Kalman Filter Toolkit Status

LArSoft Tracking Meeting
Apr. 3, 2012

H. Greenlee

Outline

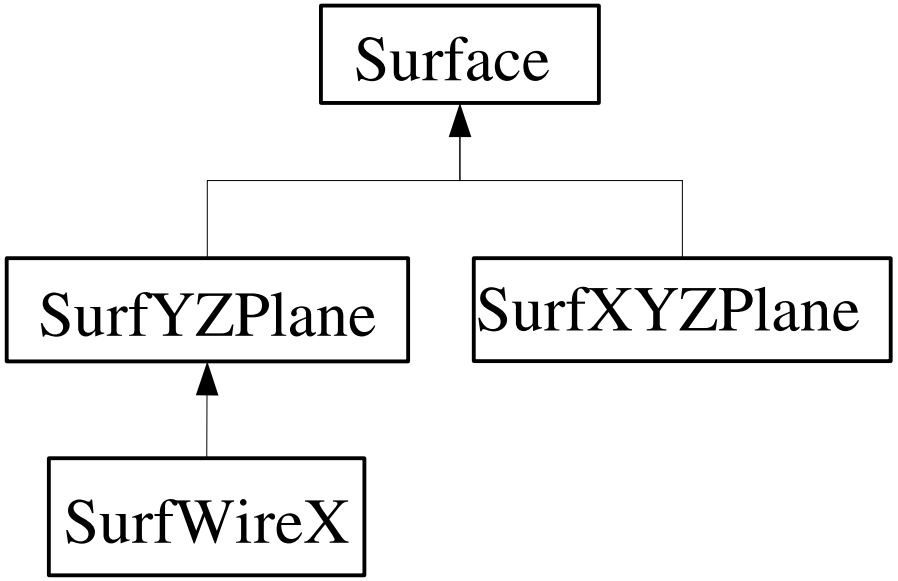
- Toolkit classes (do not depend on art).
  - Linear Algebra.
  - Surfaces.
  - Tracks.
  - Measurements.
  - Propagators.
  - Measurement containers.
- Services/modules.
  - KalmanFilterService service.
  - TrackKalmanCheater module.

Boost and Linear Algebra

- I am using boost linear algebra package (ublas).
  - In the current design, all objects are preallocated on the stack using exact dimensions specified at compile time.
  - Ublas is sophisticated in the computer science sense, not so much in a mathematical sense.
    - Good enough for Kalman filter (I still think it is the right choice).
    - Had to write my own symmetric matrix inversion.
- I found some other boost packages to be useful.
  - shared_ptr<T> reference counted smart pointer.
  - optional<double> return value with validity flag (effectively bool + double).

Surface Classes

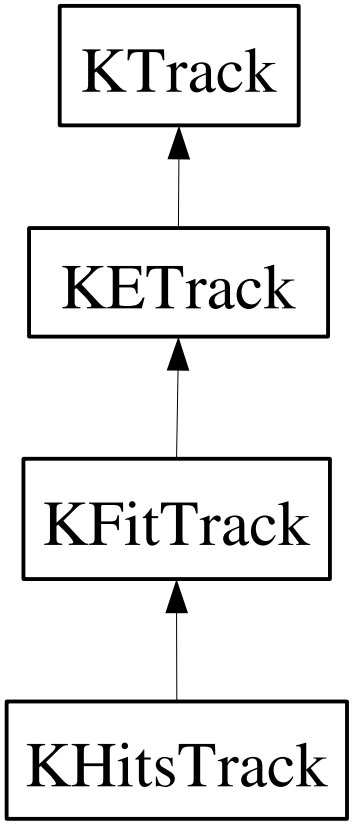
- Surface – Virtual base class.
- SurfYZPlane – Plane parallel to x-axis.
  - $y0$, $z0$, ϕ.
- SurfXYZPlane – General plane.
  - $x0$, $y0$, $z0$, θ, ϕ.
- SurfWireX – Adds constructor to calculate SurfYZPlane parameters from channel number using geomtry system.



- SurfYZPlane is good enough to implement Kalman filter for nonmagnetic detector.
- No bounded surfaces (yet).

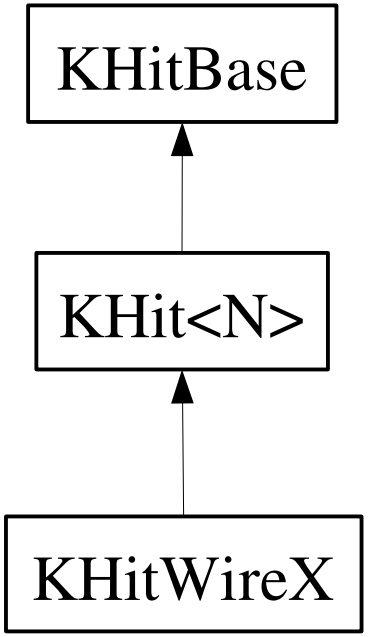
Track Classes

- KTrack – Track without errors.
  - Surface.
  - Track parameters.
  - Surface direction parameter.
- KETrack – Track with errors.
  - Error matrix.
- KFitTrack – Track with fit.
  - Propagation distance.
  - Chisquare.
- KHitsTrack – Track with measurements.
  - Associated measurements.



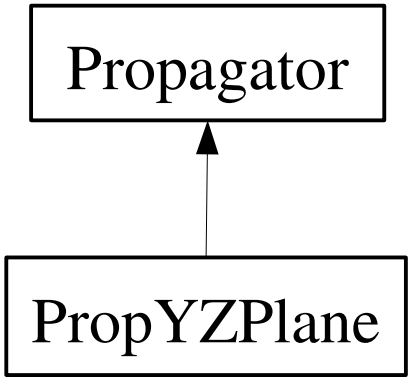
Measurement Classes

- KHitBase – Virtual base Class.
  - Measurement surface.
  - Methods “predict” and “update,” (basic Kalman filter methods).
- KHit<N> – Measurement class with dimension specified by integer template parameter N.
  - Owns linear algebra objects for measurement.
  - Full implementation of method “update.”
  - Partial implementation of method “predict.”
- KHitWireX – Time (x) measurement on wire.
  - Remaining implementation of method “predict.”



Propagator Classes

- Propagator – Virtual base class. Has three methods/use cases:
  - Method vec_prop – Propagate without error.
  - Method err_prop – Propagate with error, but without noise.
  - Method noise_prop – Propagate with error + noise.
  - Only first of above methods is virtual.
- PropYZPlane – Propagator for SurfYZPlane surfaces.
  - Only simple straight line implementation so far. No multiple scattering or energy loss. Also no magnetic field.



Measurement Containers

- KHitGroup
  - Collection of measurements (KHitBase) on the same surface.
- KHitContainer
  - Collection of KHitGroup objects.
  - Methods to sort measurements by estimated path length, based on track hypothesis (seed track or partially reconstructed track).
- Above classes exist, but are not in svn.

Kalman Filter Service

- Prototype KalmanFilterService exists to implement Kalman Filter algorithm. Various use cases are envisioned.
  - Basic Kalman fit using known set of measurements.
  - Kalman filter, including pattern recognition, starting from 3D seed track.
  - Either of above can be 2D or 3D.
  - Initial implementation only for first use case (fit using known measurements).
- Initial version exists (one use case), but not in svn.

TrackKalmanCheater Module

- Interface to KalmanFilterService known measurement use case using MC truth hits.
  - Makes KHitWireX measurement objects from RecoBase/Hit objects.
  - Fills Kalman filter measurement container.
  - Invokes fit use case in KalmanFilterService.
- Exists, but not in svn.

Performance of Kalman Fit

- Kalman fit has difficulty to converge if hits from multiple views are used, even under ideal conditions of known hits and known starting track parameters.
  - Problem seems to be lack of precision of *y* slope with a limited number of measurements.
    - Update formula sends *y* slope to some crazy value when fitting the first few measurements. This messes up subsequent propagations.
    - *Y* slope seems to need a longer lever arm than is afforded by a few adjacent hits.
    - Problem is not fundamentally different than the space point combinatoric problem.

Performance of Kalman Fit (cont.)

- Kalman fit works OK if hits from only one view are used (provided starting u and v errors are not correlated in measurement surface coordinate system).
  - $u = x$ (axis parallel to drift direction).
  - v = axis parallel to wire.
- If u and v position and slope errors are not correlated, v slope will not be updated by Kalman fit.

Some Possible Solutions to 3D Kalman Fit Problem

- Do fitting and/or pattern recognition in each view separately (or do one view first), them merge views.
- Combine hits into 2D or 3D segments. Use segments as measurements.
- Modify order that hits are added to Kalman fit to increase lever arm.

Tasks

- Implement multiple scattering and energy loss in propagator.
- Implement fully functional 2D (single view) Kalman fit.
- Try different strategies to reach 3D (e.g. merge 2D tracks).
- It seems feasible to do pattern recognition in one view using existing Kalman filter tools.
  - Similar to what Ben originally suggested.
- Extract track momentum using multiple scattering and/or range.