TWIST Pattern Recognition

The Problem: Identifying the individual tracks from the hit wires for each event.

- In most cases different tracks correspond to different hit times. However important cases such as Bhabha's deltas, hard scatters will be coincident.
- The PC's determine the track time. The DC's drift time is sufficiently long that we can have hits possibly belonging to two or more tracks (overlaps).
- The helix pitch will often be comparable to the interchamber distances except for the dense stack – not the usual situation in HEP, but the case for MEGA.
- Failure to recognise tracks within our angular and energy range could introduce a bias to our results.

Methodology

- Group hits into time windows including overlaps.
- Form hit wire clusters in each plane and possible hit *uv* points.
- Find sets of helix parameters consistent with groups of these hits.
- Use these helix parameters aka first guess tracks to guide the Kalman filter.

Organization

Module window_mod: entry point windowing

- Calculates time windows and hit clusters

Module tracking_mod: Entry point tracking

- Steering routine feeding points to pattern recognition.
- Calls Kalman filter
- We may have to iterate these.

Module track_mod: Entry point addcluster

- Contains track_type and FGTrack
- Calculates 1st guess track helicies.

Outlook

- Track down clustering glitches.
- Try search starting from dense stack points and other search strategies.
- Read simulation banks and display true helicies.
- Display by window or other attribute.
- Integrate with Kalman.