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 emittance reduction via ionization cooling. Muon ionization cooling will be required for the proposed Neutrino Factory or Muon Collider next generation facilities. The emittance before and after the cooling cell must be measured precisely and this is achieved with two scintillating fibre trackers, one upstream and one downstream of the cooling cell. This paper describes the software reconstruction for the fibre trackers, including: 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 measured.

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 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 symmetry violation, or Muon Collider [2], a potential route to multi-TeV lepton - anti-lepton collisions. Muon beams are generated via pion decay, leading to a large initial emittance, which must be shrunk in order for a reasonable fraction of the beam to fall within the acceptance of the downstream acceleration components. Without cooling much of the muon beam is lost, leading to a severe reduction in useful particle rates.

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 yet to be demonstrated. Ionization cooling reduces emittance by passing a beam through some suitable, low atomic number material, such a hydrogen. This leads to reduction in beam momentum in all directions due to ionization energy losses. After the absorber, momentum is restored in the longitudinal direction only by means of radio frequency cavities. The sequence is repeated leading to an overall reduction in the transverse phase space occupied by the beam.

MICE is based at Rutherford Appleton Laboratory, U.K., using the ISIS synchrotron as a proton driver to generate the muon beam. The MICE beamline is described in detail in [5]. A schematic of the full MICE cooling channel, including ionization energy loss and longitudinal acceleration, is shown in figure 1. MICE has completed the first step of its programme, consisting of the muon beamline with particle identification (PID). The next step in the programme, which introduces the trackers and the first absorber module, producing

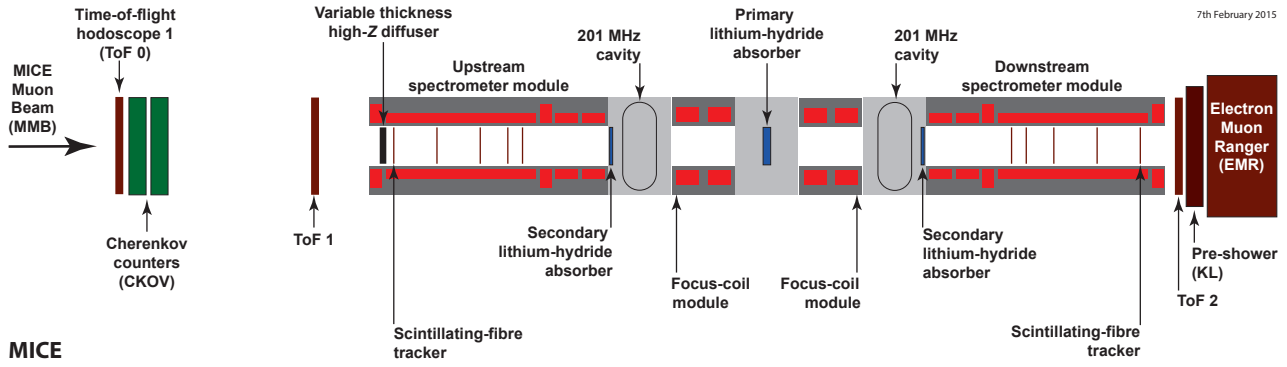

Figure 1: The downstream beam line and final cooling channel. The diffuser is used to increase the beam emittance prior to cooling. The primary absorber can be swapped between liquid hydrogen and solid lithium hydride.

transverse emittance reduction, began taking data in 2015. The demonstration of sustainable cooling, which includes 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 [6]. Each tracker is placed in a superconducting solenoid that provides a uniform field over the tracking volume. One tracker, TKU, is located 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. The upstream tracker, TKU, is orientated such that Station 5 sees the beam first, while the downstream tracker, TKD, is rotated by 180° such that Station 1 sees the beam first, thus in both trackers Station 1 is always nearest to the cooling channel (see figure 1).

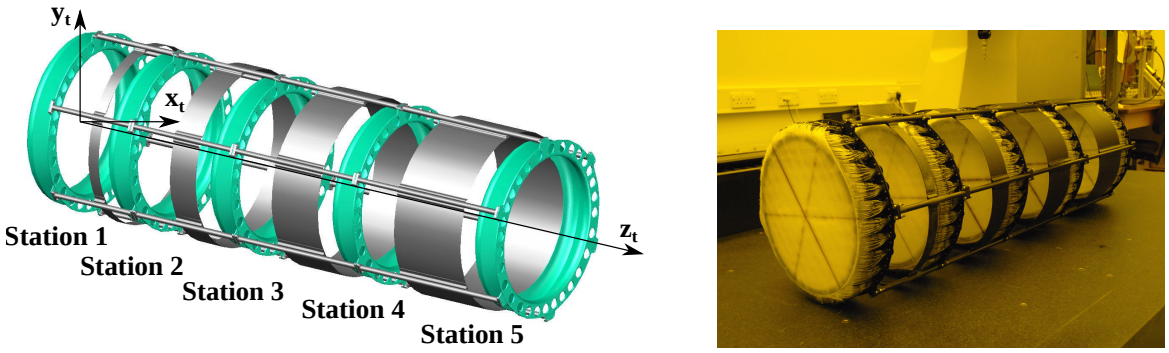

Figure 2: Left: A schematic of the tracker carbon fibre frame, showing the detector station positions. The fibre planes are glued on to the upstream edge (smaller z_t position) of the carbon fibre station frames (shown in green). Right: A photograph of a tracker. The orange tint is due to the special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

Each station is formed of three planes of 350 μm scintillating fibres, orientated at 120 degrees to each other and attached to a sheet of mylar. The fibres in each plane are arranged in a doublet-layer structure in order to give 100% coverage of the plane area as illustrated in figure 3. A mylar sheet glued to the doublet layer provides mechanical stability. 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 analgoue-to-digital converters (ADCs). See [6] for details.

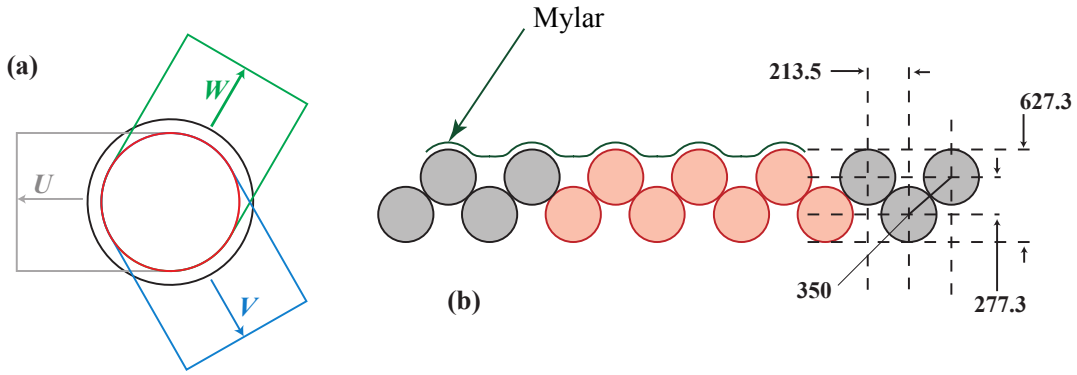

Figure 3: (a) Arrangement of the doublet layers in the scintillating fibre stations. The outer circle shows the solenoid bore while the inner circle shows the limit of the active area of the tracker. The arrows indicate the direction that the individual 350 μm fibres run. (b) Detail of the arrangement 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 μm. The pattern of seven fibres ganged for readout as a single channel, via a single clear-fibre light-guide, is shown in red. The sheet of mylar plastic glued to the doublet layer is indicated.

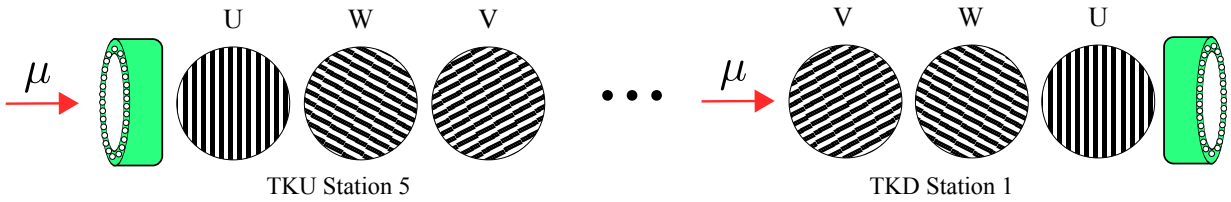

Figure 4: The orientation of the fibres in each plane, as seen by the incoming beam, for both trackers. The green object is the station frame. If TKU was rotated by 180° around the centre of the cooling channel, it would be coincident with TKD.

2 Coordinate systems and reference surfaces

2.1 Channels and digits

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

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. α 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, β, completing a right-handed coordinate system. The origin of the (α, β) 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, α 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

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.

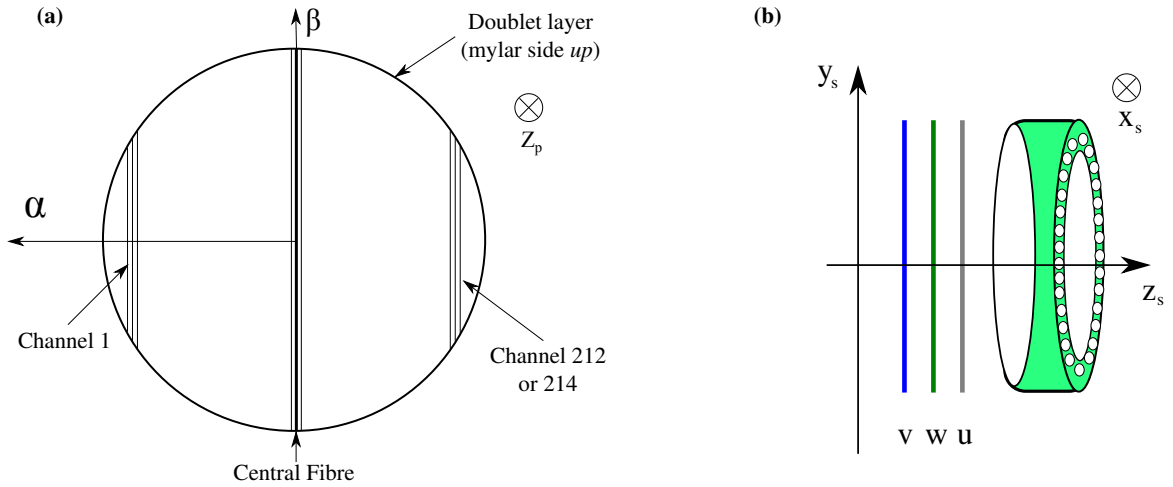

Figure 5: (a) The channel numbering within a plane, and the (α, β, z_p) plane coordinate system (a right-handed system). (b) The fibre plane ordering with respect to the station body and the station coordinate frame (a right-handed system). In TKU the beam approaches from the right, in TKD from the left.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

3 The MAUS framework

The tracker software is part of the MICE software framework, known as MAUS (MICE Analysis User Software) [7]. 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 [8], with analysis based on ROOT [9]. 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[10]. The object passed between the modules is known as a "spill", representing the data associated with one spill of particles through the MICE beamline (see [5]). 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.

The tracker software consists 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 event plots and run information. The modules contain little code themselves but instead call C++ classes to perform the work.

4 Data Structure

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 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 TrackerDAQ data from the full MICE DAQ is stored in a VLSB object or, if the data originated in a cosmic-ray test DAQ, in a VLSB_C object.

The Monte Carlo event holds data on "scifi hits" produced by tracks passing through the fibre planes and any associated 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 Monte Carlo detector-hit classes are derived (see [7]). 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).

4.2 Tracker reconstruction data structure

The reconstructed data for the tracker is held in the "scifi event" class. This contains C++ standard library vectors of the following container classes that represent the higher level reconstructed tracker data:

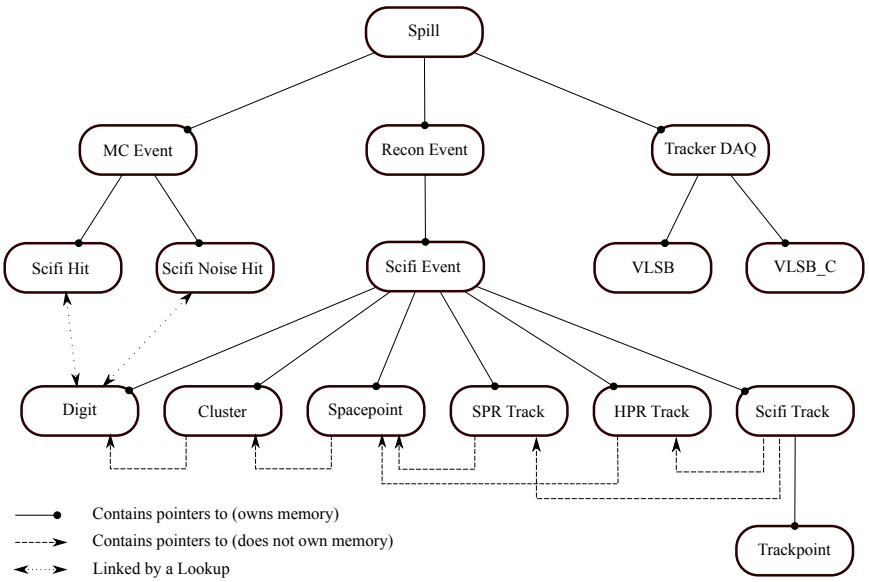

Figure 6: The tracker software data structure, and relevant MAUS data structure. The spill is the top level object below which data is split into real data, MC data and reconstruction branches. When an object owns the memory of a set of other objects, these are held as standard vectors of pointers. When 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.

- "Digits" contain the ADC counts (real or simulated) from the readout of a single channel in response to an incident track;
- "Clusters" are groups of neighbouring digits arising from a particle crossing multiple channels;
- "Spacepoints" group clusters from adjacent detector planes to give a point in space in terms of (x, y, z);
- "Straight pattern recognition tracks (SPR tracks)" group together spacepoints from different tracker stations when the originating track is straight (i.e. when the enclosing field is off);
- "Helical pattern pecognition tracks (HPR tracks)" group together spacepoints from different tracker stations when the originating track is helical (i.e. when the 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. Trackpoints are not stored directly in the scifi event, but instead in the scifi tracks to which they belong.

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.

5 Geometry

The position of each tracker station was determined with respect to the tracker reference surface by use of a coordinate measuring machine (see table 1). The station positions are stored in the MICE configuration database

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.

(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 "MICE modules", at run-time. The MICE modules 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 Monte Carlo and real data, which may be called on by the reconstruction as needed. 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. This is done by means of a simple conversion factor. At this point the NPE values are non-integer values, 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 ADCs. This is weighted channel-by-channel depending on the calibration to give the 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 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 probablility 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.

Once the final NPE value has been calculated it is combined with the channel number and timing information 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. Once the digits stage has been been reached the subsequent reconstruction proceeds identically for both real and simulated data, being encapsulated in one MAUS module (MapCppTrackerRecon). The reconstruction process is illustrated in figure 7. Each stage of the reconstruction is detailed in the sections below.

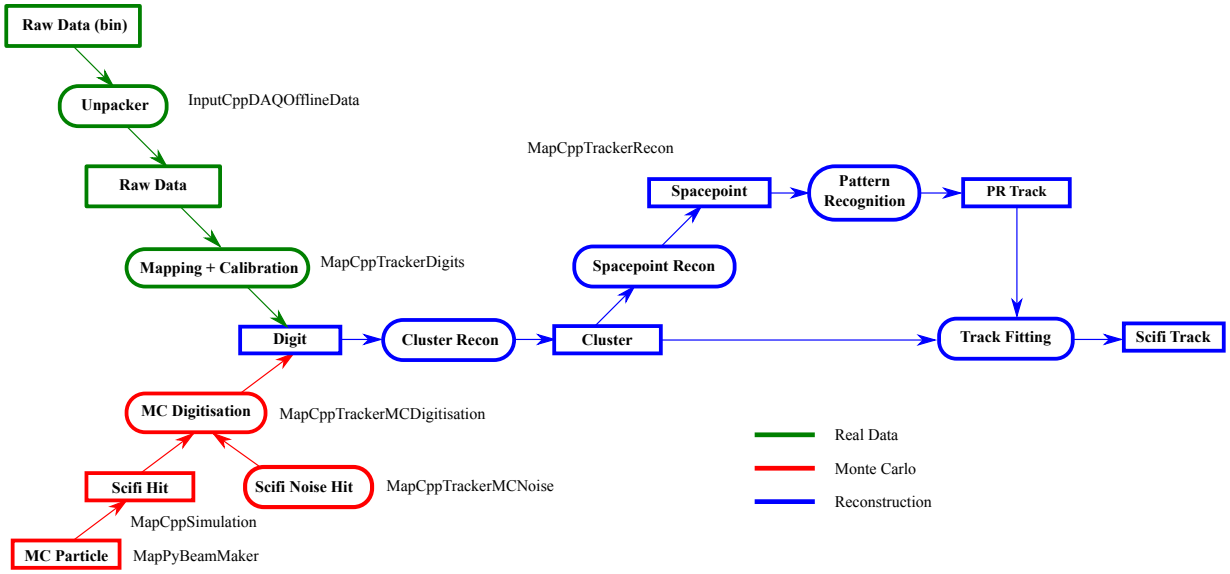

Figure 7: The reconstruction data flow. Data originates either from simulated or real data, the two branches meet after digitisation, after which the reconstruction proceeds identically for both. The relevant MAUS modules for each step are indicated.

7.1 Digitization

For real data the signal from the tracker ADCs is recorded by the DAQ system, including the pulse height for each channel. Channel-by-channel calibration constants are used to convert the ADC value to a signal in terms of 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

A particle that traverses a plane will generate a hit in one or at most two adjacent channels. An isolated hit or hits in two adjacent channels form a cluster. 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 same plane. In the case of multi-digit cluster, the average channel value is used. The cluster position is defined by the coordinates (α, β)

where α and the direction of β have been defined in section 2.2. In the case of two overlapping active channels the value for β is determined by solving:

$$\begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix} = R \begin{pmatrix} \alpha_2 \\ \beta_2 \end{pmatrix} \tag{1}$$

where R is the rotation matrix that corresponds to the orientation of the specified plane.

7.3 Spacepoint Reconstruction

For each station the constituent planes are searched for clusters which 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). From the cluster coordinates, (α, β), the (x, y) coordinates of the spacepoint are determined.

In order to determine which clusters from each plane originate from the same track Kuno's conjecture[6] is followed, which states that 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:

$$|(n^u + n^v + n^w) - (n_0^u + n_0^v + n_0^w)| < K\,. \tag{2}$$

where K is a constant, take by default as 3.0.

Once all triplet spacepoints have been found, doublet spacepoints are created from pairs of remaining clusters.

7.4 Pattern Recognition

Pattern recognition is based on looping over different combinations of spacepoints and performing a fit using a simple 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, ϕ, z), where r is the helix radius, ϕ 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 them paramterised by: the circle centre x_c, y_c and the radius, r, in the transverse plave; 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, ϕ') projection. If the χ^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 ϕ is used to generate s and a straight line fit is performed in the (z, s) plane, and if the χ^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 χ^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 done 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 χ^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 χ^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[11, 12], 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 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 parametrisation 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), in addition to the material effects of the Helium gas within the tracker, to be accounted for between each measurement point.

In order to implement a flexible re-usable Kalman filter, the core algorithm was implemented without any dependencies on phase space dimensions or physical effects. Due to this only the system specific functionality need be provided for a complete implementation. The principal components of the implementation are: propagation routines for both helical and straight tracks; a measurement routine that correctly 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 α (see section 2.2) forms a one-dimensional measurement state. The measurement noise corresponds to the statistical spread of measurements within a single channel in the tracker readout. If the channel is modelled as a top-hat function, the variance of the function is calculated as $w^2/12$, where w corresponds to the width of the channel. Therefore the measurement noise was assumed to be $w/\sqrt{12}$ for all clusters.

The process noise was implemented as a combination of MCS and energy straggling. The energy loss was is calculated using the Landau-Vavilov 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 3), in a fashion almost identical to the GEANT4 implementation:

$$\theta_{RMS} = \frac{13.6\text{MeV/c}}{\beta cp} z \frac{x}{X_0} \left(1 + 0.038 \ln(\frac{x}{X_0} \right) \tag{3}$$

where θ_{RMS} is the RMS scattering angle through a finite length of material, βc, p and z are the velocity,

momentum and charge of the particle in question, and x/X_0 represents the total path length through the material per radiation length. Although the RMS scattering angle does not form a Gaussian distribution, the first order approximation is believed to be sufficient.

8 Performance

A high statistics Monte Carlo simulation was performed in order to estimate 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, without any additional noise or biases.

The fitted transverse positions, (x, y), and momenta, (p_x, p_y, p_z), were compared against the the Monte Carlo truth on an event-by-event basis. The reconstructed events were not subject to any cuts or additional requirements over those of real data reconstruction. The Monte Carlo truth data was determined from the simulated track information, which was stored at every tracker plane to permit a direct comparison to the reconstructed data. All data sets were compared at the tracker reference surface, the designated measurement location with the MICE cooling channel.

An artificial beam was generated with uniform distributions in both the longitudinal and transverse momenta. This was to ensure that the results were not biased by the incoming beam distribution, and that the full reconstructible phase space was probed with equal statistics.

8.1 Track Finding Efficiency

For every simulated event, the number of expected tracks 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 then compared for each tracker and 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 8.

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 9 shows the trackpoint finding efficiency as a function of longitudinal and transverse momenta.

8.2 Reconstruction Resolution

Position residuals are shown in figures 10 and 11, and the momentum residuals in figures 12 and 13. The position reconstruction can be seen to agree with the Monte Carlo truth to high precision in both the upstream and downstream trackers. The momentum residuals currently display some systematic effect which is due to a known issue with the energy loss determination and is currently under study.

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 distributions. As we are concerned with the resolution following a successful pattern recognition stage, these events were removed.

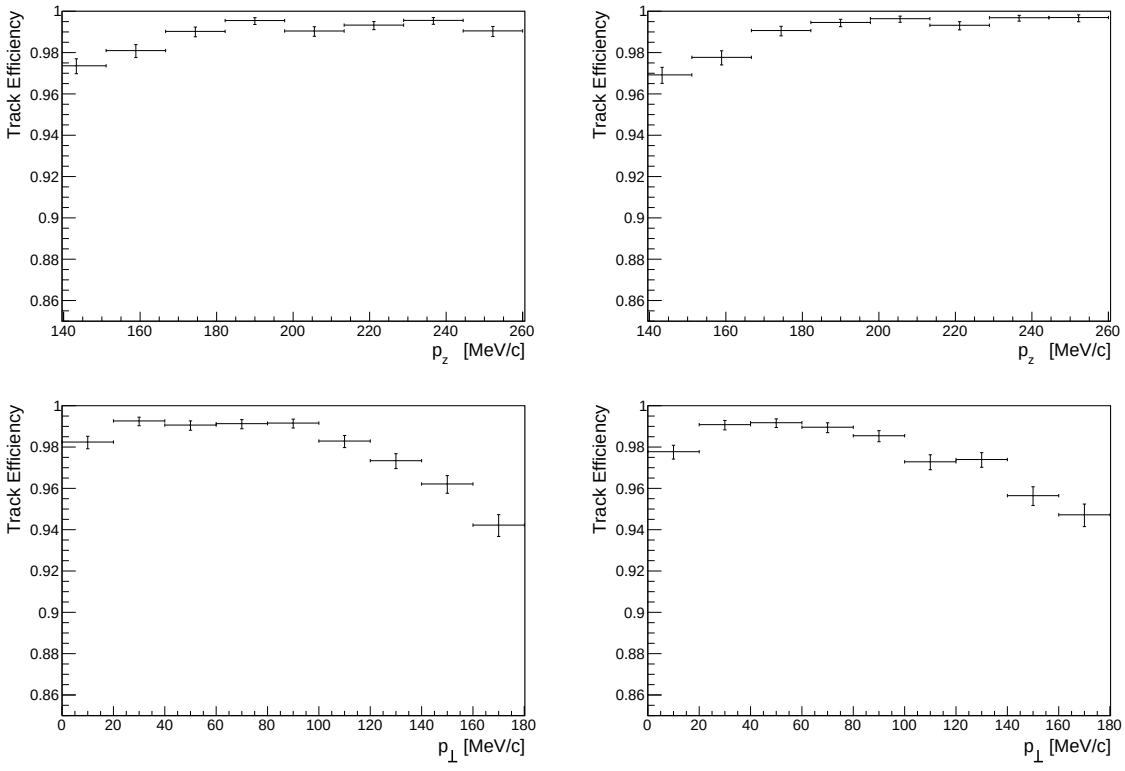

Figure 8: The efficiency of reconstructing tracks in the upstream (left) and downstream (right) trackers as a function of the simulated longitudinal (top) and transverse (bottom) momentum.

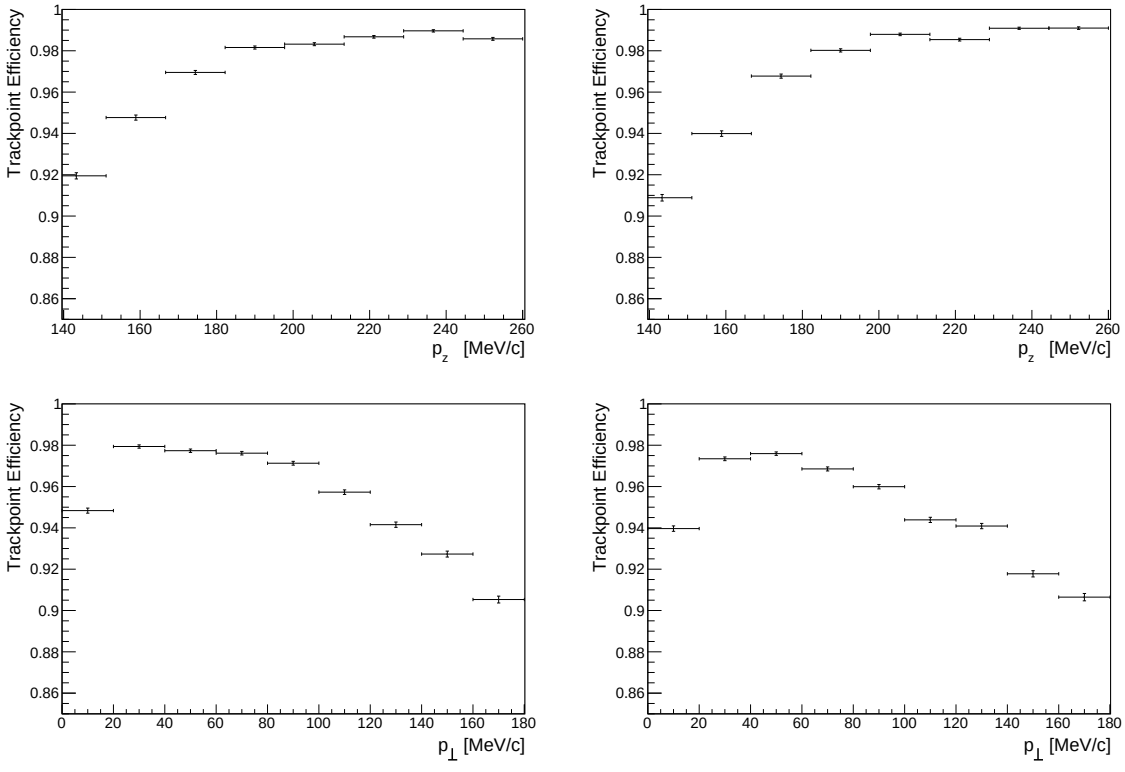

Figure 9: The efficiency of reconstructing trackpoints in the upstream (left) and downstream (right) trackers as a function of the simulated transverse momentum.

The position residuals are consistent with the expected measurement resolutions for a combined fit and the absolute spread is very close to the width of a channel (1.497mm). The transverse momentum resolution is consistent across the range of the sensitive phase space at $\sim$1 MeV/c in both trackers. The longitudinal momentum, an intrinsically more difficult measurement for the tracker, still retains an acceptable spread of ~ 2.7 MeV/c in both trackers. There is however a systematic offset present in the distributions, ~ 0.8 MeV/c in both trackers, and the distributions have more pronounced tails than in the transverse cases. These offsets require further investigation. It is believed that the root cause is a systematic under (over) estimation of the energy loss per plane in the upstream (downstream) tracker, which is realised in the 0.8 MeV/c longitudinal, and 0.2 MeV/c transverse, systematic offsets.

Trends in transverse and longitudinal momentum resolution as a function of transverse momentum are shown in figures 14 and 15. 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

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, $p_\perp$ and p_z. The observed performance in the transverse position and momentum measurements is excellent, for both the upstream and downstream trackers. A systematic offset in p_z requires further investigation, though the resolution remains acceptable.

Future Monte Carlo studies will evaluate the performance of the MICE emittance measurement and how the resolutions of the trackers could bias the measurements.

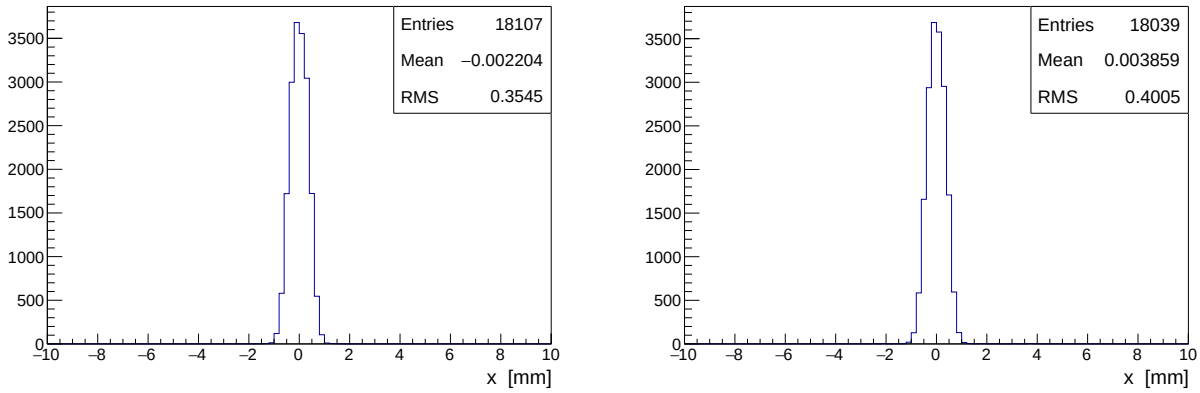

Figure 10: The x residuals of the upstream (left) and downstream (right) trackers for a 6 mm 4D emittance, and 200 MeV/c momentum beam.

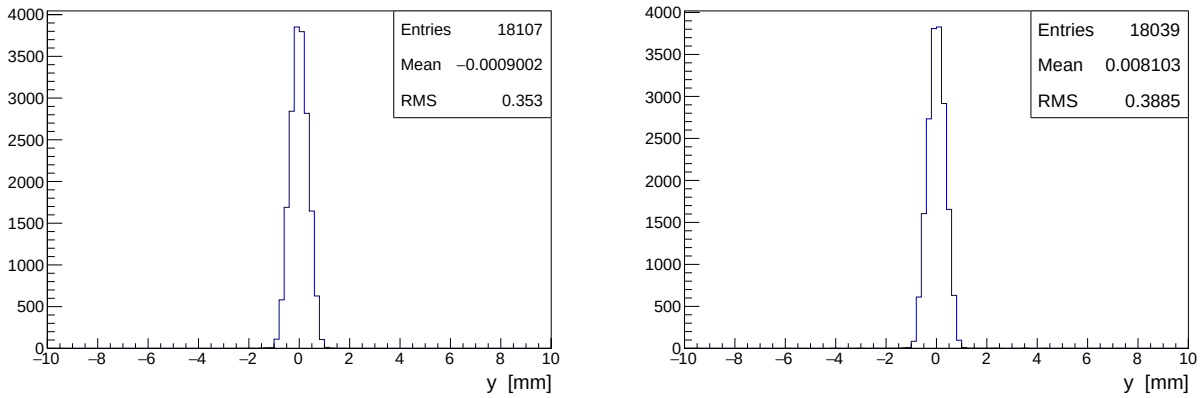

Figure 11: The y residuals of the upstream (left) and downstream (right) trackers for a 6 mm 4D emittance, and 200 MeV/c momentum beam.

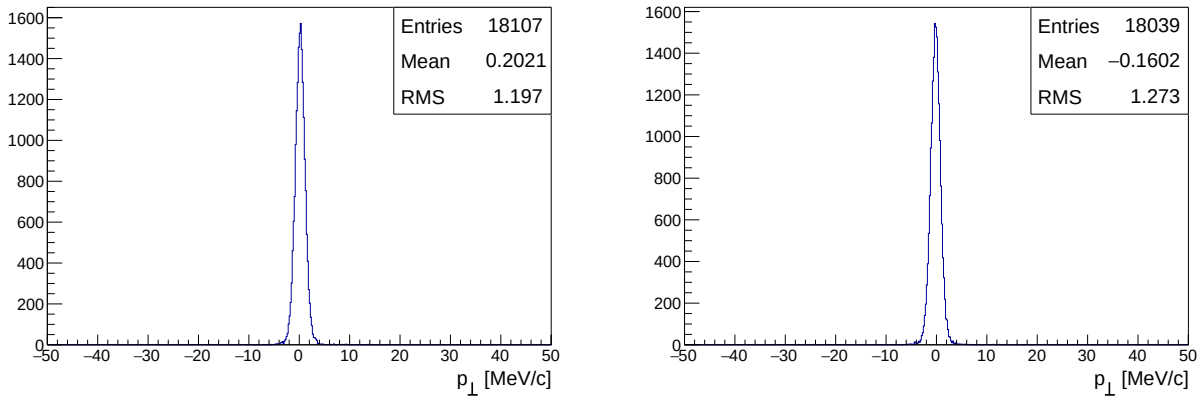

Figure 12: The $p_\perp$ residuals of the upstream (left) and downstream (right).

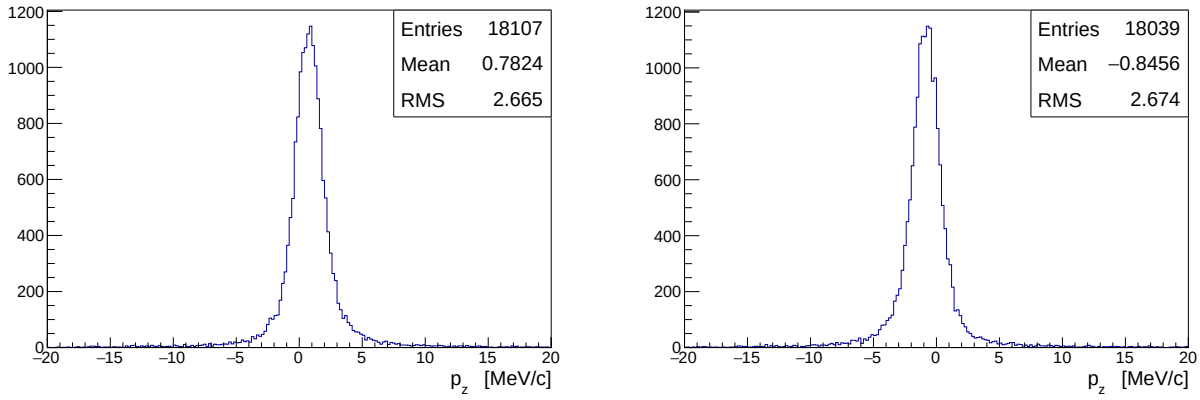

Figure 13: The p_z residuals of the upstream (left) and downstream (right) trackers for a 6 mm 4D emittance, and 200 MeV/c momentum beam.

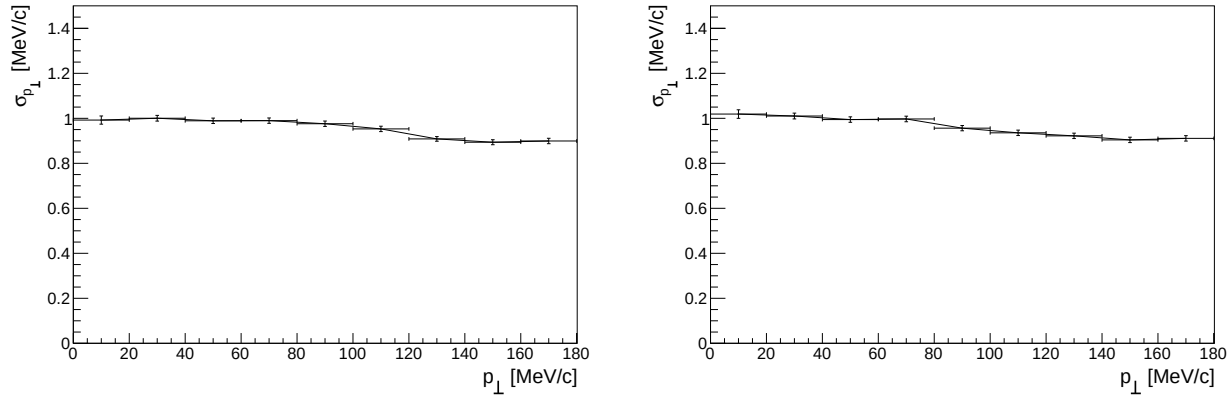

Figure 14: The $p_\perp$ resolution as a function of the $p_\perp$ of the upstream (left) and downstream (right).

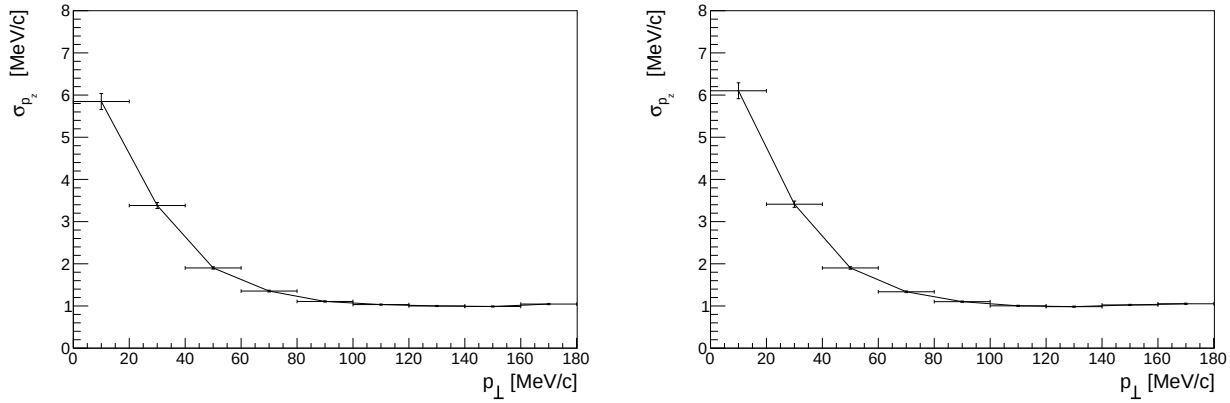

Figure 15: The p_z resolution vs the $p_\perp$ of the upstream (left) and downstream (right).

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.

[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.

[5] 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.

[6] 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].

[7] D. Rajaram, "MAUS: The MICE Analysis and User Software.". Currently in production.

[8] S. A. et al., "Geant4 - a simulation toolkit," *Nucl. Inst. Meth. A* **506** (2003) 250.

[9] R. Brun and F. Rademakers, "Root - an object oriented data analysis framework," *Nucl. Inst. Meth. A* **389** (1997) 81–86. http://root.cern.ch/.

[10] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in *Proceedings of OSDI04*. 2004. http://research.google.com/archive/mapreduce.html.

[11] 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.

[12] P. Billoir, "Track fitting with multiple scattering: A new method," *Nucl. Instrum. Methods* **225** no. 2, (1984) 352 – 366.