



# **ILDPerformance package**

Alonso Guerrero Llorente, Yorgos Voutsinas

DESY FLC

[georgios.voutsinas@desy.de](mailto:georgios.voutsinas@desy.de)

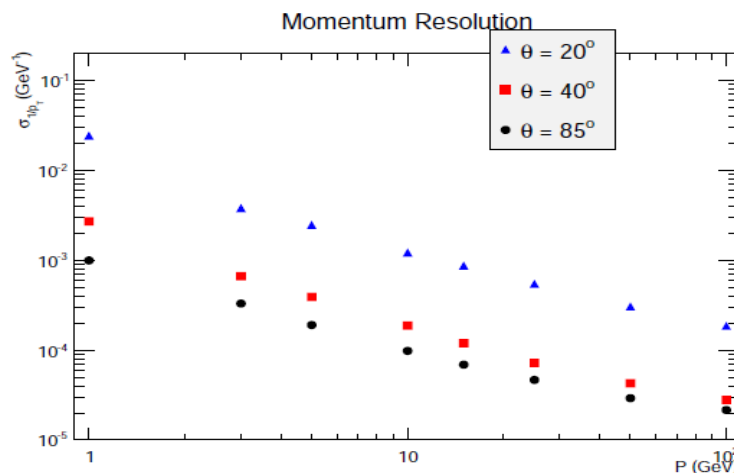
# Introduction

---

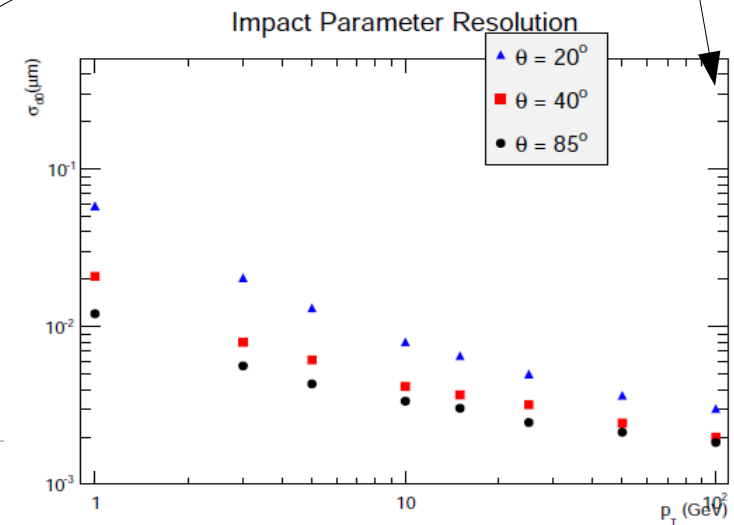
- Scope
  - Allow to evaluate the ILD tracking performance in an easy & fast way
    - Aid to detector optimisation studies / algorithm development
- Based on Diagnostics processor
  - It's run after the reconstruction
  - Input: studied track collection & relations from RecoMCTruthLinker processor
    - Processor parameters to define the examined MCParticle subsample
    - Processor parameters to define efficiency
      - Default > 75% purity &  $\geq 4$ hits in Si detectors
  - Output: ROOT file
    - ROOT tree with relevant info
      - True – reco track parameters & errors, track quality, # of hits, “bad” tracks etc
    - Canvases with track parameters residuals & pulls, finding efficiency
- svn co <https://svnsrv.desy.de/public/marlinreco/ILDPerformance/trunk>
- The user need to build the package and (for time being) export its library to MARLIN\_DLL variable

# Aid to optimisation studies

- Often it is very useful to study the momentum & impact parameter resolution, and compare them to the std DBD plots
- The user need to
  - `source tracking/scripts/runAll.sh`
  - It simulates, reconstructs (using the std reconstruction xml file) and analyses 1000 muon tracks for 8 momentum and 4 polar angle values
  - Runs the macros which produce the resolution plot
  - Initiate a ROOT gui
  - One can directly plot the resolutions
- One can do the above procedure step by step by using `runMokka.sh`, `runMarlin.sh` and `runAll.sh` scripts
  - For example, if wants to study the resolution using a different tracking algo. should modify the `stdreco_tracking.xml` and run `runMarlin.sh` & `runAll.sh`
    - Doesn't need to run simulation again



2014



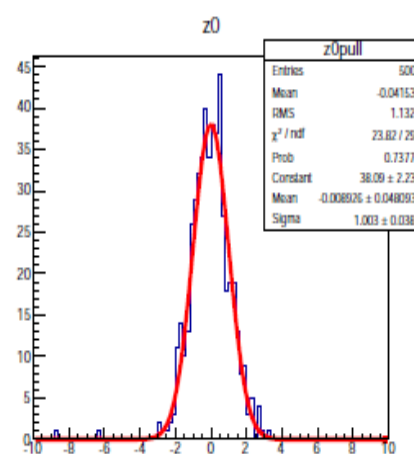
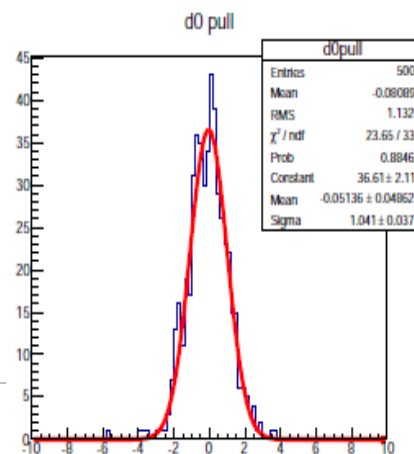
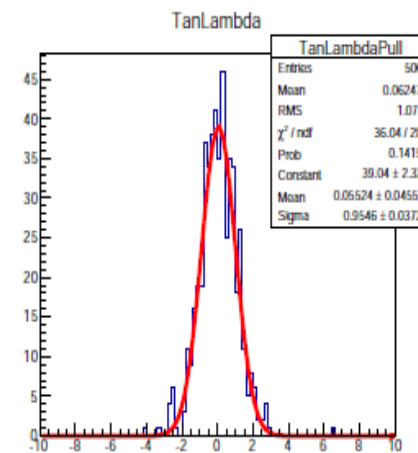
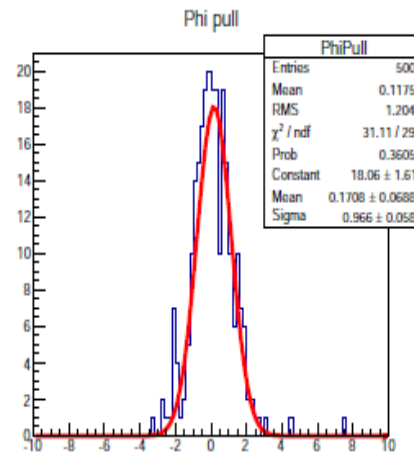
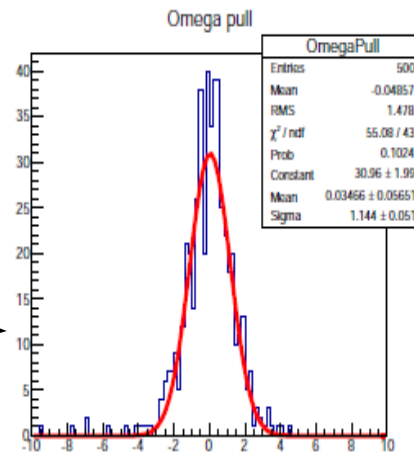
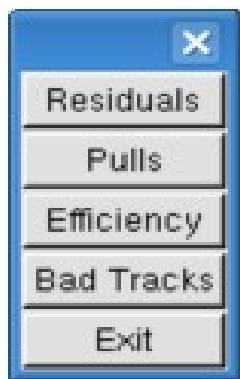
# Pattern recognition

---

- Evaluation of the efficiency / “bad” track rate of a new tracking algorithm
  - `source tracking/scripts/PatRec.sh`
  - Edit the script and assign to `INFILE` variable the relevant reconstructed `lcio` file
  - Turn the following Diagnostics processor parameters to true (`Diagnostics.xml`)
    - `PhysSampleOn`
    - `TrkEffOn`
  - One will be lead to a similar ROOT gui and plot efficiency or “bad” tracks as a function of `P`
- The script by default examines the full tracks
  - One can change track collection by appropriately editing the `RecoMCTruthLinker` & Diagnostics processor parameters inside `Diagnostics.xml`

# Track fitting studies

- Plotting of track parameters pulls and residuals
  - [tracking/scripts/PatRec.sh](#)
  - Give a reco lcio file with single tracks as INFILE
  - For example for 5 GeV muons at  $\theta = 85^\circ$



# Outlook

---

- Add a processor which calculates track – hit residuals
  - Pattern recognition & track fitting
- Add ROOT files with reference plots corresponding to DBD detector & software
  - Directly compare a new detector configuration or software tool with the standard ones