CMS

The Compact Muon Solenoid Experiment

Detector Note

The content of this note is intended for CMS internal use and distribution only

2018/02/01

EJ200 Scintillator Tile Geometry Update and Testing

A. Horst

Abstract

We study the light output and light collection efficiency of the EJ-200 Scintillator Tile by Eljen Technology, using a simulated tile created using GEANT4 software. We present new methods of testing uniformity and efficiency of irradiated and unirradiated simulated tiles.

1 CMS papers

We study the light output and collection efficiency of scintillators being considered for an update at the hadronic calorimeter of the CMS detector. Experimental tests are useful for determining properties of different tiles, however, simulations using GEANT4 could provide another more efficient way of testing different geometries and tiles. Here I test a commercial scintillator by Eljen Technology: the EJ-200 tile. It has a polyvinyltoluene (PVT) base, an emission peak of 425nm, a shaped groove with a wavelength shifting fiber inside connected to a Hamamatsu R6091 PMT and is wrapped in Tyvek.

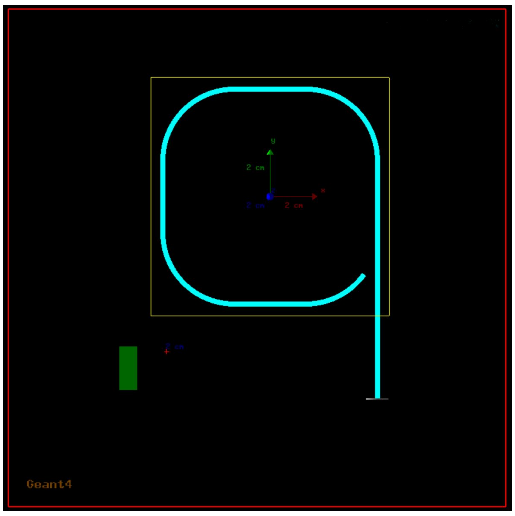

Figure 1: Visual simulation of the new fiber (blue) for the EJ-200 simulated tile.

2 Project

Our tile simulation was in need of a modified geometry for the EJ-200 material. The main section altered was the output corner, where the fiber leaves the tile, and where the opposite end of the fiber ends inside the tile. Originally, the output of the fiber was connected to the end of the fiber in a T like style on the output corner. This is not true of the EJ-200 tile, in which the end and output are not connected at this corner, nor is the end of the fiber straight. The EJ-200 fiber looks closer to a , where the output is not connected to the end of the Fiber (see Fig. 1).

2.1 Uniformity Testing

To test this new geometry, and create ways of validating new tile geometries with the GEANT4 simulation, I added a method to test uniformity of photon detection efficiency in the tile. Other experimental researchers provided absorption lengths from tests with irradiated and unirradiated tiles. Using these experimental results, I was able to find the uniformity of photon absorption in the simulated tile based on position. This involved creating a function to record the photon source position to gather position based data on the number of photons being detected by the unirradiated and irradiated tiles. Although partially used to compare the unirradiated tiles to the irradiated tiles, this process was also used to observe the accuracy of the simulation itself, by comparing the results to known, measured results. To complete this uniformity test, I needed a method of creating a changing particle source to test specific points on the tile. Using the position commands and a python script, I could run multiple jobs at different points. I also added a function that records the position of the particle source before each run, so my final

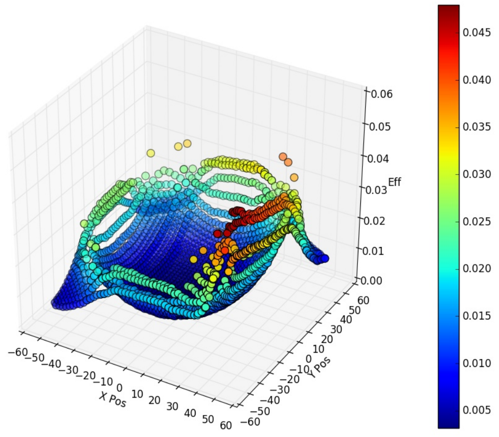

Figure 2: Example of a 2500 point efficiency simulation with 1 million events per point at 1 photon per event, on the unirradiated tile simulation (with points inside the fiber removed due to their outlying nature). Efficiencies are very low, as expected. Even with thousands of emitted photons, only a few are likely to be recorded.

results could be differentiated by their source position. Fig. 2 shows an example of the results from this testing, with a position based graph of the efficiency of the tile at picking up photons from a photon source with 2500 points total each with 1 million events of 1 photon randomly shot from the center plane of the tile in any direction. Fig. 3 displays the ratio of an irradiated tile to an unirradiated tiles efficiency.

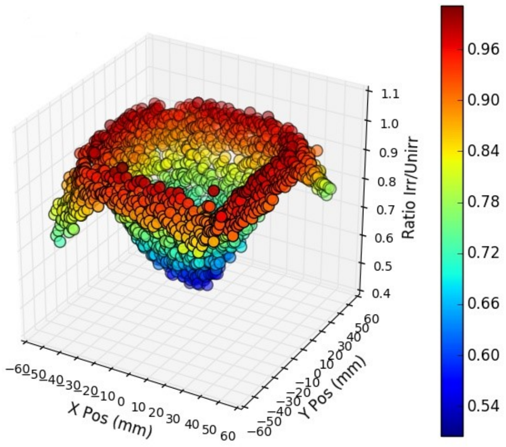

Figure 3: Ratio between the efficiencies (photons that reach the PMT vs photons released from the particle source) of the irradiated vs the unirradiated tiles at 2500 points. Results show similar efficiencies around the fiber groove and ratios as low as .5 in the center of the tile, where the light travels through more of the scintillator, and is more likely to be absorbed in the irradiated tile before reaching the fiber.

2.2 Muon Light Yield Testing

In addition to tile uniformity testing, I needed a way to find an estimated number of photons released in the EJ-200 tile by Muons interacting with it, causing scintillation. This was done in part by observing the number of detected photons in a simulation given a certain number of photons released from a particle source. I also used histograms to record the amount of energy deposited in the simulated scintillator by a Muon of energy given by a sea level muon distribution [1]. This could be used to narrow down the number of photons that might be released during scintillation of Muons in the EJ-200 tile.

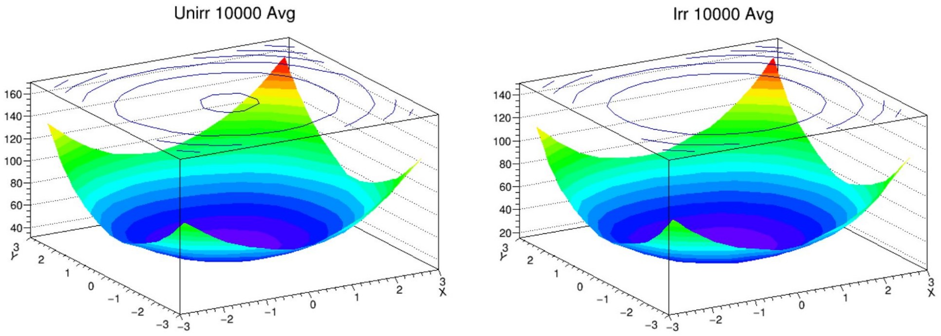

Figure 4: Avg number of photons detected out of 1000 events with 10000 photons at 400 points on the irradiated and unirradiated tiles (graphed in the center of the tile in area enclosed by fiber).

3 Function

e primary goal of this project is to modify/extend and verify our existing GEANT4 simulation, including updating the tile geometry of the scintillator tile to the commercial EJ-200 geometry, by altering the tile geometry and the fiber groove. This change was made so I could more realistically describe real measurements, such as the cosmic ray, alpha source, and test beam measurements, which could then be used to explain behaviors in the real tiles. After validating these results, others can further develop this simulation based on this simulation to morph/extrapolate it into more geometries for other tiles. By comparing/contrasting different types/geometries of tiles, one could find tiles better suited for use in the detector based on radiation damage, longevity, and accuracy. These tiles are crucial to the measurements made at CMS, and better tiles will mean more accurate measurement with reduced noise. Using GEANT4 will be a straight forward method of finding tiles that best suit the detectors.

4 Method

To test the tiles, I first updated the fiber geometry, which required mostly some slight changes to the tori/tubes that represent the fiber in the simulation. Using previously defined lengths and parameters, I added a new fiber geometry that altered the readout corner to incorporate the new geometry specified earlier.

```
G4VSolid* solidFiberCurved1R =
        new G4Torus(name+"CurvedSection1R",
                    radiusI,                        //G4double pRmin,
                    radiusO,                        //G4double pRmax,
```

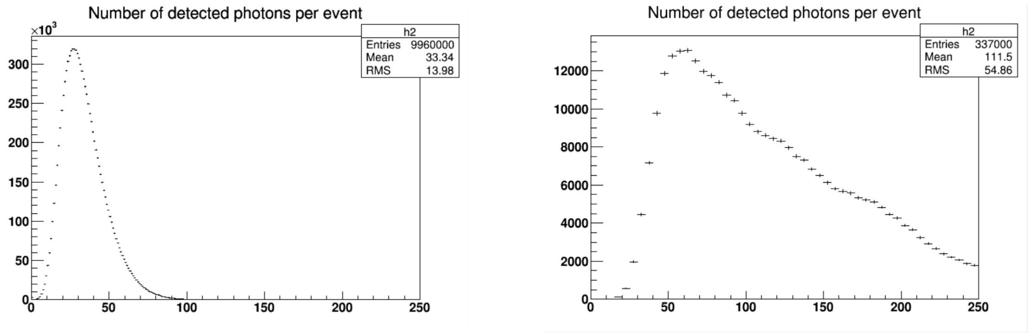

Figure 5: Fig 5. - 5000 photon source histogram of photons detected in 10 million events with randomly distributed particle source positions along a 9x9cm square plane centered in the middle of the tile. Fig 6 - 10,000 photon source histogram of photons detected in 300,000 events, from same source as fig. 5.

```
                    bendRadius,                      //G4double pRtor,
                    1.5*pi,                     //G4double pSPhi,
                    0.3*pi);                    //G4double pDPhi)
```

This code defines a torus for the readout corner, you should notice that the torus does not complete an entire 0.5pi curve (0.3pi in this case), which is what we desire here. Highlighting the fiber in a visual simulation was extremely useful for determining problems. The next step was to create a function to record the position of the particle source at the beginning of each event. This was necessary for obtaining the results for each event at each position.

```
const G4ThreeVector LYSimPrimaryGeneratorAction::GetSourcePosition()
{
    G4ThreeVector pos = particleSource->GetParticlePosition();
    return pos;
}
```

To get the output, I recalled `pos` at the end of each run, and I could then associate efficiencies with their original particle source position. I could then make these results into a dataframe using python, and represent the results in a 3d scatter graph using matplotlib. Each simulation I ran had an accompanying python script to submit each simulation job. This script would change certain parameters (source position, histogram names, etc.) and would also submit only the desired jobs, which was important when some points contained unusable points. Additionally, I had the script skip over source positions that were inside the fiber, since they were extraneous and would skew data. Histograms are easily defined in the analysis code, with binning and axis limits (changeable to some degree with root). Filling the histograms is made easy by the singleton access method `Instance()` in the G4AnalysisManager class, meaning histograms can be filled just about anywhere. This was useful when working with muons, since some histograms had to be filled every step, to record the energy deposited into the tile.

5 Results

The uniformity tests had good results, with some variation of 4% (in efficiency) from maximum to minimum, which varied according to the source distance from the fiber and the source distance from the readout corner. At these points, photons incident on the fiber are likely to be detected, of which there are many at these source points in comparison. With a median eff on the unirradiated tile of .012, and a min and max of, respectively .003 and .048 the majority of points lie below 2% efficiency. This indicates some uniformity in the areas of the tile further from the fiber. The fiber proximity issue also arises when comparing the unirradiated and irradiated tiles, as depicted in Fig. 3, where the ratios in these areas are close to 1, whereas in central/corner areas the ratios drop below 1 (as expected). These results demonstrate the capabilities of the tile simulation to test scintillator tiles for use in detectors. These tests show methods of comparing tiles in terms of uniformity, and degradation. This may prove to be an important method in testing new styles of tile geometries. The muon tests were ultimately to get a bound on the number of photons detected compared to the amount released at the particle source. Both have a positive skew gaussian distribution which gets more skewed as the source photon count increases. The skewness is likely simply caused by the bound at 0 photons. It is unlikely that 0 photons are recorded in either of these cases, so the distribution has a cut off. The left side of the distribution would be less likely to move outwards (low photon count is still possible and likely) but the right side (higher photon count) expands outwards (possible but less likely). The highest percent of photons absorbed vs. released grow also, the 5,000 photon source curve ends at around 1.5% (75 photons), the 10,000 photon distribution ends at around 3% (300). These results demonstrate how the simulation can be used to find an approximate bound on the number of photons being scintillated when a muon passes through the tile.

References

[1] C. Hagmann, D. Lange, and D. Wright, “Cosmic-ray shower generator (cry) for monte carlo transport codes”, *IEEE* (2007) 1143.