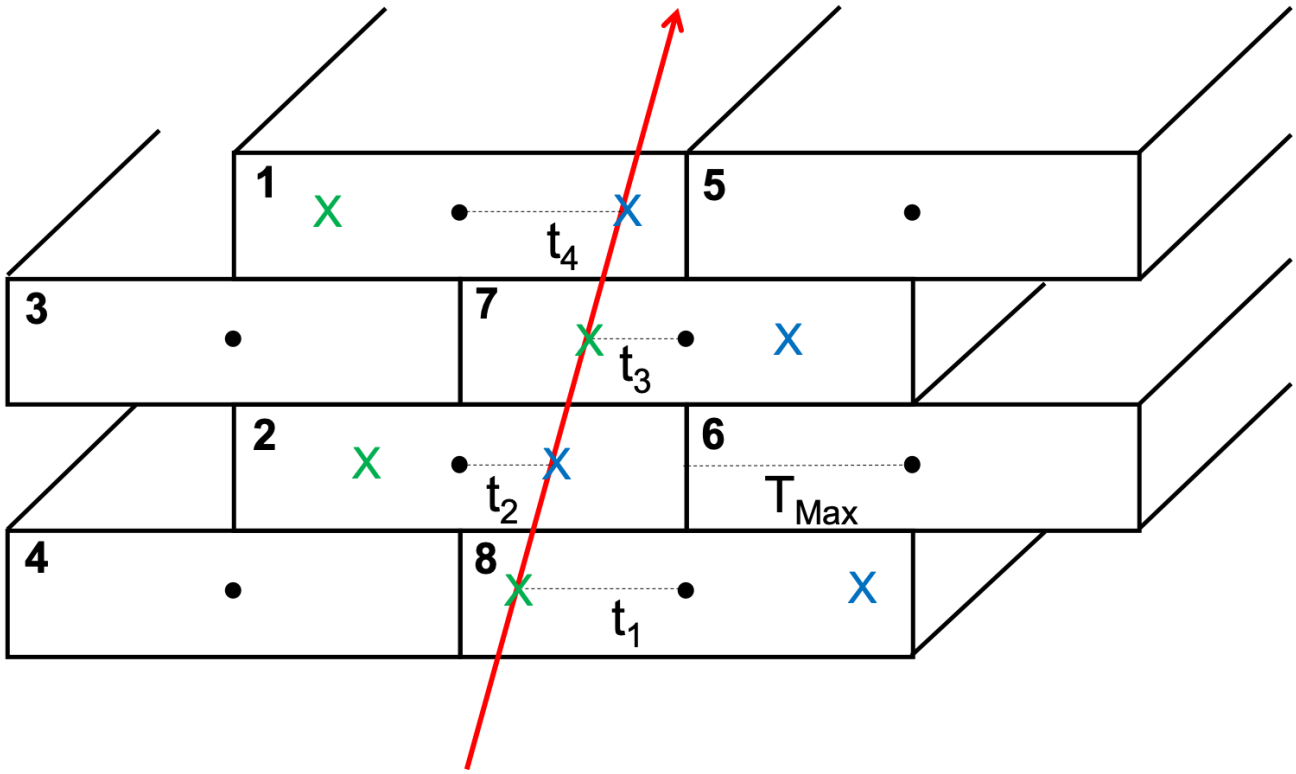
Processing of triggerlessly acquired detector's data

Description

The data to be analyzed have been gathered by a series of (4) particle detectors. Such detectors were composed by four layers of cells, each measuring the time of the electronic signal released at the passage of a charged particle. More in detail, the particle ionizes the gas within the cell and the so produced electrons drift at a constant velocity towards the anodic wire at the center of cell. The goal is to precisely assess the position where a particle traversed a given cell; combining the information of four consecutive cells it is then possible to deduce the trajectory of the particle.

Let's use the follwoing picture as reference:



The charged particle is represented by the red arrow. The four layers of cells are visible, only two cells each were displayed whereas the actual detector has 16 cells per layer. The black dots represent the anodic wires towards which the ionization produced by the charged particle drifts. Each cell measures the time it records the inonization signal with rispect to a given counter (see later). There is a left-right ambiguity on where the particle passed by within a given cell (left hypothesis represented by green X's, right hypothesis represented by the blue X's).

The numbering of the cells is somewhat weird and follows what displayed in the picture.

The actual time taken by the ionization to reach the anodic wire (drift time) is proportional to the distance:

$$x = v_d(t - t_0)$$

where v_d is the constant drift velocity, t is the time recorded by the electronics and t_0 is a time pedestal which needs to be determined for every particles crossing the detector (the drift time is $t - t_0$). This is possible thanks to the geometry of the layers, more precisely by the fact that they are staggered by exactly half a cell. This allow to exploit a (set of) mean timer equation(s) to spot (trigger) the passage of a particle, e.g., thanks to Talete's theorem, the following relation holds:

$$T_{Max} = \frac{t_1 + t_3}{2} + t_2$$

where t_1, t_2, t_3 are the drift times of those three cells and T_{Max} is the maximum drift time and it is a known quantity: $T_{Max} = \frac{L}{2v_d} = 390$ ns ($L = 42$ mm is the lenght of the cell). When hits are found in a column of cells like the one displayed in the picture, the equation above can be used to determine the time pedestal (t_0), thus the drift times in each cell and thus the position of the hit within the cell (with the left-right ambiguity).

As it can be guessed from the picture, the same equation would hold for the drift times t_2, t_3, t_4 and similar ones for the other combinations.

Datasets

Data Format

The raw data (hexadecimal format) have been preprocessed to produce csv files, each row being formatted in the same way, with 6 fields encoding the address of the read out channel (i.e. the cell) and the actual time measurement:

- *HEAD*: always 1 (useless)
- *FPGA*: 0 or 1
- *TDC_CHANNEL*: in the range 1-128, special values are present too (137, 138, 139)

- *ORB_CNT*: in the range $0 - 2^{32}$
- *BX*: in the range $0 - 3564$
- *TDC_MEAS*: in the range $1 - 30$

where the channel are mapped to the four detectors in the following ways:

- Detector 1 → FPGA 0, TDC_CHANNEL in [1-64]
- Detector 2 → FPGA 0, TDC_CHANNEL in [65-128]
- Detector 3 → FPGA 1, TDC_CHANNEL in [1-64]
- Detector 4 → FPGA 1, TDC_CHANNEL in [65-128]

the time measurement is code similary to standard time (h:m:s), the time in nanoseconds is given by:

$$t = ORB\_CNT * 3564 * 25 + BX * 25 + TDC\_MEAS * 25 / 30$$

As it will explained later, actually the orbit information will not be necessary, i.e. the relevant time information will be the one from BX and TDC_MEAS.

Every row in the dataset represents a hit in the detector (a green/blue dot in the picture above).

Trigger and trigger-less data aquisition

The data have been taken exploiting an acquisition system which read the detectors sensors at 40 MHz (every 25 ns), without the need for an external trigger. Imagine this as a video recording versus a camera taking pictures when ordered to (external trigger). To limit the bandwidth "zero-suppression" is applied, i.e. sensors' data is recorded only when they contain something (i.e. when an hit is present in a cell).

Even though not used as such, an external trigger was present anyway. When it fired a special 64 bit word (a row in the csv file) were produced enconding the trigger information, i.e. the timing at which it occurred and which of the kind of trigger fired. More precisely:

- Mean-time trigger: TDC_CHANNEL=139
- Scintillator trigger: TDC_CHANNEL=137 or 138

Data location and runs

In your assigned VM (LCP-6, IP: 10.67.22.9) you find the relevant datasets on `\data`. Datasets have been grouped into "runs", where

- 260-262 are "calibration" runs
- 333 is a "physics" run

The difference between the two is that in the calibration run, most of the hits can be associated to a particle's track, whereas in the physics run most of the hits are due to background events.

Assignments

An "event" can be thought as the passage of particle through the detector(s), i.e. a collection of hits produced by the particle's track. The datasets are not composed of events, instead they consist of just a series of hits. The goal of the project is to process the data to reformat them as list of events.

Note that the rate of passage of particles through the detector was rather limited, i.e. the chance to get two particle within the same orbit is extremely small.

The following assignments should be accomplished using Pandas and all its functionalities. The datasets from the calibration runs should be looked at first for simplicity.

- Part 1
  - Grouping the hits (rows) by orbits, identify and create events by means of the external trigger information. An event must be a collection of hits expressed in terms of coordinates (chamber, layer, cell) and position within the cell (with the left-right ambiguity NOT resolved). The hits should be close in time to the trigger
  - Verify that (at least some of) the hits of the events so produced align in a patter similar to the one displayed the the picture above
- Part 2
  - Implement the mean-timer trigger, i.e. process the data (still grouping the hits by orbit) to spot an alignment as the one displayed the the picture above. When such an alignment is found, use that as a trigger, i.e. create an event with the hits compatible in time with those giving the alignment.
  - Compute the time pedestal ($t0$) and subtract it from the measured times to obtain the drift times. Plot the distribution of the drift times
- Part 3
  - Repeat part 1 and part 2 with the dataset from the physics run, **OR**
  - Repeat part 1 with Apache Spark an analysis engine for distributed analysis based on DataFrames (contact the reference people below for that)

Contacts

- Marco Zanetti marco.zanetti@unipd.it (mailto:marco.zanetti@unipd.it)
- Jacopo Pazzini jacopo.pazzini@unipd.it (mailto:jacopo.pazzini@unipd.it)

In [ ]: