



UCL

Standardized calibration interface proposal

DUNE calibration TF meeting – 5 Dec 2017

Chris Backhouse – University College London

Introduction

- ▶ Discussion at Physics Week about implementing calibration systems
- ▶ Need to be able to tweak what reco thinks the calibration is
 - ▶ Central location affecting all algorithms
 - ▶ Ideally independently of simulation

- ▶ State of the art is `DetectorPropertiesService`?
- ▶ I suspect there is also reco code that hardcodes various numbers

- ▶ Affects other experiments too (shared `larreco` code)
- ▶ Operating expt. like MicroBooNE has much more elaborate needs
- ▶ Aim to specify a shared calibrator interface we can all use
- ▶ Individual experiments / studies will swap in their implementation

- ▶ Current draft at `/dune/app/users/bckhouse/ICalibrator.h`
- ▶ Deliberately minimalistic
- ▶ Some feedback from sim/reco last week, to larsoft meeting next

```

/// Interface class for Calibrators. Classes implementing this interface can
/// connect to a database, or use simple parameterizations, etc etc
class IChargeCalibrator
{
public:
    /// Useful in this form if you have a hypothetical hit some reconstruction
    /// algorithm needs to consider. NElectrons is useful because it's the
    /// correct value to use Poisson statistics on.
    double NElectrons(double integral, raw::ChannelID_t chan) = 0;

    /// Most people should use GeV directly though. pos is the best estimate of
    /// where the hit originated in 3D space.
    double GeV(double integral,
               raw::ChannelID_t chan,
               TVector3 pos) = 0;

    /// Convenience functions for concrete hits
    double NElectrons(const recob::Hit& hit)
    {
        return NElectrons(hit.Integral(), hit.Channel());
    }

    double GeV(const recob::Hit& hit, TVector3 pos)
    {
        return NElectrons(hit.Integral(), hit.Channel(), pos);
    }

    /// Convert a time between flash and hit into a position in the detector,
    /// using the drift velocity. In principle this could vary spatially? So
    /// include the channel number. TODO: I think wire wrapping throws a
    /// spanner in the works here and we need to include a TPC identifier too.
    double DriftCoordinate(raw::ChannelID_t chan, double dt) = 0;

    double DriftCoordinate(const recob::Hit& hit, double t0)
    {
        return DriftCoordinate(hit.Channel(), hit.PeakTime()-t0);
    }
};

```

```

/// May want to swap in dummy charge and photon calibrations in various
/// combinations.
class IPhotonCalibrator
{
public:
    double NPhotons(double integral, int opdet) = 0;
    /// Need a 3D position because result depends on position along length of
    /// bar. This is going to be pretty imprecise even so.
    double GeV(double integral, int opdet, TVector3 pos);

    /// Convenience
    double NPhotons(const recob::OpHit& oh)
    {
        return NPhotons(oh.Area(), oh.OpChannel());
    }

    double NPhotons(const recob::OpFlash& of)
    {
        // This function would be in the .cxx in practice
        const std::vector<double>& pes = of.PEs();
        double ret = 0;
        for(int chan = 0; chan < pes.size(); ++chan)
            ret += NPhotons(/* TODO: flash is already in PEs, not "area"
                           chan);
        return ret;
    }

    double GeV(const recob::OpHit& oh, TVector3 pos)
    {
        return GeV(oh.Area(), oh.OpChannel(), pos);
    }

    double GeV(const OpFlash& of, TVector3 pos)
    {
        // This function would be in the .cxx in practice
        const std::vector<double>& pes = of.PEs();
        double ret = 0;
        for(int chan = 0; chan < pes.size(); ++chan)
            ret += GeV(/* TODO: flash is already in PEs, not "area"
                       chan, pos);
        return ret;
    }
};

```

Discussion

- ▶ Sim/reco were broadly happy
- ▶ Good feedback about DP needs incorporated

- ▶ Main remaining point is handling of Birk's suppression
- ▶ NOvA story:
 - ▶ Decided the calibrator shouldn't be concerned with any correction that depends on the particle hypothesis
 - ▶ Were thinking about threshold corrections (depend on dE/dx)
 - ▶ Also applies to Birks, though that crept up on us
 - ▶ In practice "GeV" number from calibrator has Birks for muons
 - ▶ This is all reasonable, but naming is a little confusing
- ▶ Thoughts?