UCT-ALICE-TRD Documentation

Release latest

Oct 14, 2019

PyTRD Classes

The physics department at the University of Cape Town is in possession of a Transition Radiation Detector (TRD), similar to those used on the ALICE experiment at CERN. This is a python project designed to make reading, processing, and visualising the data acquired from the TRD as easy and as extensible as possible.

This project was adapted from the initial pytrd code which can be found here. A new repository was created as the whole code structure changed.

CHAPTER 1

Installation

PyTRD can be installed by downloading the pytrd file from the repository and placing it into the 'Lib' folder in the main Python directory.

CHAPTER 2

PyTRD Classes

2.1 trdfile

Creates a new instance of a TRD file using the file name provided. The TRD file is a basic list implementation of the raw data contained in the file provided. Each item in the list is a dictionary that contains the version of the data format used, the time stamp of the event, the number of data segments and the list containing the data segments. Each data segment is a dictionary containing the sfp number, the size of the segment, and the list of the raw data in the segment.

Custom methods:

from_json(filename) Returns an event dictionary from a json object file

save_json(self, filename) Saves the raw data to a json object file

Magic methods:

__getitem__(self, key) Return self[key]

__len__(self) Return len(self)

__str__(self) Return str(self)

2.2 trdevent

Class that manages the data for 1 single detection event. A TRD event has several attributes.

version: The current version of the format of the data blocks.

time: The time the run was started.

segments: The number of data segments in the event.

blocks: The list containing the raw data blocks.

data_cube: The numpy array containing the processed data.

TRD events support numpy-like indexing where trdevent[slice] is equivalent to data_cube[slice].

Custom methods:

ave_signal(self) Returns the average number of counts throught the whole data block.

get_axes(self) Returns the real space X, Y, and Z axes.

regions_of_interest(self) Returns an iterator of the all the regions of interest in the data cube.

set_drift(self, zdrift, uzdrift=0) Method for setting the drift velocity of the electrons. Defaults to 1,uncertainty default to 0.

set_mask(self, mask, value=0) Method for bulk setting values in the data cube. Useful for areas of the TRD that are returning garbage data.

Magic methods:

__getitem__(self, key) Return self[key]

__len__(self) Return len(self)

__str__(self) Return str(self)

2.3 visual

This is a class containing tools helpful for visualising the data stored in the data cube.

Methods defined here:

get_tracks(self) Finds and returns the cosmic rays based on a linear fit of interesting regions.

plot_tracks(self, tracks) Method that plots position of the intersection of the track with the xy-plane, given the depth of the plane z and the axes.

pulse_height(self, show=True) Displays a pulse height graph for the event.

slices(self) Displays an interactive graph showing the ADC data for each slice of the data cube.

stack(self) Displays a plot of the summed count values along the time bin axis.

stack_tracks(self, tracks) Displays a plot of the summed count values along the time bin axis as well as plotting lines showing the particle tracks through the detector.

2.4 linear

This is a class that contains linear fitting tools.

Methods defined here:

linear_1d(x, y, uy) Finds the values for m and c that minimises the sum of the squares of the residuals weighted by the uncertainty squared.

linear_fit(points) Returns a linear fit of 3 dimensional data.

linear_function(z, M, C) Returns the coordinates of the points generated by the vector function M*z +C

2.5 DataFormatException

This is an exception that is raided when the format of the dat received by the program is invalid. It can be caused by zero suppressing the data from the TRD. It can be an indicator that there is something wrong with the electronics on the TRD.

trdfile Class that opens and processes the raw data received from the TRD.

trdevent Class that manages the data for 1 single detection event.

visual Class that contains tools for visualising TRD events.

linear Class containing linear fitting tools.

DataFormatException Raised when the data format is invalid.

CHAPTER 3

DAQ System