Imperial College London

# Tracker Reconstruction Status

A. Dobbs

Imperial College London

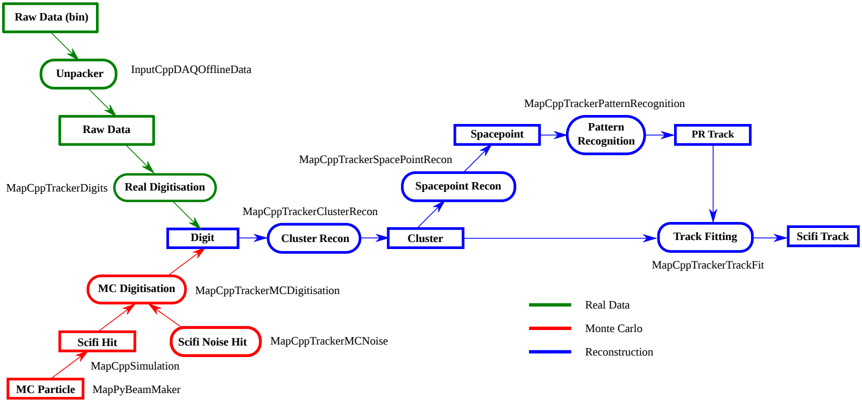
Tracker Software Review, 27th April 2017

# The Team

**Paul Kyberd**
Head of Tracker Software

- **Melissa Uchida** — Online Recon, Efficiency
- **Ed Overton** — Digitisation, Calibration, Mapping
- **Adam Dobbs** — Pattern Recognition, Efficiency
- **Chris Hunt** — Final Track Fit, Performance

# A Quick Overview of the Tracker Software




*A. Dobbs et al, The reconstruction software for the MICE scintillating fibre trackers, JINST, Vol:11, T12001*

# Reconstruction

- **Digitisation**: Channel mapping and calibration → channel hits with npe (digits)
- **Clustering**: group digits from neighbouring channels into clusters
- **Spacepoint recon**: Collect clusters from a different planes in a single station using Kuno's conjecture → spacepoints $(x, y, z)$
- **Pattern Recognition**
  - Collect spacepoints to form initial track
  - Perform initial linear least squares fit to seed final fit
- **Final track fit**: Kalman filter → final tracks and trackpoints (postion and momentum at each plane)

# Pattern Recognition Track Model

- Transverse $(x, y)$ projection: circle params $(x_c, y_c, r)$
- Longitudinal $(z, s)$ projection: straight line params $(c_{zs}, m_{zs})$
- Relations: $\phi' = \tan^{-1}\left(\frac{y-y_c}{x-x_c}\right)$, $\phi = \phi' + 2n\pi$, $s = r\phi$
- $s$ - distance travelled around circle in $(x, y)$ plane, $\phi'$ - observed turning angle, $\phi$ - true turning angle
- Finding $n$: exploit varying station separation: $\frac{\Delta\phi_{ji}}{\Delta\phi_{ki}} = \frac{\Delta z_{ji}}{\Delta z_{ki}}$
- Linearising the circle fit:

$$\alpha(x^2 + y^2) + \beta x + \gamma y + \kappa = 0$$

$$x_c = -\frac{\beta}{2\alpha}, \quad y_c = -\frac{\gamma}{2\alpha}, \quad r = \sqrt{\frac{\beta^2 + \gamma^2}{4\alpha^2} - \frac{\kappa}{\alpha}}$$

# Efficiency Methodology

- Efficiency here is analysed for Pattern Recognition specifically
- Using real data we look for events where a good 5 point track is expected:
  - 1 spacepoint only in both TOF1 and TOF2;
  - muonic time-of-flight;
  - exclude events with enough spacepoints to form 2 or more tracks;
  - optionally further require 1 spacepoint and 1 only in each tracker station ("ideal events");
  - each tracker is looked at independently of the other.
- Code available in MAUS:

```
bin/scifi/pat_rec_efficiency.py maus_output.root
```

# Progress

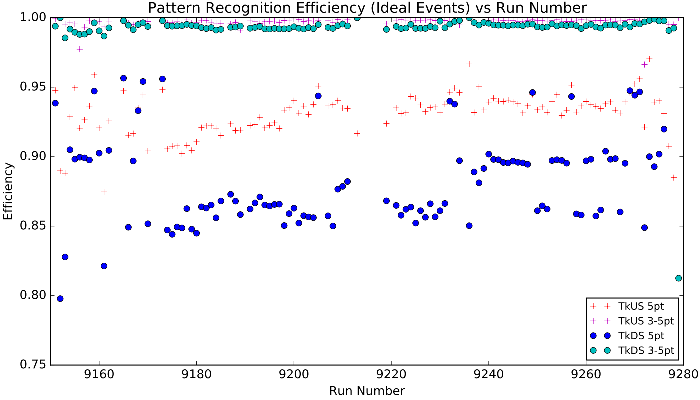
- Real data, run 8681
- Ideal event cuts

| Version | TkUS 5pt | TkUS 4-5pt | TkDS 5pt | TkDS 4-5pt |
|---|---|---|---|---|
| 2.6.5 | 0.5774 | 0.9141 | 0.3954 | 0.827 |
| 2.7.0 | 0.8754 | 0.9912 | 0.7625 | 0.9723 |
| 2.8.3 | 0.9272 | 0.9986 | 0.8452 | 0.9951 |

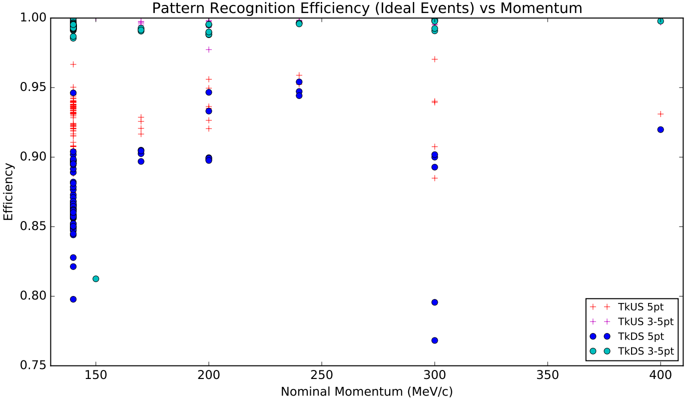
Efficiency results

# Efficiency vs Run Number

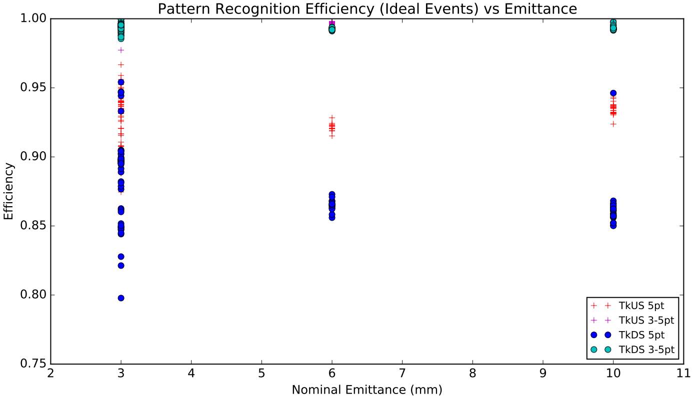
MAUS v2.8, ideal events



# Efficiency vs Momentum



# Efficiency vs emittance



# Upgrades

- MINUIT-based circle fit - *Implemented, but no increase in performance seen over our custom* $\chi^2$ *fit*
- Missing spacepoint search algorithm - if 4 point track found, check the missing the empty station for an unused spacepoint, and add if it close enough to the track - *Implemented, large increase in 5pt track efficiency, final performance plots unaffected*
- Consider changing track model to avoid singularity at $p_t = 0$
- Consider effect of scattering on efficiency

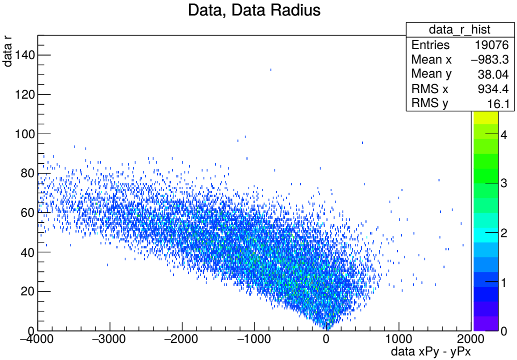
# Missing Spacepoint Search

| Search (mm) | TkUS 5pt | TkUS 4-5pt | TkDS 5pt | TkDS 4-5pt |
|---|---|---|---|---|
| 0 | 0.927 | 0.999 | 0.845 | 0.996 |
| 2 | 0.943 | 0.999 | 0.889 | 0.996 |
| 5 | 0.967 | 0.999 | 0.937 | 0.996 |
| 8 | 0.984 | 0.999 | 0.965 | 0.996 |
| 10 | 0.990 | 0.999 | 0.974 | 0.996 |
| 15 | 0.994 | 0.999 | 0.983 | 0.996 |

Spacepoint Search Efficiency results

# Low $p_t$ singularity

Courtesy of V. Blackmore (CM47, MAUS 2.7):




- Seen in real data and reconstructed MC
- Contour of missing tracks occurs at zero canonical angular momentum i.e. **when $\mathbf{p_t = 0}$**

**Q.** Is this still present in MAUS 2.8?

**Q.** Can we change the track model to remove this singularity?

# Next Steps

- Evaluate spacepoint search effect on final track performance
  - Initial results indicate it makes little to no difference
- Look for $p_t = 0$ missing tracks feature using MAUS 2.8+
- If neccessary come up with a revised track model to remove the singularity