

Data-driven Alignment of the Tracker for the Mu2e experiment

A dissertation submitted to the University of Manchester for the degree of
Master of Science by Research
in the Faculty of Science and Engineering

2020

Ryunosuke H. O'Neil

Department of Physics & Astronomy

School of Natural Sciences

The University of Manchester



The University of Manchester

Contents

1	Introduction	14
2	Charged Lepton Flavour Violation	18
2.1	Theory overview	19
2.2	Direct searches	24
2.2.1	$\mu \longrightarrow e\gamma$	25
2.2.2	$\mu^\pm \longrightarrow e^\pm e^+ e^-$	25
2.2.3	$\mu N \longrightarrow eN$	26
3	The Mu2e Experiment	29
3.1	Overview	29
3.2	Accelerator	30
3.3	Solenoids	33
3.3.1	Production Solenoid	33
3.3.2	Transport Solenoid	34
3.3.3	Detector Solenoid	34
3.4	Detector Systems	35
3.4.1	Stopping Target Monitor	35
3.4.2	Calorimeter	35
3.4.3	Cosmic Ray Veto	37
3.4.4	Tracker	38
3.5	Coordinate systems and units	41
4	Software	43
4.1	Overview	43
4.2	Validation and continuous integration	46

4.3	Alignment Conditions	51
4.4	Tracking Algorithms	51
4.4.1	Time-based no-field cosmic track fit	52
4.4.2	Kalman-filter track fit	54
4.5	STM Test Beam software	55
5	Alignment	57
5.1	Detector Alignment	57
5.2	Misalignment impact on tracking	62
5.3	Millepede for detector alignment	63
5.3.1	Local fits	65
5.3.2	Global fit	66
5.3.3	Generation of input data	68
5.4	Tracker degrees of freedom	69
5.5	Weak modes and constraints	72
5.5.1	Constraint equations and parameter fixing	74
6	Alignment Studies	75
6.1	Sel. criteria and dataset creation	75
6.2	Plane translation only	77
6.3	Plane translation and rotation	78
6.4	Alignment with Kalman-filter tracks	82
7	Conclusion	84
	References	87
A	Documentation	93
A.1	How to use the TrackerAlignment package	93
A.1.1	Setting up the software	93
A.1.2	Configuring an alignment	93
A.1.3	Acquiring tracks and running an alignment fit	94
A.1.4	Grid-based workflow	94

List of Figures

2.1	Neutrinoless double beta decay diagram	23
2.2	A SUSY BSM charged lepton flavour violating process. Reproduced from [20].	24
2.3	A SM $\mu \rightarrow eee$ process, occurring via neutral flavour oscillation in a loop. This process is heavily suppressed, with the expected rate around $O(10^{-50})$	26
2.4	A depiction of Muon-to-Electron conversion. The muon is captured by a nucleus, and ‘converts’ to an electron, followed by the recoiling of the electron off the nucleus. It is currently not clear what mechanism governs this rare process.	27
2.5	A Feynman diagram of free muon decay process, which occurs by the weak interaction.	27
2.6	The Michel electron energy spectrum, and the tail of the muon DIO electron energy (Czarnecki) spectrum for ^{27}Al . Reproduced from [28].	28
3.1	An annotated depiction of the Mu2e Experiment, reproduced from [28].	30
3.2	A top-down view of the Accelerator Complex at Fermilab. The proton beam is fed to the Mu2e building from the Delivery Ring via the M4 beam-line. Reproduced from [30].	32
3.3	Timeline after proton pulse. Reproduced from [31].	32
3.4	The Mu2e Solenoid system. Reproduced from [32].	33
3.5	A model of the Calorimeter, situated just after the Tracker. Reproduced from [33].	36
3.6	The Cosmic Ray Veto (CRV) system. Reproduced from [34].	37
3.7	The fully-assembled Mu2e Straw Tracker, reproduced from [35].	38
3.8	An exploded view of a straw and its layers. Reproduced from [28].	39

3.9	A Tracker panel, holding 2 layers of straws. Reproduced from [28]. . . .	39
3.10	Straw layer arrangement inside a panel. Each straw is 5 mm in diameter, and spaced 1.25 mm apart at the closest ends. There is a 6.25 mm spacing between each sense wire. Units on this diagram are in mm. Reproduced from [28].	40
3.11	Momemtum resolution plot	41
3.12	The Mu2e coordinate system axes, centred on the Transport Solenoid. Reproduced from [38].	42
4.1	A screenshot of a pull request, or proposed change, to the Offline software repository on GitHub. An introductory comment is posted automatically by the FNALbuild bot user.	48
4.2	A screenshot of an Offline software repository maintainer triggering a CI test, by posting a comment.	49
4.3	A screenshot of the reported status of CI test results.	50
4.4	Illustration of the principle of distance of closest approach (DOCA) between two 3-D vectors	53
4.5	A screenshot of the monitoring dashboard in a testing stage.	56
5.1	A very simple example of how detector misalignment can affect tracking.	58
5.2	A flow diagram showing the software processes involved in aligning the detector.	61
5.3	Tracking figures of merit: before and after Tracker misalignment.	63
5.4	Demonstration of weak mode behaviour in Tracker Plane z translation d.o.f's during a global fit.	73
6.1	Plane translation alignment after 3 iterations.	79
6.2	Plane translation and rotational alignment - translation shift alignment constants.	80
6.3	Alignment constants (Plane rotation angles α, β, γ) for each plane, for 3 alignment iterations.	81

List of Tables

5.1	A table of solution methods supported by Millepede-II involving the global matrix C	68
5.2	Millepede: required inputs per track measurement.	69

Word Count

This dissertation is 18,999 words in length, not including references.

Abstract

An alignment of the Mu2e Tracker is investigated using the Mu2e Offline Software and Millepede-II. Simulated misalignments of Tracker planes were inserted between Monte-Carlo event generation and before reconstruction using randomly generated (x, y, z) -translations and rotations of the rigid-body planes in the detector model. Cosmic ray muons, with no magnetic field present in the Detector Solenoid, will follow a straight-line trajectory. A maximum-likelihood fit is used to reconstruct a track and determine its parameters. Tracks traversing at least four planes, with ten registered straw hits, and a maximum time residual of 20 ns are selected to be used in an alignment sample. The cuts were chosen to discard tracks with low information content, such as those with high levels of multiple scattering or those with low numbers of measurements contributing to a single track fit. Millepede-II produces the resulting alignment constants according to a simultaneous least-squares fit of the alignment parameters and the track parameters for all input tracks. The improved alignment constants are used to repeat the process and recover an increased number of suitable tracks, further improving the estimate of the alignment constants in subsequent iterations.

A performance study of the process was conducted on a misaligned Tracker in the case of translation-only, and translation plus rotational misalignments. In both cases, the correct alignment constants were recovered to within statistical error after three global fit iterations.

Additionally, a continuous integration system was developed for the Mu2e/Offline GitHub repository, introducing a convenient automated testing process for all proposed changes (“pull requests”). The tests developed included a full build of the software, followed by tests of the validation routines, geometry overlap checking, and multi-threading mode. Additionally, code quality checks and low-statistics physics comparison checks are also available.

Declaration

I declare that no portion of the work referred to in the dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Ryunosuke H O'Neil

Copyright Statement

- The author of this dissertation (including any appendices and/or schedules to this dissertation) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.
- Copies of this dissertation, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.
- The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the dissertation, for example graphs and tables (“Reproductions”), which may be described in this dissertation, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.
- Further information on the conditions under which disclosure, publication and commercialisation of this dissertation, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy, in any relevant Dissertation restriction declarations deposited in the University Library, The University Library’s regulations and in The University’s policy on Presentation of Dissertations

Acknowledgements

I'd firstly like to thank my supervisors Mark Williams and Marco Gersabeck for the weekly meetings, helpful feedback, and impromptu conversations throughout the year that kept me on track, and particularly in the last challenging month leading up to the submission of this dissertation!

During the year I worked with many great and helpful members of the Mu2e collaboration in a number of areas, and I thoroughly enjoyed my time working with them. I'd like to thank David Brown for giving me plenty of opportunities to present my work at the weekly Comp-Soft meetings, as well as all the guidance, code reviews, feedback on my work, and answered questions throughout the year (particularly, on the `#cosmicalignment` slack channel!) I also owe many thanks to Rob Kutschke, who was extremely helpful and responsive in many matters involving the Offline software. Additional thanks to Ray Culbertson and Patrick Gartung for their assistance and answered questions while I was setting up the CI system.

I'd also like to thank Richie Bonventre, Sophie Middleton, and Roberto Soleti who were great to work with and all provided many helpful answers throughout the year, particularly when I was getting started and still trying to figure everything out.

Many thanks also to Gleb Lukicov (UCL) for the helpful discussions about Millepede, and Mark Lancaster for offering me the STM test beam opportunity.

Finally, I'd like to thank my parents for the continued contact and support, especially throughout the Covid-19 lockdown period during which I was thankfully able to continue working on most of my projects.

Preface

In this work I discuss my involvement in the Mu2e experiment over the past academic year.

At first, I involved myself in the software of the experiment, studying how it was organised, and most of all where I could get involved. The time I spent in this phase familiarising myself with the Offline software was the most valuable, otherwise I'm sure that I would not have been able to complete my research project to the current standard.

After being given access to Mu2e and Fermilab computing resources, I spent a lot of time studying a just-developed tracking algorithm used for reconstructing cosmic ray data, developed by Sophie Middleton. Another postdoc, Richie Bonventre, had been working on a revised version of this track fit which became the tracking algorithm I eventually started using with Millepede. In addition to the performance upgrades, the fact that it operated in the time domain and included a T_0 parameter as part of the track model was critically important.

Before this revised track fit was ready, I had time to investigate other aspects of the software that I needed to use, such as the “Proditions” (Pro, rather than Con-ditions) conditions system, where there were some particularly elusive bugs in the routines handling the straw model alignment using stored database values.

Additionally there seemed to be a lack of automated testing, or continuous integration (CI.) In a meeting Ray Culbertson suggested I could get involved with setting up such a system for the Mu2e/Offline GitHub repository. He referred me to a talk by Patrick Gartung, who had proposed a system called CMS-BOT, which he had adapted for use with a Fermilab-hosted Jenkins instance. This was a great starting point. I re-wrote a number of routines within his adaptation, which eventually evolved into a standalone codebase, used to trigger tests on the Mu2e/Offline GitHub repository in a consistent, secure, and helpful way.

Although continuous integration is certainly nothing new, this was a project I believed to be particularly important for the experiment, as an interactive, consistent and systematic approach to testing every single software change can save everyone a lot of time and energy in bug-fixing. Additionally, when developers spend a lot of time on repetitive testing procedures, occasionally corners will be skipped particularly if the procedures are slow. For the past 8 months or so, the testing has been used for every single Pull Request (PR) on the Offline GitHub repository, where appropriate. It can be seen in action at <https://github.com/Mu2e/Offline>. I thoroughly hope Mu2e continue to develop, extend, and utilise this system to their benefit.

After completing the first stage of the CI project, and working with the new track fit for some time and getting comfortable, I felt ready to attempt to interface this track reconstruction method with Millepede-II, a well known Alignment algorithm within the HEP community. Many bugs were squashed in the process, but eventually I'd managed to run the Alignment successfully on a concentrated selection of tracks with useful properties for aligning the Tracker. Thankfully, I'd managed to reach this stage just in time for a Plenary talk given by Dave Brown (LBNL) at the tail-end of the Summer 2020 Mu2e Collaboration Meeting.

To finish, I'd like to mention that all my software contributions to the Mu2e Offline software are publicly viewable on the GitHub repository, under the username @ryuwd.

Chapter 1

Introduction

The Standard Model (SM) of particle physics is one of the most successful and rigorously tested theories in physics today. Its success stems from many correctly-predicted outcomes in particle physics experiments, and yet some areas remain untested, incomplete, or currently inaccessible. In the last few decades, several experimental discoveries (not limited to particle physics, but also cosmology, for example) have highlighted such areas. This leaves one to wonder what the future generations of particle physics experiments will uncover moving forward.

The SM provides a description of the electromagnetic (EM), strong, and weak interactions. Additionally, a description of all known fundamental particles: quarks, leptons, and bosons. Since the conception of the SM, many of its components have been successfully validated. The Higgs mechanism, for example, was conceived to explain the origin of mass in fundamental particles. This was confirmed by the discovery of the Higgs boson at CERN in 2012 with a measured mass of $m_H = 125$ GeV [1], [2]. Whereas other discoveries, such as those of neutrino oscillations at the SNO and Kamiokande experiments have indicated an incompleteness of the SM, particularly that more work must be done to understand and explain the origin of neutrino mass and oscillations [3]–[5].

The SM is not a complete theory of all of the fundamental forces, as it does not incorporate gravitation. In addition, there are areas of the SM where after investigations, more questions have emerged than been answered, leading to a greater need to develop new experiments that may continue investigations. Some pertinent examples of Standard Model incompleteness include matter-antimatter asymmetries in the Universe, dark matter, the accelerating expansion of our Universe, and flavour violation in leptons.

Focusing more specifically on flavour violation, neutrino oscillations have been detected and as a result neutrinos are known to have mass (whereas they were initially considered massless in the Standard Model), but there are still questions that remain here, such as the fact that there is not yet any measurement of their mass. Additionally, neutrino flavour oscillations are proof that lepton flavour violating processes exist in the neutral lepton sector. However, a question still stands on whether it is possible for LFV to occur also in the charged leptons, as the rate at which these processes can occur, or if they can occur is not yet understood.

It is known that advancement of knowledge on LFV in the charged lepton sector (CLFV) can be achieved by focusing on direct searches for three charged lepton flavour violating processes, more precisely the muon processes $\mu \rightarrow e\gamma$, $\mu \rightarrow eee$, and $\mu + N \rightarrow e + N$. The current generation of experiments are in development (with some having already completed the first phase), built with the aim of either making an observation of the process or setting a new upper-bound limit on the corresponding process branching ratios. Such experiments probing these search channels include the MEG, MEG-II, Mu3e, COMET, and Mu2e experiments, which represent an emerging and now growing field of muon physics and muon physics experiments.

There is much value in pursuing CLFV searches due to extremely suppressed and currently experimentally inaccessible rates for these processes predicted by the SM (with non-zero neutrino mass and oscillations included), such that any observation of a process would indicate the presence of Beyond the Standard Model (BSM) physics. Even if no observations are made, improved upper-bound limits on the branching ratio can provide important constraints on the parameter space of many candidate BSM models. In addition to the offered value to SM and BSM physics, due to the challenges posed by these rare process searches and required sensitivity, the design, planning, and construction of these experiments alone will push the limits and assist in the advancement of detector instrumentation and technology.

The $\mu + N \rightarrow e + N$ process is considered the most sensitive to CLFV due to the monotonic energy of the produced electron (or, the conversion electron), far above the endpoint of the Michel spectrum, or the energy spectrum of electrons produced in the free decay of muons. The Mu2e experiment at Fermilab aims to detect this process occurring in muons captured within the aluminium nuclei of a fixed stopping target, with the

conversion electron energy at ~ 105 MeV [6].

In parallel to the installation of the detector systems required to detect the rare conversion electron signature, a software system is being developed to understand the physics processes occurring using information from the detectors. This includes reconstructing the paths that candidate conversion electrons take after emission from the stopping target, which involves the combination of information from a straw drift tube tracking detector (the tracker), a calorimeter system situated downstream from the tracker, and Cosmic Ray Veto (CRV) surrounding the target and detector systems. The tracker contains an array of $\sim 20,000$ straw drift chambers, the positions and orientations of which must be known accurately in a software detector model to ensure the experiment is able to achieve the goal energy resolution of 180 keV/c at full-width half-maximum (FWHM) [7].

Initially, surveys of the tracker can correct initial discrepancies (or, misalignments) between the real and software model positions and orientations of the straws. Remaining misalignments of the straws can be close to the order of millimetres. A well-known solution is the use of track-based detector alignment algorithms, which use large samples of reconstructed tracks to determine corrections to the detector model for the remaining misalignments. The application of one well-known method, Millepede, to the alignment of the Mu2e tracker will be investigated in chapter 5.

The main goal of this research is to provide a starting point for the eventual development of a capable, robust, and versatile software-based alignment procedure for the tracker that is able to meet the challenging requirements of the Mu2e experiment. A simulation-based approach to building a track-based alignment of the tracker using Millepede-II is taken, making use of a maximum-likelihood cosmic track fit and a Monte-Carlo generated sample of no-field cosmic rays. The method is validated by deliberately inserting misalignments into the detector model, followed by attempts to recover them using the alignment system.

This work does not constitute a full solution to the alignment of the tracker, as many more track trajectory types, tracking methods, and alignment degrees of freedom remain to be implemented and supported within the package. Extensions and track-based alignment methods more suited to Kalman filter tracking (the primary track fit method used in Mu2e) are also discussed.

In addition to the content in chapter 5, section 2 of chapter 4 discusses an automated

continuous integration system that was developed for regular use in Mu2e software development and review procedures. The Mu2e software is updated very regularly, with sometimes multiple, separate changes submitted by different developers. While validation utilities and tests are available and implemented, they may not always be used, whether due to lack of time or knowledge of the tests available. The continuous integration platform allows a consistent, wide-variety battery of tests to be run on proposed changes to the software on specially provisioned computing resources provided by Fermilab. Proposed changes are tested before they are approved and merged, thereby greatly reducing the likelihood of errors introduced into the shared main development branch.

Additionally, a standalone software package for a test beam involving the Stopping Target Monitor was developed. The package aims to provide components of a live monitoring system, routines to decode events received from the data acquisition (DAQ) system (more specifically, ‘banks’), and a routine to write events to easily accessible, and analysable dataset formats (see Section 4.5).

Chapter 2

Charged Lepton Flavour Violation

The significant discovery of neutrino flavour oscillations in atmospheric and solar neutrinos presented by the Super-Kamiokande (1998) and Sudbury National Observatory (2001) experiments confirmed the existence of Lepton Flavour Violating (LFV) processes [5] [4] [3]. The discovery provided a resolution to the long-standing ‘solar neutrino problem’ first raised by the results of the Homestake experiment (later shown consistent with the observations of Kamiokande-II), which had long been wrestling with a deficit of electron neutrinos inconsistent with the Standard Solar Model [8].

However, many more questions were raised. The existence of neutrino oscillations is evidence that neutrinos have mass - although exactly how they have mass, whether by being Majorana particles, Dirac particles or otherwise, is still awaiting experimental input. Furthermore, with neutrino oscillations it is evident that lepton number conservation is broken [9].

Additionally, while evidence for LFV in processes involving neutral leptons are established, there have been no observations of LFV in processes involving charged leptons. Charged Lepton Flavour Violating (CLFV) processes are forbidden in the SM assuming massless neutrinos, and strongly suppressed if non-zero neutrino masses and neutrino mixing are added. From Equation (13) of [10] combined with the now-known measurement of the PMNS-matrix mixing angle θ_{13} [11], an estimate of the branching ratio for the CLFV process $\mu \rightarrow e\gamma$ is available,

$$\text{BR}(\mu \rightarrow e\gamma) \approx 10^{-54}, \quad (2.1)$$

which suggests such a process is far too rare to access by experiment. In Beyond the Standard Model (BSM) models the rates can be significantly enhanced, such that the next

generation of experiments may be potentially sensitive to these BSM processes. While there are many models with potential to describe the nature of CLFV, referring to the review literature it is clear that a model-independent, focused experimental approach is desirable in order to elucidate further on the nature of CLFV and rates of production. The observation of a CLFV process could have profound implications for the Standard Model and the physics underpinning neutrino mass [10] [12] [6]. This has motivated a range of experiments which are conducting or planning to conduct direct searches for CLFV processes, and whether they manage to observe such processes or not they could still provide powerful constraints on the parameter space for a large number of BSM models moving forward.

2.1 Theory overview

In the Standard Model, generations of leptons can be expressed as ‘doublets’

$$\begin{pmatrix} \nu_e \\ e^- \end{pmatrix}, \quad \begin{pmatrix} \nu_\mu \\ \mu^- \end{pmatrix}, \quad \begin{pmatrix} \nu_\tau \\ \tau^- \end{pmatrix}, \quad (2.2)$$

where there are 6 leptons: the neutral electron-neutrino (ν_e), muon-neutrino (ν_μ), and tau-neutrino (ν_τ), and the charged electron (e), muon (μ), and tau (τ). Each lepton has a corresponding anti-lepton. The neutral leptons (neutrinos) have zero charge, and may only interact weakly.

A compact, simplified version of the Standard Model Lagrangian density may be written

$$\begin{aligned} \mathcal{L} = & -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + i\bar{\psi}D\psi + \text{h.c.} \\ & + \psi_i y_{ij} \psi_j \phi + \text{h.c.} + |D_m \phi|^2 - V(\phi), \end{aligned} \quad (2.3)$$

where the first term, the scalar product of the field strength tensor, includes the behaviour of the interaction particles: photons, the weak bosons, and gluons, but not the Higgs boson. The second term includes the interaction behaviours between matter (specifically, fermions - described by the field ψ) and interaction particles. The *h.c.* terms represent the Hermitian conjugate of the respective previous terms. The fourth term has particular relevance to neutrinos and their mass, as it describes, using the Yukawa matrix y_{ij} , the Yukawa couplings of matter to the Higgs field (ϕ). The sixth and seventh term describe the

interaction particle couplings to the Higgs field, and the Higgs field potential respectively [13]. Neutrinos were initially assumed massless, meaning that they had no coupling to the Higgs field, however the observation of flavour oscillations clearly suggest that neutrino mass terms are not zero.

Leptons have an individual ‘flavour’ number, dependent on their generation

$$\begin{aligned} L_e &\equiv N(e^-) - N(e^+) + N(\nu_e) - N(\bar{\nu}_e), \\ L_\mu &\equiv N(\mu^-) - N(\mu^+) + N(\nu_\mu) - N(\bar{\nu}_\mu), \\ L_\tau &\equiv N(\tau^-) - N(\tau^+) + N(\nu_\tau) - N(\bar{\nu}_\tau), \end{aligned} \quad (2.4)$$

and the total lepton number is

$$L = N(\ell) - N(\bar{\ell}), \quad (2.5)$$

and in the original Standard Model these are conserved for all interactions [9]. However, for neutrinos the observation of neutrino oscillations have shown that individual lepton flavour number conservation can be violated. Additionally, neutrinos have non-zero, small masses, and the current upper limit from the KATRIN experiment in Karlsruhe, Germany is $m_{\nu_e} < 1.1$ eV (90% CL) [14].

Flavour oscillations were observed in neutrinos arriving at the Super-Kamiokande experiment, produced in the Earth’s atmosphere due to Cosmic Ray collisions with atmospheric nuclei. The Super-Kamiokande collaboration reported a double-ratio

$$R = \frac{(N_{\nu_\mu}/N_{\nu_e})_{\text{data}}}{(N_{\nu_\mu}/N_{\nu_e})_{\text{MC}}} = 0.63 \pm 0.03(\text{stat.}) \pm 0.05(\text{sys.}), \quad (2.6)$$

where the measurement of R is not consistent with 1, which indicates a disagreement between Monte-Carlo experimental predictions and data [5]. The quantities N_{ν_μ} , N_{ν_e} are the number of identified muon-neutrinos and electron-neutrinos in each sample respectively.

It was predicted that two muon-neutrinos were expected to arrive for each electron neutrino arriving in the detector. This hypothesis held for atmospheric neutrinos arriving from directly above the detector. However, for those emerging from beneath the detector after having propagated through the Earth, the observed ratio (corresponding to the numerator of R) was approximately ~ 1.3 . This evidence was a clear indication that muon-neutrinos were oscillating into electron-neutrinos during their long flight-time through the Earth [5], [9].

The presence of flavour oscillations mean that the neutrino flavour and mass eigenstates are not the same. The neutrino flavour states are not eigenstates of the Hamiltonian. A flavour eigenstate is a superposition of mass eigenstates that travel with different velocities, resulting in the mixing of the mass eigenstates becoming time-dependent, leading to distance-dependent probabilities of observing a flavour state. The flavour and mass states are related by the Pontecorvo-Maki-Nakagawa-Sakata (PMNS) matrix, which describes a transformation between the two. This formalism can be used to further understand neutrino oscillation probabilities as a function of distance travelled, and so the probability of neutral LFV occurrence. For three flavours (electron, muon, and tau) and for a transformation of a mass eigenstate (RHS) to a flavour eigenstate (LHS) we have

$$\begin{pmatrix} \nu_e \\ \nu_\mu \\ \nu_\tau \end{pmatrix} = V \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix}, \quad (2.7)$$

where the 3×3 PMNS-matrix is

$$V = \begin{pmatrix} c_{13}c_{12} & c_{13}s_{12} & s_{13}e^{-i\delta} \\ -s_{12}c_{23} - c_{12}s_{23}s_{13}e^{-i\delta} & c_{12}c_{23} - s_{12}s_{23}s_{13}e^{-i\delta} & s_{23}c_{13} \\ s_{13}s_{23} - c_{12}s_{23}s_{13}e^{-i\delta} & -c_{12}s_{23} - s_{12}c_{23}s_{13}e^{-i\delta} & c_{23}c_{13} \end{pmatrix}, \quad (2.8)$$

and $c_{ij} = \cos \theta_{ij}$, $s_{ij} = \sin \theta_{ij}$, δ is the CP-violating phase (the Dirac phase.)

The 3×3 PMNS matrix is parameterised by three mixing angles θ_{12} , θ_{13} , θ_{23} , and the Dirac phase δ . The probability of the oscillation $\nu_\alpha \rightarrow \nu_\beta$ i.e. the oscillation of a neutrino from the flavour α to β given 3 possible flavours is

$$P(\nu_\alpha \rightarrow \nu_\beta) = \sum_{jk} V_{\alpha j}^* V_{\beta j} V_{\alpha k} V_{\beta k}^* \exp\left(\frac{i\delta m_{jk}^2 L}{2E_\nu}\right), \quad (2.9)$$

where $\delta m_{jk}^2 = m_j^2 - m_k^2$, the neutrino eigenmass squared difference, m_i is a neutrino eigenmass, L is the distance travelled, and E_ν is the neutrino total energy. If neutrinos are Majorana particles, there are two Majorana phases in the PMNS matrix. In Equation (2.8), these are taken to be zero as they have no effect on the oscillation probabilities [15].

The 3×3 PMNS matrix parameters have been determined by *Nu-FIT.org* [16] at the 3σ level by combining the latest experimental measurements. As of July 2020, the best-fit

estimate is

$$V = \begin{pmatrix} 0.801 \rightarrow 0.845 & 0.513 \rightarrow 0.579 & 0.143 \rightarrow 0.155 \\ 0.234 \rightarrow 0.500 & 0.471 \rightarrow 0.689 & 0.637 \rightarrow 0.776 \\ 0.271 \rightarrow 0.525 & 0.477 \rightarrow 0.694 & 0.613 \rightarrow 0.756 \end{pmatrix}, \quad (2.10)$$

where the lower and upper bound values for the 3σ range are shown for each element of the matrix.

The discovery of neutrino oscillations also brings into question how neutrinos obtain mass, and whether the neutrinos we know of are Majorana or Dirac particles. Dirac particles, whether left or right-handed, are those that have distinct anti-particles. Majorana particles, are particles that do not have distinct anti-particles.

Under the Standard Model, given massless neutrinos, it is impossible to distinguish between a Dirac or Majorana neutrino, however with massive neutrinos the two are distinguishable by conducting the appropriate experiments. Neutrinos and anti-neutrinos as they are currently known, if found to be Majorana particles, may really be the same particle that may have either left or right-handed chirality states.

Individual lepton number violation does not necessarily mean that the total lepton number can also be violated. If neutrinos are Dirac particles, then the total lepton number L could still be conserved in interactions. If neutrinos are Majorana particles, then the L may be ‘accidentally conserved’ but violated in interactions such as neutrinoless double beta decay ($0\nu\beta\beta$)

$${}^A_Z X \longrightarrow {}^A_{Z+2} Y + e^- + e^-, \quad (2.11)$$

where two neutrons in X have decayed into protons, see Figure 2.1. The neutrinoless final state here is only possible if the neutrino is a Majorana particle, so observing neutrinoless double beta decay would provide evidence for the existence of Majorana neutrinos [17].

With non-zero neutrino masses and neutrino flavour mixing in the SM, CLFV becomes possible through oscillations in loops, although at extremely suppressed rates (e.g. as shown in Equation 2.1) which are dependent on the neutrino masses themselves. The ‘mass-generating’ mechanism that explains the origins of neutrino mass may therefore be relevant, when considering the rates at which CLFV may be observed.

One popular candidate neutrino mass-generating mechanism is the *See-saw mechanism*, a general grand-unifying theory (GUT) model of neutrino masses. By introducing a massless neutrino that is permitted to mix with a massive Majorana neutrino it is pos-

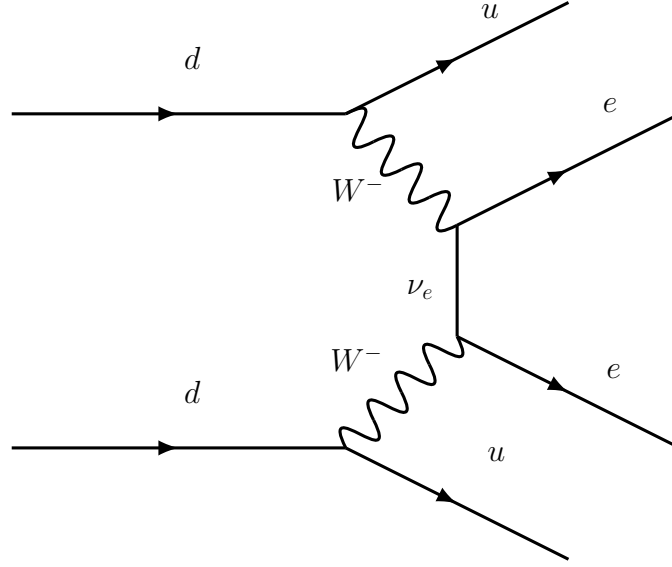


Figure 2.1: The neutrinoless double beta decay process. Two neutrons decay weakly to a neutrinoless final state with two electrons. This hypothesised process is only possible if the electron-neutrino is a Majorana particle, where the electron-neutrino is its own antiparticle.

sible to construct a model that assigns masses for the three known neutrinos (electron, muon, tau) satisfying present experimental limits. The mass matrix associated with this mechanism is

$$\mathbf{M} = \begin{pmatrix} 0 & m_D \\ m_D & m_M \end{pmatrix}, \quad (2.12)$$

where m_D is the Dirac mass (order of electroweak scale) and m_M is the Majorana mass (order of GUT scale). The eigenvalues of \mathbf{M} are

$$|\lambda_+| \approx m_M, \quad |\lambda_-| \approx \frac{m_D^2}{m_M}. \quad (2.13)$$

The name “Seesaw mechanism” is related to the fact that if one mass, say m_M increases, then the other mass m_D must decrease [18].

Lepton flavour violation in charged leptons is also possible in Supersymmetric (SUSY) extensions of the Standard Model such as the Minimal Supersymmetric SM (MSSM). Supersymmetry is a proposed symmetry between bosons and fermions, such that each elementary particle has an associated ‘super-partner’. It was introduced to address issues with grand unified theory (GUT) predictions for the proton lifetime and weak mixing angle. The MSSM contains only the minimum number of new particles needed in addition to the SM for a self-consistent framework [18].

Leptons and quarks in the MSSM have super-partners sleptons ($\tilde{\ell}$) and squarks (\tilde{q})

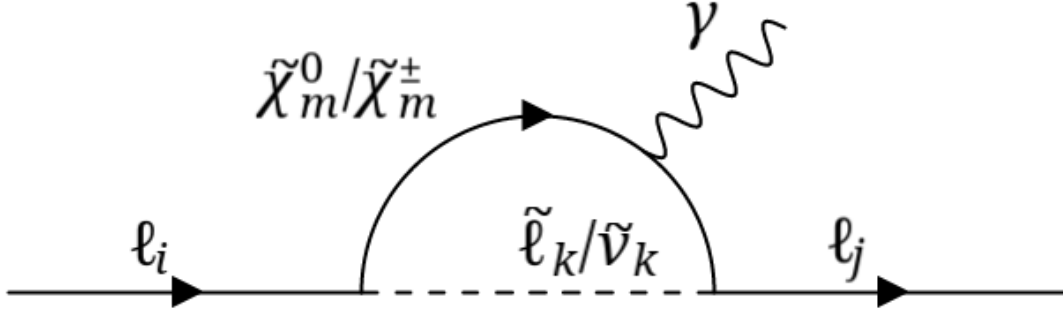


Figure 2.2: A SUSY BSM charged lepton flavour violating process. Reproduced from [20].

Gluons (g) have the “Gluino” (\tilde{G}) super-partner. Additionally there are MSSM “charginos” ($\tilde{\chi}_i^\pm$ with $i = 1, 2$) and neutralinos ($\tilde{\chi}_i^0$ with $i = 1, \dots, 4$) [19]. Diagrams such as the one in Figure 2.2, show a muon to electron conversion process made possible by mixing of these SUSY particles and the radiation of a photon [20].

The rates of LFV processes in SUSY models with a see-saw neutrino mass-generating mechanism can be as large as to be comparable to existing experimental upper bounds (discussed in Section 2.2). The Yukawa couplings resulting from such models can result in large flavour mixing in sleptons [19].

Evidence for neutrino oscillations, and therefore neutral LFV, hint strongly that CLFV is possible. It is not yet known how, or if CLFV is related to neutral LFV or a neutrino mass-generating mechanism such as the Seesaw mechanism. As previously mentioned, this has driven strong motivation to conduct direct, model-independent searches for CLFV processes in High Energy Physics experiments, since if one was to observe an CLFV process or at least narrow down potential candidates via limit-setting, the implications for Standard Model and neutrino physics would be significant. There are a slew of experiments which have or will conduct direct searches for CLFV processes, and these processes are covered in the following section.

2.2 Direct searches

In contrast to general purpose collider experiments, experiments searching for CLFV are usually designed with one particular CLFV process and search channel in mind. Due to the expected rareness of these processes, the process backgrounds encountered in the

channel need to be sufficiently reducible in order to increase the chances of making a significant observation, or where there is no observation, set a new upper bound on the limit of the process.

At the time of writing, there are three notable, active direct search channels with a history of experimental searches behind them, and with modern experiments currently operating, or expected to begin operating in the near future.

2.2.1 $\mu \longrightarrow e\gamma$

Before the discovery of the neutrino by Cowan and Reines in 1956 [21], a search for $\mu \longrightarrow e\gamma$ was conducted by Hincks and Pontecorvo [22], to investigate the possibility of the emission of a photon rather than the now firmly accepted electron neutrino.

Although the existence of neutrinos is now established, the search for this process remains relevant and active due to its charged lepton flavour violating property.

The MEG Experiment at the Paul Scherrer Institute, recently pushed the limit on the branching ratio down to $\sim 10^{-13}$, with [23]

$$\frac{\Gamma(\mu^+ \longrightarrow e^+\gamma)}{\Gamma(\mu^+ \longrightarrow e^+\nu\bar{\nu})} < 4.2 \times 10^{-13} \quad (90\% \text{ CL.}) \quad (2.14)$$

The experiment involves the firing of positively charged muons onto a plastic stopping target. A search is conducted for an electron and photon pair produced back to back, where the combined invariant mass is equal to the muon mass. A scintillating liquid-Xe photon detector with photomultipliers was used to measure the photon timing, position and energy. Positrons were detected using an array of drift chambers situated within a solenoidal magnetic field [24].

MEG-II, an upgrade to MEG, plans to push this limit further down to 4×10^{-14} by realising a rough factor of two increase in detector resolution across all components, in addition to increased muon beam intensity and detector efficiencies [24].

2.2.2 $\mu^\pm \longrightarrow e^\pm e^+ e^-$

Another direct search of interest is the $\mu^\pm \longrightarrow e^\pm e^+ e^-$ process. Such a process, although heavily suppressed, is allowed to occur in the SM via oscillations in loops, and is shown in Figure 2.3. The best limit to date was set by the SINDRUM experiment in 1987 with the

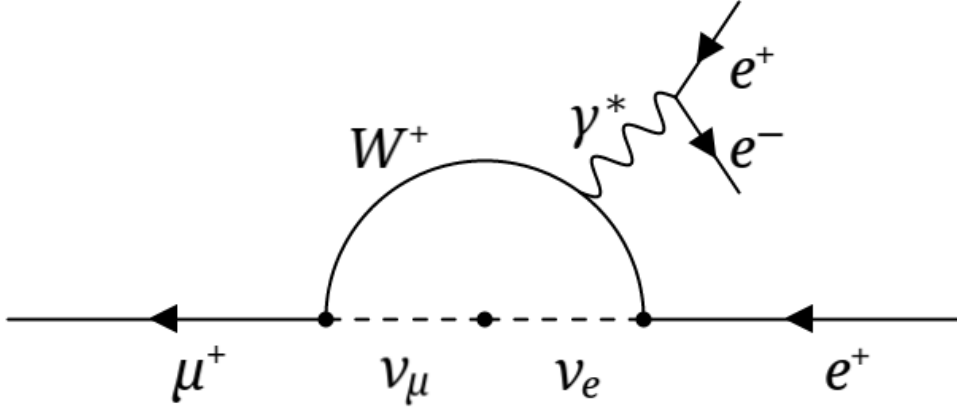


Figure 2.3: A SM $\mu \rightarrow eee$ process, occurring via neutral flavour oscillation in a loop. This process is heavily suppressed, with the expected rate around $O(10^{-50})$.

upper bound set at 1.0×10^{-12} (90% CL) [25].

The future generation of detectors accommodating the next direct searches for $\mu \rightarrow eee$ will be built for the Mu3e experiment at the Paul Scherrer Institute. The Mu3e Experiment aims to detect the positron and electron products of stopped muons in a fixed target, undergoing the $\mu \rightarrow eee$ decay using a thin, silicon strip tracker situated in a solenoidal magnetic field. Phase-I of the experiment will aim for the measurement of or upper-bound limit setting of the branching ratio to the 2×10^{-15} level [26].

2.2.3 $\mu N \rightarrow eN$

Muon-to-Electron conversion is the ‘conversion’ of a muon to an electron, with no other products in the final state. The final state for this process is a single electron, and the conversion of the muon happens while it is captured in a nucleus e.g. in a solid target downstream from a beam of muons (see Figure 2.4.) The single electron final state means that such a process cannot be governed by the weak interaction, and that individual lepton flavour number conservation is certainly violated if the process is allowed. Detecting this process and measuring the ratio $R_{\mu e}$ of conversions to muon captures would inform on the nature of the ‘new physics’ governing this process [6].

The coherent recoil of the nucleus as the muon converts to an electron means the emerging electron has the energy

$$E_e = m_\mu - B - E_{\text{recoil}}, \quad (2.15)$$

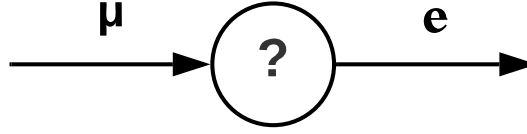


Figure 2.4: A depiction of Muon-to-Electron conversion. The muon is captured by a nucleus, and ‘converts’ to an electron, followed by the recoiling of the electron off the nucleus. It is currently not clear what mechanism governs this rare process.

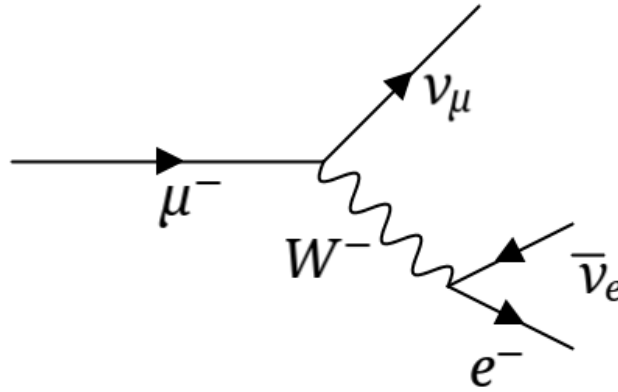


Figure 2.5: A Feynman diagram of free muon decay process, which occurs by the weak interaction.

such that the energy varies as a function of the nuclear mass and Z -number, due to dependence on the binding energy. Therefore, electrons emerging from stopped muons in the presence of e.g. ^{27}Al nuclei (aluminium is the muon stopping target material used for the Mu2e experiment at Fermilab, chosen as an example) will have a single energy, as opposed to a spectrum of energies.

If attempting to select an electron emerging from $\mu - e$ conversion, potential backgrounds may emerge from muon free-decay and muon decay-in-orbit (DIO) processes. Muons freely decay via the weak interaction into a W -boson and muon neutrino. The W -boson decays to an electron and antielectron neutrino (see Figure 2.5.) The electron energy spectrum is called the Michel spectrum, which occupies a range lower than the conversion energy.

If the muon is captured in orbit e.g. in an ^{27}Al target, it decays weakly in what is called

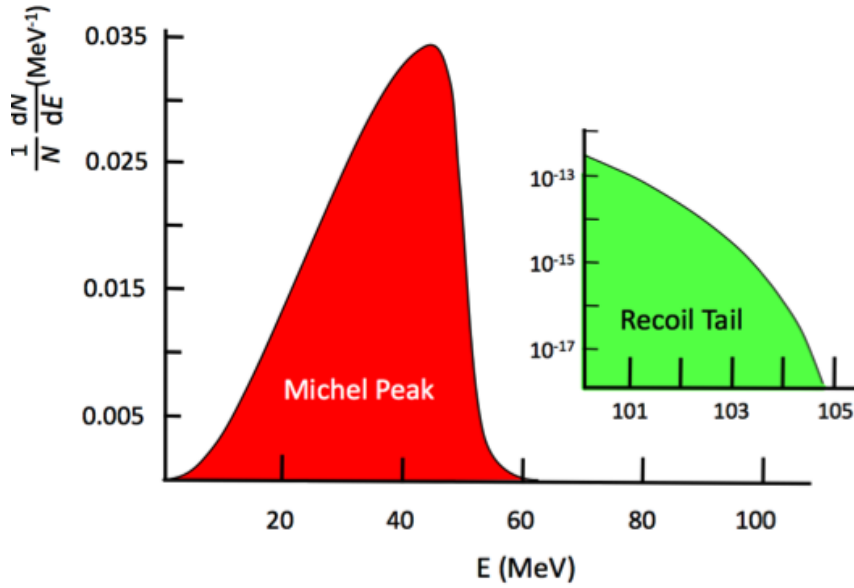


Figure 2.6: The Michel electron energy spectrum, and the tail of the muon DIO electron energy (Czarnecki) spectrum for ^{27}Al . Reproduced from [28].

a muon decay-in-orbit (DIO) process. For a muon DIO, the momentum spectrum of the emerging electrons (the Czarnecki spectrum [27]) are slightly different than for a free decay (the Michel spectrum), see Figure 2.6. The DIO electron energies will be less than that of a conversion electron. However, the mono-energetic conversion electron energy peak lies close to the endpoint of the DIO peak.

The latest limits set by recent searches for Muon-to-Electron conversion come from the SINDRUM-II experiment, based at the Paul Scherrer Institute. The SINDRUM-II experiment started in 1987 and finished data-collection in 2000. Muons were directed at and stopped in a gold target, and the experiment set the upper limit for this process at 7×10^{-13} (90% CL) [29].

The next chapter will discuss the upcoming Mu2e Experiment at Fermilab, near Chicago, U.S., where there are plans to push the upper limit for this process down to 8×10^{-17} (90% CL), an improvement of four orders of magnitude on the existing limit. Another experiment, COMET, based at J-PARC in Japan, will also search for Muon-to-Electron conversion. Phase-II of COMET aims to push the limit down to $\sim 10^{-17}$.

Chapter 3

The Mu2e Experiment

The Mu2e experiment is one of two experiments currently pursuing the search for Muon-to-Electron conversion. In this chapter, an overview of the experiment and its systems is presented.

3.1 Overview

The Mu2e Experiment (see Figure 3.1), based at Fermilab in Chicago, U.S., aims to measure the neutrinoless muon-to-electron conversion ratio

$$R_{\mu e} = \frac{N(\mu^- N \rightarrow e^- N)}{N(\mu^- \text{ captures})}, \quad (3.1)$$

resulting from the hypothesised CLFV-process $\mu N \rightarrow e N$, with N being an Al^{27} nucleus in the Mu2e stopping target. This is achieved by firing muons into a fixed aluminium target, where the muons can be stopped either in orbit or captured in the nucleus. The number of muons captured in the stopping target, $N(\mu^- \text{ captures})$, is counted by the Stopping Target Monitor (STM) [7].

To measure $N(\mu^- N \rightarrow e^- N)$, the number of conversion electrons (electrons emerging from the target as a product of Muon-to-Electron conversion), the product of each decaying bound muon is tracked and identified as they pass through the Tracker and Calorimeter, situated downstream from the stopping target. Electrons matching a neutrinoless muon-to-electron decay signature are counted by selecting for their characteristic monotonic conversion energy at ~ 105 MeV [7]. This distinct process signature makes this search channel potentially one of the most sensitive to CLFV.

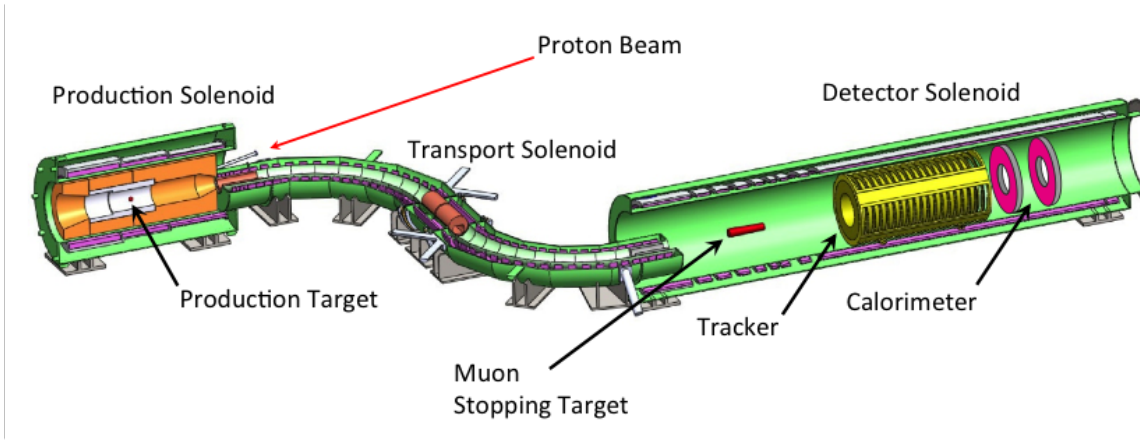


Figure 3.1: An annotated depiction of the Mu2e Experiment, reproduced from [28].

In order to measure $R_{\mu e}$ effectively, accurate measurement of particle momenta as they traverse the Mu2e Tracker is especially important. The signal process, muon-to-electron conversion, has a conversion energy of 104.973 MeV [7] [27], and due to the coherent recoil of the nucleus and the single electron in the final state, any electrons emerging as a result of this process have an energy equal to the conversion energy. However, the most common background emerges from electrons resulting from “Decay-in-orbit” (DIO) processes, and a high momentum resolution is required due to the closeness of the signal peak to the endpoint of the DIO momentum spectrum, or Czarnecki spectrum [27].

Before any muons reach the stopping target, a beam of muons must be produced. A pulsed proton beam, provided by the Fermilab Accelerator complex, is fired at a tungsten production target at 8 GeV kinetic energy [7]. The emerging backwards-going pions, which decay into muons, are directed along the “Transport Solenoid” where they are selected for charge and momentum. Muons emerge into the “Detector Solenoid” where those muons hit the stopping target. A full view of the experiment and its sections are shown in Figure 3.1.

3.2 Accelerator

The Mu2e experiment uses an 8.9 GeV pulsed beam of negatively-charged protons, focused onto a production target, as the first step in producing a muon beam. The protons have a kinetic energy of 8 GeV, the pulse width is ~ 250 ns at FWHM, with a gap of 1695 ns between peaks.

The protons are produced by the Fermilab Booster. Two batches of 4.0×10^{12} pro-

tons each are injected into the Recycler Ring where they undergo a 2.5 MHz bunching RF sequence. In the Recycler Ring, each batch is manipulated into four bunches over 90 milliseconds and transferred to the Delivery Ring. The Delivery Ring holds the proton bunches, where they are then extracted to the Production Solenoid through a new external beamline, the M4 beamline. See Figure 3.2 for a top-down view of the Accelerator complex. The Accelerator facilities are shared with Muon g-2 and Fermilab neutrino experiments [7] [28].

Protons are extracted from the Delivery Ring in pulses; however, owing to unwanted beam that may present between pulses, an ‘extinction’ system is to be installed in order to ensure uniform pulses. Beam extinction is defined as the “ratio between the number of protons striking the production target between beam pulses to the number striking it during the beam pulses.” The experiment requires an extinction below 10^{-10} , and to achieve this the M4 beamline as annotated in Figure 3.2 is fitted with oscillating AC dipoles. Protons travelling through the M4 beamline that are not part of a pulse are directed by these dipoles into collimators, ensuring that protons only arrive during pulses. Further down the beam line, past the production target, an “extinction monitor” is situated to measure the extinction of the beam [28].

The proton beam enters the Production Solenoid at 17° to the axis. Pions, which eventually decay to muons, are produced at the production target. Those travelling in the backwards-going direction (towards the Transport Solenoid), are collected and channelled through the Transport Solenoid to the Detector Solenoid.

After proton pulses are delivered to the tungsten production target, a muon beam is produced and delivered to the aluminium stopping target. The delivery of muons from the production target to the stopping target is handled by a novel “Solenoid” system.

Finally, a pulsed beam is chosen to reduce backgrounds. Many undecayed pions and other backgrounds in addition to the muons arrive at the detectors within a 600 ns window after a proton beam pulse. Un-stopped muons, undecayed pions, and other backgrounds continue down the centre of the Tracker and Calorimeter undetected. Any that do pass through detectors will be excluded as they arrived outside of the selection window. If a continuous proton beam is used, then the time of arrival could not be used to discriminate against background particles. See Figure 3.3 for more about background and signal arrival times between beam pulses.

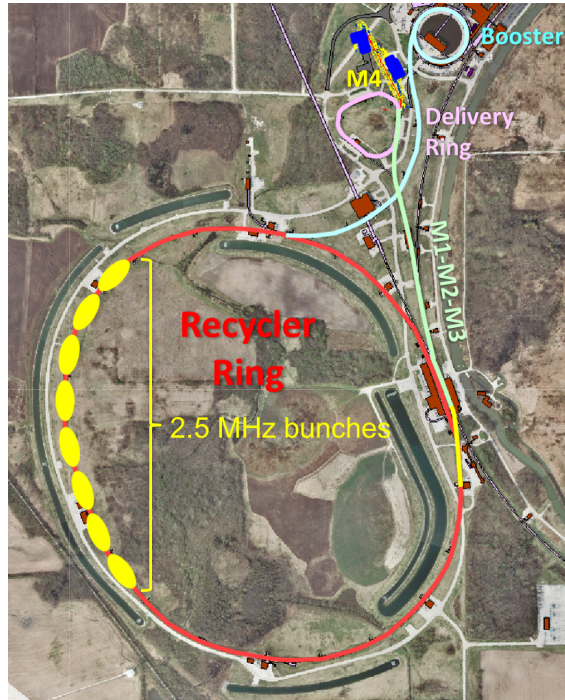


Figure 3.2: A top-down view of the Accelerator Complex at Fermilab. The proton beam is fed to the Mu2e building from the Delivery Ring via the M4 beamline. Reproduced from [30].

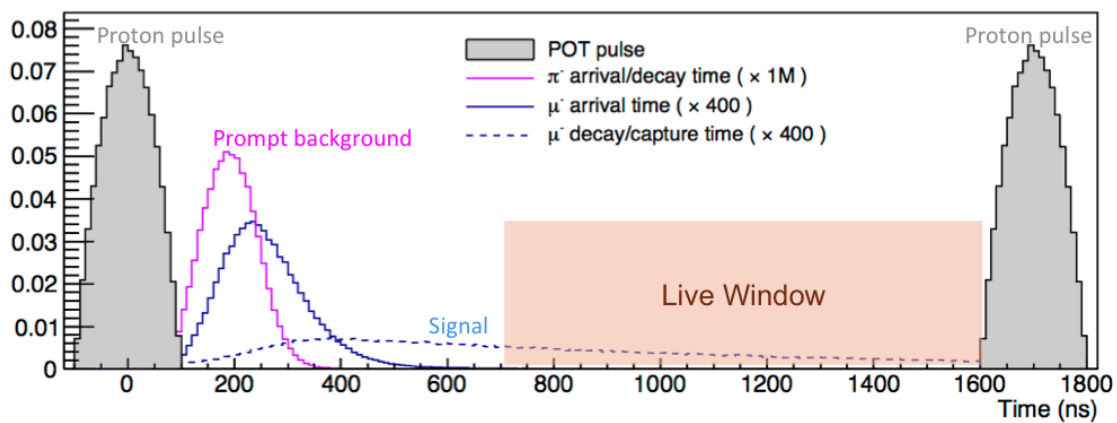


Figure 3.3: Timeline after proton pulse. Reproduced from [31].

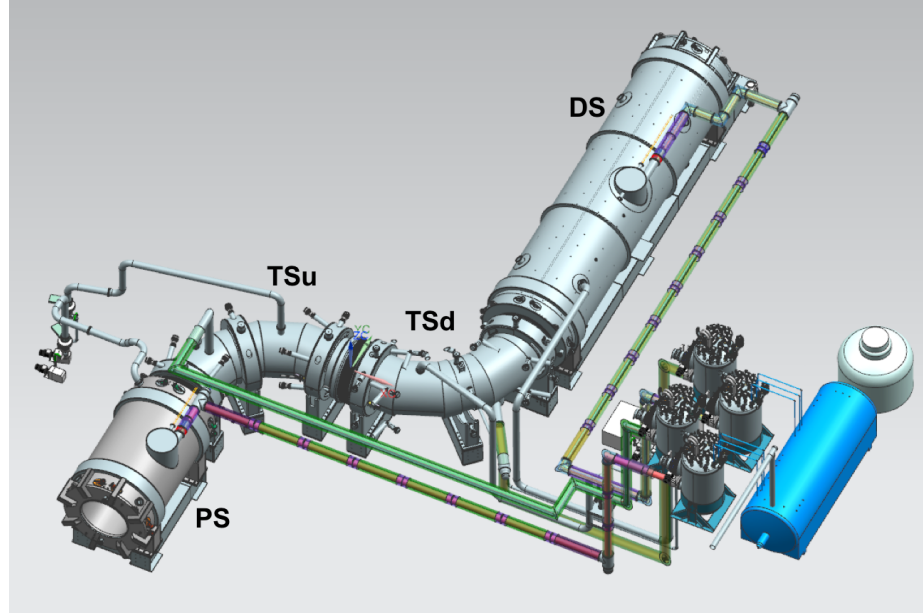


Figure 3.4: The Mu2e Solenoid system. Reproduced from [32].

3.3 Solenoids

To create and manipulate the muon beam, the experiment has a large Solenoid system composed of three main parts: the Production Solenoid, Transport Solenoid, and Detector Solenoid, respectively. A full view of the system can be seen in Figure 3.4.

The Production Solenoid is where the proton beam is received and focused onto the production target. In the Transport Solenoid, muons emerging from the production target are selected for charge and momentum, as they are ‘transported’ to the detectors. The Detector Solenoid is where these charge and momentum selected muons are received onto the stopping target. The products of stopped muon decays emerging from the stopping target are detected, tracked, and identified as they pass through the Tracker and Calorimeter. X-rays indicative of muon captures are counted by the Stopping Target Monitor further downstream.

3.3.1 Production Solenoid

In the Production Solenoid (PS), a pulsed proton beam is received onto a fixed tungsten production target situated within an evacuated 1.5 m diameter bore region. Pions are produced in proton interactions with the fixed target. Backwards-going pions travelling opposite to the proton beam direction are directed towards the Transport Solenoid by a

strong axial magnetic field with axial gradient ~ 1 T/m, with field strength ~ 4.3 T at the target, decreasing towards the Transport Solenoid. The pions eventually decay to muons to produce the required muon beam.

Surrounding the 1.5 m bore region, is the “Cold Mass” where cooled superconducting coils used to provide the magnetic field within the bore region are assembled. These coils are indirectly cooled with liquid helium to below temperatures of 5.10 K, and protected by a large bronze heat and radiation shield to reduce damage to the coils from secondary particles produced at the target, as exposure to these would raise their temperature such that their superconducting property is quickly and undesirably lost [28].

Not all protons will interact at the tungsten target, and not all produced pions will be travelling opposite to the proton beam direction. Left-over protons are directed further downstream to an absorber and the extinction monitor.

3.3.2 Transport Solenoid

The Transport Solenoid (TS) is a large S-shaped pipe transporting muons from the production target to the Detector Solenoid. A mix of positively and negatively charged muons that have decayed from pions move along the pipe, guided by a magnetic channel formed by a series of superconducting solenoids and toroids [28].

Positively and negatively charged particles moving through a toroid drift in opposite directions. Positively charged particles are thus absorbed by a collimator positioned to allow the drifting negative particles to continue through the system while stopping positively-charged particles from continuing. Neutral particles are unaffected by the magnetic field, do not travel along the S-shaped TS, and do not make it to the Detector Solenoid. High momentum particles are also eliminated as they either hit the walls of the TS or do not make it past the central collimator [28].

3.3.3 Detector Solenoid

Muons arriving from the TS incident on the aluminium stopping target are either stopped in orbit, captured in the nucleus, or not stopped at all. The detector systems required to detect, track, and identify stopped-muon decay products are housed inside a large, evacuated bore region, called the Detector Solenoid (DS), with a 1 T constant magnetic field throughout the region of the Tracker and Calorimeter. Those muons that do not stop

proceed to a beam dump.

3.4 Detector Systems

Inside the Detector Solenoid is the Stopping Target, Tracker, Calorimeter, and beam dump. There is a window to allow X-ray photons produced by muon state transitions to proceed downstream to the Stopping Target Monitor.

Cosmic rays must also be accounted for, as these can cause electrons to be produced that may be mistaken for conversion electrons. A large Cosmic Ray Veto system surrounds the entire DS for this purpose.

3.4.1 Stopping Target Monitor

The Stopping Target Monitor (STM) is a High-Purity Germanium (HPGe) detector situated ~ 35 m downstream from the muon stopping target.

The function of the STM is to provide a measurement of the number of muon captures in the stopping target, in order to complete the measurement of $R_{\mu e}$. It will achieve such a measurement by studying the number of 347 keV X-rays emitted from a $2p \rightarrow 1s$ state transition of the muon, and perhaps additionally the number of later-arriving 844 keV X-rays. It is difficult to count individual X-rays at the expected muon stop rate, due to the expected performance constraints of commercially available HPGe detector electronics [7] [28]. Referring to the Technical Design Report, the experiment's requirements on the measurement uncertainty of the stopped muon count is $\sim 10\%$ [28].

The positioning of the detector is chosen in order to reduce radiation damage effects from beam flash, which if too great would take the detector out of service in a matter of hours or days. The increased distance and shielding makes such effects manageable [7]. In addition to appropriate separation from the target, collimators are to be installed to ensure that the STM is only able to see the target, and the STM is to be situated outside of the DS magnetic field [28].

3.4.2 Calorimeter

The Calorimeter is an additional and important component in the accurate identification of conversion electrons (CEs). It is comprised of two large annular disks (see Figure 3.5)

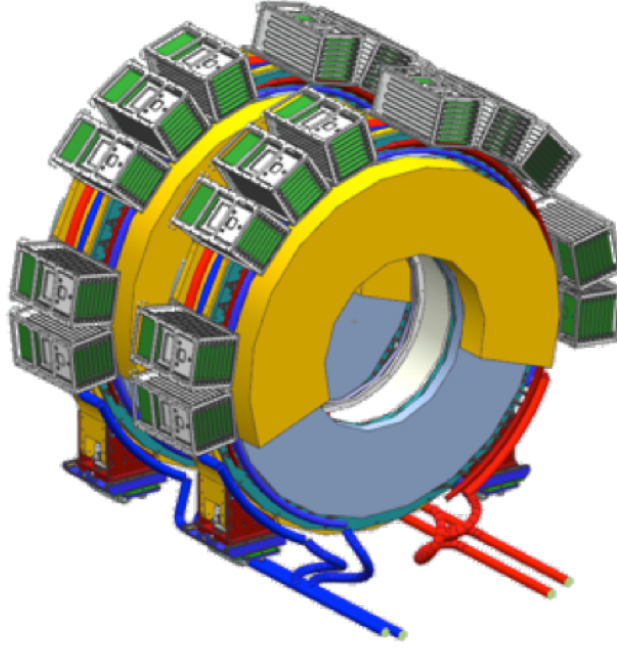


Figure 3.5: A model of the Calorimeter, situated just after the Tracker. Reproduced from [33].

situated just beyond the Tracker, surrounding the muon beamline, such that particles following a trajectory within the acceptance region of the Tracker should also traverse the Calorimeter further downstream. A genuine conversion electron should therefore generate activity in both the Tracker and Calorimeter at points in time and space consistent with the measured trajectory. In addition to position information, the Calorimeter also measures the energy of incoming particles, which may be compared with the momentum measurement extracted by track reconstruction. The additional timing, position, and energy measurements provide a crucial source of discrimination against wrongly identified conversion electron backgrounds. Additionally, the identification of electrons vs muons is important for background rejection in order to achieve the required event sensitivity. The Calorimeter also helps discriminate against cosmic ray muons not detected by the Cosmic Ray Veto [28].

A Calorimeter disk has an array of 674 CsI crystals and two UV-extended silicon photomultipliers. The Calorimeter needs to have an energy resolution of under 10%, timing resolution below 500 ps, and position resolution less than 1 cm [7]. The Calorimeter should additionally provide a trigger for events with significant energy deposits [28].

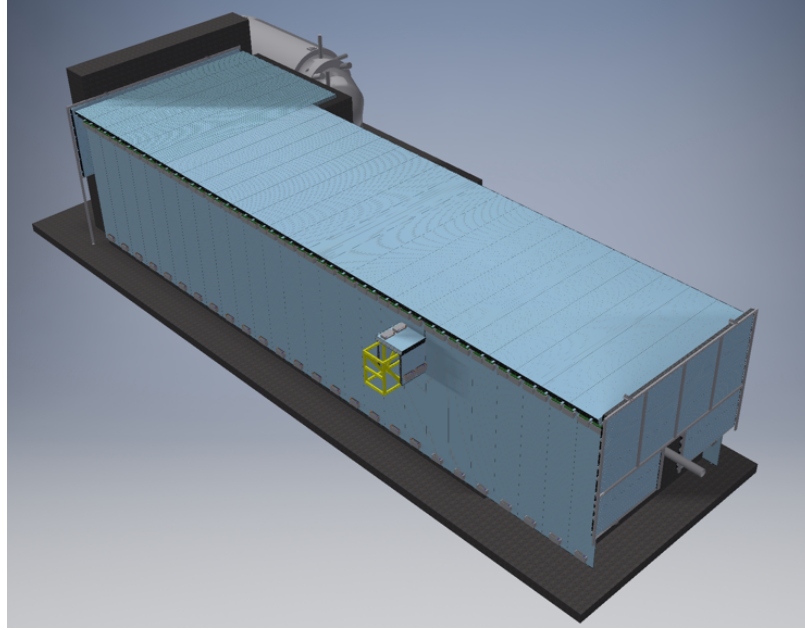


Figure 3.6: The Cosmic Ray Veto (CRV) system. Reproduced from [34].

3.4.3 Cosmic Ray Veto

Cosmic rays are a potentially significant source of background when searching for $\mu - e$ conversion. Not all Cosmic Ray background can be removed by studying whether the trajectory of reconstructed tracks are consistent with coming from the stopping target. It is still possible for a series of interactions, initiated by incident cosmic ray muons interacting with Detector Solenoid material, to produce ~ 105 MeV particles that are identified as electrons originating from the stopping target. In order to discriminate against such events (arriving at the rate of approximately one event per day), the Cosmic Ray Veto (CRV) was developed [28]. See Figure 3.6.

The CRV surrounds the Detector Solenoid and the end of the Transport Solenoid. The structure consists of 4 layers of scintillating strips with aluminium absorbers situated between each layer. Light from the scintillating strips are detected by silicon photomultipliers (SiPMs.) A cosmic muon is detected if at least three strips report detections within five ns of each other. If a conversion electron candidate is detected within 125 ns of the detected muon, the event is excluded [28].

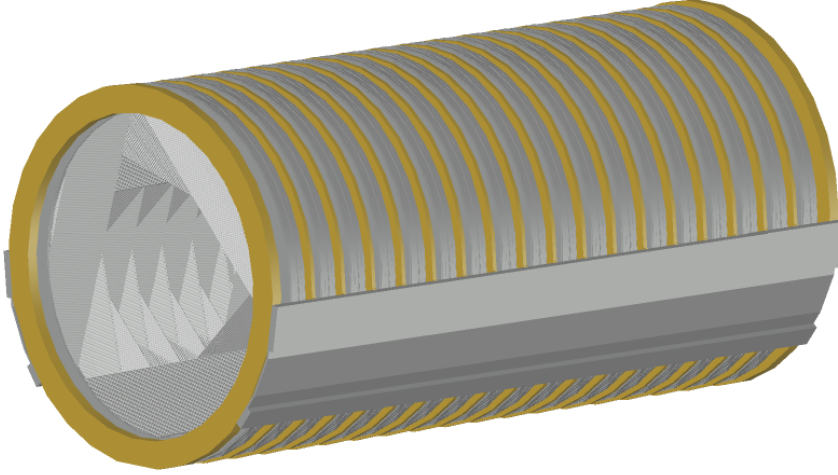


Figure 3.7: The fully-assembled Mu2e Straw Tracker, reproduced from [35].

3.4.4 Tracker

The Tracker is a straw drift tube tracking detector, 3196 mm in length and 1620 mm in diameter [28], and is to be assembled as displayed in Figure 3.7. It is situated between the Stopping Target and Calorimeter in the Detector Solenoid, where its main purpose is to detect and track electrons emerging from the stopping target. During normal operations, the Tracker operates within a 1 Tesla magnetic field present throughout the Detector Solenoid, such that particles follow a helical trajectory with a radius of curvature proportional to the momentum of the particle. The Tracker has an annular design which serves to effectively discard particles with low momenta and reduce overall detector activity.

The Tracker consists of 18 stations, with 2 Planes per station (36 total), 6 Panels per Plane (216 total), and 96 Straws per Panel (20,040 total.) These objects will be referred to as detector elements or components.

A “Straw” is a long cylindrical drift chamber, 5 mm in diameter, and between 430 - 1200 mm in length. There is a gold-plated tungsten sense wire fixed end to end, held at high voltage (the anode.) The inner wall of a straw is coated in aluminium and gold, which acts as the cathode. A straw’s “Drift Gas” is an 80:20 mix of Argon and CO₂, and the straw wall is constructed from two layers of 6.25 μm Mylar material, 15 μm thickness total accounting additionally for the adhesive between the Mylar layers. See Figure 3.8 for an exploded view.

Argon is a mono-atomic chemically inert gas, where atoms may be ionised or become excited. Mixing with CO₂ ensures that any UV photons produced by excited argon atoms

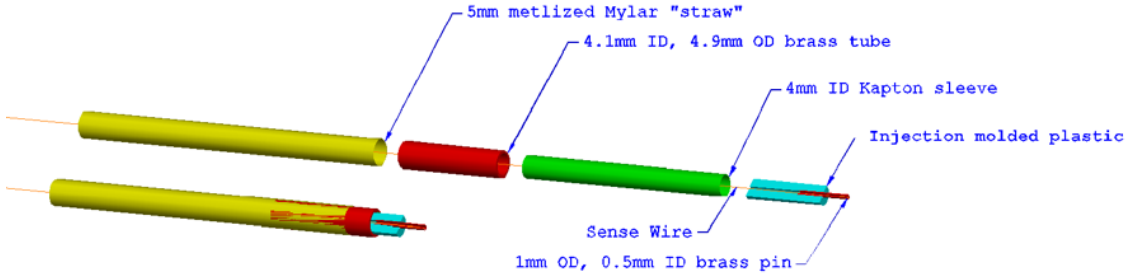


Figure 3.8: An exploded view of a straw and its layers. Reproduced from [28].

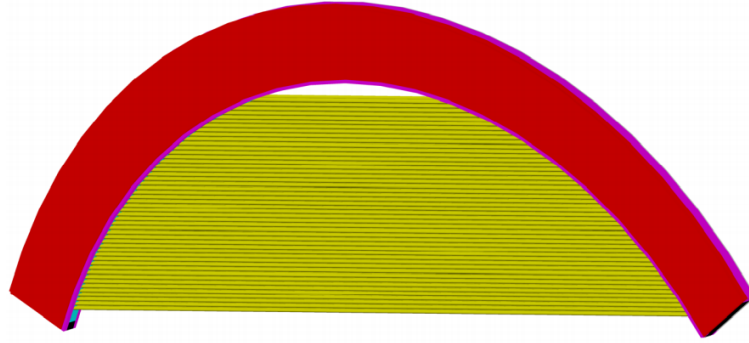


Figure 3.9: A Tracker panel, holding 2 layers of straws. Reproduced from [28].

are absorbed before they reach the cathode. UV photons incident on the cathode would produce photoelectrons, which are capable of producing undesirably large currents in the sense wire.

Straw drift tubes were chosen for their low material density, relative to other detector technologies such as silicon pixel detectors [28]. Particles traversing materially dense detectors are more susceptible to multiple scattering, which can alter their trajectory. Such effects are undesirable since the accurate measurement of particle momenta is an important goal of the experiment. A particle traversing a straw ionises argon atoms in the drift gas, liberating electrons. These electrons drift at a velocity (increasing with closeness to the wire, and also dependent on straw gas conditions) towards the sense wire, where they are collected and a measurable current produced, indicating particle presence. A particularly important measurement is the time of detection, which may be used to determine the point of closest approach of the particle to the sense wire, once combined with other straw measurements. The combination of many individual straw measurements and accurate knowledge of the straw positions allows one to find a best fit of the particle trajectory.

Straws are assembled into layers, with 48 straws per layer, and two layers of straws assembled into a “panel” as shown in Figure 3.9. Straws are staggered as shown in Figure

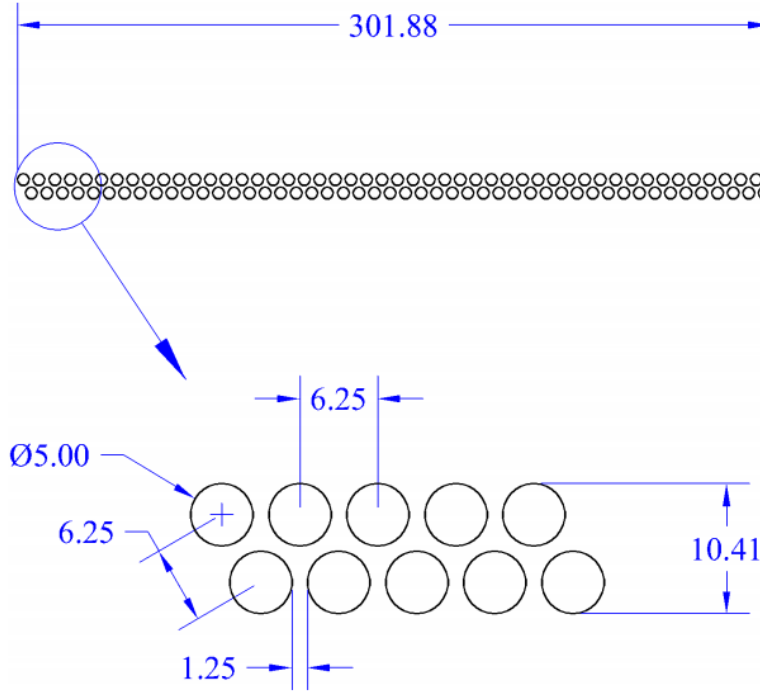


Figure 3.10: Straw layer arrangement inside a panel. Each straw is 5 mm in diameter, and spaced 1.25 mm apart at the closest ends. There is a 6.25 mm spacing between each sense wire. Units on this diagram are in mm. Reproduced from [28].

3.10 to assist in resolving left-right ambiguities, i.e. determining which side of the wire the particle passed. One panel holds a total of 96 straws and sweeps a 120° arc. To mitigate straw sag, straws are held at tension when mounted in the panel. Three panels are combined to form a “face”, an arrangement of panels such that the supports (in red) form a ring. Two faces are layered on top of one another, rotated 30° relative to each other. This arrangement of faces forms one “plane”. Two identically assembled planes are assembled flat-sides together, with one rotated 180° , to make a “station”.

The full Tracker assembly consists of 18 stations mounted in the configuration, as shown in Figure 3.7. One can see the annular geometry, with straws occupying the region $380 < r < 700$ mm with r being the perpendicular distance from the muon beamline. This region is also referred to as the ‘acceptance’ region, where particles with momenta in the region of the conversion energy are expected to pass. The region $r < 380$ mm is the ‘no-mass’ region where remaining muons and the majority of DIO backgrounds are expected to pass undetected. The region $r > 700$ mm are where the supports and readout electronics are located [36].

The Tracker is an important component in ensuring the experiment meets the required

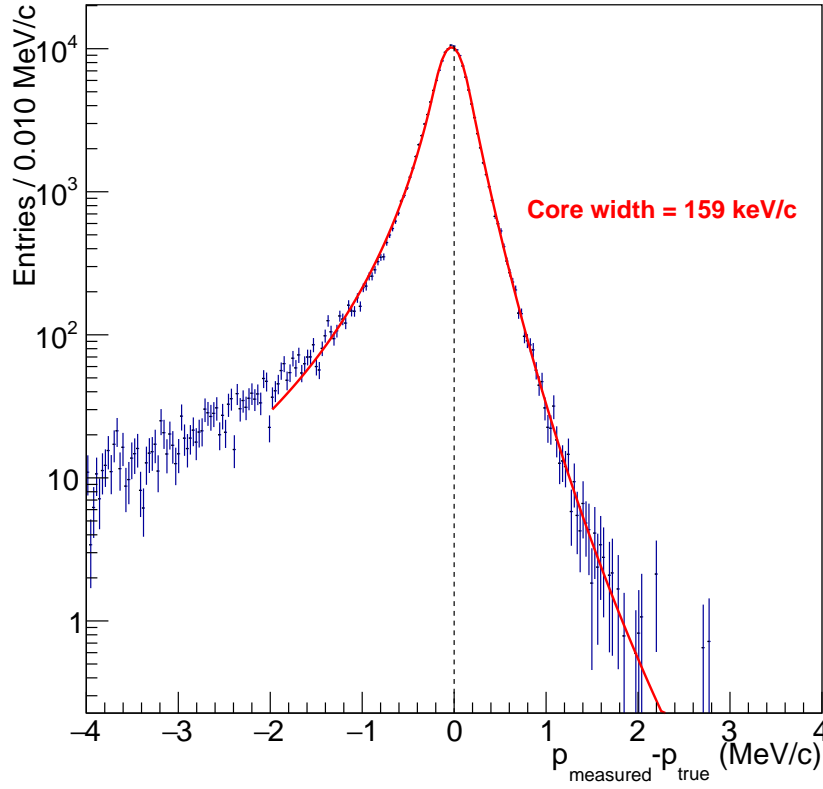


Figure 3.11: A plot of the momentum resolution, which is the difference between the measured momentum and the truth Monte-Carlo momentum (actual momentum at event generation.) The momentum is measured at the point where the particle enters the Tracker. The distribution has a Gaussian shape in the central region. The ‘Core width’ refers to the full-width half maximum of the central Gaussian region. Reproduced from [37].

momentum resolution, in order to avoid contamination from DIO backgrounds. The required momentum resolution is 180 keV/c at full-width half-maximum [7]. See Figure 3.11 for a plot of the momentum resolution from simulation.

3.5 Coordinate systems and units

In the next chapter, there will be a discussion of the experiment’s software component. In the software, there are two coordinate systems available that may be used to represent locations within the experimental geometry. The coordinate systems are the ‘Mu2e’ coordinate system and the ‘detector’ system.

The Mu2e coordinate system origin is situated at the centre of the Transport Solenoid. The z -axis points towards the DS, and is parallel to the muon beamline in the DS, while the

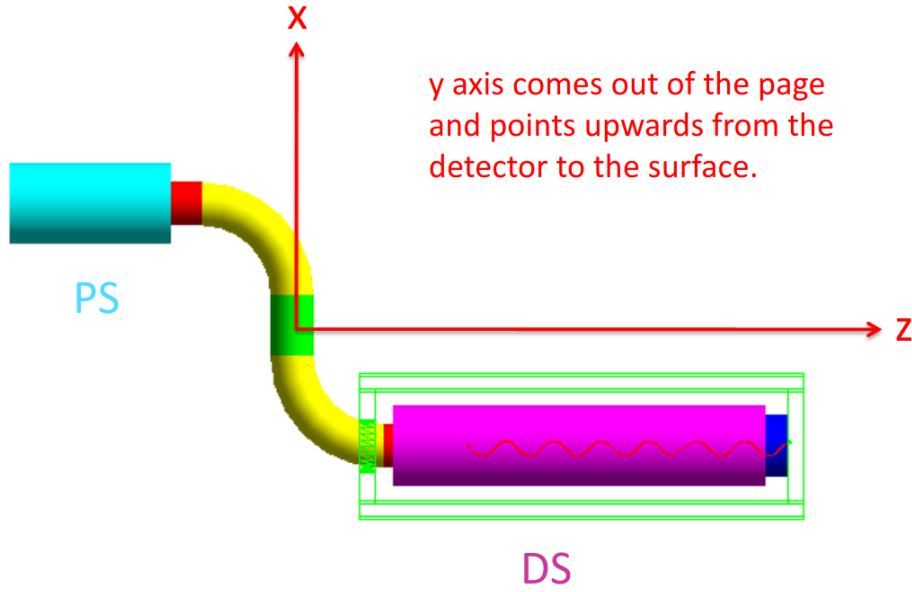


Figure 3.12: The Mu2e coordinate system axes, centred on the Transport Solenoid. Reproduced from [38].

y -axis points up towards the sky. The x -axis runs parallel to the floor and perpendicular to the direction of the muon beamline in the DS. See Figure 3.12 for an annotated diagram showing the coordinate axes. The detector coordinate system is necessary to increase the numerical stability of tracking algorithms by ensuring the individual coordinate values do not have large magnitudes. The axes are parallel to those of the Mu2e coordinate system, but the origin is in the centre of the Tracker [38]. There is also a Geant4 ‘world’ coordinate system used in simulations; however, any coordinates expressed in this system are always transformed to either the Mu2e or detector coordinate systems [39].

Chapter 4

The Mu2e Offline Software

In this chapter, the software of the experiment will be introduced and discussed. The main focus will be on how the track reconstruction algorithms work in the software Tracker environment, accompanied by discussion of the continuous integration and STM test-beam monitoring software projects.

4.1 Overview

The software infrastructure is one of the crucial pillars of the Mu2e experiment. Data are read out from Mu2e detector systems in real time by the data acquisition (DAQ) system, to build ‘events’, where each event contains digitised representations of data recorded from the experiment’s detectors closely in time. Once data are sorted into distinct ‘events’ and stored on disk as ‘data products’ during “Online” processing, these events may be further processed by “Offline” data processing stages. In the Offline phase, data products within each event, e.g. Tracker straw hits, are further processed, combined, and reconstructed into a useful “reconstructed” or “reco” products. These may undergo further processing to create flat n-tuples, stored in a format that may be analysed directly by physicists.

The software must gather unsorted data aggregated across the whole experiment by the DAQ system and recreate what actually happened in the detector. For example, starting with data acquired from the Tracker: when trying to reconstruct a particle’s trajectory, a set of signals on straws in the Tracker may be grouped closely together in time, corresponding to the incidence of one particle. These signals can be associated with a candidate particle that passed through the Tracker, and matched to straws in a software

model of the detector. The positions of the straws that registered a signal from a passing particle in the model can be used to reconstruct its trajectory. Then, the resulting trajectory may be used to derive physical quantities associated with the candidate particle, such as the momentum and particle kinematics at different locations in the Tracker (i.e. as the particle entered, and as it left the Tracker, for example). Such reconstructed data may be used in many different physics analyses, however, each track need only be reconstructed from digitised detector data once. Individually reconstructed tracks may then be combined with other tracks to find common production vertices (for example), matched to energy deposits in the calorimeter, crosschecked with output from the CRV. At each stage, hardware and software limitations will affect the quality of the reconstruction processes involved in the search for conversion electrons. For successful measurements to be made, it will be necessary to develop an understanding of the issues, mitigate or work around them where possible, and also understand the associated systematic uncertainties. These considerations should also be part of the software development process, and the software should be designed to allow the necessary studies and tests to be conducted throughout the experiment.

The DAQ system is comprised of the `otsdaq` and `artdaq` [40][41] software systems. Triggers are implemented using `art` modules within the Online software environment. Event-building, data handling, data transfer is handled by `artdaq`. A interface to these processes is provided by `otsdaq`, which provides an end-user frontend, such as a web interface [42].

The Offline software, written in C++, is constructed within the `art` [41] software framework. `art` is an event processing framework that aims to provide the necessary tools to allow a modular and configurable approach to data analysis for small experiments at Fermilab. In `art`, events are described as the “relevant data describing what happened in a particular time period of interest” [41]. These events are processed by configured ‘jobs’, which are configured to run sequences of ‘modules’ on data ‘products’. Data products are C++ object-oriented representations of data, which may be persisted in datasets, created, read, and written by modules. Modules act on data products, and different module types are available such as the Producer, Analyzer, and Filter. A Producer can create new products and/or new products based on existing products in an event e.g. reconstructing a set of hits to form a track, and storing that track in a data product. An Analyzer has

read-only access to data products, and is often used to analyse and produce histograms or write data to ‘ntuple’ files. `Filter` can be used to discard data products that do not satisfy criteria e.g. applying cuts on reconstructed quantities. Events can contain multiple data products, and modules are executed on events according to execution paths, which are ordered lists of modules.

The Offline software’s purpose is the reconstruction of ‘digis’ or digitised products produced in Online DAQ stages, read out across all detectors. Jobs are configured to produce ‘reco’ or reconstructed products, produced by art modules. These ‘reco’ products can be combined in any number of ways, by other art modules, and processed into flat ntuple ROOT files, which may be read and processed by a number of suitable data analysis tools. N-tuples are datasets holding a number of event records, where for each event a number of data fields are stored (usually of consistent types over all events.)

The Offline software has a validation package, which produces a set of histograms of reconstructed physics quantities, in addition to other parameters such as CPU time and memory use, for comparison purposes. Producing these validation histograms at different versions of the software allows comparisons to be made. By inspecting the differences, one can ascertain whether the code-content changes caused a change in the reconstruction, or Monte Carlo simulations that produce undesirable results. This package is used for the testing routines described in section 4.2.

Track reconstruction of conversion electrons (CEs), at the time of writing, is handled by a Kalman Filter library, `BTrk`, which was used in the BaBar experiment [43]. The interfaces between Offline and `BTrk` are implemented in the `TrkPatRec` and `BTrkData` packages. A new Kalman Filter fitting library, `KinKal`, will soon be introduced to replace the current Kalman Filter implementation.

The reconstruction of cosmic ray tracks, where the DS magnetic field is not present, is handled by the `CosmicReco` package. This package contains the `CosmicTrackFinder` module, which processes `ComboHits` (the Mu2e Offline art data product that represents a single Tracker straw hit, in this case) using a time clustering algorithm that groups hits into individual tracks, followed by a seed fit which performs either a linear least-squares, or brute-force algorithm to the corresponding straw mid-positions to find an initial estimate of the track parameters. This estimate is then used to fit a more accurate track model which further accounts for straw time information.

For the work covered in this dissertation, a new Offline software package was created, called TrackerAlignment, to select and collect tracks that are suitable for an alignment with Millepede-II. A Filter (AlignTrackSelector) and Analyzer (AlignTrackCollector) are created for this particular purpose, where the former performs a selection of suitable no-field CosmicReco tracks and the latter creates the Millepede input dataset in a compressed binary format.

4.2 Validation and continuous integration

The Offline software is developed and maintained in a shared, centralised git repository located on GitHub. GitHub is a git repository management system, with a focus on enabling collaboration between multiple users. git is a version control system, where collections of files are kept in a ‘repository’. Changes to those files are encapsulated in commits, which describe the line-by-line changes needed to bring one version of a file (or files) to the next version (also called a diff.) A ‘branch’ assigns a string of text such as ‘master’ or ‘development’ which points to a specific version of the repository. As it refers to a unique commit identifier (or hash) it is similar to pointers in C/C++ in that a branch always points to a single commit hash. One can backtrack through the chain of commits starting from the latest version to reach the very earliest state of the repository.

Proposed changes and developments to the software are integrated (or ‘merged’) into a main release branch or ‘master’ branch. When a release is ready, a git ‘tag’ is created, which is a string (in this case a version number) associated with a record of the state of the software at the time of creation. The process of integrating changes regularly into development and production software releases is called continuous integration. Another component of this process is an automated testing procedure for all proposed changes ensuring that they do not cause the building (or compiling) of the software to fail, or cause any software components to stop working.

Developers can carry out such tests manually. However, a consistent and automated approach to testing software changes can save developers time, and help navigate developers to the source of issues quickly by providing feedback. Additionally, special computing resources provided by Fermilab are available to experiments for this particular purpose, with adequate memory and processor (CPU) available for memory and CPU-

intensive testing and software compilation. The resources are managed by a system called ‘Jenkins’, allowing users of the system to create ‘jobs’ which may perform certain tests, software compilation, and other actions.

Automated testing is particularly useful for testing proposed changes before they are integrated into the master branch. Since developers of the Offline package commence development of software changes at the latest state or version of the this branch, the software should be functional at all times such that the development is not hindered for other developers. Such testing can be implemented by running tests on a separate, detached copy (or ‘clone’) of the repository, where the proposed changes have been integrated. Therefore, any failing tests are likely to have been caused by the proposed changes. Based on the outcome, the ‘pull request’ which is a request to integrate (or merge) the proposed changes, may be accepted or rejected.

An interface between ‘Jenkins’ managed resources and the GitHub interface, called CMS-BOT, has been developed by, and for the CMS Software (CMS-SW) repository on GitHub. CMS-BOT has been adapted to work with the Fermilab Jenkins instance by the Fermilab Scientific Computing Division (SCD) [44], and this adaptation provided a significant starting point for Mu2e to adopt automated testing procedures.

Due to the specialised nature of the original flavour of CMS-BOT, particularly the tailoring of the features to CMS-SW operations, some of which remained in the Fermilab SCD-adapted version, the system was almost completely re-written and behaviour simplified in order to cater to Mu2e needs. All dependencies (such as Python, and other libraries) were upgraded to the most recent and supported versions. The approval mechanism is removed in favour of the existing GitHub approval and review workflow, and the bot user is disallowed from making changes to the repository. The main focus of this Mu2e customised adaptation is to provide a secure and simple interface to running tests (Jenkins jobs) on Fermilab resources, reporting on results within the context of pull requests. Tests are triggered by comments on pull requests, which are submitted by authorised GitHub users. The comments act as a trigger to start a full build (or compilation) of the Offline software, followed by tests ensuring that Validation jobs are functioning correctly. The outcome of the tests are then submitted by comment to the relevant pull request, indicating success or failure. A decision to integrate the changes, or remedial action, can be taken based on the outcome, to supplement manual code review.

Remove definitions of PACKAGE_SOURCE and BUILD_BASE #251

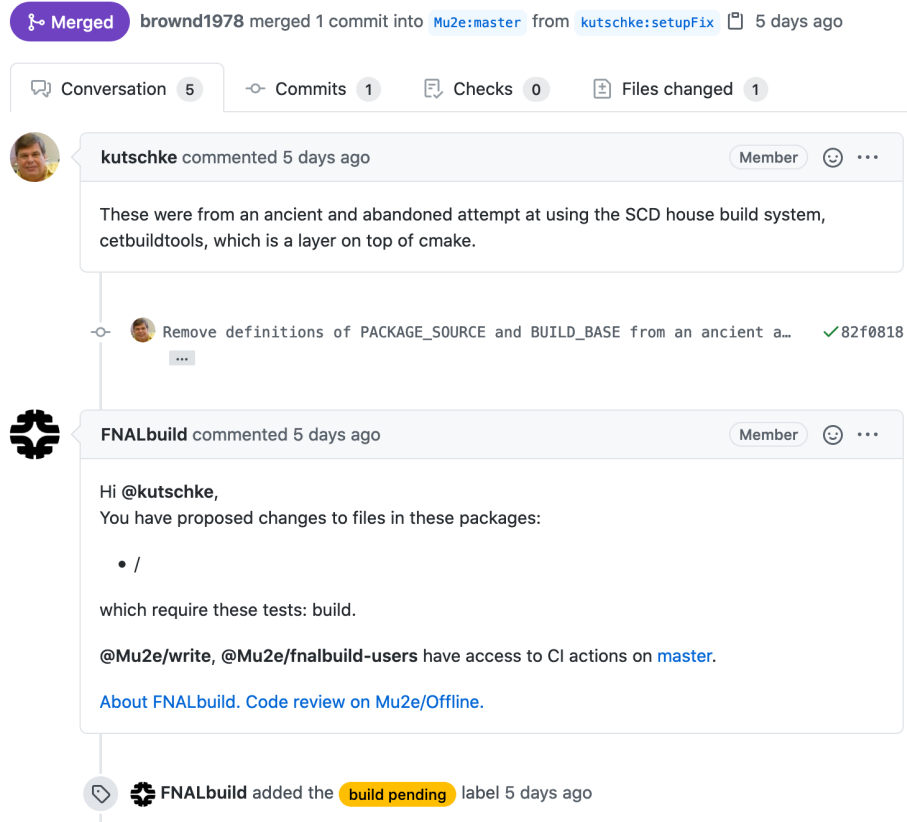


Figure 4.1: A screenshot of a pull request, or proposed change, to the Offline software repository on GitHub. An introductory comment is posted automatically by the FNALbuild bot user.

When opening a pull request proposing changes, a ‘bot’ user posts an introductory comment (see Figure 4.1.) The continuous integration tests are triggered by an authorised user. Following their comment, the Jenkins job is started, and the build status is reported as running (Figure 4.2.) The build and tests finish, and the results are reported in a summary (see Figure 4.3.)

The implementation of this system uses the GitHub Application Programming Interface (API) to receive notifications about active pull requests from the Mu2e GitHub repository. Such notifications are dispatched to the Jenkins server at Fermilab, where a custom script decides what action to take on the pull request given any recent comments or changes. If a trigger comment is detected, a Jenkins job is further triggered to build and run the tests on the Software. Log files generated during testing are uploaded to GitHub where they can be accessed to troubleshoot any issues. These scripts are versioned in the Mu2e/CI GitHub repository at the time of writing.

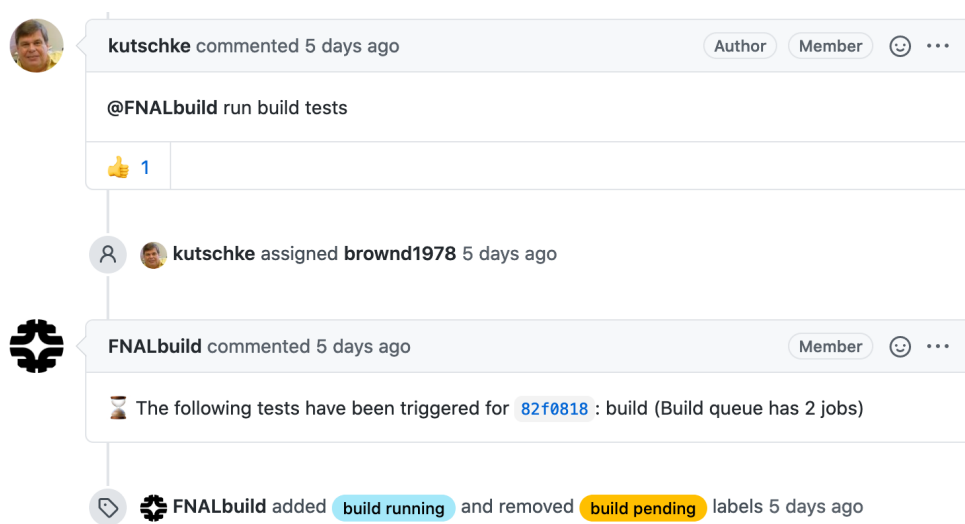
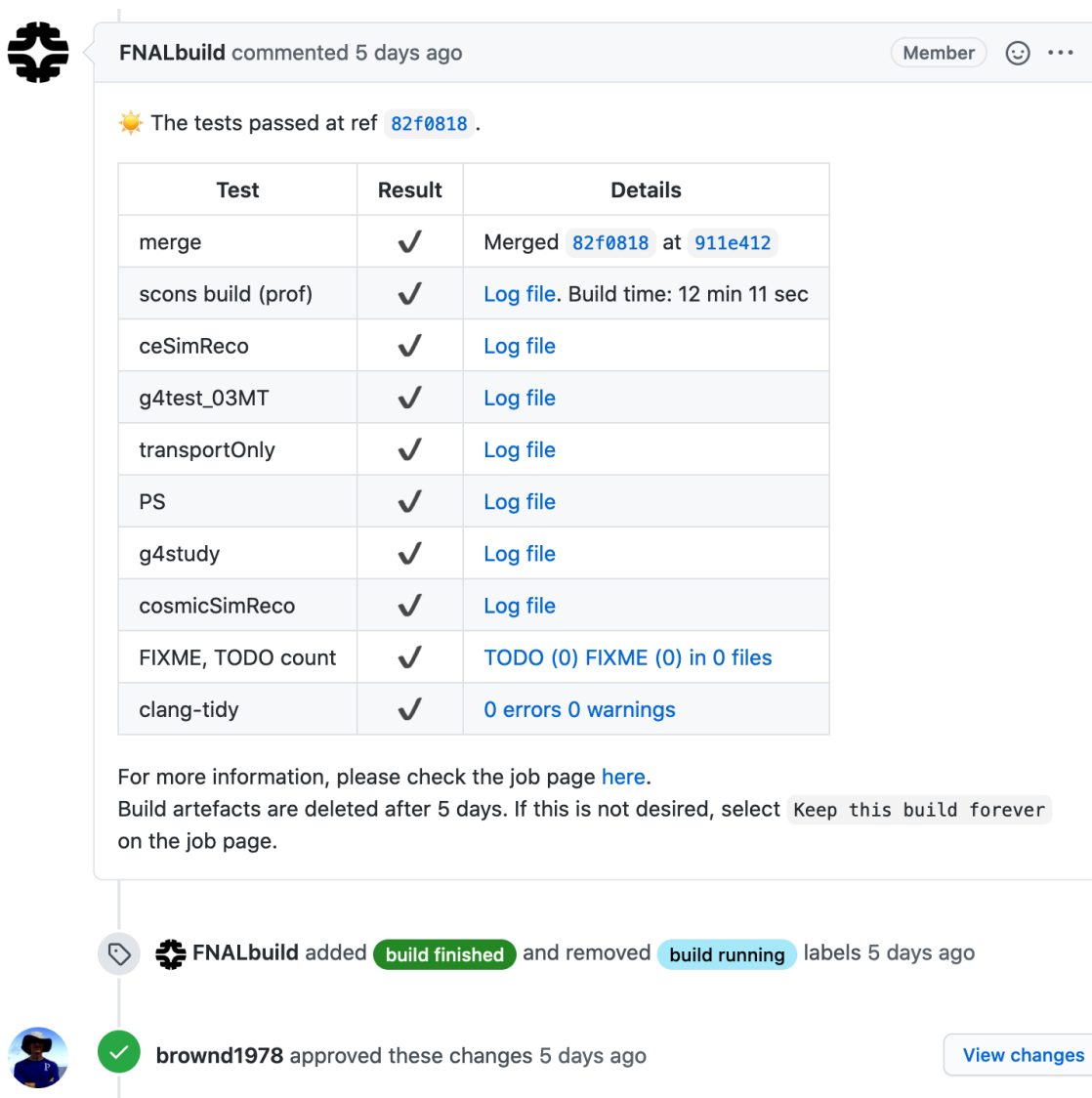



Figure 4.2: A screenshot of an Offline software repository maintainer triggering a CI test, by posting a comment.

In addition to the custom GitHub interface, a suite of bash shell scripts were written to build and test the Mu2e Offline software. These scripts download, in a local detached copy, the correct version of the software containing the proposed changes, merged into the latest version of the master branch using git, a version control system. The software is built using the ‘SCons’ build system, and a collection of ‘FHI CL’ [45] [46] [47] art job configurations are tested. In addition to these art jobs, a static code analysis on files affected by the proposed changes are performed using the clang-tidy code analysis tool. Additionally, to test any changes to the geometry, the latest version of the geometry is tested for ‘illegal’ volume overlaps. At the time of writing, these scripts are versioned at Mu2e/codetools on GitHub. The CI system has, at the time of writing, been in use in Mu2e Offline software development for 8 months and has processed 95 out of 96 pull requests in that time, of which 76 were merged. In total, 234 tests have been run over this sample of pull requests.

Extensions to this testing system may be developed moving forward, particularly to increase the coverage of the tests. Increased test coverage is likely to increase the likelihood of automated detection of software errors or undesired behaviours, if present. Any remaining lack of coverage could be eliminated by developing a comprehensive unit testing system for each package to validate the function of actively utilised art modules. Alternatively, or additionally, the Validation package may be extended to cover previously untouched areas of the codebase, and the corresponding job configurations appended to




FNALbuild commented 5 days ago Member  ...

🌟 The tests passed at ref [82f0818](#).

Test	Result	Details
merge	✓	Merged 82f0818 at 911e412
scons build (prof)	✓	Log file . Build time: 12 min 11 sec
ceSimReco	✓	Log file
g4test_03MT	✓	Log file
transportOnly	✓	Log file
PS	✓	Log file
g4study	✓	Log file
cosmicSimReco	✓	Log file
FIXME, TODO count	✓	TODO (0) FIXME (0) in 0 files
clang-tidy	✓	0 errors 0 warnings

For more information, please check the job page [here](#).
Build artefacts are deleted after 5 days. If this is not desired, select [Keep this build forever](#) on the job page.

 **FNALbuild** added [build finished](#) and removed [build running](#) labels 5 days ago



  **brownd1978** approved these changes 5 days ago [View changes](#)

Figure 4.3: A screenshot of the reported status of CI test results. Once the tests have completed, their outcomes are reported in a GitHub comment on the pull request. The pull request reviewer follows up by inspecting the results and making a decision to approve the changes.

the tests.

4.3 Alignment Conditions

As discussed in the next chapter, corrections to the detector model are determined by alignment algorithms in the form of alignment constants. These must be persisted in a database, or by some other storage method. Since alignment conditions can change over time, alignment constants at different values may be applicable to data acquired at different times. Additionally, the correct version of these constants corresponding to an event must be easily accessible from within the software.

The alignment conditions are managed by the Proditions system. Proditions is a software interface between a data storage source, such as PostgreSQL databases, text files, or art FHiCL configurations, and code in the Offline software. The interface is realised via the configuration of single objects or ‘entities’ using the values stored in the database. The objects can, for example, represent the Tracker and its components. These objects are created once at job start-up, and the memory is shared across all art modules that require access [48], [49].

Alignment constants are either stored in a PostgreSQL database, or flat text files depending on the use-case. For intermediate or unapproved alignment results, for example, a flat text file may be most suitable. For current, up-to-date versions of the alignment conditions for use by the experiment, they may be stored and versioned in a centralised database.

4.4 Tracking Algorithms

This section includes a brief description of tracking algorithms implemented and in use within the Offline software. Tracking is a crucial component of detecting and measuring the properties of candidate conversion electrons, and refers to the process of reconstructing a parameterised particle trajectory from detector measurements (referred to as hits) by use of fitting algorithms. A hit can be defined as the detection of a particle as it traverses a straw.

4.4.1 Time-based no-field cosmic track fit

A MINUIT-based [50] track fit, implemented within the CosmicReco package, reconstructs the trajectory of cosmic particles traversing the Tracker where there is no DS magnetic field. It attempts to fit a straight line trajectory best conforming to a set of Tracker straw hits, or ComboHits, grouped closely together in time. The track model is parameterised as

$$\mathbf{P} = \begin{bmatrix} A_0 \\ B_0 \\ A_1 \\ B_1 \\ T_0 \end{bmatrix}, \quad (4.1)$$

where A_0, B_0 are the x and z coordinates of the trajectory in the detector coordinate system for $y = 0$, and A_1, B_1 form two components of the track direction vector

$$\mathbf{r} = \begin{bmatrix} A_1 \\ -1 \\ B_1 \end{bmatrix}, \quad (4.2)$$

expressed as detector system coordinates. The T_0 parameter is the time that the particle was present at the intercept

$$\mathbf{r}_0 = \begin{bmatrix} A_0 \\ 0 \\ B_0 \end{bmatrix}, \quad (4.3)$$

in the detector coordinate system [49] [51] [52].

The position of the particle at a time t , the elapsed time since T_0 , is

$$\mathbf{x} = \mathbf{r}_0 + vt\hat{\mathbf{r}}, \quad (4.4)$$

where v is the speed of the particle.

For a straw hit associated with a track, the corresponding straw wire midpoint \mathbf{w} , and wire direction $\hat{\mathbf{r}}_w$ are used in conjunction with the track ‘intercept’ and direction vectors \mathbf{r}_0 and $\hat{\mathbf{r}}_t$ to calculate the points of closest approach between the track and wire, where \mathbf{p}_0 is situated on the track, and \mathbf{p}_1 on the wire. The distance of closest approach is then defined as

$$r = \text{DOCA}(\mathbf{r}_0, \hat{\mathbf{r}}_t, \mathbf{w}, \hat{\mathbf{r}}_w) = |\mathbf{p}_1 - \mathbf{p}_0|, \quad (4.5)$$

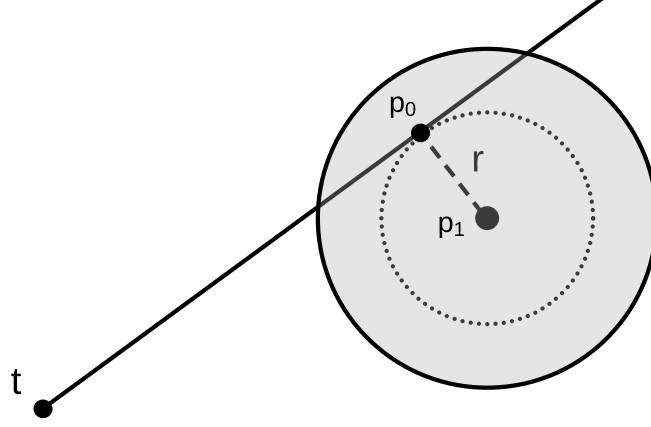


Figure 4.4: Illustration of a track (solid line) traversing a straw (solid circle) and the distance of closest approach r between p_0 on the track and the wire p_1 . The distance between t , the start point of the track as it entered the detector coordinate system, and p_0 can be used to calculate the trajectory time t_{traj} . No multiple scattering is assumed present.

and can be used to express how far the drifting charge travelled in the straw gas medium before being collected by the wire (the ‘drift radius’.)

The particle trajectory time for the hit, or the time the particle spent travelling from point r_0 to p_0 is

$$t_{\text{traj}} = \frac{(\mathbf{p}_0 - \mathbf{r}_0) \cdot \hat{\mathbf{r}}_t}{v}, \quad (4.6)$$

where v is the particle speed. The track time residual is now defined

$$z_i = T_D(r) + T_0 + t_{\text{traj}} + t_{\text{offset}} + t_{\text{prop}} - t_{\text{hit}}, \quad (4.7)$$

where t_{offset} is the additional time the particle spent travelling past p_0 before liberating an electron in the straw gas medium, t_{prop} is the wire propagation time, and t_{hit} is the measured time of the hit i.e. the time at which the collected drift charge is detected after propagation along the straw sense wire. The nonlinear function T_D performs a drift distance to time conversion, it is known as T2D in the Offline software.

An optimisation library, MINUIT [50], is used to find the maximum-likelihood best fit parameters \mathbf{P} given a set of straw ComboHits. The minimisation objective function is defined as

$$F = \sum_{i \in \text{hits}} \left(\frac{d_t - (d_h)}{(\sigma_w)} \right)_i^2 + \left(\frac{z_i}{\sigma_i} \right)^2, \quad (4.8)$$

where d_t is ‘wire distance’, or the distance between the point of closest approach (PCA) on the wire to the trajectory, and the wire midpoint. The quantity d_h is the wire distance,

where the PCA is calculated using the hit and not the trajectory. This squared term effectively serves to minimise the difference between the wire distance for the trajectory vs. measured hit information during the fit over all hits [49]. The error σ_w is the longitudinal resolution along the wire.

The second squared term is the ‘time residual’ z_i divided by the error on the drift time, σ .

Each art event is checked for straw ComboHits, where each object represents one particle detection by a Tracker straw. Hits occurring closely together in time, within a 100 ns time window, are grouped together and passed to the LineFinder algorithm, which attempts to find a starting estimate of the trajectory parameters using an iterative brute-force technique. This trajectory estimate is used as a ‘seed’ for the full fit [49].

Once the full fit converges, the result is stored inside a data product containing the ComboHits used in the fit, fit parameters and covariance matrix. The data product, which is called CosmicTrackSeed, may then be accessed by other modules. Diverged fits are also stored in this way, but with the appropriate flag set to indicate a failed fit. They may be discarded downstream by filters, or other modules created by users.

A small comment should be made here about the choice of the MINUIT library for this particular track fit, and perhaps track fitting in general. The documentation of the library itself states that “Although Minuit will of course solve easy problems faster than complicated ones, it is not intended for the repeated solution of identically parametrized problems (such as track fitting in a detector) where a specialized program will in general be much more efficient.” [53]

4.4.2 Kalman-filter track fit

The experiment uses a Kalman Filter to reconstruct the path of conversion electrons following a helix trajectory in the Detector Solenoid.

A Kalman filter uses a series of measurements made in time, hindered by some amount of ‘process noise’, to calculate previously unknown estimates of variables. The combination of multiple measurements in this way provides a better estimate of the unknown variables than a single measurement would. In the context of reconstructing a track, it may be used to estimate the track parameters at different points on the track. The application of the Kalman filtering equations to detectors and track reconstruction is outlined

in detail at [54].

The experiment at the time of writing is using the BTrk software package [43], originally developed for the BaBar experiment. BTrk implements the Kalman Filtering equations using a “hybrid” of the weighted-means and gain matrix formalism. Measurements (or ‘hits’), material effects, and magnetic-field effects are encapsulated inside C++ objects. These objects describe how to update the Kalman filter equations, and provide the manipulations necessary for a track to be filtered in the ‘forward’ (outward) and ‘backward’ (inward) direction [43].

Eventually, it is planned for this fit to be phased out and replaced by the KinKal Kalman Filter Track fitting package, which is in beta release at the time of writing. In contrast to BTrk, which provides a geometric track fit, KinKal aims to fit a track so as to provide the kinematic four-vector of particles as a function of time [55].

4.5 STM Test Beam software

The Stopping Target Monitor is due to undergo a test-beam, where the Data Acquisition (DAQ) system, detector, and detector hardware performance can be tested and studied in a live beam setting. A particularly useful component of DAQ operations, if available, is access to real-time monitoring of acquired data and DAQ system health during data collection. Real-time monitoring, or “On-line” monitoring can be developed for such a purpose to allow problems to be detected during the test-beam period, and/or study any acquired physics data if needed.

Additionally, timely decoding and storage of the physics data in an end-user friendly dataset format (accessible via widely-supported, common data analysis tools) allows analysis of the data to be conducted promptly, and with minimal special training.

For the purposes of the test-beam, the MIDAS DAQ software system is used by the experiment for the STM. A software interface (from now on, the “Middleware”) was created, and written in C++. The purpose of the middleware was to connect to a running instance of MIDAS and stream acquired event data from the STM in real-time, while managing real-time monitoring histograms and other monitoring figures, and writing the test beam data, in an analysable state, to datasets stored on disk.

A real-time monitoring dashboard was developed using `plotly dash` [56] which was

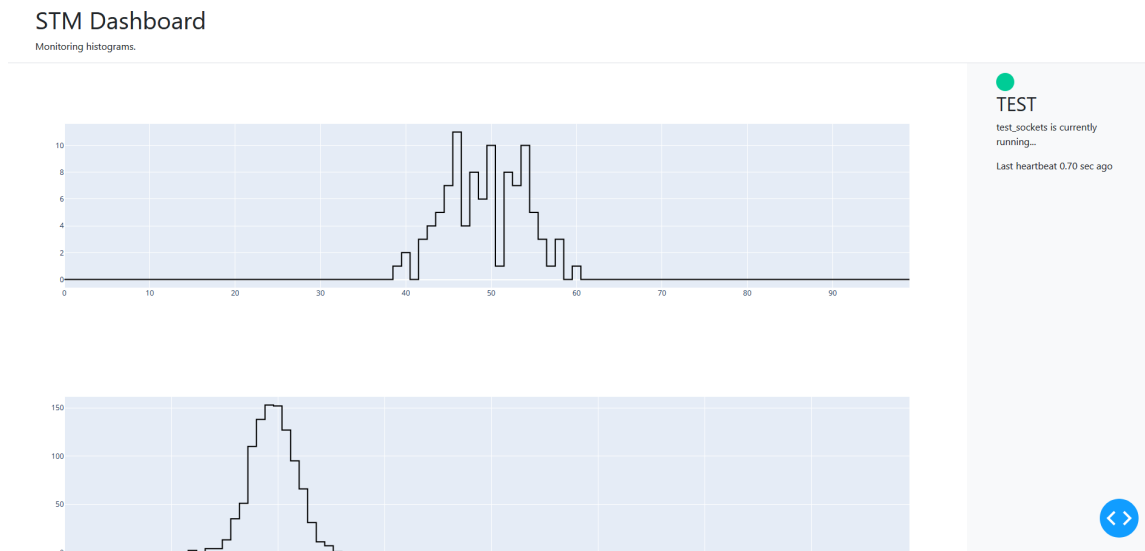


Figure 4.5: A screenshot of the monitoring dashboard in a testing stage. Two histograms with random values sampled from gaussian distributions are shown updating in real time, with a few hundred samples added each second to each histogram. There is a status bar on the right-hand side which provides information about when an update was last received from the middleware, and other information can be added to extend its usefulness.

capable of displaying histograms and other figures in real-time, while connected to the middleware. The design of this component enables the real-time dashboard to be accessed remotely over the internet using a standard web browser, for example, by users not present at the test-beam location. See Figure 4.5 for a screenshot example.

Due to time constraints, and delays caused by world events (COVID-19), the test beam could not go ahead at the expected date. As a result, the middleware remains in an incomplete state at the time of writing. Once the STM test beam is able to go ahead, the package can be revived and tailored to the specific operational needs of the programme.

Chapter 5

Tracker Alignment

5.1 Detector Alignment

A particle tracking detector involves many components, which may move around and deform over time. To track particles, a model of the detector is required, including a geometry description. A ‘geometry’ in this case refers to a representation of the detector structure. It should contain enough information to sufficiently describe the shape, position, orientation, and usually material of all components. A detector geometry cannot be assumed to be correct within the desired accuracy without some procedure to validate correctness, and determine any misalignments if needed. If a geometry does not truly represent the detector in the ‘real world’ to satisfactory accuracy, then it can be said to be misaligned, and the misalignments will need to be determined to correct inaccuracies in the model. A misalignment can be described as the geometric transformation of a component from the actual to the expected position and orientation in the detector. In an alignment, the parameters describing the reverse of this transformation are ideally determined.

A badly modelled detector, when used for particle track reconstruction, will interfere with the quality of important physics measurements. The reconstruction of the passage of a particle through the Mu2e Tracker, for example, relies on registered ‘hits’ on a sufficient number of straw drift tubes. A hit, as earlier defined, is the detection of a particle as it traverses a straw. The physics of straw drift tubes are covered in Section 3.4.4.

Knowing the straw locations and time of each hit allows one to infer how a particle moved through the detector, and the properties of the incident particle. Fitting a param-

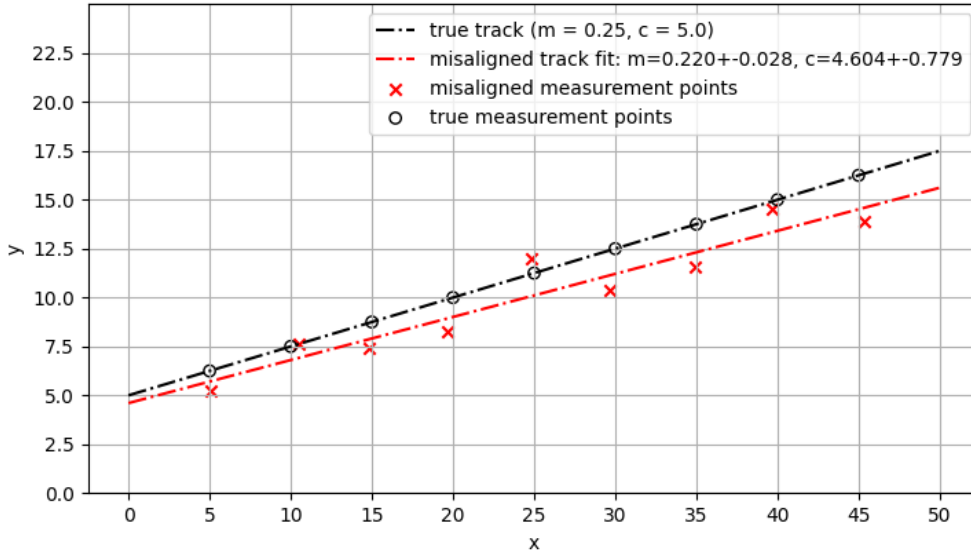


Figure 5.1: A very simple example of how detector misalignment can affect tracking. A particle is introduced, following a straight trajectory, such that its position in y given x is described by the equation $y = mx + c$, with gradient and intercept parameters m, c . Measurements are generated for x and y coordinates in the given coordinate system. The black circular points representing ‘true’ measurements are sampled directly from the track equation for a chosen set of parameters $m = 0.25, c = 5.0$ in the interval $[5, 45]$ every 5 units, where one detector element can be assumed to measure one pair of (x, y) -coordinates. The red crosses represent those same measurements now affected by detector misalignments, and the misalignment is simulated using uniform random numbers added to the true x, y values, picked from the range $[-0.5, 0.5]$ and $[-2.5, 2.5]$ for each coordinate respectively. Finally, a track is fitted (red dashed line) to the ‘misaligned’ points using a least squares fitting algorithm, in an attempt to determine the m, c parameters again. However, the hits are in the wrong location leading to larger residuals and an inaccurate fit result.

eterised trajectory (or a ‘track’) can allow further measurements to be derived from the particle’s passage, such as the momentum based on the curvature of the trajectory, caused by bending in the presence of a magnetic-field. If the straw locations are not correct, then the fitted trajectory is affected and the fit performance suffers. In other words, the detector misalignments are captured in the trajectory parameters, causing the degraded result of the track fit. See Figure 5.1 for an example.

Misalignments can be introduced during normal operation and data-taking, during the commissioning stage, and during maintenance. While a detector will be built as close to specifications as possible, it is difficult in practice to match specifications to required accuracy, due to human error, limits on precision during assembly, and other imperfections that arise during the process. A common mitigation is to do a survey which aims to measure as accurately as possible the positions of each component, imperfections in component shape (e.g. material bending and deformations), orientations, and then make corrections to the detector geometry as appropriate. However, especially in the case of the Mu2e Tracker, a survey will have limited accuracy, and some measurement techniques will not be possible without moving or disassembling the detector to some degree, so additional methods are required. It should also be noted that the Tracker is serviced yearly, and a full survey may not always be possible after reassembly.

A data-driven, track-based alignment uses a sample of reconstructed tracks to find the optimal set of corrections to a geometry by allowing components in the model to move or rotate (for example) in order to optimise some tracking performance metric e.g. the χ^2 -function over all tracks. An alignment algorithm aims to determine the ‘alignment constants’, which correspond to degrees of freedom (or parameters) introduced into the detector model, with statistical methods. The alignment degrees of freedom often involve but are not limited to the rigid-body translations or rotations of all components from nominal values. Degrees of freedom that describe shape deformations, such as the sagging of a straw drift tube wire, may also be introduced.

The benefit of data-driven alignment is that it can be performed as long as the detector can be operated, and the surveys accurate enough that the remaining misalignments are small, but not usually small enough that measurements are unaffected. Regardless, the detector does not need to be disassembled or moved from its operating configuration as would be necessary for a survey, which would cause the misalignments to vary, and so

the detector does not need to be physically disturbed.

After Tracker construction and survey, remaining misalignments may be large (order of 1-2 mm). Cosmic ray data collected without magnetic field during the initial commissioning period can be used to eliminate these misalignments before data-taking, perhaps in conjunction with magnet-on beam and cosmic ray data. After commissioning, alignments can be performed periodically (depending on misalignment variations over time) using beam and cosmic data taken during normal operation.

A track-based alignment software package, TrackerAlignment, was created for the Mu2e Offline software and is present in the GitHub repository at the version [49]. The package contains all the necessary configurations and software routines to interface with cosmic track reconstruction routines, the alignment conditions database, Tracker geometry routines, and other utilities for the purpose of producing track samples for alignment with the *Millepede-II* software package (see Section 5.3). Utilities are provided to generate Millepede job configuration files after track reconstruction. The package is compatible with standard Mu2e Grid workflows and can be scaled to process large numbers of events. A Millepede-II input dataset writer was created to enable compressed datasets to be written at double precision. A full flowchart outlining the process in software is provided in Figure 5.2.

The TrackerAlignment package is extendable such that other track types may be supported in future, and a utility (mu2ealign) is provided to simplify some tasks in the Alignment workflow. One should note that any track will be re-fit by Millepede using the least-squares method, even if the global track parameters were originally determined by a Kalman Filter, for example. Documentation is provided in Appendix A.

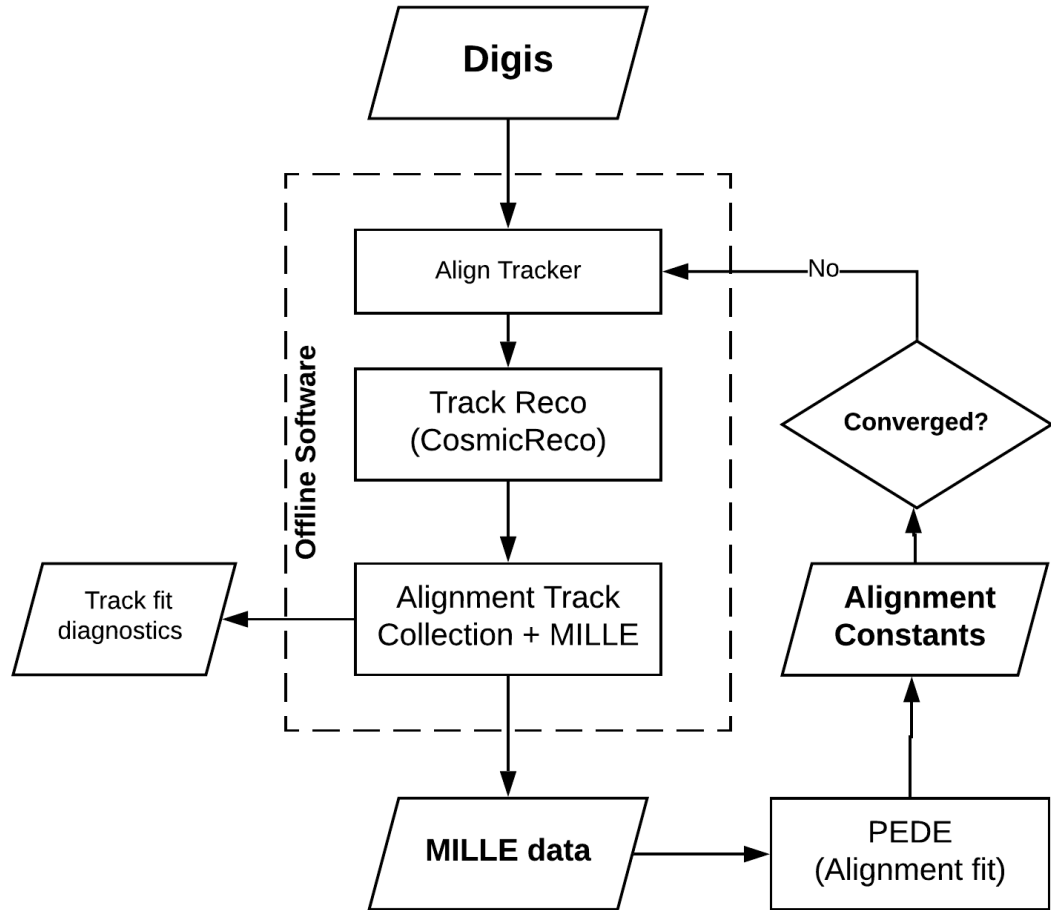


Figure 5.2: A flow diagram showing the software processes involved in aligning the detector. ‘Digis’ refer to digitised detector data, which are used to reconstruct tracks. Once the Offline software job begins, the Tracker geometry is created and aligned using the initial best-estimate of the alignment constants, either from a survey or from a previous alignment fit. This geometry is used to reconstruct the tracks from the digitised straw hits (digis), and tracks satisfying certain criteria are ‘collected’ and written to a ‘Mille’ dataset. (‘Mille’ describes the process of writing the input data for the Millepede-II program.) This ‘Mille’ data is provided to the PEDE executable, which executes the global alignment fit and returns the next best estimate of the alignment constants based on the track sample. The process continues as shown until the alignment is deemed to have converged.

5.2 Misalignment impact on tracking

It is possible to study how misalignments can affect track reconstruction performance using the Mu2e Offline software. The software is able to simulate the traversal of particles through the Tracker whether from decay-in-orbit electrons, cosmic muons, or otherwise, and the Tracker's response to those incident particles. The Monte-Carlo (MC) production jobs produce 'digitised' (or 'digi') events by simulation which contain the necessary information used by reconstruction (or 'reco') routines to fit tracks and make measurements.

In the MC digi production stage, due to software constraints, misalignments are not inserted before event generation. Therefore, misalignments must be randomly generated and applied to the geometry in the reco stage, before reconstructing any tracks. This is achieved by moving the straws to 'misaligned' positions using the ProditionsService Conditions Database in order to misalign Tracker Planes before reconstruction.

To demonstrate the impact of misalignments on the reconstruction of conversion electrons, the Tracker Planes were misaligned according to a 1 mm twist around z , 5 mm skew around x and y , and a 5 mm squeeze along z . Actual values are uniformly random in the range $[-x, x]$ [57].

Tracks were reconstructed with and without misaligning the Tracker, and a standard set of cuts were applied to select Conversion Electrons coming from the Stopping Target within the correct time window. The event was accepted if the Kalman Filter successfully converged, the track T_0 parameter was in the time window $700 \text{ ns} < T_0 < 1695 \text{ ns}$, the track 'tan-dip' (the tan of the helix angle) was consistent with a particle emerging from the target, the track impact parameter d_0 was in the range $[-80, 105] \text{ mm}$, and the track radius was in the range $[450, 680] \text{ mm}$. Each event was weighted by the Proton Bunch Intensity (PBI). The track reduced χ^2 and momentum resolution after selection can be seen in Figure 5.3. This data selection does not include any track quality metrics determined using Multivariate analyses (such as with the TrkQual module).

The figures demonstrate how the momentum resolution and reduced track- χ^2 are affected before and after Plane-only misalignments are applied. The reduced- χ^2 is no longer peaked at 1, which indicates a worsened fit performance. The momentum resolution has also widened, which demonstrates the negative impact of worsened tracking on physics measurements. If the misalignment modelling was to be extended to involve more degrees of freedom, other detrimental effects may come to light. However, from this short

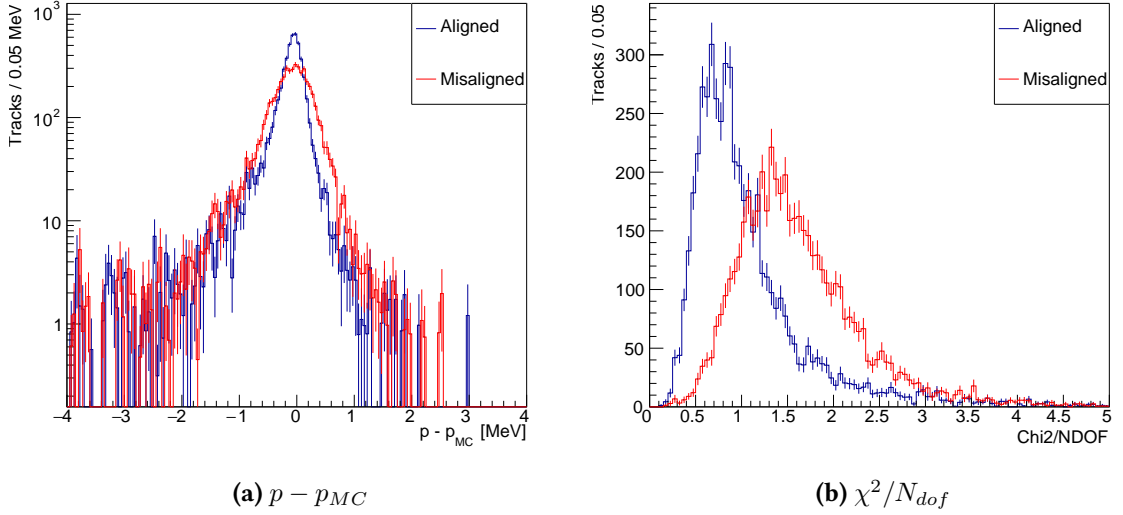


Figure 5.3: Before and after random translational and rotational misalignment of the Tracker Planes. The effect on tracking is clearly manifested in the general increase of χ^2/N_{dof} , and widening of the momentum resolution $p - p_{MC}$.

study one can conclude that an Alignment algorithm to correct at least for Plane-level misalignments can help to maintain acceptable tracking performance.

5.3 Millepede for detector alignment

The Millepede algorithm, as implemented in the *Millepede-II* software package, provides an approach to the detector alignment problem by determination of the alignment constants using a large sample of tracks and least-squares methods. Each track represents a fit of a track model to a set of measurements and their errors for one ‘event’.

A track fit may originally be carried out using Least-squares, Kalman-filter, or maximum-likelihood fit methods - but the result is a set of model parameters that describe the full passage of a particle through the detector (a *track*), and the error estimates on those track parameters or a covariance matrix. If alignment parameters, which are needed to correct misalignments, are omitted as free parameters, a track fit, based only on the track model variables, produces biased results. The bias arises since detector components are assumed to be placed differently to where they are in reality, which in turn leads to systematically displaced reconstructed hits in data.

The implementation of Millepede is in active development, and has changed since the creation of the algorithm. In the following explanations and subsections, reference is

made throughout to the source code [58], Millepede-II manual accessible at [59] (manual is versioned at [58]), and the original conference paper [60].

In Millepede, a linear least-squares “simultaneous fit of global and local parameters” is performed, and the system of normal equations corresponding to such a fit are reduced to only include the global parameters. The ‘global’ parameters in this context refer to the alignment parameters, and the ‘local’ parameters are the track model parameters for all input tracks. By simultaneously fitting the alignment parameters and the track models to the track sample the biases are eliminated, assuming that the alignment degrees of freedom model the present misalignments effectively, the track sample is sufficient, and the global fit is adequately constrained. The objective function effectively minimised by solving the system of equations is

$$F(\mathbf{p}, \mathbf{Q}) = \sum_{j \in \text{tracks}} \sum_{i \in \text{hits}} \frac{z_{i,j}^2}{2 \sigma_{i,j}^2}, \quad (5.1)$$

where j corresponds to the track index, i the index of the measurement used to fit track j , $z_{i,j}$ the residual on measurement i of track j , and $\sigma_{i,j}$ the standard deviation of the measurement. The vector \mathbf{p} holds the alignment parameters, and \mathbf{Q} represents the collection of track model parameters for all tracks where \mathbf{Q} has dimensions $N \times M$ for N tracks with M parameters. This definition of the objective function corresponds to the χ^2 -function, and will follow a χ^2 -distribution in ‘well-behaved’ data.

A residual associated with some track j is defined as

$$z_{i,j} = m_{i,j} - f(x_{i,j}, \mathbf{Q}_j), \quad (5.2)$$

where \mathbf{Q}_j are the track model parameters corresponding to the track, $m_{i,j}$ is the (independent) track hit measurement, and $x_{i,j}$ describes the point on the fitted track corresponding to the measurement location (e.g. the time, flight length, or point of closest approach.) A measurement $m_{i,j}$ is assumed to be adequately modelled by the track model f such that

$$m_{i,j} = f(x_{i,j}, \mathbf{Q}_j) + \epsilon_{i,j}, \quad (5.3)$$

for a perfectly aligned detector, where $\epsilon_{i,j}$ is the measurement error with standard deviation $\sigma_{i,j}$, and i is the index of the track measurement. Moving towards vector notation, the track- χ^2 may be defined

$$\chi_j^2 = (\mathbf{m}_j - f(\mathbf{x}_j, \mathbf{Q}_j))^T \mathbf{V}^{-1} (\mathbf{m}_j - f(\mathbf{x}_j, \mathbf{Q}_j)) \quad (5.4)$$

$$= \mathbf{z}_j^T \mathbf{V}^{-1} \mathbf{z}_j, \quad (5.5)$$

where \mathbf{m}_j is a vector of M track measurements, \mathbf{z}_j is the vector of residuals for track j with length M , and \mathbf{V}^{-1} is a symmetric covariance matrix with dimensions $M \times M$.

With the inclusion of the alignment parameters \mathbf{p} in the track model f during a fit, \mathbf{p} can be varied such that $z_{i,j}$ for one track j is at a minimum, and any remaining discrepancy between the track model and the measurement is due to random effects, or other tracking defects. However, it is not practical to attempt to determine all of or part of \mathbf{p} in each single track fit. The solution is determined by considering the entire (valid) input track sample in order to find a global minimum in $F(\mathbf{p}, \mathbf{Q})$.

5.3.1 Local fits

The first stage of a Millepede alignment iteration is to determine the ‘local fit’ parameters \mathbf{Q}_j for each track j , according to the linear least-squares method. The initial values of the fit parameters \mathbf{q}_0 are provided by track reconstruction routines in the experiment’s software, which may use different algorithms. The partial derivatives of f with respect to each variable of the track model are also required.

For a non-linear track model f , the fit is achieved by linearising f and solving for corrections $\Delta\mathbf{Q}_j$, over multiple iterations until the contributions from $\Delta\mathbf{Q}_j$ per iteration are small. For a linear model, one iteration is sufficient, since no approximation is made. If necessary, since multiple iterations of the global fit may have been performed in earlier Millepede iterations, the residuals are first corrected to account for any change in \mathbf{p} .

Following the derivation in [65], the track model (local) derivatives, for one track, with respect to the track parameter vector \mathbf{Q}_j are

$$\mathbf{H}_j = \left. \frac{\partial f(\mathbf{x}_j, \mathbf{Q}_j)}{\partial \mathbf{Q}_j} \right|_{\mathbf{q}_0}, \quad (5.6)$$

which emerge from the linearisation of the track model

$$f(\mathbf{x}_j, \mathbf{Q}_j) = f(\mathbf{x}_j, \mathbf{q}_0) + \mathbf{H}_j(\mathbf{Q}_j - \mathbf{q}_0), \quad (5.7)$$

where \mathbf{H}_j is the matrix of track model derivatives for the track j [65]. The χ^2 minimisation for a single track, now considering the linearised track model, can be expressed as

$$\frac{d\chi^2}{d\mathbf{Q}} = -2\mathbf{H}^T \mathbf{V}^{-1}(\mathbf{z} - \mathbf{H}(\mathbf{Q} - \mathbf{q}_0)) \equiv 0, \quad (5.8)$$

where the track index j has been omitted for brevity. The equation is solved for the new track parameter estimates

$$\mathbf{Q} = \mathbf{q}_0 - \mathbf{C}\mathbf{H}^T\mathbf{V}^{-1}\mathbf{z}, \quad (5.9)$$

where

$$\mathbf{C} = (\mathbf{H}^T\mathbf{V}^{-1}\mathbf{H})^{-1}, \quad (5.10)$$

is the covariance matrix of \mathbf{Q}_j [65].

The residuals are now corrected to be consistent with the new local fit parameters,

$$\begin{aligned} z'_{i,j} &= m_{i,j} - f(x_{i,j}, \mathbf{Q}_j) \\ &= z_{i,j} - \sum_k \left(\frac{\partial f}{\partial (\mathbf{Q}_j)_k} \right)_i (\Delta \mathbf{Q}_j)_k, \end{aligned} \quad (5.11)$$

where the sum is over the track model parameters, $(\mathbf{Q}_j)_k$ is the k^{th} track parameter variable, and $(\Delta \mathbf{Q}_j)_k$ is the correction to the k^{th} component of \mathbf{Q}_j .

Before processing the next track, the current track is either accepted or rejected. If the local fit fails to converge, or if the χ^2 indicates that the influence of outliers is too great, the track is rejected. The algorithm may also be configured to cut tracks exceeding some sensible threshold, and these checks also indirectly check the validity of the input data, increasing the robustness of the method.

5.3.2 Global fit

Once the local fits for all viable tracks are processed, estimates of the track parameters and track covariance matrices are now available, given an initial estimate of the alignment parameters \mathbf{p}_0 . The track model is extended to include the alignment parameter dependence

$$f(x_{i,j}, \mathbf{Q}_j) \rightarrow f(x_{i,j}, \mathbf{Q}_j, \mathbf{p}), \quad (5.12)$$

and the global alignment fit aims to find a global minimum in $F(\mathbf{p}, \mathbf{Q})$ such that the conditions

$$\frac{\partial \sum_j \chi_j^2}{\partial \mathbf{p}} = 0, \quad \frac{\partial \chi_j^2}{\partial \mathbf{Q}_j} = 0 \quad \forall j, \quad (5.13)$$

are met simultaneously [65]. In summary, the χ^2 summed over all tracks are to be minimised with respect to the alignment parameters \mathbf{p} , and the track- χ^2 are also minimised with respect to the track parameters, for each track j .

To demonstrate the derivation of the system of equations required for the global minimisation, the approach in [65] is followed, starting at the total derivative (taking into account the dependence of Q_j on p)

$$\frac{d}{d\mathbf{p}} = \frac{\partial}{\partial \mathbf{p}} + \frac{dQ_j}{d\mathbf{p}} \frac{\partial}{\partial Q_j}. \quad (5.14)$$

Since

$$\frac{\partial \chi_j^2}{\partial Q_j} = 0 \Rightarrow \frac{d}{d\mathbf{p}} \frac{\partial \chi_j^2}{\partial Q_j} = 0, \quad (5.15)$$

we may substitute Equation 5.14 into the above and rearrange to get

$$\frac{dQ_j}{d\mathbf{p}} = -\frac{\partial^2 \chi_j^2}{\partial \mathbf{p} \partial Q_j} \left(\frac{\partial^2 \chi_j^2}{\partial Q_j^2} \right)^{-1}. \quad (5.16)$$

By now considering the total derivative of χ_j^2 and substituting in Equation 5.16, it is found that

$$\frac{d\chi_j^2}{d\mathbf{p}} \equiv 0, \quad (5.17)$$

and after linearising this condition given some initial estimate of the alignment parameters p_0 , an updated estimate of the alignment parameters can now be determined by the system of N linear equations

$$\left(\sum_j \frac{d^2 \chi_j^2}{d\mathbf{p}^2} \bigg|_{p_0} \right) \Delta \mathbf{p} = \left(\sum_j \frac{d\chi_j^2}{d\mathbf{p}} \bigg|_{p_0} \right), \quad (5.18)$$

where the derivatives have been calculated for $\mathbf{p} = \mathbf{p}_0$, only N degrees of freedom are present in this minimisation for N alignment parameters, and the dependence of the track parameters on \mathbf{p} are included [65].

To find the first-order and second-order derivatives for each track that contribute to the sums in Equation 5.18, the residual vector

$$\mathbf{z}_j = \mathbf{m}_j - f(\mathbf{x}_j, \mathbf{p}), \quad (5.19)$$

is linearised about $(\mathbf{x}_j(p_0), p_0)$, and the global residual derivatives with respect to each alignment parameter are calculated,

$$\mathbf{A}_j = \frac{\partial \mathbf{z}_j}{\partial \mathbf{p}}, \quad (5.20)$$

where \mathbf{A}_j is a matrix with dimensions of M measurements $\times N$ alignment parameters. The first and second total χ_j^2 derivatives with respect to \mathbf{p} with linearised \mathbf{z}_j are

Solution Method	Description
inversion	Inversion of the matrix of linear equations
cholesky	Solution via cholesky decomposition.
diagonalization	Solution by diagonalisation, also provides eigenvalues.

Table 5.1: A table of solution methods supported by Millepede-II involving the global matrix C .

$$\frac{d\chi^2}{d\mathbf{p}} = \frac{\partial\chi^2}{\partial\mathbf{p}} - \mathbf{A}^T \mathbf{V}^{-1} \mathbf{H} \mathbf{C} \frac{\partial\chi^2}{\partial\mathbf{Q}} \quad (5.21)$$

$$= 2\mathbf{A}^T \mathbf{V}^{-1} (\mathbf{V} - \mathbf{H} \mathbf{C} \mathbf{H}^T) \mathbf{V}^{-1} \mathbf{z}, \quad (5.22)$$

and

$$\frac{d^2\chi^2}{d\mathbf{p}^2} = 2\mathbf{A}^T \mathbf{V}^{-1} (\mathbf{V} - \mathbf{H} \mathbf{C} \mathbf{H}^T) \mathbf{V}^{-1} \mathbf{A}, \quad (5.23)$$

where these equations represent the contribution of a single track, and the index j has been omitted for brevity. These quantities are calculated and summed over the entire track sample, and then used to solve Equation 5.18 for the alignment parameter corrections $\Delta\mathbf{p}$. The covariance matrix of the alignment parameters are

$$\text{cov}(\mathbf{p}) = 2 \left(\sum_j \frac{d^2\chi_j^2}{d\mathbf{p}^2} \right)^{-1}, \quad (5.24)$$

which may be used to estimate the errors on the alignment parameter estimates [65].

There are a number of methods available in Millepede-II that handle the matrix arithmetic in various ways (and at various computational efficiencies) in order to solve for the alignment parameters, listed in Table 5.1.

Further iterations of the global fit may be carried out using a line search algorithm until convergence is reached [58], [59].

5.3.3 Generation of input data

Millepede-II requires the generation of a dataset in a prescribed format to perform a track-based alignment. Each dataset is a collection of tracks, or local fits. Each track has a collection of measurements. Each measurement must provide the data summarised in Table 5.2.

Quantity	Description
z_i	Track residual
σ_i	Measurement error
$\left(\frac{\partial f}{\partial \mathbf{q}}\right)_i$	Partial derivatives w.r.t. track parameters \mathbf{q}
$\left(\frac{\partial f}{\partial \mathbf{p}}\right)_i$	<i>Non-zero</i> partial (global) derivatives w.r.t. alignment parameters
ℓ	Alignment parameter labels corresponding to the global derivatives.

Table 5.2: A table of required Millepede input data per track measurement.

There are N alignment parameters \mathbf{p} across the whole detector. For a collection of measurements used to reconstruct a track, each measurement is made by a corresponding detector element. This individual detector element has n alignment parameters that, if changed, directly affect its position and orientation within the detector model, where $n < N$, such that the track measurement model f would have a dependence on those n parameters for the corresponding measurement. Then for each measurement, the partial derivatives

$$\left(\frac{\partial f}{\partial \mathbf{p}}\right)_i, \quad (5.25)$$

for an individual measurement i on the track and the track measurement model f , which have the dimensions of N , for N alignment parameters \mathbf{p} , will have n non-zero values and $N - n$ zeroes. Each alignment parameter in \mathbf{p} has an assigned, unique integer label denoted ℓ .

Therefore, only n values need to be stored per track measurement in addition to n integer labels ℓ , which uniquely identify what alignment parameter the stored global partial derivative was calculated with respect to.

5.4 Tracker degrees of freedom

To perform any alignment, the alignment parameters, or degrees of freedom, must be defined. In the case of the Mu2e Tracker, the construction of which is described in section 3.4.4, translational and rotational degrees of freedom are introduced for the Tracker (1 total), Planes (36 total) and Panels (216 total). The Planes and Panels are treated as rigid-body objects that can be translated in x, y, z , and rotated about the x, y, z -axes. For 3 shifts and 3 rotation angles assigned to each detector component, there are 1516 alignment

degrees of freedom total. Omitting the Tracker and Panel degrees of freedom, there are 216 Plane alignment degrees of freedom in total.

The ‘alignment’ of a vector \mathbf{v} can be described by the function

$$T(\mathbf{v}) = R_x(\alpha)R_y(\beta)R_z(\gamma)\mathbf{v} + \mathbf{u}, \quad (5.26)$$

with

$$\begin{aligned} R_x(\alpha) &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix}, \\ R_y(\beta) &= \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix}, \quad \dots \end{aligned} \quad (5.27)$$

where \mathbf{v} is rotated according to the rotation matrices R_x, R_y, R_z and their angles α, β, γ , and \mathbf{u} describes a translation in x, y, z .

A ‘nominal’ geometry describing the Mu2e Tracker holds the x, y, z Tracker coordinates of all 20,736 straw wire midpoints, and unit direction vectors describing the orientation of each straw.

The ‘aligned’ version of the Tracker is constructed using a series of nested transforms, as implemented in the Offline software at [49] according to the procedure outlined in [61]. The alignment process for an individual straw begins in a local unaligned panel frame of reference, with a straw wire direction unit vector and midpoint position vector pointing from the panel centre (also the origin) to the straw wire midpoint. The series of transforms applied to these vector quantities result in a straw midpoint position vector and direction vector correctly expressed in the aligned Tracker system (or detector system.)

For each straw, a vector pointing from the center of the panel holding the straw, $\mathbf{c}_{\text{panel}}$, to the straw midpoint, $\mathbf{c}_{\text{straw}}$, is calculated using the ‘nominal’ Tracker geometry

$$\begin{aligned} \mathbf{v}_{\text{panel}} &= (\mathbf{c}_{\text{straw}}^\perp - \mathbf{c}_{\text{panel}}^\perp, z, 0), \\ z &= (\mathbf{c}_{\text{straw}} - \mathbf{c}_{\text{panel}})_z, \end{aligned} \quad (5.28)$$

where \mathbf{v}^\perp of a vector \mathbf{v} is the transverse component (a scalar)

$$\mathbf{v}^\perp = \sqrt{v_x^2 + v_y^2}, \quad (5.29)$$

and the vector $\mathbf{v}_{\text{panel}}$ can now be transformed according to

$$\mathbf{p}_{\text{panel}} = (\mathbf{u}, \alpha, \beta, \gamma), \quad (5.30)$$

which are the alignment constants for the corresponding panel, such that

$$\begin{aligned} \mathbf{v}'_{\text{panel}} &= T(\mathbf{v}; \mathbf{p}_{\text{panel}}), \\ \mathbf{v}_{\text{plane}} &= T(\mathbf{v}'_{\text{panel}}; \text{panel to plane}), \end{aligned} \quad (5.31)$$

with $\mathbf{v}'_{\text{panel}}$ now describing the straw position vector in the ‘aligned’ panel frame of reference. This aligned quantity is now transformed to $\mathbf{v}_{\text{plane}}$, which is the vector pointing from the plane centre to the straw midpoint, in the unaligned plane frame of reference. Effectively with this transform the panel has been ‘placed’ inside an unaligned plane. The straw midpoint vector is now rotated and shifted with the plane’s alignment constants, and then transformed to the unaligned Tracker reference frame

$$\begin{aligned} \mathbf{v}'_{\text{plane}} &= T(\mathbf{v}_{\text{plane}}; \mathbf{p}_{\text{plane}}), \\ \mathbf{v}_{\text{tracker}} &= T(\mathbf{v}'_{\text{plane}}; \text{plane to tracker}). \end{aligned} \quad (5.32)$$

The resulting vector $\mathbf{v}_{\text{tracker}}$ is transformed again with T using the alignment constants $\mathbf{p}_{\text{tracker}}$ which yields the final ‘aligned’ straw result $\mathbf{v}'_{\text{tracker}}$, the ‘aligned’ straw midpoint starting from the Tracker coordinate system origin.

Now the straw ‘direction’ or orientation vector is calculated. The same procedure is repeated as described above for each straw direction unit vector starting at

$$\hat{\mathbf{d}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad (5.33)$$

and a modified form of the function T , T_r is used such that

$$T_r(\mathbf{v}) = R_x(\alpha)R_y(\beta)R_z(\gamma)\mathbf{v}, \quad (5.34)$$

which is a rotation-only transformation of a vector \mathbf{v} as shown, such that no translations are applied over the series of nested transforms applied to $\hat{\mathbf{d}}$.

The alignment procedure is carried out for all straws, and the results stored in an ‘aligned’ Tracker geometry description.

The alignment constants used by each alignment transformation are stored in a Conditions Database (the ProditionsService) [61], and the Alignment procedure as explained above is carried out in the TrackerGeom package by the AlignedTrackerMaker module.

5.5 Weak modes and constraints

In a Millepede global fit, some or all alignment degrees of freedom, as they are defined in section 5.4, can vary according to ‘weak modes’ which may cause the fit to diverge, or struggle to converge to acceptable values. Weak modes can present in various ways, dependent on which constraints are in place, the track sample size, the number of tracks per alignment parameter, the track model definition, and alignment parameter definitions. In the case of the Mu2e Tracker, running Millepede with no constraints or fixed parameters results in a diverging global fit, with the alignment constants returned as large positive or negative values.

Without appropriate constraints it is still possible for track residuals to be minimised whilst allowing the alignment parameters and track parameters to vary undesirably. In this case, for a straight-line track model parameterised by a straight line gradient and intercept, a weak mode contributing to the divergence of the global fit could be described as a translation of the entire Tracker and all track intercepts in any direction by a distance of any magnitude. Without constraining to sensible values the translation degrees of freedom on the entire Tracker and some or all planes, the global fit will diverge. Panels should also be constrained according to the geometry of the Plane they are housed within.

Another weak mode found in this case is a ‘stretch’ or ‘squeeze’ of all planes in z . This weak mode has the effect of either bringing all planes further from (or closer to) the Tracker center, and is present even when enforcing a zero-average translation in x, y, z

$$\sum_{i \in \text{planes}} \mathbf{u}_i = \mathbf{0}, \quad (5.35)$$

where \mathbf{u}_i is the vector representing the translation applied to the corresponding plane i from its ‘nominal’ geometry position.

The lack of appropriate constraints to prevent this behaviour results in the effect manifesting as shown in Figure 5.4. This particular effect can be damaging due to the magnitude of the resulting misalignments after the global fit, even when the alignment constants are initially set to perfect alignment. This indicates that a global fit will fail to converge (to acceptable alignment) within a sensible number of iterations until fit constraints are imposed to suppress the effect.

Constraints can be applied to the global fit parameters in a number of ways, either by

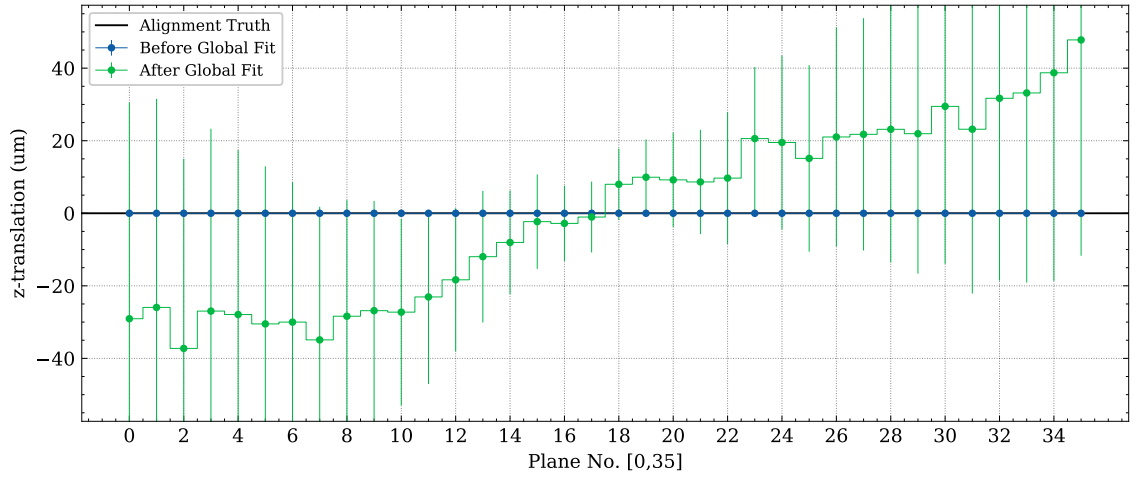


Figure 5.4: Alignment constants (z -translations are shown) before and after as determined by Millepede, where the detector is perfectly aligned and the initial alignment parameter values are already optimal. A zero-average translation is enforced on all Planes, and Panel degrees of freedom are fixed. The track sample included 45,724 cosmic tracks reconstructed at perfect alignment. The desired behaviour (in order to minimise the objective function F) is no change in the alignment constants i.e. minimal change between the blue and green data points indicating convergence behaviour. However, the pattern indicates the presence of a weak mode (stretch/squeeze) of the Planes in z which is hindering convergence of the global fit, even when given perfect initial values. Due to the highly correlated errors, the error bars in this case do not represent those of a good fit with independent statistical errors.

fixing them to ‘best-guess’ values, or by introducing constraint equations. Either method can be configured in Millepede, and may be informed by survey measurements in their application.

5.5.1 Constraint equations and parameter fixing

Millepede supports multiple instances of global fit constraints described by a linear combination of chosen alignment parameters fixed to a constant value C (usually zero)

$$\sum_{i \in \Omega} a_i p_i = C, \quad (5.36)$$

or alternatively a measurement m with associated error σ_m

$$\sum_{i \in \Omega} a_i p_i = m \pm \sigma_m, \quad (5.37)$$

where Ω represents the alignment parameters to be constrained, a_i the constant associated with parameter i , and p_i the i^{th} parameter [58] [59].

In addition to constraint equations, parameters can be variable (the default), or fixed at a constant value for the entire global fit. This feature is also useful in that quantities which are not alignment parameters can be introduced (such as detector component coordinates) to build constraint equations.

Chapter 6

Alignment Performance Studies

To test the performance of any candidate alignment procedure, Monte Carlo simulation studies are necessary and can be performed well in advance of data-taking and commissioning operations. Cosmic ray muons are a useful starting-point for an alignment and are available regardless of beam status.

For this study, a sample of MC cosmic digi events (DS-cosmic-nofield) are used, which were produced in a magnet-off scenario (no magnetic field). The dataset was created in the Mu2e MDC2018 production run, and adapted for compatibility with the latest software version, part of the MDC2020Dev collection of MC datasets [62].

The Tracker is misaligned by the same process described in Section 5.2, and tracks are reconstructed using the algorithm described in Section 4.4.1. For each alignment iteration: a track reconstruction job is run, a dataset is produced containing information about the collected alignment track sample, and a Millepede global fit is run. This process is repeated until the alignment is deemed to have converged or otherwise.

6.1 Selection criteria and dataset creation

Due to computing resources, constraints, and development requirements, a sample of tracks was preselected from the DS-cosmic-nofield dataset. In the first pass, cuts were first required to pass given that the Tracker was at perfect alignment i.e. no misalignments have been applied, so the tracks are reconstructed here in a perfectly aligned geometry. This was necessary to reduce the number of tracks with high levels of multiple scattering (e.g. low momentum cosmic muons), or generally tracks with low quality measurements

such as those with low numbers of hits and a low number of planes traversed. Such tracks failing selection in a perfect alignment pass are unlikely to be accepted in a misaligned pass. In the second pass, where the Tracker now has deliberately inserted misalignments, the cuts are loosened to account for those introduced misalignments and to reduce selection biases. The resulting ‘digi’ events passing preselection were stored in a new dataset DS-cosmic-nofield-alignselect in the MDC2020Dev collection, and the reconstruction data were discarded [63]. The reconstruction data were irrelevant determining whether the preselection cuts passed or failed for an event since the reconstruction would need to be re-evaluated when changing the alignment constants. Discarding the reconstruction products also saved a significant amount of disk space.

Reconstructed cosmic tracks were required to have at least 10 straw hits, and the number of planes traversed by a track was required to be at least 3 in the first pass, and 4 in the second pass. To suppress tracks with high multiple scattering (M.S.) effects, tracks with any residual exceeding 10 ns in magnitude were excluded from the alignment sample in the first pass [64]. This cut is loosened to 20 ns in the second pass.

No truth Monte-Carlo quantities, which are quantities that should be known only at the time of event generation, other than the truth MC alignment constants were used to select an alignment track sample. However, in a real alignment scenario it will be necessary to determine selection criteria without this information. Any observables that can assist in excluding tracks with high multiple scattering will assist global fit convergence, which may otherwise erroneously capture multiple scattering effects in the alignment parameters.

The Millepede input dataset, as summarised in Table 5.2, is generated from the selected track sample. A number of quantities must be generated, which are calculated in terms of the cosmic track model (see Section 4.4.1) and detector model (now with alignment degrees of freedom introduced.)

It is now possible to discuss the partial derivatives required by Millepede, calculated for each hit

$$\begin{aligned}\frac{\partial z_i}{\partial \mathbf{x}} &= \frac{\partial}{\partial \mathbf{x}} (D_T(r) + T_0 + t_{\text{traj}} + t_{\text{offset}}), \\ \frac{\partial z_i}{\partial \boldsymbol{\alpha}} &= \frac{\partial}{\partial \boldsymbol{\alpha}} (D_T(r) + T_0 + t_{\text{traj}} + t_{\text{offset}}),\end{aligned}\tag{6.1}$$

where the remaining terms to be differentiated are those with a dependence on the track parameters \mathbf{x} or the alignment parameters $\boldsymbol{\alpha}$ corresponding to the Plane holding the

straw. The wire midpoint \boldsymbol{w} and direction $\hat{\boldsymbol{r}}_w$ (as defined in Section 4.4.1) depend on the alignment parameters $\boldsymbol{\alpha}$, and are used to calculate \boldsymbol{p}_0 , \boldsymbol{p}_1 and the DOCA r . The partial derivatives for alignment parameters corresponding to other objects are zero, since the residual has no dependence. The derivatives are calculated numerically using the symmetric difference quotient,

$$\frac{\partial f}{\partial x} = \frac{f(x+h) - f(x-h)}{2h}, \quad (6.2)$$

for each track and alignment parameter, where h is the step size. An attempt was made to differentiate the time residual function to obtain an analytical derivative function, however this was scrapped due to the burden of ensuring that the separate program source code written to derive the analytical derivatives are consistent with definitions in the track fitting source code, which evolve as track reconstruction developments continue, and since D2T and t_{offset} both rely on numerical methods and calibration data available at runtime. In future developments however, analytical derivatives should be used if possible for the speed benefits, and lower sensitivity to numerical inaccuracy.

For each track, a total of 13 double-precision numbers per hit are stored in the dataset. This corresponds to a minimum of 130 8-byte doubles per track written to file given that at least 10 hits are required.

The global fit constraints were applied by fixing two planes (plane 5, and plane 30) to truth alignment values (perfectly aligned) for all translation shifts and rotation angles.

6.2 Plane translation only

The first alignment test was carried out by only considering Plane translation degrees of freedom, 102 in total (accounting for constraints.) A perfect alignment is achieved if all alignment constants are zero, i.e. zero misalignment from nominal Tracker plane positions.

The results of the alignment test are shown in Figure 6.1. Convergence is reached in 3 global fit iterations, with all alignment constants within 25 μm of perfect alignment.

To provide an understanding of the scale of initial misalignments, translations of the planes from perfect alignment positions were up to 4 mm, or up to 4000 μm .

The first iteration struggles to converge close to perfect alignment, returning alignment constants within 200 μm of perfect alignment. This is caused due to reduced track

acceptance under the selection cuts, caused by initial misalignments. However, by the second iteration the global fit input track sample is almost doubled due to the improved alignment constants from the first iteration. The fit performance in this iteration and subsequent iterations is improved considerably. By the third iteration, no major improvement is observed in the alignment constants nor input track sample size, and so the alignment is deemed to have converged.

This allows for the alignment study to be extended to involve rotational degrees of freedom.

6.3 Plane translation and rotation

The results of the alignment study, now with 204 degrees of freedom (taking into account that 12 are fixed) are shown in Figures 6.2 and 6.3. Once again, three iterations were needed before convergence was reached which consisted of 3 global fits and 3 track reconstruction jobs. Following the result of the previous test, convergence was judged by how much the track sample had increased from the previous iteration, and whether any noticeable improvement was seen in the alignment constants relative to the known truth values. Some mis-alignment is still observed after the 3rd global fit in the low bins of Figure 6.3(a), which could be improved by introducing further constraints on those degrees of freedom. Particularly, once it is possible to test the application of survey-based constraints rather than fixing degrees of freedom for only two of 36 total planes to their truth values, better results may be observed.

Once again, as the alignment constants approached perfect alignment, track reconstruction efficiency increased as more tracks were passing selection cuts due to reduced detector misalignments. Owing to increased sample size, subsequent global fits are able to improve the solution until no more tracks were recovered between iterations. For this test, by the second global fit, the track sample had approximately doubled to 58,591 tracks from 28,870 in the first iteration. In the third iteration, the track sample increase was small (59,569 tracks), and no remarkable improvement in the alignment constants observed. This result demonstrates that tracks are recovered rapidly between iterations, and it is therefore recommended to run multiple alignment iterations given a fixed event sample until no more tracks can be recovered.

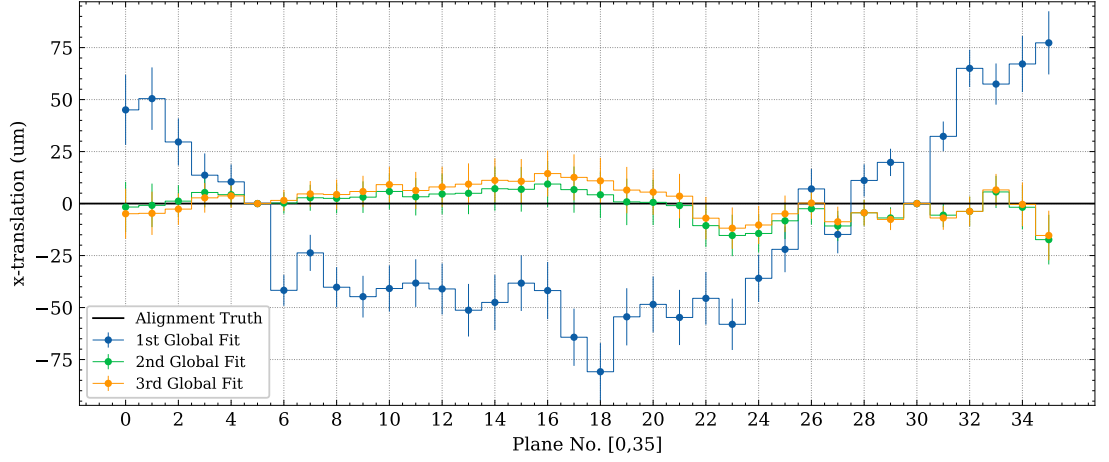
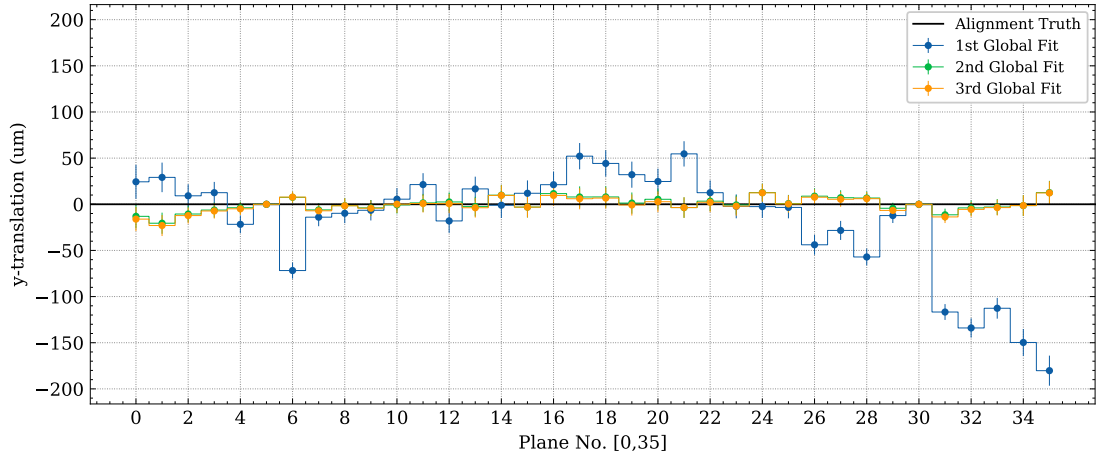
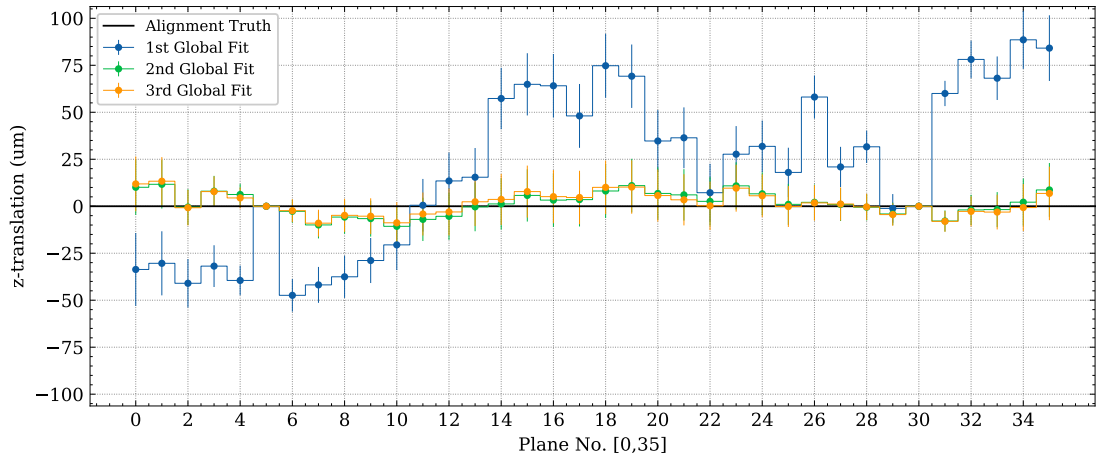
(a) Plane x -translation(b) Plane y -translation(c) Plane z -translation

Figure 6.1: Plane translation alignment (102 d.o.f. total) after 3 iterations. For this study, a perfect alignment is achieved if all alignment constants are zero. The alignment constants are initially set to non-zero values, therefore misaligning the Tracker. The alignment converged after 3 iterations, and the final alignment constants were found to be accurate to within $25\ \mu\text{m}$ for all Planes.

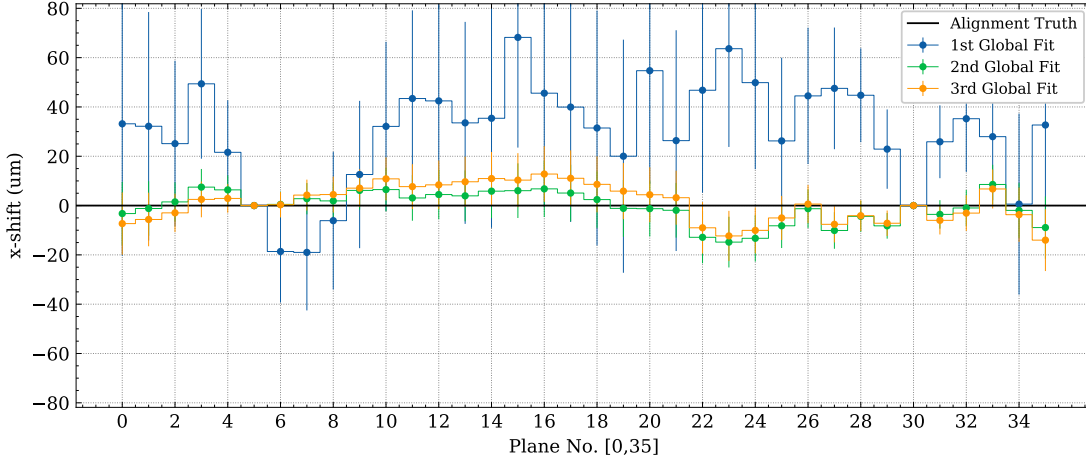
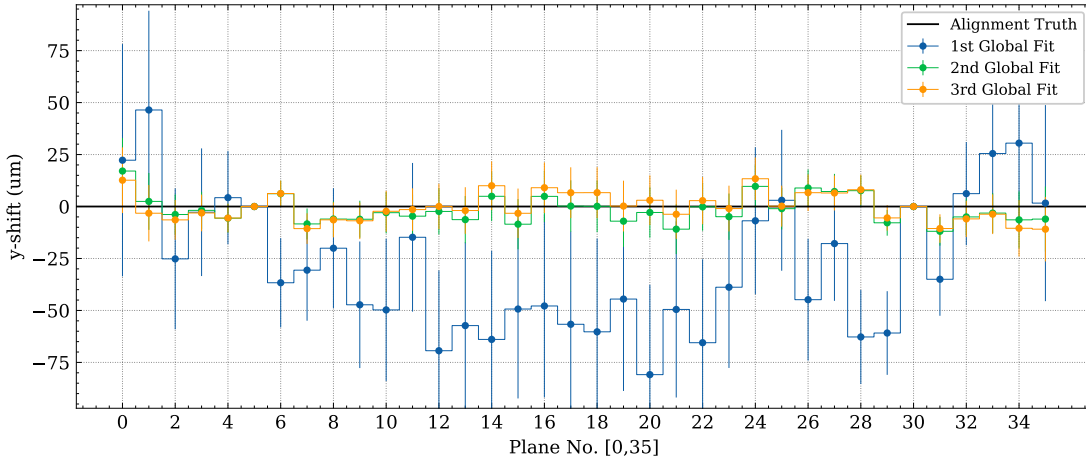
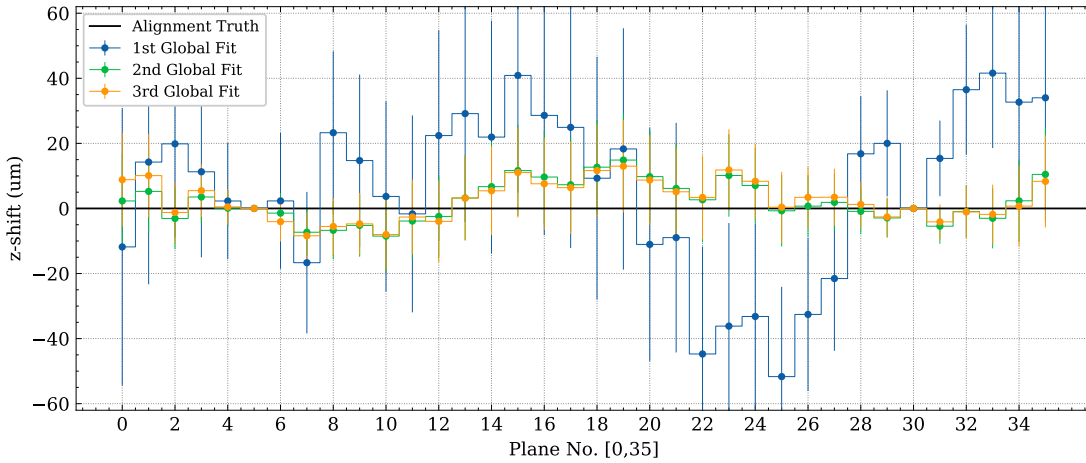
(a) Plane x -translation(b) Plane y -translation(c) Plane z -translation

Figure 6.2: Plane translation and rotational alignment (204 d.o.f. total. 102 d.o.f. shown.) Displayed above are the plane translation alignment constants, or x, y, z -translations for each plane, determined by each global fit iteration. Convergence was reached in 3 iterations, and the alignment constants were found to be accurate to within $20\ \mu\text{m}$ for all Planes.

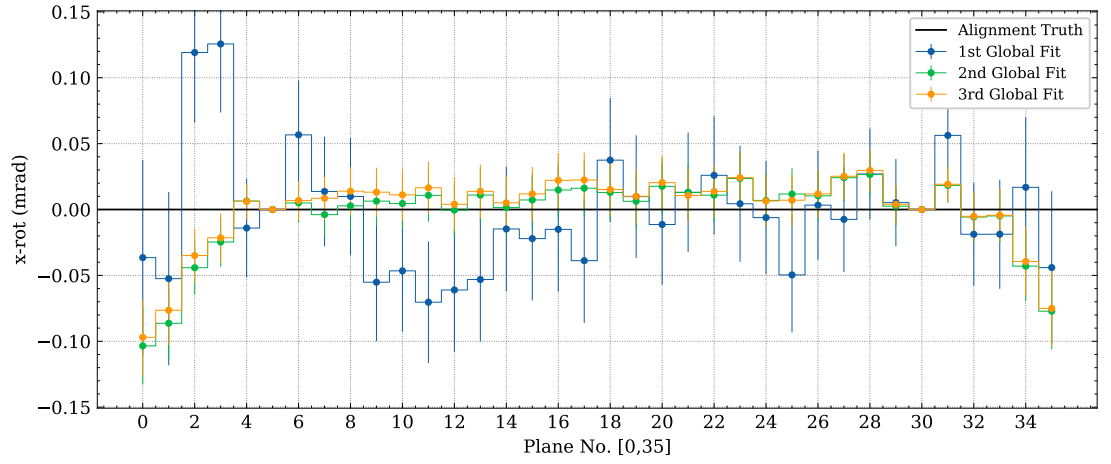
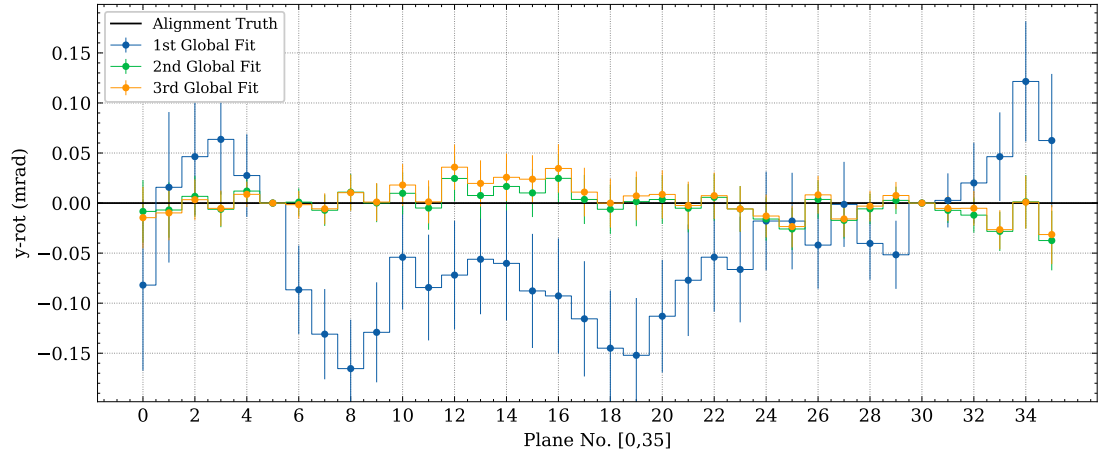
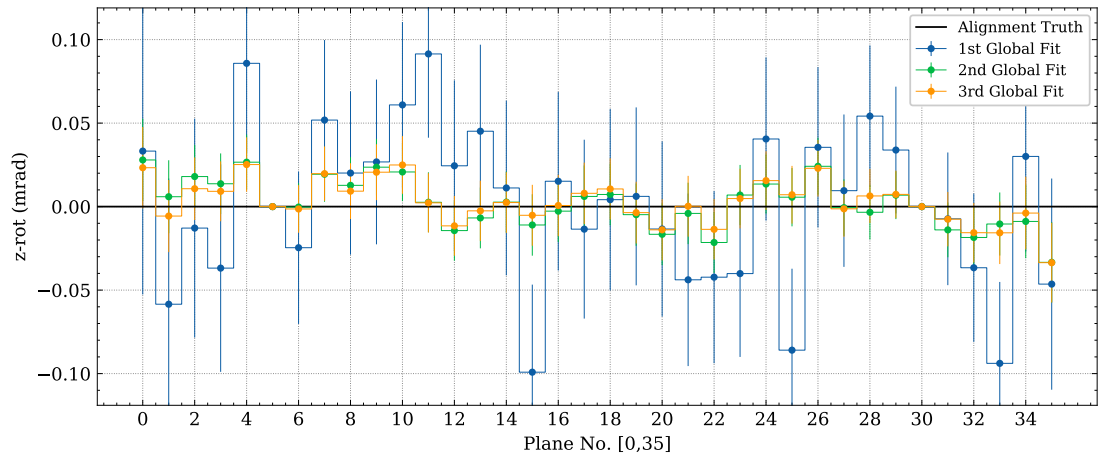
(a) Plane x rotation(b) Plane y rotation(c) Plane z rotation

Figure 6.3: Alignment constants (Plane rotation angles α, β, γ) for each plane, for 3 alignment iterations.

6.4 Alignment with Kalman-filter tracks

According to the Millepede-II Manual [59], tracks fit using a Kalman Filter method can be used to perform an alignment, by using the track parameters and numerically-determined derivatives as input. Millepede-II effectively performs a simultaneous fit of tracks and alignment parameters using a linear least squares method. The tracking model used to re-fit a track during the global alignment fit vs. during the experiment's track reconstruction process may not be the same, which may produce results where material effects and magnetic field effects may be neglected in the process of the alignment fit. Particularly this is an issue if multiple scattering and material effects are handled via process noise, and not as part of the track model [65]. If this is the case, it is important to ensure that an input track sample has negligible or low levels of multiple scattering, and no magnetic field, for example. If these effects are significant and cannot be ignored in this way, an alternative approach may be more appropriate.

A more desirable alternative is to implement a method where Kalman filtered tracks are fully supported, with all modelled effects correctly accounted for during the alignment global fit. The method discussed in [65] describes the mathematical steps required to derive the global covariance matrix of the set of Kalman filter state vectors (or piecewise track parameters), and how it may be used in the implementation of a minimum- χ^2 global alignment method. The covariance matrix derived here takes into account fully the correlations between the state vectors at different regions, or points on the track such as to account for effects modelled as process noise [65].

In contrast to the global χ^2 alignment method, the Kalman filter equations may also be used for the track-based alignment of detectors. A large system of linear equations does not need to be solved, as the method is iterative, and the estimate of the alignment constants are updated after each track is processed, and the depth of the parameter correlations considered between nodes may be adjusted [66].

Moving beyond Millepede-II to these alignment methods may require a lot of work in software implementation to produce a solution that is suitable for the Mu2e experiment at the level of alignment accuracy required. As has been seen for the no-field cosmic ray sample, if a sufficiently large high-quality (low multiple scattering, many hits) sample of cosmic rays can be selected, then alignment constants can be recovered to sufficiently high accuracy. However, an investigation into other track types should also be conducted,

particularly those available during the running of the experiment (DIOs, field-on cosmic rays).

Chapter 7

Conclusion

An alignment of the Mu2e Tracker was developed to correct the positions and orientations of misaligned planes shifted or rotated relative to their true positions and orientations in the software detector geometry.

A standalone software package was written to provide data monitoring components, plus the creation of analysable datasets. The software was designed to connect to the DAQ system and collect events from the Stopping Target Monitor in real-time during the test beam. Due to world events, the STM test beam did not go ahead, and the software remains in an incomplete state.

A Monte-Carlo simulated cosmic ray sample, where reconstructed tracks have a simple straight-line trajectory, was used to perform the track-based alignment using the Millepede-II software alignment package. Tracks with a large number of hits (at least 10) across at least four planes and low levels of multiple scattering were selected and written to a Millepede dataset. Two planes were fixed in the alignment to ensure the global fit was adequately constrained, and the global fit tested, solving for the alignment constants in two cases: plane translation misalignment only (102 d.o.f.), and plane translations plus rotations (204 d.o.f.)

In both tests, the global alignment fit had to be run at least three times in each case before no further improvement was seen in the results. The alignment constants returned were consistent with perfect alignment to within statistical error. Results improved as the track sample size increased.

These results provide a starting point for Tracker alignment with straight cosmic rays, which are available before data-taking begins. The early availability of these data

would allow for the Tracker to be aligned without an operational DS field or muon beam. The provided software tools will need to be further extended, however, to support more complex track parameterisations, and particularly Kalman Filter track fitting algorithms which are in everyday use at Mu2e. Kalman filtered tracks pose a difficulty due to the potential neglect of material effects that are often handled as process noise. A derivation of the global covariance matrix of the set of Kalman filter state vectors may be required, in order to account for these effects correctly. At this point, Millepede-II may not be suitable if it cannot support this form of the global covariance matrix, but in general, a track-based alignment achieved by solving (through other means) the minimum- χ^2 global linear least-squares equations is still a possibility and a potentially effective method. Other alignment algorithms are available, such as an iterative Kalman filter method and should also be investigated as appropriate.

At present, the only alignment degrees of freedom in play are translations of the planes and their rigid-body rotations about the x, y, z detector system axes. More investigations should be carried out to understand the complexity introduced by additional rigid-body degrees of freedom of the panels and the required constraints to avoid weak modes. At a far more precise level, misalignments could be present within the straws, due to displacement of the ends of straw wires, or sagging of the straw wires under gravity. Aligning at the straw level will be a challenge, as there are $\sim 20,000$ straws; therefore, this would introduce many degrees of freedom to the Tracker alignment. Additionally, there could be more investigation into the standard measurements available from a Tracker survey and how these should be used to constrain the alignment appropriately.

Now that a starting point for the Tracker alignment has been established for cosmic rays, there should be sufficient scope to explore other track types and methods. Extending and studying the existing Millepede-II system to understand further the limitations should also be considered. The `KinKal` track fit library now supports a wide range of trajectories needed by the experiment, and will soon be adopted in the Offline software for track reconstruction. Any future Mu2e track alignment studies should strongly consider using this library going forward. In summary, with regards to Tracker alignment, the Mu2e Offline software is in a good state, containing all the services, utilities, and frameworks (track fitting, conditions, configurable Tracker geometry, event processing) required to facilitate the further development, testing, and validation of an alignment algorithm.

In addition to the alignment project, a continuous integration workflow including a suite of bash shell scripts and bespoke GitHub interface was created for Mu2e Offline software development. The software has been used regularly and successfully to test proposed software changes over eight months (at the time of writing). The turnaround time for a test of proposed changes at a specific version is approximately ~ 20 minutes, and the results provided include log files in addition to code quality checks. It is hoped that new features and extensions to the suite of available tests can improve the effectiveness and efficiency of software development and operations during the lifetime of the experiment.

Finally, an STM test beam software package was developed to interface with the DAQ system chosen for the test beam and to provide monitoring and dataset tools. Due to world events, the software remains incomplete and should be revived and tailored to the needs of the test beam once it has been re-scheduled.

References

- [1] ATLAS Collaboration, ‘Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc,’ *Physics Letters B*, vol. 716, no. 1, pp. 1–29, 2012, ISSN: 0370-2693. DOI: <https://doi.org/10.1016/j.physletb.2012.08.020>.
- [2] CMS Collaboration, ‘Observation of a new boson at a mass of 125 gev with the cms experiment at the lhc,’ *Physics Letters B*, vol. 716, no. 1, pp. 30–61, 2012, ISSN: 0370-2693. DOI: <https://doi.org/10.1016/j.physletb.2012.08.021>.
- [3] SNO Collaboration, ‘Direct evidence for neutrino flavor transformation from neutral-current interactions in the sudbury neutrino observatory,’ *Physical Review Letters*, vol. 89, no. 1, p. 011 301, 13th Jun. 2002, ISSN: 0031-9007, 1079-7114. DOI: [10.1103/PhysRevLett.89.011301](https://doi.org/10.1103/PhysRevLett.89.011301). arXiv: [nuc1-ex/0204008](https://arxiv.org/abs/nuc1-ex/0204008).
- [4] SNO Collaboration, ‘Measurement of the rate of $\nu_e + d \rightarrow p + p + e$ interactions produced by b 8 solar neutrinos at the sudbury neutrino observatory,’ *Physical Review Letters*, vol. 87, no. 7, p. 071 301, 25th Jul. 2001, ISSN: 0031-9007, 1079-7114. DOI: [10.1103/PhysRevLett.87.071301](https://doi.org/10.1103/PhysRevLett.87.071301).
- [5] The Super-Kamiokande Collaboration and Y. Fukuda et al, ‘Evidence for oscillation of atmospheric neutrinos,’ *Physical Review Letters*, vol. 81, no. 8, pp. 1562–1567, 24th Aug. 1998, ISSN: 0031-9007, 1079-7114. DOI: [10.1103/PhysRevLett.81.1562](https://doi.org/10.1103/PhysRevLett.81.1562). arXiv: [hep-ex/9807003](https://arxiv.org/abs/hep-ex/9807003).
- [6] R. H. Bernstein and P. S. Cooper, ‘Charged lepton flavor violation: An experimenter’s guide,’ *Physics Reports*, vol. 532, no. 2, pp. 27–64, Nov. 2013, ISSN: 03701573. DOI: [10.1016/j.physrep.2013.07.002](https://doi.org/10.1016/j.physrep.2013.07.002). arXiv: [1307.5787](https://arxiv.org/abs/1307.5787).
- [7] R. H. Bernstein, ‘The mu2e experiment,’ *Frontiers in Physics*, vol. 7, p. 1, 28th Jan. 2019, ISSN: 2296-424X. DOI: [10.3389/fphy.2019.00001](https://doi.org/10.3389/fphy.2019.00001).
- [8] R. Davis, ‘A review of the homestake solar neutrino experiment,’ *Progress in Particle and Nuclear Physics*, vol. 32, pp. 13–32, 1st Jan. 1994, ISSN: 0146-6410. DOI: [10.1016/0146-6410\(94\)90004-3](https://doi.org/10.1016/0146-6410(94)90004-3).

- [9] B. R. Martin, *Particle physics*, eng, Fourth edition., ser. The Manchester physics series. Chichester, West Sussex, United Kingdom: John Wiley & Sons, ch. 2, ISBN: 9781118911907.
- [10] W. J. Marciano, T. Mori and J. M. Roney, ‘Charged lepton flavor violation experiments,’ *Annual Review of Nuclear and Particle Science*, vol. 58, no. 1, pp. 315–341, Nov. 2008, ISSN: 0163-8998, 1545-4134. DOI: 10.1146/annurev.nucl.58.110707.171126.
- [11] F. P. An, J. Z. Bai, A. B. Balantekin *et al.*, ‘Observation of electron-antineutrino disappearance at daya bay,’ *Physical Review Letters*, vol. 108, no. 17, p. 171 803, 23rd Apr. 2012, ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.108.171803. arXiv: 1203.1669.
- [12] A. de Gouvêa and N. Saoulidou, ‘Fermilab’s intensity frontier,’ *Annual Review of Nuclear and Particle Science*, vol. 60, no. 1, pp. 513–538, 23rd Nov. 2010, ISSN: 0163-8998, 1545-4134. DOI: 10.1146/annurev-nucl-100809-131949.
- [13] J. Woithe, G. J. Wiener and F. F. Van der Veken, ‘Let’s have a coffee with the standard model of particle physics!’ *Physics Education*, vol. 52, no. 3, p. 034 001, May 2017, ISSN: 0031-9120, 1361-6552. DOI: 10.1088/1361-6552/aa5b25.
- [14] M. Aker, K. Altenmüller, M. Arenz *et al.*, ‘An improved upper limit on the neutrino mass from a direct kinematic method by KATRIN,’ *Physical Review Letters*, vol. 123, no. 22, p. 221 802, 25th Nov. 2019, ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.123.221802. arXiv: 1909.06048.
- [15] V. Barger, D. Marfatia and K. Whisnant, ‘Neutrino mixing and oscillations,’ in *The Physics of Neutrinos*. Princeton University Press, 2012, pp. 33–44, ISBN: 9780691128535.
- [16] I. Esteban, M. Gonzalez-Garcia, M. Maltoni, T. Schwetz and A. Zhou, ‘The fate of hints: updated global analysis of three-flavor neutrino oscillations,’ Jul. 2020. arXiv: 2007.14792 [hep-ph].
- [17] A. de Gouvêa and P. Vogel, ‘Lepton flavor and number conservation, and physics beyond the standard model,’ *Progress in Particle and Nuclear Physics*, vol. 71, pp. 75–92, Jul. 2013, ISSN: 01466410. DOI: 10.1016/j.pnpnp.2013.03.006.
- [18] B. R. Martin, *Particle physics*, eng, Fourth edition., ser. The Manchester physics series. Chichester, West Sussex, United Kingdom: John Wiley & Sons, ch. 12, ISBN: 9781118911907.
- [19] Y. Kuno and Y. Okada, ‘Muon decay and physics beyond the standard model,’ *Reviews of Modern Physics*, vol. 73, no. 1, pp. 151–202, 12th Jan. 2001, ISSN: 0034-6861, 1539-0756. DOI: 10.1103/RevModPhys.73.151.

- [20] L. Calibbi and G. Signorelli, ‘Charged lepton flavour violation: An experimental and theoretical introduction,’ *La Rivista del Nuovo Cimento*, vol. 41, no. 2, pp. 71–174, 31st Jan. 2018, ISSN: 0393-697X. DOI: 10.1393/ncr/i2018-10144-0. arXiv: 1709.00294.
- [21] C. L. Cowan, F. Reines, F. B. Harrison, H. W. Kruse and A. D. McGuire, ‘Detection of the free neutrino: A confirmation,’ *Science*, vol. 124, no. 3212, pp. 103–104, 1956, ISSN: 0036-8075. DOI: 10.1126/science.124.3212.103. eprint: <https://science.sciencemag.org/content/124/3212/103.full.pdf>.
- [22] E. P. Hincks and B. Pontecorvo, ‘Search for gamma-radiation in the 2.2-microsecond meson decay process,’ *Physical Review*, vol. 73, no. 3, pp. 257–258, 1st Feb. 1948, ISSN: 0031-899X. DOI: 10.1103/PhysRev.73.257.
- [23] T. Mori, ‘Final results of the MEG experiment,’ *Il Nuovo Cimento C*, vol. 39, no. 4, pp. 1–8, 30th Dec. 2016, ISSN: 0390-5551. DOI: 10.1393/ncc/i2016-16325-7. arXiv: 1606.08168.
- [24] S. Mihara, ‘Status of MEG experiment at PSI,’ *PoS*, vol. NuFact2019, p. 081, 2020. DOI: 10.22323/1.369.0081.
- [25] U. Bellgardt *et al.*, ‘Search for the Decay $\mu^+ \rightarrow e^+ e^+ e^-$,’ *Nucl. Phys. B*, vol. 299, pp. 1–6, 1988. DOI: 10.1016/0550-3213(88)90462-2.
- [26] K. Arndt *et al.*, ‘Technical design of the phase I Mu3e experiment,’ Sep. 2020. arXiv: 2009.11690 [physics.ins-det].
- [27] A. Czarnecki, X. G. i. Tormo and W. J. Marciano, ‘Muon decay in orbit: Spectrum of high-energy electrons,’ *Physical Review D*, vol. 84, no. 1, p. 013 006, 14th Jul. 2011, ISSN: 1550-7998, 1550-2368. DOI: 10.1103/PhysRevD.84.013006. arXiv: 1106.4756.
- [28] L. Bartoszek *et al.*, ‘Mu2e Technical Design Report,’ Oct. 2014. DOI: 10.2172/1172555. arXiv: 1501.05241 [physics.ins-det].
- [29] The SINDRUM II Collaboration, W. Bertl, R. Engfer *et al.*, ‘A search for mu-e conversion in muonic gold,’ *The European Physical Journal C*, vol. 47, no. 2, pp. 337–346, Aug. 2006, ISSN: 1434-6044, 1434-6052. DOI: 10.1140/epjc/s2006-02582-x.
- [30] S. J. Werkema, ‘Cd3c review accelerator plenary talk,’ Mu2e Experiment, Internal Note Mu2e-doc-7530-v6, 14th Jun. 2016.
- [31] R. E. Ray and J. Whitmore, ‘Project overview plenary talk for cd-3c review,’ Mu2e Experiment, Internal Note Mu2e-doc-7084-v23, 14th Jun. 2016.
- [32] M. J. Lamm, ‘Mu2e solenoid status for october 2018 ipr,’ Mu2e Experiment, Internal Note Mu2e-doc-20562-v5, 1st Oct. 2018.

- [33] N. Atanov, V. Baranov, J. Budagov *et al.*, ‘Design and status of the mu2e crystal calorimeter,’ *IEEE Transactions on Nuclear Science*, vol. 65, no. 8, pp. 2073–2080, Aug. 2018, ISSN: 0018-9499, 1558-1578. DOI: 10.1109/TNS.2018.2790702.
- [34] E. C. Dukes, ‘Crv layout perspective drawings,’ Mu2e Experiment, Internal Note Mu2e-doc-26466-v1, 4th Jun. 2019.
- [35] A. Mukherjee, ‘Doe tracker l2 talk (ipr 2017),’ Mu2e Experiment, Internal Note Mu2e-doc-11567-v3, 29th Jun. 2017.
- [36] R. Bernstein and A. Mukherjee, ‘Tracker requirements document,’ Mu2e Experiment, Internal Note Mu2e-doc-732-v22, 7th Apr. 2016.
- [37] R. Bonventre, ‘Momentum resolution plot,’ Mu2e Experiment, Internal Note Mu2e-doc-18138-v1, 18th Jun. 2018.
- [38] R. Kutschke, ‘Units and coordinate systems used by mu2e offline,’ Mu2e Experiment, Internal Note Mu2e-doc-1120-v3, 29th Jul. 2016.
- [39] R. Kutschke, Private Communication, 2020.
- [40] K. Biery, C. Green, J. Kowalkowski, M. Paterno and R. Rechenmacher, ‘artdaq: An event filtering framework for Fermilab experiments,’ in *18th Real-Time Conference*, 2012. DOI: 10.1109/RTC.2012.6418358.
- [41] C. Green, J. Kowalkowski, M. Paterno, M. Fischler, L. Garren and Q. Lu, ‘The Art Framework,’ *J. Phys. Conf. Ser.*, vol. 396, M. Ernst, D. Düllmann, O. Rind and T. Wong, Eds., p. 022 020, 2012. DOI: 10.1088/1742-6596/396/2/022020.
- [42] R. Rivera, ‘Online daq software status,’ Mu2e Experiment, Internal Note Mu2e-doc-26941-v1, 24th Jun. 2019.
- [43] D. N. Brown, E. A. Charles and D. A. Roberts, ‘The babar track fitting algorithm,’ in *Proceedings of the International Conference on Computing in High Energy and Nuclear Physics*, p. 2001.
- [44] P. Gartung, ‘Github and jenkins integration,’ Mu2e Experiment, Internal Note Mu2e-doc-29718-v3, 6th Nov. 2019.
- [45] Fermilab. (). ‘Fhicl - fermilab redmine,’ [Online]. Available: <https://cdcvns.fnal.gov/redmine/projects/fhicl> (visited on 14/01/2021).
- [46] W. E. Brown, C. Green, K. Knoepfel, J. Kowalkowski, M. Paterno and R. Putz, ‘Quick Start Guide for FHiCL 3: The Fermilab Hierarchical Configuration Language,’ Fermilab SCD/SSA/SSI, Reference Documentation, 13th Oct. 2011.

- [47] R. Putz, ‘Specification of the fermilab hierarchical configuration language,’ Fermilab, Reference Documentation, 21st Jun. 2012.
- [48] R. Culbertson, ‘Conditions database status and plans,’ Mu2e Experiment, Internal Note Mu2e-doc-23917-v2, 6th Feb. 2019.
- [49] Mu2e Collaboration, *Mu2e offline software*, version 09_07_00, 16th Jul. 2020.
- [50] M. Hatlo, F. James, P. Mato, L. Moneta, M. Winkler and A. Zsenei, ‘Developments of mathematical software libraries for the lhc experiments,’ *IEEE Transactions on Nuclear Science*, vol. 52, no. 6, pp. 2818–2822, 2005.
- [51] R. Bonventre, ‘Reconstructing cosmic tracks from the vertical panel runs,’ Mu2e Experiment, Internal Note Mu2e-doc-33914-v2, 18th Jun. 2020.
- [52] R. Bonventre, ‘Cosmic track fitter update,’ Mu2e Experiment, Internal Note Mu2e-doc-33162-v1, 13th May 2020.
- [53] F. James, ‘MINUIT Function Minimization and Error Analysis: Reference Manual Version 94.1,’ 1994.
- [54] R. Frühwirth, ‘Application of kalman filtering to track and vertex fitting,’ *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 262, no. 2, pp. 444–450, 1987, ISSN: 0168-9002. DOI: [https://doi.org/10.1016/0168-9002\(87\)90887-4](https://doi.org/10.1016/0168-9002(87)90887-4).
- [55] D. N. Brown, ‘Kinematic kalman filter track fit update,’ Mu2e Experiment, Internal Note Mu2e-doc-33022-v1, 29th Apr. 2020.
- [56] Plotly Technologies Inc. (2015). ‘Collaborative data science,’ [Online]. Available: <https://plot.ly> (visited on 29/09/2020).
- [57] D. N. Brown, ‘Tracker misalignment proposal,’ Mu2e Experiment, Internal Note Mu2e-doc-28722-v3, 29th Sep. 2019.
- [58] V. Blobel, G. Flucke and C. Kleinwort, *Millepede-II*, in collab. with A. Miller, version 04-07-01, 16th Jul. 2020.
- [59] V. Blobel and C. Kleinwort. (16th Jul. 2020). ‘Millepede II - draft manual.’ generated from software source code using Doxygen, [Online]. Available: https://www.desy.de/~kleinwrt/MP2/doc/html/draftman_page.html (visited on 01/08/2020).
- [60] V. Blobel and C. Kleinwort, ‘A New method for the high precision alignment of track detectors,’ in *Conference on Advanced Statistical Techniques in Particle Physics*, Jun. 2002. arXiv: hep-ex/0208021.

- [61] R. Culbertson, ‘Tracker alignment infrastructure,’ Mu2e Experiment, Internal Note Mu2e-doc-24870-v1, 12th Mar. 2019.
- [62] A. Edmonds, ‘Reprocessing mdc2018 digi files for mdc2020 development,’ Mu2e Experiment, Internal Note Mu2e-doc-30838-v2, 15th Jan. 2020.
- [63] R. H. O’Neil, ‘Cosmic no-field alignment selection,’ Mu2e Experiment, Internal Note Mu2e-doc-33308-v1, 20th May 2020.
- [64] R. Bonventre, Private Communication, 2020.
- [65] W. Hulsbergen, ‘The global covariance matrix of tracks fitted with a kalman filter and an application in detector alignment,’ *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 600, no. 2, pp. 471–477, Mar. 2009, ISSN: 01689002. DOI: 10.1016/j.nima.2008.11.094.
- [66] S. Blusk, O. Buchmüller, A. Jacholkowski, T. Ruf, J. Schieck and S. Viret, Eds., *Proceedings, first LHC Detector Alignment Workshop, CERN, Geneva, Switzerland, 4-6 September 2006*, CERN Yellow Reports: Conference Proceedings, Aug. 2007. DOI: 10.5170/CERN-2007-004.

Appendix A

Documentation

This section is designed in the hope that it will provide useful step-by-step instructions and adequate descriptions of the developed software interface between the Mu2e Offline and Millepede-II alignment software.

A.1 How to use the TrackerAlignment package

A.1.1 Setting up the software

You must first set up an up-to-date release of the Mu2e software as follows:

```
setup mu2e
git clone https://github.com/Mu2e/Offline
cd Offline
source setup.sh

# introduce the mu2ealign utility to the environment.
source TrackerAlignment/scripts/setup.sh
```

This will install mu2ealign within the shell environment. Wherever mu2ealign is run, it will behave according to the contents of the current working directory. This utility was created to try and automate some of steps and bookkeeping required, since there's a lot of inputs, outputs, translation steps involved.

A.1.2 Configuring an alignment

We can create a mu2ealign working directory like this:

```
mkdir some_directory && cd some_directory
mu2ealign new [alignment constants file]
```

which creates a number of files. It is important to tune these to ensure the input data (`sources.txt`), reconstruction, and alignment behaviour (`job.fcl`) is correctly configured. The working directory is constructed with bookkeeping in mind. It contains clearly named configuration files, log files, inputs, and outputs - everything needed to reproduce the results for that iteration.

The `job.fcl` configuration file is used for configuring the track reconstruction job. It can be submitted to the grid, or run interactively. The results should be linked somewhere in the current working directory (perhaps by using symbolic links.)

A.1.3 Acquiring tracks and running an alignment fit

A track collection job can be run interactively by running the following command

```
# the simplest thing to do is to run
mu2e -c job.fcl -S $input_dataset -n $n_events
```

Convenience routines are available. The input files in `sources.txt` can be split across multiple processes using the following:

```
# files in sources.txt are split across 4 processes
mu2ealign parallel 4
# run the 4 processes interactively
mu2ealign run
```

This set of commands will start the track collection job. After this has finished, it is time to merge the output (if running multiple jobs) and run the PEDE utility.

```
mu2ealign pede
```

which will produce the `millepede.res` and `alignconstants_out.txt` text files. One can also study on the terminal the PEDE output and ensure that the alignment fit ran as expected.

A.1.4 Grid-based workflow

Realistically speaking, many digi events will need to be processed, and interactive `mu2egpvm` instances will not have enough resources to carry out these jobs effectively. The previous steps can be repeated, but with `job.fcl` submitted to the grid using `generate_fcl`

and mu2eprodsys. The Mu2e Wiki should be consulted for the latest recommended grid workflow.

The `.bin.gz` files produced by successful grid jobs should then be moved out of out-stage to a safe place. A symlink to these `.bin.gz` files can be created in the alignment working directory, and referenced from the `mp-steer.txt` file to load the alignment track data. Symlinks are recommended here, since mu2e job output files may have very long filenames as standard, which PEDE may struggle to process correctly. One can then run the `mu2ealign pede` command to produce the Proditions tables for the next iteration.

This workflow has not been automated yet, and it may be convenient to extend `mu2ealign` to facilitate this.