Science & Technology
Facilities Council

tkLayout - A Tracker Layout Modeling Tool

Jelena Ilic
STFC, Rutherford Appleton Laboratory

Intro

- What is tkLayout
- How it works
- Example(s)

Evaluation of tracker performances

Build tracker geometry?
Evaluate performances of a tracker?

detailed MC simulation

optimise event reco algorithms

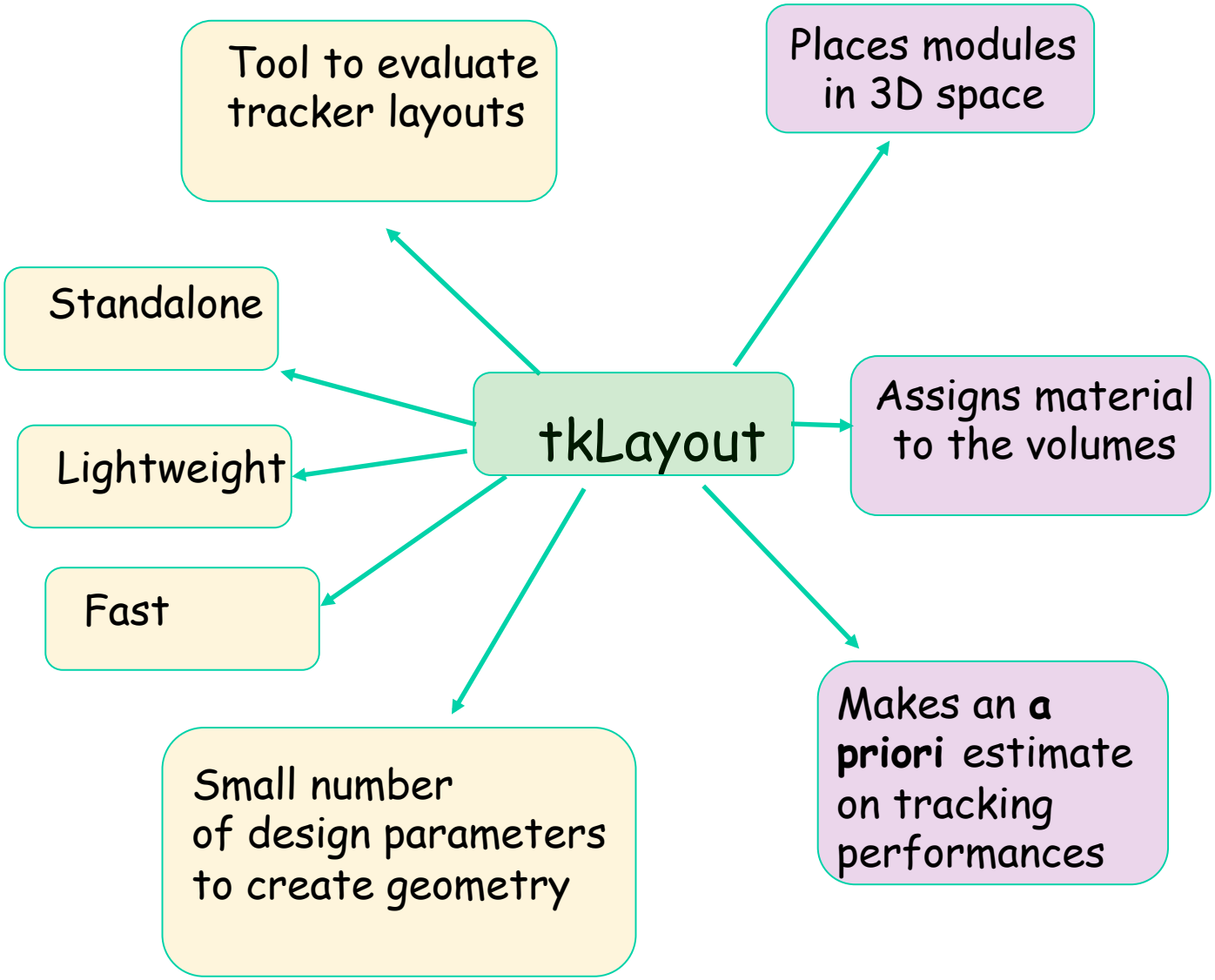
- ✓ Thorough
- ✓ Time consuming

fast-simplified geometry

Estimate the track parameter resolution from first principles

- ✓ tkLayout

What is tkLayout?



- Fair comparison of layouts
- a priori estimate of performance
- Narrow down the parameter space
- Pre-optimized designs
- Does not depend on optimised reco algorithms

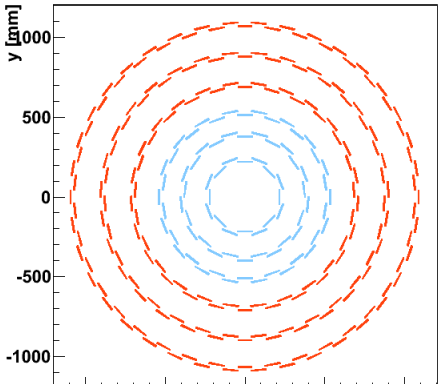
- IS NOT a replacement for the MC simulation
  - estimate impact on trigger
  - physics channels
  - occupancy
  - efficiency
  - ……

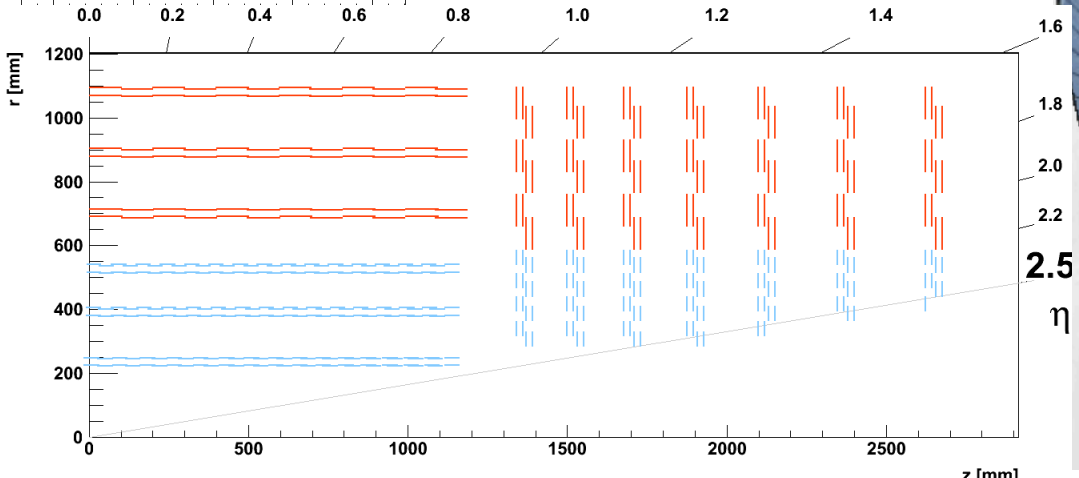
Developed by Nicoletta De Maio, Stefano Mersi

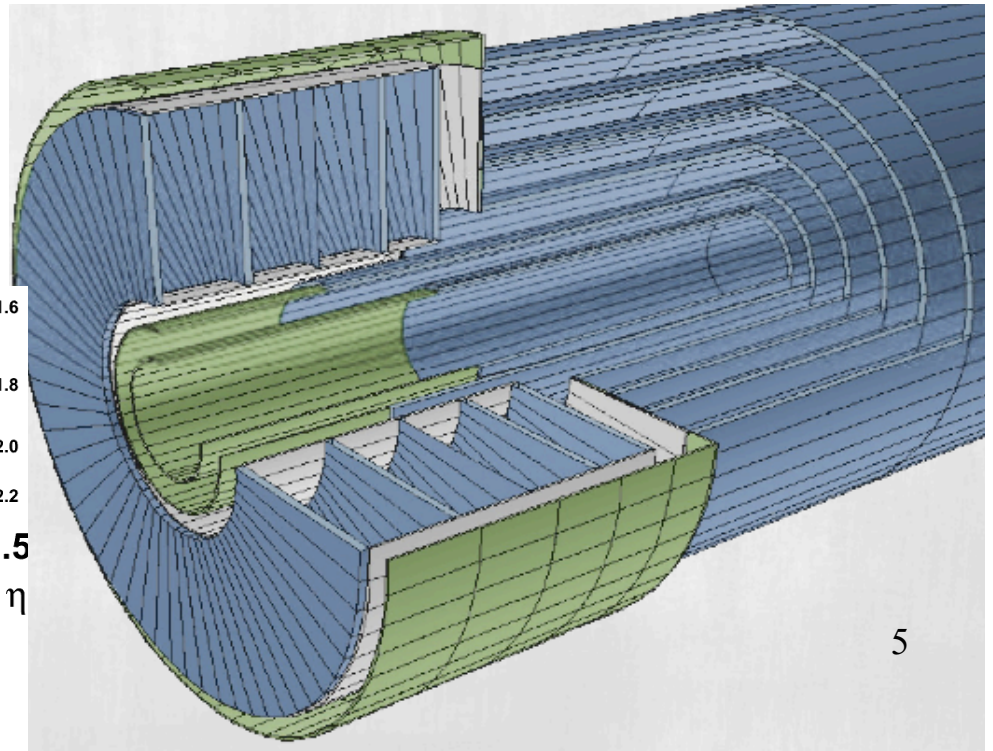
Step 1: Define Geometry

parameters:

- large-scale structure of tracker (number of layers/discs, volume boundaries)
- Details of modules used in the tracker (type of modules, no of sensors, distance between modules, size of trigger windows…)
- Support structures and services around the modules are, placed automatically (one can define additional support if needed)



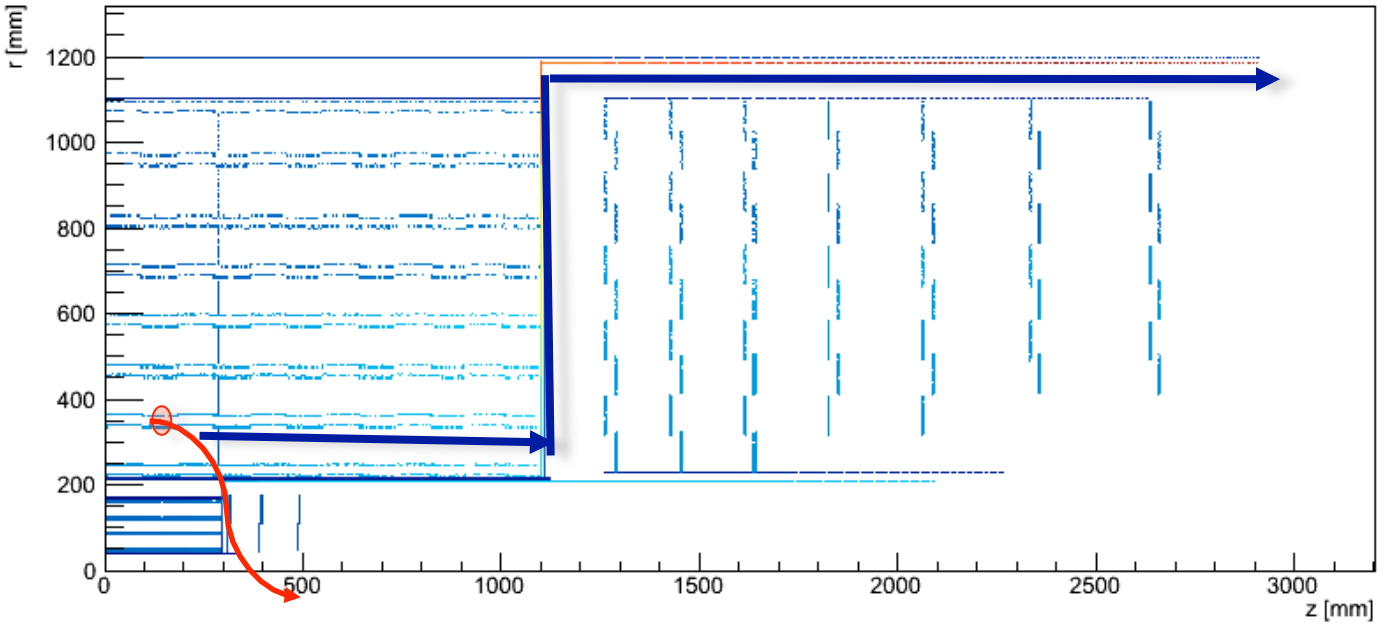




Step 2: Define Materials

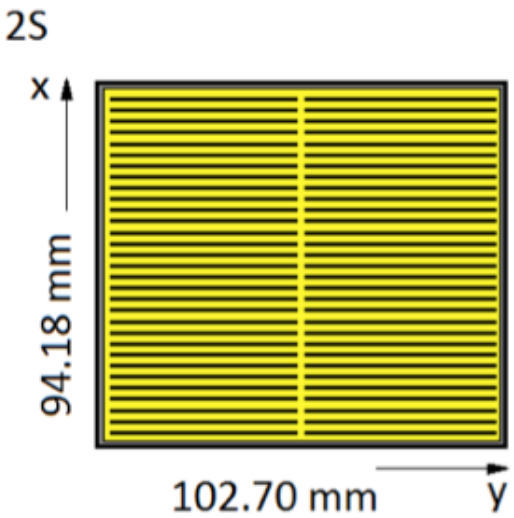
- the material file tells the application what the volumes are actually made of

Material assigned to a module without any detail about geometric distribution of material within the module itself

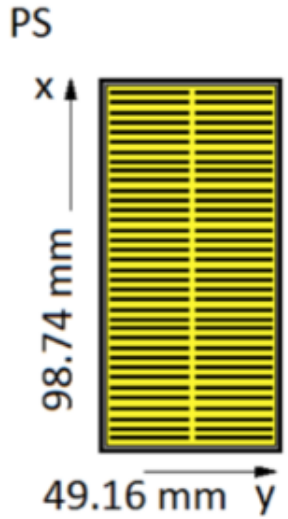


Material on active element + Material on services automatically routed

How it works? MODULES



Sensor: (94.18 x 102.70) mm
Active: (91.44 x 100.50) mm



Sensor: (98.74 x 49.16) mm
Active: (96.00 x 46.26) mm

- 2 strip sensors
- 960 strips x 2 segments
- long strip ~46mm
- 90 μm pitch
- ~1.5mm macro pixel
- 8 ROCs per segment
- p_T information

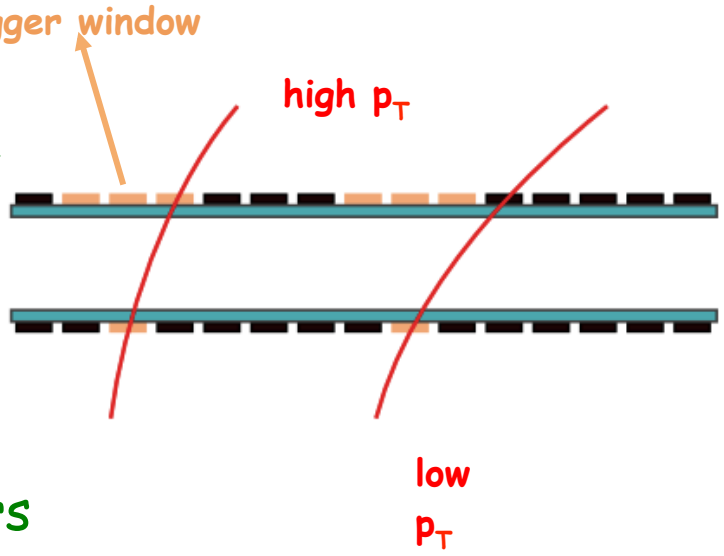
- 1 strip sensor, 1 pixel sensor
- 960 strips x 2 segments
- Short strip ~24 mm
- 960 x 16 pixels x 2 segments
- ~1.5mm macro pixel
- 8 ROCs per segment
- p_T + z information

How it works? MODULES-local p_T reconstruction

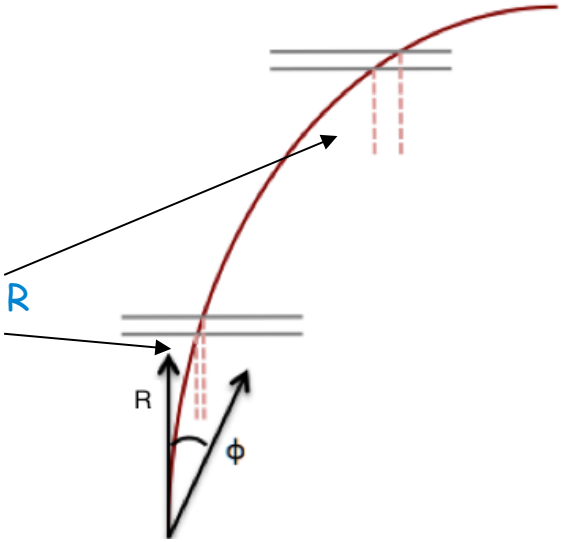
Low p_T track filtering

Measure the track crossing angle orthogonal to a layer's surface.

- The highest-p_T tracks will cross almost orthogonal to the surface
- The low-p_T tracks will cross at a wider angle
- The $R\varphi$ distance travelled between two sensors in a stack is of a similar size to the pitch of a single pixel
- Optimise trigger windows and (or) sensors spacing to obtain consistent p_T selection
- one of the parameters in tkLayout is *triggerWindowSize*



For a given p_T, $\Delta(R\varphi)$ increases with R



How it works? Performance Estimate

A priori error estimation

- No Monte Carlo
  - the accuracy of the track parameters derived from a fitting procedure
  - 2 uncorrelated fits: a circle in (r, φ), line in (r,z) plane
- No Fit actually done
  - minimimisation of χ^2 can be done analytically
- Ingredients:
  - Error propagation
  - Sensor resolution (measurement error)
  - Multiple scattering (treated as (correlated) measurement error)

Validation

- Detailed studies done by modeling current CMS tracker & comparing with full simulation
  - http://indico.cern.ch/event/113796/session/9/material/slides/0?contribId=46
- Layout studies
  - http://indico.cern.ch/event/153564/session/9/material/slides/0?contribId=36

How it works? Performance Estimate

3xPS_3x2S_5disks_baseline – Geometry

jilic.web.cern.ch/jilic/3xPS_3x2S_5disks_baseline/index.html

Bonjour razno Google Jelena Stribor

3xPS_3x2S_5disks_baseline

layouts

geometry | material (outer) | weights (outer) | resolution | resolution (trigger) | trigger | info | log page

layers and disks

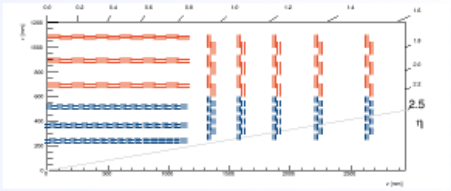
Layer	1	2	3	1	2	3	Total
r	230	357	508	686	888	1080	
# mod	1008	1320	1836	1152	1488	1824	8628
# rods	16	24	34	48	62	76	

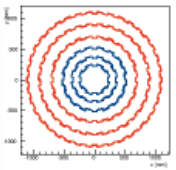
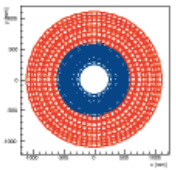
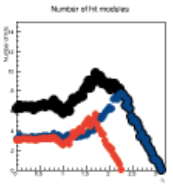
Disk	1	2	3	4	5	Total
z	1349	1597	1891	2239	2650	
# mod	680	680	680	680	680	6800

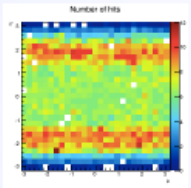
Ring	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
r_{min}	246	293	323	371	398	447	471	519	551	601	671	776	838	945	1000
r_{max}	292	339	369	417	444	494	517	566	597	701	771	876	939	1045	1100

modules

plots





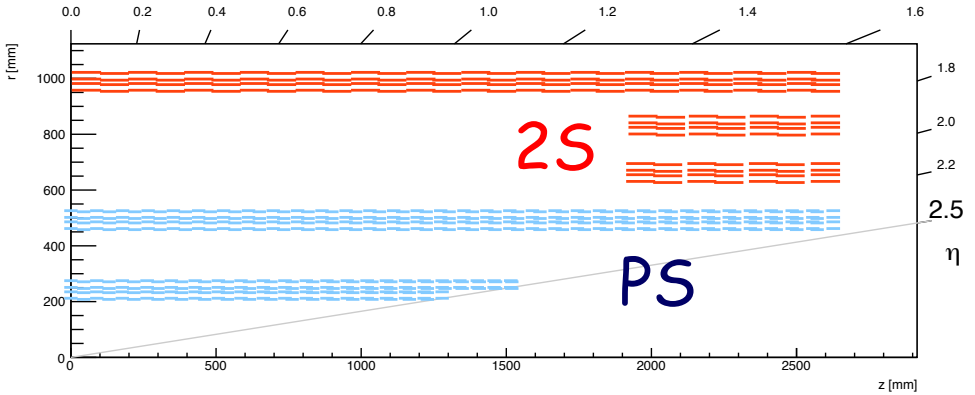
layer coverage

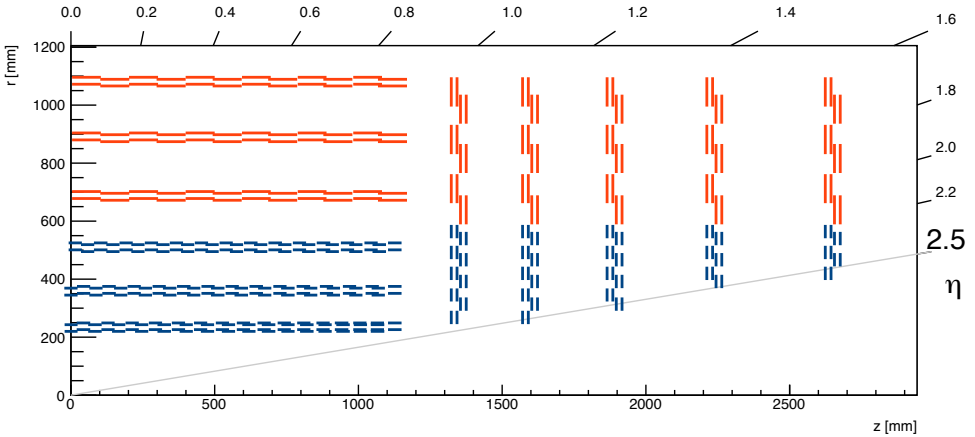
Examples

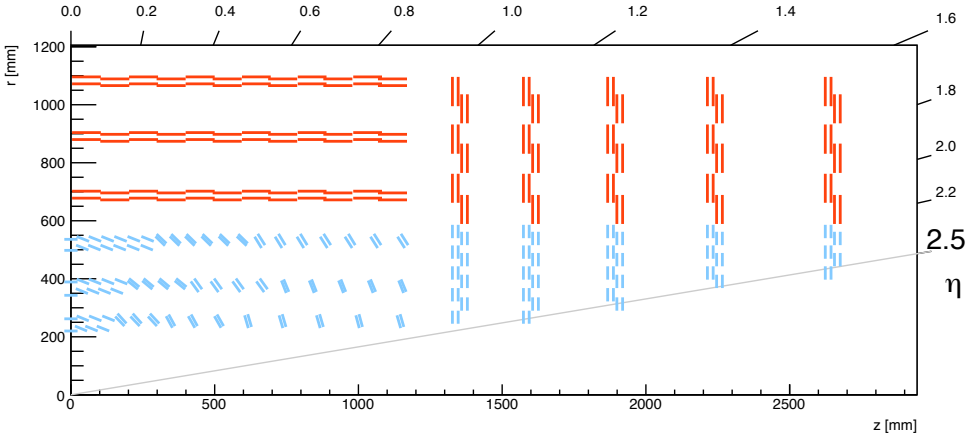
- LongBarrel
  (LB; LB_all PS)

- BarrelEndcap
  (BE_7; BE_5)

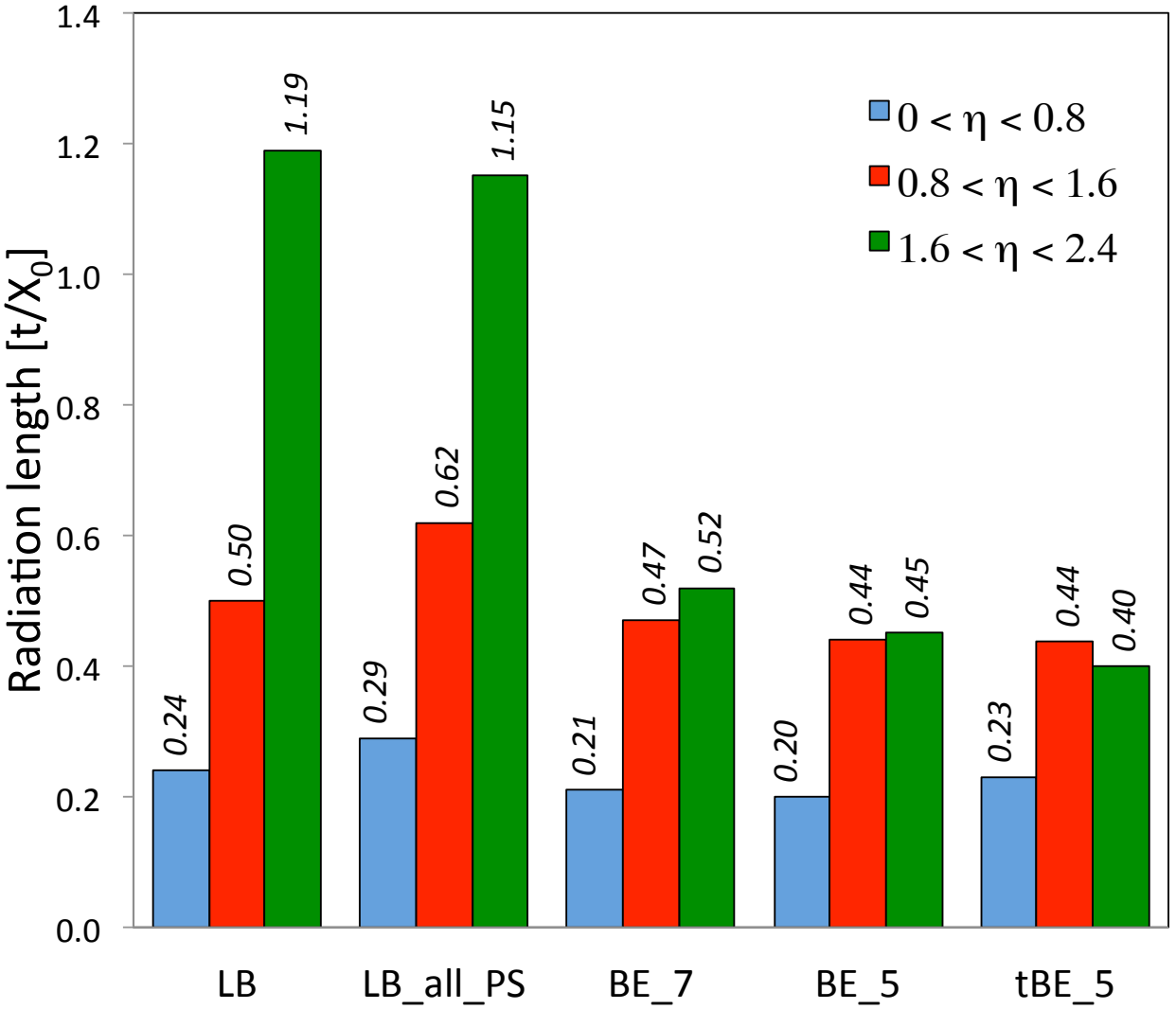
- tilted BarrelEndcap
  (tBE)





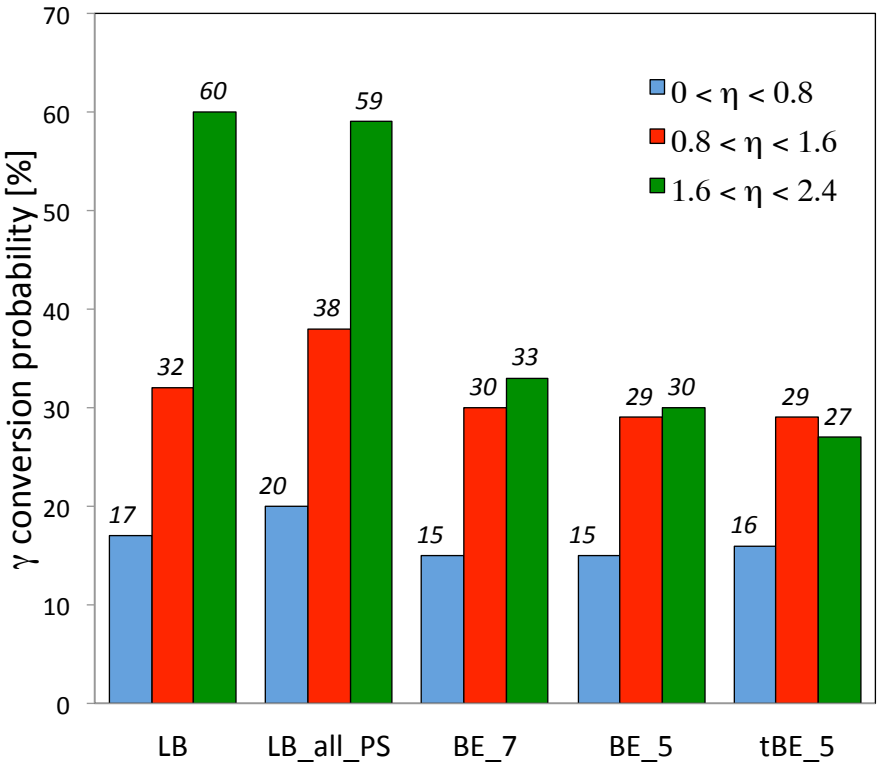


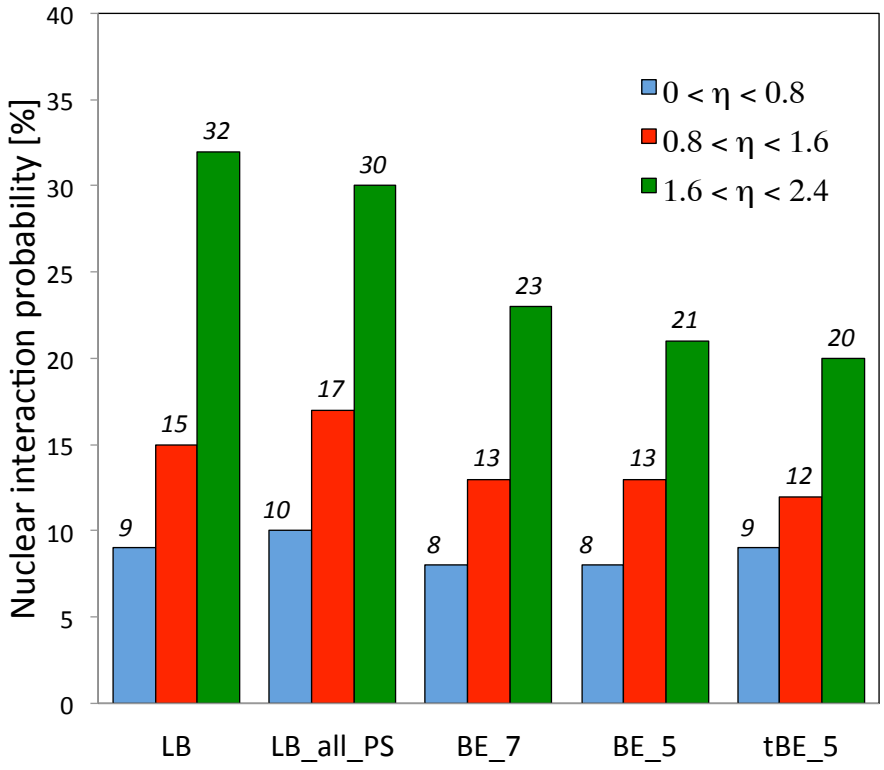
Material Budget



Examples

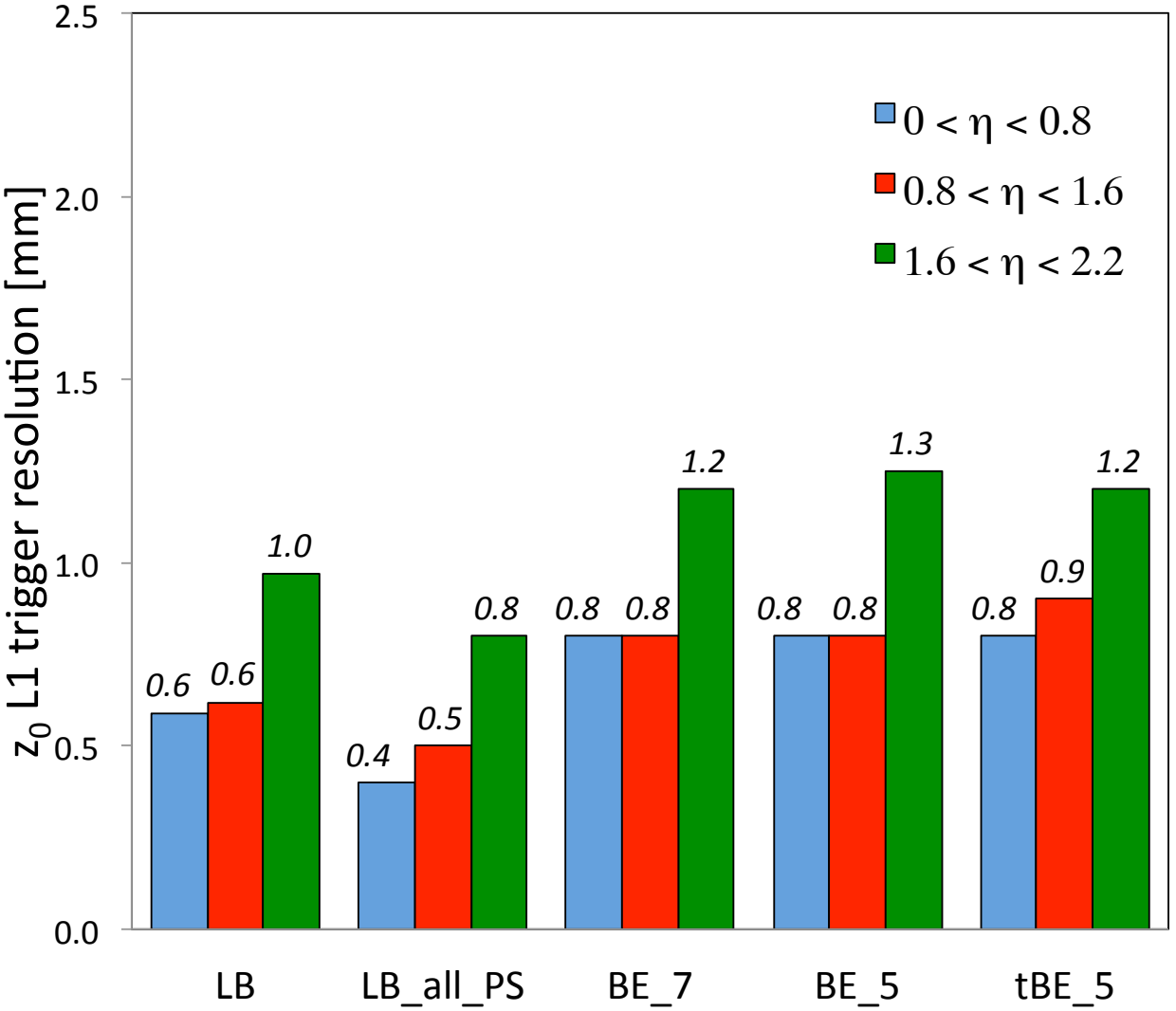
γ conversion probability & nuclear interaction probability





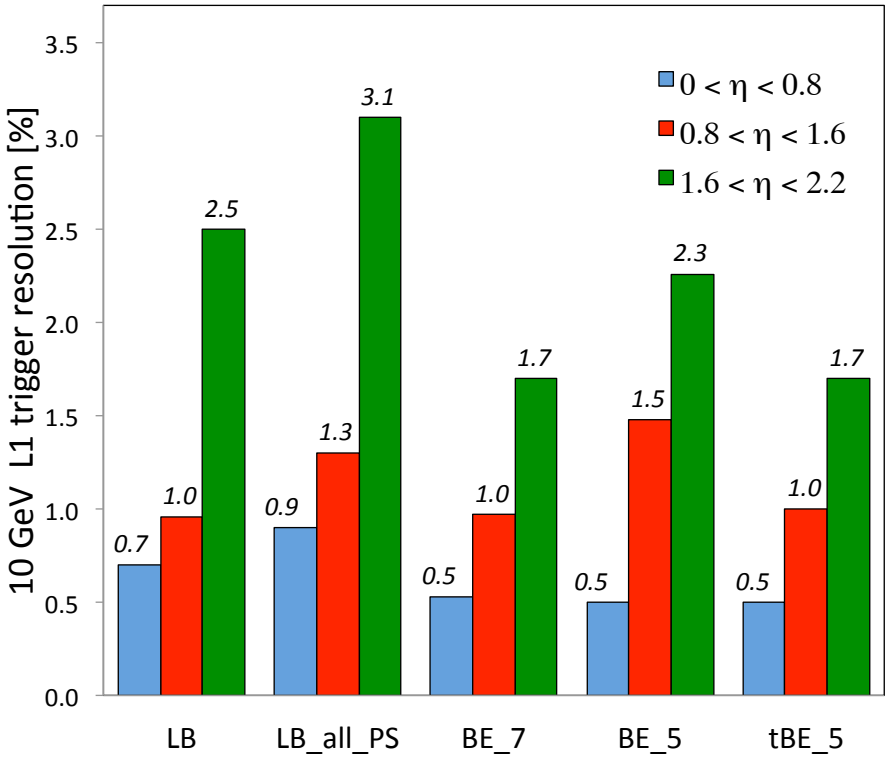
Examples

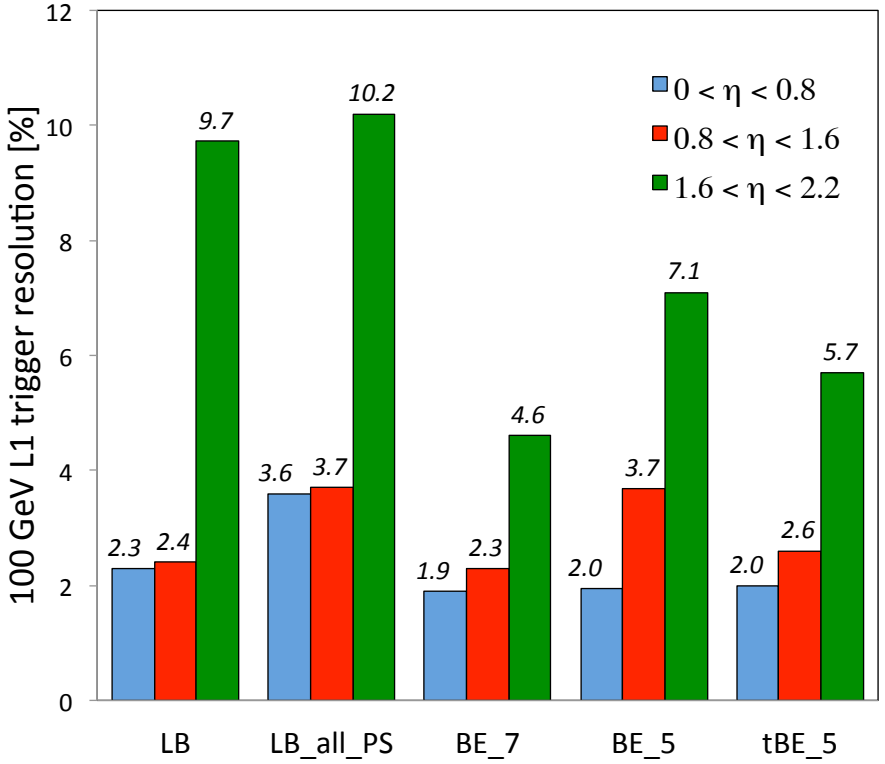
z_0 L1trigger resolution



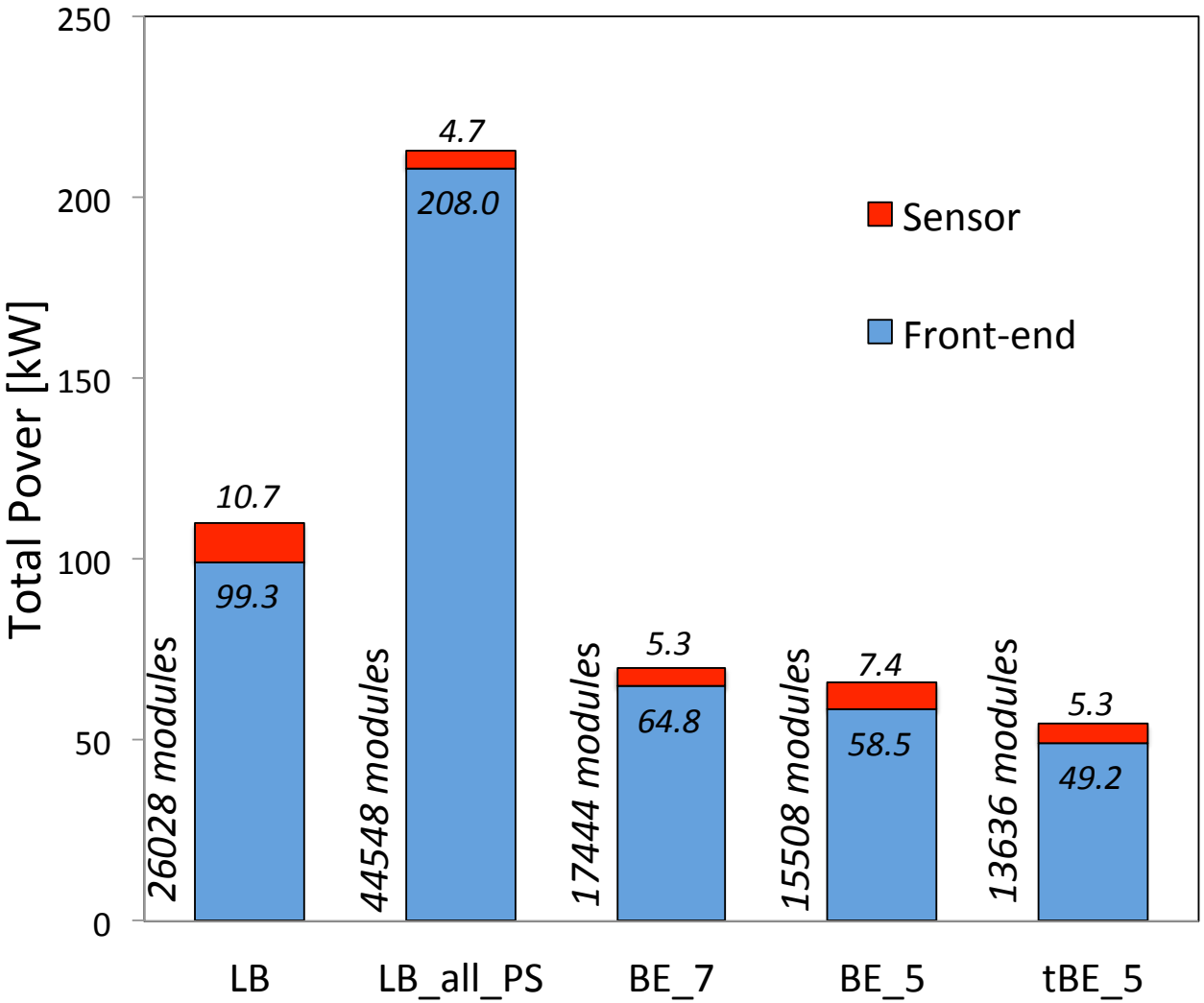
Examples

p_T L1trigger resolution





Examples


250
200
150
100
50
0
Total Pover [kW]
Sensor
Front-end
26028 modules
10.7
99.3
44548 modules
4.7
208.0
17444 modules
5.3
64.8
15508 modules
7.4
58.5
13636 modules
5.3
49.2
LB
LB_all_PS
BE_7
BE_5
tBE_5

Conclusion

- tkLayout is a free generic tool
- Fast running
- Simple
- Has been thoroughly validated
- No dependence on reco algorithm tuning
- Needs well understood model of materials to give good output
- Gives fair comparison between different geometry models
- Does not replace full simulation studies
- Produces geometry *xml*
- https://code.google.com/p/tkgeometry/source/checkout