

# The Reconstruction Software for the MICE Scintillating Fibre Trackers

A. Dobbs, C. Hunt, K. Long, E. Santos, M. A. Uchida

*Physics Department, Blackett Laboratory, Imperial College London*

*Exhibition Road, London, SW7 2AZ, UK*

P. Kyberd

*Brunel University London, Kingston Lane, Uxbridge, Middlesex, UB8 3PH, U.K.*

C. Heidt

*University of California Riverside, 900 University Ave, Riverside, CA 92521, U.S.A.*

The Muon Ionization Cooling Experiment (MICE) will demonstrate the principle of muon beam phase-space reduction via ionization cooling. Muon beam cooling will be required for the proposed Neutrino Factory or Muon Collider. The phase-space before and after the cooling cell must be measured precisely. This is achieved using two scintillating-fibre trackers, each placed in a solenoidal magnetic field. This paper describes the software reconstruction for the fibre trackers: the GEANT4 based simulation; the geometry and configuration implementations; digitisation; spacepoint reconstruction; pattern recognition; and the final track fit based on a Kalman filter. The performance of the software is evaluated by means of Monte Carlo studies and the precision of the final track reconstruction is evaluated.

## 1 The MICE Experiment

### 1.1 Overview

The Muon Ionization Cooling Experiment (MICE) will perform a practical demonstration of muon ionization cooling. Cooling refers to a reduction in the emittance of a beam, that is, the reduction of the phase-space volume occupied by the beam. Beam cooling is required for any future facility based on high intensity muon beams, such as a Neutrino Factory [1], the ultimate tool to study leptonic CP-invariance violation, or a Muon Collider [2], a potential route to multi-TeV lepton – anti-lepton collisions. Muon beams are generated via pion decay, and therefore have a large emittance, which must be reduced so that a reasonable fraction of the beam will fall within the acceptance of the downstream acceleration system.

The short muon lifetime requires fast beam cooling which traditional techniques are unable to provide. Ionization cooling was proposed in the early 1970s [3, 4], but has not yet been demonstrated at the energies of interest for the Neutrino Factory or Muon Collider. Ionization cooling reduces emittance by passing a beam through some suitable material of low atomic number such as hydrogen. This leads to the reduction of all components of momentum due to ionization energy loss. Low atomic number absorbers are preferred because they minimise multiple scattering which “heats” the beam.

MICE is based at the Science and Technology Facilities Council Rutherford Appleton Laboratory in the U.K., using the ISIS proton synchrotron to generate the muon beam [5]. The MICE beamline is described in detail in [6]. A schematic of the full MICE experiment is shown in figure 1. MICE has completed the first step of its programme, consisting of the muon beamline with particle identification in which the MICE muon beam was characterised. The next step in the programme, which introduces the trackers and the absorber module began taking data in 2015.

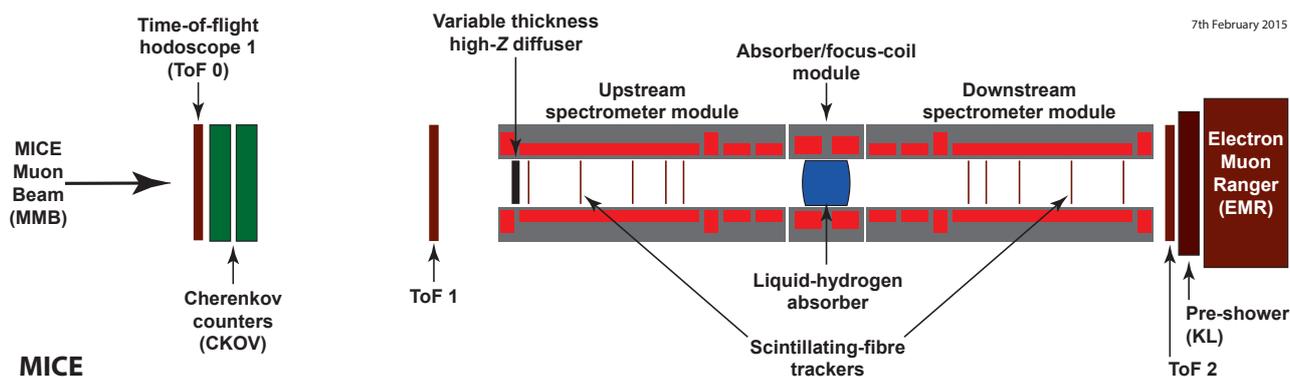


Figure 1: The beam diagnostics and cooling channel. The diffuser is used to increase the beam emittance prior to cooling.

ncludes longitudinal re-acceleration, will begin data taking in 2018.

## 1.2 The Scintillating Fibre Trackers

MICE is equipped with two identical, high precision scintillating-fibre (“scifi”) trackers, described in [7]. Each tracker is placed in a superconducting solenoid that provides a uniform field over the tracking volume. One tracker, TKU, is upstream of the cooling cell, the other, TKD, downstream. Each tracker consists of 5 detector stations, labelled 1 to 5, as illustrated in figure 2. TKU, is orientated such that Station 5 sees the beam first, TKD is rotated by  $180^\circ$  such that Station 1 sees the beam first, thus, in both trackers, Station 1 is always nearest to the cooling channel (see figure 1).

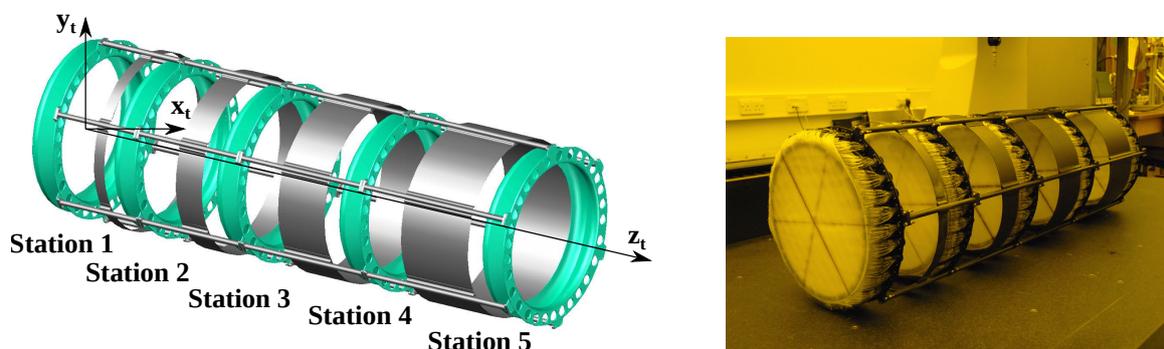


Figure 2: Left: A schematic of the tracker carbon fibre frame, showing the detector station positions. The fibre planes are glued on to the upstream edge (smaller  $z_t$  position) of the carbon fibre station frames (shown in green). Right: A photograph of a tracker. The colour is due to the filtered lighting needed to protect the fibres. The intersecting lines visible on the station faces indicate the direction of the fibres in each plane.

Each station is formed of three planes of  $350 \mu\text{m}$  scintillating fibres, orientated at 120 degrees to each other. The fibres in each plane are arranged in two layers offset with respect to each other, known as a “doublet layer”, in order to give 100% coverage of the plane area as illustrated in figure 3. The doublet layer is glued onto a sheet of mylar. The fibres are collected into groups of seven for readout, each group forming a single channel, as illustrated in figure 3b. The planes, also known as views, are labelled  $U$ ,  $V$  and  $W$ . Plane  $U$  is attached to the station frame directly, plane  $W$  on to plane  $U$ , and plane  $V$  on to plane  $W$ . The fibre-plane orientations are illustrated in figure 4. Each station is oriented such that the fibres in the  $U$  plane are vertical. The fibres produce scintillation light when ionizing radiation passes through them. Clear-fibre light guides transport the

scintillation light to visible light photon counters (VLPCs) in a cryostat, the signal from which is digitised via analogue-to-digital converters (ADCs). See [7] for details.

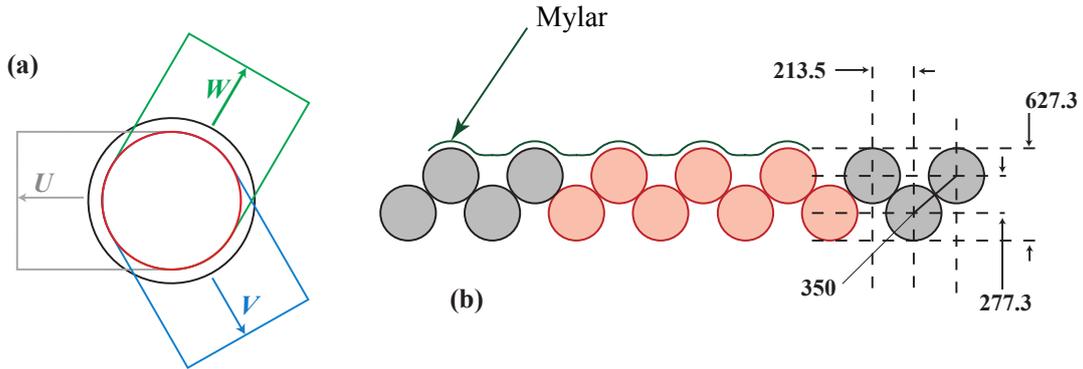


Figure 3: (a) Arrangement of the doublet layers in the scintillating fibre stations. The outer circle shows the solenoid bore while the inner circle shows the limit of the active area of the tracker. The arrows indicate the direction that the individual  $350\ \mu\text{m}$  fibres run. (b) Detail of the arrangement of the scintillating fibres in a doublet layer. The fibre spacing and the fibre pitch are indicated on the right-hand end of the figure in  $\mu\text{m}$ . The pattern of seven fibres shown in red form a single channel, which is readout via a clear-fibre light-guide. The sheet of mylar plastic glued to the doublet layer is indicated.

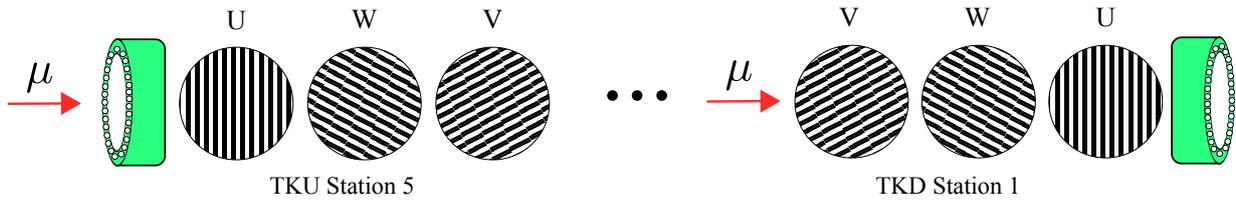


Figure 4: The orientation of the fibres in each plane, as seen by the incoming beam, for both trackers. The green object is the station frame.

## 60 2 Coordinate systems and reference surfaces

The planes, stations and trackers themselves each have a coordinate system defined on them which will be described in the following sections. Coordinates with subscript  $p$  will refer to the plane coordinate system,  $s$  to the station coordinate system and  $t$  to the tracker coordinate system.

### 2.1 Channels and digits

65 The  $V$  and  $W$  planes each consist of 214 channels, labelled 0 to 213, while the  $U$  plane has 212 channels, labelled 0 to 211. The channel number increases from left to right if a plane is placed mylar side up, with the fibre readout pointing downwards, as illustrated in figure 5.

## 2.2 Planes and clusters

Each plane is assigned an integer known as the plane number: plane  $V$  is assigned 0; plane  $W$  is assigned 1; and plane  $U$  is assigned 2. The plane reference surface is defined to be the flat plane that is formed by the outer surface of the mylar sheet. The measured position perpendicular to the direction of the fibres in each plane is labelled  $\alpha \in (v, u, w)$ , defined to increase in the *opposite* direction to the channel number with zero as the mid-point of the central channel.  $\alpha$  is then given by  $\alpha = (N_{CC} - N_{Ch}) \times d$ , where  $N_{CC}$  is the central channel number,  $N_{Ch}$  is the channel number and  $d$  is the channel width.

The  $z$  axis of the plane coordinate system is defined to be perpendicular to the plane reference surface and points in the direction from the mylar sheet towards the fibres. The direction in which the fibres run defines the final plane coordinate,  $\beta$ , completing a right-handed coordinate system. The origin of the  $(\alpha, \beta)$  coordinate system is taken to be at the centre of the circular active area of the plane.

## 2.3 Stations and spacepoints

The station reference surface is defined to coincide with the reference surface of the  $V$  doublet-layer. The station coordinate system is defined such that the  $x_s$  axis is coincident with the  $v$  coordinate (that is,  $\alpha$  for the  $V$  layer), the  $z_s$  axis is coincident with the  $z_p$  axis of the  $V$  layer and the  $y_s$  axis completes a right-handed coordinate system.

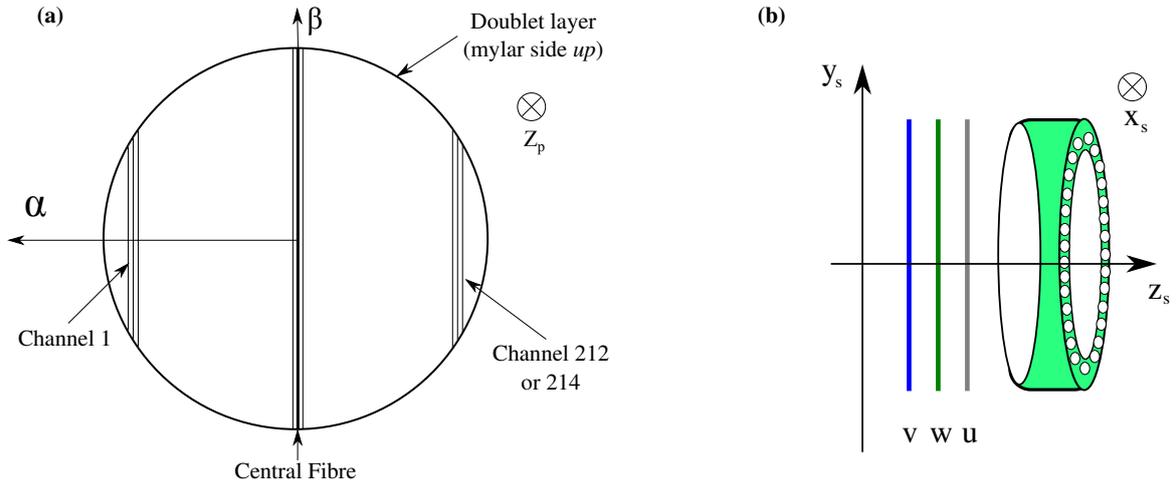
## 2.4 Trackers and tracks

Each tracker is assigned an integer known as the tracker number: TKU is assigned 0 and TKD is assigned 1. The tracker reference surface is defined to coincide with the reference surface of Station 1. The tracker coordinate system is defined such that the  $z_t$  axis coincides with the axis of cylindrical symmetry of the tracker as shown in figure 2. The tracker  $z_t$  coordinate increases from Station 1 to Station 5. The tracker  $y_t$  axis is defined to coincide with the  $y_s$  axis of Station 1 and the tracker  $x_t$  axis completes a right-handed coordinate system.

## 3 The MAUS framework

The tracker software is part of the MICE software framework, known as MAUS (MICE Analysis User Software) [8]. MAUS is used to perform Monte Carlo simulation and both online and offline data reconstruction. It is built using a combination of C++ and Python, with C++ being used for more processor-intensive tasks and Python being used more in the code presented to the user. Simulation is based on GEANT4 [9], with analysis based on ROOT [10]. ROOT files are used as both the primary input and output data format. MAUS also reads in the custom binary format written by the MICE data acquisition system (DAQ).

MAUS programmes are defined in a Python script together with a configuration file. This script allows the user to create programmes by combining different MAUS modules depending on the task at hand, following the Map-Reduce programming model [11]. The object passed between the modules is known as a “spill”, representing the data associated with one spill of particles passing through the MICE beamline (see [6]). The modules come in four types: Input; Output; Map; and Reduce. Input modules provide the initial data to MAUS, from a data file, or from the DAQ. Maps perform most of the simulation and analysis work and may be processed in parallel across multiple nodes. Reducers are used to display output, such as for online reconstruction plots, and are capable of accumulating data sent from maps over multiple spills, but must be run in a single thread. Output modules provide data persistency.



Fibre run to the station optical connectors towards the bottom of the figure

Figure 5: (a) The channel numbering within a plane, and the  $(\alpha, \beta, z_p)$  plane coordinate system (a right-handed system). (b) The fibre plane ordering with respect to the station body and the station coordinate frame (a right-handed system). In TKU the beam approaches from the right, in TKD from the left. Note that  $z_s$  is by definition equivalent to  $z_p$  of the  $V$  plane.

The tracker software consists of four maps and a reducer. The maps cover: digitisation of Monte Carlo data; digitisation of real DAQ data; the addition of noise to Monte Carlo data; and reconstruction. The reducer provides real time information on the tracker performance.

## 4 Data Structure

### 110 4.1 General MAUS, Monte Carlo and DAQ data structures

A simplified schematic of the tracker data structure, with the relevant entries from the more general MAUS data structure, is shown in figure 6. All the objects listed represent container classes for different parts of the simulation, raw data and reconstruction. The top-level object is the spill (see section 3). Within the spill the data is split into three branches: real-data from the DAQ; Monte Carlo data generated by simulation; and  
 115 reconstructed data, which is formed from data in either the real or Monte Carlo branches.

The reconstruction code makes no direct reference to the Monte Carlo information and has no way to distinguish real from simulated data, thus ensuring that they are treated equally. The DAQ data is held in an object known as TrackerDAQ, within which the data is sub-divided according to the DAQ system it originated from.

The Monte Carlo event holds data on “scifi hits” produced by tracks passing through the fibre planes and  
 120 any noise hits originating in those planes. The scifi hit is implemented as a class based on the generic hit-class template from which all the different MICE detector-hit classes are derived (see [8]). Other relevant data held in the Monte Carlo event, though not part of the tracker data structure, includes simulated track objects which hold the generated information on position, momentum and particle type. Such data is used to evaluate the reconstruction performance against the generated data (see section 8).

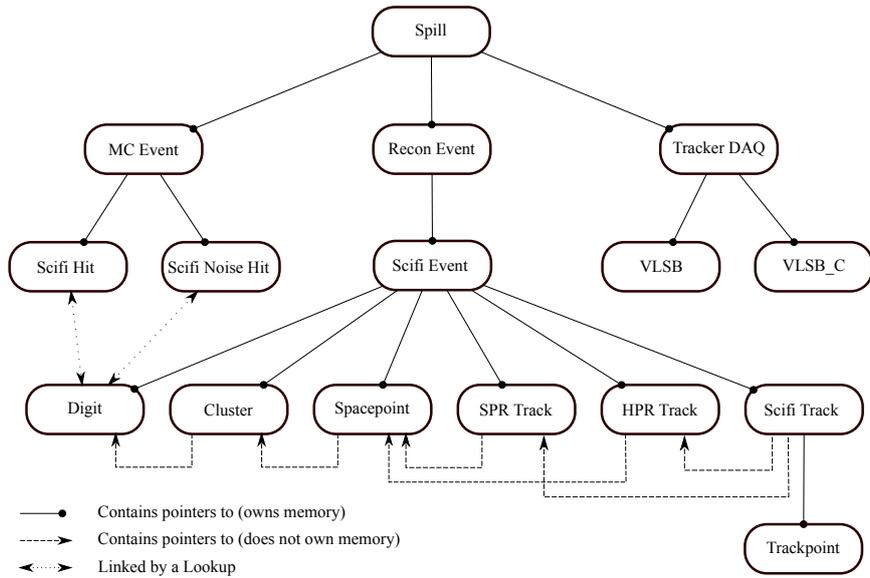


Figure 6: The tracker software data structure, and relevant MAUS data structure. The spill is the top level object below which branches hold real-data, MC data and reconstructed data objects. When an object owns the memory of a set of other objects, these are held as standard vectors of pointers. When an object contains cross links to another set of objects, without owning their memory, these are held as a ROOT TRefArray of pointers. MC = Monte Carlo, SPR = Straight Pattern Recognition, HPR = Helical Pattern Recognition.

## 125 4.2 Tracker reconstruction data structure

The reconstructed data for the tracker is held in the “scifi event” class. This contains vectors of the following container classes that represent the higher-level reconstructed tracker-data:

- “Digits” contain the ADC counts (real or simulated) from the readout of a single channel in response to an incident track;
- 130 • “Clusters” are groups of neighbouring digits arising from a particle crossing one or two channels;
- “Spacepoints” group clusters from adjacent detector planes to give a point in space in terms of  $(x, y, z)$ ;
- “Straight pattern recognition tracks” group together spacepoints from different tracker stations when the track that introduced the hits was straight (i.e. when the enclosing magnetic field is off);
- 135 • “Helical pattern recognition tracks” group together spacepoints from different tracker stations when the track that introduced the hits was helical (i.e. when the magnetic enclosing field is on);
- “Scifi tracks” hold the final Kalman fit parameters of the particle track; and
- “Trackpoints”, which hold the fit parameters at each detector reference plane, including the momentum and position of the track.

Each of these objects is stored directly in the scifi event object (the event “owns” the required memory), with the exception of trackpoints which are stored in the associated scifi tracks. Each higher-level object also contains cross links in the form of pointers back to the objects within the scifi event which were used to create it. In this manner all higher-level objects can be traced back to the original digits. In the case of a Monte Carlo run the digits themselves are linked via an ID number and lookup table back to the scifi hits used to produce them. The ID is defined as  $tspc$ , where  $t$  is the tracker number,  $s$  is the station number,  $p$  is the plane number and  $c$  is the channel number (given with three numerals e.g. 010 for channel 10). This structure means the reconstruction branch has no direct reference to the Monte Carlo data.

Tracker 1 offsets in mm					
	Station 1	Station 2	Station 3	Station 4	Station 5
X	0.0	-0.5709	-1.2021	-0.5694	0.0
Y	0.0	-0.7375	-0.1657	-0.6040	0.0
Z	-1099.7578	-899.7932	-649.9302	-349.9298	0.0
Tracker 2 offsets in mm					
	Station 1	Station 2	Station 3	Station 4	Station 5
X	0.0	-0.4698	-0.6717	0.1722	0.0
Y	0.0	0.0052	-0.1759	-0.2912	0.0
Z	-1099.9026	-899.009	-650.0036	-350.0742	0.0

Table 1: The position of the tracker stations with respect to the tracker reference surface as measured by the coordinate measuring machine.

## 5 Geometry

The position of each tracker station was determined with respect to the tracker reference surface using a coordinate measuring machine (see table 1). The station positions are stored in the MICE configuration database (CDB). The CDB is a bi-temporal database, alterations being tracked by date and run number. Information is stored in the CDB as a collection of XML files which are translated into the native MAUS format “MiceModules”, at run-time. The MiceModules are text documents and contain all the information needed to simulate the various MICE systems and detectors. MAUS uses the same geometry descriptions for both simulation and reconstruction.

In the description of the geometry MAUS adopts a passive rotation convention to be consistent with GEANT4. The active volume of each tracker is given by a cylinder of 150 mm radius, which is used to define the fiducial volume for the reconstruction. Alignment of the individual tracking stations and the trackers themselves to the solenoid axis has recently been completed using real-data.

## 6 Simulation

The simulation of the trackers makes use of the GEANT4 standard physics libraries to describe particle motion through the fields and material of the beamline. The trackers are simulated on a per-fibre basis and arranged into doublet-layer planes (as described in section 1.2). As particles pass through the fibres scifi hits are generated, containing the energy deposited in the fibre.

The scifi hits are converted by the MAUS module used to digitise simulated tracker data (MapCppTrackerMCDigitisation) into to a number of photoelectrons (NPE) produced in the tracker VLPCs by means of a simple conversion factor. At this point the NPE values are non-integers, and are quantised simply by rounding down. These NPE values are then “smeared” to simulate the detector response (the electron avalanche effect in the VLPCs). The smearing process involves modeling this response as a Gaussian, the mean being given by the quantised NPE value and the width being determined from data. Once the smearing is complete the signal is split into  $2^8$  bins to represent the sampling size of the 8 bit ADCs. The simulated ADC counts are then used together with the measured calibration for each channel to give a final NPE value.

It is also possible to add noise to digitisation process by the addition of an extra MAUS module (MapCppTrackerMCNoise). Noise may arise in the signal from thermally excited electrons within the VLPCs, known as “dark count”. The dark count is a stochastic process described by a Poisson distribution. Physically the rate

175 can be changed by altering the bias voltage on the VLPCs, with a target dark count rate of 1 NPE in 1.5% of the fibres per particle trigger, with higher numbers of NPE having a smaller probability following the Poisson distribution. The effect is modelled in the software and used to introduce additional photoelectrons to the simulated signal prior to the quantisation and smearing stage.

180 Once the final NPE value has been calculated it is combined with the channel number to form a digit object. The digits are then added to the scifi event and sent on to the reconstruction modules.

## 7 Reconstruction

The reconstruction process begins with either real or simulated data and proceeds to reconstruct progressively higher-level objects step-by-step, culminating in scifi tracks and trackpoints. The digits, real or simulated, are passed to the same MAUS module (MapCppTrackerRecon) and the reconstruction proceeds identically for either case from that point. The reconstruction process is illustrated in figure 7 and is described in the sections that follow.

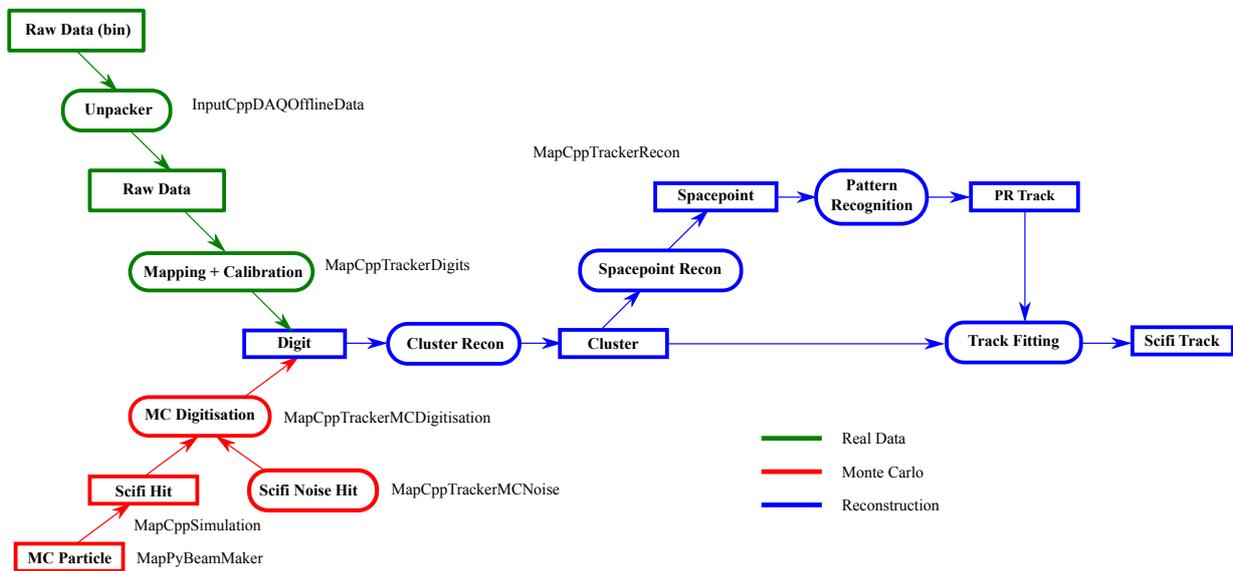


Figure 7: The reconstruction data flow. Data originates either from simulated or real-data, the two branches merge after digitisation, after which the reconstruction proceeds identically. The relevant MAUS modules for each step are indicated.

### 7.1 Digitization

190 For real-data the signal from the tracker ADCs is recorded by the DAQ system. Channel-by-channel calibration constants are used to convert the ADC value to a signal in NPE and the DAQ channel number to tracker channel number. This information is then used to form a digit. The analogous process for Monte Carlo data is described in section 6.

## 7.2 Clustering

The clustering algorithm loops over every combination of pairs of digits in a scifi event and combines any that occur in neighbouring channels. In the case of multi-digit cluster, the unweighted average channel value is used to define the plane coordinate,  $\alpha$ , and the NPE is summed.

## 7.3 Spacepoint Reconstruction

For each station the constituent planes are searched for clusters that can be used to form a spacepoint. Spacepoints are constructed from clusters from all three planes (a triplet spacepoint) or for any two out of the three planes (a doublet spacepoint).

To determine which clusters from each plane originate from the same track, Kuno's conjecture [7] is used: for a given triplet spacepoint the sum of the channel numbers of each cluster will be a constant. So if  $n^u$ ,  $n^v$  and  $n^w$  are the fibre numbers of the clusters in  $u$ ,  $v$  and  $w$  and  $n_0^u$ ,  $n_0^v$  and  $n_0^w$  are the corresponding central channel numbers. Three clusters form a space point if:

$$|(n^u + n^v + n^w) - (n_0^u + n_0^v + n_0^w)| < K . \quad (1)$$

where  $K$  is a constant, taken by default to be 3.0.

Once all triplet spacepoints have been found, doublet spacepoints are created from pairs of remaining clusters, the only selection criteria applied being that the crossing point of the two channels is within the tracker radius.

Spacepoint positioning is determined from the cluster measurements,  $\alpha$  (see section 2.2) and the known plane orientations for those measurements. All clusters selected as belonging to the same spacepoint must intersect. Using two clusters the second plane coordinate  $\beta$  may be calculated using knowledge of the plane orientations. This then expresses the crossing point in the  $(\alpha, \beta)$  coordinate system, which can be simply rotated into the station  $(x, y)$  coordinate system. In the case of a triplet spacepoint three crossing points are present, the average of which is used to define the final spacepoint position.

## 7.4 Pattern Recognition

Pattern recognition is based on looping over different combinations of spacepoints and performing a fit using a linear least squares technique. There are separate algorithms for the straight track (no field) case and the helical track case.

### 7.4.1 Helical Pattern Recognition

The helical pattern recognition is performed in cylindrical co-ordinates  $(r, \phi, z)$ , where  $r$  is the helix radius,  $\phi$  the turning angle in the transverse plane, and  $z$  is the longitudinal coordinate. An additional coordinate,  $s$  the distance the particle travels in the transverse plane, is also defined. The helix is then parameterised by: the circle centre  $x_c, y_c$  and the radius,  $r$ , in the transverse plane;  $s_0$  the value of  $s$  where the helix crosses the reference plane; and  $t_s = ds/dz$ , which describes the tightness of the coiling.

To find a track one spacepoint is selected from each station and a circle is fitted in the  $(r, \phi')$  projection. If the  $\chi^2$  of this fit is sufficiently small (by default less than 15.0 multiplied by the number of degrees of freedom, NDF) then the value of  $\phi$  is used to generate  $s$  and a straight line fit is performed in the  $(z, s)$  plane, and if the  $\chi^2$  in this projection is also small (by default less than 4.0 multiplied by the NDF) the track is accepted as a track candidate. Once all the different possible combinations of spacepoints have been tried the track candidate with

the lowest combined  $\chi^2$  from the fits is selected as the true track. Once every possible track with a spacepoint in all 5 stations has been found, the procedure is repeated looking for tracks with spacepoints in only 4 of the 5 stations.

## 7.4.2 Straight Line Pattern Recognition

The straight-track finding is performed in Cartesian coordinates. The track parameters are  $(x_0, y_0, t_x, t_y)$  where:  $(x_0, y_0)$  is the position the track crosses the tracker reference surface;  $t_x = dx/dz$ ; and  $t_y = dy/dz$ . Two spacepoints are chosen in the outer stations and a road is created between them. Any spacepoints in the road are fitted in the  $(x, z)$  and  $(y, z)$  planes. If the  $\chi^2$  of the fits is small (by default less than 4.0 multiplied by the NDF) a candidate track is formed. Once all possible spacepoint combinations have been tried the track candidate with the lowest  $\chi^2$  is selected. As in the helical case, following the completion of the full 5 point track search, tracks with spacepoints in 4 out of the 5 stations are searched for. For the straight case only, following the completion of the search for 4 point tracks, a search is also made for tracks with spacepoints in only 3 out of the 5 stations.

## 7.5 Track Fit

The final track fit was implemented using a track-orientated Kalman filter [12, 13], which can be shown to be an optimal linear fitter, that takes into account all correlations and measurements, for a linear system. The Kalman filter is an iterative algorithm that incrementally propagates an estimate of the current track state between measurement planes, using measurement information to “filter” the state, improving the estimate of the track state.

For the helical track fit, the system is only approximately linear, hence an extended Kalman filter [14] was implemented which analytically propagates the track states between measurements, while the covariance matrices are propagated using a first-order linear approximation to the non-linear system.

Pattern recognition provides a set of clusters that are associated with a track, which passed the selection criteria, and a parameterisation of that track based on a least squares fit to the points by a helix or a straight line as appropriate. The track parameters calculated from the least squares fit are used to provide the seed for the Kalman fit and the raw cluster information is used as the measurement data. The flexibility of the Kalman algorithm permits the effects of individual planes (multiple coulomb scattering MCS, and energy loss) and the material effects of the helium gas within the tracker to be accounted for between each measurement point.

The Kalman filter was written entirely from scratch. In order to make the implementation flexible and reusable the core algorithm was implemented without any dependencies on phase-space dimensions or physical effects. The principal components of the implementation are: propagation routines for both helical and straight tracks; a measurement routine that transforms the track state-space into the measurement state-space; and approximations for the process and measurement noise.

The measurements correspond to individual clusters, hence the parameter  $\alpha$  (see section 2.2) forms a one-dimensional measurement state. The measurement noise corresponds to the statistical spread of measurements of width  $w$ , within a single channel in the tracker readout. If the channel is modelled as a top-hat function, the variance of the function is  $w^2/12$ . Therefore the measurement noise was taken to be  $w/\sqrt{12}$  for all clusters.

The process noise was implemented as a combination of MCS and energy straggling. The energy loss is calculated using the Bethe formula, for the most probable energy loss, and applied during the propagation stage. The noise term itself is calculated per increment as an RMS scattering angle using an implementation of

the Highland formula (equation 2), in a fashion almost identical to the GEANT4 implementation:

$$\theta_{RMS} = \frac{13.6\text{MeV}/c}{\beta c p} Z \frac{x}{X_0} \left( 1 + 0.038 \ln \left( \frac{x}{X_0} \right) \right) \quad (2)$$

where  $\theta_{RMS}$  is the RMS scattering angle through a finite length of material,  $\beta c$ ,  $p$  and  $Z$  are the velocity, momentum and charge of the particle in question,  $x$  is the distance travelled and  $X_0$  the radiation length of the material. Although the RMS scattering angle does not form a Gaussian distribution, the first order approximation is believed to be sufficient.

265 In contrast to Pattern Recognition, the Kalman Filter uses a different state space model,  $(x, p_x, y, p_y, q/p_z)$  rather than  $(x_c, y_c, r, s_0, t_s)$ . This permits the energy loss and multiple coulomb scattering to be modelled linearly and implemented using a more simple set of algorithms. The longitudinal components however are less well modelled due to small angle scattering, such that the reconstruction of  $p_z$  from the final track fit is sensitive to the accuracy of the pattern recognition stage.

## 270 8 Performance

A Monte Carlo simulation was used in order to measure the reconstruction efficiency and resolution of the implemented track-finding and track-fitting algorithms. It was necessarily Monte Carlo based in order to compare the reconstructed events to the simulated truth, thereby highlighting any inefficiencies and inaccuracies within the algorithms.

275 The fitted transverse positions,  $(x, y)$ , and momenta,  $(p_x, p_y, p_z)$ , were compared to the Monte Carlo truth on an event-by-event basis. The reconstruction of the Monte Carlo data followed the same requirements that are used for the real-data reconstruction. The Monte Carlo truth data was stored at every tracker plane to permit a direct comparison with the reconstructed data. All comparisons were made at the tracker reference surface.

280 An artificial beam was generated with uniform distributions for both the longitudinal and transverse momenta. This ensured that the results were not biased by the incoming beam distribution and that the full reconstructible phase-space was probed with equal statistics. In order to remove unwanted non-physical particles, a cut was placed on the Monte Carlo “truth” data during the analysis phase to eliminate tracks with a large  $p_t/p_z$  ratio. Tracks with a ratio greater than 0.5 were rejected from the analysis.

### 8.1 Kuno’s Conjecture

285 All clusters selected to form triplet spacepoints should follow Kuno’s conjecture (section 7.3). Plot showing the sum of the channel numbers for the clusters in each spacepoint is shown in figure 8. As expected the sum of the channel numbers for all the clusters in a spacepoint is constant to very good approximation, with a small variation arising from scattering.

### 8.2 Track Finding Efficiency

290 For every simulated event, the number of tracks expected was calculated from the Monte Carlo truth. If the simulated track crossed enough tracker planes to create a sufficient number of spacepoints (3 for straight tracks and 4 for helical tracks), a reconstructed track was expected. The reconstructed tracks were compared to the expected track parameters. The efficiency of track finding as a function of the true longitudinal and transverse momenta is shown in figure 9.

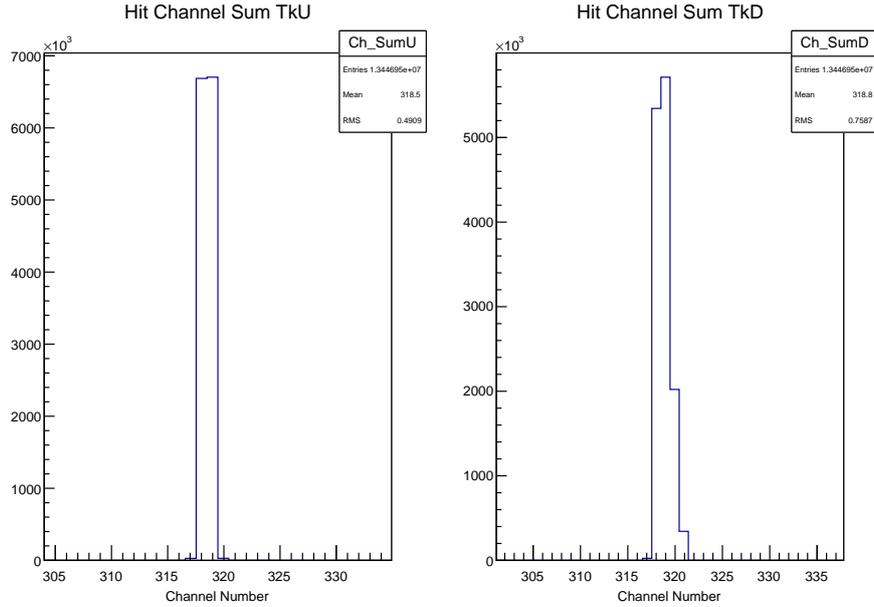


Figure 8: Plot showing the sum of the channel numbers for the clusters in each spacepoint. Kuno’s conjecture that the sum is a constant is observed, with a small variation due to scattering.

295 Once the the Monte Carlo tracks have been identified, the expected number of trackpoints can be calculated by examining the number of tracker planes that the simulated track crossed. Comparing the number of trackpoints in each reconstructed track to the expected number for each simulated track permits the efficiency of finding the correct number of trackpoints to be calculated. Figure 10 shows the trackpoint finding efficiency as a function of longitudinal and transverse momenta.

### 300 8.3 Position and Momentum Resolution

The position residuals, shown in figures 11 and 12, are consistent with the expected measurement resolutions for a combined fit and the absolute spread is very close to the width of a channel (1.497 mm). The transverse momentum resolution, shown in figure 13, is consistent across the range of the sensitive phase-space at  $\sim 1.2$  MeV/c in both trackers. The longitudinal momentum shown in figure 14, an intrinsically more difficult  
 305 measurement for the tracker, still retains an acceptable spread of  $\sim 4$  MeV/c in both trackers. There is however a small ( $< 0.1$  MeV/c) systematic effect, visible in the momentum distributions.

In order to produce these plots, a requirement that there was a cluster within the reference plane was applied. Due to the effects of Multiple Coulomb Scattering, on rare occasions a single hard scatter can cause pattern recognition to miss a single spacepoint at the reference plane, hence creating a tail that will adversely affect the  
 310 distributions. As we are concerned with the resolution following a successful pattern recognition stage, these events were removed.

The track fit assumes a simple model of energy loss based on the mean thickness of materials used in the tracker construction, however the effects of glues and resins, in addition to the non-uniform densities of the materials are not considered. This results in systematic underestimate for the energy lost at each tracker plane  
 315 and a small sytematic bias during the pattern recognition stage, which doesn’t attempt to model these effects. Due to the high-granularity of the position information utilised by the fit, the position reconstruction is much less sensitive to any discrepancies in momentum. Figure 15 shows the mean deviation in total reconstructed

momentum across the typical momentum acceptance of the trackers.

320 Trends in transverse and longitudinal momentum resolution as a function of transverse momentum are shown in figures 16 and 17. A consistently uniform distribution is found in the transverse momentum as expected, with the predominant issue found in the longitudinal reconstruction of low- $p_t$  tracks. This effect, when coupled with variations in efficiency will yield some systematic concerns in the reconstruction of statistical quantities such as emittance. Studies of these systematic biases are currently under way.

## 9 Conclusion

325 The performance for the final Kalman filter based track fit has been evaluated by comparing Monte Carlo truth with reconstructed data, for the key tracker measurements of  $x$ ,  $y$ ,  $p_t$  and  $p_z$ . The observed performance in the transverse position and momentum measurements is excellent, for both the upstream and downstream trackers. The reconstruction resolution in  $p_t$  and  $p_z$  are within expectation and will provide a sufficient degree of precision for the MICE physics program.

330 There are however systematic effects present in the momentum reconstruction due to the complexity of precisely modelling the density and thicknesses of the various materials within the tracker. The Monte Carlo model provides a more detailed description than could be feasibly implemented by the reconstruction, hence producing the residuals seen in section 8.3. This discrepancy will be representative of a physical systematic effect in the reconstruction of data and corresponds to the leading systematic effect in the tracker reconstruction.

335 Monte Carlo studies will be used to model the momentum discrepancy and provide a linear correction based on a fit to the data shown in figure 15. The linear correction can then be used during analyses to reduce the effect of this systematic. Comparisons with other detectors in the MICE experiment may provide a data-driven estimate for this correction thereby supporting the Monte Carlo model. In addition it would be possible to further extend the track fit such that an Adaptive Kalman Filter is used, whereby the energy loss per plane is both modelled and estimated by the track fit. Such extensions are currently under discussion.

340 Due to the precision of the track fit the key measurement of MICE, the transverse emittance of a muon beam, can be very well estimated. The emittance is based on the precise determination of the beam covariance matrix, and is therefore systematically sensitive to the resolution of the reconstruction. The resolutions of the individual parameters are small and symmetric which permits a simple covariance matrix correction to be applied, fully accounting for the measurement effects of the tracker. Additionally, the high resolution reduces the number of muons required to achieve the expected statistical precision of MICE.

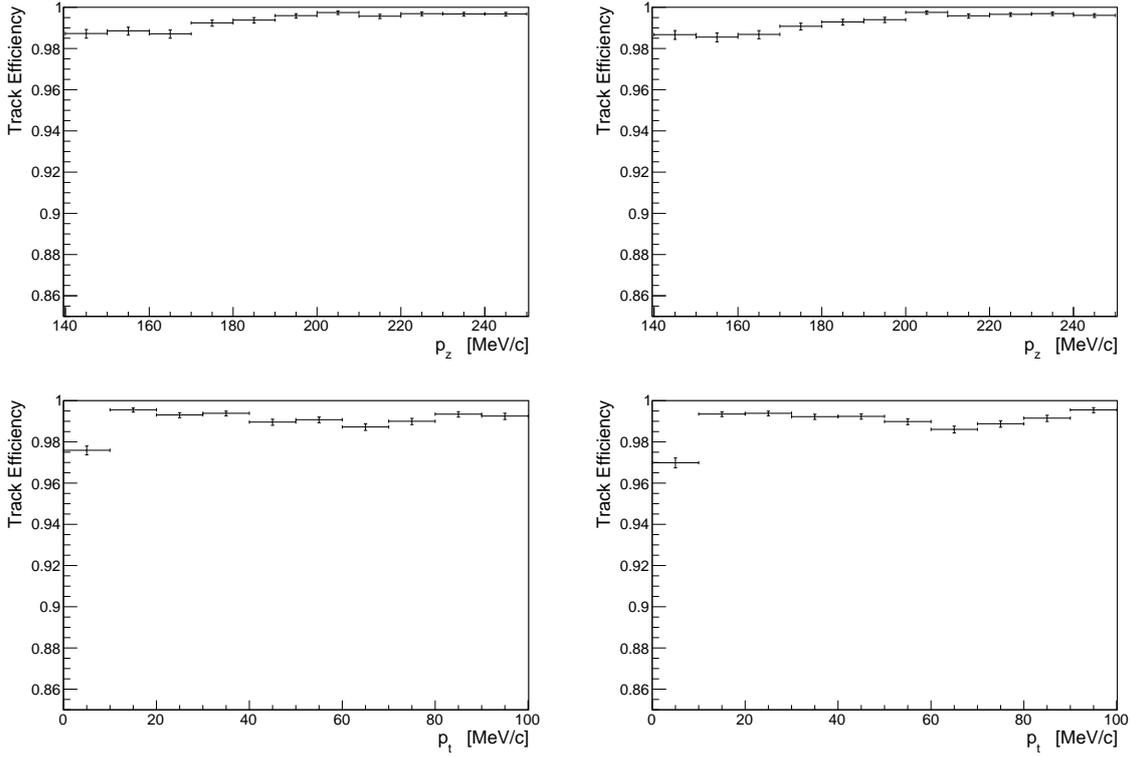


Figure 9: The efficiency of reconstructing tracks in the upstream (left) and downstream (right) trackers as a function of the simulated longitudinal (top) and transverse (bottom) momentum.

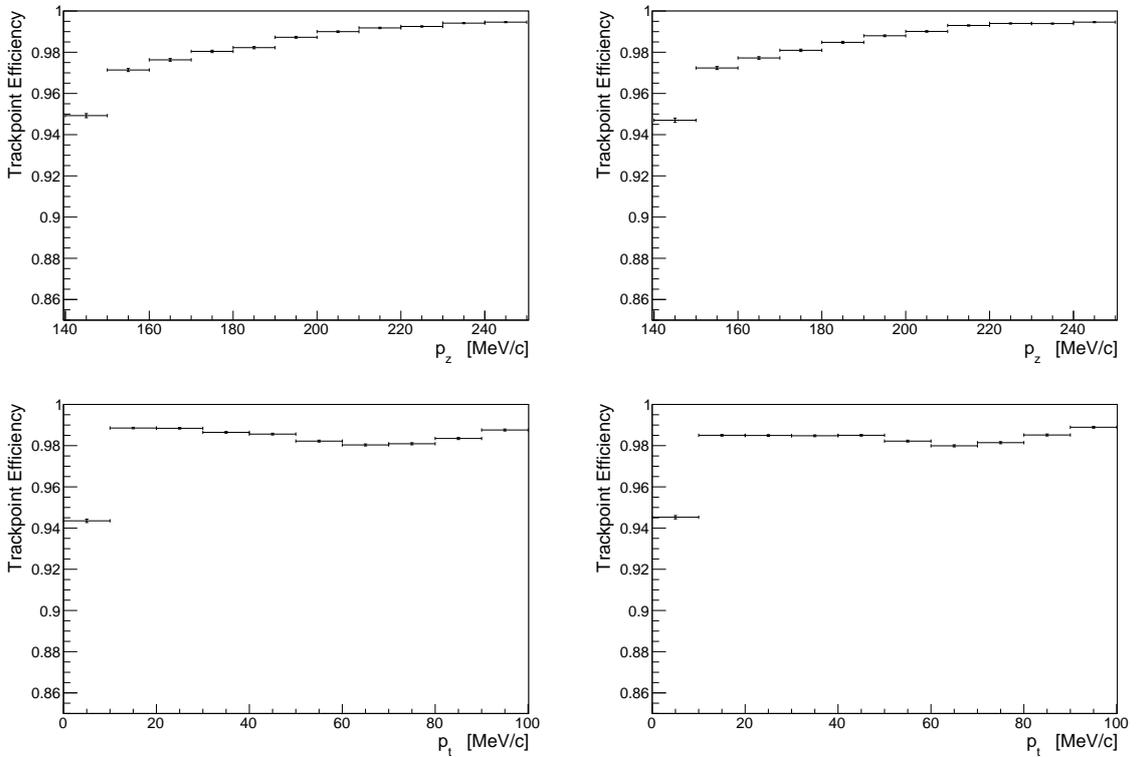


Figure 10: The efficiency of reconstructing trackpoints in the upstream (left) and downstream (right) trackers as a function of the simulated longitudinal (top) and transverse (bottom) momentum.

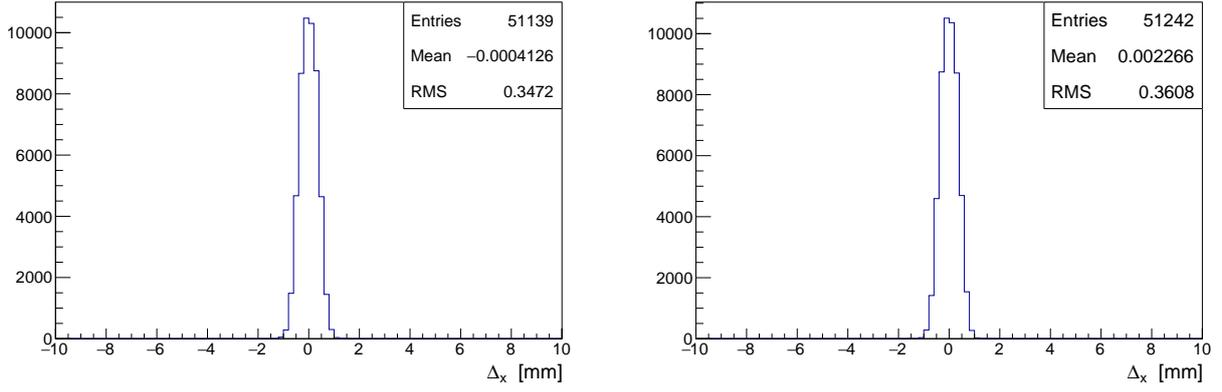


Figure 11: The  $x$  residuals of the upstream (left) and downstream (right) trackers.

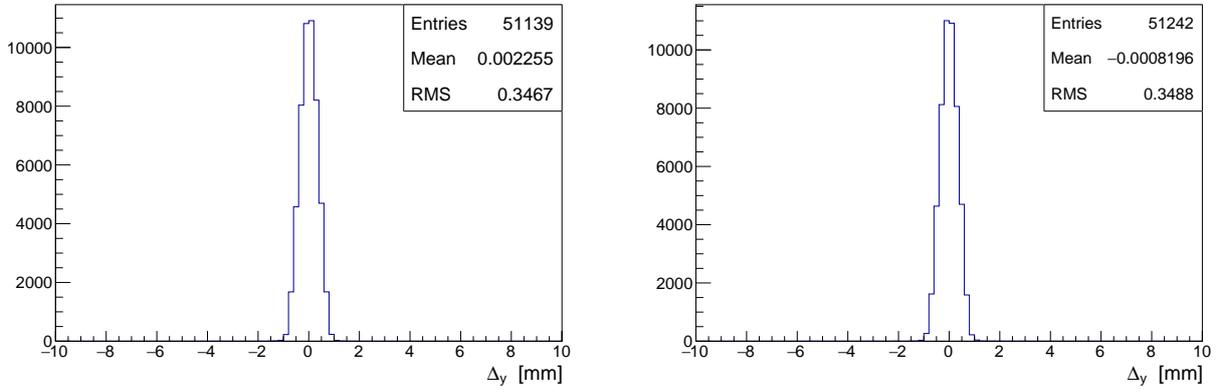


Figure 12: The  $y$  residuals of the upstream (left) and downstream (right) trackers.

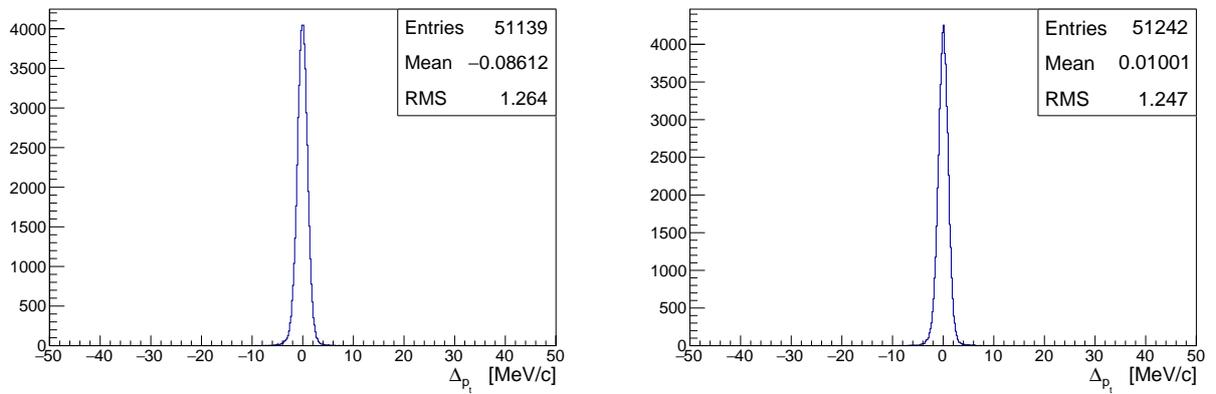


Figure 13: The  $p_t$  residuals of the upstream (left) and downstream (right) trackers.

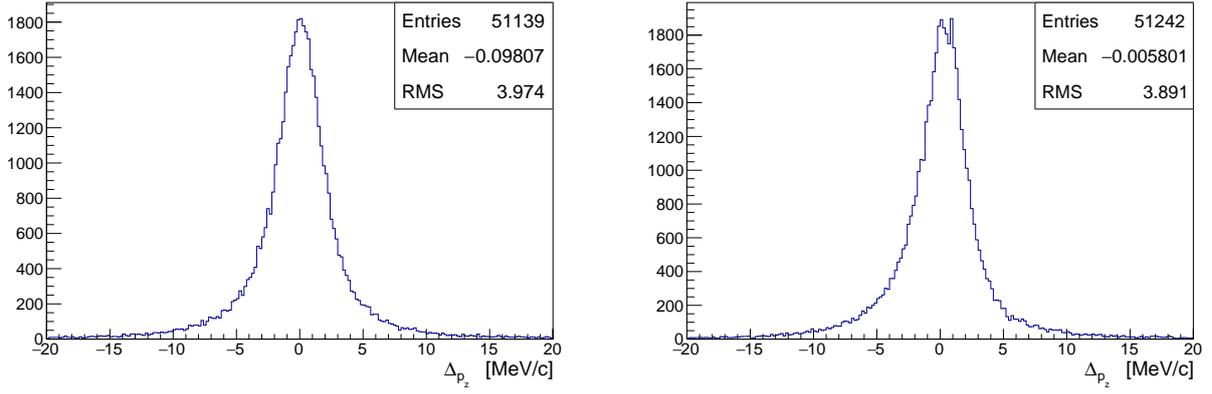


Figure 14: The  $p_z$  residuals of the upstream (left) and downstream (right) trackers.

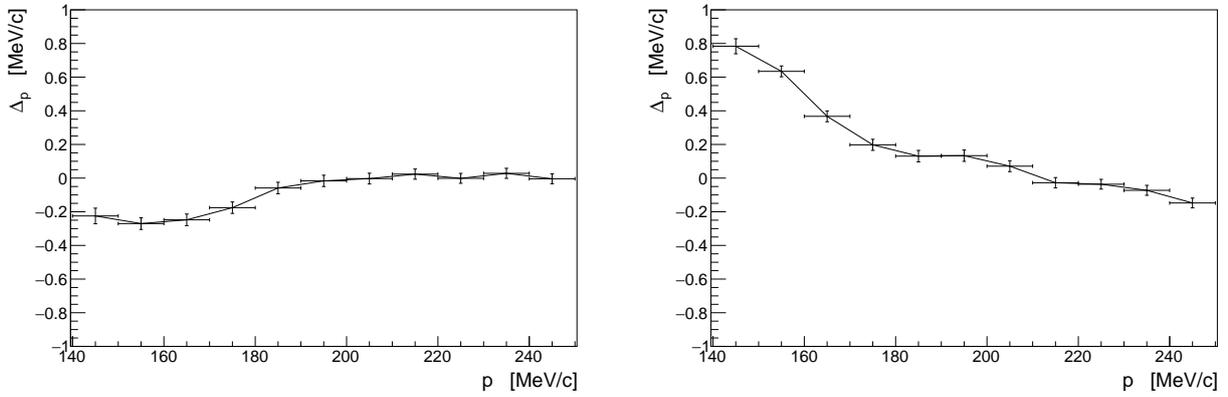


Figure 15: The mean residual between the final track fit total momentum and the true track momentum, evaluated at the reference plane.

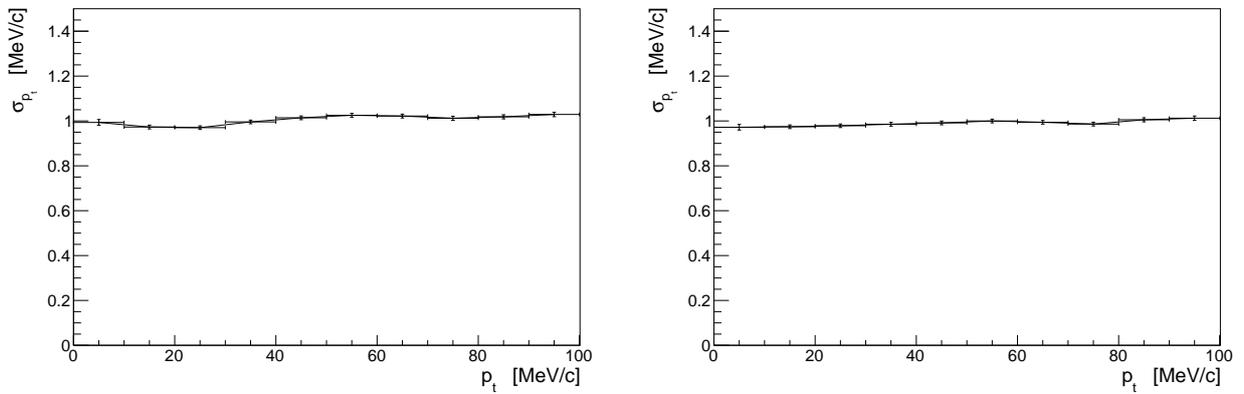


Figure 16: The  $p_t$  resolution as a function of the  $p_t$  of the upstream (left) and downstream (right) trackers.

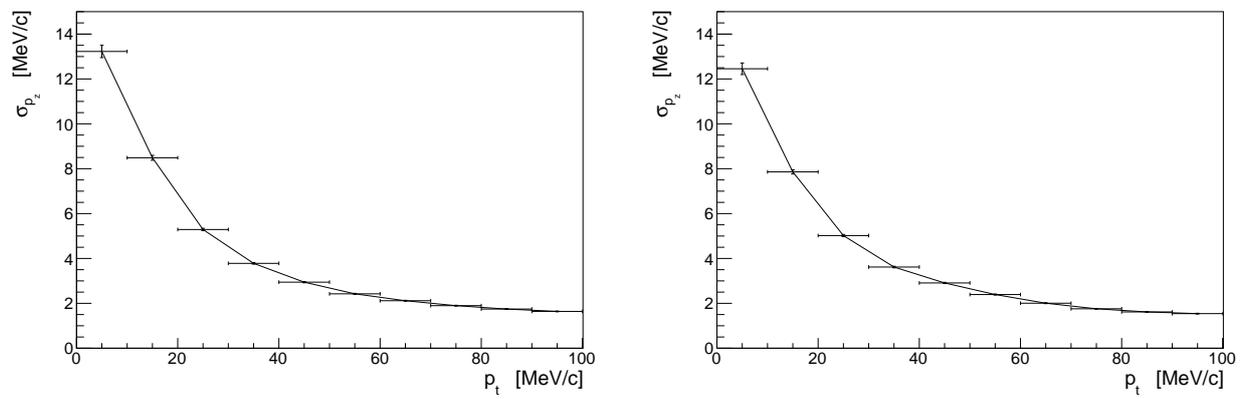


Figure 17: The  $p_z$  resolution vs the  $p_t$  of the upstream (left) and downstream (right) trackers.

## References

- [1] International Scoping Study Physics Working Group, “Physics at a future Neutrino Factory and Superbeam facility,” *Rep. Prog. Phys.* **72** (2009) 106201. arXiv:hep-ph/0710.4947v2.
- 350 [2] S. Geer, “Muon Colliders and Neutrino Factories,” *Annual Review of Nuclear and Particle Science* **59** (2009) 345 – 367.
- [3] A. Skrinsky and V. Parkhomchuk, “Cooling methods for beams of charged particles,” *Sov. J. Part. Nucl.* **12** (1981) 223 – 247.
- [4] D. Neuffer, “Principles and applications of muon cooling,” *Part. Accel.* **14** (1983) 75.
- 355 [5] C. Booth, P. Hodgson, L. Howlett, R. Nicholson, E. Overton, *et al.*, “The design, construction and performance of the MICE target,” *JINST* **8** (2013) P03006, arXiv:1211.6343 [physics.ins-det].
- [6] M. Bogomilov *et al.*, “The MICE Muon Beam on ISIS and the beam-line instrumentation of the Muon Ionization Cooling Experiment,” *JINST* **7** (2012) P05009, arXiv:1203.4089.
- 360 [7] M. Ellis, P. Hobson, P. Kyberd, J. Nebrensky, A. Bross, *et al.*, “The Design, construction and performance of the MICE scintillating fibre trackers,” *Nucl.Instrum.Meth.* **A659** (2011) 136–153, arXiv:1005.3491 [physics.ins-det].
- [8] D. Rajaram, “MAUS: The MICE Analysis and User Software.”. Currently in production.
- [9] S. A. et al., “Geant4 - a simulation toolkit,” *Nucl. Inst. Meth. A* **506** (2003) 250.
- 365 [10] R. Brun and F. Rademakers, “Root - an object oriented data analysis framework,” *Nucl. Inst. Meth. A* **389** (1997) 81–86. <http://root.cern.ch/>.
- [11] J. Dean and S. Ghemawat, “MapReduce: Simplified data processing on large clusters,” in *Proceedings of OSDI04*. 2004. <http://research.google.com/archive/mapreduce.html>.
- 370 [12] R. Fruhwirth, “Application of kalman filtering to track and vertex fitting,” *Nucl. Instrum. Methods Phys. Res. A* **262** no. HEPHY-PUB-503, (Jun, 1987) 444. 19 p.
- [13] P. Billoir, “Track fitting with multiple scattering: A new method,” *Nucl. Instrum. Methods* **225** no. 2, (1984) 352 – 366.
- [14] G. Goodwin and K. Sang Sin, *Adaptive filtering prediction and control*. Prentice-Hall, 1984.