

LumiCal LuCaS Simulation and Cluster Position Reconstruction

Oleksandr Borysov
Tel Aviv University

FCAL Clustering WG Meeting
March 16, 2015

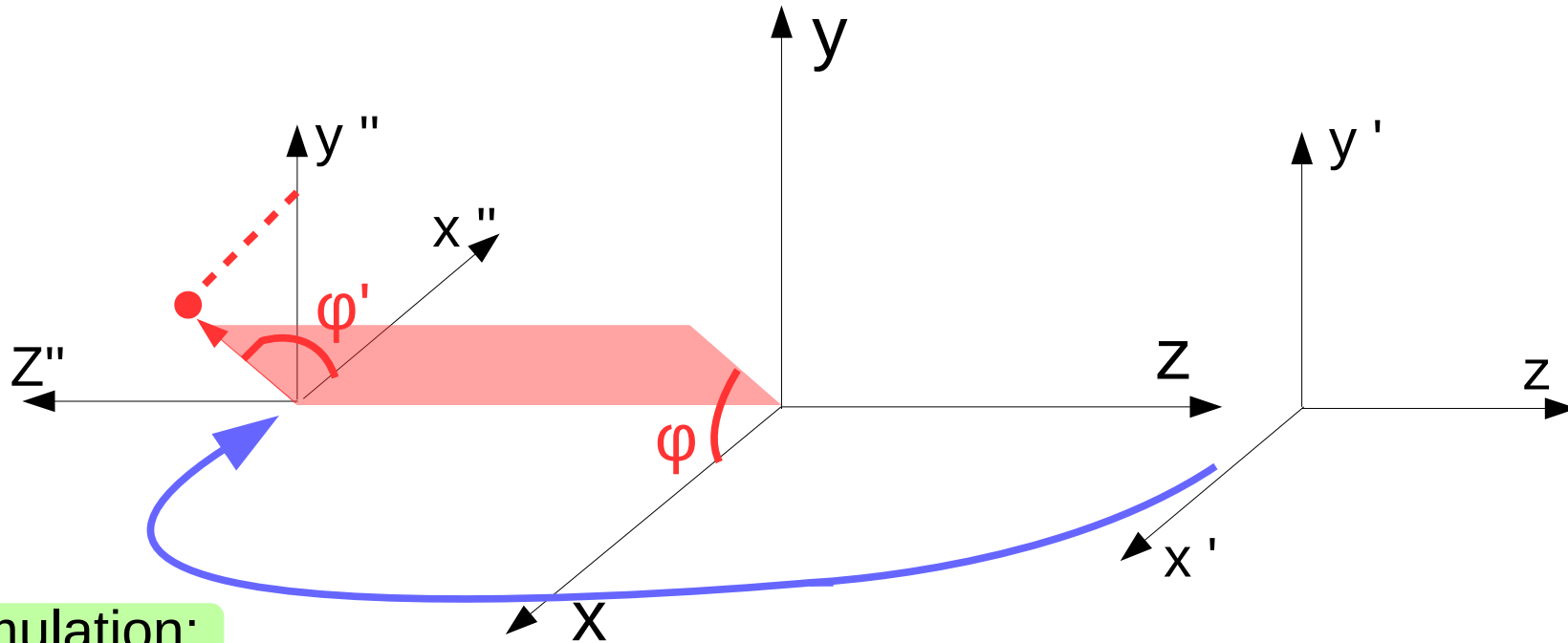
LumiCal Construction in Simulation

```
rotAng      = Setup::Beam_Crossing_Angle / 2.;
rotAng1     = 180.*deg - rotAng;
rotAng2     = rotAng;
.....
G4Transform3D trans1( G4RotationMatrix().rotateY(rotAng1),
                     G4ThreeVector( 0., 0., zpos).rotateY(rotAng1));
G4Transform3D trans2( G4RotationMatrix().rotateY(rotAng2),
                     G4ThreeVector( 0., 0., zpos).rotateY(rotAng2));

new G4PVPlacement( trans1 ,
                  logicWholeLC,
                  "LumiCalDetector1",
                  logicWorld,
                  false,
                  1);

new G4PVPlacement( trans2,
                  logicWholeLC,
                  "LumiCalDetector2",
                  logicWorld,
                  false,
                  2);
```

LuCaS – Reconstruction Coordinates Mismatch



Simulation:

Global hit coordinate

local coordinate in
LumiCal arm

R', ϕ'

Cell radial ID,
Cell azimuthal ID

Reconstruction

Global hit coordinate

R, ϕ

Cell radial ID,
Cell azimuthal ID

Cause the problem for $Z < 0$

Hit Position in LuCaS

```
G4bool LCSensitiveDetector::ProcessHits(G4Step *aStep, G4TouchableHistory *)
{
    ....
    LC_num      = theTouchable->GetHistory()->GetVolume(1)->GetCopyNo();
    // Which layer inside LumiCal?
    layer_num   = theTouchable->GetHistory()->GetVolume(2)->GetCopyNo();

    G4ThreeVector GlobalHitPos = ( (preStepPoint->GetPosition())+(postStepPoint-
>GetPosition()) ) / 2.;
    ....
    G4ThreeVector LocalHitPos = theTouchable->GetHistory()-
>GetTopTransform().TransformPoint(GlobalHitPos) ;
    ....
    G4double rho = LocalHitPos.getRho();
    G4double phi = LocalHitPos.getPhi();
    ....
    cell_num     = (G4int)floor(( rho - (CalRhoMin-cellDimRho/2.) ) / cellDimRho );
    sector_num   = (G4int)floor ( phi / cellDimPhi ) + 1;
    ....
    cellID.id0  |= (cell_num     << 0); // store the cell # in the lowest 8 digits
    cellID.id0  |= (sector_num  << 8); // shift the sector # to the next byte up
    cellID.id0  |= (layer_num   << 16); // shift the layer # to the next byte up
    cellID.id0  |= (LC_num      << 24); // shift the LumiCal # to the highest byte
    ....
}
```

Hit Azimuthal Angle

LuCaS

```
UTIL::CellIDEncoder<IMPL::SimCalorimeterHitImpl> cellid( "I:7,J:6,K:6,L:2"  
,calVec ) ; // LuCaSside uses: 1-2 for side
```

```
cellid["I"] = cellcode & 0xff ; // cell number in sector, rCell  
cellid["J"] = (cellcode >> 8) & 0xff ; // sector number, phiCell  
cellid["K"] = (cellcode >> 16) & 0xff ; // layer number  
cellid["L"] = (cellcode >> 24) & 0xff ; // side, arm;
```

```
layer = (*_mydecoder)( calHitIn )["K"] ;  
phiCell = (*_mydecoder)( calHitIn )["J"] ;  
rCell = (*_mydecoder)( calHitIn )["I"] ;  
.....  
rHit= (rCell+0.5) * _rCellLength + _rMin;  
PhiHit = (phiCell+0.5) * _phiCellLength;
```

Reconstruction

```
double LumiCalClustererClass::thetaPhiCell(int cellId, GlobalMethodsClass::Coordinate_t  
output) {  
.....  
GlobalMethodsClass::CellIdZPR(cellId, cellIdZ, cellIdPhi, cellIdR, arm);  
.....  
rCell = _rMin + (cellIdR + .5) * _rCellLength;  
zCell = fabs(_zFirstLayer) + _zLayerThickness * (cellIdZ - 1);  
thetaCell = atan(rCell / zCell);  
.....  
phiCell = 2*M_PI * (double(cellIdPhi) + .5) / _cellPhiMax;  
.....  
}
```

LumiCal Arm Encoding

```
cellId = GlobalMethodsClass::CellIdZPR(layer, phiCell, rCell, arm);
arm is ether 1 or 2 if it comes directly from LCIO file otherwise it -1 or 1
if it is converted to ClusterFinder schema (-1(10) = 1111...111(2) )
.....
int GlobalMethodsClass::CellIdZPR(int cellZ, int cellPhi, int cellR, int arm)
{
    int cellId = 0;
    cellId |= ( cellZ    << 0 ) ;
    cellId |= ( cellPhi << 10 ) ;
    cellId |= ( cellR    << 20 ) ;
    cellId |= ( arm      << 30 ) ;
    return cellId;
}
```

this is ok as `cellId |= (arm << 30);` uses 2 bits, But in

```
void GlobalMethodsClass::CellIdZPR(int cellId, int& cellZ, int& cellPhi, int&
cellR, int& arm)
    arm = (cellId >> 30 ) & (int)(( 1 << 2 ) -1) ;
```

which gives 1 same for `arm == -1` and `arm == 1`.

To recover `arm == -1` some additional efforts are required and sometime these 1 and -1 are used to get values from the map.

```
posWeightHit = log(calHit->getEnergy() / _totEngyArm[detectorArm]) +
_logWeightConst
```

Radial Cell Position

```
Setup::Lcal_CellPitch = ( Setup::Lcal_SensRadMax
                          - Setup::Lcal_SensRadMin
                          -
2.*Setup::Lcal_sector_dead_gap )/G4double(Setup::Lcal_n_rings); // result: 1.762

.....
void LCDetectorConstruction::InitDetectorParameters() CellPitch =
Setup::Lcal_CellPitch;

    SensDet = new LCSensitiveDetector("LumiCalSD", // name
                                      .....
                                      CellPitch, // radial cell size
                                      .....
                                      VirtualCell); // cell type real/virtual = false/true
.....
    SetRhoCellDim(cellDimRho);
.....
    cell_num = (G4int)floor(( rho - CalRhoMin ) / cellDimRho );
```

```
ClusterFinder:
_rCellLength = (_rMax - _rMin) / _cellRMax; // result: 1.8
.....
rHit = (rCell+0.5) * _rCellLength + _rMin;
```

Summary

- Present implementation of the LumiCal reconstruction procedure does not make clear difference between the local and global coordinates.
- Azimuthal angle is not reconstructed properly in the LumiCal arm with $Z < 0$.
- Should we consider to make changes in accordance with LC detector geometry conventions, if any.