IPATREC: inner detector pattern-recognition and track-fitting

Roger Clifft and ***Alan Poppleton***

1.0 Introduction

IPATREC is an inner detector track reconstruction package for ATLAS available within the framework of SLUG/ATRECON. It accepts events simulated using SLUG/DICE. Due to the high level of pile-up expected at LHC, reconstruction is restricted to 'regions of interest', i.e. to within roads joining the vertex region to 'trigger' seeds. The current program version takes its seeds from the electromagnetic calorimeter, and reconstructs from the digitizations in the barrel silicon detectors, SIT and SITV, plus the end-cap gas microstrip and Gallium Arsenide counters. It uses a combinatorial algorithm with a recursive track fitting and road definition procedure. Tracks are initiated from hits in the outer tracking layers to take advantage of the relatively low occupancy to reduce the combinatorial overhead. The results are expressed in a hierarchical structure of SEED/ROAD/TRAK/TFIT and HITS banks filled via window commons using the ATREBANK package [*SOFT-NO-002*].

The reconstruction has been tested with single electron and 'electromagnetically boosted' muons with up to 20 minimum bias pile-up events. Several studies of interesting physics channels have also started: $H^0 \rightarrow ZZ^*/ZZ \rightarrow 4e$, $A^0 \rightarrow \tau\tau$ and the identification of $t\bar{t}$ decays.

An initial version of IPATREC has been available within ATRECON since version 1.03/05 (02/05/94). Potential users are referred to the following section which gives some basic running instructions, plus a description of the output data structure. The subsequent section provides details of the program algorithms, and the final section illustrates some preliminary results from a comparison of the Cosenors and Panel tracking detector layouts that is currently in progress.

Further additions and improvements are under development (or are at least foreseen):

- Addition of the TRT 'continuous tracking' function. This should strengthen the pattern recognition capability and improve the track momentum resolution.
- Extra track-fit parameters to follow multiple scattering on low p_T tracks and bremsstrahlung 'kinks' on electrons [*see INDET-NO-015*]. These refinements are necessitated by the rather dense tracker material.
- Treatment of seeds from the external muon system, from narrow hadronic jets, and from stiff tracks found in the TRT stand-alone pattern recognition (XRECON).
- Secondary seed/road generation, required for the further analysis of electromagnetic calorimeter seeds. These restart the reconstruction with an enlarged road and/or relaxed tolerances to provide an electron veto for photon candidates and an e^+e^- pair veto to distinguish conversion/Dalitz backgrounds amongst electron candidates.

- Interface to a vertex fitting procedure. This will allow the association of multiple seed events to a common primary vertex. It is also intended to use this to tag secondary vertices from τ and b jets.
- Provision of some ‘standard’ histogram and NTUPLE output.

2.0 Running instructions: datacards and data-structures

2.1 Input

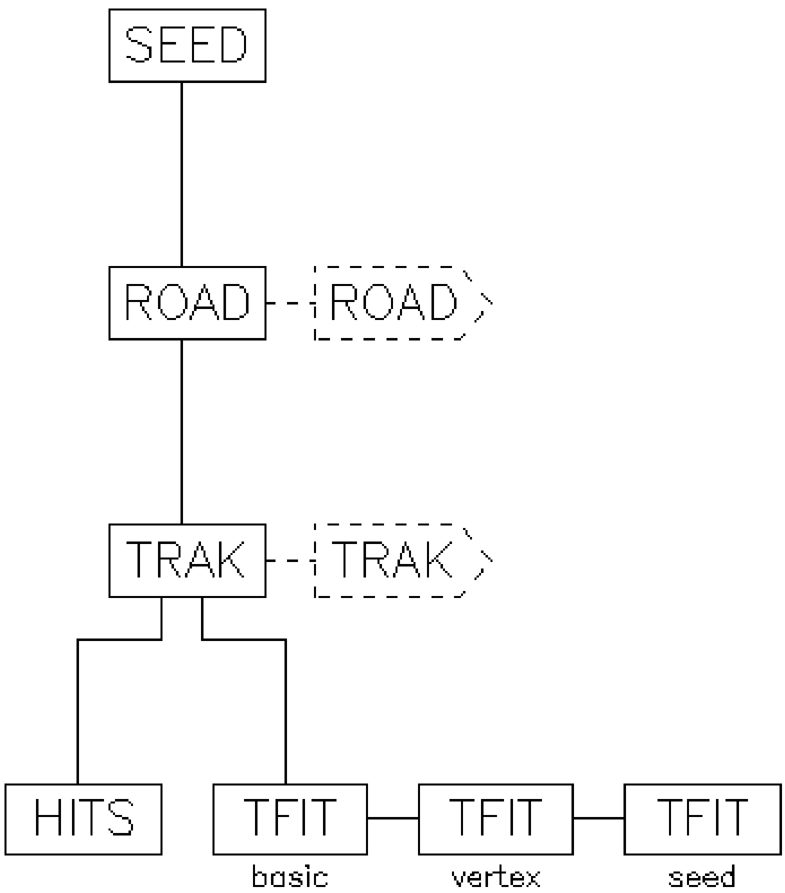
Typical SLUG datacards, relevant to the current package, might include:

```
C digitization and simulation and analysis status
SIMULATION   0
DIGI         0
RECONSTR     1
ANALYSIS     1
OUTP         0
*BKIO        'P'     'GEOM'
*BKIO        'P'     'KINE'
*BKIO        'P'     'DIGI'
*BKIO        'P'     'RUNT'
*BKIO        'P'     'EVNT'
C read events from ZEBRA file.
KINE        -1
STOP
LIST
C-------------------------------------------------------------------------C
C   Corrections to data cards
C   Other options
C    - Define analysis Et threshold for calorimeter em cluster finder
C-------------------------------------------------------------------------C
C  threshold for ARECON
 *DETA       'ECAL'   2=5.0    4=5.0
C print-out level and detectors used in IPATREC
*MODE       'IPAT'   'PRIN'   4        'RECO'   1        'HIST'   1
*MODE       'SITV'   'RECO'   1
*MODE       'SIT '   'RECO'   1
*MODE       'MSGC'   'RECO'   1
*MODE       'GAAS'   'RECO'   1
*MODE       'ECAL'   'PRIN'   0        'GRAP'   0        'RECO'   1
STOP
```

In this example, ATRECON is selected by the ‘**RECONSTR 1**’ card. The set of banks read from the ZEBRA input stream, via ‘***BKIO ‘P’** ’, is expected to contain the event simulation and digitization results from a previous job step (DICE). Reconstruction by the IPATREC module is requested by the ‘***MODE ‘IPAT’ PRIN’ 4 ‘RECO’ 1 ‘HIST’ 1**’ datacard, where the ‘**‘RECO’ 1**’ field is obligatory for module selection. The ‘**HIST**’ option is provided for implementation in a future version (where it will produce some standard histograms plus an NTUPLE for PAW analysis). The '**PRIN**' option allows for up to 6 levels of increasingly verbose print-out for debugging purposes. It is also necessary to '***MODE** ' select the tracking detectors to be used in reconstruction (SITV, SIT, MSGC and/or GaAs) plus any detectors providing 'trigger' seeds (ECAL in this case). IPATREC reconstruction is only performed in road(s) defined from these seed(s). For most layouts it makes no sense to run IPATREC without selecting at least 2 of the tracking detectors, otherwise many events terminate with diagnostics such as 'gives up as only 1 good superlayer(s)'.

2.2 Output

The output bank structure is created via 'window commons' using the ATREBANK package. The IPATREC results are expressed in a 4 level tree structure starting from a single header bank per event ('SEED'), as illustrated in the diagram below. This structure is appended to the standard ATRECON data-structure, such that the results can be retrieved by following the single-valued path: '/SECT/INNE/SEED'.



Details of each 'trigger' seed considered, along with its corresponding vertex region, transverse momenta and χ^2 cut-offs, are to be found in a horizontal chain of 'ROAD' banks. In turn each 'ROAD' supports a horizontal chain of 'TRAK' banks, one for each track found in the road. Each successful track has 3 types of track-fit stored in 'TFIT' banks and the associated tracking detector hits stored in the 'HITS' bank. There is a 'basic' fit to the associated hits, a 'vertex' fit which includes the transverse vertex as an additional measurement, and a 'seed' fit which includes the seed as a measurement. In the special case of electron candidates, the seed fit also invokes extra bremsstrahlung fit parameters.

2.2.1 SEED bank description

seed_TOTAL	# of seeds passed to track-finding
seed_ELECTRON	# of seeds which are single electron candidates
seed_EPAIR	# of seeds which are electron pair candidates
seed_PHOTON	# of seeds which are photon candidates
seed_MUON	# of seeds which are muon candidates
seed_TAU	# of seeds which are tau candidates

2.2.2 ROAD bank description

road_NUMB	current road number
road_TYPE	seed type (1=e, 2=e^+e^-, 3=γ, 4=μ, 5=τ-jet, 6=TRT stiff track)
road_INDEX	index of seed in appropriate data structure (according to type: calorimeter cluster, muon track, electron seed for e^+e^-, etc)
road_XSEED	x-coord of seed
road_YSEED	y-coord of seed
road_ZSEED	z-coord of seed
road_RSEED	r-coord of seed
road_DFSEE	σ(seed - rϕ direction) used in track fit
road_DRSEE	σ(seed - r direction) used in track fit
road_DZSEE	σ(seed - z direction) used in track fit
road_WTSEE	road half-width in transverse projection at seed
road_WLSEE	road half-width in longitudinal projection at seed
road_XVERT	x-coord of mean vertex position
road_YVERT	y-coord of mean vertex position
road_ZVERT	z-coord of mean vertex position
road_DXVER	σ(vertex x-coord) used in track fit
road_DYVER	σ(vertex y-coord) used in track fit
road_WTVER	road half-width in transverse projection at vertex
road_WLVER	road half-width in longitudinal projection at vertex
road_PTMIN	minimum p_T used in track-finding
road_CH2HI	χ^2/degf above which the worst measurement (highest residual) may be rejected. If the χ^2 remains high the track-fit is signalled bad.

2.2.3 TRAK bank description

trak_NHITS	# of hits associated (stored in HITS bank)
trak_NHOLES	# of 'holes' on track (hits expected but missing)
trak_PATTERN	hit pattern outwards from vertex (bitted word: 0=hole, 1=hit)
trak_BASEFIT	fit-code for basic fit (just to HITS)
trak_VERTFIT	fit-code for fit with vertex constraint
trak_SEEDFIT	fit-code for special seed-dependent fit

fit-codes :
- 0 = good fit - TFIT bank exists
- 1 = high χ^2 (i.e. > road_CH2HI) - TFIT bank exists
- 2 = fit not attempted }
- 3 = insufficient measurements }
- 4 = no convergence } thus no TFIT bank
- 5 = particle 'TRAPPED' in mag field }
- 6 = z-vertex outside road }
- 7 = transverse impact outside road }

2.2.4 TFIT bank description

tfit_NDEGF	# of degrees of freedom
tfit_NITER	# of track-fit iterations before convergence
tfit_IPBREM	measurement # for hard brem origin (electron fit)
tfit_IPWORST	measurement # with largest χ^2 contribution
tfit_CHISQ	fit χ^2 (per degree of freedom)
tfit_X0	}
tfit_Y0	} fitted vertex coordinates - at closest approach to
tfit_Z0	} input transverse vertex (road_XVERT, road_YVERT)
tfit_R0	}
tfit_A0	transverse impact parameter to (road_XVERT, road_YVERT)
tfit_FI0	fitted ϕ at vertex
tfit_THETA0	fitted θ at vertex
tfit_COSFI0	cos (tfit_FI0) }
tfit_SINFI0	sin (tfit_FI0) } transverse direction cosines of initial track direction
tfit_DZDR0	1/tan (tfit_THETA0) }
tfit_PTINV0	$1/p_T$ at vertex (signed with charge sign)
tfit_PTINV1	$1/p_T$ after hard brem (measurement #tfit_IPBREM)
tfit_ERR11 ...	} error (covariance) matrix on fit parameters:
... ERR55	} 1=A0, 2=Z0, 3=FI0, 4=THETA0, 5=PTINV0
tfit_ERR66	diagonal covariance for PTINV1 (electron fit)

2.2.5 HITS bank description

hits_IGEOM	row number of detector element in geometry table
hits_XCOOR	X coordinate of hit
hits_YCOOR	Y coordinate of hit
hits_ZCOOR	Z coordinate of hit
hits_EDEPO	energy deposition or pulse height
hits_DELTA	coordinate error (projected in clustering direction)
hits_CLWID	cluster width (# of strips)
hits_KINE	generated hit association (track id in GEANT KINE)

2.3 Program structure

The program is stored as patches in the ATRECON CMZ file. These patches include some program modules with a single entry point:

P=IPATREC Overall steering routine,

P=INIPAT Initialization and geometry table building,

P=ITRAKS Track finding algorithm within a given road,

P=IFITTER Track fitting,

plus some routines grouped according to function:

P=ISEEDS Seed definition routines,

P=IROADS Road building routines,

P=IUTILITY Utility routines possibly required from several places.

3.0 Program description (algorithms)

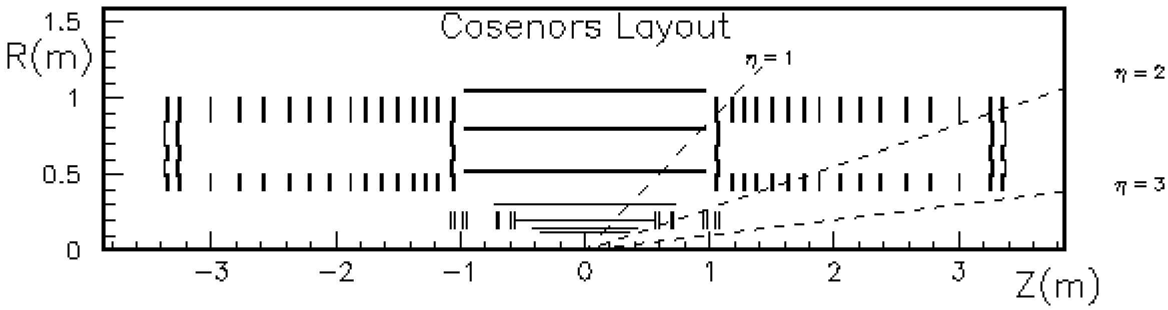
3.1 Detector representation: the 'Geometry Table' interface

The four discrete tracking detectors (SITV, SIT, MSGC and GaAs) are represented in GEANT using detector specific code introduced by DICE. The parameters relevant to each of these detectors are found in the corresponding banks:

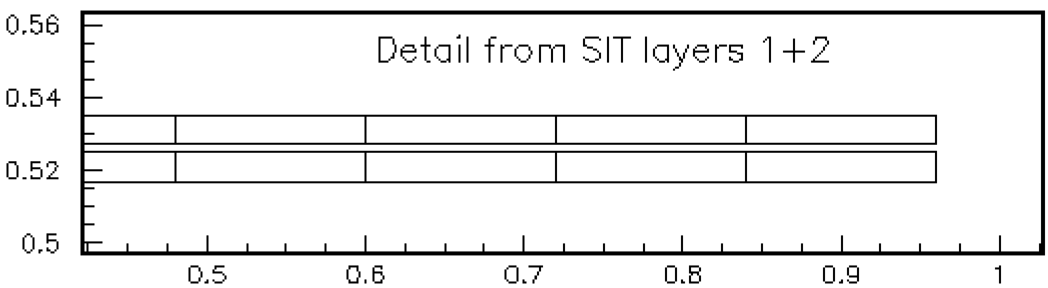
DETP - detector geometry parameters
DETD - digitization parameters
DETG - calculated geometry parameters

Note that these banks are defined differently for each detector (see eventual DICE write-up). However, this detailed detector knowledge is not really required during the reconstruction phase, and in fact only the decoding stage requires any detector dependent code. A general procedure, maintaining flexibility towards different detector layouts, has been ensured by collecting all the geometrical parameters required for reconstruction into a 'Geometry Table'. This 'Geometry Table' is created during an initialisation step from the DETP/DETD/DETG banks of the first input event. At the same time a few additional detector specific parameters necessary for decoding are stored separately as they don't conveniently fit into the table description. The subsequent processing then proceeds in a detector independent manner without further recourse to the DETP/DETD/DETG banks.

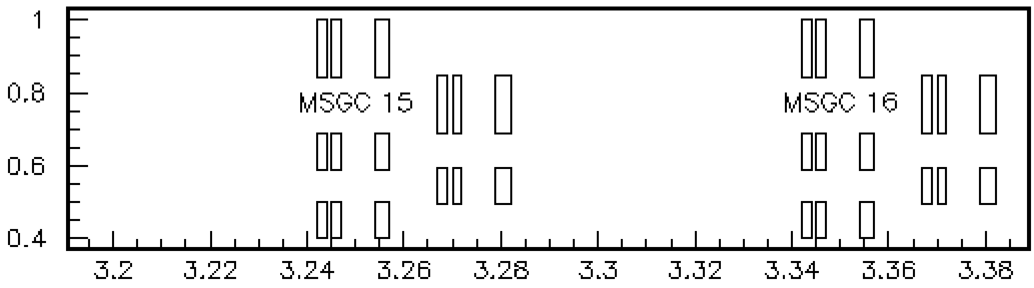
Since the detector is approximately ϕ symmetric, a simple description of differing detector layouts can be given by a line diagram in the R-Z plane, here illustrated using the Cosenors layout:



Each detector consists of line 'elements' which represent silicon crystals folded through 2π in the barrel, or MSGC/GaAs rings in the end-cap. Conceptually, each detector element corresponds to a row in the 'geometry table'. The lines are replaced by thin rectangles to take into account factors such as the tilt angle of the barrel detector for tiling and Lorenz angle compensation, as shown by zooming in onto part of the SIT detector:



The thickness represents the gas volume for the MSGCs:



For each detector ‘element’ the table contains the position, size, orientation, granularity (strip pitch, length and thickness), stereo and tilt angles, material thickness, details of the module arrangement in transverse projection and some decoding parameters. To facilitate the subsequent track-finding the elements are sorted: in order of increasing radius in the barrel, increasing |*z*| in each end-cap.

A further concept, that of superlayers, has also been introduced, where a superlayer is a set of neighbouring elements, sufficient to define a space point. Elements are assigned to superlayers by the program: thus the elements in SIT layers 1+2 form one superlayer, while the elements of MSGC15 and MSGC16 make 2 separate superlayers (see above diagrams). The Cosenors and Panel layouts each consist of around 500 elements which get assigned to 47 and 32 superlayers respectively.

3.2 Seeds and roads

The reconstruction is driven by seeds derived from ‘triggers’ occurring in the different subdetectors of ATLAS. Each seed is processed in turn. The philosophy is to find not only ‘the’ track (if any) creating the seed, but also any other nearby tracks which may affect subsequent physics analysis. For each seed a road connects the vertex and seed regions. The latter is the region of interest for future analysis (such as isolation criteria) rather than the intrinsic precision of the seed. In the same spirit there is a low p_T cutoff appropriate to the seed type. Initially the road-centre is simply a line joining vertex to seed. It is represented by the result of a helix fit at the vertex, since this allows the road-width at any radius to be obtained by error propagation.

After road definition the first task is to produce an ordered list of detector elements which intersect the road. This is stored in common /iRdParm/ along with other parameters of the road. The digitizations in these elements are decoded, with adjacent strip clustering, and any clusters in the road are kept in the /iRdClus/ common block.

3.3 Pattern recognition algorithm

Track finding starts by initiating a track ‘skeleton’, which is a fit to the road parameters (i.e. seed and vertex regions) plus 2 space points (from different superlayers). To account for moderate detector inefficiencies, 3 or 4 superlayers are designated to provide these space points, normally 2 at the outer tracker radius (where the road width and occupancy give the least pile-up combinatorials) and 2 in the ‘sagitta’ region (mid-radius) to give sharp p_T thresholds and to provide a reasonable extrapolation precision. The superlayer choice takes into account the degree of road ‘containment’ and only considers superlayers with space point candidates in the road. All combinatorials amongst these superlayers are taken as ‘skeleton’ tracks provided they pass checks on fit quality, road parameter consistency and track uniqueness.

The ‘skeleton’ tracks are then built into track ‘segments’ by associating the closest measurements from intermediate elements in a redefined (narrow) road. In turn a ‘segment’ fit again redefines a road to restrict space point finding in the high occupancy inner layers. Finally tracks are built by picking up measurements by interpolation in a narrow road throughout the tracker. The segmentation procedure allows for deviations from a pure helical trajectory due to material effects - such as bremsstrahlung and multiple scattering.

The checks on track quality are grouped in Subroutine itChoos. They include cuts on fit χ^2, on the fitted p_T and vertex z-coordinate (road consistency), on a minimum number of associated hit clusters, on maximum numbers of missing clusters (elements intersecting the track with no hit found) and clusters shared with higher quality tracks. Track ambiguities are resolved by sorting on track quality (by definition the sum of the fit χ^2 and the number of missing clusters), then keeping only those tracks with no ‘skeleton’ cluster shared with a higher quality track.

3.4 Track fitting

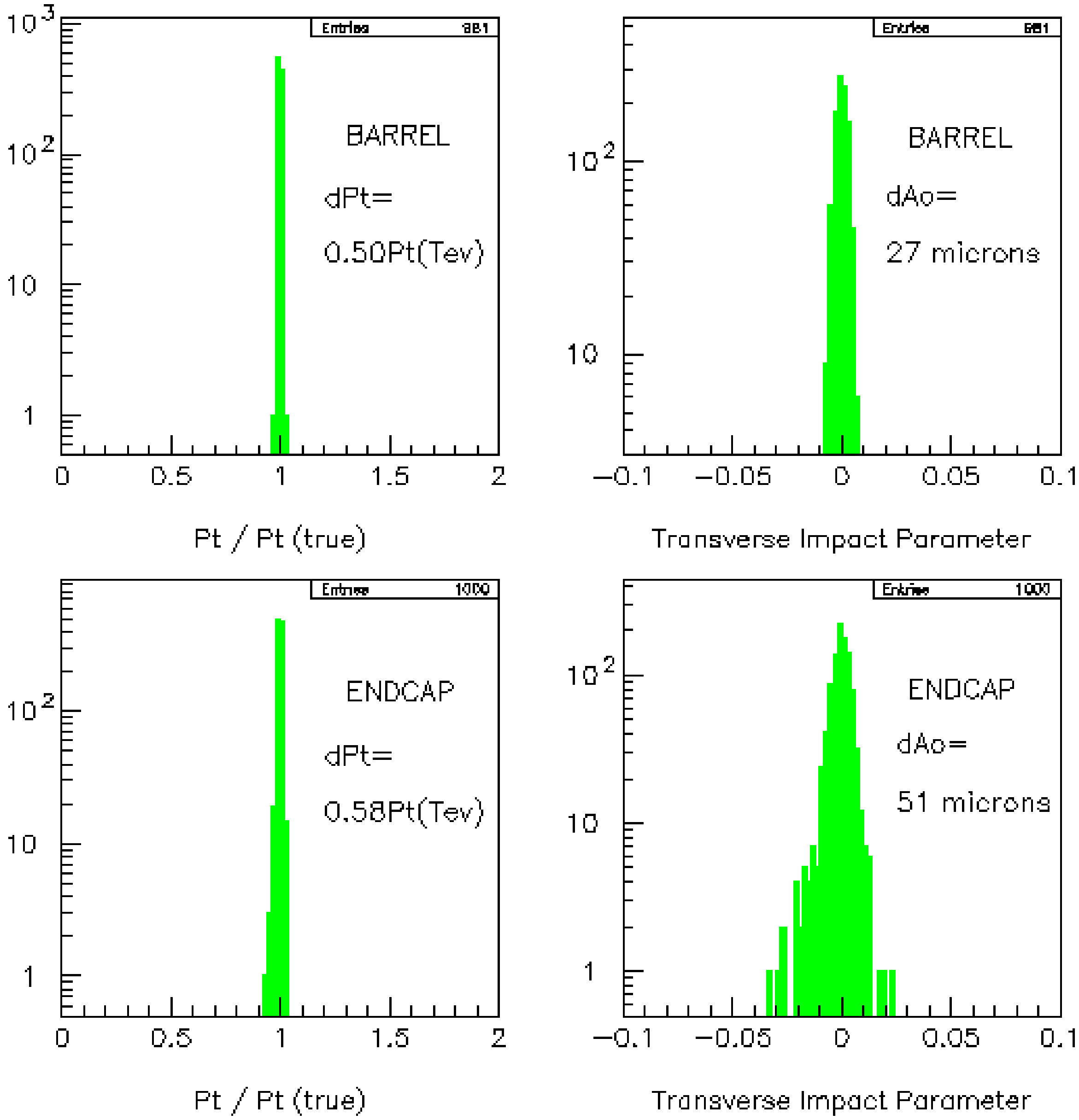
Tracks are expressed in terms of helix parameters which are specified at the closest point of approach to the road-defined transverse vertex. These parameters are stored in the TFIT window common (see section 2.2.4). A first approximation is given by taking a 3 point circle in the transverse projection with the line joining 2 of these points in the rz-plane. The subsequent track fitting uses an iterative tracking algorithm with a least squared parameter correction obtained from the first derivatives of the track residual(s) at each hit computed with respect to the helix parameters. Analytical derivatives are taken for high energy tracks in a constant magnetic field. In other cases numerical derivatives are computed. The hits are assumed to be cluster centroids on planar detector elements described in the ‘Geometry Table’. These elements are foreseen to be arranged, apart from a possible small angle (tilt), with either constant r (barrel type) or constant z (disk type). There are 3 non-correlated error components: a component in the direction orthogonal to the detector plane (the element thickness), plus components in the detector plane perpendicular and parallel to a direction obtained by rotating the $r\phi$ projection by the stereo angle. For each hit, the quantity to be minimized is the distance between the cluster centroid and the intercept of the fitted helix trajectory with the detector plane. Hits may have both the perpendicular and parallel components minimized (as appropriate for pad and strip detectors), otherwise only the perpendicular component is considered.

The tracking assumes a constant magnetic field in the z-direction inside a cylindrical ‘coil’ volume, with zero field outside. The track parameters are propagated to any requested r or z value by an execution-speed optimized subroutine which approximates the circular transverse trajectory by a parabola at high p_T to avoid rounding errors. As the solenoidal field in the inner detector is not expected to deviate far from a constant field, it is hoped that it may be possible to extend this approach to a more realistic configuration by the addition of higher order terms, avoiding expensive integration procedures such as the Runge-Kutta method.

4.0 Applications

The program has been used for some initial studies of the Cosenors House and Panel default layouts, using single muons and electrons. The parameters A_0 (the transverse impact parameter) and p_T(fit) / p_T(true), derived from the track fits, are used to show the main features of the studies. The former parameter gives a measure of the track fit accuracy at the vertex, without any vertex constraint. The latter ratio, which includes the vertex constraint, measures the momentum resolution achievable.

Difference between Barrel and Endcap
for Muons
Cosenors House Layout , NO Multiple Scattering

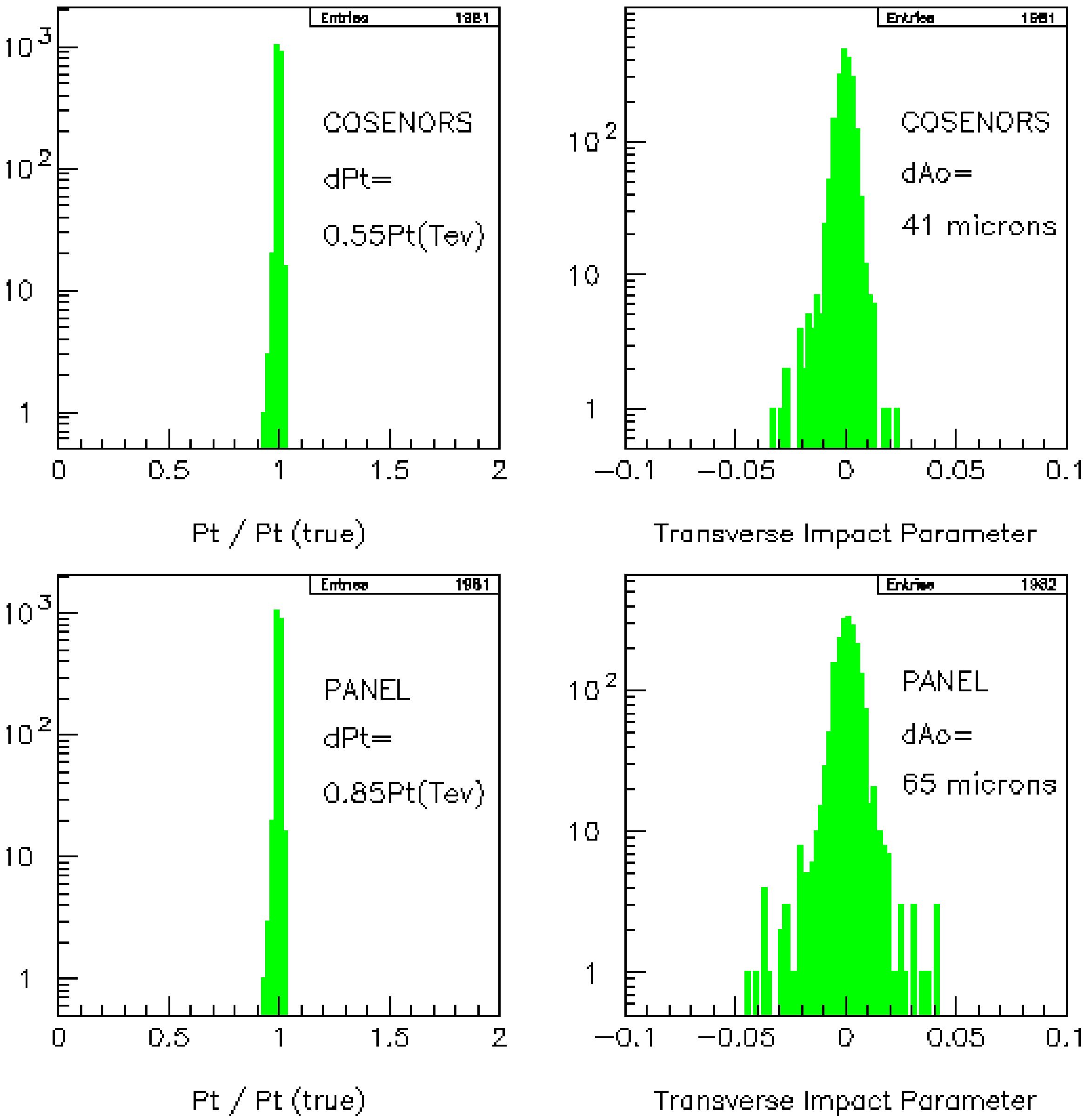


Muon triggers (seeds) were created as electromagnetic clusters by enhancing the energy deposition of the muons in the calorimeter. To test the correctness of the program 20Gev muons were generated with all physics processes turned off. The barrel and end-cap performances are contrasted in the preceding figure. It is seen that the barrel is more precise than the end-caps for impact parameter physics, whereas barrel and end-caps are reasonably well balanced for momentum resolution measurement.

The following figure shows that both impact parameter and p_T resolution are better with the Cosenors House design at higher p_T:

Difference between Cosenors House and Panel Layouts

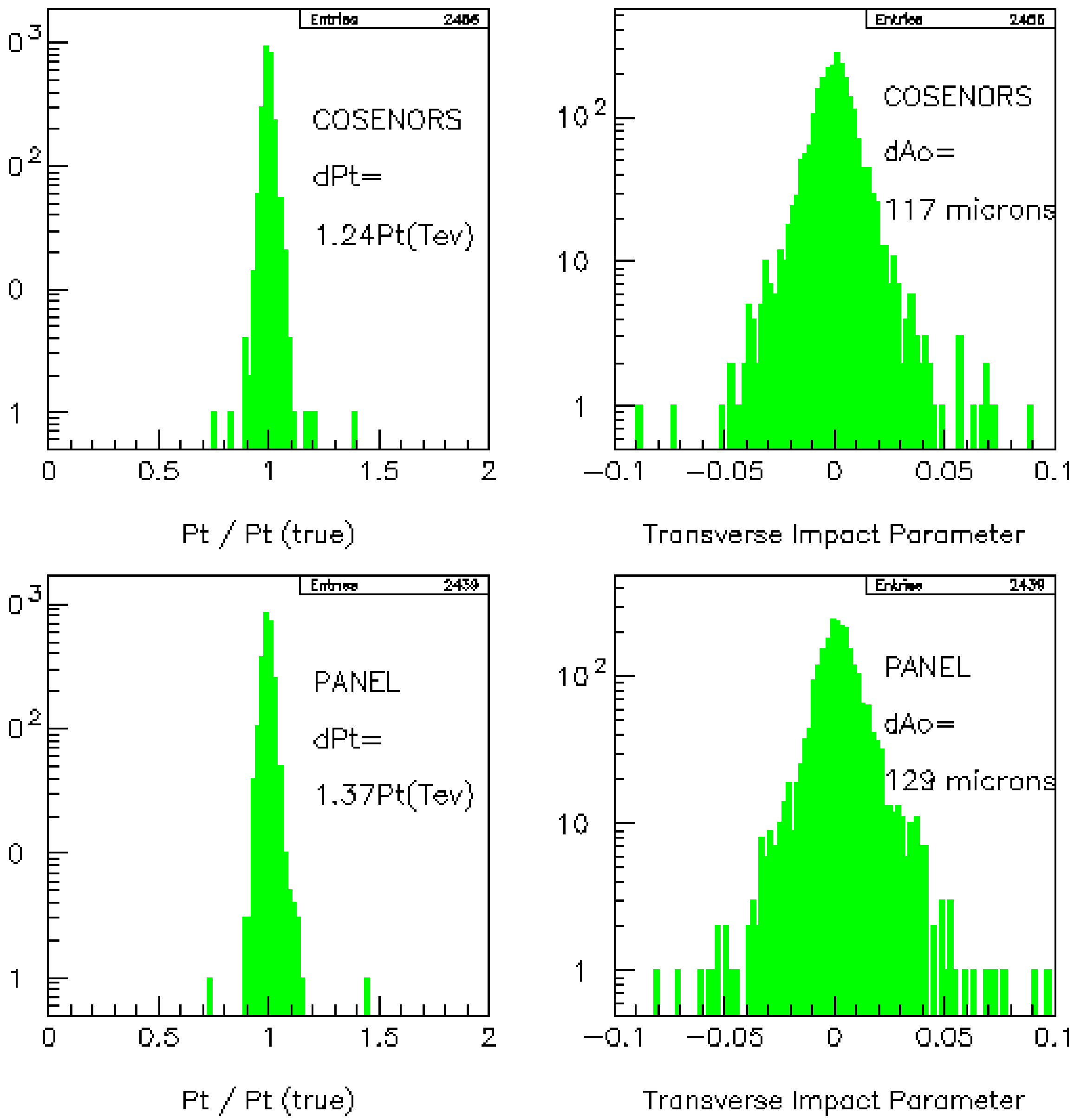
Muons : NO Multiple Scattering



Minimum values for $\Delta p_T/p_T$ of $0.55p_T$ (Tev), and $0.85p_T$ (Tev), are found for the Cosenors House and Panel layouts respectively. Of course the addition of precision TRT measurements should reduce this difference.

Muons of 7Gev, with all physics processes turned on, provide the limiting case of minimum momentum, enabling the effects of multiple scattering along the track to be studied (see following figure). Below ~20Gev, the effects of material give rise to multiple scattering dominated transverse impact parameters, whereas the p_T resolution is dominated by multiple scattering up to ~40Gev.

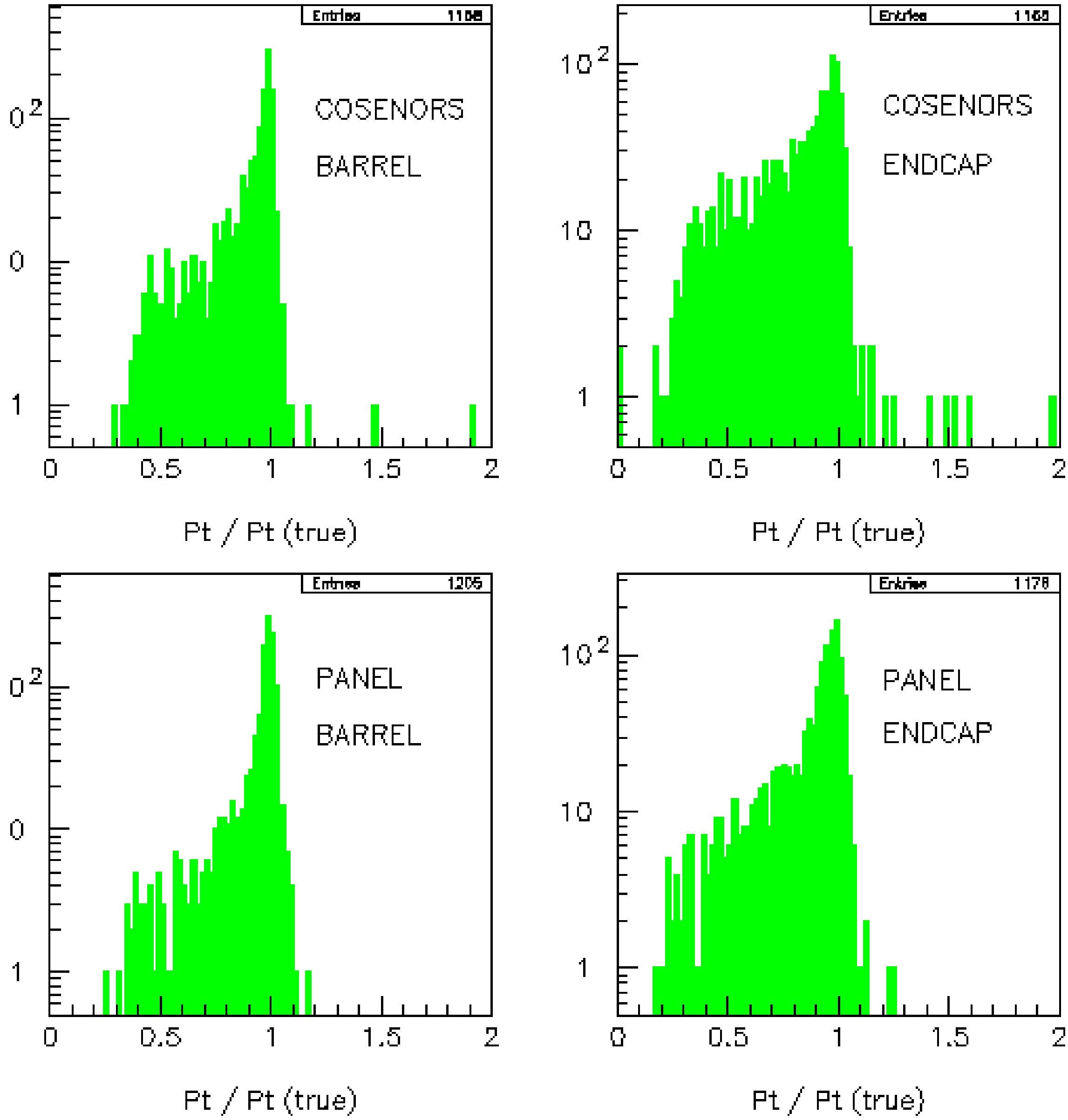
Difference between Cosenors House and Panel Layouts
Muons : WITH Multiple Scattering



Further studies using electrons of 20Gev show the effects of bremsstrahlung on the track fits as shown below. Methods to minimise the resulting tails have been developed (*INDET-NO-015*), but are not included in the current code. They will appear in the next release. It is seen that the end-caps give a worse resolution than the barrel, and that the Cosenors House layout gives a significantly worse p_T resolution than the Panel one, probably due to the relative material position along the tracks.

Difference between Cosenors House and Panel Layouts

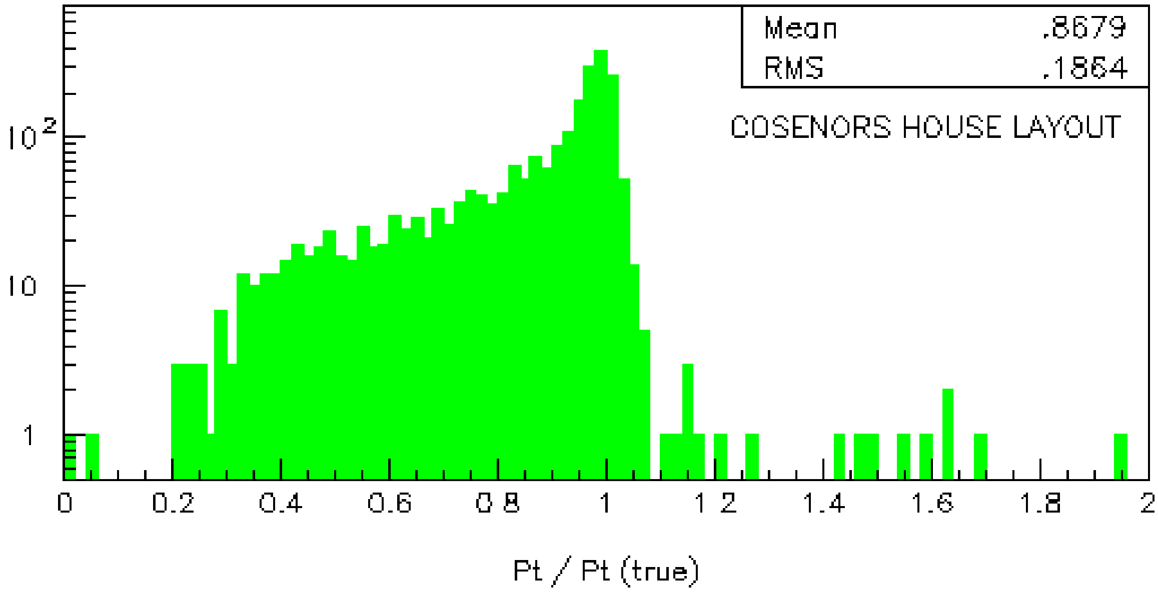
Electrons

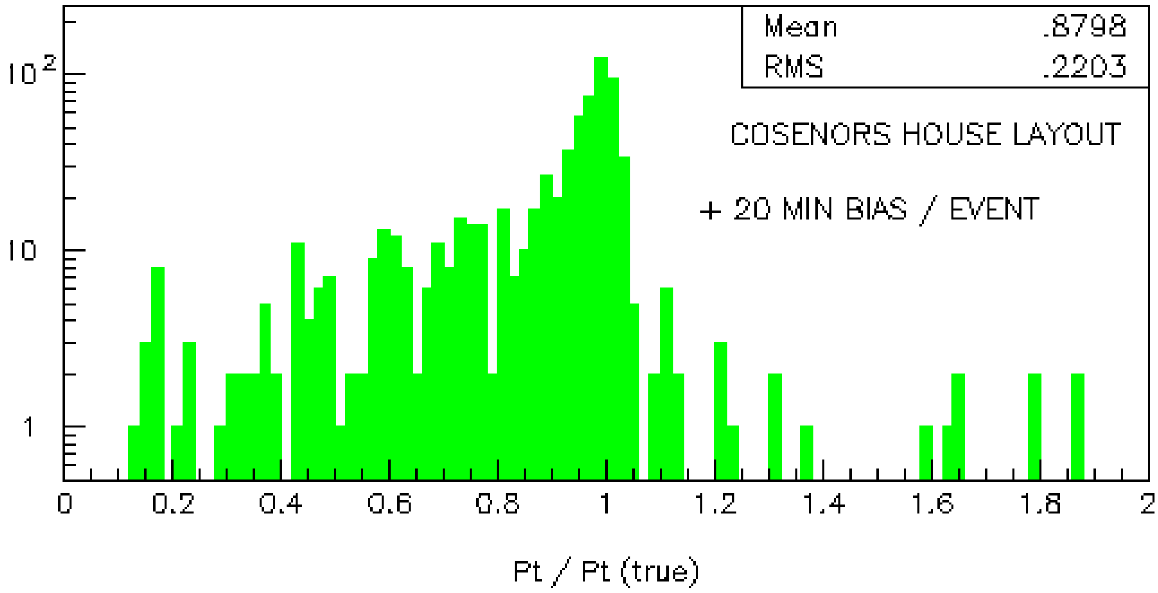


Finally, a brief look at the effects of pile up on 20Gev electrons is shown below. A mean of 20 minimum bias events per electron was used. Within the limited statistics available, this initial result suggests that pile up will not be a serious problem.

Effect of pile-up

Electrons : Cosenors House Layout





All runs made with muons, give a track finding inefficiency per trigger of ~1%, dominantly caused by the alignment of the MSGC frames in ϕ. For electrons the inefficiencies are of the order of 5% at 20 Gev due to bremsstrahlung. This will be reduced when the special electron fit is introduced.