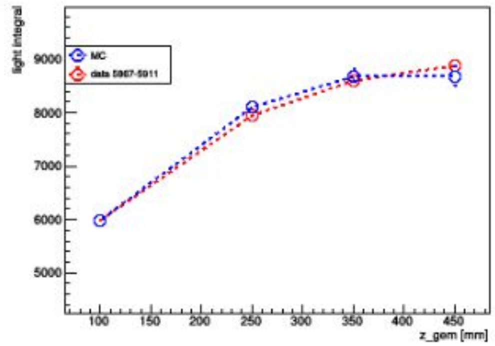
Status of digitization

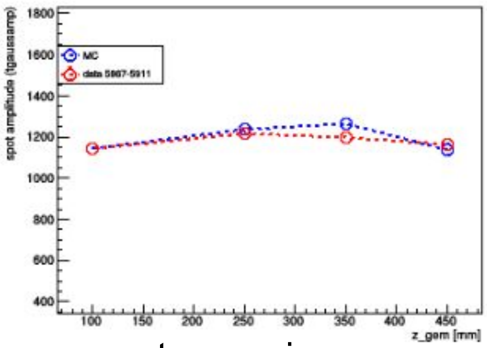
16-10-2023

Pietro Meloni, Fabrizio, Petrucci, Davide Pinci

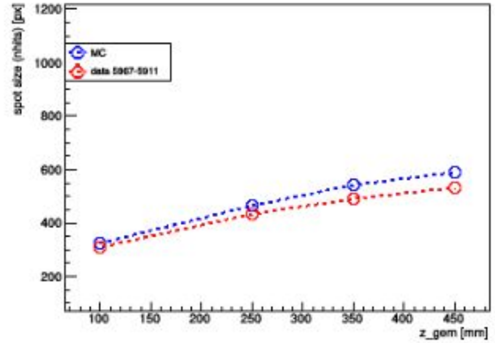
In 2022 we optimized the digitization on these LNF runs: 5867-5911

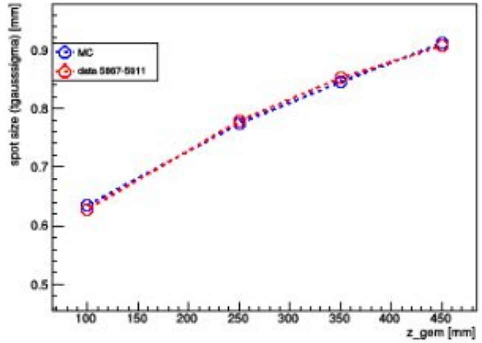
integral





tgausssigma





Parameters:

events per run= 100 events
detector dimensions = 346cm x 346cm
pixels = 2304 x 2304
pedestal = run 5861

beta= 1e-5
A= 1.52
z_vox_dim=0.1 mm
x_vox_dim= 346/2304 mm
y_vox_dim=346/2304 mm
abs_len= 1400mm
z_gem = 10, 25, 35, 45 cm
GEM1_HV= 440V
GEM2_HV= 440V
GEM3_HV= 440V
random_z = 0

diff_const_sigma0T= 0.1225 mm^2 (350 µm)
diff_coeff_T= 0.013225 [mm/sqrt(cm)]^2 (115 µm/sqrt(cm))

diff_const_sigma0L= 0.0676 mm^2 (260 µm)
diff_coeff_L= 0.00978 [mm/sqrt(cm)]^2 (99 µm/sqrt(cm))

ion_pot = 0.0462 keV
photons_per_el = 0.07
counts_per_photon = 2.,

But GEM gain varies a lot

So far, we have used the GEM gains measured at LNF.

But now we know that the gain is highly dependent on the humidity, and other environmental factor

```
## fit from Fernando Amaro's single GEM gain measurement
GEM1_gain = 0.0347 * np.exp((0.0209) * opt.GEM1_HV)
GEM2_gain = 0.0347 * np.exp((0.0209) * opt.GEM2_HV)
GEM3_gain = 0.0347 * np.exp((0.0209) * opt.GEM3_HV)
print("GEM1_gain = %d" % GEM1_gain)
print("GEM2_gain = %d" % GEM2_gain)
print("GEM3_gain = %d" % GEM3_gain)

## dividing Fernando's to Francesco&Karolina's single GEM gain measurement
extraction_eff_GEM1 = 0.87319885 * np.exp(-0.0020000000 * opt.GEM1_HV)
extraction_eff_GEM2 = 0.87319885 * np.exp(-0.0020000000 * opt.GEM2_HV)
extraction_eff_GEM3 = 0.87319885 * np.exp(-0.0020000000 * opt.GEM3_HV)
print("extraction eff GEM1 = %f" % extraction_eff_GEM1)
print("extraction eff GEM2 = %f" % extraction_eff_GEM2)
print("extraction eff GEM3 = %f" % extraction_eff_GEM3)
```

Let's keep the gain free and check if we can reproduce LNGS data adjusting its value.

We do a scan in z and gain values and we'll find the best gain for a given calibration run at LNGS.

```python
## Fit from Fernando Amaro's single GEM gain measurement
GEM1_gain = 0.0347 * np.exp((0.0209) * opt.GEM1_HV)
GEM2_gain = 0.0347 * np.exp((0.0209) * opt.GEM2_HV)
GEM3_gain = 0.0347 * np.exp((0.0209) * opt.GEM3_HV)
print("GEM1_gain = %d" % GEM1_gain)
print("GEM2_gain = %d" % GEM2_gain)
print("GEM3_gain = %d" % GEM3_gain)

## dividing Fernando's to Francesco&Karolina's single GEM gain measurement
extraction_eff_GEM1 = 0.87319885 * np.exp(-0.0020000000 * opt.GEM1_HV)
extraction_eff_GEM2 = 0.87319885 * np.exp(-0.0020000000 * opt.GEM2_HV)
extraction_eff_GEM3 = 0.87319885 * np.exp(-0.0020000000 * opt.GEM3_HV)
print("extraction eff GEM1 = %f" % extraction_eff_GEM1)
print("extraction eff GEM2 = %f" % extraction_eff_GEM2)
print("extraction eff GEM3 = %f" % extraction_eff_GEM3)
```

```python
GEM1_gain = opt.GEM1_gain
GEM2_gain = opt.GEM2_gain
GEM3_gain = opt.GEM3_gain
print("GEM1_gain = %d" % GEM1_gain)
print("GEM2_gain = %d" % GEM2_gain)
print("GEM3_gain = %d" % GEM3_gain)

## dividing Fernando's to Francesco&Karolina's single GEM gain measurement
extraction_eff_GEM1 = 0.87319885 * np.exp(-0.0020000000 * opt.GEM1_HV)
extraction_eff_GEM2 = 0.87319885 * np.exp(-0.0020000000 * opt.GEM2_HV)
extraction_eff_GEM3 = 0.87319885 * np.exp(-0.0020000000 * opt.GEM3_HV)
print("extraction eff GEM1 = %f" % extraction_eff_GEM1)
print("extraction eff GEM2 = %f" % extraction_eff_GEM2)
print("extraction eff GEM3 = %f" % extraction_eff_GEM3)
```

No dependency on HV anymore

The best gain will be the one that minimize the root-mean-square difference between data and MC.

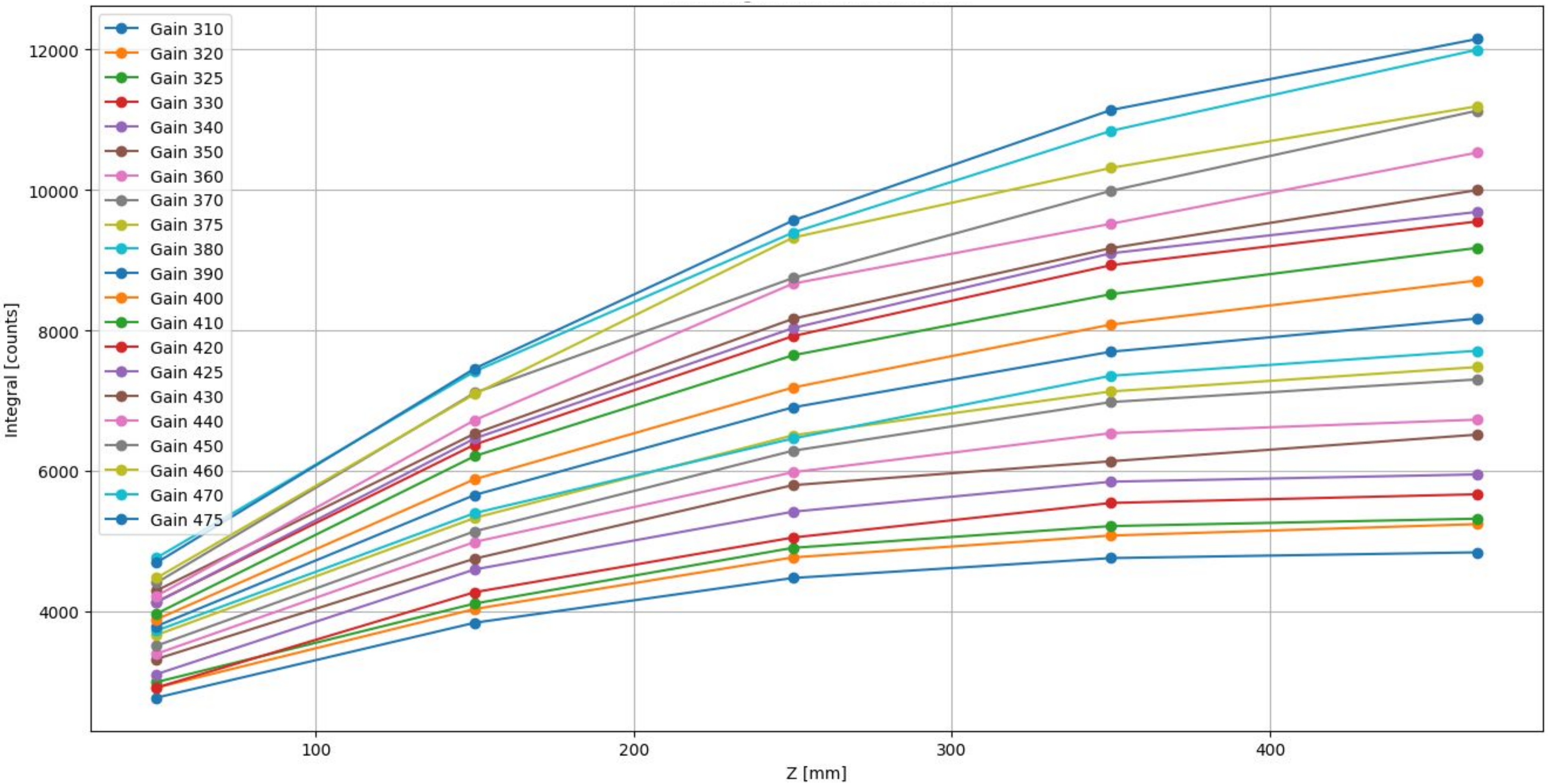
Two important notes:

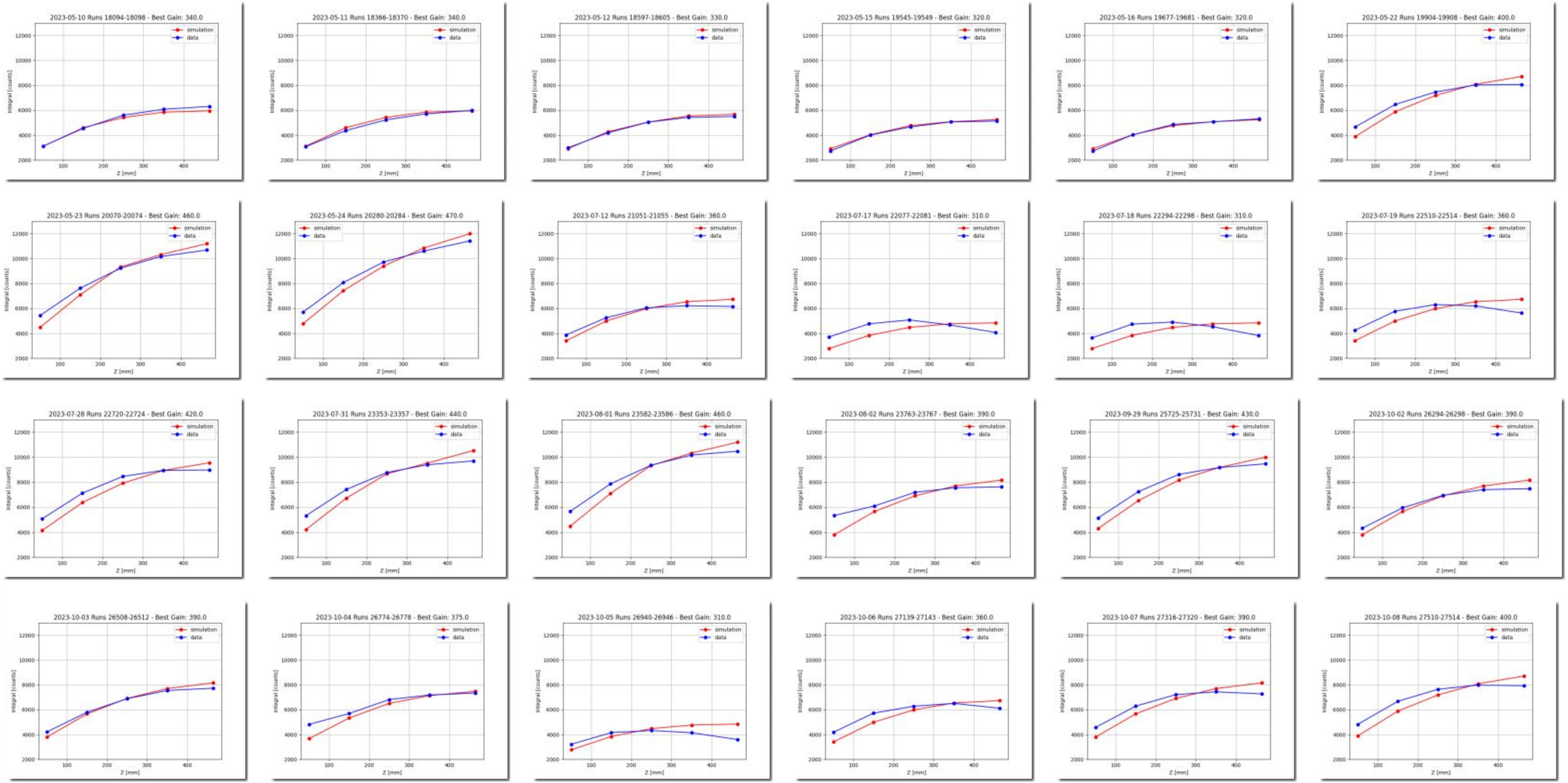
2. Saturation parameter A (normalization factor) is now set to 1. In fact, the A parameter was set to 1.52 to reproduce LNF data (since we fixed the gain).

$$G_3 = A \sum_{voxels} \frac{e^{\alpha \Delta V}}{1 + \beta n_0 e^{\alpha \Delta V}}$$

2. Now we are setting the drift field to 800 V/cm (still using diffusion parameters simulate in Garfield by Francesco Renga). Previously, we were using 1000 V/cm, but the working point underground is at 800 V/cm.

NOTE: **gain(440 V) = 342** and **gain(420 V) = 225** (measured at LNF)

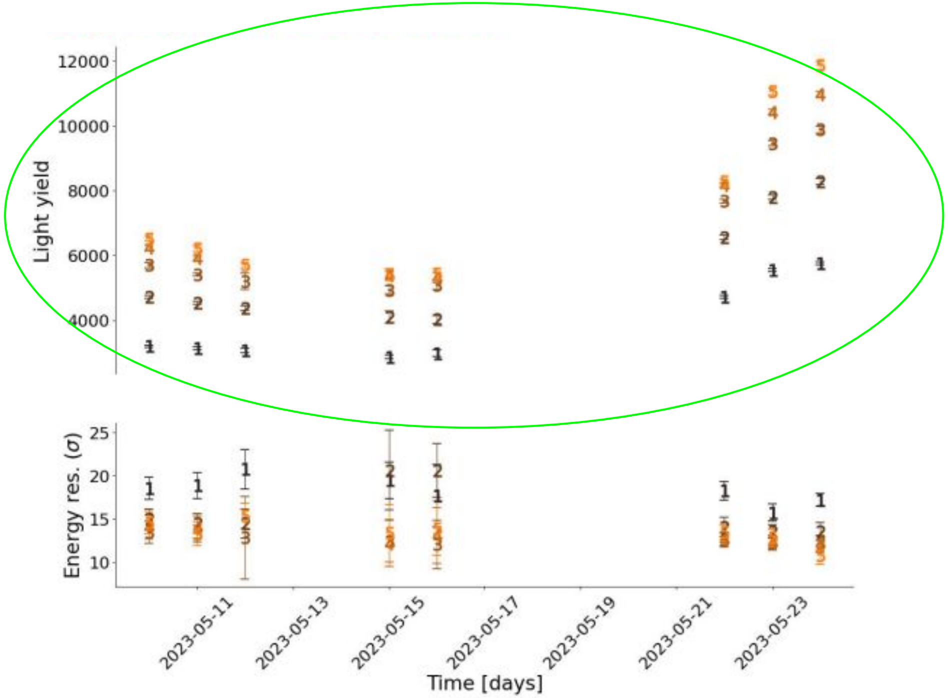




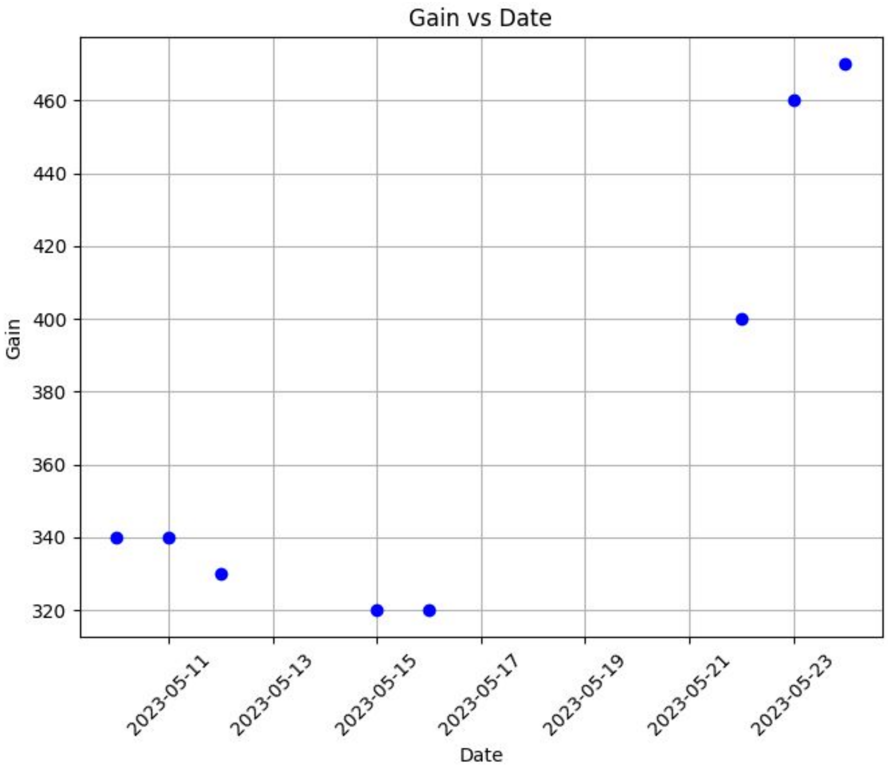
The simulation reproduces really well data in May,then it worsens a bit: especially at high z where MC is higher
Maybe we have a different attenuation due to the change in humidity? (I'll come back to this)

In May 2023 the best gain parameter has the same trend as the LY in data

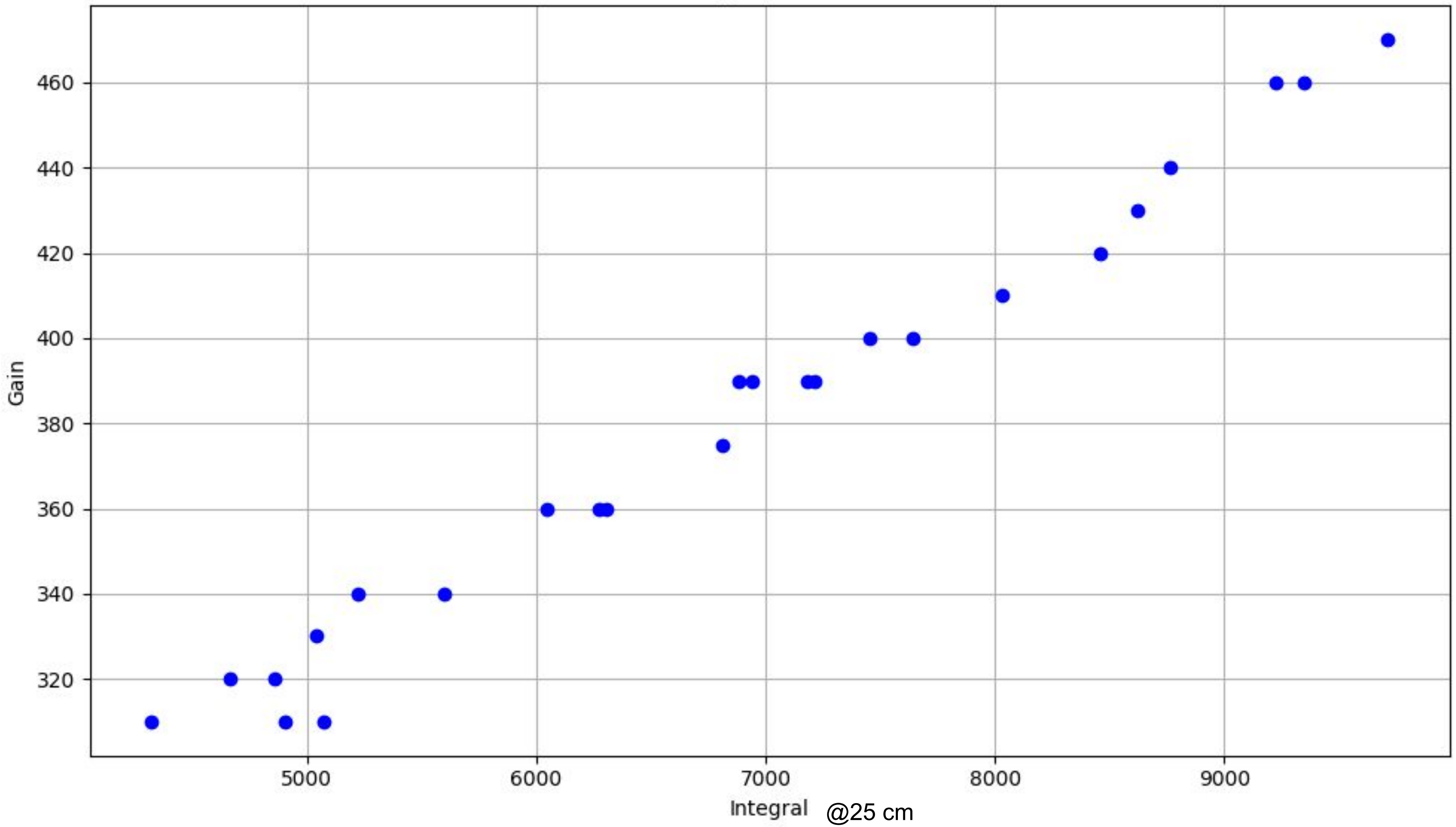
Rita Roque analysis (Coimbra 2023)



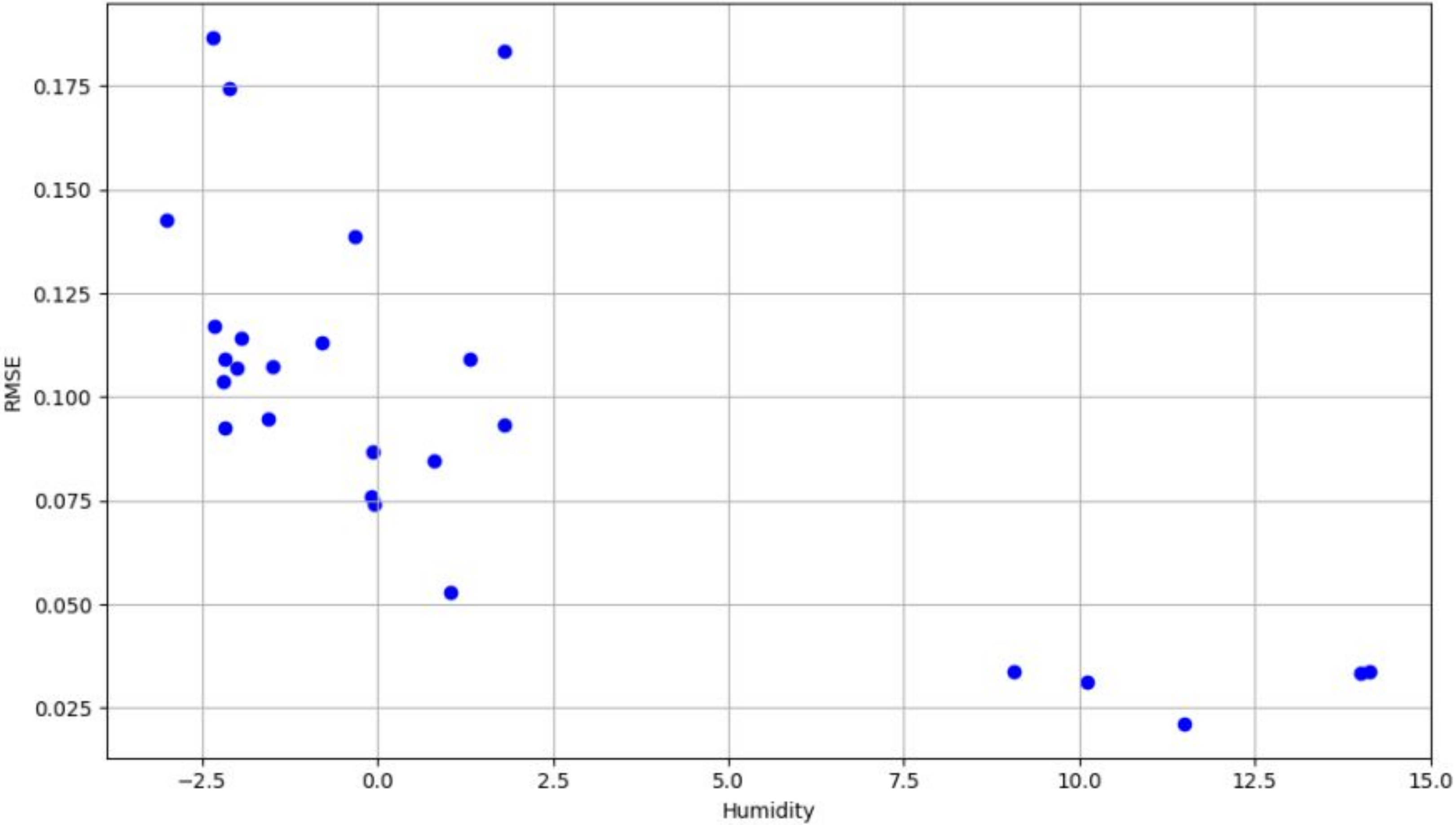
Best gain parameters for digitization



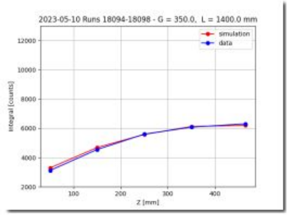
As expected, the integral @ 25 cm (data) increase with the best gain paramter

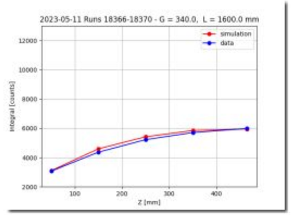


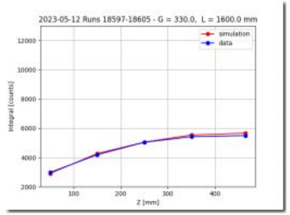
Strangely, the simulation appears to better match the data in low humidity environment.
Is this true because we tuned the digitization on data with high humidity at LNF?

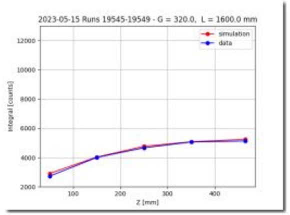


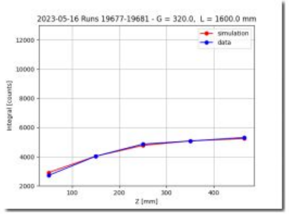
Let’s try to find the best pair (gain, abs_length)...
(to improve simulation at high z)

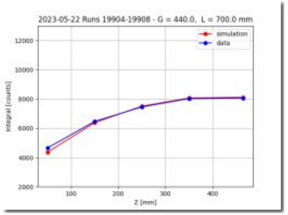

2023-05-10 Runs 18094-18098 - G = 350.0, L = 1400.0 mm
simulation
data
Integral [counts]
Z [mm]

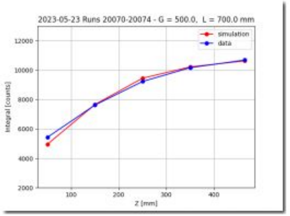

2023-05-11 Runs 18366-18370 - G = 340.0, L = 1600.0 mm
simulation
data
Integral [counts]
Z [mm]

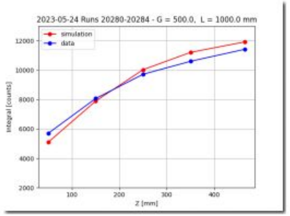

2023-05-12 Runs 18597-18605 - G = 330.0, L = 1600.0 mm
simulation
data
Integral [counts]
Z [mm]

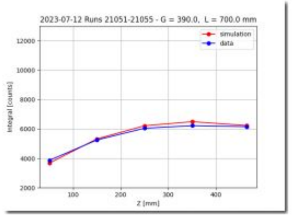

2023-05-15 Runs 19545-19549 - G = 320.0, L = 1600.0 mm
simulation
data
Integral [counts]
Z [mm]

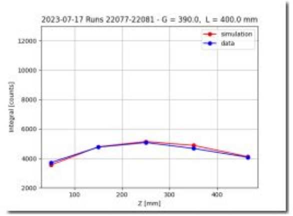

2023-05-16 Runs 19677-19681 - G = 320.0, L = 1600.0 mm
simulation
data
Integral [counts]
Z [mm]

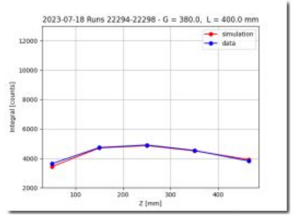

2023-05-22 Runs 19904-19908 - G = 440.0, L = 700.0 mm
simulation
data
Integral [counts]
Z [mm]

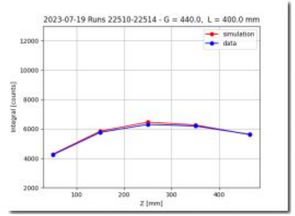

2023-05-23 Runs 20070-20074 - G = 500.0, L = 700.0 mm
simulation
data
Integral [counts]
Z [mm]

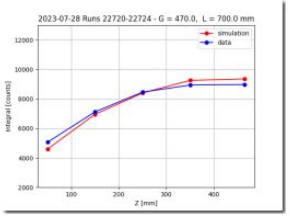

2023-05-24 Runs 20280-20284 - G = 500.0, L = 1000.0 mm
simulation
data
Integral [counts]
Z [mm]

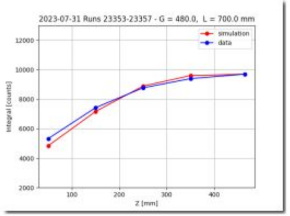

2023-07-12 Runs 21051-21055 - G = 390.0, L = 700.0 mm
simulation
data
Integral [counts]
Z [mm]

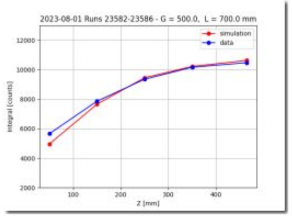

2023-07-17 Runs 22077-22081 - G = 390.0, L = 400.0 mm
simulation
data
Integral [counts]
Z [mm]

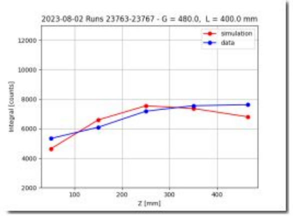

2023-07-18 Runs 22294-22298 - G = 380.0, L = 400.0 mm
simulation
data
Integral [counts]
Z [mm]

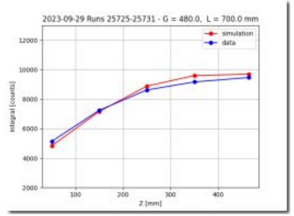

2023-07-19 Runs 22510-22514 - G = 440.0, L = 400.0 mm
simulation
data
Integral [counts]
Z [mm]

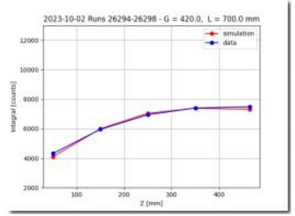

2023-07-28 Runs 22720-22724 - G = 470.0, L = 700.0 mm
simulation
data
Integral [counts]
Z [mm]

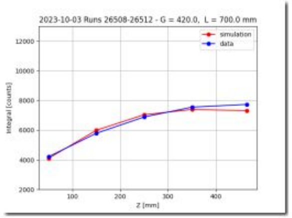

2023-07-31 Runs 23353-23357 - G = 480.0, L = 700.0 mm
simulation
data
Integral [counts]
Z [mm]

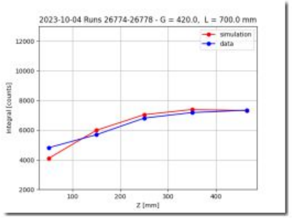

2023-08-01 Runs 23582-23586 - G = 500.0, L = 700.0 mm
simulation
data
Integral [counts]
Z [mm]

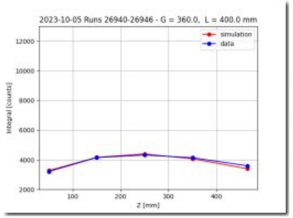

2023-08-02 Runs 23763-23767 - G = 480.0, L = 400.0 mm
simulation
data
Integral [counts]
Z [mm]

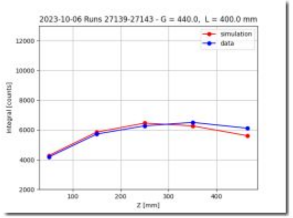

2023-09-29 Runs 25725-25731 - G = 480.0, L = 700.0 mm
simulation
data
Integral [counts]
Z [mm]

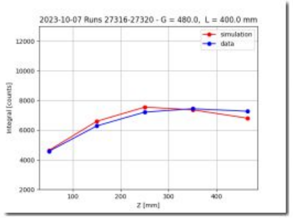

2023-10-02 Runs 26294-26298 - G = 420.0, L = 700.0 mm
simulation
data
Integral [counts]
Z [mm]

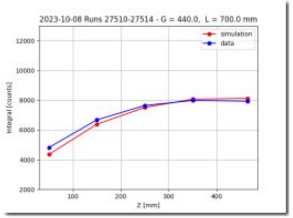

2023-10-03 Runs 26508-26512 - G = 420.0, L = 700.0 mm
simulation
data
Integral [counts]
Z [mm]


2023-10-04 Runs 26774-26778 - G = 420.0, L = 700.0 mm
simulation
data
Integral [counts]
Z [mm]

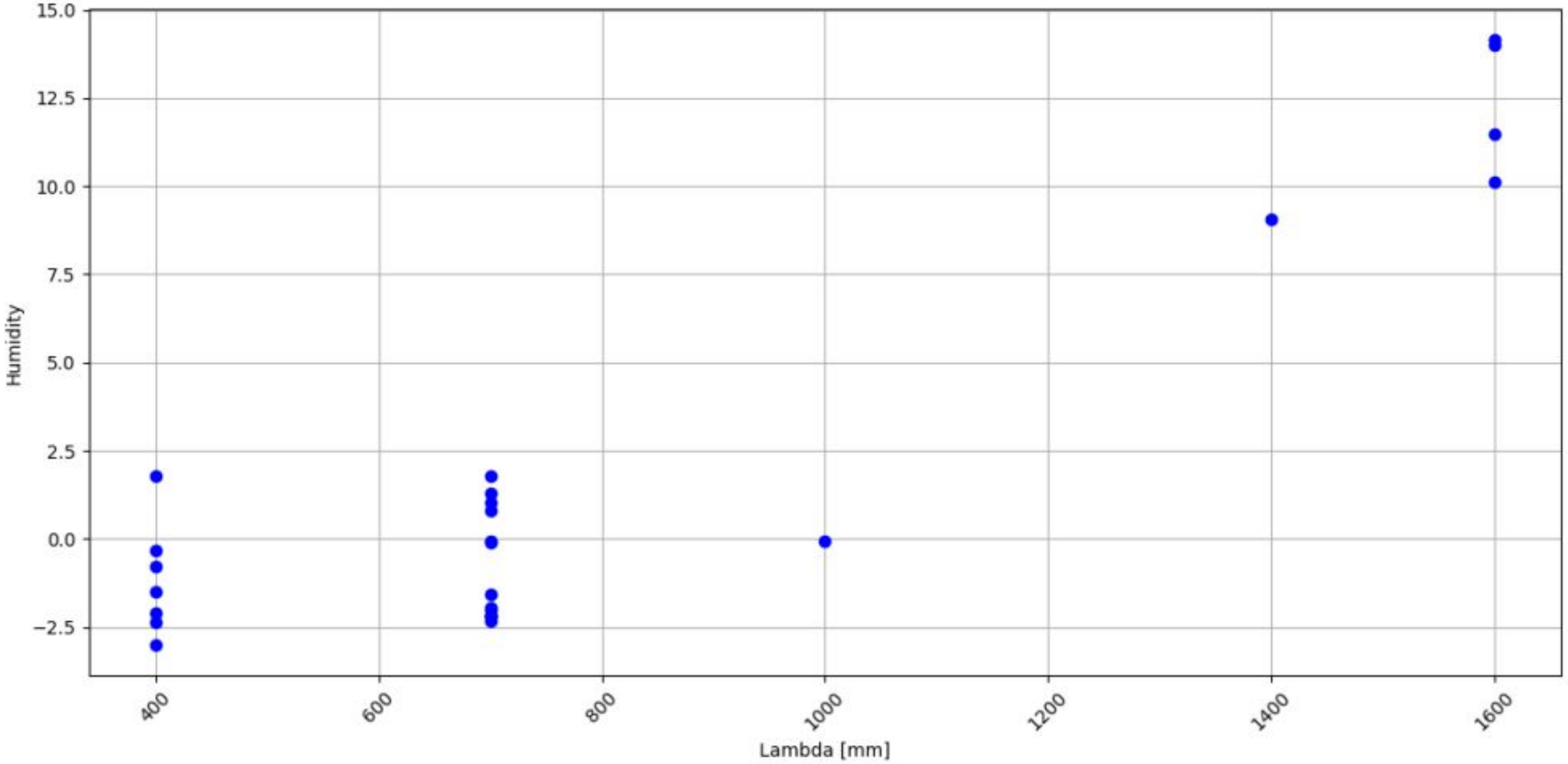

2023-10-05 Runs 26940-26946 - G = 360.0, L = 400.0 mm
simulation
data
Integral [counts]
Z [mm]


2023-10-06 Runs 27139-27143 - G = 440.0, L = 400.0 mm
simulation
data
Integral [counts]
Z [mm]


2023-10-07 Runs 27316-27320 - G = 480.0, L = 400.0 mm
simulation
data
Integral [counts]
Z [mm]


2023-10-08 Runs 27510-27514 - G = 440.0, L = 700.0 mm
simulation
data
Integral [counts]
Z [mm]

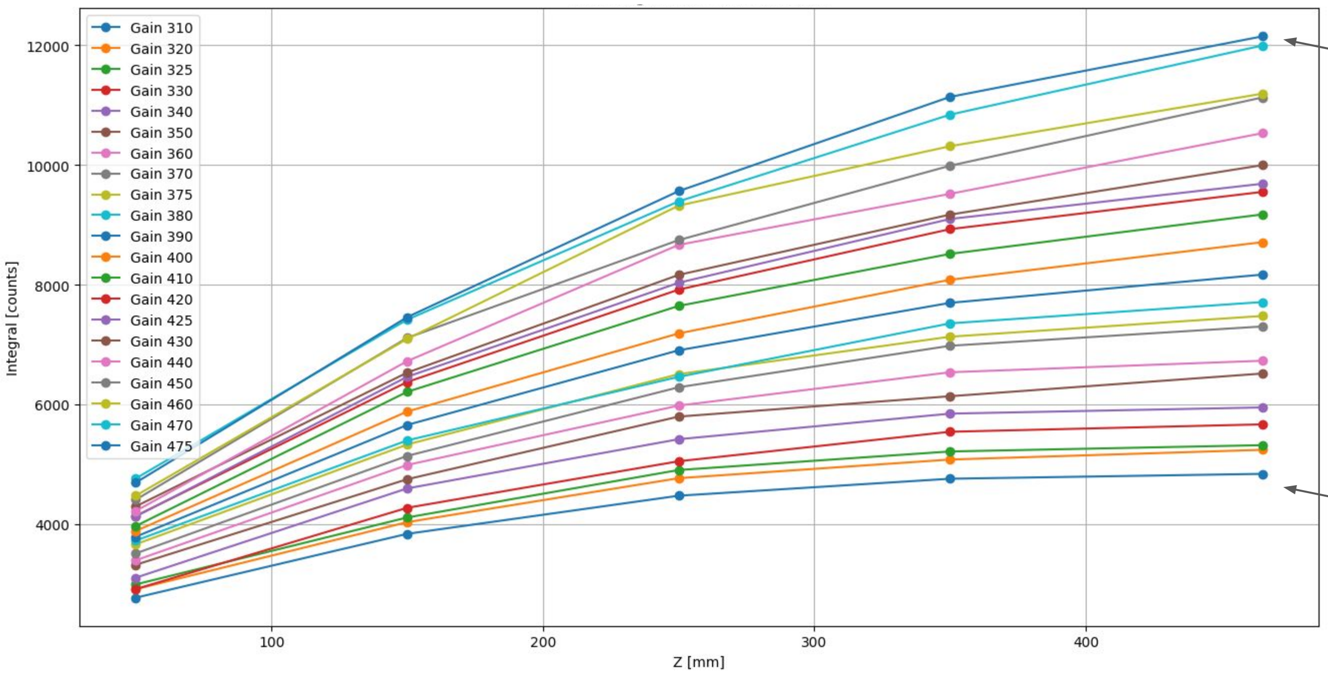
In theory, we expect more attenuation at high humidity. But it seems to be the other way around.



Also, we don't expect such a big difference in attenuation

Maybe the difference is not due to attenuation?

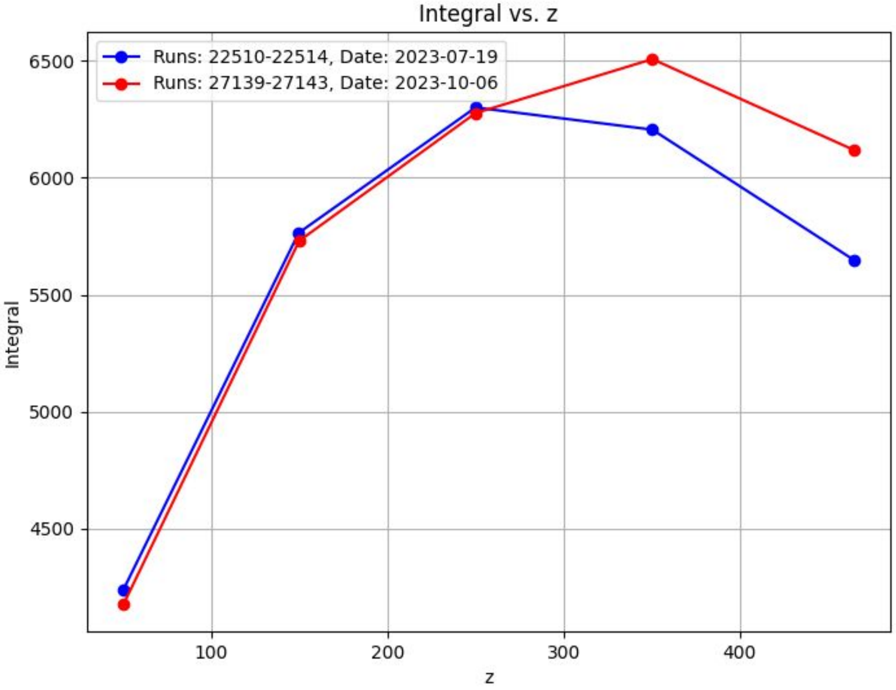
As you see, different LY/gains can make the attenuation more evident.

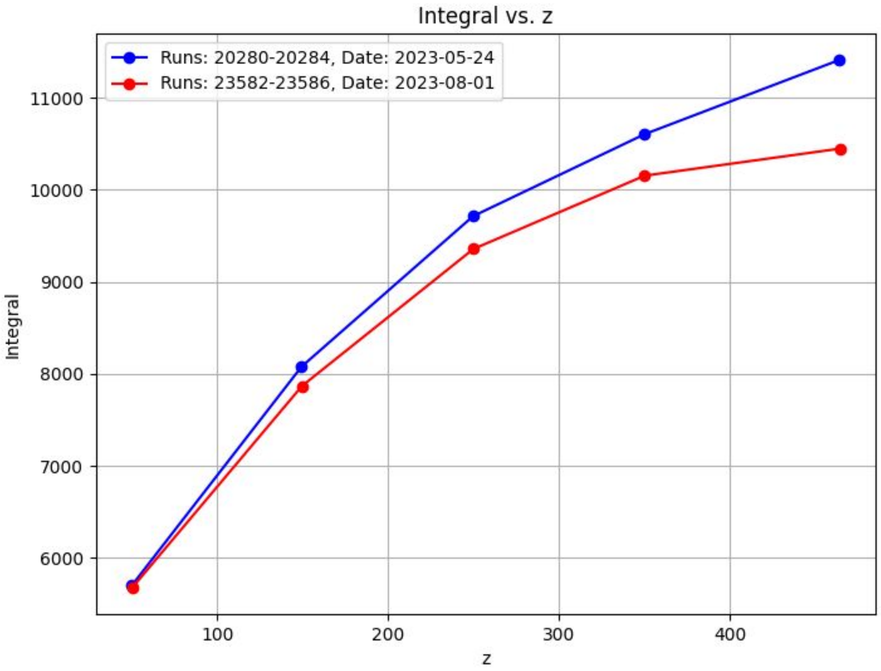


high gain attenuation not evident

low gain attenuation starts to be important

However, we have pairs of runs (data) that saturates in the same way and then diverge. So it really seems to be a difference in attenuation.

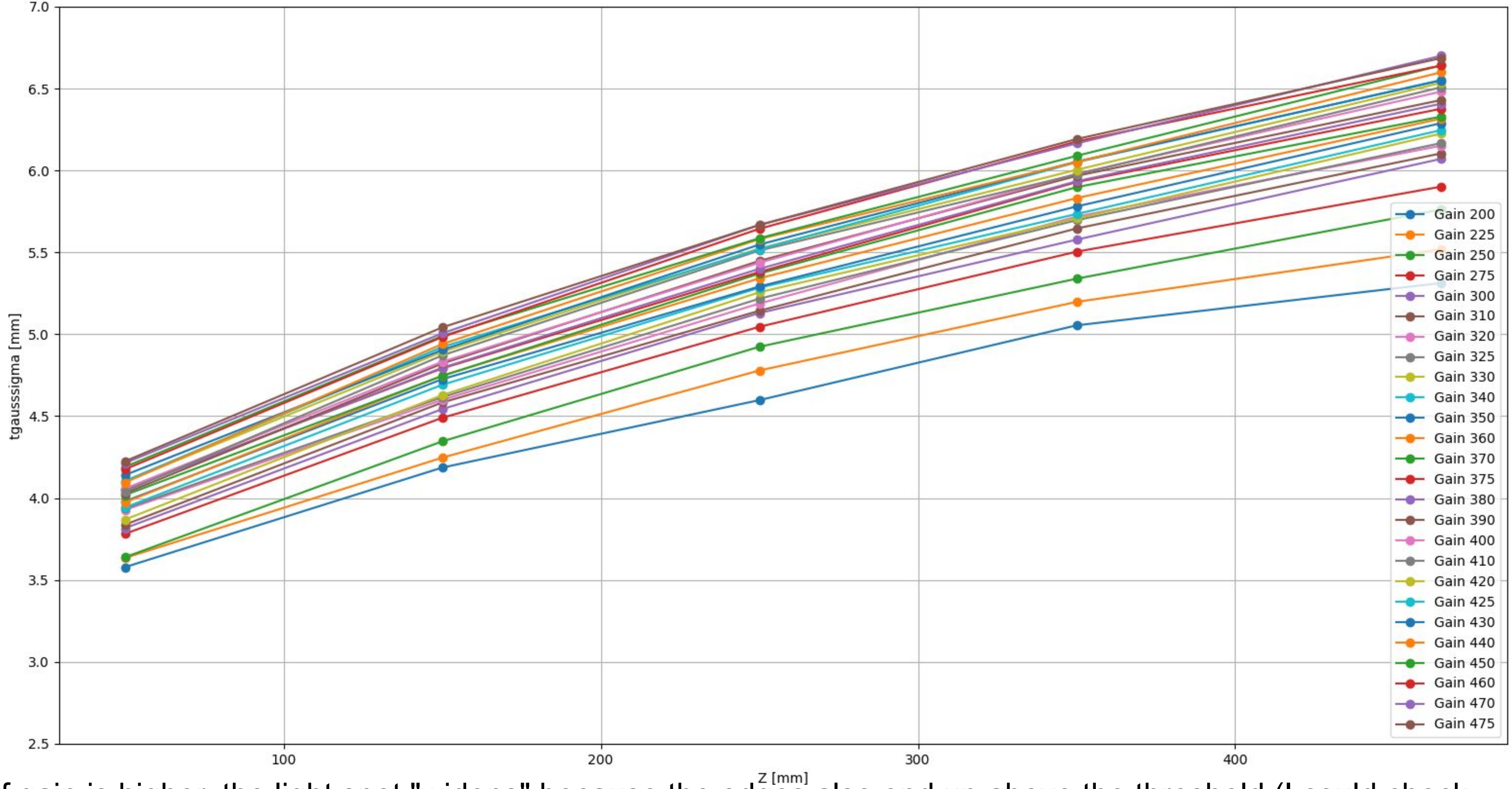




So, we can’t reproduce these different behaviors in MC without changing the attenuation

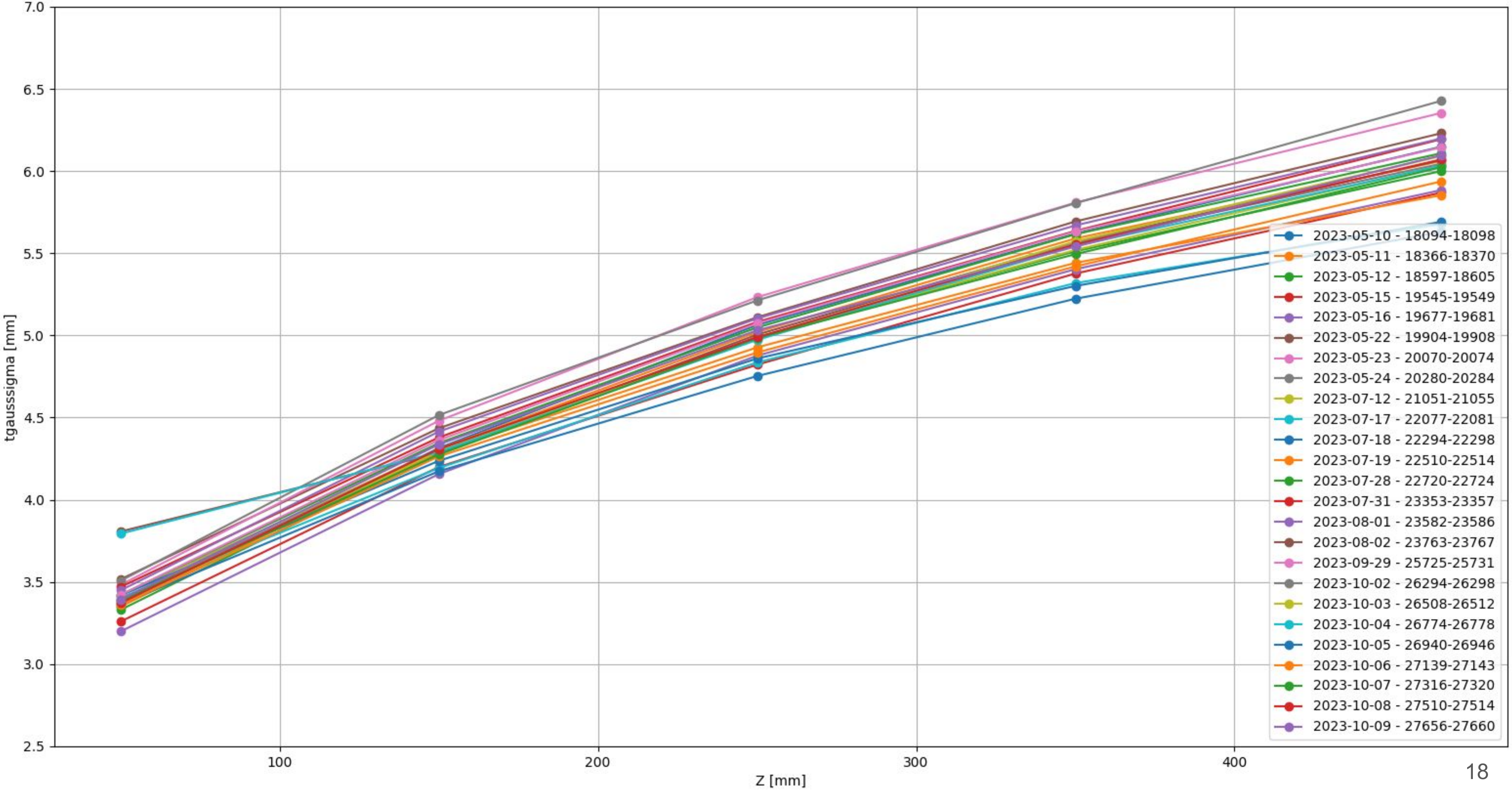
Let’s looking at tgausssigma (spotsize)….
Forgetting the attenuation problem

Simulated data



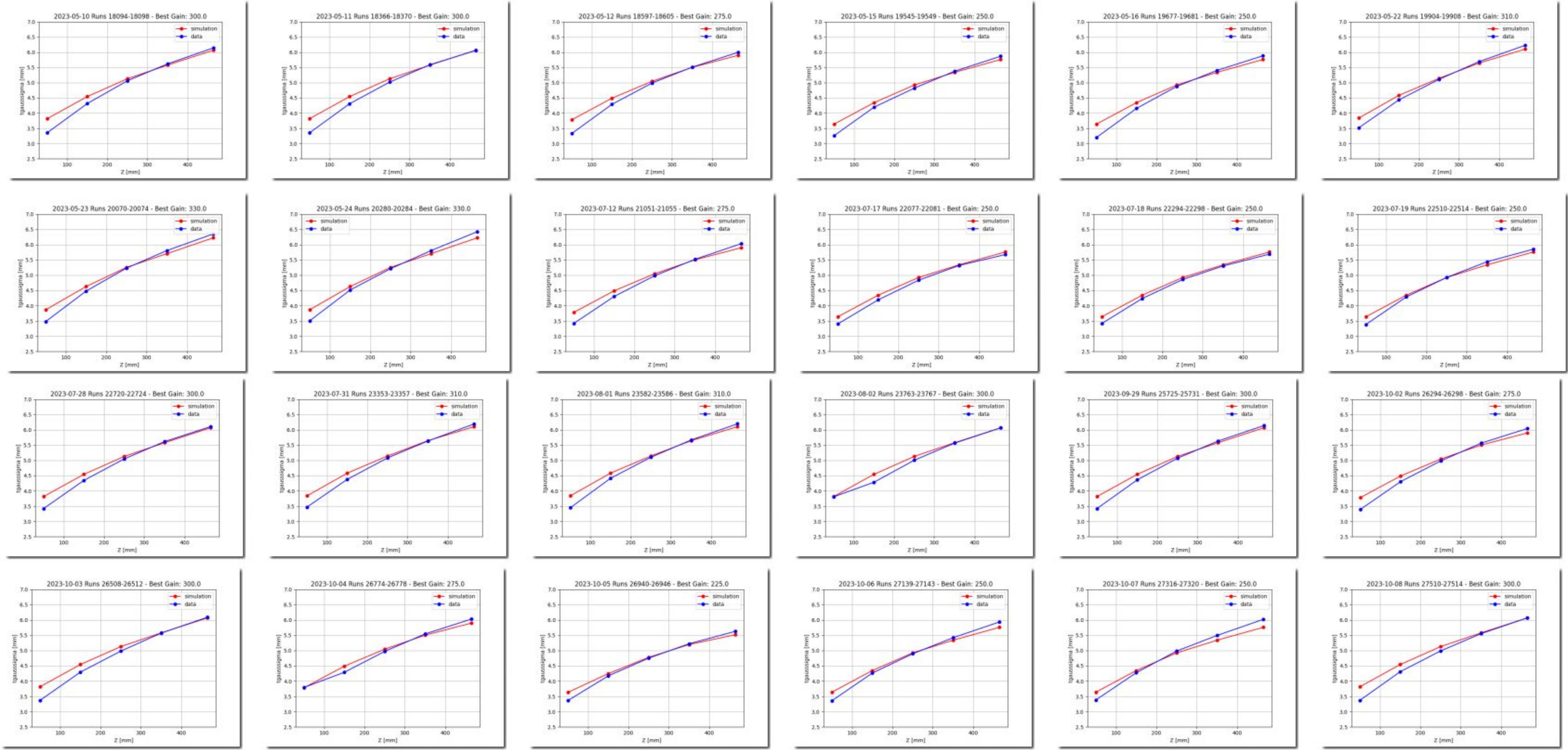
If gain is higher, the light spot "widens" because the edges also end up above the threshold (I could check if this doesn’t happen when not applying camera noise and reconstruction)

Real data



7.0
6.5
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
tgaussigma [mm]
100
200
300
400
Z [mm]
2023-05-10 - 18094-18098
2023-05-11 - 18366-18370
2023-05-12 - 18597-18605
2023-05-15 - 19545-19549
2023-05-16 - 19677-19681
2023-05-22 - 19904-19908
2023-05-23 - 20070-20074
2023-05-24 - 20280-20284
2023-07-12 - 21051-21055
2023-07-17 - 22077-22081
2023-07-18 - 22294-22298
2023-07-19 - 22510-22514
2023-07-28 - 22720-22724
2023-07-31 - 23353-23357
2023-08-01 - 23582-23586
2023-08-02 - 23763-23767
2023-09-29 - 25725-25731
2023-10-02 - 26294-26298
2023-10-03 - 26508-26512
2023-10-04 - 26774-26778
2023-10-05 - 26940-26946
2023-10-06 - 27139-27143
2023-10-07 - 27316-27320
2023-10-08 - 27510-27514
2023-10-09 - 27656-27660

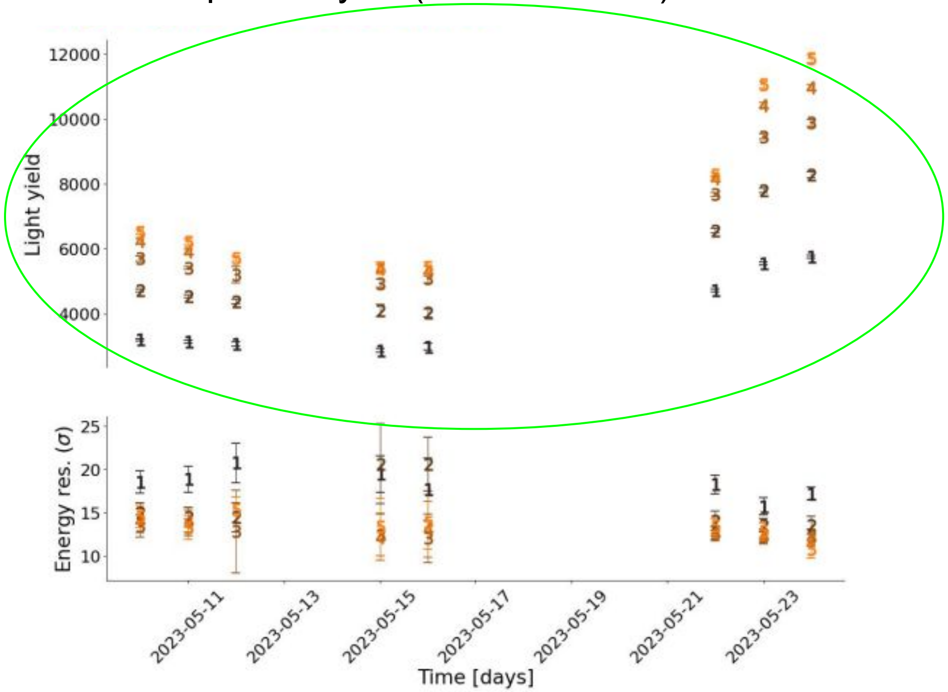
Now, best gain parameters go from **225 to 330** (while the fit on the integral gave values from **310 to 470**)



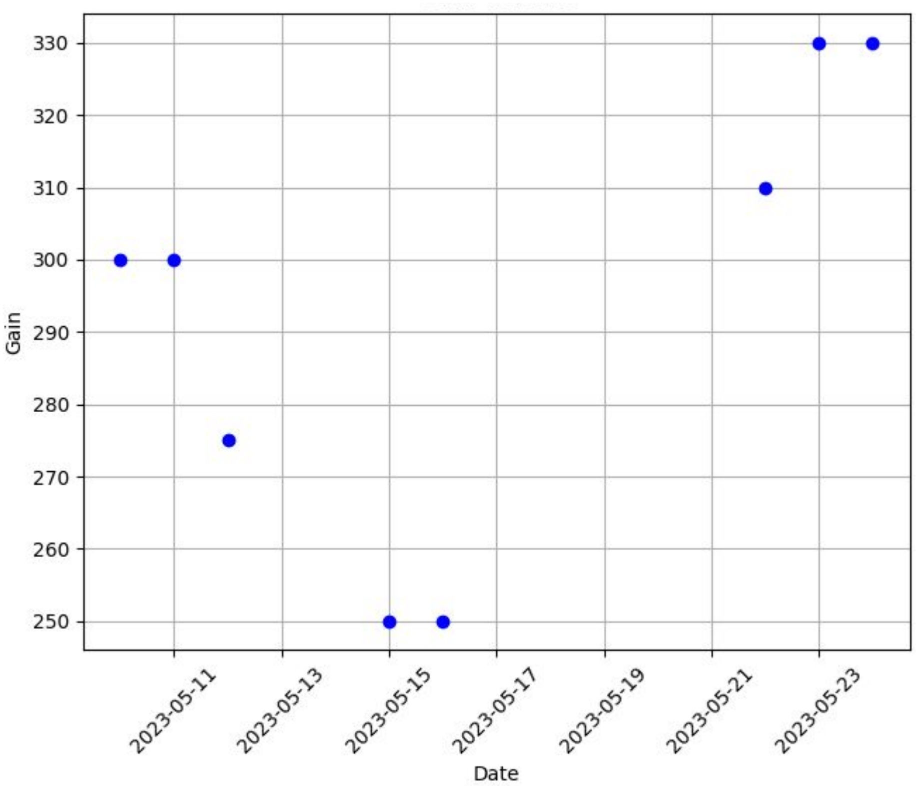
Discrepancies at low z mean we diffuse too much in the GEM. Discrepancies between the range of 'best gains' also mean the GEM diffusion in real data is less than in MC.

In May 2023 the best gain parameter has the same trend as the LY in data

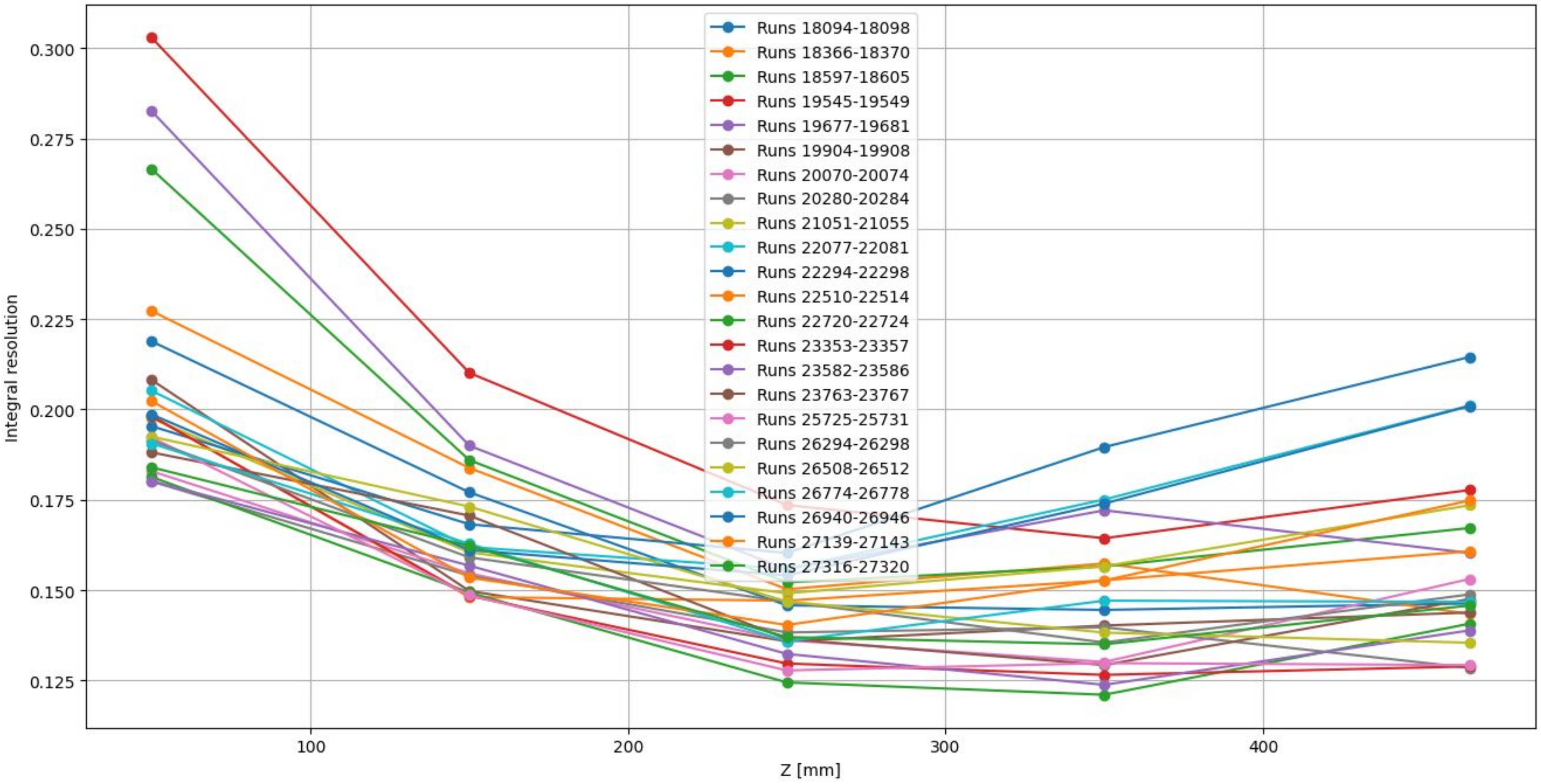
Rita Roque analysis (Coimbra 2023)



Best gain parameters for digitization

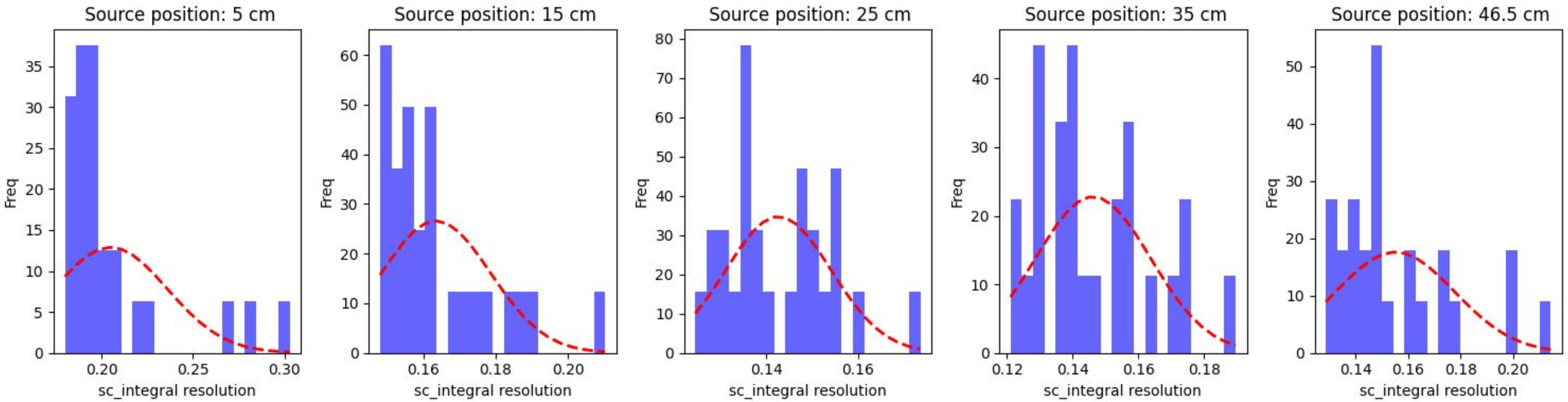


Let’s have a look at the integral resolution vs z…

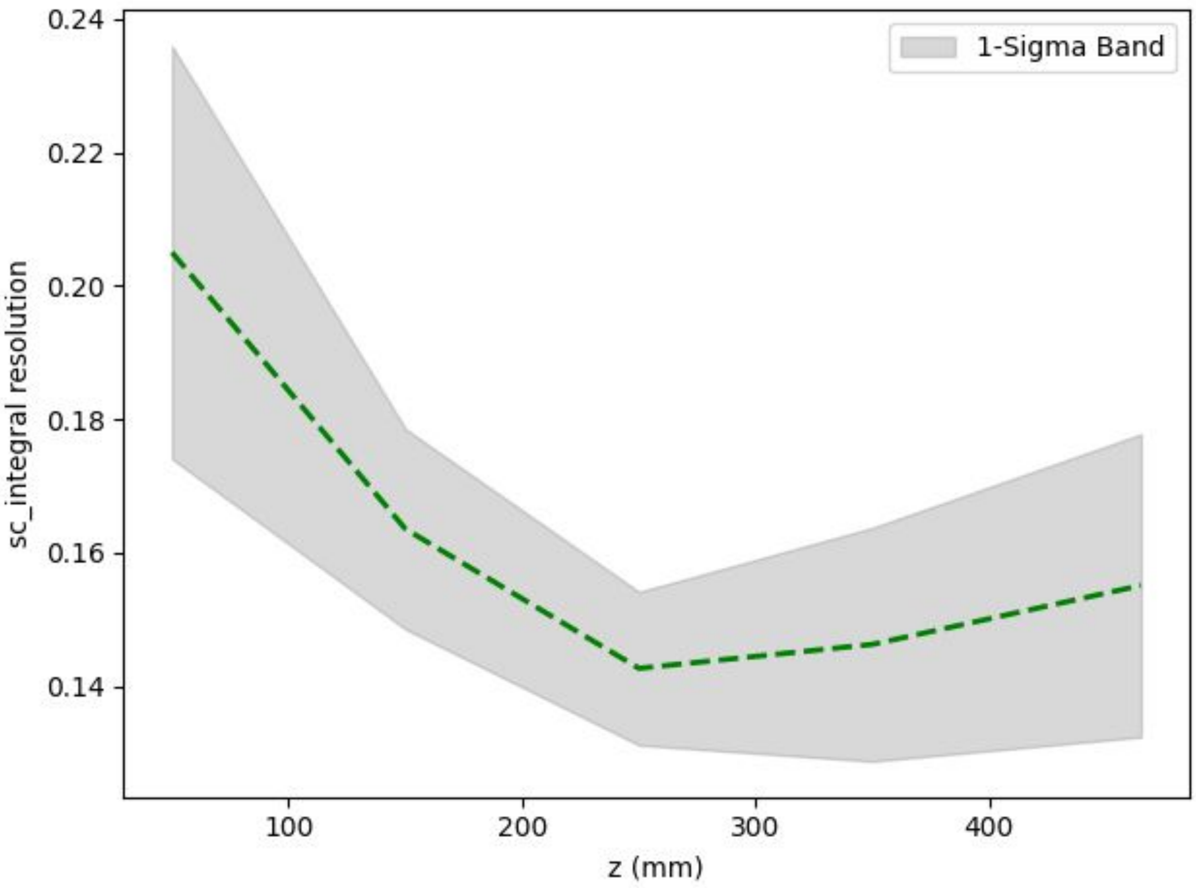


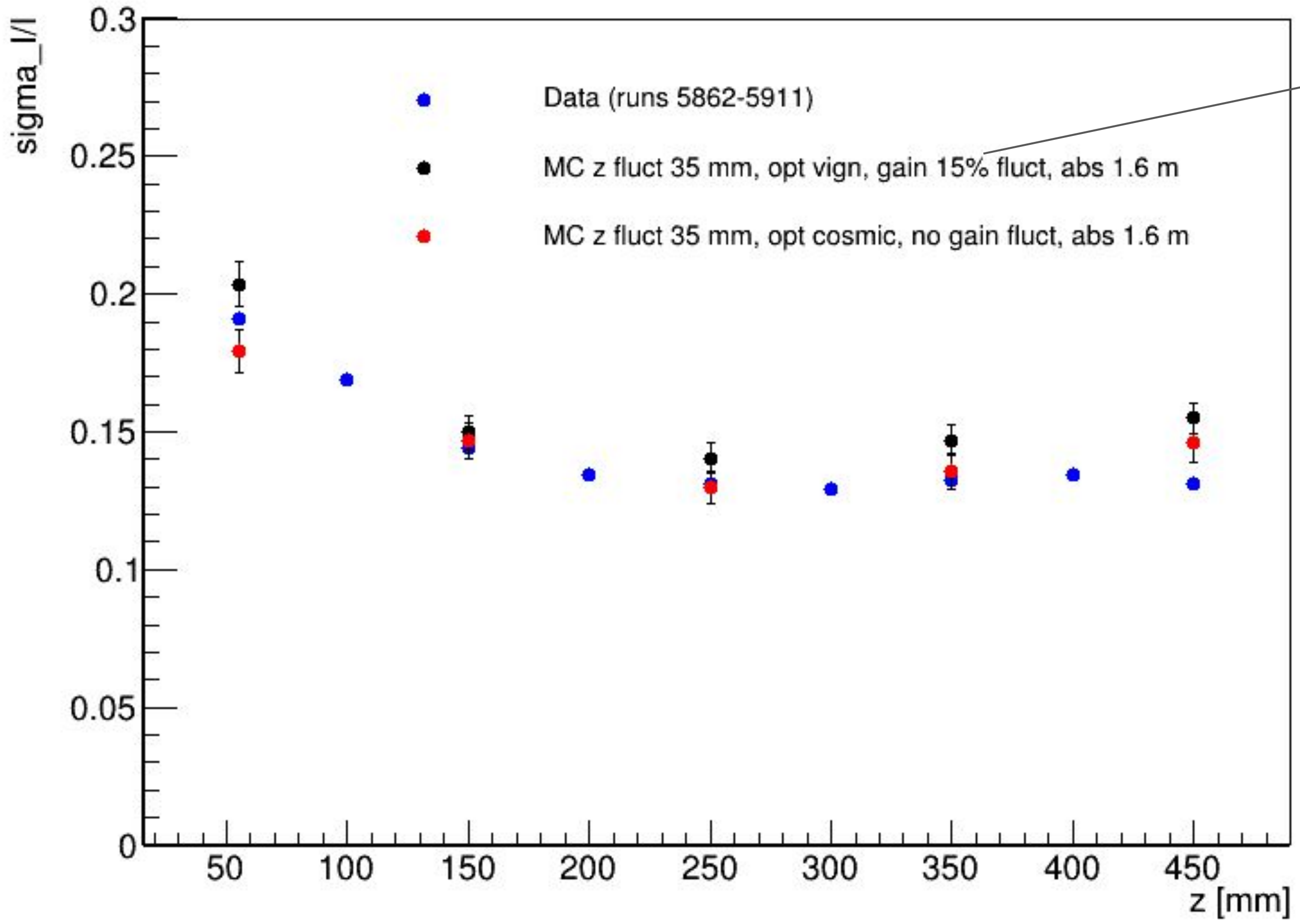
Runs 18094-18098
Runs 18366-18370
Runs 18597-18605
Runs 19545-19549
Runs 19677-19681
Runs 19904-19908
Runs 20070-20074
Runs 20280-20284
Runs 21051-21055
Runs 22077-22081
Runs 22294-22298
Runs 22510-22514
Runs 22720-22724
Runs 23353-23357
Runs 23582-23586
Runs 23763-23767
Runs 25725-25731
Runs 26294-26298
Runs 26508-26512
Runs 26774-26778
Runs 26940-26946
Runs 27139-27143
Runs 27316-27320
Integral resolution
0.300
0.275
0.250
0.225
0.200
0.175
0.150
0.125
100
200
300
400
Z [mm]

Gaussian fit on the 23 resolution values at given z:



We build a 1-sigma band, that we should use for comparison with MC





Gain fluctuations to simulate gain disuniformities

The band created with LNGS data overlaps well with both LNF data and the MC

Conclusions

1. The good comparison between data/MC at LNF was obtained by keeping the GEM gain fixed (as measured at LNF) and by adjusting the normalization parameter in the saturation A
2. We now know gain fluctuates a lot, so we can’t keep it fixed in MC.
3. As soon as we have a measurement of the absolute gain/LY in data we should use this information to set the gain in digitization
4. There are still things we don’t understand in data (attenuation)
5. Diffusion in GEM is probably overestimated in MC
6. Integral resolution also varies a lot -> we should consider those fluctuations when comparing with MC, in simulation paper