4D Track Reconstruction on Free-Streaming Data at PANDA at FAIR

On Behalf of the PANDA Collaboration

Jenny Taylor[1,2*], Michael Papenbrock[1], Tobias Stockmanns[3], Ralf Kliemt[2,4], Tord Johansson[1], Adeel Akram[1], Karin Schönning[1]

[1*]Uppsala University, Uppsala, Sweden.
[2*]GSI Helmholtzzentrum für Schwerionenforschung GmbH, Fair Forschung NRW, Darmstadt, Germany.
[3]Forschungszentrum Jülich GmbH, Jülich, Germany.
[4]Ruhr-Universität Bochum, Institut für Experimentalphysik I, Bochum, Germany.

*Corresponding author(s). E-mail(s): j.taylor@gsi.de;

Abstract

A new generation of experiments is being developed, where the challenge of separating rare signal processes from background at high intensities requires a change of trigger paradigm. At the future PANDA experiment at FAIR, hardware triggers will be abandoned and instead a purely software-based system will be used. This requires novel reconstruction methods with the ability to process data from many events simultaneously.

A 4D tracking algorithm based on the cellular automaton has been developed which will utilize the timing information from detector signals. Simulation studies have been performed to test its performance on the foreseen free-streaming data from the PANDA detector. For this purpose, a quality assurance procedure for tracking on free-streaming data was implemented in the PANDA software. The studies show that at higher interaction rates, 4D tracking performs better than the 3D algorithm in terms of efficiency, 84% compared to 77%. The fake track suppression is also greatly improved, compared to the 3D tracking with roughly a 50% decrease in the ghost rate.

Keywords: 4D Tracking, PANDA, Quality Assurance, Free-Streaming Data, Cellular Automaton

1 Introduction

The search for rare processes and previously unknown particles push the frontier of modern experimental hadron physics to higher and higher reaction intensities. We are even about to reach a point where our ability to produce abundant data outpaces our ability to retain them. This fact is correlated with the rapid increase in processing power of computing hardware, in contrast to the slower evolution of data transport and storage. As a result of the ever-improving accelerators and detectors world-wide, the past decades have seen a remarkably rapid progress, with discoveries such as the Higgs boson [1, 2] and exotic multiquark states [3–5]. In order to solve current puzzles in hadron and particle physics, a new generation of facilities is underway. For all of these, a crucial component will be to discriminate between signals of interest and background at an early stage.

For a long time, hardware triggers served this purpose very well, selecting potentially interesting physics events by identifying specific signatures *e.g.* high energy loss, and storing the corresponding events for reconstruction and more refined filtering later. As more and more of the low-hanging fruits of subatomic physics have been harvested, the hunting ground for unknown physics has to a large extent become processes whose signatures are expected to be very similar to those of background. Hence, more sophisticated data filtering methods are required – and the hardware trigger paradigm falls short. The new generation of facilities is responding to this challenge by moving to different variations of a trigger-less data acquisition, having data streaming continuously from the detectors and selecting the data only after they have been reconstructed online during their acquisition. These approaches are referred to as *software triggers*.

Currently operating and upcoming experiments are embracing the new paradigm in various degrees. The high-level trigger of the ALICE experiment performs the full event reconstruction, calibration and high-level data quality assurance in almost real time [6]. The LHCb collaboration is currently making efforts to implement full online analysis chains in the software trigger apart from computing observables for the event filtering [7]. XENON1T has been collecting data altogether with a software-trigger [8]. At the future Facility for Antiproton and Ion Research (FAIR), the CBM [9] and PANDA [10] experiments will also employ fully software-based trigger systems.

The introduction of software triggers calls for a different treatment of the detector signals. In the traditional approach, data are processed event-by-event in the form of contained data packages, containing all detector signals collected within a pre-defined time-window as a response to the hardware trigger. In the approach needed for free-streaming data, a continuous stream of data is processed and organised based on spatial and timing information. Tracks can then be formed *e.g.* by 4D track reconstruction where time is used in addition to spatial coordinates. A 4D Cellular Automaton (CA) is utilized at CBM [11] with excellent performance.

Track reconstruction algorithms are evaluated using Quality Assurance (QA) methods. For tracking event-by-event, there are established methods such as those used and outlined in Ref. [12]. However, QA methods for tracking on free-streaming data are hard to find within hadron and HEP literature.

In this article, a 4D CA track finding algorithm developed for a continuous stream of hit data from the PANDA experiment is presented. In addition, a quality assurance
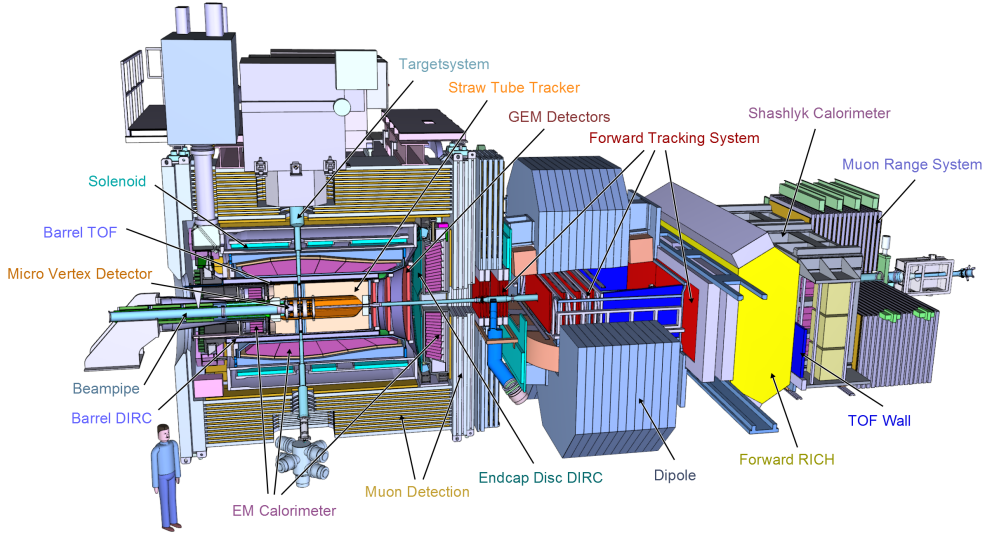
Fig. 1: The PANDA detector. All subdetectors are labeled.

method to evaluate the performance of the track finding was developed which is the first of its kind and is outlined in detail in this article.

2 The PANDA experiment

PANDA (antiProton ANnihilations at DArmstadt) [13] is a future multipurpose detector designed to investigate QCD in the energy region where quarks form hadrons. The broad physics program include four main pillars: strangeness physics, charm and exotics, nucleon structure and hadrons in nuclei [14].

The antiproton beam will be delivered by the High Energy Storage Ring (HESR), a racetrack-shaped ring with a circumference of 575 m. The beam will cover a momentum range from 1.5 GeV/c to 15.0 GeV/c. In Figure 1, the beam enters from the left.

The detector, shown in Figure 1, is divided into a target solenoid spectrometer surrounding the interaction point and a forward dipole spectrometer to detect particles emitted at small scattering angles. Both spectrometers comprise subdetectors for tracking, particle identification and calorimetry. The track reconstruction algorithm presented in this work treats signals from the central Straw Tube Tracker (STT) [15] and can be extended to also include the the Micro Vertex Detector (MVD) [16], the Gas Electron Multipliers (GEM) [17] and the Forward Tracking System (FTS) [18]. The coordinate system is defined with the z-direction along the beam line and the xy-plane perpendicular to the beam line.

2.1 The Straw Tube Tracker

The main tracking detector of the central part of PANDA is the STT, Figure 2. It consists of 4224 closely packed single-channel readout drift tubes placed in a hexagonal pattern. The detector has a cylindrical shape with roughly 27 tube layers in the radial direction, 19 parallel layers for xy-reconstruction and 8 central layers of tilted tubes, $\pm 2.9°$ with respect to the beam line, for z-reconstruction. The detector will be placed between 15.0 cm and 41.8 cm in radial direction and the tubes are 140.0 cm long with a diameter of 1.0 cm. The tubes will be filled with a 90/10 gas mixture of Ar/CO_2. The walls consist of a 27 μm thick layer of aluminized mylar foil and the tubes will have 20 μm thick anode wires in the center that are made of gold-plated tungsten.

When a charged particle traverses a tube, the gas will be ionized and free electrons are created along the trajectory of the particle. An applied electric field will make the electrons accelerate towards the anode wire. Close to the wire, an avalanche of electrons will be created that give rise to a signal that is large enough for readout. By measuring the arrival time of the electrons precisely and in combination with other detectors it is possible to determine the *drift* time of the electrons. Based on that the *isochrone* radius is calculated, a circle centered at the tube wire and going through the point of closes approach of the particle to the wire. The maximum drift time, *i.e.* the time it takes the electron to drift to the wire, is around 250 ns for electrons created close to the tube wall.

2.2 Free-streaming detector data

The HESR antiproton beam will be quasi-continuous, and give rise to Poisson distributed instantaneous interaction rates when interacting with the fixed internal proton or nuclear target. The revolution time of the beam is around 2000 ns, corresponding to a frequency of 500 kHz, but the exact number depends on the beam momentum. The beam will be active during 80% of the time and a gap with no beam during 20%. This structure creates natural separation for data packaging.

During the first phase of operation, the luminosity will be 2×10^{31} cm^{-1}s^{-1} resulting in an average interaction rate of 2.0 MHz. The final design luminosity of PANDA is 2×10^{32} cm^{-1}s^{-1} resulting in an average interaction rate of 20.0 MHz. The raw data rate before the software trigger will be around 200 GB/s. This needs to be reduced by a factor of 1000.

The HESR beam structure in combination with the foreseen high intensities implies that signals originating from different physics reactions will overlap in time. At an average interaction rate of 2.0 MHz, the average time between the start of two consecutive events will be 500 ns. In Figure 3, it is however shown that many consecutive events are separated by much less than 500 ns. At the PANDA design interaction rate of 20.0 MHz, the average time between two events will be 50 ns.

Each subdetector at PANDA will be self-triggering so the detector information is read out independently of other subdetectors. The information is then synchronized and sorted in time by using the time-stamp each detector has to assign to its data. The reconstruction and event building can subsequently take place. This, combined with the Poisson distributed interaction rates, means that the hits from several events will
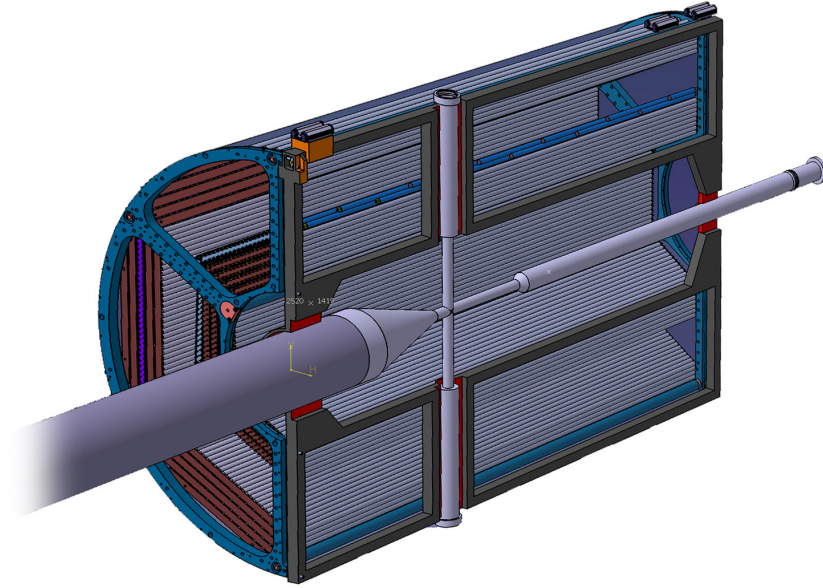
Fig. 2: Cutaway view of the Straw Tube Tracker. The detector surrounds the beamline and has vertical cavities for the target beamline.

be processed simultaneously. Since the STT has a dift-time of up to 250 ns, signals from different events will have a higher tendency to overlap in time in this detector compared to others, as illustrated in Figure 4.

2.2.1 Simulations for Free-Streaming Data

The free-streaming data is simulated in FairRoot [19] upon which the PANDA software PandaRoot [20] builds. Figure 5 illustrates the difference between time-sorted and event-sorted simulations. In both cases, the events are generated one-by-one at the *simulation stage*. This is followed by the *digitization stage* where the output from the simulation is converted to signals that realistically mimics the detector output, including spatial resolution, detector granularity, timestamps and pulse shapes. At this stage, event-sorted and time-sorted simulations differ: while in the first case, the digitized hits are sorted event-wise, in the time-sorted case, the digitized hits are sorted according to time instead, and passed on to the next *reconstruction* stage in pre-defined time intervals into *bursts*. Hence, the event-sorted simulation chain is synchronous while the time-sorted chain is asynchronous. The latter means that data reconstruction operates on detector hits coming from several events simultaneously.
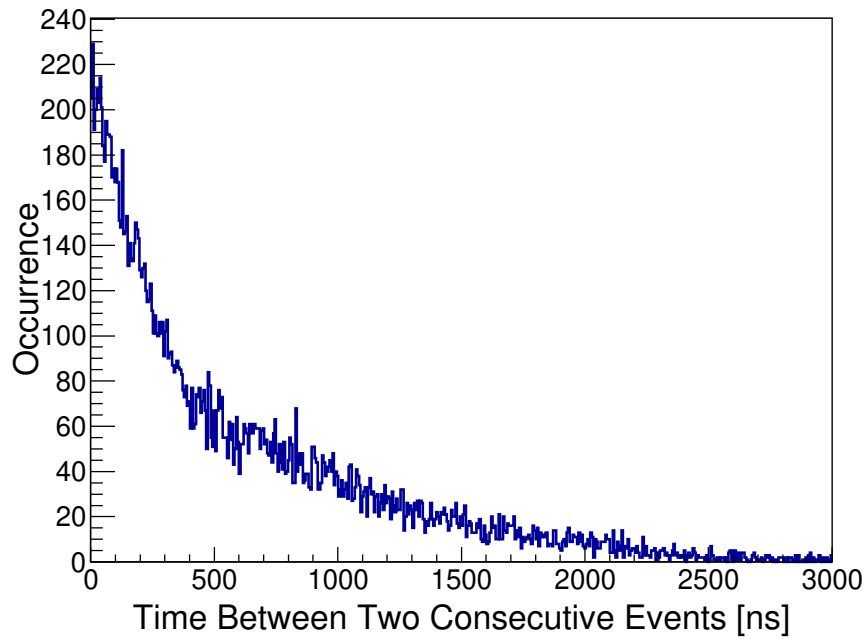
Fig. 3: The time between the start of two consecutive events at 2.0 MHz. The average time is 500 ns at this interaction rate. Many events occur closer in time as well as more separated.
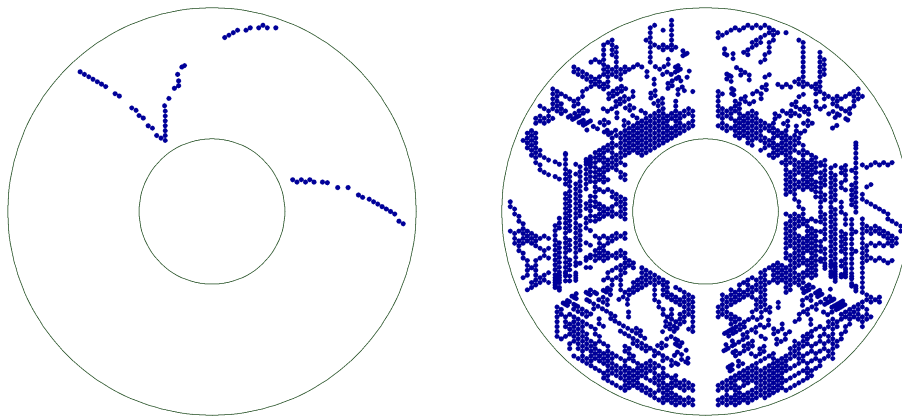
Fig. 4: Detector occupancy of the PANDA Straw Tube Tracker at low intensities, where events do not overlap in time (left) and high intensities, where up to 40 events overlap (right).
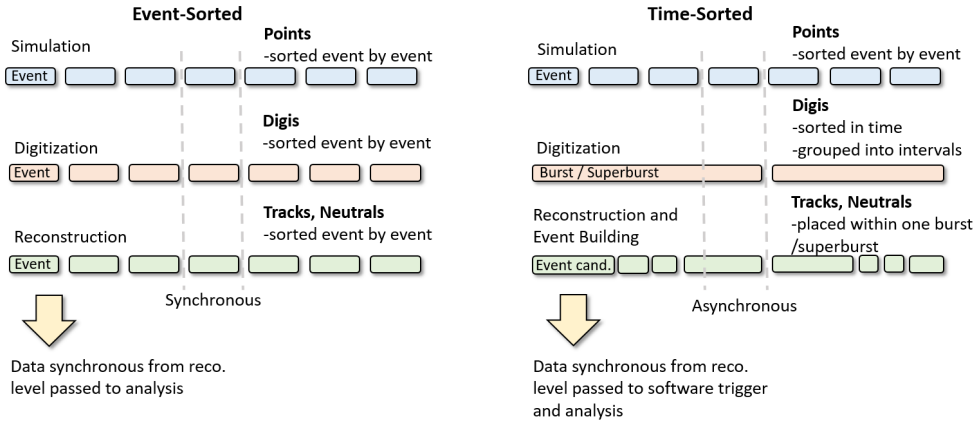
Fig. 5: The event-sorted simulation (left) compared to the time-sorted simulation (right). The event-sorted simulation is synchronous until the software trigger and the time-sorted simulation is asynchronous until the reconstruction and synchronous following that. One burst of data correspond to events within a certain time-interval and a superburst consists of a number of bursts
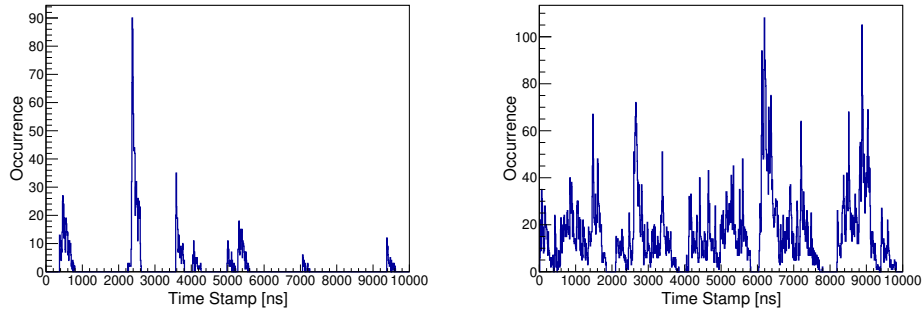
Fig. 6: Time-stamps of the hits in the STT at 2.0 MHz (left) and 20.0 MHz (right). The time-stamp peaks at 2.0 MHz correspond roughly to one event.

When processing time-sorted data, a time interval for obtaining the hits must be chosen. In the PANDA STT detector, the time structure of the hits is shown in Figure 6 for two cases. At an interaction rate of 2.0 MHz (left panel), the events are fairly well separated in time since the average time between two events exceeds the maximum drift time of the electrons. However, at 20.0 MHz (right panel), signals from different events are overlapping in time. The larger gaps at around 2000, 4000, 6000, 8000 and 10000 ns reflect the time structure of the beam.

7

3 4D track reconstruction

The advent of fast pixel detectors has led to the introduction of 4D track reconstruction, where a timestamp is assigned to each point (or hit) along the track. In environments with high track multiplicity, timing information is expected to help distinguishing between hits from different particles.

In the following, an algorithm is presented that was originally developed to operate on event-sorted data but later updated for time-sorted data. This track finder combines a CA [21] for pattern recognition with a Riemann mapping procedure for track parameter estimation. The algorithm is based on pattern recognition in the PANDA STT, and is general in the sense that it does not require tracks to originate from the beam-target interaction point. Hence, it should be able to reconstruct tracks from secondary vertices, such as tracks from hyperon decay products. In this work, the algorithm was developed to perform tracking in 4D, utilizing the timestamp of each hit and to work on time-sorted data.

3.1 Track reconstruction event-by-event

The CA-based tracking algorithms operate on a grid of cells, which can have *e.g.* a 1D or a 2D/3D rectangular or hexagonal layout. The state of a cell is determined by the state of its neighbours. Since CA algorithms are inherently local, they are suitable for parallelization which has made them popular for track reconstruction purposes.

Given its tightly packed, hexagonal layout of the STT, CA algorithms are suitable for track finding in this detector and was first applied in Refs [22, 23]. Here, each STT tube translates into a cell.

In CA algorithms, the state of a cell is either active or inactive. In this case with the STT tracker, active means that a hit is read out from the corresponding straw. Whether an active cell is a part of a pattern or not is uniquely determined by the states of its neighbours. Considering only the number of the closest, active neighbours $N_{neighbours}$, the following classification can be done:

1. $N_{neighbours} = 0$ — The hit is not associated with a track or tracklet.
2. $0 < N_{neighbours} \leq 2$ — The hit is unambiguously associated with a track or tracklet.

3. $N_{neighbours} > 2$ — The hit is ambiguously associated with one or several tracks or tracklets.

The reasoning behind the classifications is the following. If a hit is surrounded by no other hits (category 1), it is not considered to be part of a track or tracklet, but rather a random background hit. However, being surrounded by one or two other hits (category 2) suggests that the hit is part a track or tracklet that either ends in this hit or passes through. Any additional neighbours beyond two (category 3) are usually caused by several tracks crossing each other or a random background hit occurring close to a track, hence the ambiguity. However, the classification is made independently of any cells outside of the neighborhood. In case of ambiguous hits within a track candidate, the algorithm produces separate track segments which are disconnected at the point of the ambiguous hit. Consequently, the algorithm creates many track segments that
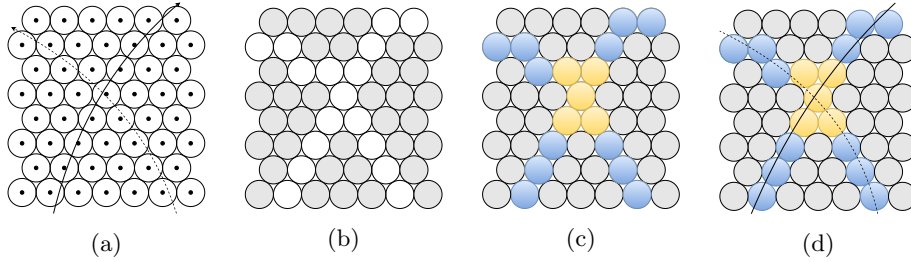
Fig. 7: Illustration of the Cellular Automaton workflow. (a) Particles traverse the detector. (b) Cells corresponding to detector hit are marked as active. (c) Active cells are classified as *unambiguous* (blue) or *ambiguous* (yellow). (d) Track fits are used to resolve ambiguities and merge tracklets.

still have to be connected. Track segments that are connected via ambiguous hits can be connected using a preliminary parameter estimation of the tracks in the xy-plane by a Riemann paraboloid fitting procedure [24]. Based on these parameters, track segments are combined into common tracks.

The procedure is illustrated in Figure 7. After particles traverse the detector (Figure 7(a)), the cells that correspond to detector hits are activated (Figure 7(b)). All activated cells are categorized as unambiguous (blue), if they have one or two neighboring activated cells, or ambiguous (yellow), if they have more activated neighboring cells (Figure 7(c)). All unambiguous cells are assigned to one *tracklet*, preliminary track segment, and all ambiguous cells are assigned to the closest final track after a Riemann track fit is performed (Figure 7(d)). The algorithm for selecting the neighborhood is shown in Alg. 1.

Algorithm 1: 3D Cellular Automaton

Data: STT hits
Result: clustered STT hits
begin
    for *each STT hit* **do**
        if *STT hit is neighbouring another STT hit* **then**
            accept hits as neighbours
        else
            reject hits as neighbours
    return *clustered hits*

3.2 Track reconstruction of free-streaming data

To adapt the reconstruction algorithms that work event-by-event to also work well for the free-streaming data with event-mixing, spatial information from the hits alone

will not be sufficient to perform the pattern recognition (see Sec. 2.2). The timing information of the electronic signals has to be taken into account as well.

In PandaRoot simulations, the time-stamp of a hit in the STT, t_{hit}, is calculated by the time at which a reaction occurred t_{event}, the travel time of a particle from its point of creation to the detector element that was hit t_{travel}, and the drift time of the electrons inside a straw tube from the point of closest approach between the particle trajectory and the tube's anode wire *i.e.* the time it takes the first electrons to arrive, t_{drift}. The final expression becomes

$$t_{hit} = t_{event} + t_{travel} + t_{drift}. \tag{1}$$

Under realistic conditions in the PANDA experiment, the orders of magnitude for these values would be quite different. While t_{event} would be continuously increasing, the difference between adjacent events would be $\mathcal{O}(\Delta t_{event}) = 10 - 100\,\text{ns}$. The travel time of a particle is considerably shorter with $\mathcal{O}(t_{travel}) = 0.01 - 1\,\text{ns}$. It becomes negligible when compared to the drift time of the electrons, i.e. $\mathcal{O}(t_{drift}) = 100\,\text{ns}$. However, with the mean time between events being of a similar or even smaller order of magnitude than the drift time, mixing of hit signal between two or more events becomes possible and even common.

Algorithm 2: 4D Cellular Automaton

Data: STT hits, threshold time
Result: clustered STT hits
begin
    for *each STT hit* **do**
        if *STT hit is neighbouring another STT hit* **then**
            if *time stamps of both STT hits ¡ threshold time* **then**
                accept hits as neighbours
            else
                reject hits as neighbours
    return *clustered hits*

To perform tracking in this high occupancy environment, the approach in Sec. 3.1 was modified to include the hit time stamps as given in Eq. 1. The resulting algorithm is shown in Alg. 2. If spatially neighboring hits occur closer in time than 250 ns, they are classified as neighbors, otherwise discarded and can not be contained in the same tracklet.

4 Quality assurance for track reconstruction on free-streaming data

A class for evaluating the efficiency and other quality variables is available in Panda-Root in order to have a common basis for evaluating the performance of the PANDA

tracking algorithms. The quality of the reconstruction algorithms is assessed by comparing the contents of the reconstructed tracks with their generated counterparts. It is not known, a priori, which generated track corresponds to the reconstructed track. The *true* track is therefore defined as that to which the majority of the hits in the reconstructed track belong.

A reference track set must be chosen in order to have relevant generated tracks to compare the reconstructed tracks to. This is done using an ideal track finder that uses the generated track with the hits to create a reconstructed track object from this. The performance is then only limited by the detector acceptance and efficiency. One benefit of creating such an object is that the hits in the ideal track and the realistically reconstructed track can be directly compared since it has the same data structure. Another benefit is that different hit conditions, that correspond to a minimum hit requirement in different detectors for a track to be reconstructed, can be imposed on the ideal track. This is needed since the efficiency should be normalized to those tracks that can be reconstructed in the relevant subdetectors. The condition used for this study is ≥ 6 STT hits.

The matching of the ideal track to the realistically reconstructed track is performed through a system of links that contain the information about where an object is stored and to which objects it relates to. For example a hit contains links to the monte-carlo particles that created it.

The hit finding efficiency is defined as

$$\epsilon_{hit} = \frac{N_{correct}}{N_{gen}}, \tag{2}$$

where N_{gen} is the number of hits in the generated track and $N_{correct}$ is the number of found hits in the reconstructed track from the generated track. If $\epsilon_{hit} = 1$, all hits from the generated track were found. The hit purity within a track is defined as

$$p_{hit} = \frac{N_{correct}}{N_{found}}, \tag{3}$$

where N_{found} is the total number of found hits from any generated track and $N_{correct}$ is the number of found hits from the true generated track. If $p_{hit} = 1$, only hits from the true generated track were found and if $p_{hit} < 1$, hits from additional generated tracks were found. If hits from generated tracks that are not the true track are included in the reconstructed track, these hits are referred to as *impurities*. No noise hits were included in this study.

Track categories are defined in terms of the hit finding efficiency and purity:

1. **Fully, purely, found**: all hits from a generated track were found and there are no impurities in the reconstructed track. $\epsilon_{hit} = 1$ and $p_{hit} = 1$.
2. **Fully, impurely, found**: all hits from a generated track were found and at least 70% of all hits are from the true track. $\epsilon_{hit} = 1$ and $0.7 < p_{hit} < 1$.
3. **Partly, purely, found**: not all hits from a generated track were found and there are no impurities in the reconstructed track. $0.7 < \epsilon_{hit} < 1$ and $p_{hit} = 1$.
4. **Partly, impurely, found**: not all hits from a generated track were found and at least 70% of all hits are from the true track. $0.7 < \epsilon_{hit} < 1$ and $0.7 < p_{hit} < 1$.

5. **Ghost**: no true generated track could be identified. $p_{hit} < 0.7$.
6. **Clone**: one generated track was reconstructed more than once. The track with the most hits from the true generated track is referred to as the real track and other reconstructed tracks which have been created from the same generated track are referred to as clones.

The tracks in the first four categories constitute the set of correctly reconstructed tracks whereas the tracks in the final two categories are referred to as wrongly reconstructed or false tacks.

The track finding efficiency is defined as

$$\epsilon_{track} = \frac{N_{Reco}}{N_{Gen}}, \tag{4}$$

where N_{Gen} is the number of generated tracks and N_{Reco} is the number of reconstructed tracks. Ghost and clone tracks are not included in the calculation of the track finding efficiency.

The two main differences between the event-sorted and time-sorted tracking quality assurance is 1) the way the data is read and 2) the way the matching between tracks is carried out.

1. In the simulation for free-streaming data, the data is retrieved within a certain time interval. This can be done in two ways: all data within a specified time interval can be retrieved or all data before a gap in the data stream has been found. This is not the case for data event-by-event where the data is always retrieved event-wise.
2. In the simulation for free-streaming data, the matching between the realistically reconstructed track and the ideal track is performed by matching a unique link to each track of the tracks to each other. In the event-by-event reconstruction, this is done using an index.

The benefit of the method for free-streaming data is that it is more general than the event-by-event method and works for both ways of retrieving the data. The disadvantage is that it is considerably slower using the full link system. However, since the quality assurance is run offline, this is not problematic.

5 Results

5.1 Data Samples

In order to benchmark the algorithms, a data sample of 1000 events based on the Fritiof Model (FTF) [25, 26] was generated with PandaRoot. This was done at four different interaction rates: 0.5, 1.0, 1.5, 2.0 and 4.0 MHz. The number of possible tracks, within the generated event samples of 1000 events, for each interaction rate is around 2670. The exact values differ slightly because of how the hits occur in time with respect to the time interval for retrieving the data at the different interaction rates. This results in a difference in the number of reconstructed tracks within the time interval. The interaction rates were chosen within an interval around 2.0 MHz which is the expected average interaction rate at the start of PANDA. The beam time structure and Poisson distributed interaction rate is included in the simulations. A
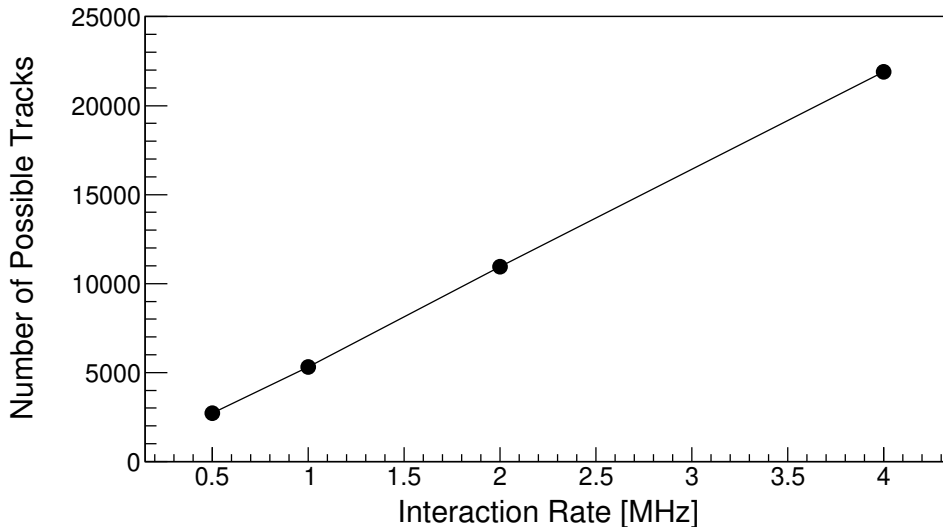
**Fig. 8**: The number of reconstructed ideal tracks in 1000 bursts.

beam momentum of 6.2 GeV/c was chosen as a mid-range point to benchmark the algorithm. For a reference track set, the ideal track finder is used with the requirement that ≥ 6 STT hits are present in the track. Moreover, the data is sorted in time and then retrieved in bunches of 2000 ns for both the ideal and realistic tracking. The time interval mimics the beam revolution time.

There is no pre-filtering of tracks on *e.g.* transverse momentum or point of origin in the simulations presented in this study. This means that challenging cases such as particles that are trapped in the magnetic field and curl as well as decay products from long-lived particles are included.

The number of reconstructed tracks within 1000 bursts at the different interaction rates can be seen in Fig. 8. The number of ideal tracks scales linearly with the interaction rate. In order to reduce the runtime of the QA, 8557 tracks from 400 bursts are used in this data point to evaluate the efficiency of the reconstruction.

5.2 Varying interaction rates

The efficiency is presented for both the 3D version of the CA as well as the 4D version for different interaction rates in the left panel of Figure 9. The efficiencies for the four track categories, fully pure, fully impure, partially pure and partially impure as well as the full efficiency are shown. At 0.5 MHz there is little temporal overlap between the events in the STT and the 3D and 4D version of the CA are expected to have a similar efficiency. It can be seen from the figure that this is also the case. However, at higher interaction rates the 4D version yields a larger total efficiency. The largest contribution to the full efficiency comes from partly pure tracks. The impure tracks

13
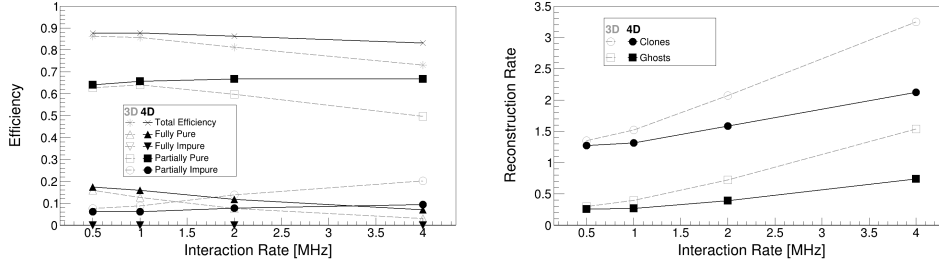
Fig. 9: Efficiency (left) and the fake rate (right) for the 3D CA (dashed, gray, lines) and the 4D CA (solid, black, lines) vs the interaction rate. The time-cut for the 4D CA is 250 ns.

contribute with almost 20% for the 3D CA at 4.0 MHz but contributes with around 10% over the full range for the 4D CA.

The fake rate (ghost and clone rate) is presented for varying interaction rates in the left panel of Figure 9 for both the 3D and 4D version of the CA. The time-cut is 250 ns. The number of fake tracks in each category is normalised to the number of possible true tracks. It is clear that including the time information decreases the fake rate and the effect is most prominent at higher interaction rates.

5.3 Varying time-cut

The effect of the time-cut value on the efficiency for the 4D CA is presented in Figure 10, left panel. The interaction rate is 2.0 MHz in all data points. The efficiency is evaluated at a time-cut of the nominal 250 ns as well as 50, 100, 150, 200 and 300 ns. If the time-cut is below 250 ns, the number of fully pure tracks decrease while the number of partially pure tracks increase. Hence, the full efficiency remains stable. This is because the time-cut is so tight, true hits start to get discarded from the track. At a time-cut less than 100 ns, the full efficiency start to drop since too few hits are allowed to form a tracklet.

The fake rate as a function of the value of the time-cut in the 4D CA at 2.0 MHz interaction rate can be seen in Figure 10, right panel. The fake rate, for both ghosts and clones, is reduced for stricter time-cuts.

5.4 Run-Time Performance

Since the reconstruction algorithm is being developed to operate online, speed is a consideration. The computational performance for the full reconstruction is shown in Figure 11 for the different interaction rates. The tests were performed on a laptop with the following hardware specifications:

- Processor: Intel Core i7-1185G7, 3.00 GHz
- Memory: 32 GB RAM

14
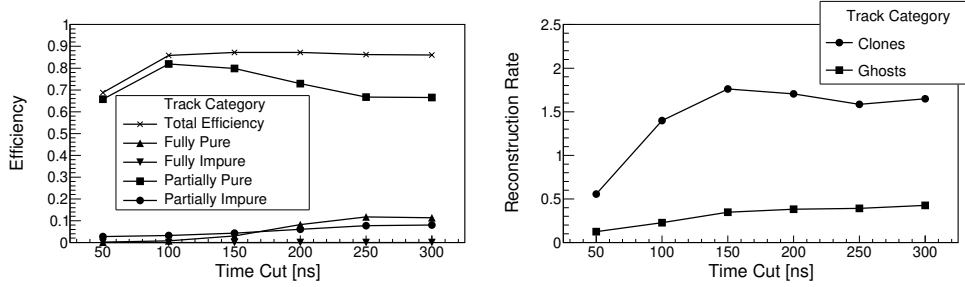
Fig. 10: Efficiency (left) and fake rate (right) for the 4D CA vs the time-cut at 2.0 MHz interaction rate.
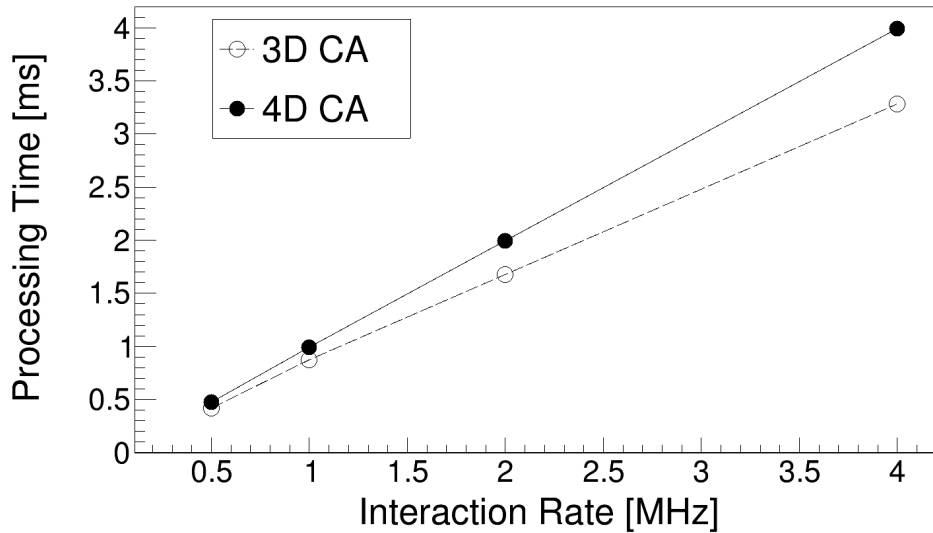
Fig. 11: The time per event for running the full reconstruction for different interaction rates.

The time is calculated as the average processing time for 100 data bursts for each interaction rate. The increasing number of hits per burst makes the processing time per event longer at higher interaction rates. There is a linear dependence between the processing time and the interaction rate.

The time-cut inclusion in Alg. 2 is the only part that separates the 4D version from the 3D version. The run-time for this specific part of the code was therefore examined more closely. The increase in run-time when moving from the 3D version to the 4D is negligible compared to the full time for the reconstruction at all interaction rates. A larger increase occur when going from the event-sorted reconstruction to the time-sorted and processing more data simultaneously in bunches.

15

6 Conclusions and Outlook

A Cellular Automaton-based track reconstruction algorithm and the tracking quality assurance procedure in PandaRoot has been adjusted to work for time-sorted data. The tracking algorithm has also been developed to work in 4D, taking into account both the spatial dimensions and time. A comparison between the 3D method and 4D method shows that at 4.0 MHz interaction rate, the efficiency is higher for the 4D version, 83%, than for the 3D version with a 73% efficiency. In addition, the 4D method yields a much lower fake rate of tracks at higher interaction rates.

Some improvements to the algorithm will be made in the future. The signal from a straw tube is characterized by two timestamps, the leading edge time, t_{LE}, when the signal in the readout electronics reaches above threshold and the trailing edge time, t_{TE} when the signal drops below threshold. Currently, the procedure is used with t_{LE}. However, in the future it will be used with t_{TE}, when this is implemented in the software. This is because t_{TE} is independent of the isochrone radius and therefore contributes with less uncertainty.

This is the first time the quality of a track reconstruction algorithm that works on free-streaming data for PANDA has been evaluated with the standard quality assurance of PandaRoot. Therefore, this marks an important step towards a realistic track reconstruction for PANDA. The quality assurance procedure will be useful for the developments of further algorithms for free-streaming data at PANDA.

7 Conflict of Interest

The authors declare that we have no conflicts of interest.

8 Acknowledgments

References

[1] The 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 **716**, 1–29 (2012) https://doi.org/10.1016/j.physletb.2012.08.020

[2] The CMS collaboration: Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. Physics Letters B **716**, 30–61 (2012) https://doi.org/10.1016/j.physletb.2012.08.021

[3] The Belle Collaboration: Observation of a narrow charmoniumlike state in exclusive $B^{\pm} \rightarrow K^{\pm}\pi^{+}\pi^{-}j/\psi$ decays. Phys. Rev. Lett. **91**, 262001 (2003) https://doi.org/10.1103/PhysRevLett.91.262001

[4] Aubert, B.t.: Observation of a broad structure in the $\pi^+\pi^- j/\psi$ mass spectrum around 4.26 GeV/c^2. Phys. Rev. Lett. **95**, 142001 (2005) https://doi.org/10.1103/PhysRevLett.95.142001

[5] Ablikim, M.t.: Observation of a charged charmoniumlike structure in $e^+e^- \rightarrow \pi^+\pi^- j/\psi$ at $\sqrt{s}$=4.26 GeV. Phys. Rev. Lett. **110**, 252001 (2013) https://doi.org/10.1103/PhysRevLett.110.252001

[6] Acharya, S., *et al.*: Real-time data processing in the ALICE High Level Trigger at the LHC. Comput. Phys. Commun. **242**, 25–48 (2019) https://doi.org/10.1016/j.cpc.2019.04.011 arXiv:1812.08036 [physics.ins-det]

[7] Aaij *et al.,*: Design and performance of the LHCb trigger and full real-time reconstruction in Run 2 of the LHC. JINST **14**(04), 04013 (2019) https://doi.org/10.1088/1748-0221/14/04/P04013 arXiv:1812.10790 [hep-ex]

[8] Aprile, E., *et al.*: The XENON1T Data Acquisition System. JINST **14**(07), 07016 (2019) https://doi.org/10.1088/1748-0221/14/07/P07016 arXiv:1906.00819 [physics.ins-det]

[9] Höhne, C. *et al.,*: The CBM Physics Book: Compressed Baryonic Matter in Laboratory Experiments, pp. 849–972. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-13293-3_7 . https://doi.org/10.1007/978-3-642-13293-3_7

[10] Lutz, M.F.M., et al.: Physics Performance Report for PANDA: Strong Interaction Studies with Antiprotons (2009) arXiv:0903.3905 [hep-ex]

[11] Akishina, V.: Four-dimensional event reconstruction in the CBM experiment. PhD Dissertation, Frankfurt am Main, Germany (2016)

[12] Ikegami Andersson, W., et al.: A Generalized Approach to Longitudinal Momentum Determination in Cylindrical Straw Tube Detectors. Comput. Softw. Big Sci. **5** (2021) https://doi.org/10.1007/s41781-021-00064-0

[13] The PANDA Collaboration: Technical Progress Report for PANDA - Strong Interaction Studies with Antiprotons. FAIR-ESAC (2005)

[14] Barucca, G., et al.: PANDA Phase One. The European Physical Journal A **57** (2021) https://doi.org/10.1140/epja/s10050-021-00475-y

[15] Erni, W., *et al.*: Technical design report for the PANDA (AntiProton Annihilations at Darmstadt) Straw Tube Tracker. Eur. Phys. J. A **49**, 25 (2013) https://doi.org/10.1140/epja/i2013-13025-8 arXiv:1205.5441 [physics.ins-det]

[16] Erni, W., et al.: Technical Design Report for the: PANDA Micro Vertex Detector (2012) arXiv:1207.6581 [physics.ins-det]

[17] Divani Veis, N., Ehret, A., Firoozabadi, M.M., Karabowicz, R., Maas, F., Saito, N., Saito, T.R., Voss, B.: Implementation of the $\bar{P}$ANDA Planar-GEM tracking detector in Monte Carlo simulations. Nucl. Instrum. Meth. A **880**, 201–209 (2018) https://doi.org/10.1016/j.nima.2017.10.036

[18] Smyrski, J., *et al.*: Design of the forward straw tube tracker for the PANDA experiment. JINST **12**(06), 06032 (2017) https://doi.org/10.1088/1748-0221/12/06/C06032

[19] Stockmanns, T.: Continuous-readout simulation with fairroot on the example of the panda experiment. Journal of Physics: Conference Series **664**, 072046 (2015) https://doi.org/10.1088/1742-6596/664/7/072046

[20] S. Spataro for the PANDA Collaboration: The PandaRoot framework for simulation, reconstruction and analysis. J. Phys.: Conf. Ser. **331**(032031) (2011) https://doi.org/10.1088/1742-6596/331/3/032031

[21] Wolfram, S.: Statistical Mechanics of Cellular Automata. Rev. Mod. Phys. **55**, 601 (1983) https://doi.org/10.1103/RevModPhys.55.601

[22] Schumann, J.: Entwicklung eines schnellen Algorithmus zur Suche von Teilchenspuren im "Straw Tube Tracker" des PANDA-Detectors. Bachelor Thesis, FZ Jülich, Germany (2013)

[23] Schumann, J.: Beschleunigung eines Spurfindealgorithmus für den Straw Tube Tracker des PANDA-Detektors durch Parallelisierung mit CUDA C. Master Thesis, FZ Jülich, Germany (2015)

[24] A. Strandlie, J. Wroldsen and R. Frühwirth: Treatment of multiple scattering with the generalized Riemann sphere track fit. Nucl. Instrum. Meth. A **488**(1-2), 332–341 (2002) https://doi.org/10.1016/S0168-9002(02)00465-

[25] B. Andersson, *et al.*: A model for low-p_T hadronic reactions with generalizations to hadron-nucleus and nucleus-nucleus collisions. Nucl. Phys. B **281**, 289–309 (1987)

[26] Nilsson-Almquist, B., Stenlund, E.: Interactions between hadrons and nuclei: The Lund Monte Carlo - FRITIOF version 1.6 -. Comp. Phys. Commun. **43**, 387–397 (1987)