ATLAS Tracking Detector Upgrade studies using the Fast Simulation Engine

This content has been downloaded from IOPscience. Please scroll down to see the full text.
(http://iopscience.iop.org/1742-6596/664/7/072005)

View the table of contents for this issue, or go to the journal homepage for more

Download details:

IP Address: 137.138.124.206
This content was downloaded on 24/02/2016 at 13:18

Please note that terms and conditions apply.
ATLAS Tracking Detector Upgrade studies using the Fast Simulation Engine

Noemi Calace\textsuperscript{1}, Andreas Salzburger\textsuperscript{2} on behalf of the ATLAS Collaboration, CERN, Switzerland

\textsuperscript{1} Département de physique nucléaire et corpusculaire, Section de Physique, Université de Genève, 24 Quai Ernest-Ansermet, Genève, Switzerland
\textsuperscript{2} European Organiz. for Nuclear Res. (CERN)

E-mail: Noemi.Calace@cern.ch

Abstract. The successful physics program of the LHC Run-1 data taking period has put a strong emphasis on design studies for future upgrades of the existing LHC detectors. In ATLAS, testing alternative layouts through the full simulation and reconstruction chain is a work-intensive program, which can only be carried out for a few concept layouts. To facilitate layout prototyping, a novel technique based on the ATLAS reconstruction geometry and a fast simulation engine have been established that allow fast layout iterations and a realistic but fast Monte Carlo simulation. This approach is extended by a fast digitisation and reconstruction module.

1. Introduction
With the first long shutdown (LS1) ending and Run-2 of the LHC data taking starting, the LHC experiments shift part of their focus to detector upgrade. For the ATLAS experiment \cite{atlas}, the most ambitious upgrade is planned as a Phase-2 upgrade during the third long shut down phase (LS3) of the LHC starting in 2021. A complete replacement of the innermost tracking device is foreseen to cope with the expected running conditions in Run-3, which are characterised by a highly increased event pile-up. Given the long preparation time for such an update, feasibility and design studies have to be performed at present to allow component prototyping and production to be completed in time for installation during LS3.

The simulation of different detector concepts is one key ingredient for designing a new inner tracker. In high energy physics, detector simulation is generally being performed using the Geant4 \cite{geant4} \cite{geant4a} toolkit, however, also fast simulation strategies are widely used. In ATLAS, a fast tracker simulation FATRAS \cite{fatras} exists, which is based on components from the offline software.

2. Set-up description
The ATLAS detector description for simulation, digitisation and reconstruction is based on a common geometry source, the GeoModel \cite{geomodel} library. Dedicated converters then change this description into either Geant4 or a reconstruction geometry. Implementing different layouts can be a very time-consuming process and for many studies including basic layout prototyping of the geometry setup fast simulation programs can be used efficiently. In ATLAS, several studies for the Phase-2 Letter of Intent \cite{atlas_phase2} have been performed with the semi-analytical IdRes program \cite{idres}.
which allows quick evaluation of track resolutions and hit statistics of layout candidates. However, IdRes fails to accurately describe efficiencies, details about layout hermicity and has no concept about the effects from pile-up and dense environments.

To address these questions before spending the time to implement the candidate layout into the full simulation setup, a novel workflow based on FATRAS has been recently deployed. The key idea behind this approach is to facilitate the layout definition while at the same time guaranteeing a fair amount of accuracy in the simulation.

The fast tracker simulation FATRAS reads the detector description from GeoModel and builds the tracking geometry\(^1\). The TrkDetElementBase is the base class for all tracking detector elements with methods for reading out all relevant information, extended by the geometry description implemented in GeoModel.

A new technique for building a customised tracking geometry has been developed reading the detector description from XML configuration files in order to remove the dependence on the GeoModel library. The PlanarDetElement class represents the smallest entity of the custom geometry and extends the common ATLAS TrkDetElementBase class. The PlanarDetElement has an attached Surface representation and contains the information about the position and rotation of the module, its dimensions, its shape, as well as its readout segmentation. The module’s material is customizable and attached to the single Surface using the classes already implemented in the ATLAS framework. Each module includes information about surrounding elements for navigation purpose.

A new builder tool has been developed to build custom geometries following the detector and material description. The layout building and material implementation are defined through a limited number of interesting parameters read in by XML. These simplifications decrease the CPU time consumption. Figure 1 shows the comparison of the hit distribution of single $\mu$ tracks in the ATLAS Pixel Detector using the FATRAS tracking geometry from GeoModel and from XML configuration file.

![Figure 1: Comparison of hit distribution of single $\mu$ tracks in the ATLAS Pixel and SCT Detectors using FATRAS tracking geometry from GeoModel and from XML configuration file \[9\].](image)

Once the geometry is built and loaded, several particles are simulated by propagating through the detector. Hits are produced in sensitive detector elements, the same hit format as in the full simulation is used. This step is performed using a fast tracker simulation FATRAS \([4]\) based on the offline extrapolation engine. The main element for the extrapolation is the ExtrapolationCell that contains the extrapolation’s configuration, caches material, track parameters and jacobians. The engine navigates through the volumes and layers and creates the

\(^1\) The Tracking Geometry \([8]\) is a simplified version that is based on a surface description of the detector. Material components are projected onto these surfaces.
trajectories by propagating through the magnetic field and intersecting sensitive detector elements. During the propagation through the detector, the extrapolation engine takes care also of performing the interaction with the detector material according to different particles types. It samples the material effects from parametrised functions taking ionization, bremsstrahlung photon emission, photon conversion, positron annihilation, multiple scattering effects and hadronic interaction into account. The tracks parameters are then updated and the extrapolation is carried on until the geometry’s boundary is reached or the particle is killed because of material effects or decay. Figures 2 and 3 show the comparison of the energy loss in the silicon system for full and fast simulation.

![Figure 2: Comparison of the energy loss distributions for 1 GeV single $\mu$ tracks in the ATLAS Pixel and SCT Detectors for full simulation (based on the Geant4 toolkit) and FATRAS simulation [9].](image1)

![Figure 3: Comparison of the energy loss $\eta$ distributions for 1 GeV single $\mu$ tracks in the ATLAS Pixel and SCT Detectors for full simulation (based on the Geant4 toolkit) and FATRAS simulation [9].](image2)

Hits are produced in the detector where intersections of the track with sensitive surfaces occur. They are the inputs to the very fast simulation module based on a smeared digitisation approach related to the pitch size: a gaussian smearing is applied to the hit’s positions providing the optimisation of pitch sizes. During the digitisation step, PlanarCluster objects are produced from hits. These clusters extend the ATLAS tracking event data model [10] and can therefore be used transparently in most common tracking tools. The adapted truth seeded reconstruction [11] constitutes the last step of the chain: it accepts as input the cluster produced during digitisation and provides a fast track reconstruction.

3. First results
A first demonstration of working implementation has been carried out simulating a custom tracking system that emulates the actual ATLAS pixel detector. The description of the layout
has been read, the geometry has been built and loaded and single muons of different momenta have been simulated. Single muons have been simulated and their tracks extrapolated using the ExtrapolationEngine running through the FATRAS transport. The main result achieved using the offline extrapolation engine is the very low time and CPU consumption needed for extrapolating thousands of tracks of different momenta through the tracking geometry. The needed time for track extrapolation of $10^5$ muons with momentum of 10 GeV and 1 GeV (using 8 x Intel Xeon E312xx series, 2.6 GHz, RAM 16 GB) are reported in Table 1. The time spent for extrapolation is shown to be almost the same for 1 GeV and 10 GeV muons.

The hits produced were distributed on the surfaces of the geometry in very good agreement with the ones produced by very detailed simulation based on the Geant4 toolkit.

Other results have been produced comparing the material distribution of the ATLAS pixel detector using GeoModel implementation and of an ATLAS pixel-like custom detector. Although introduced just recently, the custom geometry builder tool has been already heavily used to simulate and visualize several layouts under investigation for the new ATLAS inner tracker and, together with the new extrapolation, to produce first coverage studies. The custom tracking geometry has been visualised using the Virtual Point 1 tool (VP1) [12], as shown in Figure 4.

Table 1: Time needed for tracks extrapolation of $10^5$ muons with momentum of 10 GeV and 1 GeV (using 8 x Intel Xeon E312xx series, 2.6 GHz, RAM 16 GB).

<table>
<thead>
<tr>
<th>muon $p_T$</th>
<th>Time needed for extrapolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_T = 10$ GeV</td>
<td>Tot = 31.6 [s] Ave/Min/Max = 31.6 ($\pm$ 8.04)/18/76 [ms]</td>
</tr>
<tr>
<td>$p_T = 1$ GeV</td>
<td>Tot = 32.8 [s] Ave/Min/Max = 32.8 ($\pm$ 8.22)/18/96 [ms]</td>
</tr>
</tbody>
</table>

Figure 4: A sample event visualized with the ATLAS Virtual Point 1 tool. The surfaces belonging to the custom tracking geometry are shown. Extrapolated tracks are also shown.
4. Conclusions
A new technique for building custom tracking geometry has been developed reading the detector description from XML configuration files removing any dependence on GeoModel libraries. It has been used for simulation and performance studies of the innermost tracking device for the very challenging pile-up scenario in Run-3.

Based on the FATRAS framework, the simplification in the material and geometry description using XML configuration files facilitates the simulation of several kinds of layout reducing considerably the CPU time consumption. Indeed, the custom geometry builder tool and the new extrapolation engine have already been using intensively for first coverage studies for several prototypes for the new ATLAS tracking detector.

In a following step, hits produced after transporting simulated particles through the detector will be then digitised, using a very fast digitisation module. The out-coming clusters are planned to be used in the truth seeded reconstruction step.

Validation studies for checking material description and interaction with the detector material are ongoing.

References
[1] The ATLAS Collaboration 2008 The ATLAS Experiment at the CERN Large Hadron Collider JINST vol 3 S08003
[1] www.guit.sssup.it.it