



DD4hep-ACTS integration for B0

Sakib Rahman
University of Manitoba

EPIC Far-Forward Weekly Meeting
January 19, 2023

Integrating the B0 Trackers into ACTS tracking for EPIC

Add B0tracker detector plugin
to digitise and reconstruct
hits with detector info

```
void InitPlugin(JApplication *app) {  
    InitJANAPugin(app);  
    using namespace eicrecon;  
    app->Add(new JChainFactoryGeneratorT<SiliconTrackerDigi_factory>  
             ({"B0TrackerHits"}, "B0TrackerRawHits"));  
    app->Add(new JChainFactoryGeneratorT<TrackerHitReconstruction_factory>  
             ({"B0TrackerRawHits"}, "B0TrackerRecHits"));  
}
```

Step 5:
Activating the
tracker in
ElCrecon

```
std::vector<std::string> EICRECON_DEFAULT_PLUGINS = {  
    ...  
    "B0TRK"  
    ...  
}
```

Add B0tracker plugin to default list
of plugins to activate in
src/utilities/eicrecon/eicrecon.cc

Add B0TrackerRecHits to be used in
src/global/tracking/tracking.cc

```
app->Add(new JChainFactoryGeneratorT<TrackerHitCollector_factory>(  
    {  
        "SiBarrelTrackerRecHits",  
        "SiBarrelVertexRecHits",  
        "SiEndcapTrackerRecHits",  
        "TOFBarrelRecHit",  
        "TOFEndcapRecHits",  
        "MPGDBarrelRecHits",  
        "MPGDDIRCRHits",  
        "B0TrackerRecHits"  
    },  
    {  
        "CentralTrackingRecHits"  
    }  
});
```

Passed
to
tracker
source
linker

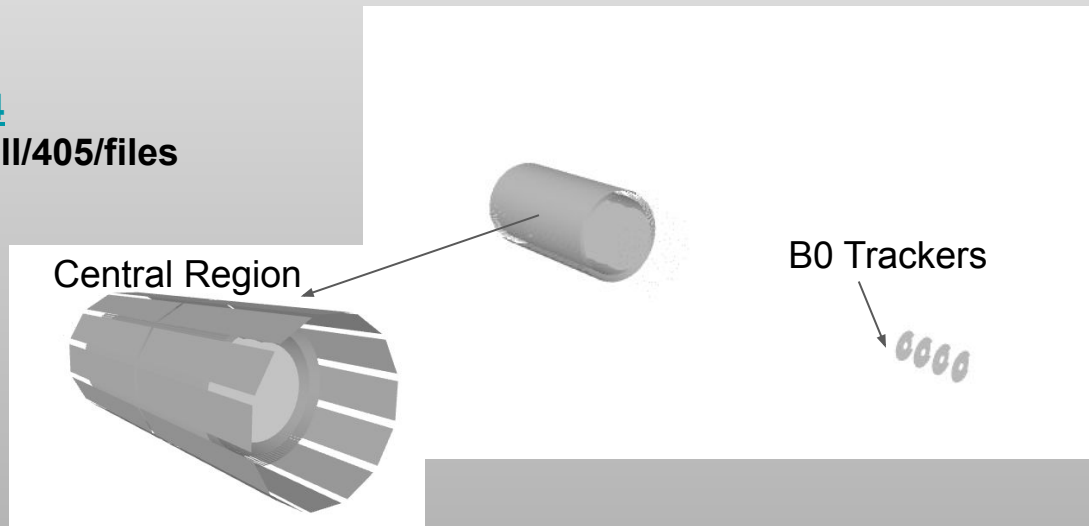
Test simulation and reconstruction chain

```
ddsim -compactFile epic_tracking_only.xml -G -gun.particle "pi-" -gun.momentumMin  
"1*GeV" -gun.momentumMax "20*GeV" -gun.distribution "uniform" -N 2000 -outputFile  
sim_pi-_1GeV_20GeV_tracking_only.edm4hep.root
```

```
install/bin/run_eicrecon_reco_flags.py sim_pi-_1GeV_20GeV_tracking_only.edm4hep.root  
rec_result -Pplugins=dump_flags,janadot -Pdump_flags:json=tracking_only_flags.json
```

<https://github.com/eic/epic/pull/344>

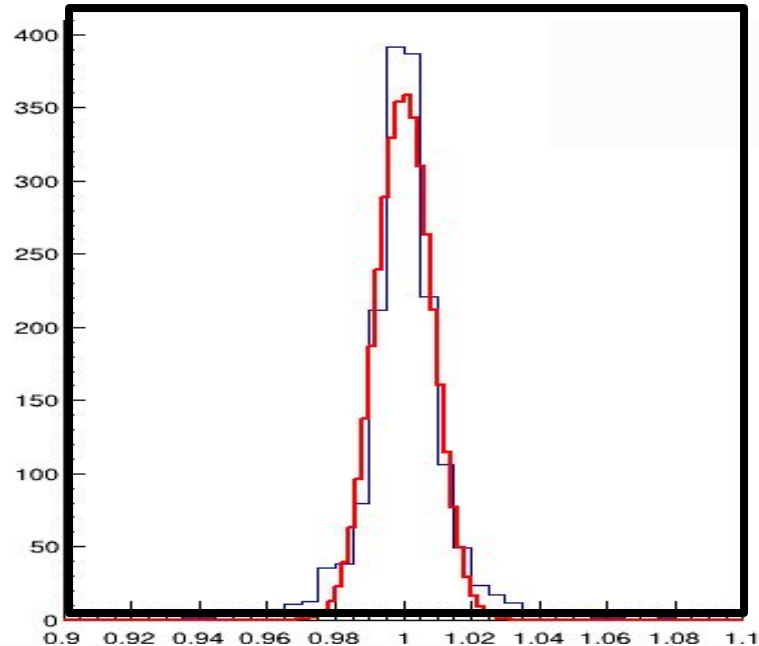
<https://github.com/eic/ElCrecon/pull/405/files>



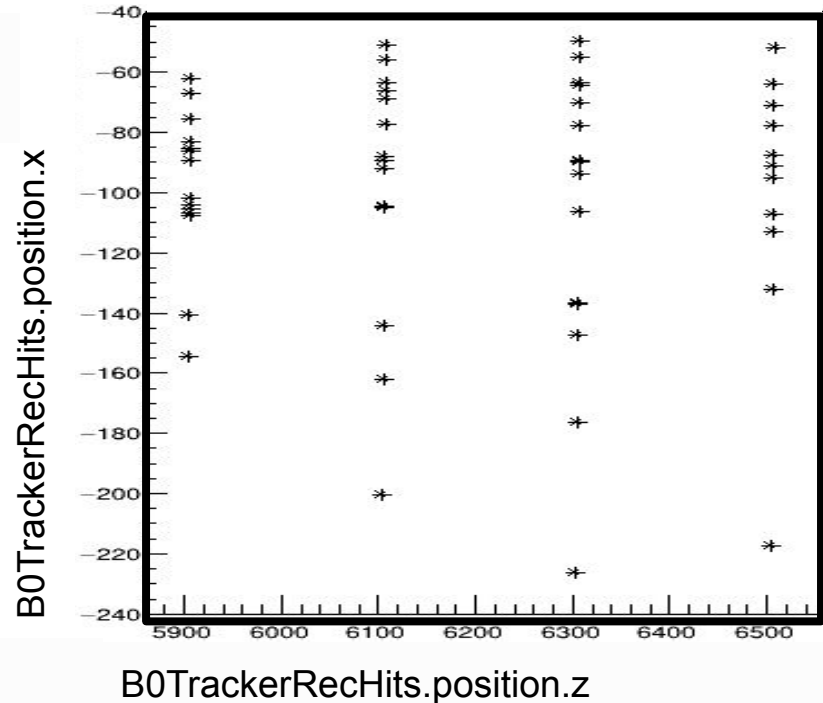
Overall reconstruction seems to produce reconstructed momentum fit with reasonable width but fails for events recording hits on B0

Warning indicates that no surface in the acts tracking geometry can be associated with hits on B0. Currently debugging this issue with help from ACTS and EICrecon experts.

ReconstructedZMomentum/TruthZMomentum



B0TrackerRecHits



Relevant matrix operations in ACTS core

```
Acts::Result<Acts::Vector2> Acts::PlaneSurface::globalToLocal(  
    const GeometryContext& gctx, const Vector3& position,  
    const Vector3& /*unused*/, double tolerance) const {  
    auto inverse = transform(gctx).inverse();  
    Vector3 loc3Dframe = inverse * (position);  
    Vector3 global3Dframe = localToGlobal(gctx,{loc3Dframe.x(),loc3Dframe.y()}, {0,0,0});  
    if (loc3Dframe.z() * loc3Dframe.z() > tolerance * tolerance) {  
        .....  
        return Result<Vector2>::failure(SurfaceError::GlobalPositionNotOnSurface);  
    }  
    return Result<Vector2>::success({loc3Dframe.x(), loc3Dframe.y()});  
}
```

Round trip test.
Log the inverse
matrix and local
and global
positions.

Manual matrix operations in ElCrecon

```
auto inverse =  
((surface->transform(Acts::GeometryContext()))).inverse();  
local_x = hit_pos.x*inverse(0,0)+hit_pos.y*inverse(0,1)  
          +hit_pos.z*inverse(0,2),  
local_y = hit_pos.x*inverse(1,0)+hit_pos.y*inverse(1,1)  
          +hit_pos.z*inverse(1,2),  
local_z = hit_pos.x*inverse(2,0)+hit_pos.y*inverse(2,1)  
          +hit_pos.z*inverse(2,2));
```

ACTS global x y z

-92.3752 -123.472 5906.54

ACTS inverse

-0.906025 -0.422618 -0.0226553

0.422486 -0.906308 0.0105644

-0.0249974 1.16226e-16 0.999688

ACTS local x y z ztolerance

-2.44 45.16 -0.000180419 0.0001

ACTS 360 global x y z

-92.3752 -123.472 5906.54

ElCrecon global x y z

-92.37521 -123.47248 5906.5386

ElCrecon inverse

-0.9060245806039117 -0.4226182617408825 -0.022655334573139643

0.4224862004124976 -0.9063077870365647 0.010564356009524422

-0.024997395914712218 1.1622647289044608e-16 0.9996875162757026

ElCrecon local x y z

2.0613318104478537 135.27559407627325 5907.002016835014

Sample point that fails the “On Surface Test”

ACTS core matrix operations
passes round trip test
(global=>local=>global)

Data point is passed properly
from eicrecon to acts

Local coordinates from manual
multiplication do not match that
from ACTS core. Why?

ACTS core local z is only
slightly higher the threshold for
on surface tolerance. Can we
increase this to 1 micron just for
the B0?

```

    auto hit_det = hit->getCellID() & 0xFF;
    auto onSurfaceTolerance = 0.0001;          // By default, ACTS uses 0.1
micron as the on surface tolerance
    if (hit_det == 150) {
        onSurfaceTolerance = 0.001;           // Ugly hack for testing B0.
Should be a way to increase this tolerance in geometry.
    }

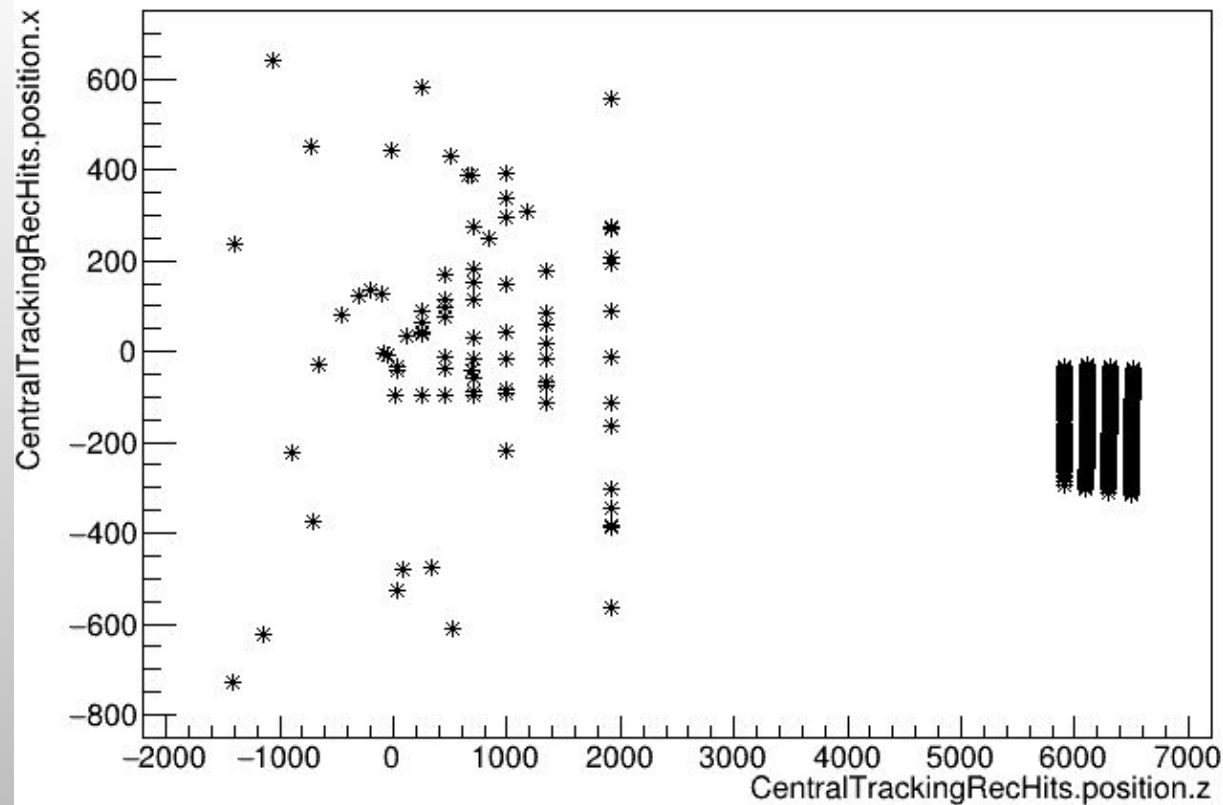
    try {
        // transform global position into local coordinates
        // geometry context contains nothing here
        pos = surface->globalToLocal(
            Acts::GeometryContext(),
            {hit_pos.x, hit_pos.y, hit_pos.z},
            {0, 0, 0}, onSurfaceTolerance).value();

        loc[Acts::eBoundLoc0] = pos[0];
        loc[Acts::eBoundLoc1] = pos[1];
    }

```

Change
onSurfaceTolerance
for just B0

CentralTrackingRecHits.position.x:CentralTrackingRecHits.position.z



Now hits from B0 no longer fail the on surface test.

However, reconstruction still seems to be not working for these events. Don't have any error messages to go by.

