EPJ Web of Conferences **251**, 04011 (2021) https://doi.org/10.1051/epjconf/202125104011
CHEP 2021

Streaming Readout of the CLAS12 Forward Tagger Using TriDAS and JANA2

Fabrizio Ameli[5], *Marco* Battaglieri[1,4], *Mariangela* Bondí[4], *Andrea* Celentano[4], *Sergey* Boyarinov[1], *Nathan* Brei[1], *Tommaso* Chiarusi[2], *Raffaella* De Vita[4], *Cristiano* Fanelli[6], *Vardan* Gyurjyan[1], *David* Lawrence[1,*], *Paolo* Musico[3], *Carmelo* Pellegrino[3], *Ben* Raydo[1], and *Simone* Vallarino[4]

[1]Thomas Jefferson National Accelerator Facility
[2]Istituto Nazionale di Fisica Nucleare - Sezione di Bologna
[3]Istituto Nazionale di Fisica Nucleare - CNAF
[4]Istituto Nazionale di Fisica Nucleare - Sezione di Genova
[5]Istituto Nazionale di Fisica Nucleare - Sezione di Roma
[6]Massachusetts Institute of Technology - M.I.T.

Abstract. An effort is underway to develop streaming readout data acquisition system for the CLAS12 detector in Jefferson Lab's experimental Hall-B. Successful beam tests were performed in the spring and summer of 2020 using a 10GeV electron beam from Jefferson Lab's CEBAF accelerator. The prototype system combined elements of the TriDAS and CODA data acquisition systems with the JANA2 analysis/reconstruction framework. This successfully merged components that included an FPGA stream source, a distributed hit processing system, and software plugins that allowed offline analysis written in C++ to be used for online event filtering. Details of the system design and performance are presented.

1 Introduction

An effort was started in early Spring 2020 to develop a prototype streaming data acquisition system (DAQ) for the CLAS12 detector[1] in experimental Hall-B at Jefferson Lab. This system brought together components from the existing CODA DAQ system[2][3], the TriDAS DAQ system[4] and the JANA2 software framework[5]. The prototype system was used to successfully read out the CLAS12 Forward Tagger(FT) and Forward Hodoscope(FH) detectors in a streaming mode during an active beam test. The COVID-19 pandemic halted the beam operations early in the testing period, but the test was resumed once beam operations started back up in the summer. The longer term goals for the project are to expand the prototype to the full CLAS12 DAQ system and eventually deploy the system to other experiments at Jefferson Lab and elsewhere. One of the main benefits of a streaming readout (SRO) system is that it allows custom hardware triggering systems to be replaced with software algorithms run on cheaper commodity hardware. In addition to removing the deadtime inherent with traditional hardware triggers, this allows more complex triggering algorithms that can operate on whole detector events (as opposed to only fast subdetectors).

*e-mail: davidl@jlab.org

The following sections give details on the setup for the beam test and the various components of the prototype system. This is followed by some analysis results of the data taken with the system.

2 Instrumentation and Resources

2.1 Experimental Setup

The beam test focused on the reaction $e\,\mathrm{X} \rightarrow \pi^0\,X$ with $\pi^0 \rightarrow 2\gamma$ produced by the interaction of 10.6 GeV, ~100 nA, CEBAF electron beam on 125μm lead and 40 cm gaseous deuterium targets. The inclusive π^0 electro-production was chosen because the two decay γs can be detected by a single detector (the CLAS12 Forward Tagger) reducing the complexity of the experimental set up. Moreover, the invariant mass of the two photons to form a π^0, provides a clean signature over the background due to the scattered electron and other electro-magnetic processes.

The Forward Tagger or FT[6] is part of the CLAS12 detector[7] hosted in Hall-B at Jefferson Lab. It is composed of a lead-tungstate electromagnetic calorimeter (FT-Cal), used to measure the photon energy and position, and a plastic scintillator hodoscope (FT-Hodo), used to distinguish neutrals from charged particles and, in turn, identify gammas. The FT covers a small solid angle ($0^o < \phi < 360^o$ and $2.5^o < \theta < 5^o$) in the beam direction. It is mainly used to detect electrons scattered at small angles from the target and forward-going neutral particles, such as π^0, produced in the interaction of the electron beam with the target. The 332 PbWO crystals of the FT-Cal and the 232 tiles of the FT-Hodo were read out by JLab digitizers. The limited number of channels and the combination of two different detectors (calorimeter and plastic scintillators) usually used to trigger the experiment DAQ, represents the ideal bench test for a real on-beam set up. For a quantitative assessment of the streaming DAQ chain, some data was also collected in standard triggered mode for later comparison.

2.2 Readout Electronics

In our streaming readout (SRO) system, physics signals were continuously digitized by the fADC250 flash ADC. The fADC250 is a VME64x 16-channel direct-conversion ADC module, conforming to the VITA-41 switch serial standard (VXS). These high-speed flash ADCs were developed at JLAB as part of the 12 GeV upgrade. Currently these modules are deployed in many JLAB experiments, providing energy deposition, timing, as well as hit and trigger information. The fADC250 is equipped with an FPGA, that receives 12-bit datawords streaming at 250 MHz from 16 fADC channels in a module. The fADC250 FPGA performs data processing for each fADC channel, computes energy sum of all fADCs, and generates acceptance pulses for each fADC. Each VXS crate houses 16 fADC modules that are managed by the VXS Trigger Processor (VTP) module. The VTP is a VXS switch card module, that was designed to play the leading role in the level1 trigger formation as a central or a global trigger processor (CTP, GTP) in a traditional triggered DAQ system.

Figure 1 shows a diagram of the VTP module used to communicate and process data from the fADC250 modules in the crate. The design of the VTP contains high speed backplane serial links to each front-end payload module in the crate. Fiber optic serial links provide communication to other crates. In addition, the VTP has more FPGA resources for data processing logic, and a dual-core 1GHz ARM processor, capable of running data processing components, such as event building, trigger and processing diagnostics. These features make the VTP module an ideal candidate for designing a streaming data acquisition system.

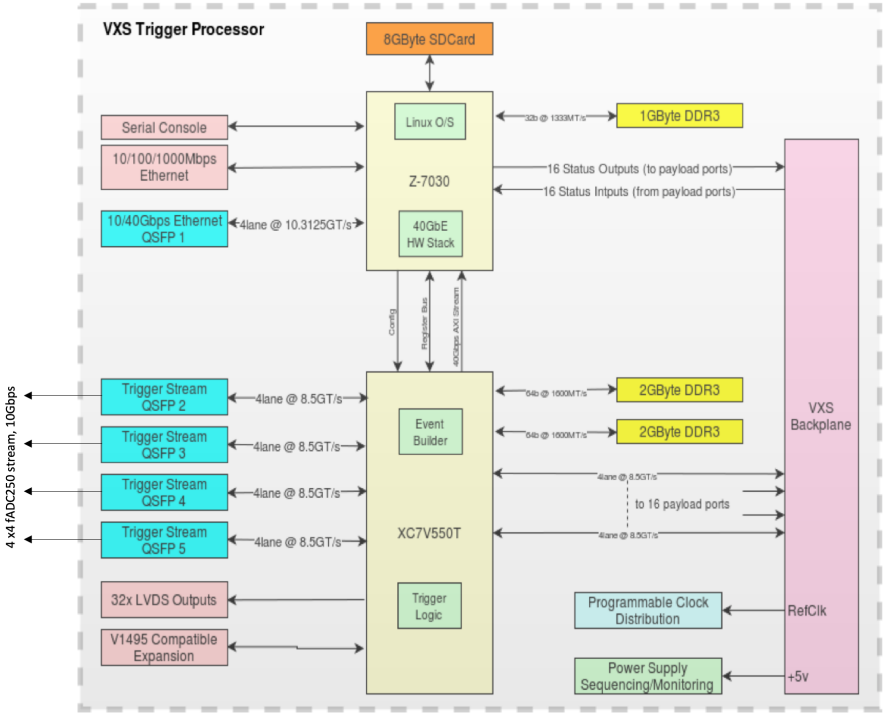

Figure 1. VXS Trigger Processor (VTP) architecture.

In the presented streaming DAQ system prototype the fADC250 data is received at 10Gb/s from each payload slot of the VXS crate. Data from each fADC250 is then buffered into DDR3 memory, where each module has a dedicated 256MB space for data buffering. The role of these buffers is to allow a significant burst of physics input signals, as well as for handling substantial network delays or downstream processor latencies. The VTP streaming firmware implements 4 parallel instances of fADC250 streaming systems, each feeding a 10Gb/s Ethernet link. Each instance handles 4 slots (i.e. 4 fADC250 modules), with a 1 GB memory buffer. Every 8 slots of fADC share a 2GB DDR3 buffer. Ethernet was chosen for the streaming readout interface because of its widespread support and compatibility. The VTP is programmed with the destination IP address and socket, where streaming data from 4 fADC slots are to be sent. Taking into account the computing power of contemporary servers, a socket/server per 10Gb/s link is feasible for data transfer. The fADC data-payload is packed into a TCP data frame, containing a header that includes information about the frame number and the timestamp. This information is necessary for ensuring the data coherency and synchronization. Streaming TCP frames correspond to a programmable time-span (typically 65536ns) for which the reported fADC hits are collected. At the beginning of the streaming readout, the VTP module synchronously starts its own timer that is used to timestamp fADC hits. At every elapse of the frame-time, a TCP data frame is sent containing hits for the corresponding time-span. VTP is responsible for dropping streaming data-frames in case the downstream receivers are not able to accept a higher input rate for a long periods of time. Burst conditions in the order of 100msec at 32MHz/channel (for all channels) are handled without data loss by the VTP DDR3 memory buffers. When the buffers are full, an entire frame (65536ns chunk of data) is dropped. These losses should not happen under normal

conditions when the network and downstream processing chain are efficient enough to keep up with the readout rate. The VTP frame counter (the record-number in the TCP header) is used to identify the number of dropped frames, thus representing the efficiency of entire data stream processing pipeline.

3 Implementation

3.1 Front end Streaming Source

The JLAB data acquisition system called CODA was designed to work with trigger-based readout systems. A key component is the Event Builder, which collects data from 100+ Readout Controllers (ROCs) and VXS Trigger Processor Boards (VTPs). In the traditional triggered mode of operation, the event builder builds events based on event number. Also used for triggered mode is the Trigger Supervisor (TS) module which synchronizes all components using clock, sync, trigger and busy signals. ROCs would read front-end electronics over a VME bus, and VTPs are used to help form trigger decisions and report some trigger-related information. A detailed description of the CLAS12 triggered mode system can be found in the literature[8].

To use the available front-end electronics in streaming mode, the role of the TS was reduced to clock distribution, and the Event Builder was replaced with new SRO components and back-end software capable of gluing the front-end module information together based on timestamp instead of event number. In addition the role of the ROCS and VME bus were reduced to just the initial configuration of the front-end modules. In streaming mode, all front-end electronics readout is performed by the VTP boards over the VXS serial lines rather then VME bus. This allows an increase in the bandwidth limit from about 2Gb/s to 20Gb/s for each of the participating electronics crates, with the possibility to be increased to 40Gb/s if needed. New firmware was developed for the VTPs to implement streaming mode.

Figures 2 and 3 show the original version of CODA, as well as the streaming version of CODA (without back-end which notes as TriDAS). In short, the front end readout software running inside the VME controllers and VTP boards was modified to stream data out freely, and a new SRO component was developed to be the intermediate translator between front-end and back-end. The SRO component is a multi-threaded and multi-node component capable of handling the 20Gb/s data rate from every electronics crate. It gets data from VTPs, converts it into TriDAS format applying appropriate format checks, and then supplies the results into the TriDAS.

3.2 Online Streaming DAQ

The Trigger and Data Acquisition System[4] (TriDAS) is software originally designed and implemented for streaming read-out of Astroparticle Physics events, specifically for the NEMO project which aimed at developing the technologies to build a cubic-kilometre sized telescope for high-energy cosmic neutrinos. The TriDAS scalable, modular, and flexible design made it also adaptable to the requirements of a beam-based experiment with minimal development effort.

TriDAS is made by several software components, each devoted to a specific task in the data-processing chain and implemented in C++11.

In this paper we describe the TriDAS as sketched in fig. 4, which represents the implementation realised for the CLAS12 streaming readout test, in the summer of 2020. The HitManagers (HMs) represent the first data aggregation stage. They receive the data streams from a pre-defined number of CODA translators, topologically corresponding to a sector of

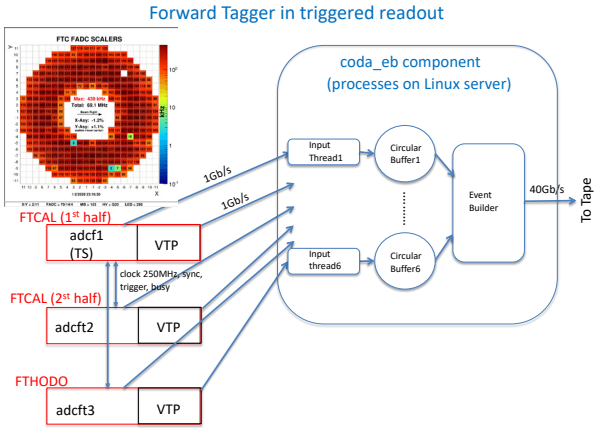

Figure 2. Forward Tagger in triggered readout: Trigger Supervisor supplies 250MHz clock, sync and trigger signals to all front end components, and collects busy conditions

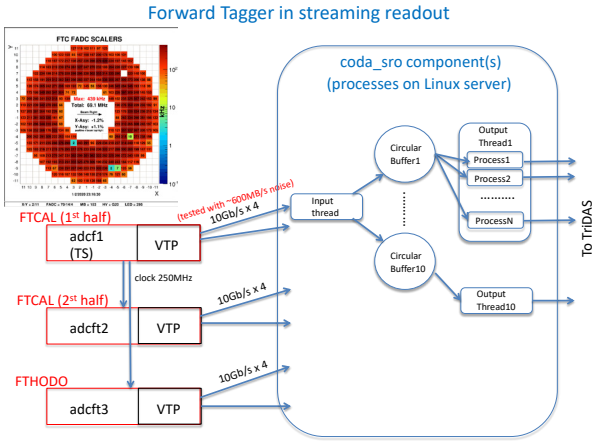

Figure 3. Forward Tagger in streaming readout: Trigger Supervisor supplies 250MHz clock only, data is streaming freely from front end

the detector. Each HM subdivides the collected data into a sequence of time-ordered bunches of data, called Sector Time Slices (STS). The sequence of STSs reproduces the succession of time windows of fixed duration, typically 50 ms. Sharing a common time reference, all HMs arrange their STSs according to the same intervals of time, which are referred to as Time Slices (TS).

The TriggerCPUs (TCPUs) receive the STS assembled by all HMs referring to the same TS and apply the event building and the classification/selection algorithms to the data (see section 3.3). Time Slices are processed in parallel by multiple threads of the same TCPU and by multiple TCPU processes running on the computing farm. For the CLAS12 FT and FH application, the Level 1 events (L1) consisted of the detector data within a time window of 200 ns around a hit whose energy exceeded a threshold of approximately 2 GeV. No attempt

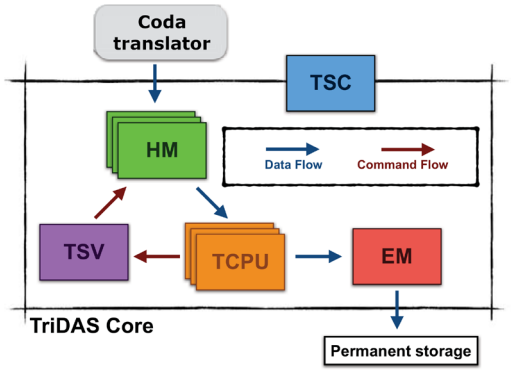

Figure 4. The TriDAS flow diagram. Data flow from the translators (block on the upper-left corner) to the HitManagers (HM) where time-slicing takes place. Each time slice (TS) is coherently routed to a single TriggerCPU (TCPU) process for event building and classification/selection with JANA. The TriDAS SuperVisor (TSV) manages the load balancing via token passing. Selected events are stored to permanent media by the Event Manager (EM). The TriDAS Core application is steered by the TriDAS System Controller (TSC) which responds to the user commands throughout the TriDAS state machine.

is currently made to check for and recover events spanning two STS's due to the detector window (200 ns) being so much smaller than the STS (50 ms) that it represents a negligible amount of potential data loss. Level 1 events identified within a TS are then fed to the L2 classification/selection algorithms that are implemented in separate binaries that are specified in the run configuration file. These binaries are loaded and configured at run-time, allowing one to easily change the L2 algorithms or their parameters on a run-by-run basis without the need for recompiling while still keeping the highest possible computation efficiency.

A token-based mechanism is at the base of the TriDAS SuperVisor (TSV) load balancing. Each TCPU thread owns a token that is given to the TSV on completion of the TS processing. The TSV, then, maintains a pool of "free to use" TCPU threads which are then matched to the new Time Slices that are continuously assembled by the HMs.

The Event Manager (EM) collects the selected L2 events and then writes them to the so called Post Trigger (PT) file.

The TriDAS System Controller (TSC) is the part of the system with which users directly interact. Through it, users may configure and control the TriDAS activities. For the aforementioned test with CLAS12, a simple interface to the TSC was built in order to steer TriDAS along the hierarchical state machine sketched in fig. 5.

In the IDLE state, only the TSC process is running and waits for user commands. Upon the *Init* transition, the TSC retrieves a JSON-formatted run configuration file, called *Datacard*. The Datacard describes the geometry of the detector and the configuration of the TriDAS system for a given run. If the transition is successful, the state machine moves into the Initiated sub-state machine. TriDAS is now in the STANDBY state, where still no other process than TSC is running. During the *Configure* transition, The TSC decides the run number and then starts the HM, TCPU and EM processes on the corresponding nodes. If all the processes

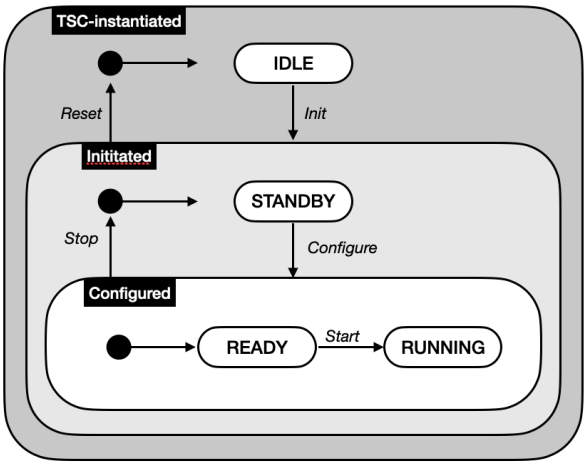

Figure 5. The TriDAS state machine. See text for details.

start successfully the state machine moves into the Configured sub-state machine. The TriDAS is now in the READY state. During the *Start* transition, the TSC computes the start date and time of the run (which for the CLAS12 case is always the fixed value 01/01/2020 00:00:00) and starts the TSV. If this transition is successful, TriDAS moves into the RUNNING state.

3.3 Software Trigger

The TriDAS system supports user-level plugins to allow implementation of custom processing algorithms which can be used to implement a software trigger. For this prototype system, a TriDAS plugin was constructed that implemented the JANA2 framework. JANA2 is a multi-threaded event processing/analysis framework designed for both offline and streaming applications. User algorithms written within the JANA2 framework were then made available for forming software triggers in the form of JANA2 plugins. The benefit of this is that the full suite of reconstruction algorithms used in the offline reconstruction are available for use as triggers/filters in the streaming system. This includes accessing translation tables, and calibration constants.

The software triggering itself was done by using multiple JANA plugins, each implementing their own trigger(s). Each plugin produced one or more *TriggerDecision* objects for each potential "event" identified by the TriDAS system. The decision for each algorithm was in the form of a 16-bit integer where a value of zero meant *no-keep* and any non-zero value meant *keep*. If any trigger algorithm indicated a *keep* condition then The TriDAS system was told to keep the event. A unique 16-bit ID was assigned to each trigger algorithm (passed in the *TriggerDecision* object). The 16-bit ID and 16-bit decision for each "event" was given to TriDAS so it could store the decision for each algorithm with each event written out.

The JANA2 plugins list was determined by the JANA configuration file. Configuration settings for the individual triggers were also set in this file. The system allows different individuals to maintain the code base for their trigger separately while the selection of which triggers are used and what their configurations are is kept in a single configuration file that is read in at run time. Figure 6 shows a snippet of the configuration file with settings for the FT calorimeter multi-cluster trigger.

```
#------------------------------------------------------------------------
# FT Calorimeter based trigger
# This triggers on a minimum number of cluster objects being
# reconstructed in the event. There are multiple options here
# and each clustering algorithm is effectively its own trigger.
TRIGGER:FTCalClus:ENABLED 1
#TRIGGER:FTCalClus:TRIGGER_MODE 0  # 0=smple, 1=advanced (i.e. only count
    clusters matching MIN_CLUSTER_SIZE, MIN_SEED_ENERGY, and MIN_CLUSTER_ENERGY)
TRIGGER:FTCalClus:MIN_CLUSTERS 2
#TRIGGER:FTCalClus:MIN_CLUSTER_ENERGY 500
#TRIGGER:FTCalClus:MIN_CLUSTER_SIZE 2
#TRIGGER:FTCalClus:MIN_SEED_ENERGY  10.0
#TRIGGER:FTCalClus:MIN_CLUSTER_ENERGY 30.0
```

Figure 6. Snippet from the JANA configuration file showing settings for one of the software triggers implemented. The format is simple key-value pairs. Lines starting with "#" are comments. Having the keys start with "TRIGGER:FtCalClus:" was just a choice of convention for this particular beam test.

The on-demand design of JANA2 specifically supports multi-tiered triggering. This means trigger algorithms can be designed such that more expensive algorithms are only run for events or time slices when a decision cannot be made using the output of less expensive algorithms. The benefit of this is that the compute resource required for the software trigger can be provisioned for the average time needed for a keep/no-keep decision rather than for the most expensive algorithm. For example, consider a situation in which one wishes to trigger on events with a detected proton track and two other charged tracks in the forward direction such as a rare Primakoff reaction $\gamma p \rightarrow p\pi^+\pi^-$. Even the rough tracking algorithm used to identify the two very forward going tracks can be expensive. At the same time, one wants to take a significant amount of of pre-scaled events using a minimum bias trigger where only a hit count or minimal calorimeter energy is needed. This would be a very fast algorithm. JANA2 can be configured to only run the more expensive Primakoff tracking trigger for those events which were not already flagged for saving by the fast minimum bias trigger. Figure 7 illustrates how this can work.

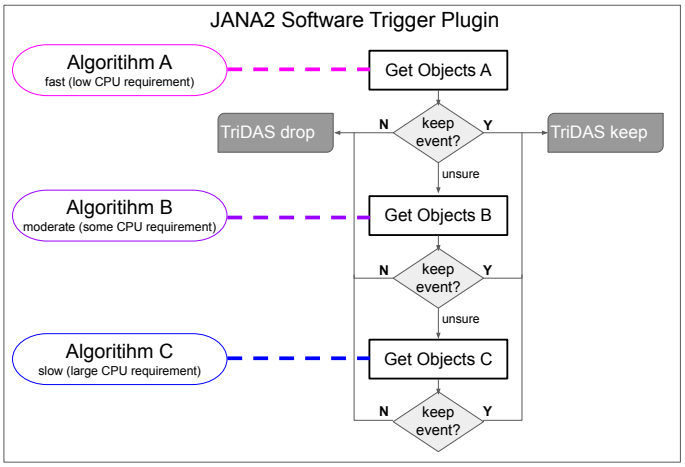

Figure 7. Illustration of how the JANA2 on-demand design can be leveraged to reduce the overall CPU required to implement a software trigger. In this scenario, the more expensive algorithms are only run on an event/time slice when a keep or drop decision cannot be made using a less expensive algorithm.

4 Data Analysis

The off-line data analysis is focused on identification of the $\pi^\circ \rightarrow \gamma\gamma$ decay events where both photons are detected in the calorimeter. In particular, in this paper we report the result obtained by the electron-beam on lead target test.
In the off-line data reconstruction, performed by applying the same full suite of reconstruction algorithms used in the on-line reconstruction, the recorded signal of each crystal was converted into energy by applying proper calibration constants. The latter were determined in a previous calibration run performed in standard trigger mode. The standard calibration procedure is described in [9].
Figure 8 shows the reconstructed $\gamma\gamma$-invariant mass spectrum. It is characterized by two peaks on the π° mass region: the first peak at higher mass is associated to π° production from the lead target, while the second one is related to π°s production from the aluminum target window.

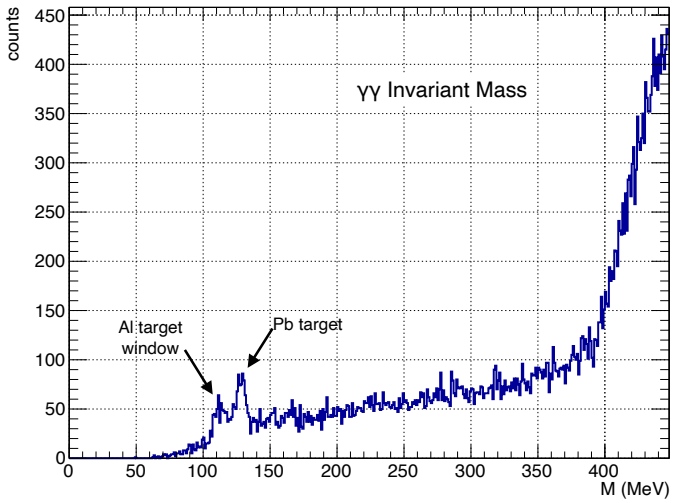

Figure 8. $\gamma\gamma$ invariant mass spectrum. The labeled peaks are both due $\pi^\circ \rightarrow \gamma\gamma$ decays. The peak marked *Al target window* has its position shifted to a lower invariant mass due to the assumption that the vertex is located at the Pb target position when calculating the invariant mass.

4.1 Artificial Intelligence

SRO can further the convergence of online and offline analyses allowing to incorporate new emerging software approaches. For example, the inclusion of high level A.I. algorithms in the analysis pipeline can foster better data quality control during data taking and shorter analysis cycles. A.I. is becoming ubiquitous in nuclear and particle physics and encompasses all the concepts related to the integration of intelligence into machines; unsupervised learning is a type of algorithm able to learn patterns from untagged data (*i.e.*, no training phase) offering new solutions to near real-time reconstruction problems.

An unsupervised hierarchical clustering algorithm inspired by hdbscan [10] has been developed as a plugin within the JANA2 framework. The following are essential features of this unsupervised approach: (i) can be easily ported to other experiments; (ii) formally does not depend on cuts making it less sensitive to variations in experimental conditions during data taking; (iii) able to cope with large number of hits; (iv) excels when dealing with challenging topologies/arbitrary shaped clusters, different cluster size and noise.

The main idea behind the hierarchical clustering is to consider all the information at the hit-level in the detector (spatial, time, and energy) and look at the density of the hits in that

space of parameters, after defining a metric (*e.g.*, euclidean) which allows to define what is called "mutual reachability" among points. In this way, clusters can be interpreted as more likely (higher density) regions separated by less likely regions (lower density). Within this framework all hits have a probability of belonging to a cluster as well as of being outliers and one can make decisions when forming clusters based on these probability values.

Tests have been performed both online and offline (on collected data) to analyze and reconstruct clusters in the FT-Cal, and provided results consistent with the π° yields already discussed.

5 Summary

A prototype streaming data acquisition system was successfully tested in beam conditions during the summer of 2020. The prototype system combined several different software systems with an existing hardware DAQ system for the test. These included the CLAS12 detector and CODA DAQ system, the TriDAS streaming DAQ system, and the JANA2 event processing framework. The test successfully read out the CLAS12 Forward Tagger detector and subsequent analysis was able to extract a clean physics signal in the form of a π° invariant mass peak. The prototype system is being used as the basis for developing a larger system planned for the entire CLAS12 detector and its future physics program.

6 Acknowledgements

We would like to acknowledge the CLAS12 collaboration as well as the JLab technical staff for their accommodation and support of this effort.

The INFN Group has been supported by Italian Ministry of Foreign Affairs (MAECI) as Projects of great Relevance within Italy/US Scientific and Technological Cooperation under grant n. MAE0065689 - PGR00799.

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics under contracts DE-AC05-06OR23177 and DE-SC0019999.

References

[1] V. Burkert, L. Elouadrhiri, K. Adhikari, S. Adhikari, M. Amaryan, D. Anderson, G. Angelini, M. Antonioli, H. Atac, S. Aune et al., Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **959**, 163419 (2020)

[2] W. Watson, J. Chen, G. Heyes, E. Jastrzembski, D. Quarrie, IEEE Transactions on Nuclear Science **41**, 61 (1994)

[3] B. Moffit, D. Abbott, W. Gu, V. Gyurjyan, G. Heyes, E. Jastrzembski, C. Timmer, Real Time Conference, 2012 18th IEEE-NPSS (2012)

[4] T. Chiarusi, M. Favaro, F. Giacomini, M. Manzali, A. Margiotta, C. Pellegrino, Journal of Physics: Conference Series **898**, 032042 (2017)

[5] D. Lawrence, A. Boehnlein, N. Brei, D. Romanov, Journal of Physics: Conference Series **1525**, 012032 (2020)

[6] A. Acker et al., Nucl. Instrum. Meth. A **959**, 163475 (2020)

[7] V.D. Burkert et al., Nucl. Instrum. Meth. A **959**, 163419 (2020)

[8] S. Boyarinov, B. Raydo, C. Cuevas, C. Dickover, H. Dong, G. Heyes, D. Abbott, W. Gu, V. Gyurjyan, E. Jastrzembski et al., Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **966**, 163698 (2020)

[9] A. Acker, et al., Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **959**, 163475 (2020)

[10] R.J. Campello, D. Moulavi, A. Zimek, J. Sander, ACM Transactions on Knowledge Discovery from Data (TKDD) **10**, 1 (2015)