

The design and performance of the ZEUS global tracking trigger

P.D. Allfrey^a, M.A. Bell^a, N. Coppola^{b,1}, R. Devenish^a, S. Dhawan^c, W. Dunne^d,
J. Ferrando^d, D. Gladkov^e, R. Hall-Wilton^{f,2}, M. Hayes^g, H.-P. Jakob^e, J.H. Loizidies^f,
A. Pellegrino^b, A. Polini^{g,3}, V. Roberfroid^{h,4}, M. Soaresⁱ, P.B. Straub^a, A. Stifutkin^e,
M.R. Sutton^{a,5}, S. Topp-Jørgensen^a, B.J. West^f, N.P. Woolley^j, C. Youngman^{g,*}

^aDepartment of Physics, University of Oxford, Keble Road, Oxford OX1 3RH, UK

^bNIKHEF, Amsterdam, Netherlands

^cDepartment of Physics, Yale University, New Haven, CT 06520-8121, USA

^dDepartment of Physics and Astronomy, University of Glasgow, Glasgow, UK

^ePhysikalisches Institut der Universität Bonn, Bonn, Germany

^fDepartment of Physics and Astronomy, University College London, Gower Street, London WC1E 6BT, UK

^gDeutsches Elektronen-Synchrotron DESY, 22603 Hamburg, Germany

^hMarie Curie Fellowship, Department of Physics, University of Oxford, Keble Road, Oxford OX1 3RH, UK

ⁱDepartment of Physics, York University, Ont., Canada M3J 1P3

^jThe Sanctuary, 23 Oakhill Grove, Surbiton, Surrey KT6 6DU, UK

Received 27 January 2007; received in revised form 27 June 2007; accepted 27 June 2007

Available online 12 July 2007

Abstract

The Global Tracking Trigger (GTT) of the ZEUS experiment is described. The GTT is data driven at the ZEUS first level trigger rate of ≤ 600 Hz and performs event-based track finding on data from the experiment's Central Tracking Detector (CTD), silicon Micro Vertex Detector (MVD) and Straw Tube Tracker (STT) forward detectors. The resulting track-based trigger quantities calculated (track multiplicity, vertices, vector meson masses, background event probabilities, etc.) are available within ~ 9 ms and are used in the experiment's second level trigger to improve the selection of physics events. Detector information is pushed to the PC farm of the GTT using PowerPC VME board computers which are either embedded within the detector's frontend readout system (MVD) or are parasitically attached to them via multiple serial transputer links (CTD and STT). Data flow and control is performed via point-to-point Fast and Giga ethernet switched network connections using the TCP protocol. The principal design challenges were: integrating new and interfacing to existing frontend systems, providing a useful trigger result, satisfying the rate and latency requirements and not interfering with ongoing data taking during commissioning. These aims have been achieved. The GTT has been actively used in the ZEUS trigger since 2004 when an initial CTD-only algorithm was used; in 2005 this was upgraded to use MVD information which significantly improves track and primary vertex resolutions. Commissioning problems delayed the STT implementation and its use in the GTT has only been tested.

© 2007 Elsevier B.V. All rights reserved.

PACS: 29.50.+v; 29.40.Gx; 29.85.+c

Keywords: Trigger

*Corresponding author.

E-mail addresses: youngman@mail.desy.de, Christopher.Youngman@desy.de (C. Youngman).

¹Now at DESY, Hamburg, Germany.

²Now at CERN, Geneva, Switzerland.

³Now at INFN, Bologna, Italy.

⁴Now at Inst. de Physique Nuclaire, Univ. Catholique de Louvain, Belgium.

⁵Now at UCL, London, UK.

1. Introduction

The ZEUS detector [1] is designed to record final state particles produced by colliding 920 GeV protons and 27.5 GeV electrons (or positrons) at the HERA accelerator at DESY. Its principal detector components are a tracking system, consisting of forward⁶ (proton beam direction) and central (barrel) tracking detectors, surrounded by a superconducting thin coil 1.43 T solenoid magnet, a high resolution calorimeter, and a muon detector. HERA operates with contra rotating electron and proton bunch trains each of 220 bunches whose bunches collide every 96 ns at the ZEUS interaction point. The detector has been taking data since 1992.

During the HERA luminosity upgrade [2] shutdown in 2000/2001 new tracking detectors were added to the experiment, including a silicon strip Micro Vertex Detector (MVD) located between the beampipe and Central Tracking Detector (CTD), and a Straw Tube Tracker (STT) which replaced the transition radiation detector in the forward region.

The Global Tracking Trigger (GTT) evolved from the requirement of using the high spatial resolution of MVD hits in the second level of the ZEUS trigger to improve selection of physics events using track finding results. The limited number of MVD hits per track—around 2–4 depending on the track flight direction—requires that additional tracking information from other tracking detectors be used during track finding. Hit information from these detectors is interfaced to the GTT via specially developed hardware interfaces.

A three-step GTT development was planned: (1) interface data from the CTD to a PC farm and evaluate the primary vertex with a resolution better than that of the CTD standalone trigger, (2) significantly improve the track and primary vertex reconstruction by including MVD hits, and (3) extend the trigger into the forward STT region. The GTT, with its significantly improved track finding performance and quicker execution latency, replaced the CTD standalone trigger.

This paper is organized as follows. Section 2 contains an overview of the ZEUS trigger system into which the GTT is integrated and brief descriptions of the MVD, CTD and STT tracking detectors whose data are used. The conceptual design of the GTT and the hardware solutions used are described in Section 3. The event control and handling software are described in Section 4. The tracking algorithm design and track finding resolutions and efficiencies are reviewed in Sections 5 and 6. Derived trigger quantities and the impact of the GTT on the ZEUS trigger are reviewed in Section 7. The DAQ performance of the GTT is described in Section 8. Simulation, testing, and monitoring of the GTT system are described in Sections 9–11, respectively. Finally, conclusions are drawn in Section 12.

⁶ZEUS coordinate system: x —centre of the HERA ring, y —up, and z —proton beam direction.

2. ZEUS environment

2.1. Trigger

The ZEUS trigger and DAQ [3] systems were designed and implemented in 1988–1992. The high beam crossing rate, 10.4 MHz, and large event data digitization volume led to the implementation of a three-level trigger system which rejects uninteresting events as early as possible, allowing progressively more time for rejecting more complicated background events later. The trigger and DAQ are pipelined up to the Second Level Trigger (SLT) decision, thereafter the system is not event ordered.

The First Level Trigger [4] (FLT) system consists of a Global FLT (GFLT) controller connected, via ECL signal cables clocked at the beam crossing rate, to FLT units in the component frontends. Control signals exchanged between GFLT and FLT units synchronize the trigger system and its data pipelines. On receiving a bunch crossing signal the FLT units start preparing component trigger information (energy deposits, track multiplicities, etc.) which are quickly available. These are sent to the GFLT after 46 bunch crossings (4.4 μ s). The GFLT searches the information received for pre-defined patterns (slots) of quantities above threshold and if found sends a GFLT accept decision to the FLT units. On GFLT accept the components stop their pipelines to digitize and buffer the corresponding event data and additionally send improved quality trigger data to the next trigger level. Events for which no GFLT accept is generated fall off the end of the trigger pipelines and are lost. Typically above-noise FLT input rates to the GFLT are 10–200 kHz depending on beam background which are reduced to GFLT accept rates of ≤ 600 Hz.

The GFLT accept decision can be aborted by the Fast Clear [5] (FCLR) calorimeter sub-system. On GFLT accept the FCLR receives a dump of energies in electromagnetic and hadronic towers of the calorimeter, performs cluster finding, evaluates energy flow and sums and then rejects events consistent with beam-gas signatures. Processing typically takes 24 μ s, but will timeout after 36 μ s or on the arrival of a new GFLT accept. The FCLR typically aborts 10–20% of GFLT accepts and has been enabled since 2004.

The SLT [6] is designed around the INMOS transputer⁷ [7] (TP). The system consists of a Global SLT (GSLT) controller connected, via 1.8 MHz TP serial links, to SLT units in the component frontends. The component SLT units are required to send improved trigger data within ≤ 10 ms of the GFLT accept to the GSLT. The GSLT trigger decision, available ~ 1 ms after all component data are received, is sent to the event builder (EVB) which

⁷INMOS Transputers were an advanced technological development in the early 1990s when the ZEUS experiment was designed. Provided with a 32 bit 25 MHz processing unit, onboard memory, four 20 MHz serial input links for processor interconnection and a high-level parallel programming language (OCCAM), transputers were ideal for parallel processing and data transfer. Transputer development ceased in the mid 1990s.

distributes it to the component frontends via transputer links, or, as in the case of the MVD, via ethernet. For accepted events the components send their event-specific data to the EVB which builds complete events before sending them by ethernet to the Third Level Trigger (TLT). Typical GSLT accept rates are ~ 80 Hz.

The PC farm of the TLT runs event reconstruction code which is similar to the reconstruction code used offline. The event filter cuts applied reduce the GSLT output trigger rate by a factor of ~ 5 to ~ 20 Hz.

The configuration flexibility of the trigger system allows operation with $\leq 2\%$ deadtime during data taking under normal HERA background conditions. Deadtime occurs when the GFLT is held busy when component FLT units digitize and buffer data on GFLT accept, or when no more buffer space is available.

2.2. Tracking detectors

The MVD [8,9] consists of three concentric barrel layers surrounding the beampipe with radii 4.0, 8.6 and 12.3 cm and length 65.2 cm and four wheel layers in the forward direction. The polar angles, defined with respect to the proton beam direction, covered by the barrel and wheels are $22^\circ \leq \theta \leq 160^\circ$ and $7^\circ \leq \theta \leq 22^\circ$, respectively. Each layer is instrumented with pairs of single sided, $120 \mu\text{m}$ readout pitch, silicon detectors mounted together to provide 2D coordinate measurement. In the barrel region, pairs are 60×60 mm squares with strip readout oriented at 90° , in the wheels they are trapezoidal with strips tilted by $\pm 12.9^\circ$ about the radial. Barrel detector pairs are mounted on ladder support structures which traverse the z -coordinate; each ladder supports 10 pairs (five modules) and the number of ladders per layer is 4, 10 and 15, depending on the radius. The wheel structures support 14 detector pairs. A hit resolution of $\sim 30 \mu\text{m}$ is currently achieved in the barrel.

The readout of the 207360 channels is performed via HELIX-128-3.0 [10] frontend chips. On GFLT accept analog, data from the chip pipelines are serialized and the data are sent through passive copper links to 10 bit ADC VME modules [11]. The ADC system performs common noise suppression, pedestal subtraction, and strip clustering. Event cluster and strip data are stored in separate output buffers, cluster data are sent to the GTT on GFLT accept and strip data to the EVB on GSLT accept. Readout is performed via VME Power PC single board CPUs. The MVD readout is not fast enough to provide input to the FLT.

The CTD [12] consists of 72 cylindrical drift chamber layers, arranged in nine superlayers of length 205 cm and inner and outer radii of 20.1 and 76.2 cm, respectively, covering the polar angle $15^\circ \leq \theta \leq 164^\circ$. The transverse-momentum resolution for tracks traversing all layers was measured, in the pre-upgrade data taking, to be $\sigma(p_T)/p_T = 0.0058 p_T \oplus 0.0065 \oplus 0.0014/p_T$, where p_T is in GeV. Odd numbered superlayers contain drift wires that run parallel (axial) to the beamline. Even numbered

Table 1

GTT rate, latency and data size trigger/DAQ design requirements estimated in 1999

| Condition | Requirement |
|-----------------------------------|-------------------|
| GFLT accept rate | ≤ 600 Hz |
| GSLT accept rate | ≤ 100 Hz |
| GTT mean latency at GSLT | ≤ 15 ms |
| GTT mean latency at EVB | ≤ 50 ms |
| MVD mean cluster data size | ~ 4 kB/event |
| CTD mean axial + stereo data size | ~ 4 kB/event |
| CTD mean z -by-time data size | ~ 1 kB/event |
| STT mean data size | ≤ 8 kB/event |

superlayers contain wires that are oriented at a small stereo angle ($\sim \pm 5^\circ$) with respect to the z -axis. This wire arrangement allows both $r-\phi$ and z coordinates to be measured. The resolution for a track which has traversed all superlayers is $180 \mu\text{m}$ in $r\phi$ and ~ 2 mm in z . In addition to cell drift-time measurement all eight wire layers in superlayer 1, and four wires in superlayers 3 and 5 are instrumented with a z -by-timing system [13] for z hit position measurement calculated from the difference of the arrival time of the signal at each end of the sense wire. This has a post-upgrade resolution of ~ 6.5 cm, which is much cruder than the stereo result, but provides a space point available for use in the trigger algorithms. Transputer networks are used to readout [14] the FADC and TDC systems, perform data compaction and calculate trigger quantities. The CTD component provides information to the FLT, which includes crude track multiplicities evaluated from hit pattern matching in superlayers 1, 3 and 5. Extended information (multiplicities, z -vertex, track p_T and directions, etc.) is sent to the SLT by the CTD second level trigger, CTD-SLT [15], which was targeted for replacement by the GTT.

The STT [16,17] consists of four superlayers of straw drift tubes of 7.75 mm inner diameter covering the polar region $6^\circ \leq \theta \leq 23^\circ$. Each superlayer contains two planes composed of six trapezoidal sectors covering the full azimuthal angle. A sector consists of three vertical straw tubes oriented in the azimuthal direction providing an accurate measurement of the radial coordinate. The STT superlayers are mounted in groups of two between planes 1 and 2, and 3 and 4 of the ZEUS FDET detector. The readout of the STT is based on the CTD design.

The rate, latency and readout data size design requirements evaluated in 1999 for the GTT are summarized in Table 1. These values were derived from pre-HERA upgrade physics runs or were estimated for new detector components. The values measured during running in 2005 are shown in Table 6.

3. GTT hardware architecture

Calculations [18], in 1999, using expected mean data sizes and FastEthernet point-to-point connections showed

that rate and latency requirements could be satisfied by an architecture based on a PC farm interfaced to the data sources and GSLT using single-board VME computers. This result was confirmed by rate tests performed with the then available hardware without algorithm processing by transferring configurable fixed-size data blocks. By rewriting the CTD-SLT algorithm to run on a single PC, instead of a transputer network, it was also shown that a CTD-only algorithm could satisfy the latency requirements, provided data were available at the GTT computing node with acceptable latency.

The interfacing of the GTT to the ZEUS trigger and DAQ systems is shown schematically in Fig. 1. In the remainder of this section the PC farm is described, followed by a description of the data and GSLT interfaces. Table 2 lists the GTT hardware used.

3.1. PC farm

In the farm, dual CPU nodes are used due to the assumption that two essentially single-threaded tracking

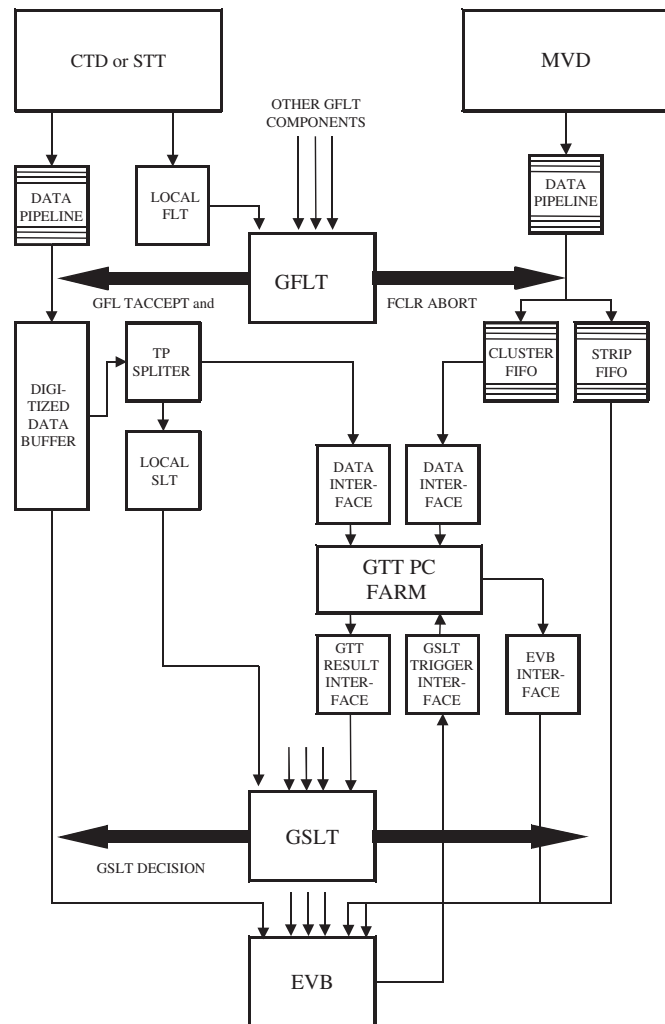


Fig. 1. Schematic showing how the GTT is interfaced to the ZEUS trigger and DAQ system.

Table 2
GTT PC farm, network and interface hardware

| Number | Hardware | Purpose |
|--------|--|--|
| 1 | DELL PowerEdge 6450 Quad 700 MHz 1 GB | NFS file server and run control |
| 1 | DELL PowerEdge 4400 Dual 1 GHz 256 MB | Credit list sync. and GSLT decision recv. |
| 12 | DELL PowerEdge 4400 Dual 1 GHz 256 MB | Algorithm processing |
| 2 | Intel Express 480 T Fast/Giga- 16Port Cu switch | Network connections |
| 5 | Motorola MVME2400 450 MHz 64 MB | MVD, CTD and STT interfaces |
| 1 | Motorola MVME2700 367 MHz 64 MB | GTT to GSLT trigger result interface |
| 4 | NIKHEF-2TP VME- Transputer modules | Transputer protocol conversion |

LynxOS 3.0.1 and Suse 8.1, later 9.2, operating systems were installed on the VME and farm CPUs. At the end of 2005 farm nodes were replaced by SuperServer 6014P dual 3 GHz machines from Supermicro Computer, Inc., which reduced the GTT's GSLT trigger latency by ~ 1.5 ms in spite of the additional MVD tracking load.

algorithms, barrel and forward, would be running simultaneously. The number of machines required was estimated by simulating the performance of the farm as a function of rate, input data size, processing time and data pusher control protocols. The control protocols tested were Round-Robin, where the order of the host list used to identify the recipient of the next event is fixed, and Credit-Based, where the host list is updated when processing finishes. Simulations showed that the Credit-Based system performance, in view of minimizing trigger latency at high rates, was superior and that ~ 10 nodes were required. An additional limit on the node count arises from the minimum number of component event buffers, 12, available for storing events before a GSLT decision is issued. Farm PCs and interface CPUs are connected to the network switches via separate network ports.

3.2. Data interface

Two distinct types, see Fig. 1, of data interfaces are used to forward event data to the GTT: embedded MVD type, where data are read directly from the frontend ADC, and parasitic CTD and STT type [19] where data are duplicated on-the-fly by a splitter transputer, specifically inserted into the detector's readout network, and sent to the interface via serial transputer links. The difference reflects the fact that the GTT was designed as part of the MVD, which does not use the now obsolete transputer in its readout. The use of the splitter transputer provides simple hardware access to the data with minimal software changes to the host transputer executable.

The implementation of the data interface types are similar allowing duplication of hardware and reuse of software. Each data interface resides in a VME crate containing a single-board computer responsible for reading

out and sending the event data via FastEthernet. Details of the data interfaces used by the GTT are given in Table 3.

The decision to use Motorola PowerPC readout CPUs was made in 2000 after feasibility studies based on the event sizes and trigger rates given in Table 1.

The use of the LynxOS operating system is essential as it provides a UNIX environment with a real-time kernel. Inhouse software including a VME driver and a library [20] providing flexible memory mapping, process synchronization, VME interrupt and DMA transfer to fully exploit the characteristics of the onboard VME chipset [21] was developed. A block diagram of the data readout and network transfer software running on the VME computers is shown in Fig. 2. A software pipeline consisting of three separate programs is responsible for receiving event information via TCP, perform the VME data readout, and transfer data to the appropriate destination computer. In MVD-type interfaces, data readout and transmission to the GTT are initiated by an interrupt generated by the last ADC module to complete data digitization. CTD- and

STT-type interfaces access detector data using NIKHEF 2TP [22] VME modules. These modules contain two transputers and a triple port memory (TPM) also accessible through the VME backplane. Each onboard transputer receives data from up to four serial links and places the data into a circular buffer. The PowerPC polls for complete events through the VME backplane and sends the detector data to the appropriate GTT node. The number of TP links used by the interface depends on the segmentation of the host detector’s transputer network where the splitter TPs are inserted. On MVD-type interfaces a second software pipeline is running to perform the final event building and data transfer for those events accepted by the GSLT. All processes on the PowerPC are synchronized to each other via system semaphores and different execution priorities are set to minimize the latencies. Details on the readout latencies are reviewed in Section 8.

3.3. GSLT and EVB interfaces

The GTT algorithm result is sent to the GSLT via a Motorola PowerPC VMEbus CPU, which writes the result to NIKHEF-2TP TPM from where it is sent by a transputer to the GSLT. As the order of trigger results arriving at the GSLT is strictly sequential, the interface is required to order the results from the different GTTs according to the event number before sending.

The GSLT sends its event trigger decision to the EVB via a transputer link where it is fanned out to the detector component subsystems through transputer links, or, as in the case of the MVD, via ethernet. Not all detector components send a trigger result to the GSLT, but all need to receive the GSLT decision. At the MVD the result is received by a process running on a Linux PC and is relayed

Table 3
Summary of GTT component data interfaces

| Name | Data handled | Readout modules |
|------|-----------------------|-----------------------------|
| MVD0 | Upper barrel clusters | 10 ADCs boards |
| MVD1 | Lower barrel clusters | 10 ADCs boards |
| MVD2 | Wheel clusters | 8 ADCs boards |
| CTDF | Stereo and axial hits | 2 NIKHEF-2TP 16 TP links |
| CTDZ | z-by-time hits | 2 NIKHEF-2TP 16 TP links |
| STTF | Straw tube hits | 1 NIKHEF-2TP 8 TP links |

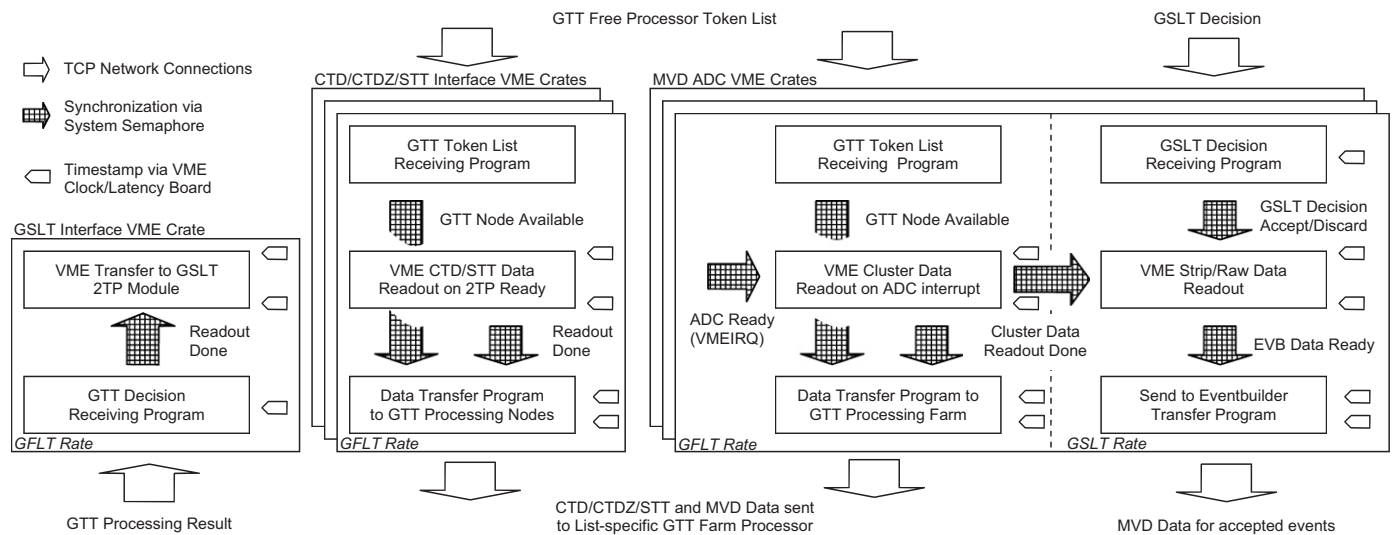


Fig. 2. Block diagram showing the functionality of the VME data and the GSLT interfaces. For the data interfaces the software pipeline consisting of three (from top to bottom) programs is shown. On MVD type interfaces a second software pipeline for the data to be sent on GSLT accept is also shown. Detector data and monitoring information are exchanged via contiguous non-paged shared memory. Synchronization among the different processes is performed using system semaphores. Latency measures are available to precisely monitor the performance.

to the MVD data interfaces and GTT node which processed the event. On accept, the data interfaces send strip data and the GTT PC sends cluster and algorithm calculation results and monitoring data to the EVB interface. During normal running cluster data were sent every 100th event which reduced the network load whilst allowing a check of the cluster information reconstruction from strip data required during monitoring, see Section 11.

The EVB interface runs on a PC and waits for event data fragments associated with GSLT accepted events. These are formatted into data banks and send to the EVB via TCP.

4. Algorithm software environment

In this section the process environment [23] within which the track finding algorithms run is described. The environment is required to receive event data from the data interfaces on GFLT accept, perform track finding, calculate trigger quantities and send the results to the GSLT. When a GSLT accept decision for the event is received the environment sends event data and trigger result banks to the EVB before deleting the event. Rejected events are marked for deletion. The control and data messages sent between the environment and the data, GSLT and EVB interfaces are shown in Fig. 3.

Since many instances are running concurrently it is necessary to define which event is sent to which environment by the data interfaces. This has been implemented using an ordered list of credits. Each environment is assigned, during run configuration, a unique credit number and the complete credit list is transferred to the data interfaces by a list synchronizing process. During the run data interfaces send event data to the environment whose credit is removed from the head of the list. Environment lists are updated, at their tail, by credit update messages forwarded by the synchronizer from environments which have finished event processing.

To perform the above tasks the environment contains seven functionally different thread types.

- The main thread which performs all startup operations (allocates event buffers, initializes histogram and statistics monitoring shared memory, builds network communication links to the data, GSLT and EVB interfaces, and finally creates all worker threads) before entering the event synchronization loop.
- Data interface threads which receive event data sent from an MVD0, MVD1, MVD2, CTDZ, CTDZ or STTF data interface.
- Decoding threads which wait for completion of single or sets of data interfaces before starting.

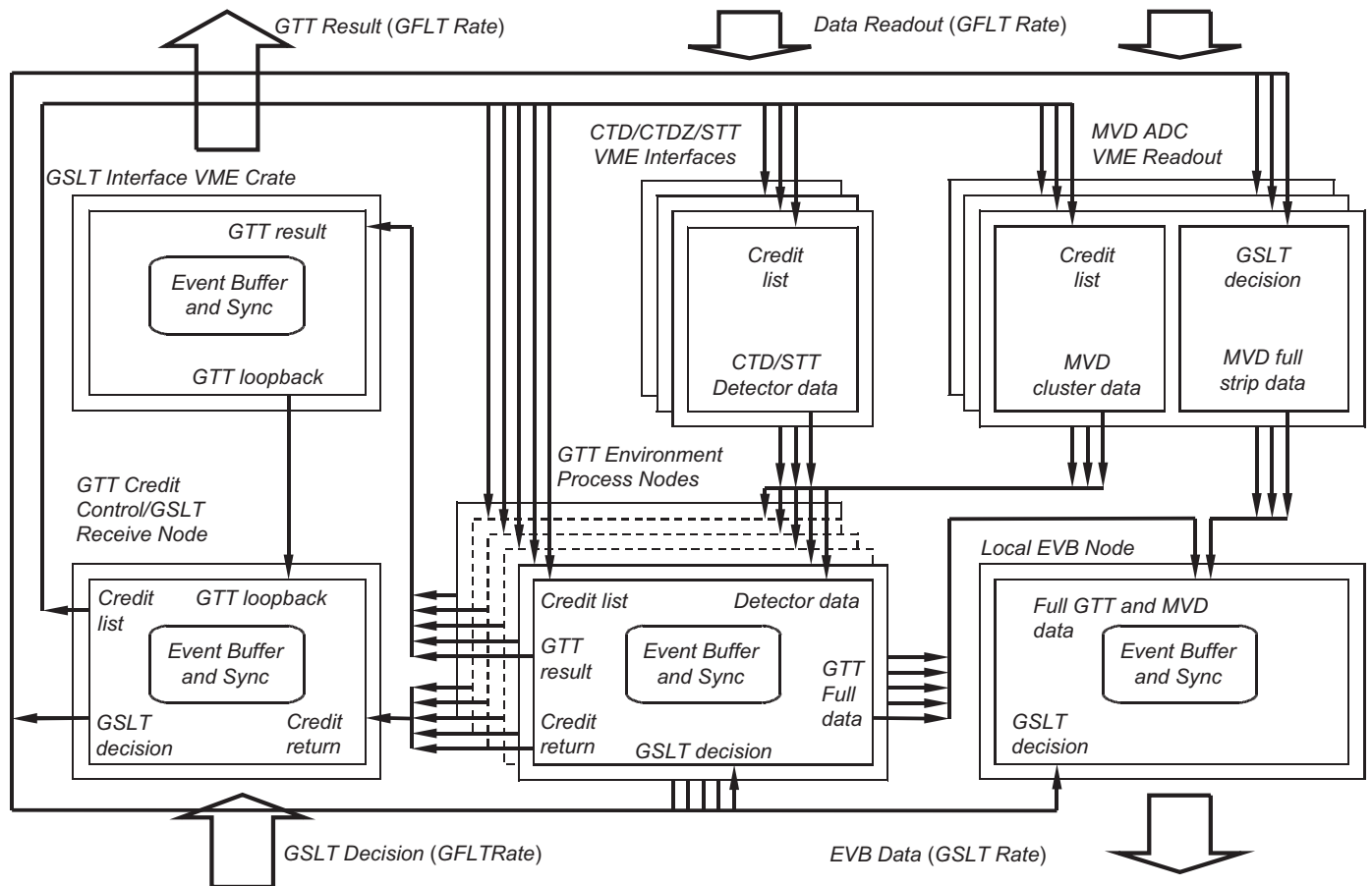


Fig. 3. Block diagram showing the data and control logic of the GTT.

- Tracking algorithm threads which run a barrel or forward algorithm on all, or a subset, of the data received.
- A GSLT result thread which receives the GSLT decision and, if an accept, sends MVD cluster and GTT algorithm result banks to the EVB interface.
- A watchdog thread which imposes, if required, a time-limit on sending the GTT result to the GSLT.
- A shutdown thread which catches signals sent to the environment and manages its orderly shutdown.

Which data interface and algorithm threads are started is determined by the run configuration. Threads are started once during the lifetime of the environment. Histogram and statistics monitoring is delegated to independent server processes by storing information in shared memory.

Synchronization functions are used to wait, or test, for operation completion of OR's or AND's of single or multiple threads. These functions are used by the environment to efficiently organize waits for completion of data arrival, decoding and algorithm processing. Thread operation completes when the thread finishes its work (data receiving, decoding or algorithm processing) on the event data which it processes and the data product (read or decoded data or algorithm result banks) are buffered in memory.

The watchdog is used to control the tail of the GTT latency distribution by generating a timeout result after 30 ms of receiving the first data interface data, which is sent to the GSLT. GSLT filters using GTT results pass timeout events. Typically 0.3% of events timeout, due primarily to late arrival of interface data.

Error recovery is performed only for the case where a tracking algorithm thread does not finish within 15 s. In this case a timeout result, see above, is sent and the environment's credit is permanently removed from the credit list preventing the environment being sent events. If the failed event is accepted by the GSLT, empty result banks are sent to the EVB. Algorithm thread deaths are infrequent, normally thread errors are removed during the offline debugging phase or during online simulation runs made before committing a new version. In this respect a useful feature of the environment is the ability to dump a subset of events, including the last event seen in a run, to local disk during running, which allows offline debugging.

5. GTT algorithm

The principal design challenge in developing the GTT tracking algorithm was how to efficiently reconstruct tracks within the millisecond trigger time budget available at the SLT. In 2000, the TLT mean processing time for track finding in the CTD data was ~ 100 ms on an 800 MHz CPU, precluding its use in the GTT. Instead the distributed algorithm of the CTD-SLT [15] was used as a starting point. The algorithm was rewritten to run on a single host

and to include CTD stereo and MVD data, in addition to the CTD axial and z -by-timing data used in the original CTD-SLT algorithm. The processing steps performed by the algorithm are:

- Axial tracking:
 - all (single superlayer) track segments in the axial and stereo layers of the CTD are found;
 - tracks are fitted in $r-\phi$.
- z tracking and vertex identification:
 - z -by-timing hits are matched to the axial ($r-\phi$) tracks and fitted by a straight line in $z-s$ where s is the track path length in $r-\phi$;
 - 3D tracks are improved using stereo segments matched to tracks;
 - the primary vertex is calculated;
 - the stereo segment matching is repeated using the vertex as a constraint and the z -tracks are refitted;
 - the primary vertex is recalculated.
- MVD matching and vertex refinement:
 - $r-\phi$ MVD hit matching is performed and $r-\phi$ tracks are refitted;
 - z MVD hit matching is performed and z -tracks are refitted;
 - the primary vertex is recalculated.
- Computation of trigger quantities.

In the following sections the algorithm stages used for processing the different CTD (axial, z -by-timing and stereo) and MVD (cluster) data types are described followed by a review of the vertex finding procedure. Note that the information from the different stages of algorithm processing are stored uniformly with appropriate weights for the resolution of the hits which allows the multiple vertex and track refit passes applied above. Finally, configuration and alignment data handling is addressed.

Algorithm performance was tuned using samples of clean, low multiplicity events containing at least one track reaching superlayer 9 in the CTD.

5.1. CTD axial and stereo segment finding

Segment finding in the CTD lies at the heart of the algorithm, since it provides the basic information from which all tracks are initially found. The segment finder operates on the unsigned drift-time data and wire numbers from each CTD cell grouping the hits together into short straight track segments. Since the drift-time information is unsigned it is not known on which side of the wire plane the hits lie and each contributes a real and a ghost candidate segment, as shown in Fig. 4. Considering real and ghost segments during track finding more than doubles the reconstruction time; to avoid this time penalty only the segment pointing closest to the origin, after hit assignment, is used. This choice is motivated by the fact that the wire plane for each cell is oriented at 45° with respect to the cell radius and the gradients of real and ghost candidate

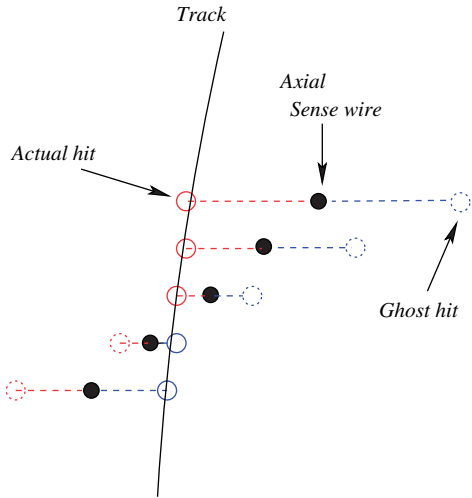


Fig. 4. The left-right drift-time ambiguity in the CTD.

segments will usually be considerably different. This choice has a high efficiency for identifying the correct candidate for high p_T tracks, but for lower p_T tracks, it leads to a charge asymmetry due to the ϕ asymmetry of the CTD geometry. The same association procedure is applied to both axial and stereo superlayers. However, since the ϕ —and hence z —position of a stereo segment is not known until the segment has been matched to a track, the dependence of the gradient of the stereo segment on the z position is ignored at this stage.

The assignment of hits in a cell to segments is performed as follows:

- Wire numbers and drift times are extracted and a global time offset (T0) value is subtracted.
- Seed pairs of hits, predominantly on adjacent wires, are found. If no seed pair is found the cell is abandoned.
- Using a linear extrapolation of the seed pair, the drift time on the next wire in increasing wire number is predicted. If a hit lying within the segment search road is found it is added to the candidate segment and the hit together with the adjacent hit from the seed pair is used as a fresh seed pair.
- This is repeated until the cell boundary is reached or no more hits within the search road are found. The procedure then returns to the original seed pair and searches for segment hits in decreasing wire number until the cell boundary is reached or no more hits lie within the search road.
- If the number of hits in the completed segment is greater than 2, the segment is deemed complete and the hits used are removed from the list of available hits for this cell and the segment finder begins searching for a fresh seed pair in the cell.
- As soon as the number of hits for the cell falls below three the cell is abandoned, since no more segments can be found. Additionally segment finding stops as soon as

the number of segments exceeds the segment count limit for a cell.

Since the segment finder processes all cells which contain data, the time consumption during segment finding is sensitive to the maximum number of segments found per cell. The segment finder algorithm limits the number of segments found in a cell to four to reduce the processing time in very busy events which often result from beam-gas interactions rather than ep collisions.

5.2. Identifying CTD r - ϕ tracks

The search for r - ϕ tracks is performed by matching CTD axial segments whose position and gradient are consistent with tracks from the interaction region. Prior to the HERA upgrade, the beamline position in the xy plane coincided with the origin of the ZEUS coordinate frame. Following the upgrade, the nominal beamline position was shifted by ~ 1 cm in x , hence before the track finding is performed, the coordinates of the segments are transformed by the known beamline position so that the origin and the beamline position coincide. After the track finding, the track parameters are translated back to the ZEUS coordinate frame.

For each segment on a track the angle of the segment, χ , with respect to the segment radius vector should be consistent with a single value of the curvature, $\kappa = 1/R_0$, where R_0 is the track radius.

Track finding starts using seed segments in the outer superlayers, where spatial separation of candidate segments is largest. It continues inwards, see Fig. 5, by associating segments that have consistent azimuthal angle, ϕ , and the correct gradient angle, χ , for a track from the interaction region using the steps described below.

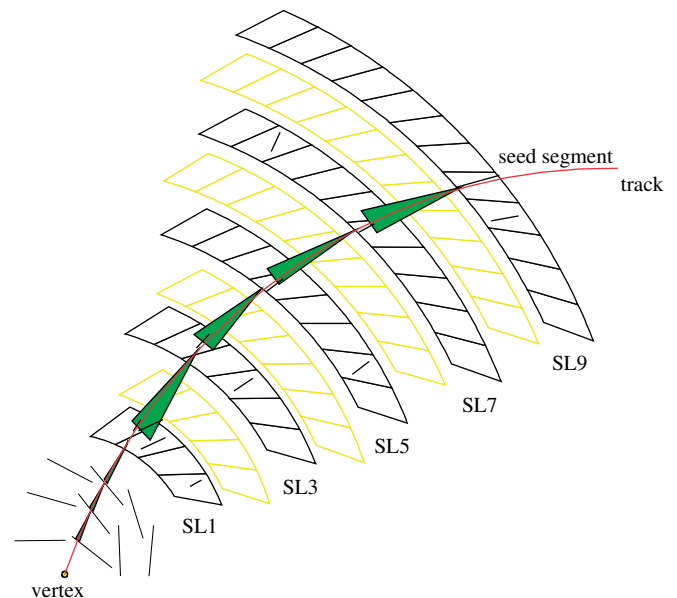


Fig. 5. Schematic showing the search roads used during r - ϕ track finding.

- A seed segment not yet matched to a track is found.
- The expected azimuthal position of the hit in the next innermost axial superlayer, $\phi_{\text{ext}}(\text{SL})$, is calculated using the linear extrapolation

$$\phi_{\text{ext}}(\text{SL}) = \phi(\text{SL} + 2) + \chi\omega(\text{SL})$$

where the values of the parameters, $\omega(\text{SL})$, used in the parameterization are those from the fit used for the CTD-SLT. The segment angle with respect to the segment radius, χ , segment radius vector, length r , is calculated and used to calculate the segment curvature κ using

$$\kappa = \frac{1}{R_0} = \frac{2 \sin \chi}{r}.$$

If segments have already been found in superlayers 3, 5 and 7 a quadratic approximation of the track trajectory into superlayer 1 is made using

$$\phi_{\text{ext}}(\text{SL1}) = \phi(\text{SL3}) - 3\phi(\text{SL5}) + 3\phi(\text{SL7}).$$

Since all segments belonging to a track should have the same curvature, segment candidates are added to a track if the ϕ position with respect to the predicted position, $\Delta\phi$, is within 3.44° and the curvature, κ , is within 0.04 of the previous segment for the track. Up to four candidate segments can be initially matched in each superlayer, the one with the smallest value of $\Delta\kappa + 10\Delta\phi$ being retained. The tolerance values used here are those obtained from optimizing the CTD-SLT algorithm.

- As shown in Fig. 5, a segment can be split across a cell boundary. If only one of the segments is matched to a track, then the additional segment can give rise to a second track. To reduce this possibility, a second segment can also be matched to this track if $\Delta\phi \leq 1^\circ$ and $\Delta\chi \leq 15^\circ$.
- The segment last matched is then used as a fresh seed and the matching proceeds into the next inner axial superlayer until at least one segment is found in superlayer 1 and at least one in a higher superlayer, or one segment in superlayer 3 and all higher superlayers with at most one gap in one of these higher superlayers. At this point the segments are added to the track and removed from the list of segment candidates.

The efficiency for finding tracks is not significantly affected by the order of segment processing since the occupancy in superlayer 9 is low. The matching of stereo segments to the tracks can in principle be affected by the order in which the tracks are processed; however, trials where stereo segments matched to a track were not removed from the list of segments for subsequent matching to a track did not cause a significant improvement in the efficiency.

Once the segment matching is complete, track parameters are calculated assuming a circle in r - ϕ . The parameters for a track not constrained to the origin are illustrated in Fig. 6. In the special case of a track constrained to the origin, the distance, d_0 , from the origin

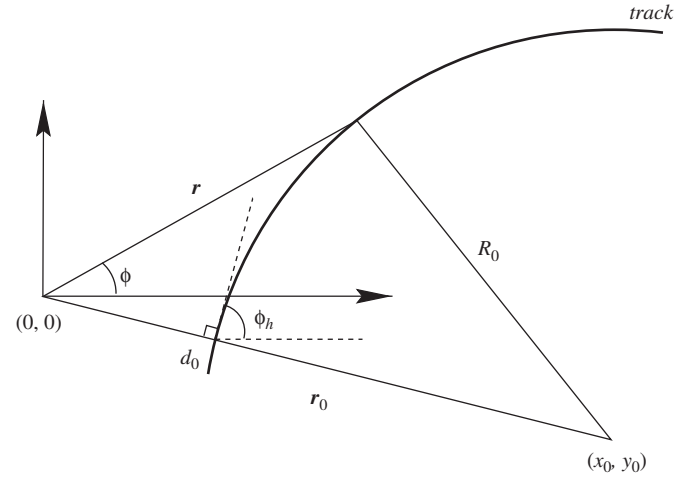


Fig. 6. r - ϕ track parameters.

to the point of closest approach of the track to the origin is zero and r_0 , the distance from the origin to the circle centre becomes identical to the circle radius, R_0 .

For CTD, only r - ϕ track finding a fast circle fit [15,24] constrained to the origin in xy is used. The circle radius, R_0 , is calculated using

$$R_0^2 = \frac{(b^2 + e^2)c^2 + (a^2 + e^2)d^2 - 2cdef}{4(ab - e^2)^2} \quad (1)$$

where the variables a , b , c , d , e and f are calculated summing over the axial segments, i , on the track, with positions x_i , y_i , radii r_i , and azimuthal angle ϕ_i

$$\begin{aligned} a &= \sum_i x_i^2, & b &= \sum_i y_i^2 \\ c &= \sum_i r_i^2 x_i, & d &= \sum_i r_i^2 y_i \\ e &= \sum_i x_i y_i, & f &= \sum_i r_i^2. \end{aligned} \quad (2)$$

The circle centre, (x_0, y_0) , is obtained using

$$\begin{aligned} x_0 &= \frac{bc - ed}{2(ab - e^2)} \\ y_0 &= \frac{ad - ec}{2(ab - e^2)}. \end{aligned} \quad (3)$$

The track charge, Q , is obtained from the sign of the cross-product of the vector from the circle centre to the position of the outermost axial segment with the track tangent at the origin with angle ϕ_h . This tangent is obtained from the perpendicular to the vector from the origin to the circle centre in the xy plane.

The addition of MVD r - ϕ hits to the track after the MVD hit matching is achieved by including the MVD hit r - ϕ positions in the above fit, but with an appropriately higher weight to represent the higher spatial resolution of the MVD. The pattern recognition to identify which hits should be added is discussed in Section 5.4.

5.3. Identifying CTD z -tracks

The CTD provides information on the z position of hits from the z -by-timing and stereo systems. The full 3D track is found by adding the hit to a simple linear fit in the z position and the path length, s , along the r - ϕ track trajectory in r - ϕ with an appropriate weight to reflect the different resolutions of the z -by-timing and stereo hits.

The z -by-timing hits are added first to aid the matching of the stereo segments, as described below.

5.3.1. z -by-timing track finding

The hits from the z -by-timing system are true space points, with both r - ϕ (drift time) and z digitization information. Therefore, hits on wires in axial segments in superlayers 1, 3, and 5 which are instrumented with z -by-timing electronics can be matched to r - ϕ tracks directly by comparing the drift time from the z -by-timing system, t_{digi} , with that expected from the axial FADC system for the same wire, t_{hit} . The z -by-timing and FADC systems use different digitization clocks; hence for comparison, the z -by-timing time, t_{digi} , measured in 48 ns bins, is first corrected for the global T0 of the z -by-timing system and converted to the 2.4 ns units of the FADC system. Unphysical values are considered to be noise and are removed. A match is found if

$$|t_{\text{hit}} - 30 < t_{\text{digi}} < |t_{\text{hit}} + 6$$

where the times t_{hit} and t_{digi} here are both measured in the (2.4 ns) units of the FADC system. The z position of the hit is then calculated from the corresponding digitized z position from the electronics, z_{digi} , using a fifth-order polynomial parameterization which corrects for small non-linearities in the z -by-timing electronics. The path length along the track, s , between the point on the track at the calculated radius of the hit and the point of closest approach of the track to the beamline is calculated. These z and s values are then added to the linear fit for the track. The matching process repeats until all z -by-timing hits on wires consistent with the track are exhausted after which the fit is performed.

The track fit in z - s results in the track gradient and z -position of the track at the point of closest approach to the beam line. This gradient is then converted to the track polar angle, θ , with respect to the beamline. The z position is referred to as the track vertex.

The mean number of z -by-timing hits matched to tracks traversing all superlayers is around 6.

5.3.2. Stereo track finding

The design of the CTD is such that the ends of each stereo wire at the front and back of the CTD are displaced by four cells. This means that the angle spanned by each wire in ϕ is as much as 36° in superlayer 2 and 16° in superlayer 8. It is possible for any stereo hit to potentially match any track passing through the large (four cell) angle in ϕ spanned by the wire.

To calculate the z position of a stereo hit, the ϕ distance along the wire, $\Delta\phi$, must be found by subtracting the hit drift vector from the calculated ϕ position of the track at the radius of the hit as shown in Fig. 7. The fraction of the total ϕ angle spanned by the wire represented by $\Delta\phi/\phi$ is then used to calculate the fraction of the distance in z along the wire. For a ghost hit, the calculated $\Delta\phi$ would be considerably different, leading to a spurious z position. In the inner superlayers, the track occupancy is large; hence, the number of possible track-hit matching combinations also becomes large. Thus, to reduce the latency, only the end points of segments in the stereo layers found by the segment finder are used. Since the segment finder uses the segment gradient to resolve the ghost ambiguity, only two hit positions for each segment have to be matched to each track rather than eight separate hits and eight ghosts.

The algorithm for the stereo segment matching is recursive and matches segments to tracks using a tree structure. For any matching segment in a stereo superlayer, it tries to follow the track into the innermost superlayer matching one segment from each layer. All combinations of segments that can match a given r - ϕ track are computed and give rise to a number of candidate z -tracks. To avoid unnecessary calculation, computed segment positions when matched to a r - ϕ track are saved in a segment cache for this track.

The association of stereo segments with a track is performed as follows:

- All stereo segments from cells which overlap with the track in the outer stereo superlayer, where the spatial separation of tracks is highest, are considered as seed segments for matching. If no segments in the outer layer match, the next stereo superlayer in is considered.
- Taking each seed segment at a time, the intersection of the segment end points with the track is found, and the

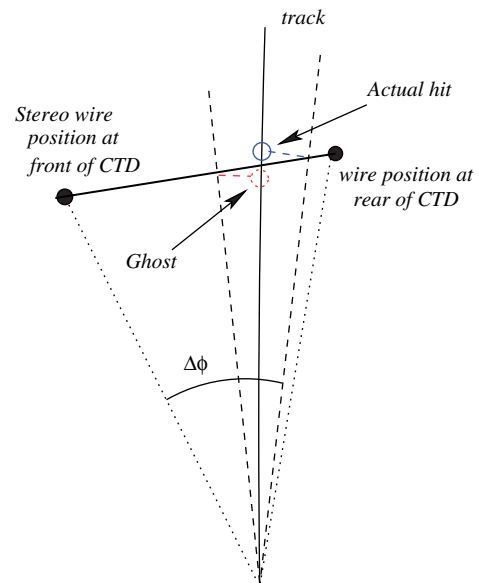


Fig. 7. The left-right stereo drift-time ambiguity in the CTD.

position in z and s , the path length along the track in the transverse plane, are determined.

The intersection of the track with the hit must be calculated considering the drift displacement of the hit with respect to the wire. This is performed using a custom implementation of the algorithm from the ZEUS offline pattern recognition package, VCTRAK.

The calculated positions and the track identifier for the track to which this segment has been matched are cached.

- The z -track parameters (gradient and intercept) are calculated. The intercept is the z position at the track's distance of closest approach to the origin in x and y . The position of the track extrapolated into the next stereo inner layer is then calculated.
- If not already in the segment cache, the z positions of the next unmatched segment in the next inner layer are calculated by matching to the track as above and added to the cache. To be used to extend the track inwards, the computed positions for the segment must lie both within the CTD and within a window of z of the extrapolated position and have a realistic gradient.

If no matching segment is found, the algorithm skips the superlayer and attempts to match segments in the next inner layer. If no matching segments are found the track candidate for this combination of segments is abandoned.

- Segment association continues inwards, with the fit being recalculated at each layer for each possible matched segment, until a track candidate with segments in all layers from the outer layer is found. The track must have a matching segment in superlayer 2 with at most one missing superlayer. The track parameters are then calculated and the list of matched segments for this track is cached. The χ^2 for the track candidate is calculated, defined as the sum of the squares of the residuals of the segment end points divided by the number of segment end points and this added to the cache of candidate z -tracks for this r - ϕ track.
- The algorithm continues finding new potential z -tracks for this r - ϕ track, until all possible combinations of segments matching to this track have been calculated. The z -track candidate with the lowest χ^2 per segment end point is then chosen at which point the segments from this candidate are flagged as belonging to this track and the parameters for this z -track candidate are added to the r - ϕ track to give the full 3D track. If the track has a single stereo segment, then the χ^2 will be zero if there are no matching z -by-timing hits. These single segment candidates are considered only if no other candidate z -tracks are available.

Fig. 8 shows the z -residuals of the segment end points with respect to the intermediate fits extrapolated into the appropriate superlayers during the stereo matching. As more segments are matched the residuals become smaller as

expected. Similarly, for longer tracks the predicted position improves.

Once the z -track finding has taken place for all r - ϕ tracks, the vertex is found (Section 5.5) and the stereo matching repeated with the vertex as a constraint to improve the pattern recognition. This increases the track finding efficiency by around 10% in events where the track multiplicity is high and improves the pointing for tracks with a single stereo segment.

5.4. MVD tracking

Since the MVD hits from both the r - ϕ and z wafers (detectors) within an MVD half module are multiplexed together, it is not known which are z and which are r - ϕ hits. Consequently all hits are considered as potential r - ϕ or z hit candidates.

5.4.1. Including MVD r - ϕ hits

Association of MVD hits with CTD tracks starts in the MVD outer layer where the extrapolation from the CTD track is least uncertain and occupancies are lowest. The hit matching proceeds as follows:

- The track's ϕ position in the MVD layer is calculated, and those ladders within a ϕ window are marked for use. Because ladder edges within a layer overlap it is possible for a track to have up to six MVD hits in r - ϕ . The track's z position at the layer radius is calculated, and only those modules from the matching ladders in r - ϕ which match within a window of z of the predicted position are considered.
- All unmatched hits on these modules are considered as potential r - ϕ hits. For each hit the difference between the ϕ of the track at the hit radius and the ϕ of the hit, $\Delta\phi$, is calculated. The closest hit from each matching module in the ladder, which minimizes $\Delta\phi$, is then matched to the track if $\Delta\phi$ is less than the layer-specific matching criterion (0.025–0.003 rad). Matched hits are removed from the hit candidate list. Note that since the angle $\Delta\phi$ is used only to sort the candidate hits, to save time, the predicted ϕ position is not corrected for the variation in radius across the wafer.
- When a maximum of two hits from this layer (one per ladder for overlapping ladders) have been found, these are added to the track and the track parameters are recalculated using the constrained fit. MVD hits are included with a higher weight.
- The algorithm then proceeds to the next innermost MVD layer, repeating the hit matching until all layers have been considered.

The measured $\Delta\phi$ residual distributions, for each MVD layer, are shown in Fig. 9. The improvement in the predicted ϕ positions once the first MVD hit has been matched to the track is evident. The widths and means of the residual distributions are tabulated in Table 4.

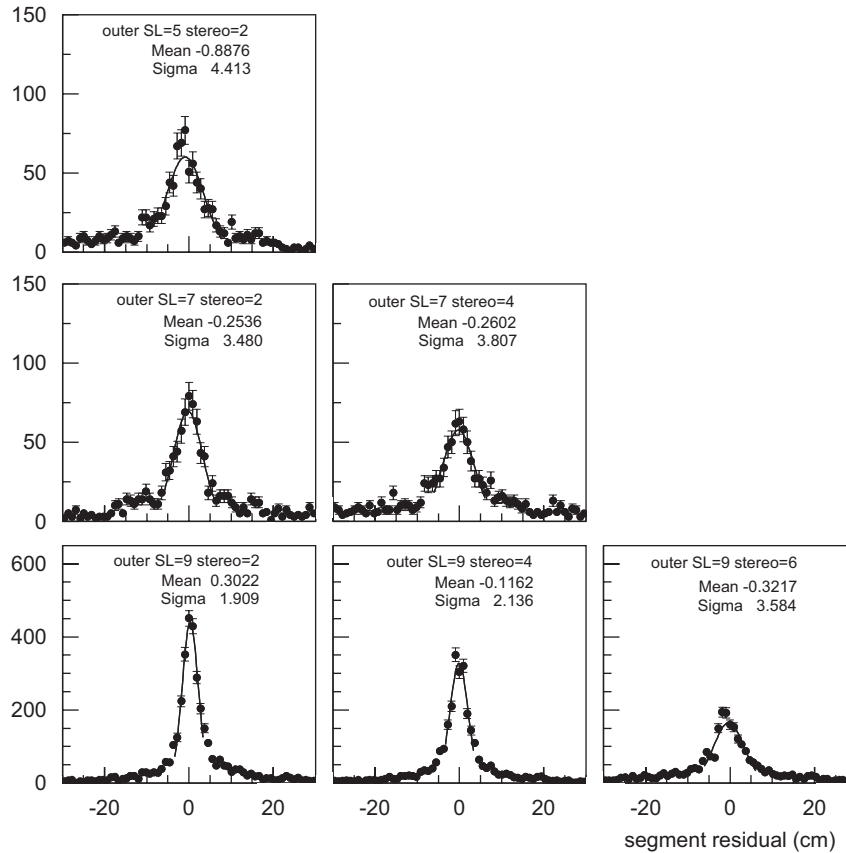


Fig. 8. The z -segment residuals of the segment end points with respect to the extrapolated track position during the stereo segment finding. From the top: tracks with outer superlayer 5, 7 and 9, respectively; from left to right: the extrapolated positions in superlayer 2, 4 and 6 (see text).

The increase in the ϕ residual seen for the third MVD hit is a geometrical effect driven by the fixed strip width and decreased radius of the inner MVD layer.

5.4.2. Including MVD z hits

Once MVD r - ϕ hit matching is completed, MVD z -hits are matched to the track as follows:

- Hit matching starts in the outer MVD layer using unassigned hits on those MVD modules containing matched hits for this track.
- Because each module consists of two wafers glued on top of each other, each r - ϕ hit from the first wafer must have an associated z hit from the second wafer. If an MVD r - ϕ hit has been previously matched to the track, the algorithm looks first for unmatched hits in the corresponding z -wafer belonging to the same module.
- If no r - ϕ hits have been found, the algorithm looks for unmatched hits in the other possible z -wafer and calculates the track intersection with these wafers.
- All unmatched hits are then compared with the predicted z -position, and the hit closest to the track is matched if it lies within a window of z around the predicted position. This window ranges from 0.3 to 3 cm depending on the quality of the CTD track.

- The hit is then added to the track with its corresponding weight and the fit in s and z is recalculated.
- The algorithm then proceeds to the next innermost MVD layer, repeating the hit matching until all layers are exhausted.

Fig. 10 shows the residual of the z point of the MVD z hit candidate with respect to the predicted track position during track matching. As with the first ϕ position, the improvement in the predicted z positions once the first MVD hit has been matched to the track is evident. The widths and means of the residual distributions are tabulated in Table 4.

5.5. The primary z -vertex algorithm

The z position of the primary vertex is a principal trigger quantity generated by the GTT. It is recalculated at several stages during algorithm processing. The x and y coordinates are not calculated since the primary interaction is assumed to originate from the beamline.

The primary z -vertex is calculated using an overlapping binning algorithm with bin width 13 cm. For each 3D track found a weight, w_i , reflecting the quality of the track is calculated by summing the weights of stereo segment end

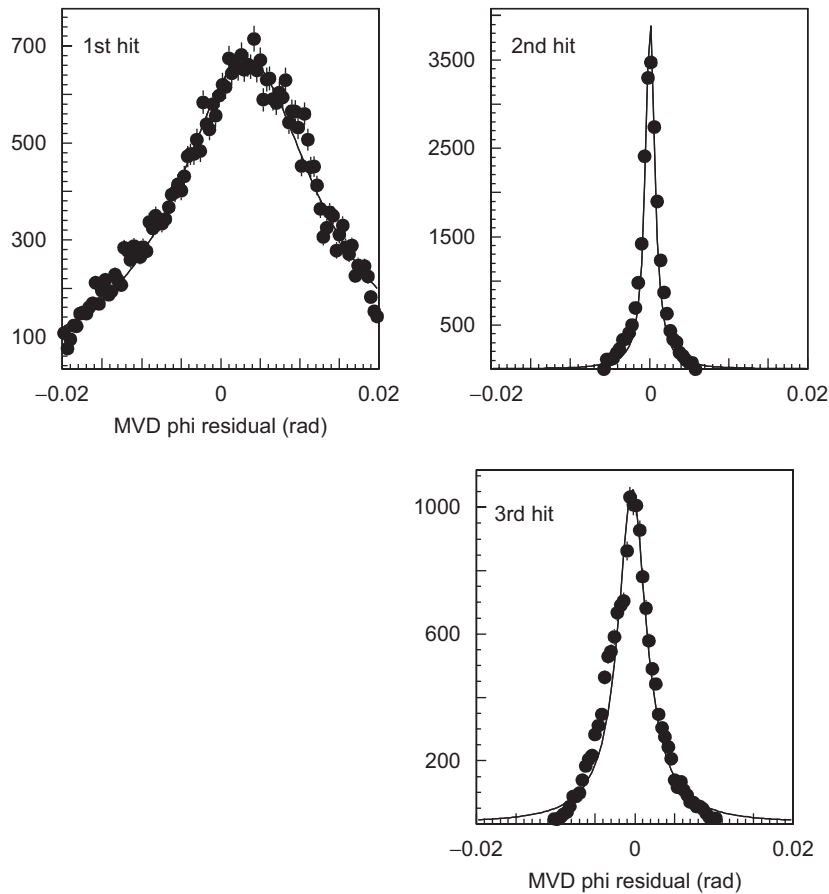


Fig. 9. The ϕ residuals of the closest MVD r - ϕ hits to the extrapolated track during the MVD r - ϕ matching stage.

Table 4

Means and widths of z and ϕ residuals of matched MVD hits measured during matching to the extrapolated track

| Matched MVD hit: | First | Second | Third |
|---|-------|--------|--------|
| $\bar{\phi}$ ($\times 10^{-3}$ rad) | 2.7 | 0.11 | -0.52 |
| σ_{ϕ} ($\times 10^{-3}$ rad) | 11.0 | 1.35 | 3.20 |
| \bar{z} (mm) | 30.0 | 0.068 | -0.048 |
| σ_z (mm) | 16.0 | 1.9 | 2.1 |

points, z -by-timing hits and MVD z -hits. The weight squared is then added to the two overlapping bins in which the track vertex lies. The square is used to give more weight to high-quality tracks and increase the robustness of the fit. After processing all tracks, the bin with the highest sum of squared weights is used to derive a prototype vertex position for the tracks in that bin using

$$z_{\text{proto}} = \frac{\sum_{i \in \text{bin}} z_i w_i^2}{\sum_{i \in \text{bin}} w_i^2}. \quad (4)$$

New running sums of the weights and weighted z positions are then calculated for all tracks in the most probable bin and the two adjacent, non-overlapping bins lying within

6.7 cm of this provisional vertex position, and the final vertex position is recalculated.

This primary vertex calculation is found to be very stable against the presence of outlier tracks. For the reconstructed vertex the sum of the weights and the root-mean-square can be used to estimate the quality of the vertex. Additionally, the fraction of the sum of the weights assigned to the vertex with respect to the total sum of weights gives a measure of the likelihood that the vertex has been correctly identified.

5.6. Alignment and parameter configuration

Optimal algorithm performance can be guaranteed only when all external variables used during track reconstruction are known accurately. The geometry of the CTD and MVD elements is defined to be time independent and has been implemented using lookup tables. Second-order effects, e.g. wire sag and non-planar wafers, are not corrected for. Time-dependent constants such as CTD drift velocity, Lorentz angle and beamspot position (used during the constrained track fit) are sufficiently slowly changing to allow the values to be determined offline and inserted into the GTT as run startup parameters. This is achieved using a simple database, referenced by run

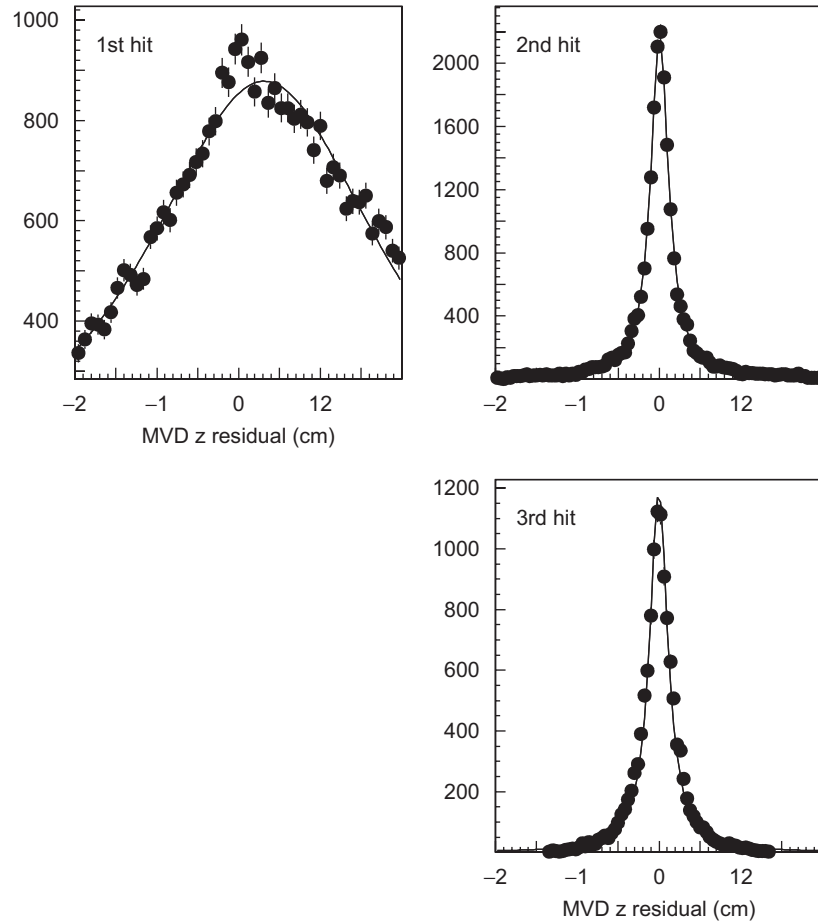


Fig. 10. The z residuals of the closest MVD z -hits to the extrapolated track during the MVD z matching stage.

number, which assures that the appropriate value of the configuration is used both online and offline when running on real or simulated data.

6. Algorithm performance

For backwards compatibility with the CTD-SLT based triggers the GTT generates trigger quantities, see Section 7, from tracks at two distinct stages in their reconstruction; CTD-only (using only CTD hits) and CTD + MVD (using both CTD and MVD hits).

In this section tracking and vertex finding resolutions and efficiencies of the algorithm for tracks from both stages of reconstruction are evaluated using events recorded during data taking. Finally, a comparison of the GTT tracking performance with those of the CTD-SLT and offline tracking reconstruction, based on simulated Monte Carlo events, is reviewed.

6.1. Track finding efficiency and resolution

The track finding resolutions and efficiencies have been evaluated using a sample of 6.8×10^6 dijet photoproduction events which were selected from events written to tape.

Events were required to have an offline vertex with $|z| < 50$ cm and a transverse distance from the nominal beam position of less than 1 cm. All offline tracks with $p_T > 200$ MeV and assigned to the primary vertex were considered.

6.1.1. Track matching

In comparison with the GTT, the ZEUS offline tracking algorithm does extensive combinatorial pattern recognition and uses a sophisticated calculation of the expected drift distance which takes into account the track angle and z position, the position within the drift cell, non-linearities in the time–distance relation, and pulse-height effects. The offline algorithm also uses the drift times for each wire rather than combining all wires in a superlayer into a segment and applies wire-by-wire T_0 calibration. In addition, the offline algorithm also allows for energy loss along the track by fitting the transverse trajectory as a set of interlinked circles rather than as a single circle as is done in the GTT. Also the offline tracking uses a more detailed description of the alignment of the MVD. As a result, the offline algorithm is more efficient and more precise than the GTT, especially for low-momentum tracks.

In order to evaluate the performance of the GTT algorithm on data, an algorithm which matched GTT tracks to offline tracks was employed. For each GTT-offline track pair, a χ^2 was calculated by comparing the measured residuals in ϕ , η , and Q/p_T with parameterized resolutions, taking into account correlations. Since the GTT measurement of Q/p_T is biased towards smaller absolute values (e.g. higher p_T), an estimation of the bias as a function of the offline Q/p_T was subtracted from the GTT-offline residual before χ^2 was computed. The error matrices were determined as a function of the length of the offline tracks. Offline-GTT track pairs for which $\text{prob}(\chi^2, 3) > 10^{-3}$ were considered as candidate matches. For an event, a set of matched track pairs was selected from the candidate matches, such that no track was used more than once. The optimal matched set of n track pairs is the one with the smallest value of $\chi^2_{\text{event}}(n)$, the sum of the χ^2 's for each track pair. The matching algorithm determined the optimal matching for the largest value of n for which $\text{prob}(\chi^2_{\text{event}}(n), 3n) > 0.01$.

The choice of χ^2 as a measure of the compatibility of an offline track with a GTT track is sensible if the differences in the track parameters are Gaussian distributed. This assumption holds approximately for long tracks and not so well for shorter tracks, where resolutions are large compared with the separations between tracks and purity is low. Therefore, for shorter tracks there is some subjectivity in the choice of input track parameter resolutions. Choosing a larger value for the resolution used in computing χ^2 will increase the likelihood that an offline track will be matched to a GTT track (higher efficiency), while increasing the observed GTT resolution. In contrast, choosing a smaller value for the input resolution will result in a lower measured efficiency and better resolution.

6.1.2. Track resolution

Fig. 11 shows the distributions of the bias-corrected residuals of full length—those ending in superlayer 9—GTT tracks which match offline tracks in ϕ , Q/p_T , η , and z .

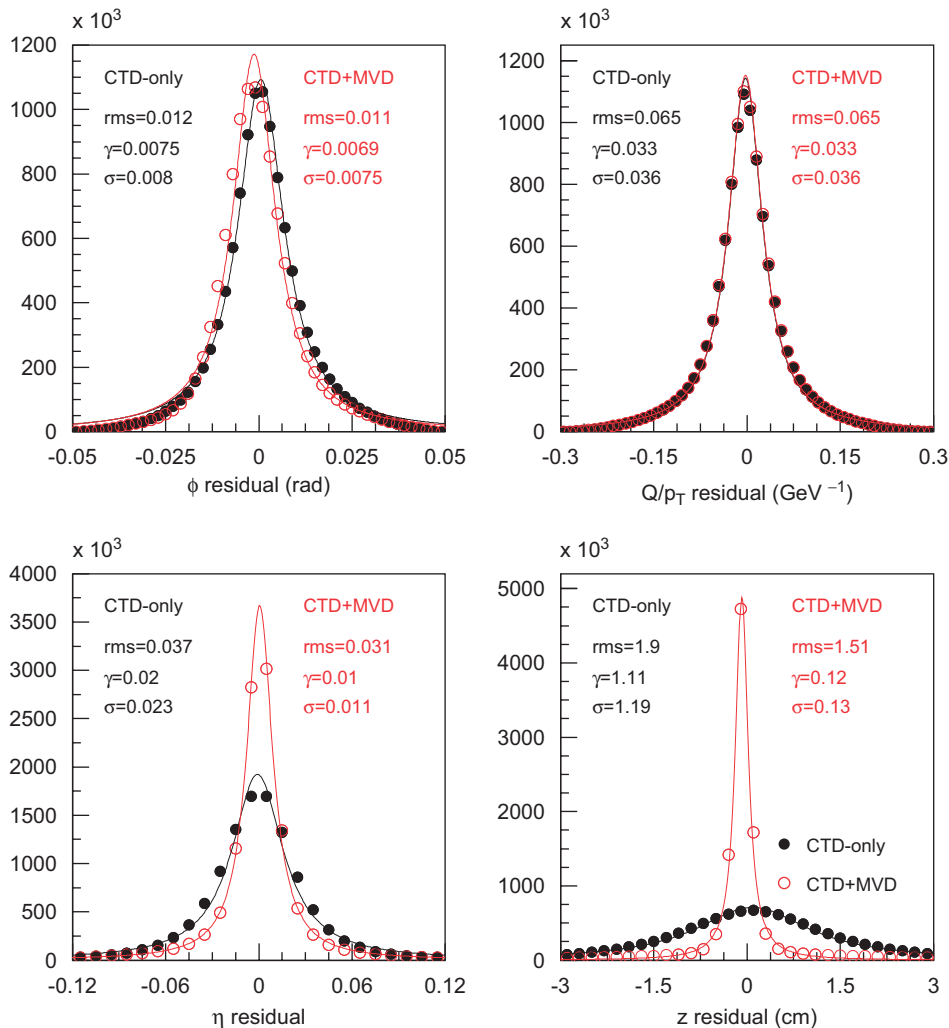


Fig. 11. Residuals between full-length GTT tracks and matched offline tracks in ϕ , Q/p_T , η , and z_{track} . The Q/p_T residuals are corrected for the bias of the GTT. The solid points are for CTD-only GTT tracks and the hollow points are for the CTD + MVD GTT tracks. Also shown are fits of a Cauchy distribution and the related width, γ , together with the width, σ , of a Gaussian fit to the central peak of each distribution.

Since the distributions show significant non-Gaussian tails, a fit to a Cauchy distribution:

$$f(x) = \frac{A_0 \gamma^2}{\gamma^2 + (x - \mu)^2} \quad (5)$$

has been applied, and the half-width at half-maximum, γ , quoted on the figure. Also shown in the figure are the values of the widths, σ , resulting from a Gaussian fit to the central region of each peak. It should be noted that 50% of the area under a Cauchy distribution lies within $\pm\gamma$ of the median value, μ , and 68%—approximately the area within $\pm\sigma$ of the mean of a Gaussian—lies within $\pm 1.84\gamma$. In addition, in the vicinity of the maximum, a Cauchy distribution with a half-width of γ and a Gaussian with $\sigma = \gamma/\sqrt{2}$ coincide exactly. Moving further from the maximum, the two distributions begin to differ such that the width of the Gaussian that best approximates to the Cauchy distribution gradually increases from this value. Where the Gaussian is required to approximate to the Cauchy distribution for distances within around 2γ of the maximum the width of the best approximating Gaussian is around 1.5γ .

Also shown in the figure are the root-mean-square values for the bias-corrected distributions. The Cauchy fits model the central peaks and tails reasonably well, although the

peaks are slightly narrower than the Cauchy fits, and give rise to the slightly larger than expected Gaussian widths. The tails of the Cauchy distribution fit—particularly for the CTD + MVD track z -vertex residual—differ from the data, although not as significantly as those from the Gaussian fit. In the case of the z -vertex distribution, this is because the distribution is actually a supposition of many distributions from tracks with differing numbers of MVD hits each with a different resolution; hence, the tails are underestimated by the Cauchy fit. For this reason, the root-mean-square has been taken to quantify the resolution of the individual track quantities for the remainder of the track quality discussion. It should be noted that although the root-mean-square of a Cauchy distribution or one with larger tails than a Cauchy distribution is not mathematically defined, the matching criterion applied for the tracks effectively limits the maximum allowed residuals in a correlated way; hence, the divergence for large residuals is prevented.

The upper two plots of Fig. 12 show the dependence of the ϕ and Q/p_T resolutions versus the offline z for the same track for full-length GTT tracks. The ϕ resolution drops from 16 mrad for tracks with p_T between 200 and 400 MeV (the first bin) to 6 mrad for high- p_T tracks. Similarly, the resolution of Q/p_T drops from 0.085 GeV^{-1} at low p_T to

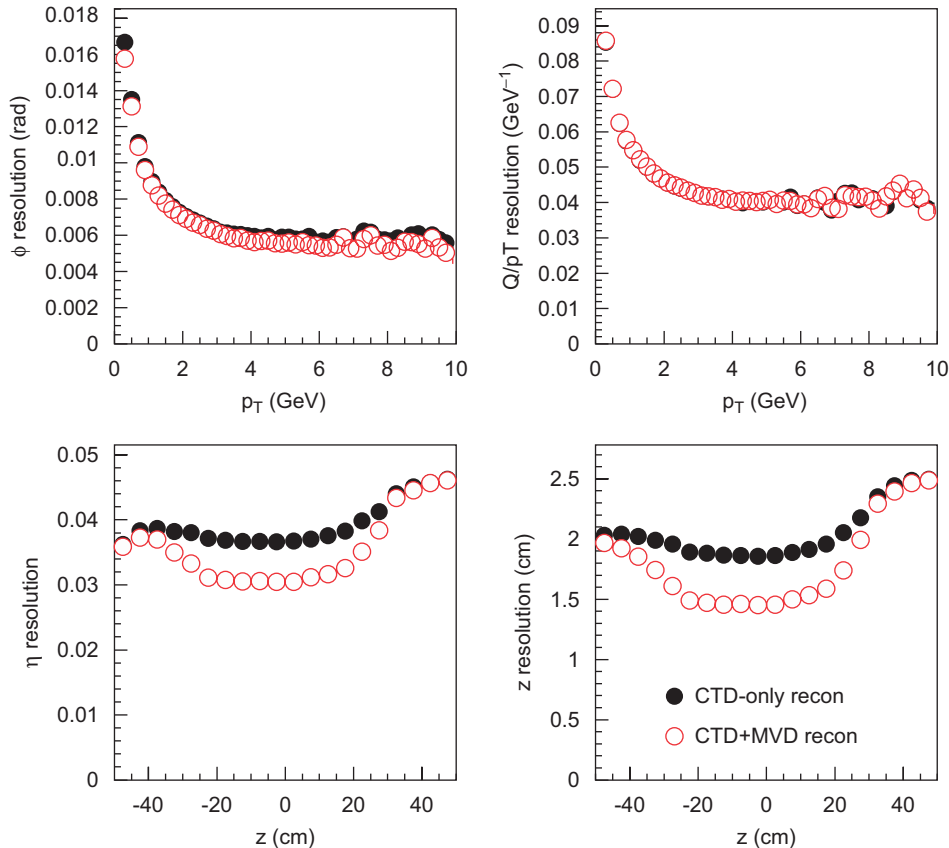


Fig. 12. The RMS of the residuals between full-length GTT tracks and matched offline tracks for ϕ vs. p_T , Q/p_T vs. p_T , η vs. z and z vs. z . The Q/p_T residuals are corrected for the bias of the GTT. The black symbols are for CTD-only GTT tracks and the lighter symbols are for the CTD + MVD GTT tracks.

0.04 GeV^{-1} at high p_T . This should be compared with offline resolutions of Q/p_T which are below 0.01 GeV^{-1} for $p_T > 1 \text{ GeV}$. The worsening of the Q/p_T and ϕ resolutions at low p_T is due to the influence of multiple scattering. Little gain is seen in adding the MVD information to the tracks as the circle fit in x and y is constrained to the nominal beam position.

The lower two plots of Fig. 12 show the dependence of the η and z resolutions on the offline z for full-length GTT tracks. The η resolution for tracks with $z \sim 0$ is about 0.038 for CTD-only tracks and 0.030 for CTD + MVD tracks. The z resolution for central tracks is 2.1 cm for CTD-only tracks and 1.6 cm for CTD + MVD tracks. Clear improvements in resolution due to the inclusion of MVD information can be seen. However, the CTD + MVD tracks include tracks for which no matching MVD hits were found. To see the effect of the MVD hits on the resolutions, Fig. 13 shows the resolutions for ϕ , Q/p_T , η , and z vs. the number of MVD hits matched to the track. For ϕ and Q/p_T , shown in the upper two plots, the resolution for tracks with five or more matched MVD r - ϕ and z hits is about a factor of two better than for tracks with zero or one matching hit. For η and z , the resolution for tracks with five or more MVD hits is about five times better than for tracks with zero or one hit. The fraction of

full-length tracks, those reaching superlayer 9, with 0 or 1 MVD hits are approximately 10% and 10%, respectively. For the shortest tracks, reaching superlayer 3 only, these fractions increase to approximately 25% and 25%. Averaging over all tracks the fractions are approximately 14% and 11%.

6.1.3. Track finding efficiency and purity

The efficiency for offline vertex tracks to have a matching GTT track is shown in Fig. 14 as a function of the offline p_T , η and the number of offline tracks and, for low p_T it shows the efficiency separately for both positive and negative tracks. The GTT efficiency drops sharply as p_T drops below 0.4 GeV . This is expected since the GTT segment finding is biased towards higher p_T tracks. For $p_T > 0.5 \text{ GeV}$, the matching efficiency is about 68%. The ϕ asymmetry of the CTD geometry leads to an asymmetry between positive and negative tracks in the track finding efficiency at low p_T which is clearly seen in the figure. The efficiency shows a dip at central pseudo-rapidity (lower-left plot), which is due to the reduced pulse heights for tracks which traverse the CTD perpendicular to the wires. The lower right plot shows that the matching efficiency of the offline tracks with the GTT tracks drops steeply with the track multiplicity. This is partly due to the nature of the

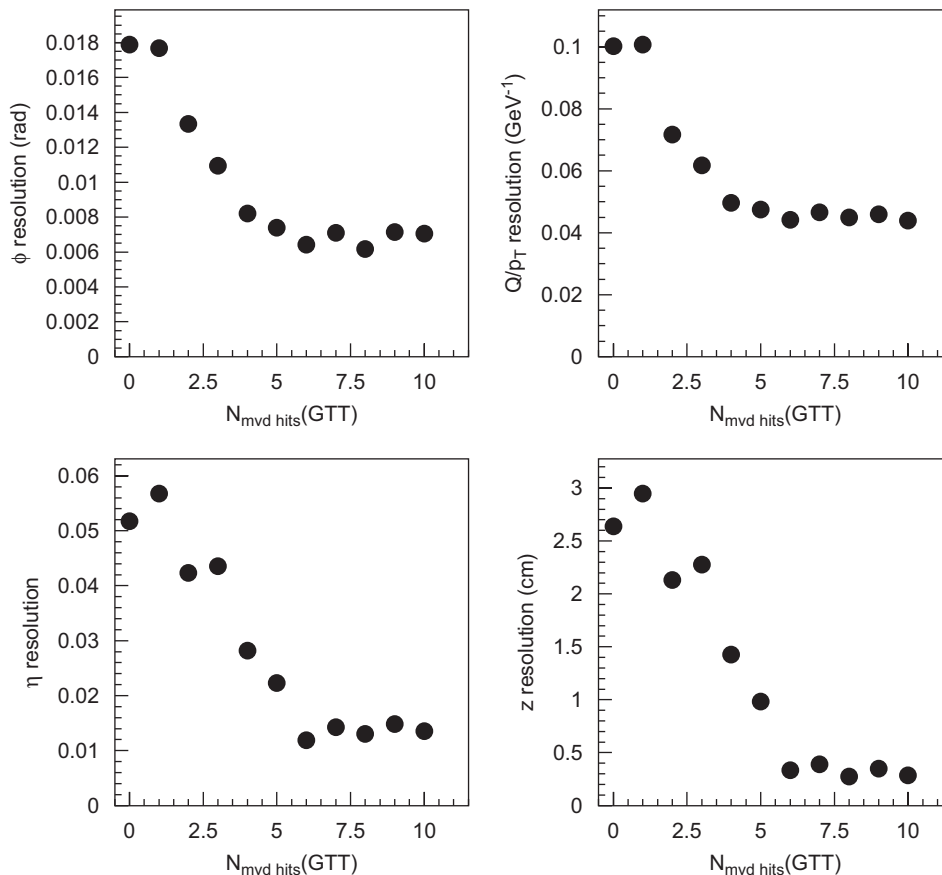


Fig. 13. The RMS of the residuals between full-length CTD + MVD GTT tracks and matched offline tracks in ϕ , Q/p_T , η and z is plotted vs. the number of MVD hits matched to the track.

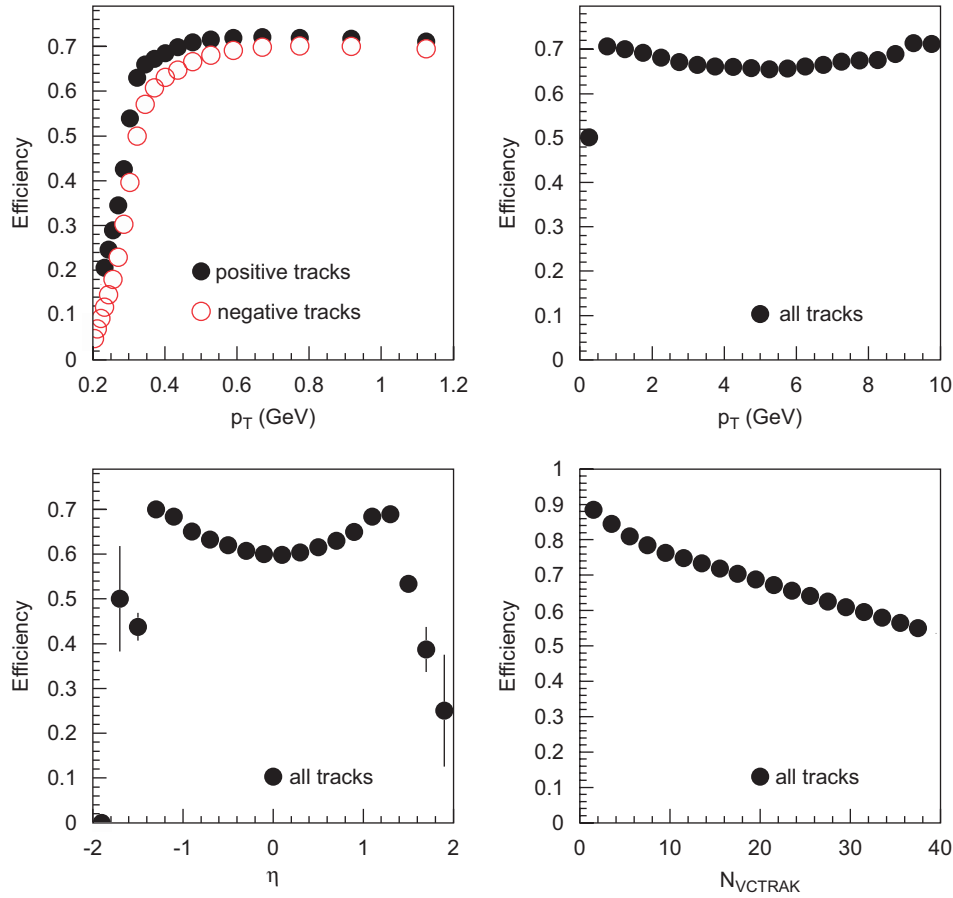


Fig. 14. Efficiency for an offline full-length vertex track with $p_T > 200$ MeV to have a matching GTT track vs. the offline track variables p_T , η , and the total number of offline tracks. In the first plot, the positive and negative tracks are shown separately to illustrate the charge asymmetry at low transverse momentum arising from the CTD geometry.

matching algorithm which breaks down when the separations between tracks becomes smaller than their resolutions and partly due to the confusion of the GTT track finding in a crowded environment. It is noteworthy that in low multiplicity events, the track matching efficiency approaches 90%.

In order to access the quality of the GTT tracks, we define the purity as the probability that a given GTT track has a matching offline track. Fig. 15 shows the purity vs. the number of MVD hits on the track for tracks ending in superlayers 3, 5, 7, or 9. In the lower right plot it is seen that full-length tracks with no matched hits have a purity of only 5%. The purity increases rapidly with the number of MVD hits and exceeds 90% for tracks with five or more MVD hits. GTT tracks that have none or a low number of associated MVD hits are predominantly badly reconstructed tracks resulting from deficits in the algorithm. Examples of these tracks can be seen in the event display of a dijet event shown in Fig. 16. The likelihood that a badly reconstructed track will pass the criterion for matching an offline track is reduced.

Thus, a key advantage of the CTD + MVD tracking is the possibility to reject badly reconstructed tracks by requiring a minimum number of MVD hits.

6.2. Primary vertex resolution and efficiency

The primary vertex finding resolutions and efficiencies have been evaluated using a large sample of events selected from events written offline. An unbiased sample, with no trigger vertex requirement, was selected by requiring the events to have been accepted by the SLT passthrough trigger. Additionally, the events were required to have a good vertex reconstructed offline and to pass the selection criterion for Deep Inelastic Scattering (DIS) events.

The results of the vertex algorithm are reported both after the CTD-only stage of track finding (including a second pass of the stereo pattern recognition to improve the track finding efficiency) and after MVD hits have been assigned to the tracks and the tracks refitted.

Fig. 17 shows the residual between the GTT vertex and the offline primary z -vertex after CTD-only and CTD + MVD tracking. A fit of a Cauchy distribution (Eq. (5)) is also shown for each distribution. Since there is at most one primary vertex identified per event, no special matching criterion has been applied. The half-widths of the Cauchy distributions were found to be 1.40 cm after CTD-only reconstruction and 0.27 cm after the CTD + MVD reconstruction.

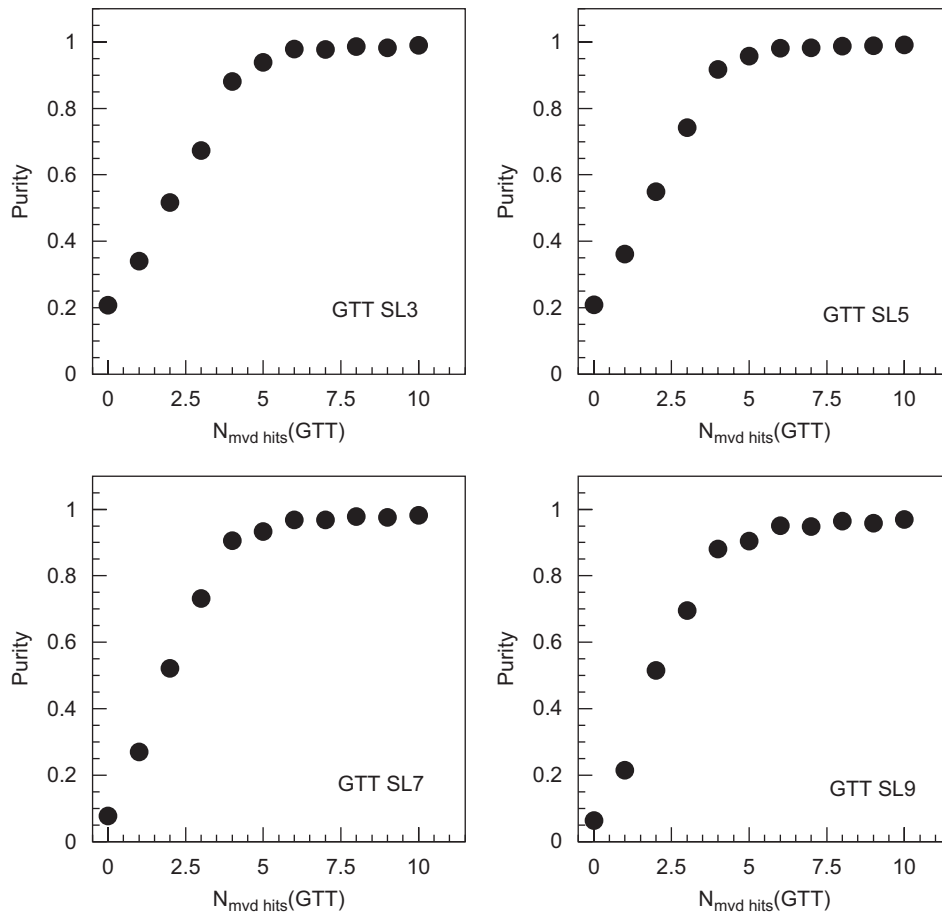


Fig. 15. Purity vs. the number of matched MVD hits for CTD + MVD GTT tracks ending in superlayer (SL) 3, 5, 7, or 9.

The vertex resolution as a function of offline z vertex position and number of offline vertex tracks is shown in Fig. 18. Both distributions show the significant improvement gained by adding MVD hits. The resolution, including the MVD data, as a function of z is ~ 5 times better than that for the CTD-only processing and is constant over the geometrical acceptance of the MVD barrel, $|z| \leq 30$ cm. The vertex resolution is flat and largely independent of the number of vertex tracks if MVD hits are present, whereas the CTD-only resolution improves slowly with the number of tracks found. These effects are driven by the superior spatial resolution of MVD hits.

The vertex finding efficiency after the CTD-only and CTD + MVD stages of processing is shown in Fig. 19 as a function of the number of offline vertex tracks. The curves correspond to the efficiency of finding a vertex within the specified $|\Delta z|$ of the offline vertex. The effect of the improving resolution with the number of tracks is clearly seen by the increase in the vertex finding efficiency. The benefit of the superior resolution of the MVD processing stage can be seen in the efficiency for finding the vertex. For $|\Delta z| \leq 2$ cm, the CTD-only and CTD + MVD vertex efficiencies are $\sim 50\%$ and $\sim 80\%$, respectively, for events with 10 on vertex tracks.

Fig. 20 shows the efficiency for reconstructing the vertex position within $|z| \leq 60$ cm—the value used typically by the physics trigger filters. For vertices reconstructed using MVD information, the efficiency rises from 67% for events with two tracks to better than 97% for events with more than 10 tracks.

6.3. Comparison with CTD-SLT and offline tracking

A comparison of the GTT CTD-only and CTD + MVD track finding resolutions with CTD-SLT and offline tracking has been made with a sample of 6.5×10^4 Monte Carlo dijet photoproduction events.

In the sample of simulated events, GTT tracks were compared with stable charged particles which were emitted from either the ep collision or from the decay, inside the CTD inner radius, of a secondary particle. The creation point of the particle was required to have $|z| < 50$ cm and only particles with $p_T > 150$ MeV and $|\eta| < 1.5$ were considered. The algorithm described in Section 6.1.1 was used to match GTT tracks with either true particles or offline reconstructed tracks.

The resolutions, taken as the σ of a Gaussian fitting the central peak of the distributions measured from Monte Carlo data, are shown in Table 5.

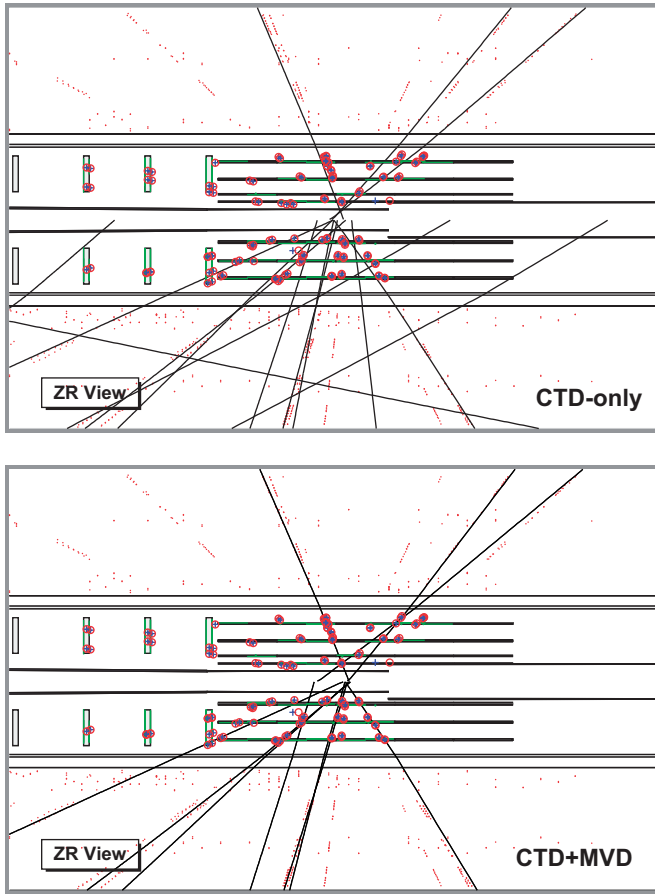


Fig. 16. Event display showing a dijet event candidate as seen with the CTD-only and CTD + MVD tracks with at least 1 MVD hit reconstructed by the GTT algorithm. Note the improved vertex and removal of badly reconstructed tracks when MVD information is used.

The comparison of GTT CTD-only with CTD-SLT track resolutions shows that the p_T and ϕ resolutions improve significantly due to calculating ϕ from the fit parameters rather than the outer segment, thus using all the available segment data in the calculation. The reason only the outer segment was used previously is due to the track ϕ information from the CTD-SLT being needed predominantly for matching the track with a calorimeter cluster; hence, the exit point of the track from the CTD using the single outer segment was sufficient.

Additionally, the η and z_{track} resolutions improve by almost an order of magnitude due to the use of the stereo data combined with the z -by-timing data. This is consistent with the expectations from hit resolutions of $\sigma_{\text{hit}}(z\text{-by-timing}) \sim 6.5$ cm and $\sigma_{\text{hit}}(z\text{-by-stereo}) \sim 2$ mm.

The comparison of CTD + MVD with CTD-only track resolutions shows only small improvements in the p_T and ϕ parameters when MVD information is added to the tracks as the circle fit in x and y is constrained to the nominal beam position. The η and z_{track} resolutions improve significantly due to the improved z hits resolution of the MVD compared with the CTD.

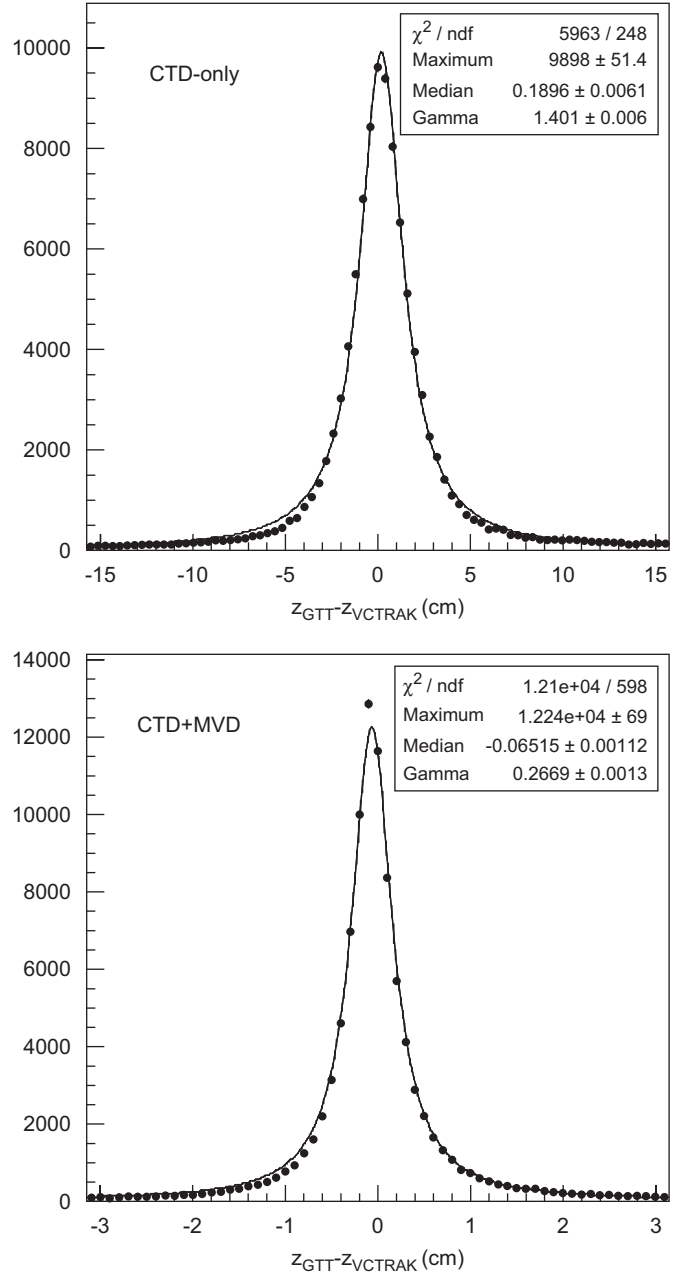
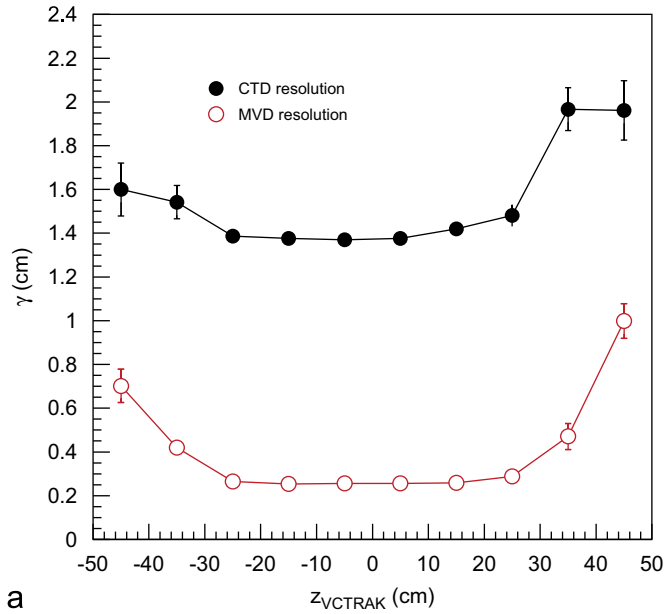


Fig. 17. The residual $z_{\text{GTT}} - z_{\text{VCTRACK}}$ for the GTT vertex algorithm after the CTD-only stage of processing, and the MVD hit matching stage. Also shown are fits using a Cauchy distribution to model the non-Gaussian tails.

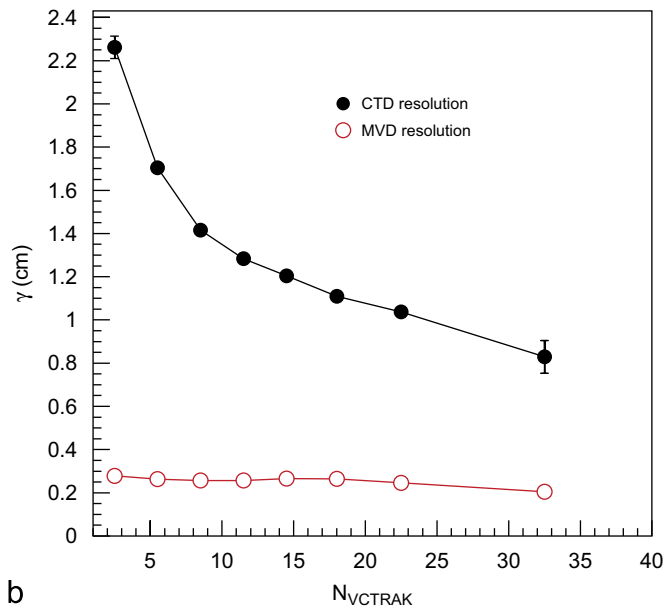
The offline resolutions are, as expected, significantly better than the CTD + MVD values.

7. Trigger quantities

The GSLT generates a trigger accept or reject decision by evaluating multiple expressions grouped into slots; currently ~ 80 slots are defined. Each slot is designed to select interesting physics event patterns derived from track finding, energy and particle identification provided by detector components. Expressions are essentially cuts on



a



b

Fig. 18. The fitted half-width at half-maximum for the vertex residual distributions as a function of (a) the offline z position of the vertex and (b) the number of identified offline vertex tracks. The solid and empty points are for CTD-only and CTD + MVD tracking, respectively.

trigger quantities sent to the GSLT. The final trigger result is evaluated by a logical OR of the slot results. A complete discussion of how the GSLT generates a trigger decision, including vetoing, pre-scaling, and passing minimum bias events, is described elsewhere [6].

As the data transfer and computing capabilities of the GSLT are moderate, component trigger data should be immediately usable in the expressions. GTT quantities are sent in two tables. One contains the source event data sizes in bytes received by the data interfaces. The other

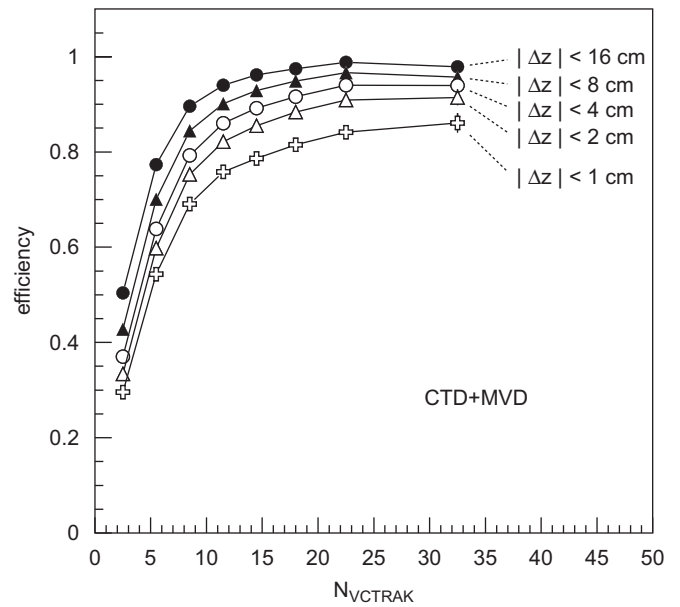
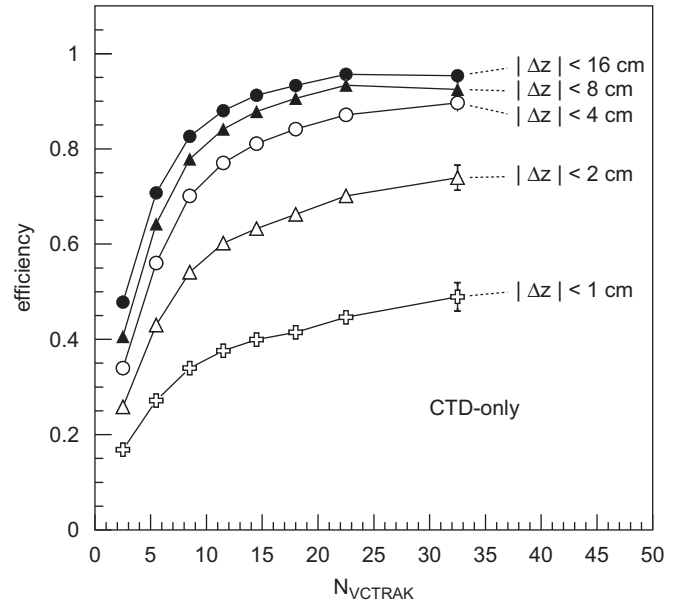


Fig. 19. The vertex finding efficiency vs. the number of offline tracks for the GTT vertex found after matching CTD-only and CTD + MVD information.

contains two rows of calculated quantities, row one values are derived from CTD-only tracking results and row two from CTD + MVD tracking, which allows GSLT filters to use CTD-only or the more accurate CTD + MVD results. The trigger quantity results fall into the following categories:

- primary vertex position and related quality indicators;
- track multiplicities both on and off the primary vertex track counters;
- track p_T related quantities for the highest two p_T and the scalar p_T sum of all vertex tracks;
- candidate J/ψ and Υ resonance meson masses;

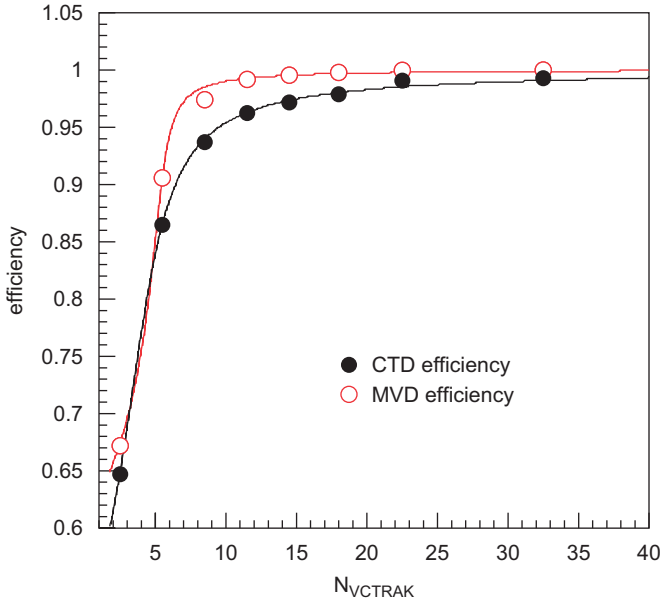


Fig. 20. The efficiency for reconstructing an event vertex within $|z_{\text{GTT}}| < 60$ cm vs. the number of offline tracks.

Table 5

The ϕ , η , z_{track} and fractional p_T resolutions for CTD-SLT, GTT, and offline tracks

| | $\sigma_{p_T}/p_T (\times 10^{-2})$ | $\sigma_\phi (\times 10^{-2} \text{ rad})$ | $\sigma_\eta (\times 10^{-2})$ | $\sigma_{z_{\text{track}}} (\text{cm})$ |
|-----------------|-------------------------------------|--|--------------------------------|---|
| CTD-SLT | 6.64 | 3.90 | 18.30 | 12.40 |
| GTT (CTD-only) | 2.77 | 1.22 | 3.98 | 1.96 |
| GTT (CTD + MVD) | 2.76 | 1.05 | 1.69 | 0.21 |
| Offline | 1.00 | 0.31 | 0.26 | 0.01 |

Only the offline algorithm is not constrained to the vertex in r - ϕ and none of the algorithms are constrained in z . The CTD-SLT values are calculated for the pre-upgrade, HERA-I data period when the MVD was not installed.

- event background likelihood; and
- control flags, including the timeout bit, defining how the algorithm ran.

7.1. Background probability

At ZEUS there is a large background rate from proton interactions with residual gas and from off-orbit proton interactions with material upstream of the detector. While ep events have vertex positions with a Gaussian distribution centred on $z = 0$ and a width of ~ 12 cm, background events are characterized by having: large track multiplicities, relatively low vertex track multiplicity and a vertex z distribution which slowly increases towards negative values.

The GTT reduces this background using an algorithm tuned by comparing a sample of offline selected dijet (signal) events with an unbiased sample of GFLT accept (GSLT passthrough) events. Offline suppression of beam-

gas background is very effective due to the precise 3D vertex reconstruction and cuts which are applied on the shape of the energy deposition in the calorimeter. Both the signal and the FLT data sets were binned in two GTT variables: $N_{\text{vtx-seg}}$, the total number of stereo segments for all tracks which were assigned to the vertex, and $N_{\text{non-vtx}}$, the number of non-vertex tracks. A simulated annealing [25] procedure was used which, for a given input signal efficiency ε , determined upper and lower cuts, $z_{\text{cut-high}}$ and $z_{\text{cut-low}}$, on z_{vtx} , for each bin in $(N_{\text{vtx-seg}}, N_{\text{non-vtx}})$.

When evaluating the background probability of an event with $(N_{\text{vtx-seg}}, N_{\text{non-vtx}})$, the value of $1 - \varepsilon$ for which z_{vtx} is equal to either $z_{\text{cut-low}}(1 - \varepsilon)$ or $z_{\text{cut-high}}(1 - \varepsilon)$ is calculated. For signal events, the returned value of $p = 1 - \varepsilon$ is an approximately uniform distribution, while for proton-induced background, the returned values were peaked near zero.

In the trigger for Charged-Current DIS, which suffers from a particularly high rate of proton-induced background, a cut of $p < 0.015$ was applied to events for which the calorimeter energy deposition was compatible with proton-induced background. This resulted in a 17% decrease in the trigger rate with essentially no loss of efficiency.

7.2. Data size cuts

Event data sizes arriving at each data interface are sent to the GSLT as part of the GTT result and many of the GSLT's physics filters use a CTD data size cut to reject events, typically beam-gas, where a large fraction of wires are hit. The heavy flavour filters reject events when the data size exceeds 90% of the maximum, which removes $\sim 13\%$ of events with a negligible loss of physics events. Physics event losses are calculated using passthrough event samples. Removing large events improves the ZEUS DAQ performance by reducing the bandwidth load.

7.3. J/ψ mass

The superior reconstruction of the GTT makes it possible to reconstruct particle invariant masses at the SLT for the first time at ZEUS. This has been exploited to produce a trigger quantity suitable for selecting vector mesons, such as the J/ψ and Υ , and any two body decays of other neutral particles.

The algorithm for producing this quantity, referred to as the J/ψ mass, is as follows:

- Reject events containing ≥ 100 tracks.
- Loop over all opposite charged pairs of tracks.
- Reject pairs where the tracks are not consistent with coming from the same z .
- Calculate the invariant mass of all pairs assuming they are muons.
- Select the highest invariant mass.

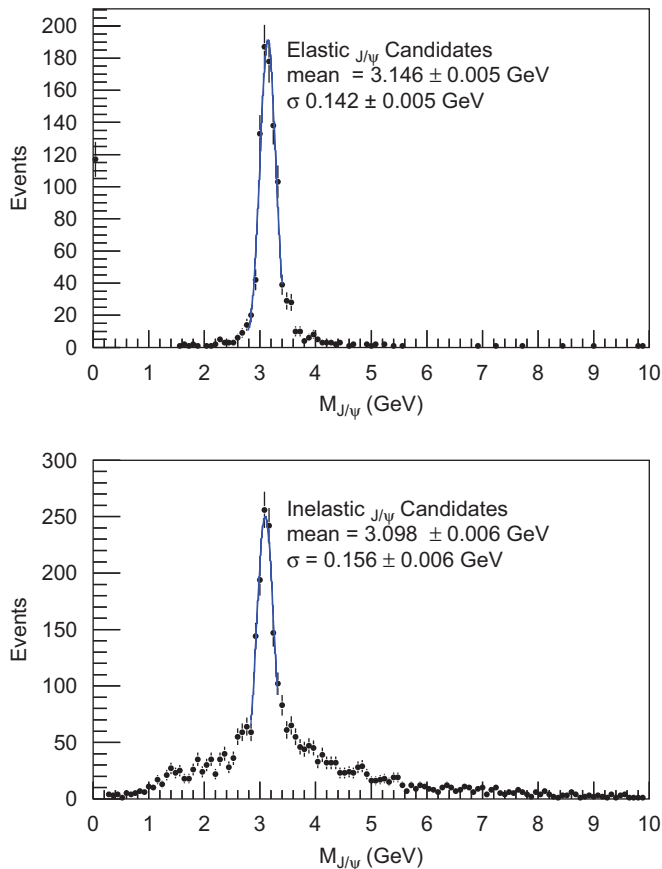


Fig. 21. The online $M_{J/\psi}$ distributions for elastic and inelastic J/ψ production.

The GTT mass distribution for elastic and inelastic $J/\psi \rightarrow \mu^+\mu^-$ production events selected offline is shown in Fig. 21. The mass resolution is found to be 140–150 MeV, compared with the offline resolution of 60–80 MeV. Selecting elastic (inelastic) events from the data, collected with an independent (inclusive muon) trigger and with an online mass > 2.7 GeV, gives an efficiency of 86% (77%) for J/ψ s decaying within the CTD acceptance. The contribution of this calculation to the overall barrel algorithm latency is negligible.

7.4. Highest p_T tracks

The GTT does not send complete information concerning tracks found to the GSLT, instead only the transverse momenta of the two highest p_T tracks are sent. Experience gained from the CTD-SLT, where many track parameters including the p_T were sent for the first 20 tracks found, showed that only the knowledge of the two tracks with the highest p_T values was useful.

The p_T s are used in the heavy flavour and exotics triggers to improve the selection of physics events coming from the decays of heavy intermediate particles like D^* . The efficiency of the heavy flavour trigger for selecting low multiplicity D^* channels has been improved by $\sim 8\%$ whilst reducing the rate by $\sim 35\%$.

7.5. Summary of trigger usage

The GTT has made an important contribution in all the the Physics filters based mainly on tracking quantities such as heavy flavour filters, but also for QCD-jet, inclusive deep inelastic and some exotics filters where the improved quality of the vertexing information and the presence of a fine-grained beam-gas background calculation have improved the selection capabilities and the rejection of proton-rest-gas events.

In the case of heavy flavour, jet physics, inclusive deep inelastic and exotics the capability to send the value of the CTD data size to the GSLT has reduced further the fraction of proton-gas events that survive GSLT selection.

For heavy flavour physics the availability of improved tracking and vertexing information from the GTT has increased the selection efficiencies and reduced the rates of the existing filters. Moreover, it has added the possibility to directly select at the second level specific meson and specific decay channels further increasing the efficiency for this kind of event selection. In the case of single slots, the improved tracking capabilities have increased up to three-fold the efficiency in selecting events with two muon candidates.

8. DAQ performance

The rate dependence of the GTT decision latency at the GSLT is the principal measure of GTT compliance with the experiment's trigger/DAQ system. The measurements of the DAQ performance presented have been made during physics runs taken during the last weeks of 2005 when the CTD-SLT was still running and CTD + MVD tracking was enabled in the GTT algorithm.

The information used consists of timing and counter data written offline with accepted and passthrough events. Absolute timing information is obtained using an RS 422 4 μ s clock bused to all interface VME CPUs and relative timing is derived from the CPU clock at each GTT PC. The measured DAQ performance seen during physics runs during 2005 is summarized in Table 6. These can be compared with the 1999 design requirements summarized in Table 1. The maximum input GFLT rate is lower due to the GFLT accept abort of the FCLR.

Table 6
Measured GTT rates, latency and data sizes seen in 2005 physics runs

| Condition | Measured |
|-----------------------------------|---------------------|
| GFLT accept rate | ≤ 400 Hz |
| GSLT accept rate | ≤ 70 Hz |
| GTT mean latency at GSLT | ~ 10 ms |
| GTT mean latency at EVB | ~ 50 ms |
| MVD mean cluster data size | ~ 2.5 kB/event |
| CTD mean axial + stereo data size | ~ 3.6 kB/event |
| CTD mean z -by-time data size | ~ 1.0 kB/event |
| STT mean data size | Not measured |

The GSLT continuously measures component trigger latency during data taking and the GTT and CTD-SLT latencies are shown in Fig. 22. The mean latency of the GTT is significantly lower than the CTD-SLT, 9.4 ms compared with 13.2 ms. The delay of event data being available at the CTDf and MVD0 interfaces is shown in

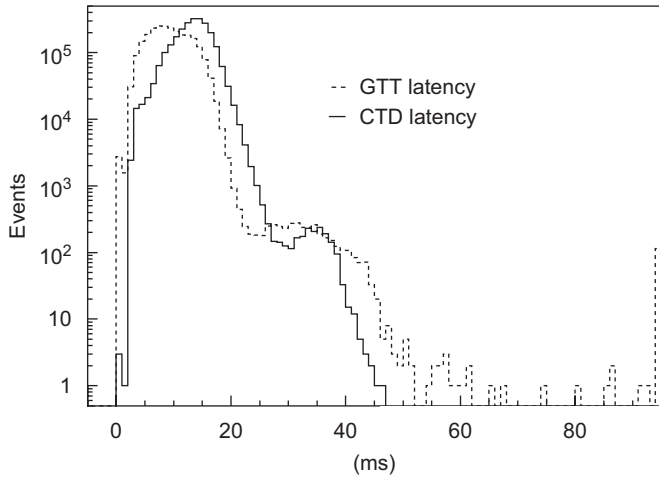


Fig. 22. Distributions of the CTD-SLT and GTT result latency at the GSLT.

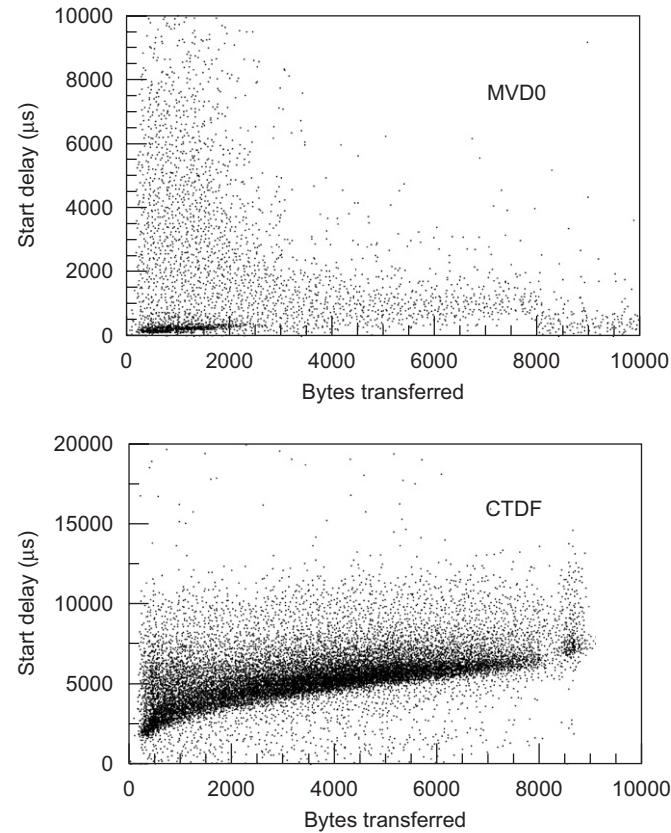


Fig. 23. The delay of event data being available at the MVD0 and CTDf data interfaces. This corresponds to the delay in starting to send event data to the GTT processing node.

Fig. 23. MVD data are quickly available with a weak linear dependence on data size, whereas CTDf (and CTDZ not shown) have an offset of 2 ms and a pronounced size dependence. These effects are due to differences between embedded and parasitic interfaces, CTDf data are delayed by passage through the transputer network. Both MVD0 and CTDf interfaces show tails to larger delays for all data sizes. The effect is larger for the MVD and results from concurrent activity at the interface such as transferring strip data for the last accepted event to the EVB.

The event data size distributions for MVD0 and CTDf are shown in Fig. 24. The mean data sizes are consistent with the expected design data sizes in Table 1.

The times required for decoding event data, algorithm processing, and waiting for data arrival are shown in Fig. 25. The mean times are 0.3, 1.4 and 7.7 ms, respectively. The GTT decision latency, both the mean and the tails, is dominated by the delay of arrival of event data at the processing node. The arrival time of data from a data interface is a convolution of the time when the complete data for an event is available at the interface and how much time is required to transfer the data over the network. Measurements of the achieved network throughput show that the full FastEthernet speed is achieved.

The maximum throughput rate of the GTT has been measured using the GTT playback system. This uses the ZEUS DAQ run control to feed data sets of dumped GSLT passthrough, Monte Carlo or other event samples through

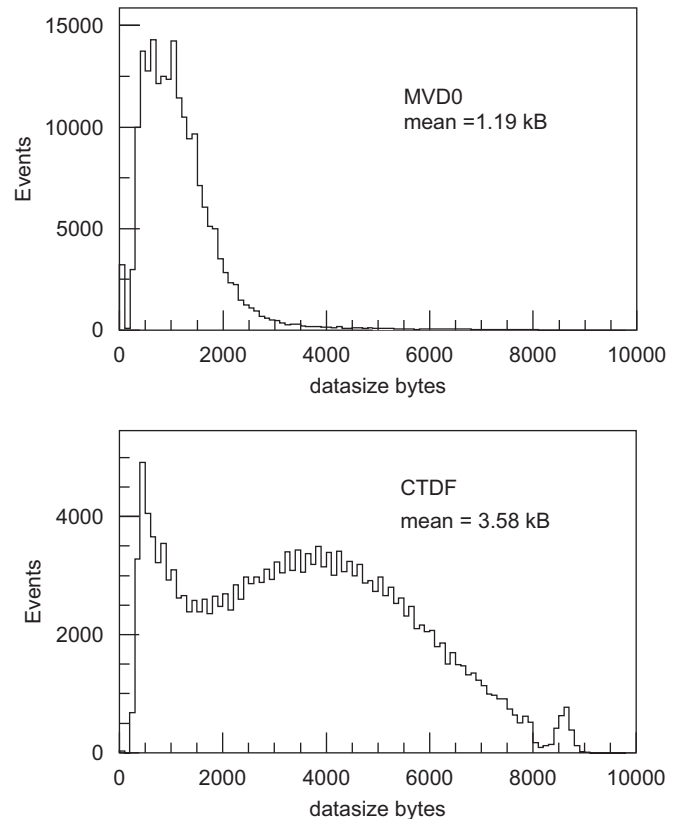


Fig. 24. MVD0 and CTDf event data sizes.

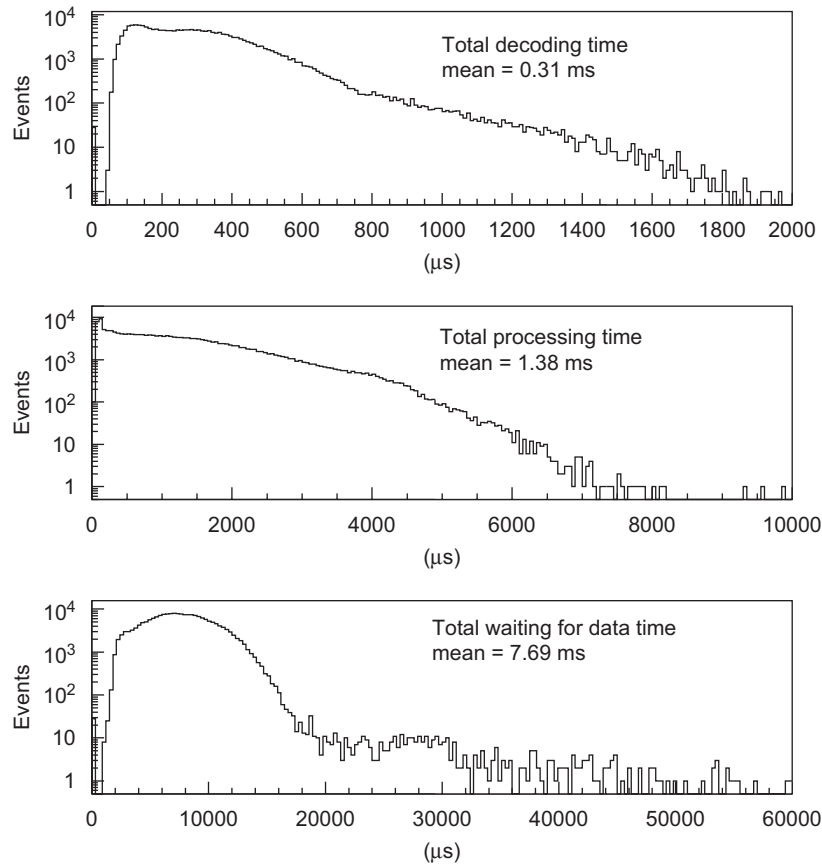


Fig. 25. Distributions showing the time required for decoding event data, algorithm processing, and waiting for data arrival.

the GTT during a ZEUS rate test run. Events stored at run configuration into data interface CPU memory are swapped into the readout chain on GFLT accept during the run. This allows DAQ performance as a function of GFLT and GSLT accept rates with arbitrary detector data sizes to be measured. GSLT passthrough events have been used to measure the performance of the GTT system as a function of input rate for runs where only the MVD and CTD detectors are included and where the GSLT accepts 1 in 6 events. Below a GFLT accept rate of 800 Hz the deadtime is $\leq 3\%$. The deadtime increases rapidly above 880 Hz reaching 30% for 830 Hz. As the number of free credits available does not change significantly between 800 and 830 Hz it is likely that the bottleneck is not the GTT.

9. Simulation

To measure interaction cross-sections reconstructed data from the detector has to be unfolded to take account of the finite resolution of the detector and the trigger efficiency. To calculate the transfer function, Monte Carlo data including a simulation of the detector response and trigger is used.

The ZEUS trigger is simulated by the ZGANA software package [26]. A significant advantage of the GTT is that the algorithm code can be used directly within ZGANA

without modification as the GTT runs on Linux PCs as does offline processing, and not a transputer network. Algorithm software is interfaced to ZGANA using wrapper functions which also simulate aspects of the online electronics to reconstruct the input cluster data sent to the algorithm environment from the raw data banks stored offline. The results of the simulation are the trigger table results sent to the GSLT, which allows the GSLT to be simulated.

10. Testing

The algorithm can only be used online when: its crash resilience has been proven, its results are consistent with expectations and its latency is acceptable.

Crash resilience is tested by feeding a large standard data set of GSLT passthrough events through the algorithm. This is performed using a full run control simulation system running on a single GTT node. The use of dumped passthrough events is of particular importance as they contain a sample of all events with which the algorithm will be confronted during normal running. This is not true for events written to tape or Monte Carlo data sets.

Algorithm data and algorithm results dumped during ZEUS and simulation runs can be compared, event-by-event, against results from previous versions of the

algorithm or offline generated results from a new version. This capability allows deviations between different algorithm versions to be quantified. The CVS tag of the algorithm version running is written to the run summary database.

Finally the DAQ performance is certified by comparing the trigger latency measured on the single node simulation for passthrough events with previous algorithm versions.

11. Monitoring

The trigger and DAQ performance of the GTT algorithm and hardware system is monitored at a number of levels.

During data taking online histograms, from individual environments and summed over all environments, can be viewed by the data taking shiftcrew for anomalies. This allows only rudimentary statements to be made about how the GTT is working.

Online algorithm performance is measured soon after a data run has completed by running the simulation on data sent to offline express monitoring queues. The results of the online processing and the offline simulation are compared and histogrammed and the histograms are examined and compared with reference histograms on a daily basis by the CTD data quality monitoring shift. The comparison with simulation ensures that the online algorithm is behaving correctly.

More detailed offline reconstruction is performed later. At this stage high statistics measurements of track finding resolutions and efficiencies and other parameters are performed.

12. Conclusions

The GTT has been actively used in the ZEUS trigger since 2004 when an initial algorithm performing track finding on data from the CTD was used. At the end of 2005 the track finding was further improved by adding MVD information. The tracking, primary vertex and other trigger quantities calculated are significantly improved when compared with those of the standalone CTD-SLT; consequently, this trigger component was disabled at the end of 2005 data taking.

The design of the GTT which uses VME interfaces with data and trigger components and track finding algorithms running on a PC farm has been a resounding success. The

DAQ performance of the GTT has met the design requirements and the trigger latency at the GSLT is significantly better than CTD-SLT.

The trigger potential of the GTT has been fully exploited and the GTT has made an important contribution in all the the physics filters based mainly on tracking quantities (heavy flavour filters, and also for QCD-jet, inclusive deep inelastic and exotics filters) where the improved quality of the vertexing and tracking information and the presence of a fine-grained beam-gas background calculation have improved the selection efficiencies of the filters whilst reducing their rates.

Acknowledgements

We would like to thank the Intel Corporation for their generous support in providing the initial GTT PC-farm and network hardware.

References

- [1] ZEUS Collaboration, Status Report, ZEUS 93.
- [2] U. Schneekloth (Ed.), DESY HERA 98-05, 1998.
- [3] C. Youngman, in: Proceedings of CHEP92 Annecy, CERN, 1992, p. 222.
- [4] R. Carlin, et al., Nucl. Phys. B (Proc. Suppl.) 44 (1995) 430.
- [5] B.G. Bylsma, et al., Nucl. Instr. and Meth. A 337 (1994) 512.
- [6] H.A.J.R. Uijterwaal, Ph.D. Thesis, University of Amsterdam, 1992.
- [7] INMOS Ltd, The Transputer databook, second ed., 1989.
- [8] A. Garfagnini, et al., Nucl. Instr. and Meth. A 435 (1999) 34.
- [9] A. Coldewey, et al., Nucl. Instr. and Meth. A 447 (1999) 44.
- [10] M. Feuerstack, Nucl. Instr. and Meth. A 447 (2000) 89.
- [11] T. Fusayasu, K. Tokushuku, Nucl. Instr. and Meth. A 436 (1999) 281.
- [12] B. Foster, et al., Nucl. Instr. and Meth. A 338 (1994) 254.
- [13] N. Harnew, et al., Nucl. Instr. and Meth. A 279 (1989) 290.
- [14] B. Foster, et al., Nucl. Phys. (Proc. Suppl.) B 32 (1999) 181.
- [15] A. Quadt, et al., Nucl. Instr. and Meth. A 438 (1999) 472.
- [16] S. Fourletov, Nucl. Instr. and Meth. A 535 (1–2) (2004) 472.
- [17] S. Goers, Proceedings of IMTC/04, Como, 18–20 May 2004.
- [18] M. Sutton, et al., ZEUS Note 99-007, 1999, unpublished.
- [19] A. Polini, et al., ZEUS Note 99-034, 1999, unpublished.
- [20] A. Polini, ZEUS Note 99-071, 2003, unpublished.
- [21] UNIVERSE 2 VME-to-PCI bus bridge, Tundra Semiconductor Corporation.
- [22] H. Boterenbrood, 2TP VME Module, Electronics Department, NIKHEF-H, Amsterdam, Ver. 2.0, September 1990.
- [23] C. Youngman, ZEUS Note 06-004, 2006, unpublished.
- [24] J.F. Crawford, Nucl. Instr. and Meth. A 211 (1983) 223.
- [25] S. Kirkpatrick, et al., Science 220 (4598) (1983) 671.
- [26] G.F. Hartner, Y. Iga, ZEUS Note 90-084, 1990, unpublished.