

The DTFitter

DTFitter: environment

The same environment provided by the [guide](#) is suitable.

DTFitter: downloading and building the code

The source code for the DTFitter is stored in [Baltig](#).



Since the source code of the DTFitter can be downloaded only by the authorized developers, credentials for Baltig are necessary.

The simplest way is using RSA keys, the public key must be uploaded into Baltig and the private key must be saved into `${HOME}/.ssh/id_rsa`

Once git client has been correctly configured the commands are:

```
git clone git@baltig.infn.it:muontomography/DTFitter.git
cd DTFitter
```

The commands to build the code are:

```
cmake3 <path-to-source>
make
```



<path-to-source> is the path to the CMakeLists.txt file in DTFitter directory, e.g.: `/home/centos/DTFitter`

The executable is found in the directory **run/**

DTFitter: running the code

The DTFitter application performs the reconstruction of tracks of the Drift Tubes (DT) detectors from:

- dataset produced with the castor-simulator software, see [The OpenCMT Castor Simulator](#);
- dataset collected with the MUTOMCA apparatus.

Preliminary settings: the configuration file

All the input parameters needed to run the code must be set in a dedicated configuration file **config.ini**. A template of this file can be found in the **utils/** directory and must be copied in the **run/** directory. Now you can modify the **config.ini** file in your **run/** directory.

- General information

Variable	Type	Description
runNumber	<i>int</i>	number that identifies the dataset to be analyzed
rawDirName	<i>string</i>	path of the input file directory
rawFileName_1	<i>string</i>	name of the input DATA file for DET 0 (file 1_)
rawFileName_2	<i>string</i>	name of the input DATA file for DET 1 (file 2_)
rawFileNameMC	<i>string</i>	name of the input MC file
t0FileName	<i>string</i>	name of the file with the initial t0 (for DATA only)
noiseFileName	<i>string</i>	name of the file with noisy channels (for DATA only)
outputDirName	<i>string</i>	path of the output file directory, e.g. ../output
outputFileName	<i>string</i>	name of the output file
logFileName	<i>string</i>	name of the log file

maxEventNumber	int	maximum number of events to be read
----------------	-----	-------------------------------------

- Prototype

Variable	Type	Description
detector	int	detector to be analyzed: 1 for DET 0 (file 1_ of DATA), 2 for DET 1 (file 2_ of DATA), or 4 for both

- Execution mode

The application can be run in three different modes (please choose only one mode at each execution)

1. *Display mode*: show event-per-event display
2. *Histogram mode*: fill histograms for track reconstruction analysis, save them in a root output file and eventually display them at the end of execution
3. *TTree mode*: fill a root TTree with the reconstructed track parameters and save them in a root output file

Variable	Type	Description
display	bool	set to 1 to run in <i>Display mode</i> , 0 otherwise
start	uint	first event to be visualized with display
histos	bool	set to 1 to run in <i>Histograms mode</i> , 0 otherwise
showHistos	bool	set to 1 to display the histograms at the end of execution
tree	bool	set to 1 to run in <i>TTree mode</i> , 0 otherwise

- Debugging

Variable	Type	Description
debug	bool	set to 1 to dump debug messages

- Castor MC

Variable	Type	Description
isCastorMC	bool	set to 1 for simulated datasets (0 not implemented for now)
det0Rot	float	DET 0 rotation angle set in the simulation [deg]
det1Rot	float	DET 1 rotation angle set in the simulation [deg]
det0Dist	float	distance of DET 0 center from MC reference frame origin in (x,y) plane [cm]
det1Dist	float	distance of DET 1 center from MC reference frame origin in (x,y) plane [cm]
det0Z	float	z coordinate of DET 0 center in MC reference frame [cm]
det1Z	float	z coordinate of DET 1 center in MC reference frame [cm]
useTrueDriftTimes	bool	use true drift times for the fit (1), or smeared drift times (0)
generateNoise	bool	if true (1), generate randomly distributed noise hits

- Fit

Variable	Type	Description
minNHits	int	minimum number of hits accepted in a collection
maxNHits	int	maximum number of hits accepted in a collection
maxPattReclter	int	maximum number of fit iterations for Pattern Recognition (PR)
sigmaPR	float	3*sigmaPR [mm] set the threshold to reject hits after the PR
maxVertFitter	int	maximum number of fit iterations for Vertical Fit (VF)
sigmaVF	float	3*sigmaVF [mm] set the threshold to reject hits after the VF
maxDriftTimeFitter	int	maximum number of fit iterations for the Drift Time Fit (DTF)
sigmaDTF	float	3*sigmaDTF [mm] set the threshold to reject hits after the DTF
maxChi2MinimizationIter	int	maximum number of iterations for the Chi2 minimization procedure in the DTF

minDeltaChi2	float	minimum value of the Chi2 variation between consecutive iterations needed to stop the Chi2 minimization procedure in the DTF
reducedChi2Limit	float	threshold for Chi2/(n. points - 3): above this value, re-do the DTF with sigmaDTF/2

Preliminary settings: other input files

In order to analyze data, the following **.txt** files should be added in the **/utils/** directory:

- in **/DTfitter/utils/T0Map/**, a file containing the t0, tMin and tMax in ns calculated from the CMS SLs with this format:

```
[event ID] [t0 info] [t0 SL 0] [tMin SL 0] [tMax SL 0] [t0 SL 1] [tMin SL 1] [tMax SL 1]
```

where [t0 info] is set to MT if a mean timer is present in that event, otherwise to minSL if the minimum of the drift times of that event must be used. Run the **Preprocess** software with the flag 't0' to obtain a T0Map from un-processed SL data; run the macro **utils/readT0FromSL.C** to obtain the T0Map from processed SL data.

- in **/DTfitter/utils/NoiseMap/**, a file containing information on eventual noisy channels, with this format:

```
[detector] [board ID] [channel ID] [SW layer ID] [SW tube ID]
```

where [detector] could be either *DET0* or *DET1*.

Other files in the **/DTfitter/utils/** directory are:

- Space-Time_Parameters.txt**: it contains the parameters of the polynomial used to describe the relation between drift distance and the drift time. It should be updated any time a new dataset is analyzed.
- ChannelMapTubes.txt**: it contains the channel map, that has the following format:

```
[board ID] [channel ID] [HW layer ID] [HW tube ID] [channel FW] [SW layer ID] [SW tube ID]
```

- DeadChannels.txt**: it contains a list with the dead channels, with the same format of the noisy channels file.

Utility

- The macro **utils/readT0FromSL.C** is used to read the output tree files from the **PattRec** software and get the input t0 for the reconstruction of drift tubes track. How to run (3 possibilities):

```
root 'readT0FromSL.C(runNumber) '
root 'readT0FromSL.C(runNumber, nEvents) '
root 'readT0FromSL.C(runNumber, nEvents, filePath) '
```

In the first case, the number of events to be read from the SL is set to 100M. You can set a different path for the input files. The T0Map file will be saved in the **utils/T0Map/** directory.

- The macro **utils/readDTTree.C** can be used as a template to read the output tree files from the **DTfitter** software itself. How to run:

```
root 'readDTTree.C(filePathAndName) '
```

How to run

```
cd run
./runDT
```