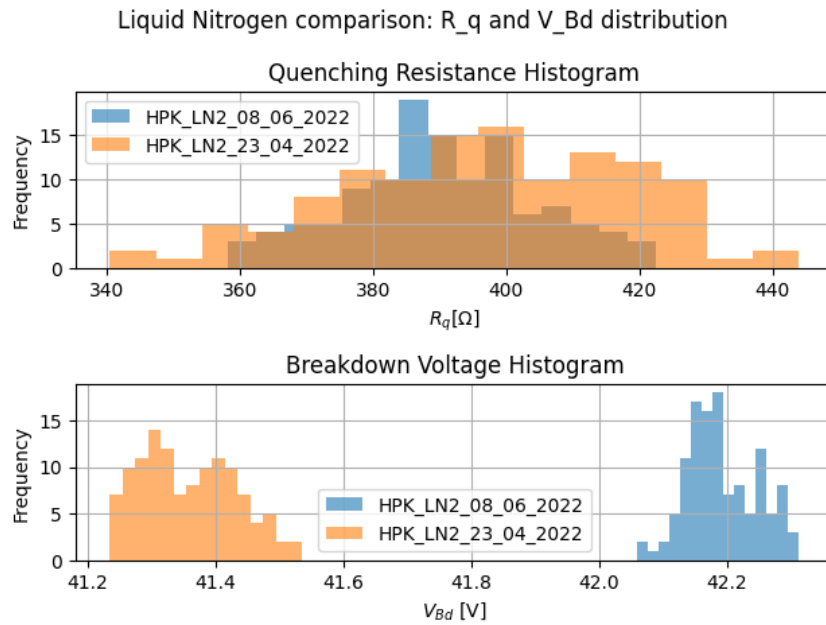


Figura 3.4: Istogrammi comparativi delle misure effettuate ad azoto liquido.