PROCEEDINGS OF SCIENCE

Real-Time Track Reconstruction with FPGAs for MUonE

Michael McGinnis for the MUonE Collaboration[a,*]

[a] *Department of Physics and Astronomy, Northwestern University, 633 Clark St, Evanston, IL 60208*

E-mail: michaelmcginnis2026@u.northwestern.edu

Event selection is central to high-intensity particle physics experiments to allow DAQ systems to manage the corresponding data rates. This is true for the MUonE experiment, which will be composed of 40 tracking stations and targets to reconstruct elastic scatters of muons on atomic electrons. Since the signal process and some backgrounds can be distinguished using track information alone, online track reconstruction can result in lower data rates with higher purity. Online track-fitting and vertexing for event selection will be implemented directly on FPGAs, using High-Level Synthesis to convert C++ code into an HDL description. This contribution will focus on results from a test beam at the M2 beamline at CERN in 2023, subsequent improvements to the algorithm, and plans for a 2025 test beam in the same location. In 2023, the standalone reconstruction of tracks from high-rate muons was performed, and the algorithm showed good agreement with offline reconstruction. Later developments yielded significant timing and resource use improvements, which allow for single track fit results to be calculated above 50 MHz. This was achieved through pipelining the algorithm and implementing custom linear algebra functions which include basic assumptions on the fits. These results demonstrate the feasibility of FPGA-based online reconstruction for event selection for the MUonE experiment. In the 2025 test beam, a more general reconstruction algorithm will be demonstrated. This includes reconstructing events with higher occupancy, and implementations of candidate track selection and vertexing. Expected milestones and planned improvements to the more general algorithm will also be described. Finally, it will be shown that this algorithm and the assumptions made to improve performance can be generalized to applications beyond MUonE.

33rd International Workshop on Vertex Detectors (VERTEX2025)
25-29 August 2025
University of Tennessee, Knoxville, USA

[*] Speaker

PoS(VERTEX2025)010

1. Introduction

The MUonE experiment [1] aims to extract the leading-order hadronic vacuum polarisation contribution to the anomalous magnetic moment of the muon, an input with large systematic uncertainty to the Standard Model prediction of the muon magnetic moment, which has been precisely measured by the Muon g-2 experiments [2][3]. MUonE will extract this value by measuring the elastic scattering of muons on atomic electrons, requiring the reconstruction of tracks and vertices to calculate scattering angles. The experiment is proposed to use the M2 beam at CERN's north area, where several MUonE test beams have occurred. This beam line delivers an asynchronous muon beam at 160 GeV, with a maximum intensity of 2×10^8 muons per spill, corresponding to a peak muon rate of 50 MHz.

The MUonE tracker is composed of stations, one of which is shown in Fig. 1. The station comprises six silicon strip modules with a scattering target in front. Since there are no macroscopic electric or magnetic fields in MUonE, the tracks are straight lines, hence the X and Y projections of the tracks are independent. The tracking modules are mounted in pairs to measure perpendicular directions, and the middle pair is rotated by 45 degrees. The rotation requires hits to be combined to form a single virtual hit before rotating to the global frame. This allows for X and Y track projections to be coupled to form 3D tracks, as matching tracks will share hits in the rotated modules, also called the UV modules. So, the modules are oriented as: XY UV XY.

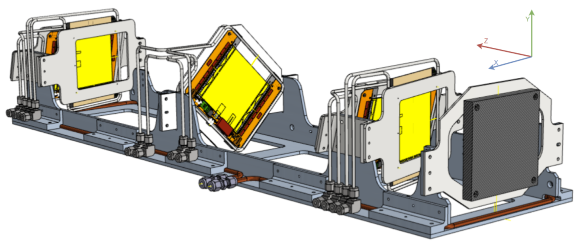

Figure 1: A CAD drawing of a single MUonE station. The 2S sensors are in yellow, and the middle modules are rotated by 45 degrees. The scattering target is in dark gray at the front of the station.

The tracker modules are the CMS two-strip (2S) tracker modules developed for the HL-LHC upgrade [4]. Each module is constructed with two strip sensors, with two columns of 1016 strips, each 5 cm long and 90 μm wide for a total active area of $10 \times 10\,\text{cm}^2$ per module. The readout ASICs allow for the readout of the module at 40 MHz, the bunch crossing rate at the LHC. For its back-end system, MUonE uses read out electronics developed by CMS for the HL-LHC, specifically the Serenity ATCA board [5]. It houses Field Programmable Gate Arrays (FPGAs) which are used for readout and configuring modules. These FPGAs will run the real-time track reconstruction outlined in these proceedings.

An estimate of data rates in the final version of MUonE is 600 Gb/s, with 16 bits per hit and 40 MHz readout dictated by the modules, and assuming 3 hits per module, 240 modules, with padding. Because of this, and the low cross-section of elastic scatters, it will be necessary to select events for long-term storage. So, real-time track and vertex reconstruction is being explored as part of this event selection. It will be implemented on an FPGA to reach rates similar to readout. Tracks and vertices are apt for event selection as they form the basis of the analysis, and even simple information can be used to distinguish between signal and some backgrounds. For example, a signal event has two outgoing tracks from a vertex, where the pair production background has three. More complex calculations of vertex position and reconstructed angles can further be used to select signal events for storage and reject backgrounds.

2. High-Level Synthesis

Typically, algorithms are developed to run on FPGAs in a hardware description language (HDL), which is then synthesized to a register transfer logic (RTL) description to be run on an FPGA. This becomes arduous for complex tasks like track and vertex fitting, so high-level synthesis (HLS), specifically AMD Vitis HLS [6] was used. This allows for the fast development of firmware by synthesizing C or C++ code into an IP block containing RTL. These blocks can then be placed in larger firmware designs. Vitis HLS also estimates performance and resource use, validates the software and firmware implementations of the block with a test bench, and provides libraries optimized for use with hardware. The Vitis BLAS library [7] was used in this project. Further, HLS pragmas afford greater control over the implementation, such as specifying resource use or enforcing pipelining, where the block continually accepts new inputs rather than waiting until the whole block finishes.

3. 2023 Results

An implementation of the track-fitting was tested live in the M2 beam in 2023 with a station without a target. Only events with a single hit in each module were selected to fit, both due to the latency of the fitting process and to simplify the algorithm. This only requires a single block to convert local module coordinates to global coordinates and blocks to fit the X and Y track projections. The outputs are the track parameters and parameter covariances. This implementation operated with a latency of about 1200 cycles at 320 MHz, corresponding to 3.75 μs. The design required 7.75% of the available resources when implemented on a Xilinx KU15P FPGA [8].

The algorithm was validated by comparing results between track-fitting done in software and those running on the FPGA. Figure 2 shows the differences in fit parameters between the two methods. The standard deviations of the distributions are well below relevant uncertainties, for example that of the intercept residual is four orders of magnitude smaller than the single hit resolution of a 2S module, 26 μm. Some small differences in the methods are expected, for instance due to fixed-precision variables. These distributions illustrate that the hardware implementation is consistent with the software implementation.

Testing of the FPGA-based track-fitting in the M2 beam allowed the real-time reconstruction of tracks and beam parameters. The beam spot position and beam direction are visualized in Fig. 3 by plotting the track intercepts and the track angles, respectively.

3.1 Further Development

Following the 2023 beam test, the track-fitting algorithm was improved in two main ways. First, pipelining was used to allow for parallel execution. When pipelining is enforced, multiple calculations can be done in parallel as each step continuously accepts new inputs. This decreases

PoS(VERTEX2025)010

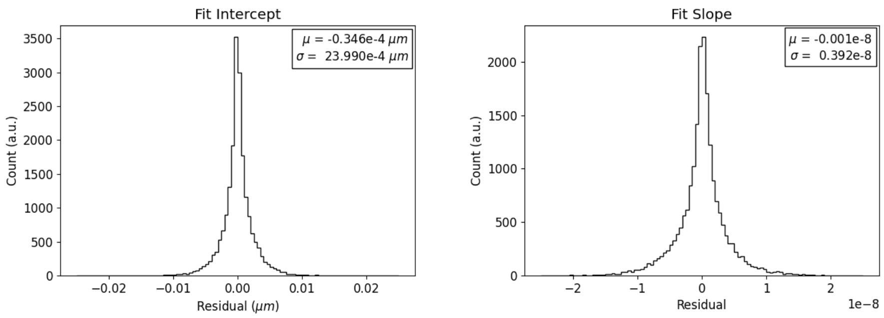

Figure 2: Distributions showing the differences between the intercepts (left) and slopes (right) from track-fitting running on an FPGA and in software. The fit parameters for X and Y fits are combined as they are independent. Notably, the distribution standard deviations are much smaller than relevant uncertainty scales, showing good agreement.

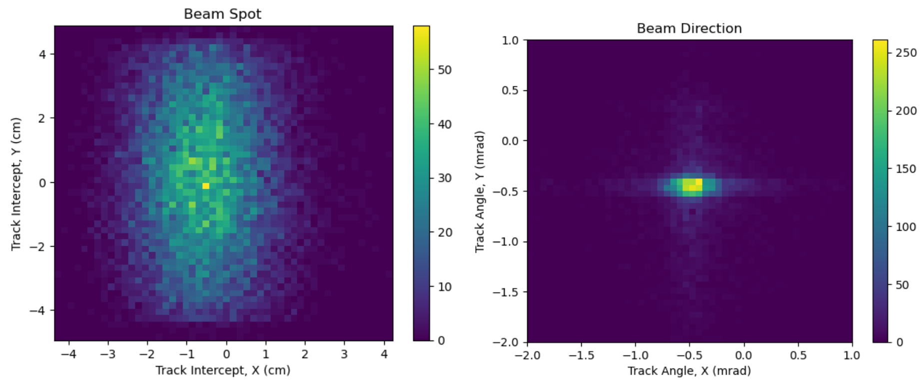

Figure 3: Plots of beam characteristics derived from real-time track-fitting in beam. The beam spot (left) is found by plotting the fit intercepts, while the beam direction (right) is calculated from the fit slope.

the time required between successive fits, called the initiation interval (II). Pipelining an algorithm typically requires more resources, and does not itself change the latency. However, the latency can decrease due to changes to the implementation to pipeline more effectively.

The second development was to replace HLS library functions with custom, optimized versions which include assumptions about inputs which follow from the experimental setup and having straight tracks. The track-fitting algorithm includes multiple steps of linear algebra, including QR factorization, matrix inversion, and matrix multiplication. Some values are always zero. These were encoded in the steps, simplifying the algorithm. Additionally, one column of the matrix to be QR factorized is always a column of ones. It was thus implemented using the Gram-Schmidt procedure, in which an orthonormal basis spanning the columns of the matrix is found. The first basis vector is chosen to be the normalized first column, and its values added as constants in the algorithm. Further, this column of ones simplifies other operations, like multiplication.

The results of these two changes are shown in Table 1. The first two rows show timing information, the latency and II, while the following four show the usage for some FPGA resources, LUTs, flip-flops, DSPs, and block-RAMs. The first column shows the values of the version which was tested at the M2 beam in 2023, and the second column show the values of the pipelined version with custom functions. The improved version uses fewer resources, while having a much lower latency and II. Most notably, an II of 6 cycles at a 320 MHz clock corresponds to fit calculated at 53.33 MHz, which is in line with the peak muon rate of 50 MHz at the M2 beam. These results, combined with the accuracy of the hardware implementation, shows that real-time track reconstruction is feasible for the MUonE experiment.

	HLS	Custom + Pipelined
Latency (cycles [μs])	1369 [4.28]	431 [1.35]
II (cycles [μs])	1372 [4.28]	6 [0.02]
LUTs (percent)	7.74	4.57
Flip-Flops (percent)	7.02	4.42
DSP (percent)	9.30	6.66
BRAM (percent)	4.47	2.24

Table 1: The timing information (first two rows) and some resource usage (bottom four rows) for two different versions of the real-time track-fitting. The first column corresponds to the version testing in beam in 2023, which used the HLS linear algebra library functions and no pipelining. The second column shows the results of enforcing pipelining and using custom linear algebra functions. Notably, an II of 6 cycles achieved by this version with a 320 MHz clock corresponds to track-fitting at 53.33 MHz, which is in line with the peak muon rate of 50 MHz at the CERN M2 beam.

4. Online Reconstruction in 2025

The online reconstruction was expanded beyond fitting single-track events during a 2025 test-beam. To simplify development and constrain resource use and latency, focus was placed on events with 2 hits per module in a station with a target, which is the minimum required to reconstruct a vertex. The aim of this test is to validate the full chain, and improvements such as enforcing pipelining are not included. The entire chain is illustrated in Fig. 4.

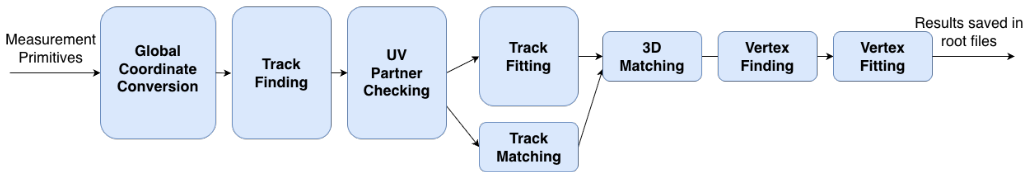

Figure 4: A block diagram showing an overview of the real-time reconstruction chain which will be tested in beam in 2025. With two hits per module, it is possible to reconstruct vertices as the end result.

The first step is again the conversion of local coordinates from the tracker modules to global coordinates. The positions in global coordinates are then passed to the track-finding block, which selects hits to form track candidates for fitting. These candidates are formed by selecting pairs of hits, one from the first and one from the last module, then projecting a line between them to the UV module. Then UV hits which are within a window about the projection are selected, and the three hits form a track candidate.

Track candidates are passed to the first of three blocks used to match X and Y track projections to form 3D tracks, the UV partner checking. This block verifies that each candidate shares a UV hit with a candidate in the other projection. Candidates which do not share a UV hit are discarded, saving time. The remaining candidates are passed in parallel to the fitting blocks as well as the the track matching block. The track matching block forms all possible combinations of X and Y projections being fit. Finally, the 3D matching block combines the fit results from the track-fitting block and the combinations from the track matching block to form 3D track objects which are used in vertex-finding.

The track-fitting occurs in parallel to track matching. Two blocks are used to fit the X and Y projections separately, each calculating the track parameters and parameter covariances. The version tested in 2025 includes the custom linear algebra functions, and is adapted to having more tracks per events, but pipelining is not enforced.

Once the tracks are fit and 3D tracks formed, vertex candidates are selected by the vertex-finding block. Due to time constraints, this was implemented as simply selecting pairs of 3D tracks which do not share hits.

Finally, the vertex candidates are fit, resulting in an estimated vertex position for each pair of 3D tracks from the vertex-finding block. This block relies on Newton's method, which is iterative. Since each iteration depends on the previous one, pipelining is limited and a balance between latency and accuracy will need to be struck.

4.1 Current Status for the 2025 Test Beam

Before testing the full online reconstruction in beam, each block needs to be implemented and validated individually in HLS. Then the blocks, connections between them, and control firmware simulated in VHDL. Next it is tested on an FPGA, injecting previously recorded data and comparing the results with those from software. As of writing, each of these steps has been completed, showing that the algorithm is accurate and the implementation works as expected in hardware. Efforts are underway to integrate the online reconstruction with the nominal DAQ to run live in beam. While operating live, the fit tracks and vertices are saved for comparison with the same events reconstructed offline.

4.2 Planned Improvements

Several developments are planned after testing the current implementation in beam. Of prime importance to improving performance and resource usage is the implementation of custom linear algebra functions and assumptions based on linear tracks as well as enforcing pipelining. A more robust vertex-finding will be developed as well. This will not in itself require fewer resources or decrease the latency, but it will result in fewer vertex candidates to be fit. This is important as the vertex-fitting is the most complex block. As described in a previous section, the current vertex-finding only selects pairs of 3D tracks which do not share hits. The future version will include

PoS(VERTEX2025)010

approximate calculations of the vertex position and the distance between the closest points on the two tracks.

The online reconstruction has several configurable parameters which affect the number of track and vertex candidates which are fit, and how many are fit in parallel. Selections such as the track-finding window, track quality cuts, and selections in vertex-finding can reduce the number of objects at each step. It is crucial to understand the selections and tune their use to decrease the time required for each event while minimizing efficiencies.

It may also be required to have multiple instances of the track and vertex-fitting blocks, as they require the most cycles, at the cost of more resources. For these developments, it will be important to understand the latency and resource budget, as well as the expected number of objects at each step to balance performance, resource usage, and selection efficiencies.

5. General Application

The central blocks, track-finding, track-fitting, vertex-finding, and vertex-fitting, are implemented such that they are general to straight tracks. As such, they are applicable to other contexts, such as different test beam setups, or for example cosmic ray tracking. For this, only the number of hits per track might need to be changed. Assumptions which are included in the implementations, such as the X and Y track projections being independent, or the linear algebra assumptions encoded in the track-fitting, remain valid for straight tracks. The other blocks are specific to MUonE and depend on details of the tracker geometry. The developments outlined in Section 4.2 will not affect the general applicability of the algorithm.

6. Conclusion

Data rates for the MUonE experiment necessitate event selection, and real-time track reconstruction is apt to do so. It has been shown that real-time tracking with HLS is feasible through testing in beam and expected throughput and resource use. Real-time reconstruction, including vertex-fitting, will be tested during a 2025 test beam, and there is a clear path forward to improve the current implementation to meet the latency and resource needs of the experiment. There are also plans to study the parameters of the reconstructions and tune them to decrease the number of fake objects at each stage and exploit multiple instantiations of blocks depending on the number of objects and available resources. Finally, the implementations of the track-finding, track-fitting, vertex-finding, and vertex-fitting are general to straight tracks and so can be relevant to contexts other than MUonE.

References

[1] G. Abbiendi et al., *Measuring the leading hadronic contribution to the muon g-2 via μe scattering*, *Eur. Phys. J. C* **77** (2017) 139 [1609.08987].

[2] Muon g-2 collaboration, *Final Report of the Muon E821 Anomalous Magnetic Moment Measurement at BNL*, *Phys. Rev. D* **73** (2006) 072003 [hep-ex/0602035].

[3] Muon g-2 collaboration, *Measurement of the Positive Muon Anomalous Magnetic Moment to 0.20 ppm*, *Phys. Rev. Lett.* **131** (2023) 161802 [2308.06230].

[4] The CMS Collaboration, *The Phase-2 Upgrade of the CMS Tracker*, Tech. Rep. , CERN, Geneva (6, 2017), DOI.

[5] A. Rose et al., *Serenity: An ATCA prototyping platform for CMS Phase-2*, in *Proceedings of Science*, vol. 343, 2018, DOI.

[6] AMD Xilinx, *Amd vitis high-level synthesis*, 2022.

[7] AMD Xilinx, *Amd vitis blas library*, 2022.

[8] AMD Xilinx, *UltraScale™ Architecture and Product Data Sheet: Overview*, Tech. Rep. (11, 2022).