TPPTcoinc description

1. Introduction

TPPTcoinc is a part of the TPPT software framework, developed at the LIP-Coimbra group in the frame of the TPPT collaboration. The framework consists of the TPPTsim (simulator), TPPTbuilder (event builder) and TPPTcoinc (coincidence sorter). The purpose of TPPTcoinc is to find coincidences in the scintillation events, constructed by TPPTbuilder using the simulation data from TPPTsim. The identified coincidence pair data are saved to ASCII or binary output files according to the format defined by the image reconstruction group.

The coincidences in scintillation events are found based on the user-configured time and energy windows: e.g. both events should be separated in time by less than 10 ns and their energies should be inside ±10% range from 0.511 MeV. The method used in the process can be selected from two available options: a *Basic* and *Advanced* ones. The *Basic* method rejects all multiple coincidences and considers scintillation events in different scintillators always as independent ones. The *Advanced* method offers flexibility in the multiple event rejection and allows to consider scintillation events in the same 8x8 assembly of scintillators as a single event if their timestamps are close enough.

2. Configuration

The simplest way to configure TPPTcoinc is to edit the settings directly in the coinc.cc file. The configuration section can be easily identified as it starts with

```
// --- Start of user inits ---
```

and ends with

```
// --- End of user inits
```

comments. An example of a configuration is given below, with the variables explained in the following text.

```
Config.WorkingDirectory = "/home/user/tmp";
Config.InputFileName = "BuilderOutput.txt"; Config.BinaryInput = false;
Config.LutFileName = "LUT.txt";
Config.OutputFileName = "CoincPairs.txt"; Config.BinaryOutput = false;
Config.HeaderFileName = "Header.hlm";
Config.ExportLutFileName = "CrystalLUT.txt";
Config.FinderMethod = FinderMethods::Basic;
Config.RejectSameHead = true;
Config.CoincidenceWindow = 4.0; // ns
Config.EnergyFrom = 0.511 * 0.95; // MeV
Config.EnergyTo = 0.511 * 1.05; // MeV
Config.TimeFrom = 0; // ns
Config.TimeTo = 1e50; // ns
```

- WorkingDirectory gives the path where the input files are located. The output files will also be created in this directory.
- BinaryInput / BinaryOutput flags set to true signify that the input / output files are binary, otherwise they are in the ASCII format. Note that the LUT and header files are always ASCII files.
- InputFileName is the name of the file with the scintillation events created by TPPTbuilder
- LutFileName is the name of the file containing the indexing/positioning data of all scintillators, created by the TPPTsim
- OutputFileName is the output file name
- HeaderFileName is the output header file with the format defined by the reconstruction group
- ExportLutFileName is the output LUT file with the positions of the scintillators in the format defined by the reconstruction group
- FinderMethod is the selector for the coincidence finder method: it can be set to *Basic* and *Advanced* method
- RejectSameHead flag set to true signifies that coincidences with both event in the same detector head (“crescent”) should be disregarded
- CoincidenceWindow defines the maximum time (in ns) between two scintillation events to consider them coincident
- EnergyFrom defines the minimum energy for the scintillation event to be considered for coincidences. When the *Advanced* finder method is activated, the meaning of the EnergyFrom setting can be differently (see section 3.3.2)
- EnergyTo EnergyFrom defines the maximum energy for the scintillation event to be considered for coincidences. When the *Advanced* finder method is activated, the meaning of the EnergyFrom setting can be differently (see section 3.3.2)
- TimeFrom can be used to define the minimum timestamp (in ns) of the scintillation events considered by the coincidence finder
- TimeTo can be used to define the maximum timestamp (in ns) of the scintillation events considered by the coincidence finder

Note that each run of the TPPTcoinc creates a configuration file CoincConfig.json in the working directory, which lists all the parameter values (JSON format). Such files provide an alternative way to configure TPPTcoinc: the parameter values could be read from such files. To use this configuration mode, provide the file name as the first argument when starting the TPPTcoinc executable. In this case the values given directly in the coinc.cc have no effect.

3. Workflow details

The workflow consists of four phases: (1) import the scintillator positioning information from a file generated by TPPTsim; (2) load scintillation events from a binary or ASCII file generated by TPPTbuilder; (3) sort events and find coincidences; (4) export results (coincidence data, header file, scintillator LUT file).

3.1 Import of the LUT with scintillator positioning information

The file with the scintillator positioning information is automatically generated by TPPTsim. It is an ASCII file which lists information for all 6144 scintillators consequently (index from 0 to 6143) in the following format:

FacePosX FacePosY FacePosZ NormX NormY NormZ HeadNumber Angle AssemblyNumber

where

- FacePosXYZs are the XYZ coordinates of the center of the inner surface (the one oriented towards the isocenter) of the scintillator
- NormXYZs are the XYZ components of the unit vector normal to the inner surface of the scintillator and oriented towards the isocenter
- HeadNumber is the index of the detector head (“crescent”): 0 or 1
- Angle is the rotation angle in degrees around the detector axis, which gives the position of the scintillator in respect to the one with index of 0
- AssemblyNumber is the index of an 8x8 scintillator assembly, containing this scintillator.

Each line is terminated with ‘\n’ (end of line) symbol. Space character is used as the delimiter.

3.2 Loading of the scintillation events

Depending on the configuration of the event builder (TPPTbuilder), the input file can be either in ASCII or binary format. The file lists the scintillation event records (pairs of time[ns] and energy[MeV]), grouped by the scintillator index.

For the ASCII files the format is the following:

```
# ScintIndex
Time_1 Energy_1
Time_2 Energy_2
…
Time_n Energy_n
# ScintIndex+1
Time_1 Energy_1
Time_2 Energy_2
…
Time_m Energy_m
…
```

Each line is terminated with ‘\n’ (end of line) symbol. The ‘#’ symbol marks the lines which contain the scintillator index (ScintIndex). Space character separates # symbol and the scintillator index, as well as the time mark and the energy values.

For the binary files the format is the following:

```
0xEE ScintIndex(int)
0xFF Time_1(double) Energy_1(double)
0xFF Time_2(double) Energy_2(double)
…
0xFF Time_n(double) Energy_n(double)
0xEE ScintIndex+1(int)
0xFF Time_1(double) Energy_1(double)
0xFF Time_2(double) Energy_2(double)
…
0xFF Time_m(double) Energy_m(double)
…
```

Note that 0xEE and 0xFF chars are used to indicate the type of the following record: scintillator index or time/energy record. There are no other delimiters.

The loaded scintillation events are stored in a container with the **event records** containing the scintillator index, timestamp and energy information. Note that events with the timestamp smaller than TimeFrom or larger than TimeTo configuration parameter values are discarded. Also, if the *Basic* coincidence finder method is selected (see next section), all events with the energy below EnergyFrom or larger than EnergyTo configuration parameter value are discarded as well.

3.3 Coincidences finding

As the first common step, all loaded scintillation event records are sorted by the event timestamp in the increasing order. The procedure for finding coincidence pairs is different for the *Basic* and *Advanced* methods, therefore, they are described separately in the following subsections.

Note that for both methods, if the RejectSameHead configuration flag is set to true, the pairs with both events in the same detector head (“crescent”) are disregarded.

3.3.1 *Basic* coincidence finder method

When this method is selected, TPPCcoinc is selecting pairs of events with the timestamp difference within the CoincidenceWindow configuration parameter. All multiple events (those which have more than two events within the CoincidenceWindow) are rejected. It is enforced that the coincidence pair should involve two different scintillators.

For diagnostics, the number of found coincidences, the number of single events and, if activated, the number of rejections for events belonging to the same detector head are reported in the terminal.

Note that for simulations performed with the probability-based approach for generation of positron emitting species this is the recommended method as multiple coincidences should not be present (see manual for TPPTsim).

3.3.2 *Advanced* coincidence finder method

If this method is selected, two additional parameters have to be configured by the user:

- GroupByAssembly - if set to true, events for different scintillators belonging to the same 8x8 assembly and with the timestamp within the CoincidenceWindow are considered as a single event with energy equal to the sum of those of the individual sub-events.
- RejectMultiples - this is the selector defining the approach in handling multiple events. The available options are RejectionMethods::All (all multiple coincidences are rejected), RejectionMethods::EnergyWindow (reject those multiple coincidences which have more than two scintillation event within the defined EnergyFrom and EnergyTo energy range) and RejectionMethods::None (multiple coincidences with more than two scintillation event within the defined EnergyFrom and EnergyTo energy range are allowed, the scintillation indexes for the reported coincidence pair are given by the two events with the largest energy).

The algorithm of finding coincidence pairs is the following.

1) The procedure starts from the first event record of the sorted event data
2) If the timestamp of the next record is different from this event by more than CoincidenceWindow (thus it is a single event), the cycle is repeated with the next event
3) All event records with the timestamp within the CoincidenceWindow are identified and added to the selection list
4) If the GroupByAssembly configuration flag is set to true, the records in the list belonging to the same assembly are combined (merged) to single records. The combined record’s parameter values are set to the following:
    a) the energy is the sum of the energies
    b) the scintillator index is the one of the record with the largest energy
    c) the timestamp is the minimum timestamp value over the records
5) If the number of remaining records is more than two and the rejection option is set to RejectionMethods::All, this coincidence pair candidate is rejected. In this case the cycle continues with the next event record after the ones identified in step (3) - this continuation is designated below as “→CONTINUE”
6) The energy selection criteria is applied: all events with the energy below the EnergyFrom and above the EneryTo configuration parameter values are removed from the selection list
7) If less than two records remains in the selection list → CONTINUE
8) Except the case when the rejection setting is RejectionMethods::None, if the number of remaining records in the list is more than two →CONTINUE

9) The remaining records in the list are then sorted to have continuously decreasing value of the event energy
10) The coincidence pair is formed using the first two records: the ones having the maximum energy. Note that the number of records can be above two only if the RejectionMethods::None option is activated
11) →CONTINUE

For diagnostics, the number of found coincidences, the number of single events, and some other parameters including the histograms of the event multiplicity before and after merging of records by assemblies are reported in the terminal.

3.4 Saving the results

3.4.1 Coincidence pairs

Depending on the BinaryOutput configuration flag, the coincidence pairs can be saved in an ASCII or binary file.

For an ASCII file, data for each pair is saved in a new line. The format of the line is:

```
Index1,Index2,DeltaT,T
```

where Index1 and Index2 ate the first and second scintillator indexes, DeltaT is time difference between the scintillations events in ps (DeltaT = T1 - T2), and T is T1 in ms. The lines are terminated with '\n' (end-of-line) character, and comma is used as the delimiter.

For binary files the format is the following:

```
Index1(int) Index2(int) DeltaT(float) T1(float)
```

The parameters are the same as for the ASCII files. There are no delimiters.

3.4.2 Header file

The header file is an ASCII file "Header.hlm" containing the following lines:

```
"Format version",1.1
"Format type",1
"Data file",”[OUTPUT_FILE_NAME]”
"Data type",1
"Number of events",[NUMBER_OF_EVENTS]
"Data mode",[DATA_MODE]
"Crystal LUT file","CrystalLUT.txt"
```

Here [OUTPUT_FILE_NAME] is the name of the output file with coincidence pairs, configured by the user, [NUMBER_OF_EVENTS] is the number of coincidences in the file and [DATA_MODE] is set to 1 for ASCII and to 2 for binary format of the file with the coincidence data. Note that the name of the output file with the scintillator positioning is always the standard one: "CrystalLUT.txt"

3.4.3 LUT file

The LUT file is an ASCII file "CrystalLUT.txt" containing information on the scintillator positions. Each line lists the XYZ positions of the center position of the scintillator front surface (the one oriented towards the isocenter) in the order given by scintillator indexes (from 0 to 6143) with comma as the delimiter:

```
X,Y,Z
```