Ecogas resistive measurements April 2025

M. Verzeroli

ECOGAS@GIF++ COLLABORATION

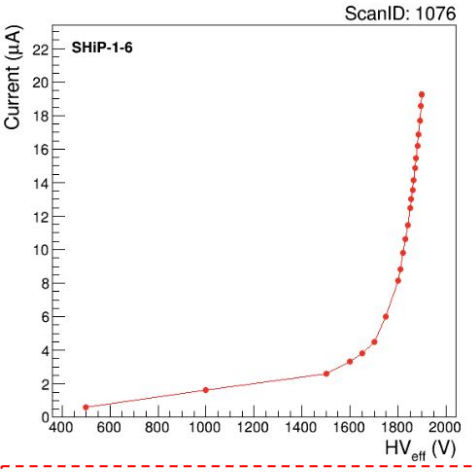
EP-DT
Detector Technologies

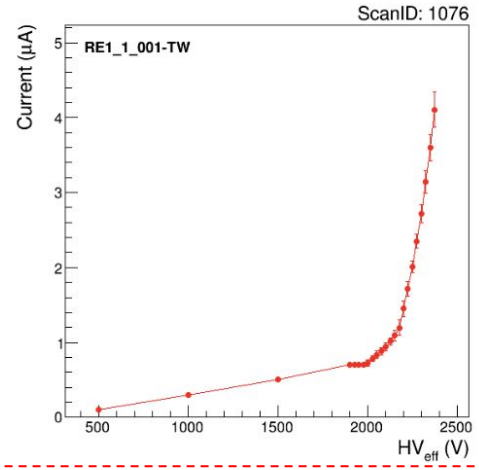
Resistive measurements

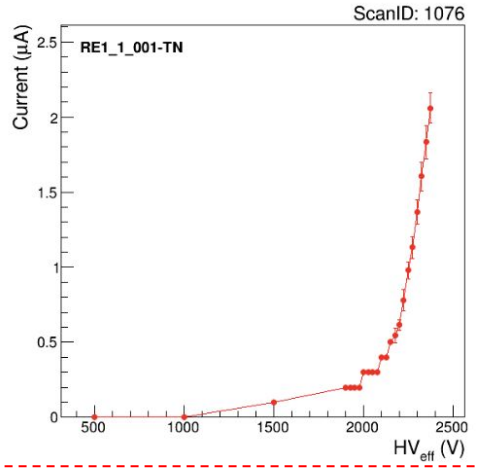
Two scans performed:

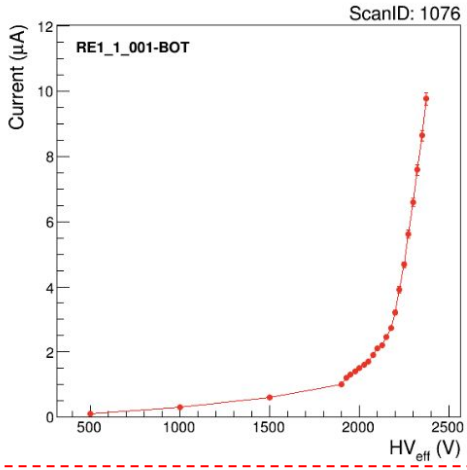
- 1076: with the standard HV point and with all the detectors
- 1077: with more points, only for Kodel, ALICE and EP-DT

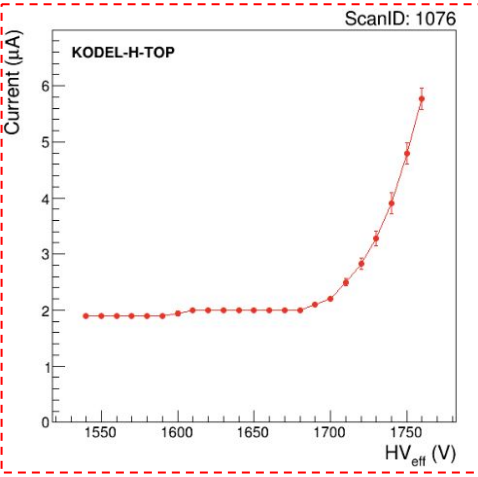
Resistive measurements

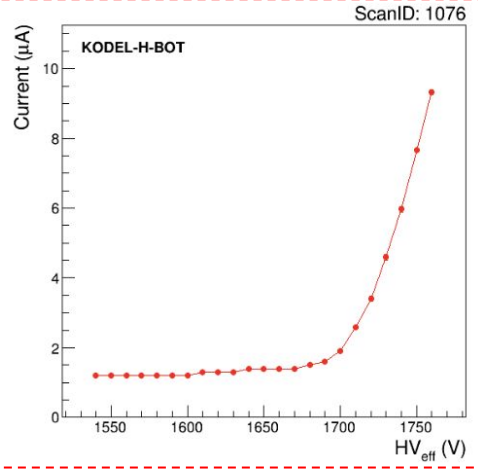


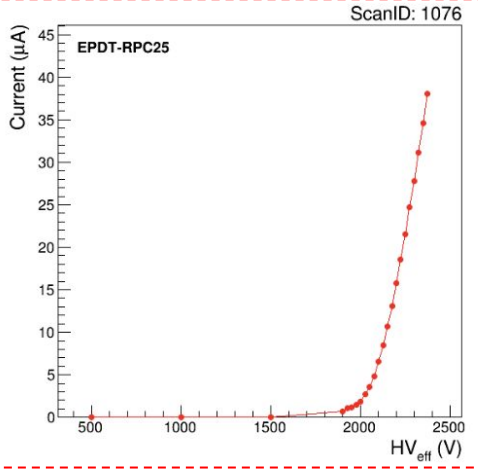


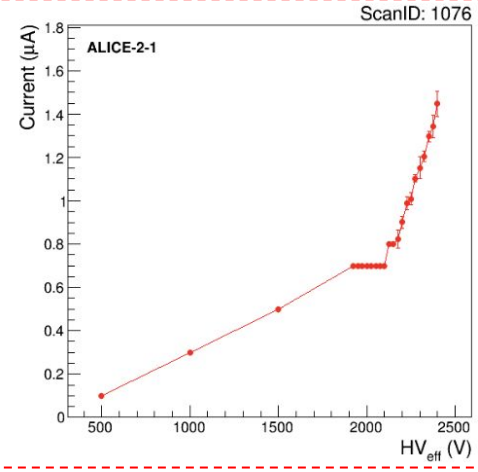




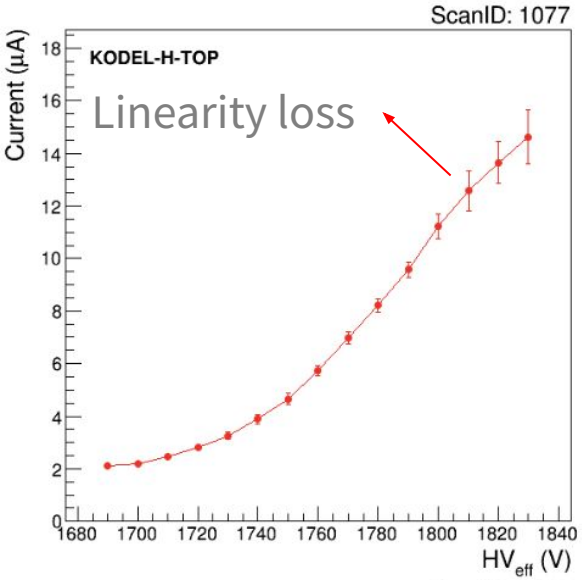


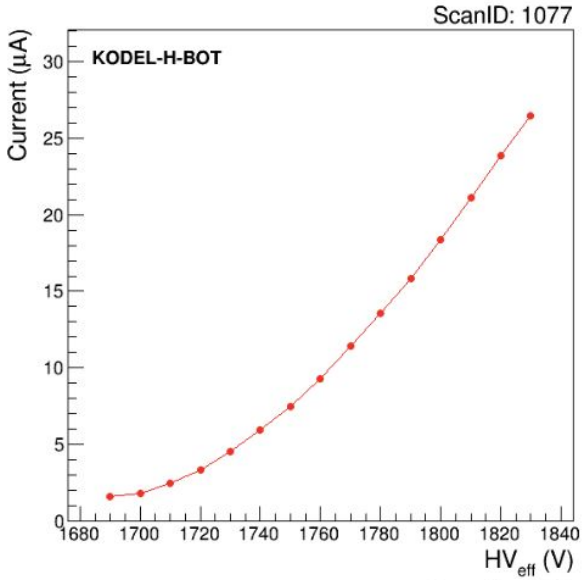


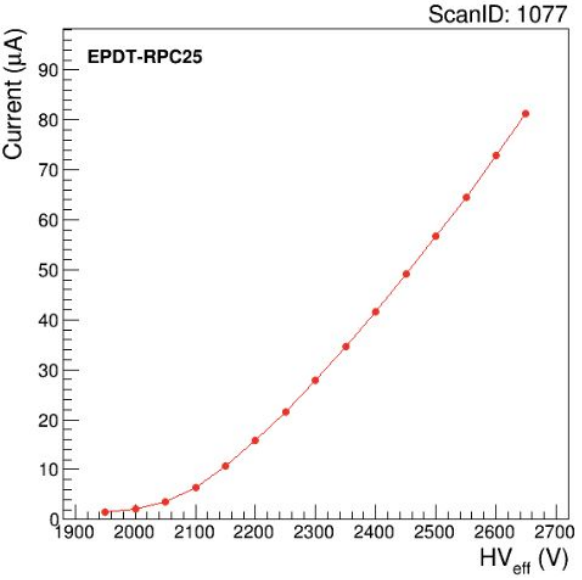


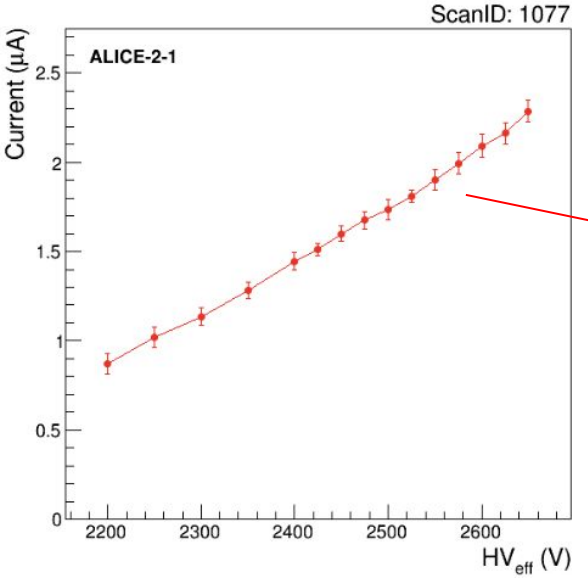


Resistive measurements



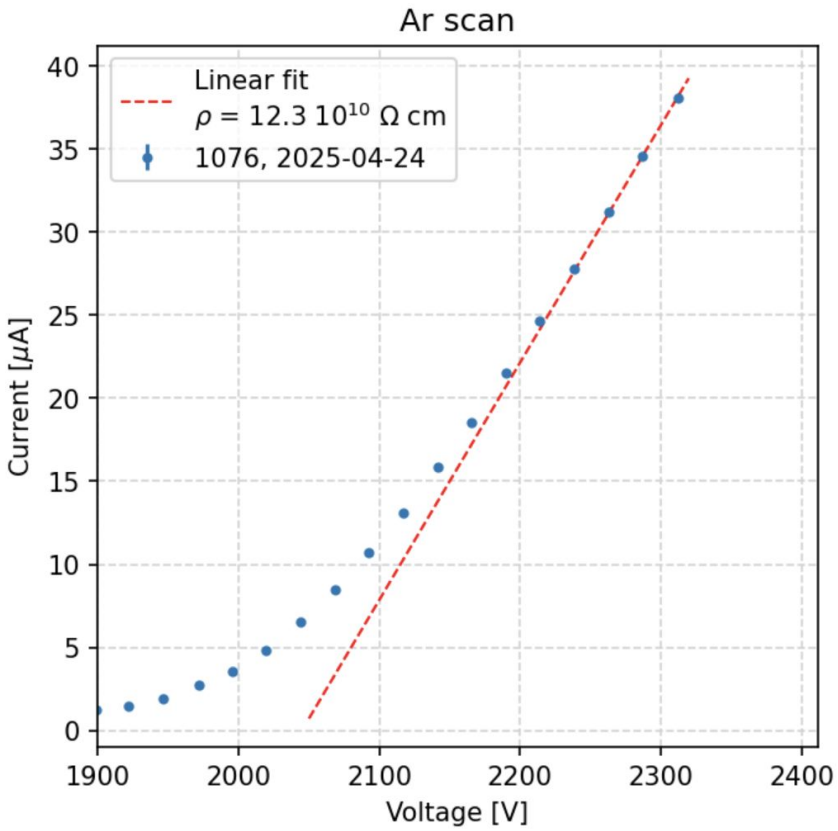


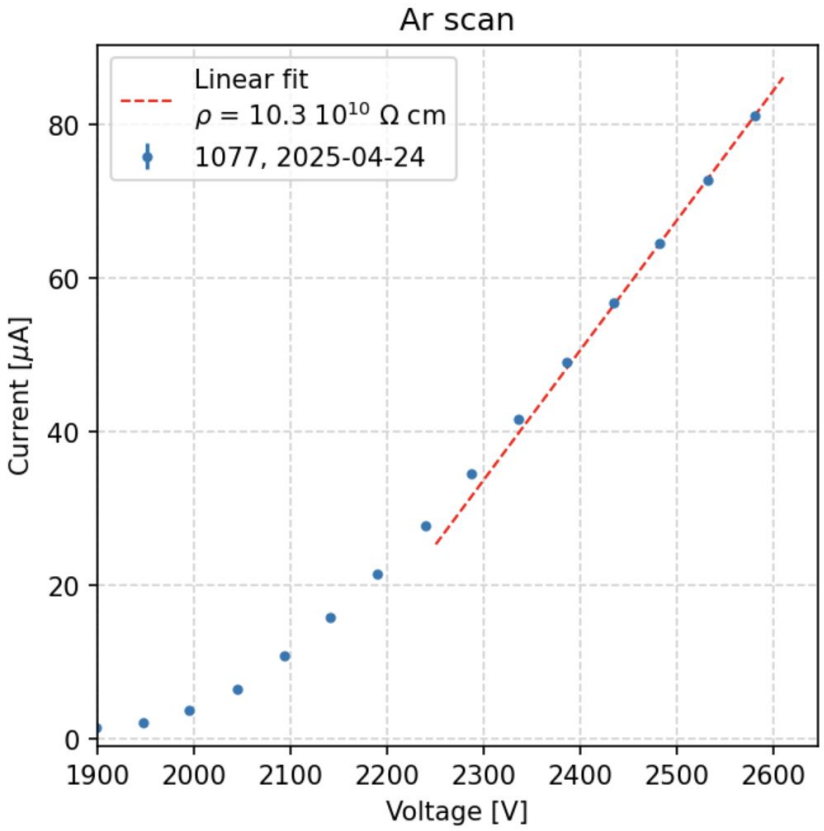




No big change even if with more point

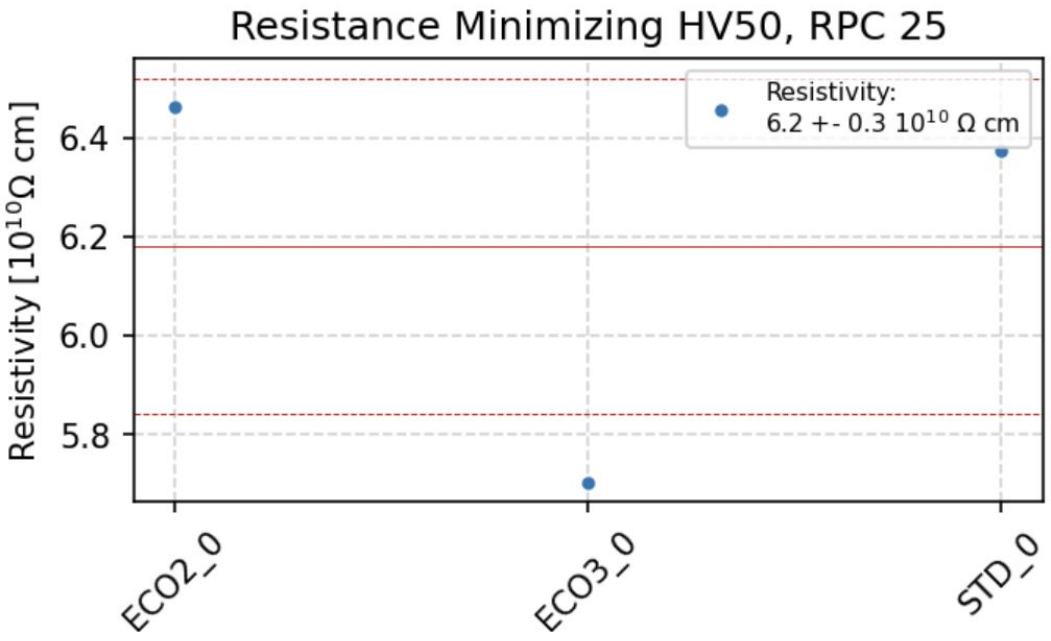
EP-DT Ar





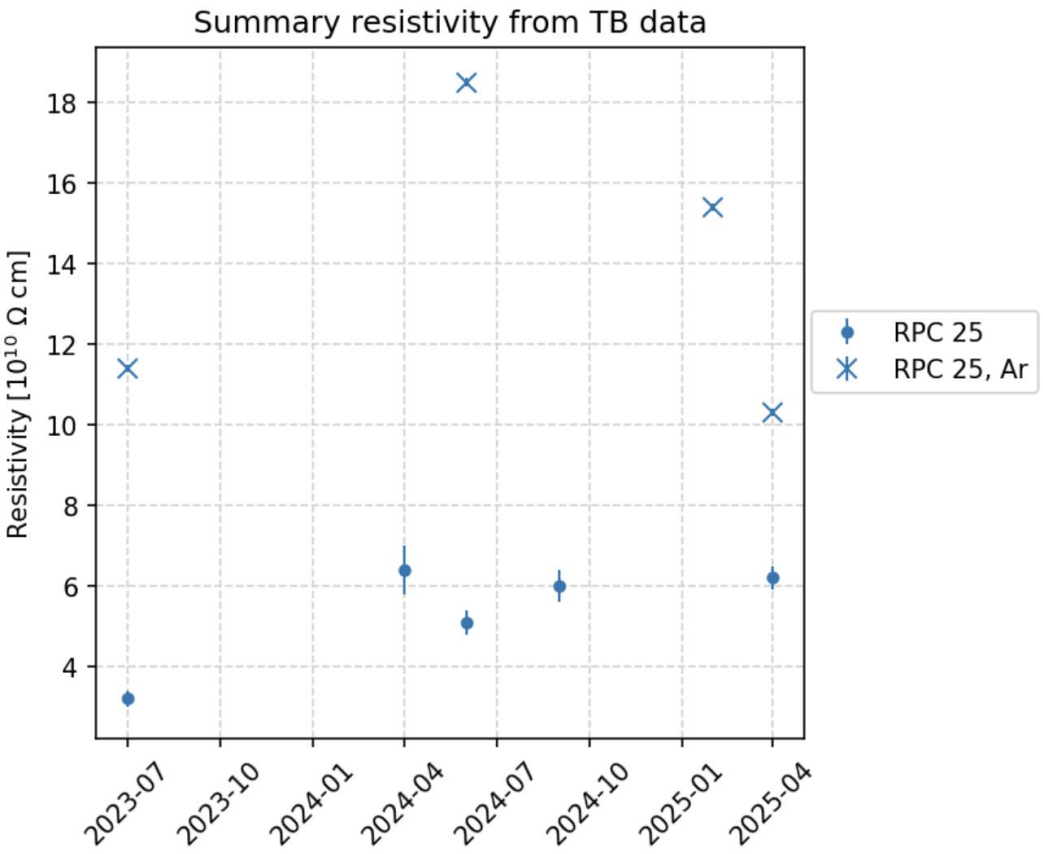
- Considered just the last 3 points for the linear fit;
- 2x10^{10} Ohm cm decrease if higher HV values are considered

EP-DT Test Beam



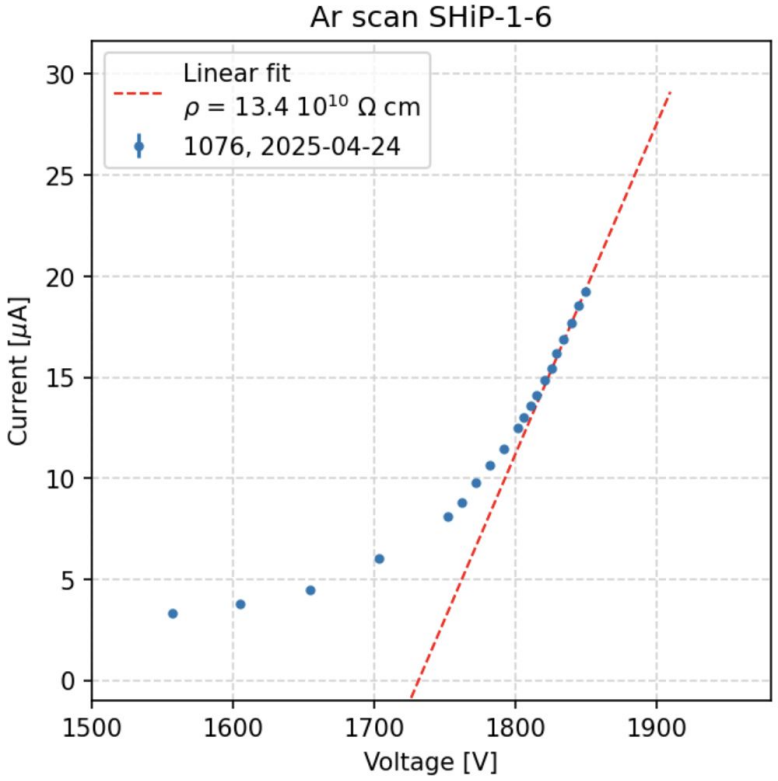
- Considering the minimization of the difference of HV50 from the abs test beam efficiency curve, found a lower values of the resistivity

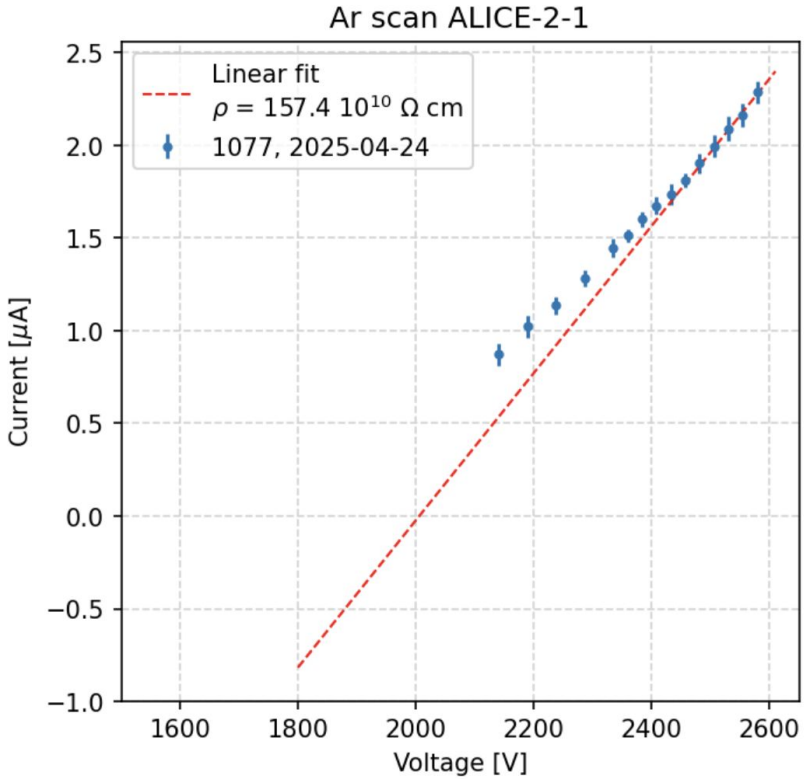
EP-DT All measurements



- In the latest measurements, the values are closer to the TB value;
- During the Ar scan, it is difficult to find stable results since they are strongly dependent on the data selection criteria used for the fit;
- Considering the TB data, the resistivity has remained similar since January 2024;
- Considering the Ar data, the resistivity shows a significant decrease - not reflected in TB detector performances

SHiP & ALICE



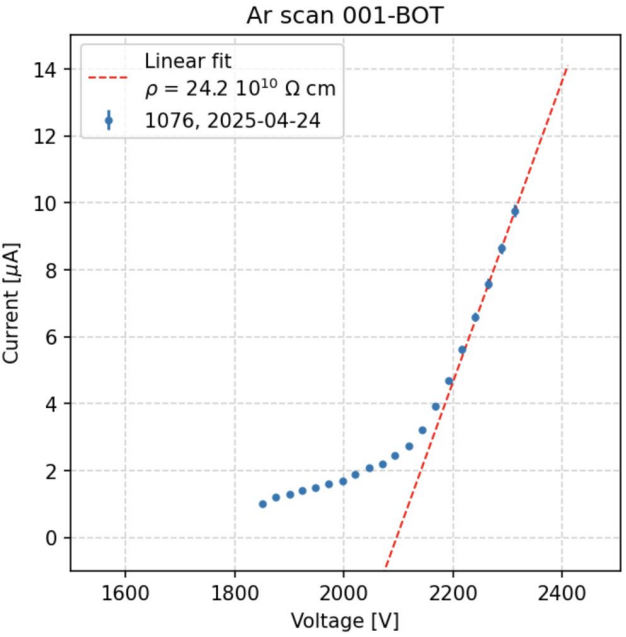


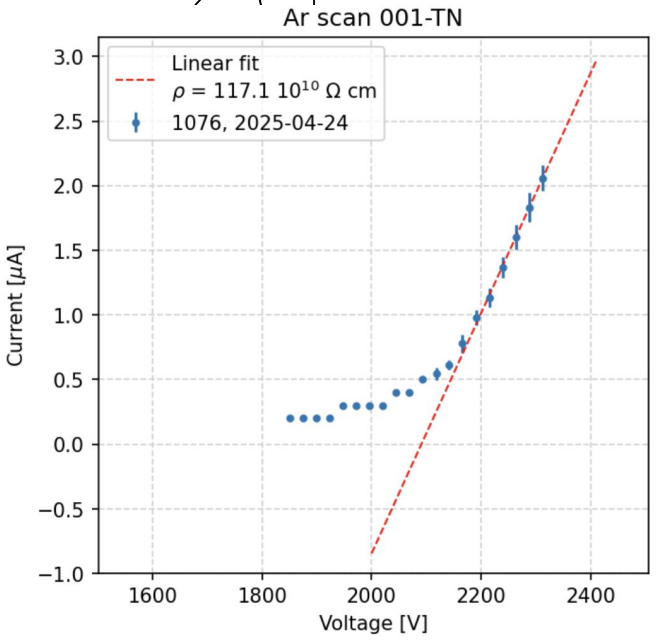
Area: 7000 cm^2
El. Thickness: 16 mm

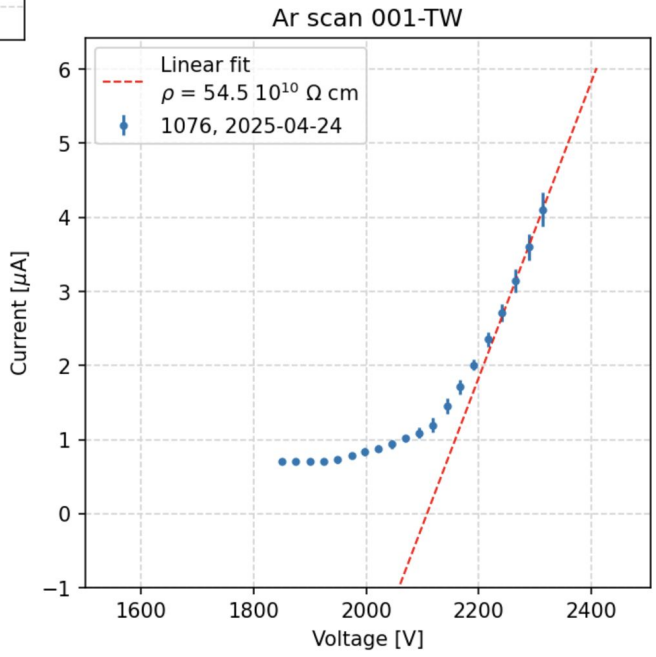
Area: 2500 cm^2
El. Thickness: 2 mm

CMS RPC

EP-DT
Detector Technologies

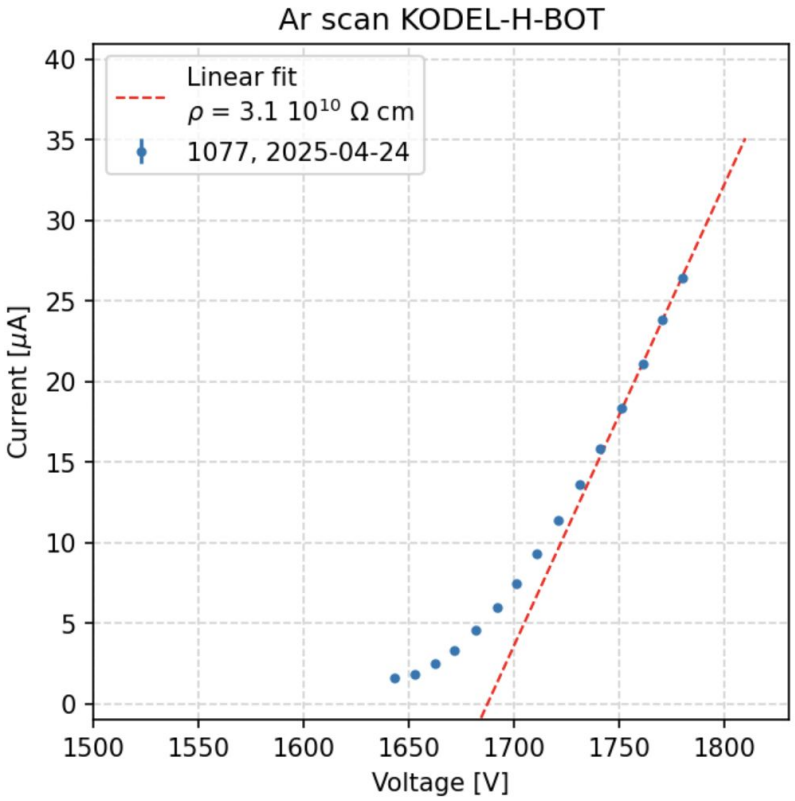


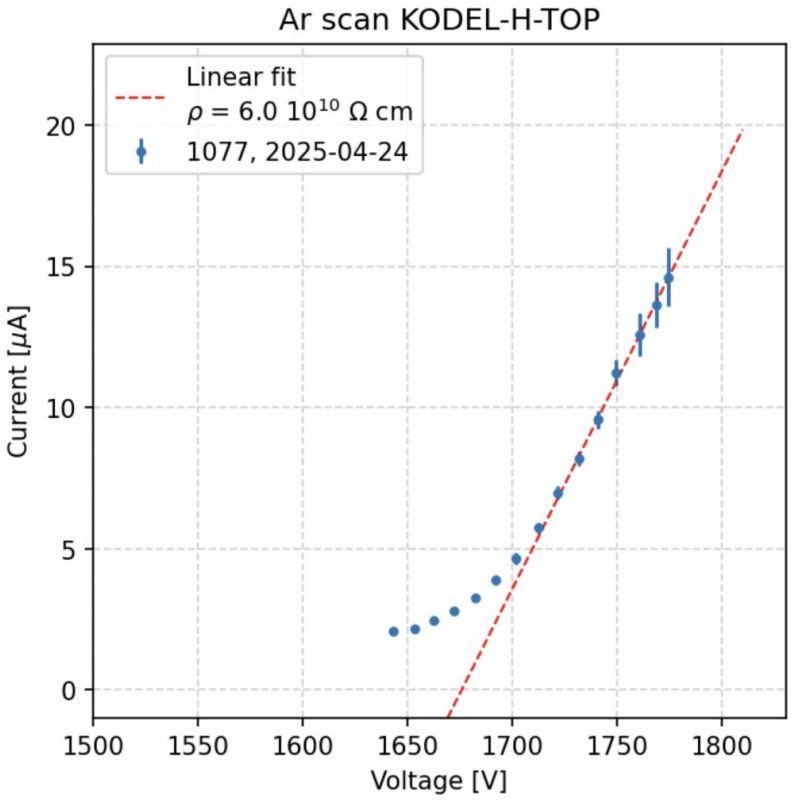




Area: 4350 cm^2
El. Thickness: 2 mm

KODEL H





Area: 2500 cm^2
El. Thickness: 14 mm

Backup

Backup

New functions that allows to perform the data analysis in a fast, easy way with the possibility to export them in a dedicated library

data_loading

It loads all the information and setup parameter, such as temperature,

pressure...

```
def data_loading(path, rpc, mix_dict, client, run_number = 0):
    '''
    function to load all the data:

    INPUT:
    - path: path of the test beam run
    - rpc: rpc number for the selected run
    - mix_dict: dictionary to convert the gas composition into the plot label
    - client: client needed to download infos from InfluxDB
    - run_number: number of the run for fixed gas mixture

    OUTPUT:
    a dataframe with the following columns:

    - run_name: name of the hv scan
    - rpc: number of the rpc
    - voltage: voltage with no correction
    - voltage_app: voltage corrected
    - efficiency: detector's efficiency indata
    - currents_standard_mean_indata: currents indata
    - currents_beam_mean_indata: beam currents indata
    - currents_standard_std_indata: currents standard deviation indata
    - effmax: maximum efficiency
    - gamma: sigmoid fit parameter
    - hv50: sigmoid fit parameter
    - working point: working point of the detector
    - up_abs: filter values as str, if source is off, up_abs = 'off'
    - mix: gas mixture composition
    - mix_label: gas mixture plot label
    - init_time: time of the beginning of the scan -> needed to download the environmental params
    - dew point: humidity of the gas system
    - o2: oxygen of the gas system
    - p_bunker: pressure inside the bunker
    - p_outside: pressure outside the bunker
    - t_detector: temperature on the detector frame
    - t_bunker: temperature inside the bunker
    - t_outside: temperature outside the bunker
    - t_gas_room: temperature inside the gas room

    for the next step, this df need to be merged with all the scan for fixed source, gas mixture and rpc and different abs
    '''
```

Backup

currents_fit

```
def physics_fit(Vgap, a, V_T, sigma_T):
    '''
    Function to fit all the hvscan plot.
    '''
    term1 = (sigma_T / np.sqrt(2 * np.pi)) * np.exp(-((Vgap - V_T) ** 2) / (2 * sigma_T ** 2))
    term2 = 0.5 * (Vgap - V_T)
    erf_term = 0.5 * (1 + erf((Vgap - V_T) / (np.sqrt(2) * sigma_T)))
    return a * (term1 + term2 * erf_term)
```

It performs the current fit over the selected run, adding the fit parameter to the dataframe

resistivity_minimization_hv50

It consider the sigmoid fit for each dataset, and compute the minimization, as a function of R, of the following function over all the filter in the dataset wrt the source off scan considering HV50:

```
def objective(R):
    return np.abs(np.sum(V_off - V_fil + R * I_fil))
```

The resistivity parameter is then added to the dataframe, as 'resistance_minimization_hv50'