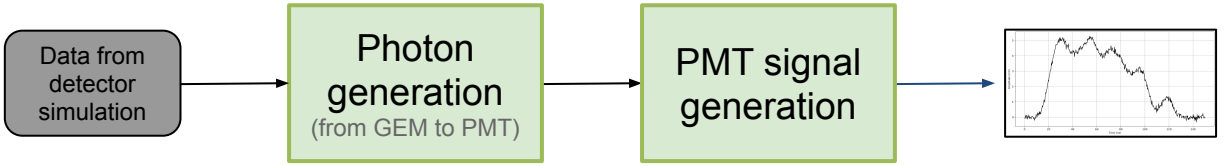
PMT Signal Simulation

Presented by: ***Rafael A. Nóbrega***
Universidade Federal de Juiz de Fora (UFJF)

with ***Davide Pinci*** *(INFN-Roma I),* ***Luan Gomes*** *and* ***Mariana Migliorini*** *(UFJF)*

January 30, 2023

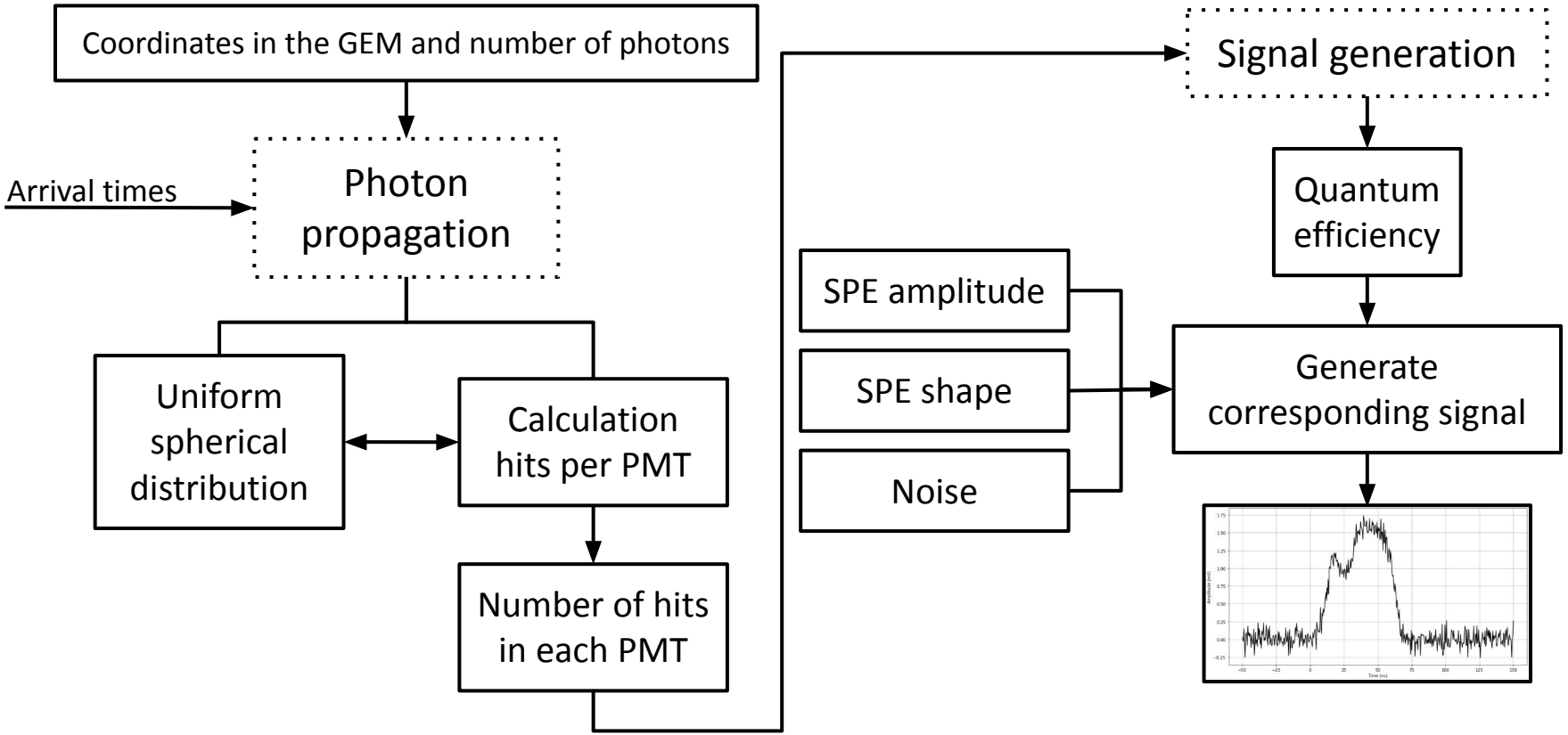
Summary



- PMT Signal Simulation
  - General Description
  - Simulation Functions
  - Next Steps

General Description

Modular implementation



Simulation functions

Example

- A particle with 3 clusters and 1000 photons each, was simulated in the following GEM positions:
  - Cluster 1: x = z = 50
  - Cluster 2: x = z = 100
  - Cluster 3: x = z = 150
- The PMTs are in a distance y = 134 from the GEM
- Each cluster arrives 10 ns after the other.

```
# Input parameters:
x0 = np.array([50, 100, 150])
z0 = np.array([50, 100, 150])
n_fotons = np.array([1000, 1000, 1000])
arr_times = np.array([0, 10, 10])
```

↓

```
ptc_object = PhotonPropagation(x0, z0, n_fotons, arr_times)
pmt_hits = ptc_object.pmt_hits()
```

↓

```
pmt_hits = {'cluster_0': {'pmt1': 0, 'pmt2': 0, 'pmt3': 2,
                          'pmt4': 1, 'arrival_time': 0},
            'cluster_1': {'pmt1': 0, 'pmt2': 0, 'pmt3': 1,
                          'pmt4': 0, 'arrival_time': 10},
            'cluster_2': {'pmt1': 0, 'pmt2': 0, 'pmt3': 0,
                          'pmt4': 1, 'arrival_time': 10}}
```

Example

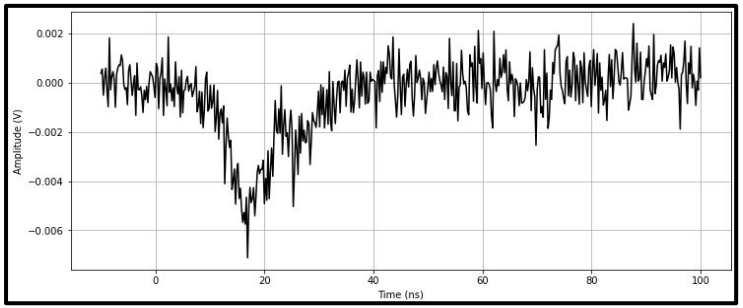
- Using a PMT quantum efficiency of 95%, the particle previously simulated in the photon generation part produced the following signal in PMT 3:

```
pmt_hits = {'cluster_0': {'pmt1': 0, 'pmt2': 0, 'pmt3': 2,
                          'pmt4': 1, 'arrival_time': 0},
            'cluster_1': {'pmt1': 0, 'pmt2': 0, 'pmt3': 1,
                          'pmt4': 0, 'arrival_time': 10},
            'cluster_2': {'pmt1': 0, 'pmt2': 0, 'pmt3': 0,
                          'pmt4': 1, 'arrival_time': 10}}
```

```
ptc_simulation = SignalSimulation(pmt_hits)
pmts_signal = ptc_simulation.particle_signal()
```

```
pmt3_signal = pmts_signal['cluster_0']['pmt3'] + \
              pmts_signal['cluster_1']['pmt3'] + \
              pmts_signal['cluster_2']['pmt3']
```

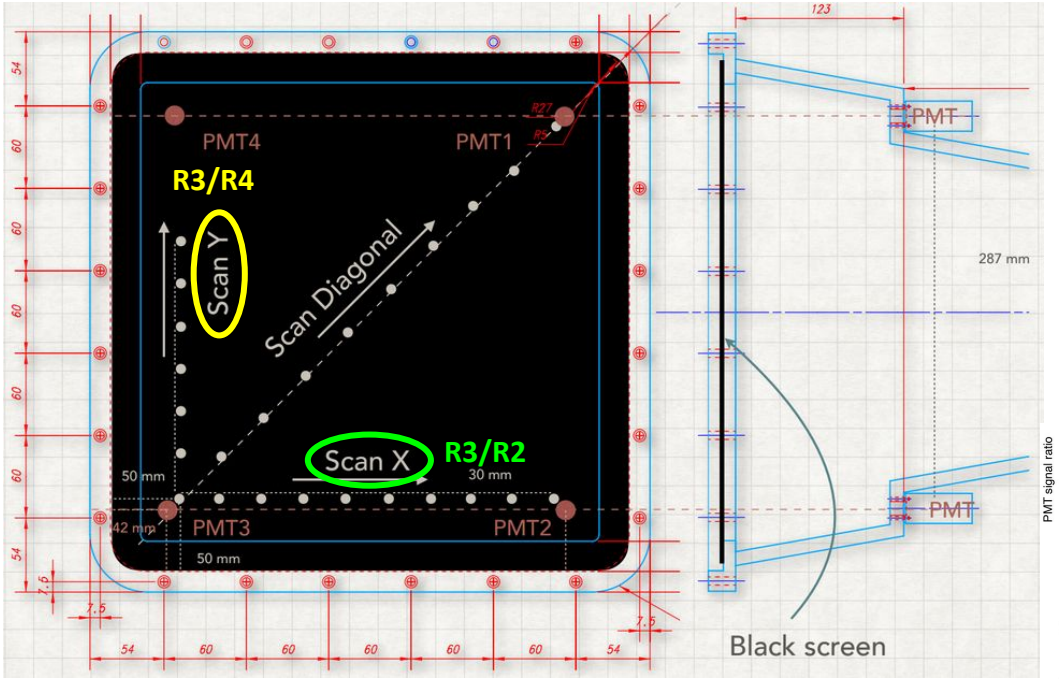
```
plt.figure()
plt.plot(pmts_signal['time'], pmt3_signal, 'k')
plt.ylabel('Amplitude (V)')
plt.xlabel('Time (ns)')
plt.grid()
plt.show()
```

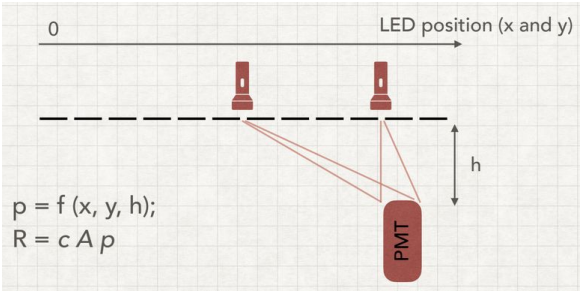


Next Step

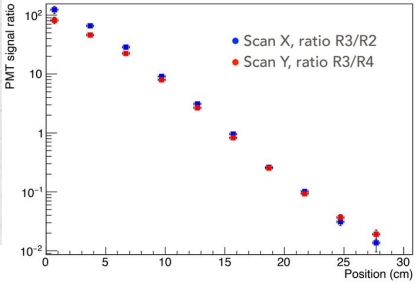
Photon Propagation Test

Preliminary tests

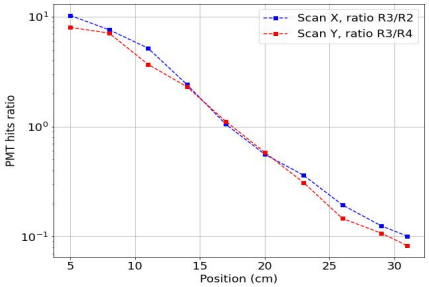




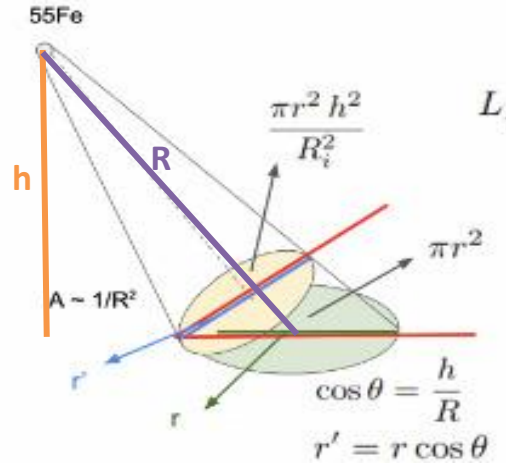
REAL DATA



SIMULATION



PMT hits



$$\frac{\pi r^2 h^2}{R_i^2}$$

$$L_{PMT} = \frac{L_{spot}}{4\pi R_i^2} \frac{\pi r^2 h^2}{R_i^2}$$

$$\pi r^2$$

$$\cos\theta = \frac{h}{R}$$

$$r' = r\cos\theta$$

PMTs positions

$$Y_{PMT} = 134$$

$$PMT_1 : X_1 = 312, Z_1 = 312$$

$$PMT_2 : X_2 = 312, Z_2 = 42$$

$$PMT_3 : X_3 = 42, Z_3 = 42$$

$$PMT_4 : X_4 = 42, Z_4 = 312$$

Equation variables

$$R = \sqrt{(X_{PMT} - x_0)^2 + (Y_{PMT} - y_0)^2 + (Z_{PMT} - z_0)^2}$$

$$h = Y_{PMT} = 134$$

$$r = 11$$

$$L_{spot} = 5 \times 10^5$$

Next Steps

- Test the photon propagation code;
- Add the noise characteristics to the signal generation module;
- Make the simulation algorithm available in the CYGNO repository.

Thank you!