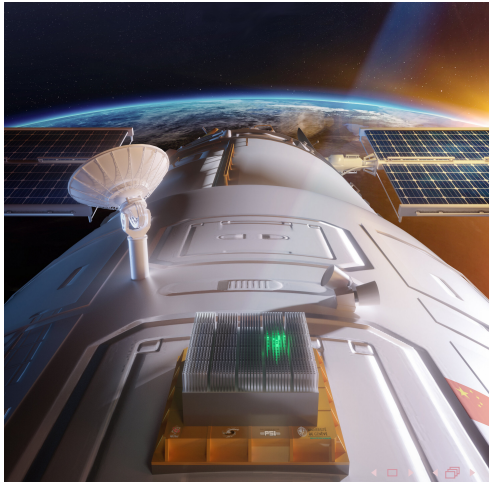


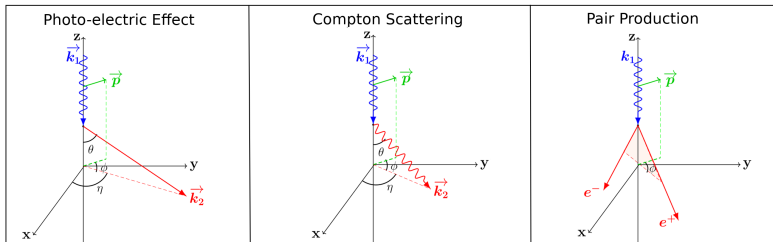
Simulating POLAR in Geant4

1st National Workshop on GEANT4

Merlin Kole



- POLAR is a dedicated Gamma-Ray Burst Polarmeter
- Measures polarization of photons in the 50-500 keV energy range
- Employs Compton scattering to measure polarization



Compton Polarimetry: A Quick Intro

- Azimuthal scattering angle (ϕ) depends on polarization vector
- Preferential scattering perpendicular to polarization vector
- Segmented detector allows to measure angle
- Scattering angle distribution \rightarrow Modulation Curve

$$\frac{d\sigma}{d\Omega} = \frac{r_o^2}{2} \frac{E'^2}{E^2} \left(\frac{E'}{E} + \frac{E}{E'} - 2 \sin^2 \theta \cos^2 \phi \right). \quad (1)$$

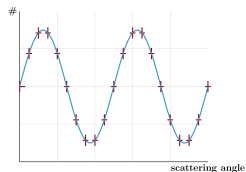
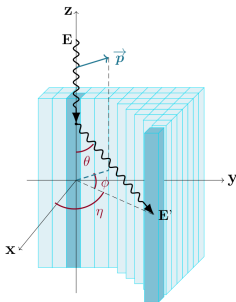
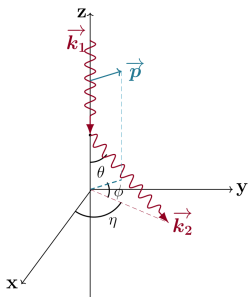
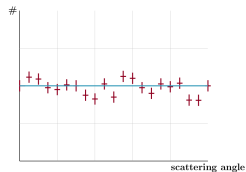
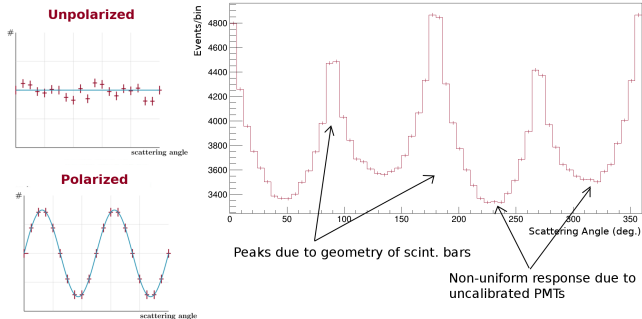


Figure 8: Modulation Curve

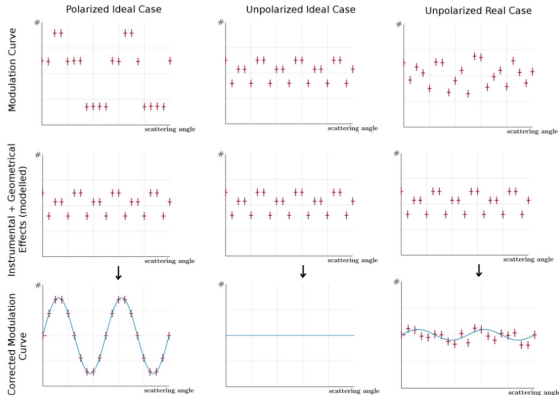


Real Modulation Curves

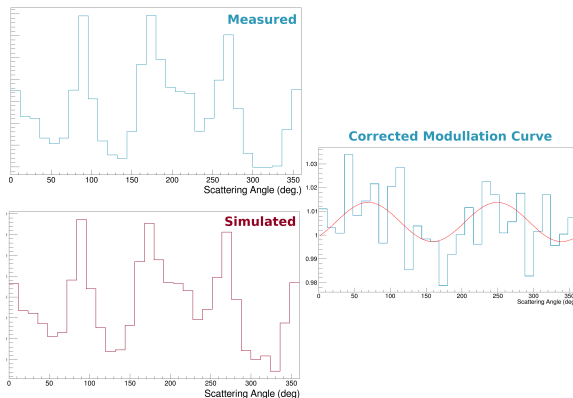


- Theory: plot the scattering angles \rightarrow check amplitude of 180° modulation \rightarrow convert to polarization \rightarrow publish
- Reality: Modulation curves look complex
- Wide field of view results in off-axis effects \rightarrow 360° modulation
- Instrument geometry and surrounding material add further complexities
- Instrument non-uniform response
- etc.
- etc.

Reconstructing PD & PA

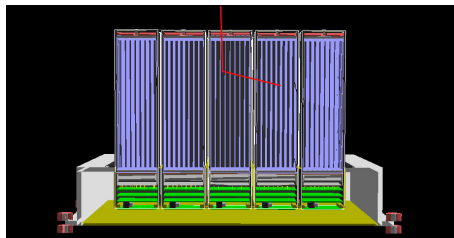
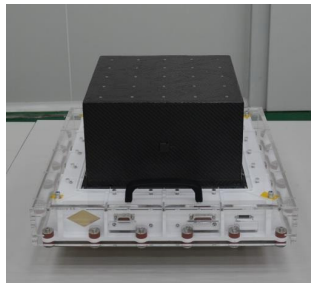


- Analysis proceeds typically by dividing the measured curve by simulated unpolarized curve
- Divide out all the extra effects and only leaves polarization induced effects
- Simulate scattering angle distribution for specific GRB (spectrum, angle, temperature etc.)
- Any error in the MC produces an positive PD

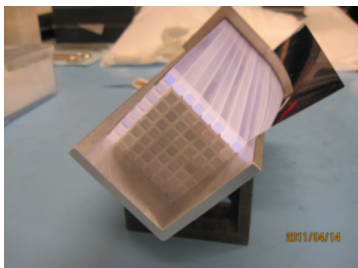
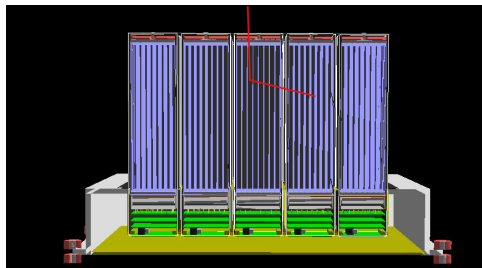
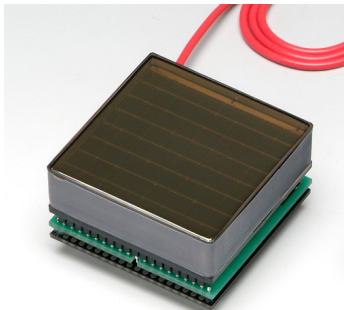


- Modelling of all details complex
- Any deviation from reality leads to "polarization"
- Second law of thermodynamics → Making a line more flat (unpolarized) is unlikely
- Compare to spectral measurements where error can go either way

- Effective area of $\approx 300\text{cm}^2$ at 400 keV
- Small pixels allows for high precision scattering angle measurements
- Field of View of half the sky

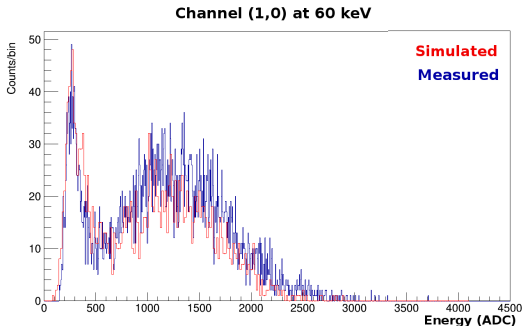


- Plastic scintillator array readout using MAPMTs
- Scintillators of $5.9 \times 5.9 \times 172 \text{ mm}^3$
- MAPMTs reads out 64 scintillators
- Carbon fibre shielding around the scintillators
- Aluminium frame housing the electronics
- Mounted on Tiangong-2 spacelab



MC rules for POLAR collaboration

- 1 Only one MC simulation package
- 2 Keep it as simple as possible
- 3 Output data from MC framework is identical to real data
- 4 Geant4 handles only the physics, electronics is performed in a second stage
- 5 No data analysis until MC is fully understood

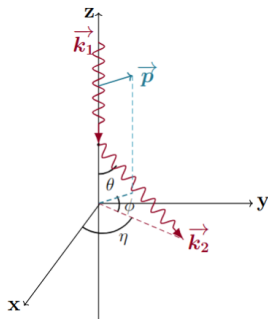


Get code: <https://github.com/POLAR-2/POLAR-SIM>

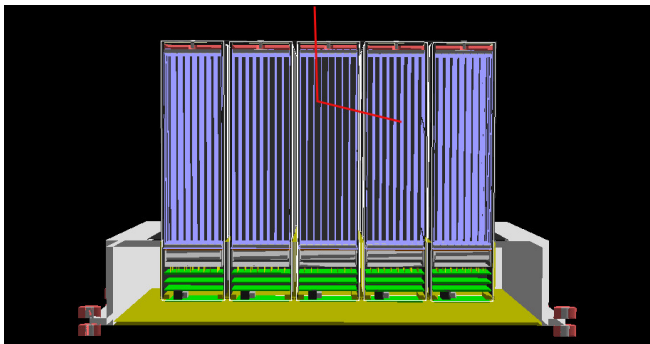
- DetectorConstruction.cc → contains all the detector details
- Physicslist.cc → contains links to physics lists to be used
- ActionInitialization.cc → Defines format of output data file (ROOT files in out case)
- Run.cc → extracts necessary info from each step
- ConfigMessenger.cc → passes configuration information to the simulations
- main.cc wraps everything together
- 2 or 3 other files might be needed but that is it
- Complexity of the code tends to increase when copying from another project

```
168K Sep 29 2017 Bgo/src/DmpSimuBgoDigitizer.os
21K Sep 29 2017 Bgo/src/DmpSimuBgoHit.os
468K Sep 29 2017 Bgo/src/DmpSimuBgoHitEventManager.os
249K Sep 29 2017 Bgo/src/DmpSimuBgoHitNTupleMaker.os
381K Sep 29 2017 Bgo/src/DmpSimuBgoSptStructEventManager.os
567K Sep 29 2017 DmpSim/src/DmpAlgRayTracer.os
138K Sep 29 2017 DmpSim/src/DmpAlgRayTracer.os
262K Sep 29 2017 DmpSim/src/DmpDetectorConstruction.os
58K Sep 29 2017 DmpSim/src/DmpDigiAlg.os
111K Sep 29 2017 DmpSim/src/DmpDetectorMessenger.os
25K Sep 29 2017 DmpSim/src/DmpG4RunManager.os
163K Sep 29 2017 DmpSim/src/DmpGeoRayTraceSvc.os
115K Sep 29 2017 DmpSim/src/DmpRootEventDataMessenger.os
779K Sep 29 2017 DmpSim/src/DmpBindingSim.os
191K Sep 29 2017 DmpSim/src/DmpSimuConfigParser.os
414K Sep 29 2017 DmpSim/src/DmpRootEventManager.os
203K Sep 29 2017 DmpSim/src/DmpSimAlg.os
512K Sep 29 2017 DmpSim/src/DmpSimuOrbitCheckAlg.os
84K Sep 29 2017 DmpSim/src/DmpSimuPhysicsList.os
297K Sep 29 2017 DmpSim/src/DmpSimuOrbitEventManager.os
178K Sep 29 2017 DmpSim/src/DmpSimuPrimaryGeneratorAction.os
72K Sep 29 2017 DmpSim/src/DmpSimuPrimaryGeneratorMessenger.os
308K Sep 29 2017 DmpSim/src/DmpSimuPrimaryEventManager.os
238K Sep 29 2017 DmpSim/src/DmpSimuEventAction.os
23K Sep 29 2017 DmpSim/src/DmpSimuTrackingAction.os
321K Sep 29 2017 DmpSim/src/DmpSimuRunEventHeaderMaker.os
107K Sep 29 2017 DmpSim/src/DmpSimuRunAction.os
105K Sep 29 2017 DmpSim/src/DmpSimuSteppingAction.os
17K Sep 29 2017 Nud/src/DmpSimuNudDigi.os
79K Sep 29 2017 Nud/src/DmpSimuNudDetectorDescription.os
332K Sep 29 2017 Nud/src/DmpSimuNudDigiEventManager.os
435K Sep 29 2017 DmpSim/src/DmpSimuTrajectoryEventManager.os
42K Sep 29 2017 Nud/src/DmpSimuNudDigitizerMessenger.os
175K Sep 29 2017 Nud/src/DmpSimuNudDigitizer.os
117K Sep 29 2017 Nud/src/DmpSimuSensitiveDetector.os
21K Sep 29 2017 Nud/src/DmpSimuNudHit.os
526K Sep 29 2017 Psd/src/DmpAlgPsdDigi.os
17K Sep 29 2017 Psd/src/DmpSimuPsdDigi.os
412K Sep 29 2017 Nud/src/DmpSimuNudHitEventManager.os
103K Sep 29 2017 Psd/src/DmpSimuPsdDetectorDescription.os
19K Sep 29 2017 Psd/src/DmpSimuPsdHit.os
18K Sep 29 2017 Stk/src/DmpSimuStkDigi.os
168K Sep 29 2017 Psd/src/DmpSimuPsdDigitizer.os
227K Sep 29 2017 Psd/src/DmpSimuPsdSensitiveDetector.os
398K Sep 29 2017 Stk/src/DmpSimuStkDigitizer.os
381K Sep 29 2017 Psd/src/DmpSimuPsdHitEventManager.os
426K Sep 29 2017 Stk/src/DmpSimuStkDigiEventManager.os
43K Sep 29 2017 Stk/src/DmpSimuStkDigitizerMessenger.os
21K Sep 29 2017 Stk/src/DmpSimuStkHit.os
88K Sep 29 2017 Stk/src/DmpSimuStkDetectorMessenger.os
459K Sep 29 2017 Stk/src/DmpSimuStkStepEventManager.os
320K Sep 29 2017 Stk/src/DmpSimuStkHitEventManager.os
125K Sep 29 2017 Stk/src/DmpSimuStkDetectorDescription.os
36K Sep 29 2017 OrbitSimulation/src/Main.o
193K Sep 29 2017 Stk/src/DmpStkSensitiveDetector.os
```

- Use G4EmLivermorePolarizedPhysics.hh for normal science simulations
- Use different physics list for background simulations (e.g. G4HadronPhysicsQGSP_BERT.hh)
- Do not blindly trust physicslists
- Example: first version of G4LowEPPolarizedComptonModel did not work at all
- Always check for changes when updating Geant4 versions



- GDML vs. coding by hand
- GDML is not easy to read
- Using CAD to gdml converters tends to make code slow
- writing by hand is a pain... but probably worth it



POLAR Geant4: Detector Construction

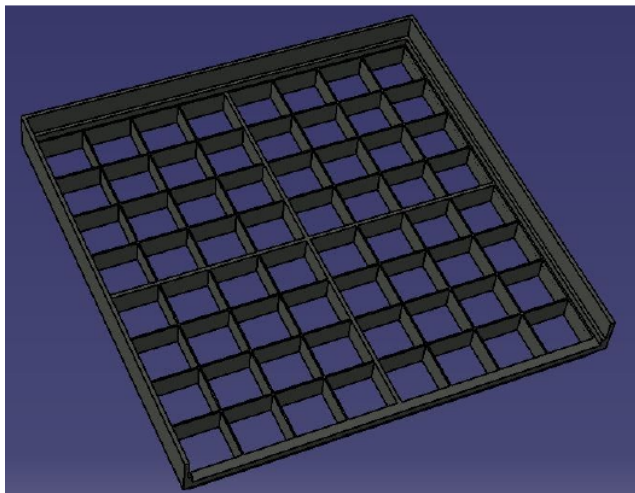
```
// Sylgard 184
Sylgard 184 = new G4Material(name = "Sylgard_184", density = 0.965*g/cm3, ncomponents = 4);
Sylgard 184 ->AddElement(elC, natoms = 2);
Sylgard 184 ->AddElement(elH, natoms = 6);
Sylgard 184 ->AddElement(elO, natoms = 1);
Sylgard 184 ->AddElement(elSi, natoms = 1);
```

```
// AlloyAl 7075
AlloyAl 7075 = new G4Material(name = "AlloyAl_7075", density = 2.81*g/cm3, ncomponents = 9);
AlloyAl 7075 ->AddElement(elSi, fractionmass = 0.4*perCent);
AlloyAl 7075 ->AddElement(elFe, fractionmass = 0.5*perCent);
AlloyAl 7075 ->AddElement(elCu, fractionmass = 1.6*perCent);
AlloyAl 7075 ->AddElement(elMn, fractionmass = 0.3*perCent);
AlloyAl 7075 ->AddElement(elMg, fractionmass = 2.5*perCent);
AlloyAl 7075 ->AddElement(elCr, fractionmass = 0.23*perCent);
AlloyAl 7075 ->AddElement(elZn, fractionmass = 5.6*perCent);
AlloyAl 7075 ->AddElement(elTi, fractionmass = 0.2*perCent);
AlloyAl 7075 ->AddElement(elAl, fractionmass = 88.67*perCent);
```

```
// AlloyAl 2219
AlloyAl 2219 = new G4Material("AlloyAl_2219", 2.85*g/cm3, 7);
AlloyAl 2219 ->AddElement(elV, fractionmass = 0.1*perCent);
AlloyAl 2219 ->AddElement(elSi, fractionmass = 0.1*perCent);
AlloyAl 2219 ->AddElement(elFe, fractionmass = 0.15*perCent);
AlloyAl 2219 ->AddElement(elZr, fractionmass = 0.15*perCent);
AlloyAl 2219 ->AddElement(elMn, fractionmass = 0.3*perCent);
AlloyAl 2219 ->AddElement(elCu, fractionmass = 6.3*perCent);
AlloyAl 2219 ->AddElement(elAl, fractionmass = 92.98*perCent);
```

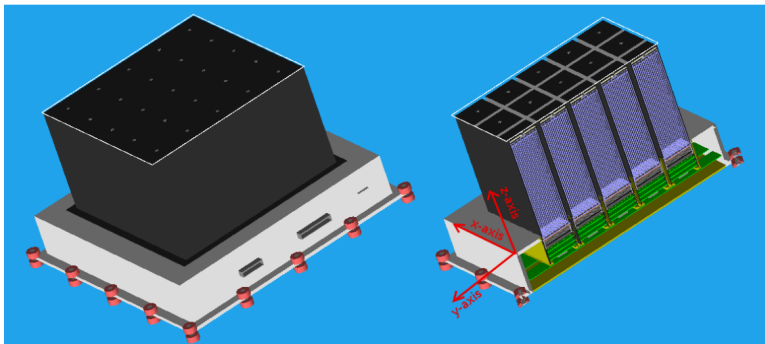
```
// AlloyAl 5083
//taken from https://www.smithmetal.com/pdf/aluminium/5xxx/5083.pdf
AlloyAl 5083 = new G4Material("AlloyAl_5083", 2.65*g/cm3, 8);
AlloyAl 5083 ->AddElement(elFe, fractionmass = 0.4*perCent);
AlloyAl 5083 ->AddElement(elCu, fractionmass = 0.1*perCent);
AlloyAl 5083 ->AddElement(elMg, fractionmass = 4.45*perCent);
AlloyAl 5083 ->AddElement(elSi, fractionmass = 0.4*perCent);
AlloyAl 5083 ->AddElement(elZn, fractionmass = 0.25*perCent);
AlloyAl 5083 ->AddElement(elCr, fractionmass = 0.15*perCent);
AlloyAl 5083 ->AddElement(elTi, fractionmass = 0.15*perCent);
AlloyAl 5083 ->AddElement(elAl, fractionmass = 94.1*perCent);
```

- Include materials in as much detail as possible
- Example: Aluminium alloys used in POLAR result in measurement background



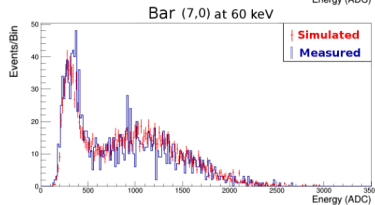
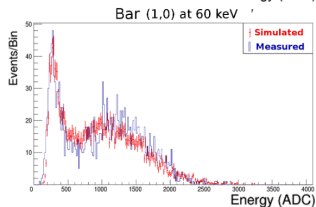
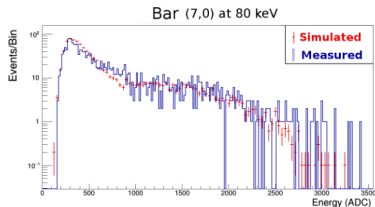
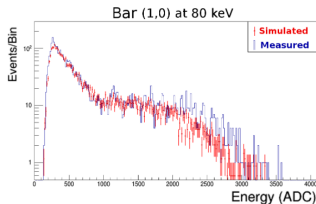
- Complex geometries can be done using things like `G4SubtractionSolid`
- Draw one `G4Box` and subtract another `G4Box` → clean code
- Downside: slows down simulations
- Ugly coding can be worth it at times

- Geant4 only handles the physics
- Information on primary particles is stored for debugging purposes
- For each physical interaction we store 5 parameters
 - 1) Detector channel number
 - 2) X,Y,Z interaction position in this channel
 - 3) Deposited energy in that location
- Deposited energy is corrected for Birks' effect

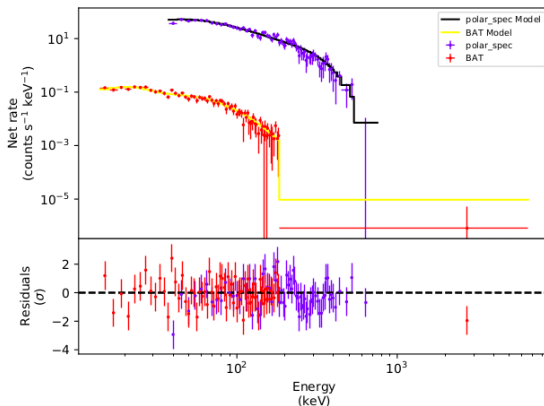


POLAR Geant4: Digitization

- Electronics response is handled in second MC C++ package
- Converts deposited energy to optical photons
- Converts optical photons to photo-electrons in PMT
- Conversion based on optical simulations (separate in G4)
- Converts PMT signal into electrical signal
- Handles trigger logic
- Output is equivalent to POLAR data



- Digitization contains all parameters and their dependence on environment
- e.g. channel gain vs HV, channel gain vs temp, threshold vs temp etc etc
- Splitting digitization from G4 speeds things up
- G4 part is relatively simple and can be run once
- G4 part is not dependent on instrument conditions (e.g. temperature, HV etc.)
- Cross calibration against other satellites



- Calibration shows ARF and RMF of each channel match measurement
- Once this works, polarization is easy

