ECal amplification chain: simulation

Gabriel Charles

Institut de Physique Nucléaire d'Orsay

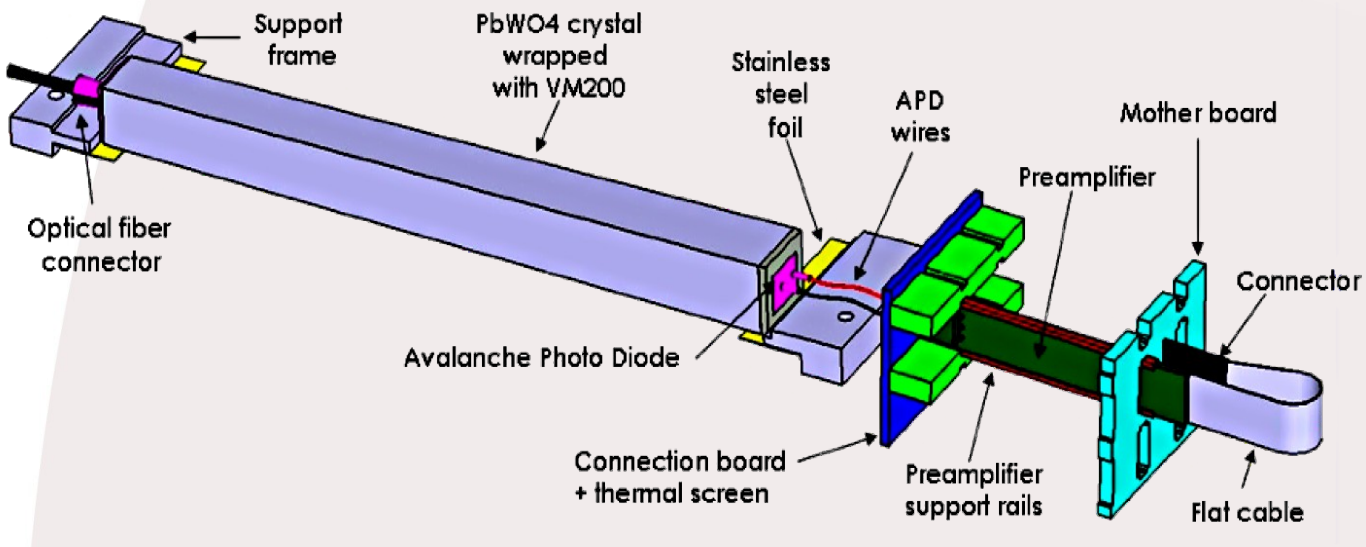
CNRS-IN2P3
Université Paris-Sud

Unité mixte de recherche

CNRS-IN2P3
Université Paris-Sud

91406 Orsay cedex
Tél. : +33 1 69 15 73 40
Fax : +33 1 69 15 64 70
http://ipnweb.in2p3.fr

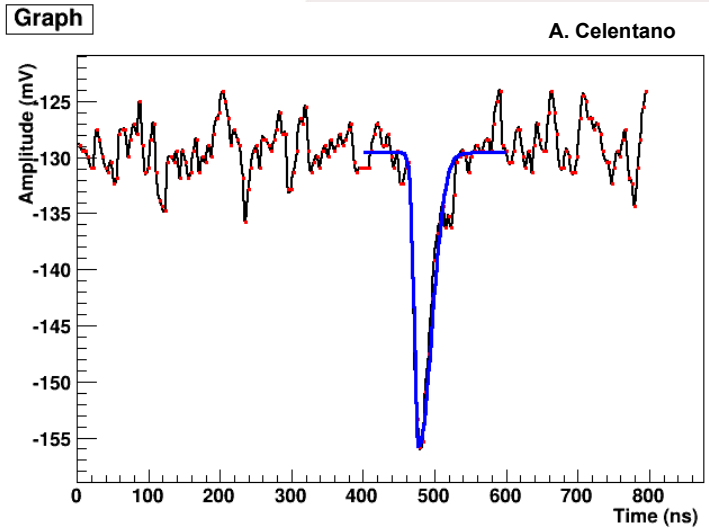
Ecal amplification chain



Goals:

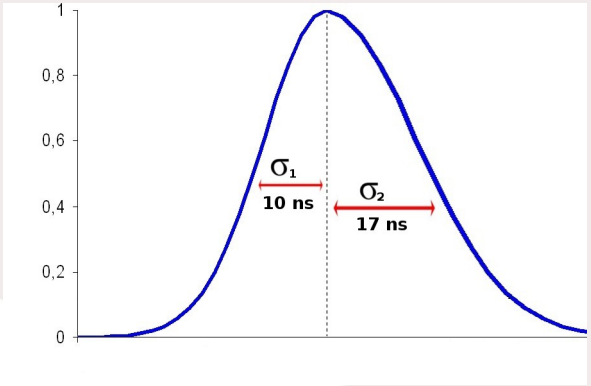
➔ **Simulate the response of the new amplification chain**

Shape formula



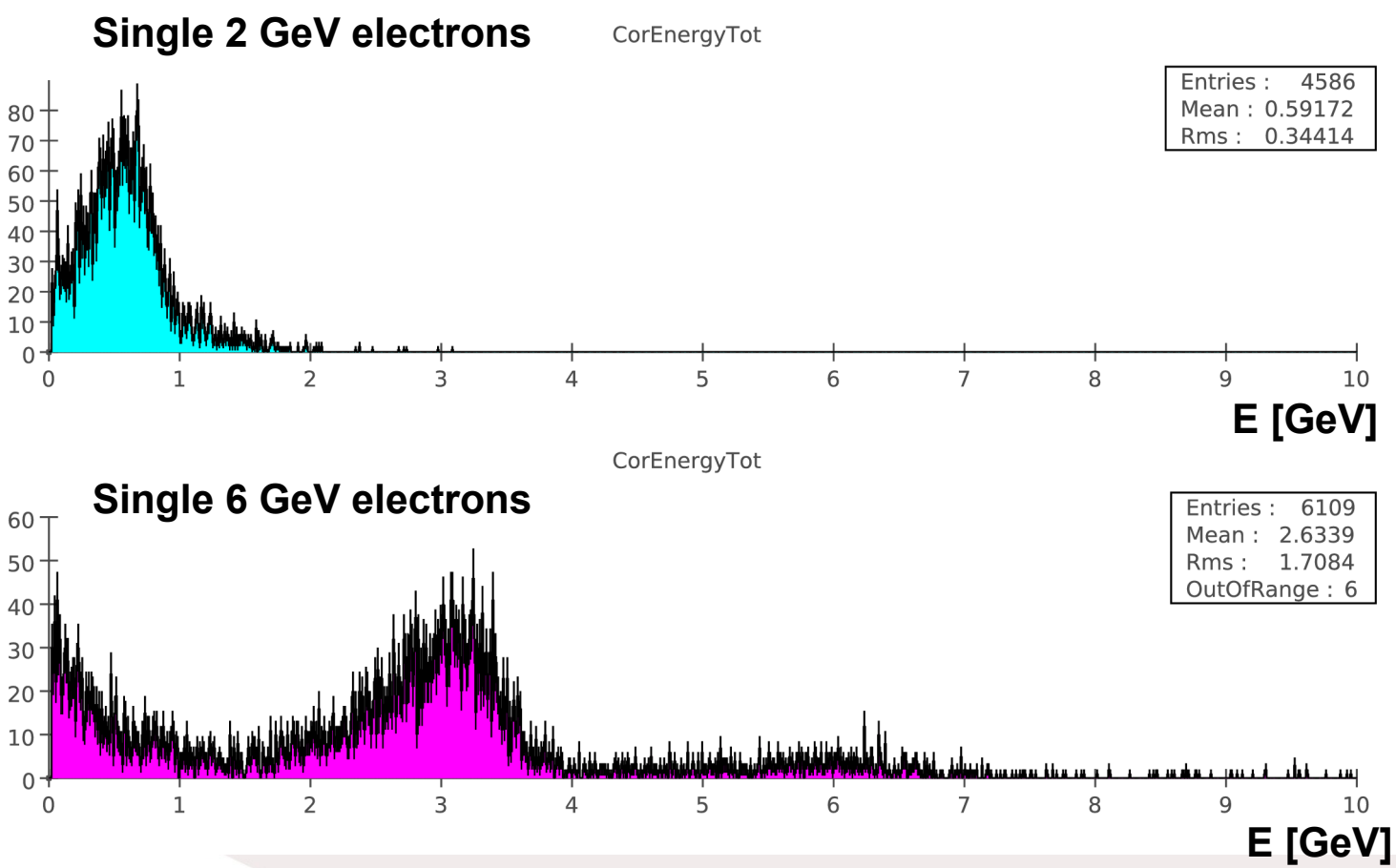
Output signal obtained with a crystal + APD + preamplifier

The new pulse shape impulsion is based on this measurements. It is composed, as shown by Andrea, by two Gaussians of different standard deviations but with the same minimum amplitude



Energy reconstruction

Sum of the energy collected by all the crystals with a threshold of 10 MeV



Use of the new shape impulsion

```
public class FADCSignalAnalysis extends Driver {

    private AIDA aida = AIDA.defaultInstance();
    FADCEcalReadoutDriver readoutDriver = new FADCEcalReadoutDriver();
    EcalRawConverterDriver converterDriver = new EcalRawConverterDriver();
    IHistogram1D h1d_sumbuffer_en = aida.histogram1D("signal buffer energy", 1000, 0.0, 10);

    IHistogram1D hitEnergyPlot = aida.histogram1D("Energy Plot", 1000, 0.0, 10);
    IHistogram1D hitCorEner = aida.histogram1D("CorEnergy", 1000, 0.0, 10);
    IHistogram1D hitCorEnerTot = aida.histogram1D("CorEnergyTot", 1000, 0.0, 10);
    IHistogram2D h2d_ehit_ebuffer = aida.histogram2D("signal buffer energy vs. EcalHit energy", 1000, 0.0, 2.2, 1000, 0.0, 2.2);

    @Override
    public void startOfData() {

        add(readoutDriver);
        readoutDriver.setCoincidenceWindow(2);
        readoutDriver.setEcalName("Ecal");
        readoutDriver.setEcalCollectionName("EcalHits");
        readoutDriver.setEcalRawCollectionName("EcalRawHits");
        readoutDriver.setConstantTriggerWindow(false);
        readoutDriver.setScaleFactor(1);
        readoutDriver.setFixedGain(1);
        readoutDriver.setUseCRRCShape(false);

        add(converterDriver);
        converterDriver.setRawCollectionName("EcalRawHits");
        converterDriver.setEcalCollectionName("EcalCorrectedHits");
        converterDriver.setGain(1.0);
        converterDriver.setUse2014Gain(true);

        super.startOfData();
    }
```