UNIVERSITY OF OXFORD

Department of Physics

PARTICLE AND NUCLEAR PHYSICS

**THE DESIGN AND PERFORMANCE OF THE ZEUS
CENTRAL TRACKING DETECTOR SECOND LEVEL
TRIGGER**

A. Quadt, R.C.E. Devenish, S. Topp-Jørgensen, M. Sutton,
H.A.J.R. Uijterwaal, A.F. Byrne and D.M. Gingrich

Submitted to *Elsevier Preprint*

SCAN-9906052

CERN LIBRARIES, GENEVA

Ref: OUNP-99-07

Address: Department of Physics
Particle & Nuclear Physics
Keble Road
Oxford OX1 3RH
U.K.

The Design and Performance of the ZEUS Central Tracking Detector Second Level Trigger

A. Quadt ^{a,1}, R.C.E. Devenish ^{a,2}, S. Topp-Jørgensen ^a,
M. Sutton ^a, H.A.J.R. Uijterwaal ^{a,3}, A.F. Byrne ^{a,4} and
D.M. Gingrich ^{a,5}

^a*University of Oxford, Department of Physics, Keble Road, Oxford OX1 3RH, UK*

Abstract

This paper describes the Second Level Trigger for the Central Tracking Detector (CTD-SLT) of the ZEUS experiment. The CTD-SLT consists of a network of micro-processors running a track finding algorithm. Operational experience gained during the 1995 to 1997 HERA data taking periods show that the maximum processing rate approaches 800 Hz, with almost 100% track finding efficiency and adequate p_t and vertex resolution of $\sigma_{p_T} \sim 0.05 p_T^2$ (p_T in GeV) and $\sigma_{z-vtx} \sim 9$ cm.

The online performance of a highly parallel system with dynamic loading such as the CTD-SLT is characterised by two quantities: the average throughput (speed) and the processing time per event (latency). Detailed understanding of these quantities for the CTD-SLT from operational experience, simulations and a simple model is presented.

Key words: Track Trigger; Parallel Processing; Trigger Simulation

---

¹ Now at EP Division, CERN, Geneva, Switzerland.

² Corresponding author, r.devenish@physics.ox.ac.uk.

³ Now with RIPE Network Coordination Centre, Amsterdam.

⁴ Now with Oxford Magnet Technology, Eynsham, Oxford.

⁵ Now at the Centre for Subatomic Research, University of Alberta, Edmonton, Alberta, Canada T6G 2N5.

1 Introduction

This paper describes the design and performance of the ZEUS Second Level Trigger for the central Tracking Detector (CTD-SLT). ZEUS is a multipurpose detector, designed to record the final state particles of collisions between 820 GeV protons and 30 GeV electrons⁶ at the ‘Hadron Elektron Ring Anlage’ (HERA) at DESY. Its main elements are an inner tracking system, consisting of a forward, central and rear tracking detector surrounded by a superconducting thin coil providing a solenoidal magnetic field of 1.43 T, a high resolution calorimeter, muon detectors and several special purpose detectors [1].

HERA is designed to run with 220 bunches of electrons and protons, which in turn collide every 96 nsec in the ZEUS detector. The resulting maximum event rate of 10 MHz in conjunction with the data volume of up to 500 kBytes (80 kBytes on average) per event put stringent requirements on the pipelined ZEUS readout and trigger system. ZEUS uses a three level trigger system in order to separate the interesting ep events from the very large background. The latter originates mainly from interactions between the beam particles and the remaining gas in the beam pipe. By selecting events in a layered system, the higher levels have more time in which to run more sophisticated algorithms.

The first level trigger (FLT) essentially performs a search for predefined patterns in the different detector components to give a maximum output rate of 1 kHz. At the second level (SLT) transputers perform a first event reconstruction, which allows a rate reduction to 100 Hz. The third level trigger (TLT) consist of a computer farm. It carries out an offline type event reconstruction and reduces the output rate to ~ 10 Hz.

The CTD is a cylindrical drift chamber. It provides measurements of charged particle momenta and dE/dx in the polar angle range $7.5^\circ < \Theta < 164^\circ$. It also provides track and event vertex information to all three levels of the ZEUS trigger system and for offline analyses. The CTD has a multi-cell superlayer wire geometry. The wires in superlayers 1, 3, 5, 7 and 9 run parallel to the z -axis⁷ (axial layers) whereas the wires in superlayers 2, 4, 6 and 8 are tilted by $\pm 5^\circ$ with respect to the z -axis (stereo layers). The stereo data allow an accurate offline measurement of the z -coordinate of hits on a track. However they cannot be used in the CTD-FLT or CTD-SLT because of insufficient processing time. Instead the CTD triggers use fast z information based on the

---

⁶ From 1994 to 1997 HERA ran with positrons instead of electrons. In this paper ‘electron’ is used as generic term for the scattering lepton.

⁷ The ZEUS coordinate system is a right handed system with the z -axis pointing in the proton beam direction, the x -axis pointing to the centre of HERA and the y -axis pointing upwards. In the following the proton ($+z$) direction is referred to as ‘forward’, the electron ($-z$) direction as ‘rear’.

difference of arrival times of pulses at each end of the wire.

The CTD-SLT is a software trigger running on a transputer network and is a vital part of the readout electronics. Its goals are to find and measure tracks with high transverse momentum (p_T) and from these to form an event vertex. The CTD-SLT trigger algorithm uses a 4-step approach: (1) track-segments in (r, ϕ) ⁸ are found on a cell-by-cell basis; (2) the local track segments are then transformed into three-dimensional vectors using information from the z-by-timing system; (3) the vectors are then combined into tracks and finally (4) all tracks together are used to determine the event vertex. The CTD-SLT output can be used to trigger positively on certain classes of physics events or to reject particular backgrounds using their specific signatures.

This paper is outlined as follows. An overview of the ZEUS trigger system is given in Sec. 2. The CTD and its transputer readout and trigger system are described in Secs 3 and 4. Sec. 5 describes the CTD-SLT algorithm and Sec. 6 its timing performance. The use of CTD-SLT information to select physics events is described briefly in Sec. 7. Improvements in ZEUS tracking triggers for the HERA upgrade are touched upon in Sec. 8 and a short summary is given in Sec. 9.

2 The ZEUS Trigger System

The ZEUS data acquisition (DAQ) and trigger system [2] has been designed to satisfy the following broad principles:

- essentially dead time free ($< 1\%$);
- large data compaction (3 MBytes/event $\Rightarrow$ 100 kBytes/event);
- acceptable data storage rate (80 kBytes/s compared to the total input rate of 3 TBytes/s);
- easy to maintain and flexible in operation.

The solution adopted has the following characteristics:

- A 3 level trigger system is used [3] (Figure 1).
- Extensive use is made of pipelines for trigger and event data buffering.
- Extensive front end CPU power is used for signal processing.
- Parallel processing is applied wherever possible.

---

⁸ Here r is defined as the radial distance from the z -axis, ϕ is the azimuthal angle measured with respect to the x -axis and the polar angle Θ is defined with respect to the proton beam direction.

Fig. 1. Schematic diagram of the ZEUS trigger and data acquisition system.

The First Level Trigger (FLT) [4] is designed to start the trigger processing of a new event at each bunch crossing, i.e. every 96 nsec. It consists of a set of pipelined hardware processors, mostly based on lookup tables, fast digital adders and programmable gate arrays. The Global First Level Trigger (GFLT) and the component FLT systems synchronise the detector to a cyclic HERA bunch crossing number. The bunch crossing rate of 10.4 MHz typically results in an event rate of 10 – 200 kHz, which is predominantly background. In order to reduce this rate to 1 kHz, the maximum input rate that the Second Level Trigger (SLT) can cope with, the GFLT compares the FLT data of the single components with predefined patterns and forms a decision correspondingly.

The components' FLT and the GFLT have 46 bunch crossings, i.e. $4.4 \mu\text{sec}$, to make the FLT decision. In the meantime the data are stored in pipeline buffers. On a GFLT accept, the data are buffered and the trigger data are sent to the SLT. The typical FLT output rate is at present 200 – 500 Hz where a dead time of at most 2% has been observed.

The Global SLT (GSLT) [5] receives trigger results from the GFLT and, more detailed trigger information from the components. It receives the components' SLT data within 4 msec of the interaction and generates a decision within 7 msec of the interaction. Using data from the component SLTs the GSLT runs algorithms to reject background and to accept events of interest. The SLT output event rate is 30 – 75 Hz (design < 100 Hz), most of which originate from *ep* collisions. The SLT is a transputer based parallel path computational device. The communication between the transputers is achieved via 20 Mbit/s data links. The local event buffers on the SLT transputers, in which the data are stored during the SLT processing, are of variable length. After a GSLT accept the component data are sent to the event builder (EVB) and converted into a standard format.

From the EVB the data are passed on to the Third Level Trigger (TLT) [6] which consists of a computer farm of 30 Silicon Graphics Workstations ⁹. As the TLT runs accelerated FORTRAN offline reconstruction and filter code on complete events, physics based selections are very precise. The maximum TLT output rate is ~ 10 Hz.

Finally the events are written to tape via an optical fibre link (FLINK) for full reconstruction and offline analysis. A small fraction of events are passed to a separate computer for data quality monitoring and detector calibration purposes.

Figure 2 shows the background and trigger rates for all three levels as a function of instantaneous luminosity for the years 1992 - 1997. The trigger rates show approximately a linear rise with the luminosity as expected for *ep* events. One also clearly observes the rate reduction following the sequence of trigger levels. Despite the increased instantaneous luminosity over the years the trigger rates at the different levels are almost unchanged, a result of a better understanding of the trigger and of tighter selection cuts.

⁹ The workstation farm was replaced by 14 Challenge S machines at the beginning of 1996 and in 1997 two Origin 200 machines were added.

Fig. 2. Background and trigger rates on all three levels as a function of instantaneous luminosity for the years 1992 to 1997.

3 The Central Tracking Detector

The central tracking detector (CTD) [7] has an inner and outer radius of 18.2 cm and 79.4 cm, respectively. The chamber axis coincides with the beam axis. The CTD is located from $z = -100$ cm to $z = +105$ cm when measured from the nominal interaction point.

The 24192 wires, out of which there are 4608 signal wires, are organised in nine superlayers consisting of 32 to 96 cells, each cell containing 8 sense wires (Fig. 4). The cell boundaries are defined by planes of field wires. The eight sense wires, separated by field-shaping wires held at ground potential, are arranged in a straight line at 45° with respect to a radial line from the chamber axis and are separated by ≈ 8 mm (Fig. 3). The cells are 5 cm wide yielding a maximum drift distance of 2.5 cm.

The chamber is operated with a gas mixture of argon (83%), CO_2 (5%) and

ethane (12%) with a small admixture of ethanol (0.84%) at 3 mbar above atmospheric pressure, resulting in a drift velocity of $\sim 50 \mu\text{m}/\text{nsec}$ and a maximum drift time of 500 ns. These conditions, together with a drift field of 1.2 kV/cm, give a Lorentz angle of 45° ensuring an azimuthal drift direction of the electrons and ions in the cell.

Fig. 3. CTD sector. The field (highlighted dots) and sense wires in the drift cells are shown along with stereo angles in the different superlayers.

The sense wires in the superlayers 1, 3, 5, 7 and 9 run parallel to the beam direction (axial) layers whereas those in superlayers 2, 4, 6 and 8 are tilted by $\pm 5^\circ$ with respect to the z -axis (stereo layers). All sense wires in superlayer 1 and every other sense wire in superlayers 3 and 5 are equipped with the z -by-timing system (704 wires), mainly for fast trigger purposes.

The CTD hit resolution in (r, ϕ) is $\sigma(r, \phi) \sim 230 \mu\text{m}$. The offline full track reconstruction use both axial and stereo layers in a two stage process. After all corrections (multiple scattering, individual wire corrections etc.) offline tracks give an event z -vertex resolution of $\sim 1 \text{ mm}$. The average z hit resolution from the z -by-timing system is 4.5 cm. Long tracks with hits in at least the first three superlayers in the CTD¹⁰ are reconstructed offline with transverse momentum resolution of

$$\frac{\sigma(p_T)}{p_T} = 0.0058 p_T \oplus 0.0065 \oplus \frac{0.0014}{p_T} \quad (p_T \text{ given in GeV}).$$

More details of the CTD offline reconstruction and charged particle track resolutions may be found in [8; 9].

¹⁰ In fact the majority of such tracks reach superlayer nine.

4 The CTD Readout and Data Acquisition System

4.1 Introduction

The CTD Data Acquisition (DAQ) system [10], of which the CTD-SLT is a vital part, has been designed and developed to fulfil the following requirements. The readout of the CTD must be fast and dead time free. It provides the CTD-FLT with hit information and stores new event data in pipeline buffers during FLT processing every 96 nsec. On a GFLT accept the data is read out, sent to the GSLT and buffered in a transputer memory during the SLT processing. On a GSLT accept the data is sent to the Event Builder for further processing. The design rate of the ZEUS SLT is 1 kHz and the component latency (the time between a GFLT-accept signal and the component sending its result to the GSLT) should be less than 16 ms in order not to cause deadtime by overflowing buffers. As the component FLTs are synchronised the inputs to the components SLTs are also synchronised.

Fig. 4. Rear view of the CTD. The 9 radial superlayers with the up to 96 cells, labelled with small numbers, are shown. The grey shading of the cells indicates, whether they are read out by electronics in an even or odd numbered crate corresponding to one of the 16 ϕ sectors of the CTD (bold numbers).

For readout and trigger purposes, the CTD is divided into 16 ϕ -sectors as indicated by the grey and white shading in Fig. 4. Pre-amplifiers are directly

mounted on the rear endplate of the CTD and in case of superlayers 1, 3 and 5 the z -by-timing system requires pre-amplifiers at both chamber ends. From the chamber the signals are transmitted via 42 m long cables to the rucksack, which accommodates all the ZEUS readout and trigger electronics. Here each sector has three stacked crates with electronics, one for the z -by-timing system, one for the post-amplifiers and one for the ' $r\phi$ ' FADC system.

The main characteristics of the CTD electronics are described in the following.

4.2 The $r\phi$ System

All 4608 sense wires of the CTD are read out by the $r\phi$ system [11], in which 104 MHz Flash Analogue to Digital Converters (FADCs) digitise the pulses into 8-bit words every 9.6 nsec. The on-board Digital Signal Processors (DSPs) perform a zero-suppression and extract the useful hit information. The pulse height, proportional to the energy loss (dE/dx) of the charged particle, is also reconstructed and stored as well as the drift time.

One FADC card reads out 2 cells, one in an inner and one in an outer superlayer, each containing 8 sense wires. This guarantees approximately equal load on every card, an essential characteristic of a parallel processing system.

4.3 The z -by-timing System

The z -by-timing system [12] determines the z -position of a hit along the sense wire from the difference in the arrival times of the pulses at either end of the CTD. The time difference Δt of the amplified pulses is converted to a voltage which is sampled every 48 nsec by an 8-bit FADC to produce a 7-bit value for the z -position of the hit (bin 0 at the rear end and bin 127 at the forward end). The single hit resolution achieved under operational ep conditions is 4.5 cm. The z -by-timing system is used in all three trigger levels and in the offline track reconstruction. All sense wires in superlayer 1 and every other wire (odd numbered) in superlayer 3 and 5 are equipped with the z -by-timing electronics.

4.4 The Transputer

The second part of the DAQ system, including the CTD-SLT, is based on digital electronics and consists of a network of 131 (127 in 1995) transputers

[13]. Transputers and their programming language, OCCAM [14], were developed together, one being optimised for the other. OCCAM may be viewed as a formalism describing a collection of processes operating simultaneously and communicating via channels (point to point connections). Some of the great advantages of using OCCAM and transputers in the CTD-SLT and DAQ are:

- Fast processing¹¹
- A memory large enough to be used as an event buffer removing the need for synchronisation within the SLT. Only input and output have to be synchronised.
- The structure is process-oriented and not processor-oriented. That means several processes can run on one CPU performing many different tasks (multi-tasking).
- The identical communication protocol between processes on one transputer and on different transputers make it easy to add transputers and move processes around in the network without any major redesign.
- Fast communication between processes and transputers is provided via channels and links.

Fig. 5. Photograph of a typical TRAM used for the CTD-SLT processing and general CTD-DAQ. The INMOS T800 transputer (element on the right), the TRAM clock (square module next to it in the middle) and 4 of the 8 RAM chips (lower left) can be seen.

The transputers used in the CTD SLT are INMOS T800s (see Fig. 5), clocked at 25 MHz or 30 MHz. Their CPUs consist of a RISC integer and floating point processors with up to 4 kBytes of quick access on-chip memory. Each transputer has 4 bi-directional links for the communication with other transputers. Via these links the code, running on the transputer, is loaded, data is transferred, monitoring information sent and state transition commands communicated at a rate of approximately 1.8 Mbytes/s. The transputer is mounted on a board together with eight 1 MBit random access memory chips (RAMs). These together with a clock chip build a unit called a TRAM (Transputer plus RAM). The CTD-SLT TRAMs sit on readout controller boards (ROCs)

¹¹ Fast is of course a relative term, transputers were respectably fast microprocessors in 1990/91 when the SLT was being designed.

together with the readout for the z-by-timing electronics.

Fig. 6. Schematic diagram of a transputer. The input and output channels of the four links are connected to 'direct memory access controllers' (DMA). They communicate with the book keeping processes (BK), that control data flow from or to the external memory and talk to other processes. In addition to these the main CPU also runs the actual algorithm process.

A schematic diagram of a transputer with its internal structure and processes is shown in Fig. 6. Each channel, input and output, of the 4 transputer links is connected to a direct memory access controller (DMA), a small unit dealing with external data transfer. Each DMA is controlled by a book-keeping process (BK), which runs in addition to the actual CTD-SLT algorithm on the transputer CPU. This procedure ensures that the communication to other transputers via a link does not absorb any CPU power, as essentially only one command needs to be executed on the transputer CPU, from then on the DMA operates autonomously. All BK processes as well as the algorithm process have access to the external memory (RAM). When the algorithm is ready to process a new event, it sends a request to the appropriate BK process and receives a pointer to a memory address along with a length in words, uniquely specifying the new data set. These are also the two pieces of information passed on to another BK process when the algorithm finishes processing an event. The BK process takes the responsibility for transmitting the event data to the next transputer.

4.5 The CTD Transputer Network

The schematic diagram in Fig. 7 shows the CTD DAQ and trigger electronics in one of the 16 racks, dealing with a ϕ sector of the CTD. Transputers are used in the FADC and the z -by-timing/SLT crate. The FADC and the z -by-timing ReadOut Transputers (ROT) read out the digital hit information from the respective FADC cards. The hit data are then sent to the SLT transputers, placed in the bottom most crate, and to merge transputers, which collect the hit data from this and adjacent crates and pass the merged data set on to the next crate. In the end this results in a chain of data, describing all $r\phi$ and all z hits in the event. Depending on the crate number the merged data might be passed to the following or previous crate or to a separate merge transputer in a tree structure.

The SLT task in a ϕ sector is divided into three processes, the segment finding (SEG), the vector hit finding (VHIT) and the track finding (TRK). They are described in detail in section 5. This process splitting has the effect, that several events can be processed on different stages of the algorithm at the same time. One event is processed in 16 crates simultaneously and several events are processed in one crate at the same time, yielding a 2-dimensional parallel processing structure. In the complete DAQ system, shown in Fig. 8, this configuration of 7 transputers is repeated 16 times, once for every ϕ sector. The reason for the FADC merge transputers being arranged in a tree structure is explained later.

Fig. 7. Schematic diagram of a CTD rack, showing the FADC crate, the post-amplifier crate and the z -by-timing/SLT crate. Also shown are the transputers used in the CTD-DAQ and SLT network.

The transputers dealing with the global CTD tasks such as final data merging, high voltage control, communication to the Global Second Level Trigger, the ZEUS event builder and the CTD equipment computer as well as the SLT vertex finding are situated in the 'regional box' (RBOX). The SLT tracks are first merged in groups of four crates, the result is sent to the transputers 'merge 1' and 'merge 2'. The result of the merging there is finally merged on the SLT driver, which also analyses the tracks found, removes duplicate tracks

due to the 2-crate mask in the track finding and performs the event vertex finding. Event parameters such as the vertex and the number of tracks found are sent to the GSLT along with the track parameters.

Following a GFLT accept decision the FADC data are read out and similarly to the z -data sent to the SLT and to the merge transputer, where they are buffered. Only for a GSLT accept decision, which is received on the GSLT decision receiver transputer, are the FADC and z -data merged. The data are then passed via the FAN in/out transputer to the ZEBRA builder, where the data structure is converted into the format of ZEBRA data banks [15]. Finally the complete CTD data set for this event is transmitted to the ZEUS event builder via the EVB interface transputer. Here the task of the CTD DAQ system ends and global ZEUS processes such as the event builder and the Third Level Trigger (TLT) take over.

4.6 The Playback System

In order to be able to test modifications to the SLT and readout OCCAM code and to measure the time spent on the single calculation steps, which can not easily be simulated in the FORTRAN version of the code, a playback system has been developed. This system allows one to feed selected data or Monte Carlo events from the equipment computer into the DAQ system, where they are distributed and stored in the buffers of the readout transputers in each crate. While running the playback system, events are shipped as quickly as possible into the SLT on crate-by-crate basis. Apart from synchronisation of the CTD-SLT input with a GFLT-accept decision the playback system provides an authentic picture of the online processing performance, including the data shipping, running the algorithm and merging the results. The transputers internally provide timing in units of $64\mu\text{sec}$. Setting a start and stop signal at the beginning and the end of a processing step, one can determine the maximum and average time per step on each transputer.

Fig. 8. 1996 setup of the CTD DAQ transputer network. The thick lines indicate the data flow, the thin lines represent links used for communication between transputers.

5 The CTD-SLT Algorithm

5.1 Introduction

The CTD-SLT algorithm is divided into four steps: (i) track-segments in (r, ϕ) are found on a single cell basis; (ii) the track-segments are then transformed into ‘vector hits’ using information from the z -by-timing system; (iii) the vector-hits of two adjacent crates are combined into tracks; (iv) after duplicate track removal the tracks are used to determine the event vertex. Each step is coded in OCCAM running on a separate transputer. A fifth step, that is not strictly part of the algorithm but affects the latency of the system, is the ‘merge’ process that occurs between steps (iii) and (iv) above. The 5 steps are described in subsections 2-6.

The structure and data flow of the CTD-SLT FORTRAN simulation [16] closely reflect those of the transputer network. The only exception is that the inherent parallelism of the transputer network has here been realised in sequential loops. Hence complete events are always synchronised and the timing performance can not be properly simulated in this approach. The simulation is used at the end of this section to study the resolution and efficiency of the CTD-SLT algorithm.

5.2 The Segment Finding

The segment finder is the first stage of the CTD-SLT and receives its input data from the FADC readout transputer (ROT). Because it sees all data in a crate the segment finder is particularly sensitive to the large input data volume and the hit data quality. In order to cope with this particularly demanding task the segment finder, which only uses integer arithmetic, runs on the fastest transputers available clocked at 30 MHz.

In order to control processing time the input data volume and the number of segments found must be limited. Fig. 9 shows the FADC data volume per crate. If only the essential axial (r, ϕ) data is used, the maximum volume is reduced by almost a factor of 4. However this is not enough as the time taken to transfer data across the transputer links is a significant contribution to the overall latency of the CTD-SLT as will be discussed in more detail in Sec. 6. The right-hand plot of Fig. 9 shows the FADC axial data per crate and the data volume actually used in segment finding before the maximum allowed number of segments is reached. The latter shows a sharply falling edge at about 60 words per crate and this value is used as a cut in the FADC-ROT sending process to limit both the data volume and transfer time.

Fig. 9. *FADC input data volume to the segment finder per crate. The left plot shows the reduction in data volume as a function of data type. The right plot shows how many words are actually analysed in the segment finder given the limit on the number of segments found, compared to all FADC axial data per crate.*

The segment finder operates on hits from one cell at a time, using the integer drift-time bins and wire numbers to find straight-line track segments in the $r - \phi$ projection. Since the sense wire plane and the drift direction are not orthogonal in a CTD cell, this system is called the 'local non-orthogonal coordinate system' (LNO). The algorithm is split into three tasks: the unpacking of cell data; the actual segment finding in a cell mask; sorting of the reconstructed segments in terms of increasing superlayer and cell number and segment output to the vector hit finder. Sorting the segments is not essential for the finding task, but saves considerable time in the following vector hit finder.

The segment finder proceeds as follows:

- First of all the segment finder receives the drift-time data from the FADC ROT. The data consists of a header and the cell data, which consists of a cell header word followed by the bit-packed wire hit data from the cell. Information from up to two hits is stored per word. The wire hits are ordered by increasing wire number and by increasing drift time on each wire. The drift time is always positive and given in units of DSP bins, which corresponds to 2.4 nsec.

At the unpacking stage the FADC T0 constant, arising from signal delay in electronics and cables, is subtracted from the drift times. Even though the T0 correction can vary slightly for the different wires, the CTD-SLT only applies a global correction, in order to save time and keep the correction simple. This T0 correction was 25 DSP bins in 1995 and 24 in 1996 - 1997.

Fig. 10. Schematic diagram of the segment finder algorithm in the LNO coordinate system.

- The actual segment finding starts at a pair of hits ('seed') on adjacent wires (see Fig. 10). In case of multiple hits only the first hit, the one with the smallest drift time, is considered for the seed finding. A seed in the centre of the cell is sought first. Using the wire pairs (4,5), (6,7), (2,3), (7,8), (1,2), (5,3), (3,1) the search is continued until a seed is found or all seven attempts to find a seed fail. In the latter case the cell is discarded and the data of the next cell are unpacked and processed.
- Based on a linear extrapolation of the seed pair, a prediction is made for the drift time on the next wire in increasing wire number. If a hit on this wire is found with a drift time lying in the segment road width of the prediction, this hit is added to the segment candidate and the procedure is continued, always extrapolating the last two hits. In 1995 to 1997 the segment road width was ± 30 DSP bins.
- If no hit is found within the road, the road width is increased by half its value and the search is continued. If more than 2 wires are left without matched hit or if the end of the cell is reached the segment finding is continued from the original seed pair, now in direction of decreasing wire number.
- The minimum number of matched hits for a valid segment is 3. This allows for the cell boundary crossing of a track and possible hit inefficiencies of the chamber. In most cases of cell boundary crossing segments are found in both cells.
- The hits on a valid segment are removed from the cell hit list and the segment finding in this cell is restarted. Segment finding in a cell stops when either less than 3 hits remain or 3 segments have been found.

- The segment finder finishes as soon as 6 segments, the upper limit per crate, are found or the entire hit data set in this crate has been processed. The segments found are ordered in increasing superlayer and cell number and sent to the vector hit finder.

The segment finder resolves the ambiguity between hits and ghost hits (Fig. 10) such that the segment direction points closest to the $r - \phi$ origin.

5.3 The Vector Hit Finder

The vector hit finder is the second stage of the CTD-SLT. Its purpose is to transform the segment information from LNO to CTD/ZEUS coordinates and to match z-by-timing hits to $r - \phi$ hits to form 'vector hits'. Since the coordinate transformation and corresponding approximations play a central role in this operation they are given first, followed by a description of the different algorithm steps.

Fig. 11. CTD system, showing a cell with LNO origin C and a segment at V .

Fig. 11 shows the geometry of a track and its segments in the CTD system. Assuming a uniform magnetic field, a track from the nominal interaction point projects onto a circle in the CTD $r - \phi$ view. A track segment with centre V

on this circle at a radial distance r_v and an angle Φ_v with respect to the origin satisfies:

$$r_v = 2R_0 \sin \chi ,$$

where R_0 is the track radius of curvature and χ is the angle between the radius vector and the segment. Both, R_0 and χ , are signed quantities, depending on the charge of the track. Φ_c is the angle of the CTD cell centre C, with $\Phi_c = \frac{2\pi n}{N_{SL}}$ ($1 \leq n \leq N_{SL}$) where $n \hat{=}$ cell number (see Fig.4) and $N_{SL} \hat{=}$ number of cells in superlayer SL. R_c is the distance of the cell centre from the (x, y) origin and depends on the superlayer of the cell. The angle between the segment and the line of sense wires is denoted by Θ_v . For convenience the angles $\alpha_v = \Phi_v - \Phi_c$ and $\Phi_0 = \Phi_v + \chi$ are also introduced.

Fig. 12 shows the detailed geometry of a segment in a CTD cell and the LNO coordinate system. The ionisation drifts towards the sense wires along the drift direction v , which is shown as dotted lines. The sense wire plane u is given by the eight sense wires, here indicated as stars. The angle between the sense wire plane and the drift direction is call Lorentz angle. It is usually close to 45° , but depends on the CTD gas mixture and varies slightly with external parameters such as the gas pressure and temperature. Hence this quantity is constantly monitored. The corresponding angle shown here is $\beta_d = 180^\circ - \text{Lorentz angle}$ or in radians $\beta_d = \frac{3}{4}\pi + \delta$, where $\delta = \frac{\pi}{4} - \text{Lorentz angle}$. For the design gas mixture $\delta = 0$ and hence $\beta_d = \frac{3}{4}\pi$. β_c is the angle between the sense wire plane and the radius vector. This angle is a CTD construction parameter and is identical for all CTD cells, $\beta_c = \frac{7}{4}\pi$.

Fig. 12. LNO system, showing the u (sense wire plane) and v axes (drift direction) and a segment BT . The angles are defined in the text.

In the LNO system the wires are numbered -3.5 to 3.5 . The coordinates of the bottom and the top hits in this LNO system , the hits with smallest and largest

wire number, are given by (u_B, v_B) and (u_T, v_T) in units of cm. The straight line $b = v - m \cdot u$ through these two outermost hits has slope $m = \frac{v_T - v_B}{u_T - u_B}$ and intercept $b = \frac{v_B u_T - v_T u_B}{u_T - u_B}$.

The LNO segment parameters m and b are related to those in the CTD coordinate system by [17]:

$$\begin{aligned}\Theta_v &= \arctan \left(\frac{m \cdot \sin \beta_d}{1 + m \cdot \cos \beta_d} \right) \\ r_v^2 &= R_c^2 + b^2 + 2bR_c \cdot \cos(\beta_d + \beta_c) \\ \sin(\alpha_v) &= \frac{b}{r_v} \cdot \sin(\beta_d + \beta_c) \\ \Phi_v &= \Phi_c + \alpha_v \\ \chi &= 2\pi - \beta_c + \alpha_v - \Theta_v \\ \Phi_0 &= \Phi_c + \alpha_v + \chi\end{aligned}$$

Since δ is small, the relations involving trigonometric functions are approximated by:

$$\begin{aligned}r_v &\simeq R_c \cdot \left(1 + \frac{b^2}{2R_c^2} - \frac{b\delta}{R_c} \right) \\ \alpha_v &\simeq \frac{b}{R_c} \left[1 - \frac{1}{2} \left(\frac{b^2}{R_c^2} + \delta^2 - 2\frac{b\delta}{R_c} \right) \right] \\ \Theta_v &\simeq \frac{\pi}{2} - \frac{1/\tan \Theta_v}{1 + 0.28 \cdot (1/\tan \Theta_v)^2} && \text{for } \tan \Theta_v > 1; \Theta_v \geq 0 \\ \Theta_v &\simeq \frac{\tan \Theta_v}{1 + 0.28 \cdot (\tan \Theta_v)^2} && \text{for } \tan \Theta_v \leq 1; \Theta_v \geq 0\end{aligned}$$

For negative Θ_v the symmetry of the tan-function is exploited. These approximations are valid to 1.4 % if the Lorentz angle is 45° , and to within 6 % if the Lorentz angle deviates from its design value by up to 10° .

The vector hit finding algorithm proceeds in the following steps :

- Read in segment finder output and z-by-timing data for the given crate.
- One segment at a time is unpacked and processed. Using the known spatial separation of the sense wires ($\sim 8 - 9$ mm), the time width of the FADC drift time bins (2.4 nsec) and the drift velocity ($\sim 48 \mu/\text{nsec}$) the LNO straight line slope (m) and intercept (b) are calculated in units of cm from the wire number and FADC drift time bin of the bottom and top hit on the segment.

- The z-by-timing data is read in ordered in terms of superlayers, cells and wires. A pointer to the z-data is arranged such that it points either to the z-data of the cell the currently processed segment lies in, or to the next cell in the z-data set. This structure is extremely efficient for finding the z-data provided that the ordering of the segment and z-data are compatible (hence the previous ordering of the segment finder output).
The z-data drift times are corrected for T0 time delay effects (12 drift time bins from 1993 to 1997). Spurious hits, such as noise hits or hits where either start or stop pulse are missing, are removed by requiring that the T0 corrected drift time is not negative and that the z-digitisation bin is in the range 0 – 127.
- For every FADC hit on a segment, a z-hit is matched if $\text{drift bin}(zBt) - 0.3 \leq \text{drift bin}(FADC) \leq \text{drift bin}(zBt) + 1.49$ where the FADC drift time bin (2.4 nsec width) is transformed into units of the z-by-timing drift time binning (48 nsec width). In case of several z-hits the one with the smallest drift time is chosen. The z-digitisation bin of every matched hit is transformed into a z-value in cm using the linear relation¹² $z = (z_{\text{bin}} + 0.5) \cdot \frac{203}{114} - 115$ which is an approximation to the precise relationship good to ± 5 cm over the length of the chamber of ± 1 m.
- The segment coordinates are then transformed from the LNO system to the CTD/ZEUS coordinate system, using the approximations described previously. The resulting parameters r_v , the radial segment distance to the origin, Φ_0 , the starting azimuthal angle of the track from which the segment originates, and χ , the angle between the segment and its radius vector, are stored for every segment.
- For segments in superlayer 7 and 9 the z-matching steps are skipped. The vector hit finder finishes its task as soon as all segments are converted into vector hits. Their order is kept in increasing superlayer and cell number. For monitoring and debugging purposes intermediate results can also be written out to the event data stream.

5.4 The Track Finder

The track finder is the third stage of the CTD-SLT. Its tasks are to find track candidates from vector hits matched between superlayers and to determine the parameters of valid tracks. The track finder operates on the vector hit data from two adjacent crates, covering a 45° sector of the CTD in $r - \Phi$, the so-called track finding mask. The track finder runs on 25 MHz transputers and is the central stage of the CTD-SLT. The quality of the CTD-SLT data

¹² Up to 1996 the offset in this relation was -113, which introduces a 2 cm bias of the track and event vertex towards the forward direction.

relies on the efficiency and precision of this stage, which has to be very robust and stable against small variations in the data taking conditions and the CTD response. The steps in the track finding algorithm are:

Fig. 13. *Schematic diagram of the track finding algorithm. A vector hit (thick short line) is matched to a track if it lies within the angular road, indicated by the shaded triangle, predicted by the last matched vector hit.*

- The reconstructed vector hits from crate N and crate N-1 are read in (see Fig. 8). If there are at least two vector hits, the minimum requirement for track finding, they are unpacked and the curvature of the corresponding track,

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

is determined for each vector hit.

- The track finding starts with a ‘seed’ vector hit (see Fig. 13). The search loop for the seed starts in the outermost superlayer, where the vector hits are usually well separated in space, and is carried out over the vector hits of this, or if necessary the inner superlayers, until a suitable vector hit, not yet matched to any other track, is found. The search for seeds finishes if the maximum number of tracks per mask, at present 6, is reached or if no candidate vector hits are left or if they have all been considered as seeds.
- Having chosen the seed vector hit the expected z-position for the next inner vector hit is calculated based on a linear interpolation to the nominal

interaction point, if previously matched hits provide z-information.

- Working inwards, the expected azimuthal position of the next vector hit, Φ_v^{expected} , is estimated using linear extrapolation based on the position (Φ_v) and the direction (χ) of the current vector hit.

$$\Phi_v(SL)^{\text{expected}} = \Phi_v(SL + 2) + \chi \cdot \omega(SL)$$

SL	1	3	5	7	9
$\omega(SL)$	0.30	0.25	0.19	0.16	0.0

The correction factors $\omega(SL)$ to χ are extracted from single track Monte Carlo and selected cosmic events, recorded in 1993. They decrease with increasing superlayer. The least precise prediction is for the vector hit position in superlayer 1. This SL usually contributes the largest number of z-hits to a track, while its lever arm in (r, z) is shortest. Hence effects of possible mismatches are largest here. If vector hits have already been matched in superlayer 3, 5 and 7 a parabolic approximation of the track trajectory projection in (Φ_v, SL) -space is applied, which is more stable and precise than the χ -based approach :

$$\Phi_v(SL = 1)^{\text{expected}} = 1 \cdot \Phi_v(SL = 3) - 3 \cdot \Phi_v(SL = 5) + 3 \cdot \Phi_v(SL = 7)$$

- A vector hit is matched to the track candidate if

$$(\Delta\Phi_v \leq (\Phi_v - road = 3.44^\circ)) \text{ .AND. } (\Delta\kappa < (\kappa - road = 0.04) \text{ .OR. } SL = 1),$$

where $\Delta\Phi_v$ and $\Delta\kappa$ are the absolute differences of the vector hit quantities and the expected Φ_v or the κ -value of the last hit. All vector hits originating from the same track should have the same κ -value. Out of up to 4 matching vector hits the best one, i.e. the one with the smallest

$$\Delta_{product} = \Delta z + 125 \cdot \Delta\kappa + 1250 \cdot \Delta\Phi_v \text{ is chosen.}$$

- As depicted in superlayer 1 of Fig. 13 a track can produce two short vector hits in adjacent cells of one superlayer when crossing a cell boundary. These split vector hits each carry essentially half of the z-information in that superlayer. Hence a second vector hit in a superlayer can be matched if $\Delta\Phi_v < 1^\circ$, $\Delta\chi < 15^\circ$ and $\Delta z < 40$ cm (if available).
- The most recently matched vector hit is treated as a new seed and the matching procedure is repeated until the innermost superlayer is reached.
- A matched set of vector hits is accepted as a valid track if it consists of either a vector hit in superlayer one plus at least one more hit in a higher superlayer or a vector hit in superlayer three with at most one gap in the higher superlayers. Track candidates consisting only of vector hits from the adjacent crate are rejected since they will be found in the adjacent overlapping track mask. Vector hits from a valid track are removed from the hit list in the track finding mask.

Once a track is accepted as valid its parameters are determined as follows:

- The track radius of curvature, R_0 , is determined using a fast circle fit in (r, Φ) as follows. Let (r_i, Φ_i) be the set of vector hit parameters r_v and Φ_v on the track. R_0 is then given by :

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

where

$$\begin{aligned} a &= \sum_i (r_i)^2 \cos^2 \Phi_i & d &= \sum_i (r_i)^3 \sin^2 \Phi_i \\ b &= \sum_i (r_i)^2 \sin^2 \Phi_i & e &= \sum_i (r_i)^2 \sin^2 \Phi_i \cos^2 \Phi_i \\ c &= \sum_i (r_i)^3 \cos^2 \Phi_i & f &= \sum_i (r_i)^2 \end{aligned}$$

- Subsequently the track vertex and polar angle Θ are calculated with a weighted two-step least squares straight line fit in (r, z) . This fit is designed such that it is as precise as possible, whilst reducing the impact of outliers and spurious hits, which show up in the z-by-timing system from time to time. Concerning the track gradient (Θ -angle) all points are of equal importance, but on the intercept (vertex) mismeasurements in z of inner points have a larger effect than those of points further out in radius. Therefore the weight of points with $r_i > 25$ cm is doubled ($\omega_i = 2$). After a first calculation of the straight line gradient and intercept in (r, z) , z-hits are declared to be outliers and rejected if

$$\left(\frac{dev_i^2}{sigma} > 3 \right) .AND. (dev_i^2 > 100 \text{ cm}^2),$$

with

$$\begin{aligned} dev_i &= z_i - gradient \cdot r_i - intercept \\ sigma &= \sum_i \omega_i \cdot dev_i^2 \end{aligned}$$

A second straight line fit is performed on the new hit set, resulting in the track vertex and the track direction $\cot \Theta$.

5.5 Merging of the Track Finder Output

Each of the 16 track finder processes performs its pattern recognition and track fits independently. In order to transfer the results to the last step of the CTD-SLT, the vertex finder, a merge process, running on the track finder

transputers, merges the output data from the local and the adjacent track finder and passes it on to the next crate. This sequential data merging is done in blocks of four crates so that the merged data volume does not get too large and a certain degree of parallelism is kept (see Fig. 8). The advantages of this arrangement for minimising latency are discussed in Sec. 6. The four data blocks are sent to the SLT merge transputer and finally combined on the ‘SLT MERGE 1’ transputer, where the vertex and event reconstruction algorithm runs.

Since the GSLT only accepts up to 20 CTD-SLT tracks and each track can be found twice due to the overlapping 2-crate track finding masks, the data of at most 40 tracks need to be merged and transferred to the vertex finder. This cut is applied ‘on the fly’ while merging the data, so that only a small amount of processing is required resulting in a limited data volume to be transferred, in particular for high multiplicity events.

5.6 *The Vertex Finder*

The vertex finder is the last stage of the CTD-SLT. Its tasks are threefold:

- Due to the overlapping 2-crate masks used in the track finding, tracks can be found twice. These duplicate tracks have to be removed from the list. Since the GSLT can only receive a certain data volume from each component, the CTD-SLT can only send up to 20 tracks per event at present. This limit has to be respected and tracks have to be selected.
- Further track parameters to be used by the physics filters on the GSLT have to be determined.
- The event vertex has to be determined and possible characteristics of beam gas events have to be recognised.

The vertex finder is different from the first three CTD-SLT stages as it has access not only to crate or mask data, but to all tracks reconstructed in the entire CTD. Therefore the algorithm, running on a 25 MHz transputer¹³, is accommodated in the regional box (RBOX), where the CTD-DAQ system is coordinated and controlled (see Fig. 8). The vertex finder proceeds as follows:

- A loop over all tracks, found in the 16 track finders, operates until either the end of this data set is reached or the maximum number of tracks that the GSLT can cope with (at present 20), has been processed.
- Each track is compared to all tracks reconstructed and stored so far. A track is declared to be a duplicate one and hence rejected if it has at least

¹³ From October 1997 on a 30 MHz transputer was also installed on this CTD-SLT stage.

one vector hit in common with a track already stored. Otherwise the track reconstruction continues.

- The track vertex has been determined by the track finder. The charge (± 1) is stored as the sign of R_0^2 , which is otherwise obviously a positive quantity. The transverse momentum in GeV is given by

$$p_T = \sqrt{|R_0^2|} \cdot 0.003 \cdot B ,$$

where B is the magnetic field in the CTD in Tesla and R_0^2 the squared radius of curvature given in cm^2 . An upper limit of 157 GeV is placed on p_T , so that numerical instabilities in the calculation do not crash the system.

- For extrapolation of the track to the outer detectors the CTD exit point and direction are estimated as shown in Fig. 14.

$$\begin{aligned}\Phi_v &= \Phi_0 - \chi \\ x_{\text{exit}} &= r_{\text{exit}} \cdot \cos(\Phi_v) \\ y_{\text{exit}} &= r_{\text{exit}} \cdot \sin(\Phi_v) \\ z_{\text{exit}} &= r_{\text{exit}} \cdot \cot \Theta + z_{\text{vtx}} \\ \Phi_{\text{exit}} &= \Phi_v - \chi = \Phi_0 - 2 \cdot \chi\end{aligned}$$

Fig. 14. Geometry of track exit point in the CTD.

- The event vertex is determined by the so-called ‘binning-algorithm’, developed in 1994. The basic idea is to divide the z-range of the CTD in 21 bins, each 20 cm wide and overlapping with its adjacent bins on either side by 10 cm (see Fig. 15). The number of tracks originating in each bin is counted, which gives a distribution with a maximum very close to the event vertex. The centre of the bin containing the maximum is called the most-probable vertex (MPV). The MPV is determined as follows. The entries in the first and last four bins are weighted by 0.5 in order to select a more central vertex in events that show beam gas as well as physics characteristics. Then the z-bins with maximum entries in the forward (bin 2 - 11) and backward region (bin 11 - 20) are determined separately. The MPV is the bin centre of the maximum of these two. If they contain equal number of tracks the one closest to the origin is chosen. In case of equal distance to the origin the one with the largest number of entries in the adjacent bin towards either end of the CTD is chosen. Finally if any ambiguity remains the forward vertex is declared to be the MPV (this introduces a negligible bias).

Fig. 15. Schematic diagram of the vertex finding algorithm ('binning algorithm'). top: An example constellation of tracks as reconstructed by the track finders is shown. bottom: The z-range of the CTD with the applied binning can be seen. The small numbers indicate the bin-number, while the bold numbers represent the number of tracks originating from this particular bin. In this example the most-probable vertex (MPV) lies in bin number 11.

- It is then decided whether an event is beam gas background or physics. The following quantities are used:

nr_tracks	=	the number of tracks found by the SLT
MPV	=	the most probable vertex, in the centre of the bin with the most entries.
vtx_tracks	=	number of tracks coming from the bin next to the MPV-bin
beam_type	=	number of tracks with $ \text{gradient} \geq 1.0$ and $ z_{\text{vtx}} \geq 95.0$ cm.
product	=	Average product of track gradient and track intercept, characterising the global event tracking topology. This quantity is close to zero for physics events and can get very large and negative (~ -500) for beam gas events.
z_bin(index)	=	number of tracks pointing to bin number 'index', where the index numbering is done according to Fig. 15.

An event is declared to be a beam gas type event if one of the following conditions is fulfilled:

- $((z\_bin(1) \geq 2) \text{ .OR. } (z\_bin(21) \geq 2)) \text{ .AND. } (z\_bin(2 \dots 20) = 0)$

- $(z\_bin(1) + z\_bin(21) - z\_bin(MPV) \geq 10) \text{ .AND. } (vtx\_tracks \leq 4)$
- $(\overline{product} \leq -500) \text{ .AND. } (nr\_tracks \geq 8) \text{ .AND. } (vtx\_tracks \leq 4) \text{ .AND. } (beam\_type \geq 4) \text{ .AND. } (beam\_type \geq 0.5 \cdot nr\_tracks)$

Events, which are declared to be of beam gas type are flagged by setting the SLT vertex +100 cm by default. If an event does not have at least one track originating from the CTD-region and is not declared to be of beam gas type, the CTD-SLT cannot draw any reliable conclusion whether it is a physics or a background event. These events do not contain tracks, which could be reconstructed by the CTD-SLT. Hence the events are either empty, as far as the tracking is concerned, or the tracks go in the very forward or backward direction and leave hits in the CTD only in superlayer 1, 2, and maybe very few in superlayer 3. The CTD-SLT does not reject these events but flags them as ‘undecided’ by setting the vertex to 9999.0 cm.

Reconstructed track and event data are sent to the GSLT for further processing and for offline monitoring of the CTD-SLT performance.

5.7 Resolutions and Efficiencies

This sub-section covers the CTD-SLT tracking efficiency and resolution for the period 1995 to 1997 during which the algorithm was relatively stable. Further details are to be found in Quadt [18, Ch. 5].

5.7.1 Track Finding Efficiency

The CTD-SLT is designed to find long tracks of high transverse momentum. On the trigger level it is more important to obtain an impression of the event topology than to find every single track. For low track multiplicities the SLT and offline track reconstruction code agree rather well in the number of reconstructed tracks, while the SLT tends to find a lower fraction of tracks in high multiplicity events. The major differences stem from the limits in the SLT on the hit data volume and the maximum number of segments and tracks allowed per sector. The differences are stronger for negative than for positive tracks because of the asymmetry of the CTD geometry.

The question under investigation here is, how the single track finding efficiency depends on the track finding algorithm, the CTD geometry and various approximations. A second parameter, indicating the stability and quality of the CTD-SLT, is the fraction of tracks found twice (‘split tracks’). These two quan-

tities are obviously closely related, since a track can only be found a second time if not all of its vector hits have been matched the first time.

The single track finding efficiency depends mainly on the number and quality of segments found. Track losses are mostly caused by geometrical effects: segment losses due to cell boundary crossing of negative charge or low- p_T tracks; segments pointing in the wrong direction due to mismeasured FADC hits at the cell ends; segments being tilted or shifted due to shifts in the Lorentz angle or the T0. In addition the requirement of matching at least two vector hits to a track limits the CTD-SLT acceptance at small and large polar angle Θ and hence its track finding efficiency in the very forward and backward direction.

These effects have been studied in detail on a sample of 50000 single track $\mu^\pm$ Monte Carlo-events generated over the full CTD acceptance in polar angle Θ and for $0.03 \text{ GeV} \leq p_T \leq 4 \text{ GeV}$.

Fig. 16. Single track finding efficiency (filled circles) and fraction of split tracks (open circles) as function of p_T (left) and Θ (right).

As depicted in the left-hand plot of Fig. 16 by the filled circles the single track finding efficiency in the central CTD region ($45^\circ - 135^\circ$) rises very quickly for negatively charged particles with low p_T and reaches $\simeq 95\%$ at about $p_T = 250 \text{ MeV}$ and even close to 100% level is reached from $p_T = 800 \text{ MeV}$ onwards. The efficiency for positively charged particles with low p_T (not shown) reaches the 95-100% level already at about $p_T = 250 \text{ MeV}$. The fraction of split tracks, shown as open circles in the diagram, scatters around the 10% level, independent of track charge, but rises steeply at p_T values below 200 MeV. These tracks curl so strongly in the CTD, that the segment finder makes the wrong choice in resolving the drift ambiguity. The trigger is simply not designed to find such low- p_T tracks.

For tracks of reasonably high transverse momentum ($p_T \geq 400 \text{ MeV}$) the track finding efficiency rises rapidly at a polar angle of $\Theta \simeq 17^\circ$, which corresponds to superlayer 3 in the CTD as seen from the nominal interaction point (see

the right-hand plot of Fig. 16). The efficiency stays almost constantly at the 100% level up to a polar angle of $\Theta \simeq 162^\circ$, where it drops quickly to zero due to the requirement of at least two matched vector hits per track and the superlayer 3 boundary at this Θ -value. The rate of split tracks scatters around 10% and 15% for negative and positive tracks, independent of Θ .

Variations of the Lorentz angle by up to $+18^\circ$ or -4° have no impact on the single trackfinding efficiency while the rate of split tracks varies in this range by at most 5%. The Lorentz angle is designed to be 45° and varied over the last four years by less than 2° either way. The efficiency has also been found to be stable against variation of the FADC time offset T_0 by up to 10 DSP bins, i.e. 24 nsec. The fraction of split tracks varies between 8 and 11% in this region and shows a slightly stronger dependence for negative tracks, again due to the CTD geometry since the corresponding segments cross the sense wire plane.

5.7.2 Z-Vertex Resolution

A parameter with very powerful potential for the discrimination of background events is the z-vertex position. For this potential to be realised in the online selection of events the SLT reconstruction of the z-vertex position must be very reliable and give the best possible resolution.

Since the CTD-SLT determines the event vertex with a binning algorithm based on track vertices, as described previously, their quality and quantity will influence the event vertex resolution most. It is naively plausible, that the binning algorithm will work best if there are enough data to be binned, i.e. in events with high track multiplicity. On one- or two-track events the expected gain from the binning compared to a simple track vertex average is small.

The resolution of the CTD-SLT event vertex in comparison to that reconstructed by the offline code, which has a resolution of ~ 1 mm for most events, is shown in Fig. 17. The gaussian fit to the distribution gives a χ^2/ndf of 0.5 and a width of 8.8 cm. Given that an SLT track can only have up to 8 z-hits assigned to it, of which about half are usually lost due to cell boundary crossing of the track, spurious hits or noise of the z-system, and recalling that these points are spread only over a lever arm as short as 30 cm in radius, this resolution is quite an achievement.

For one-track events the resolution is found to be only ~ 12 cm with a clear 3 – 4 cm shift towards larger SLT vertices. But as soon as five or more tracks are sent to the GSLT, the vertex resolution is $\sim 8 - 9$ cm, essentially without any noticeable shift. The dependence of the vertex resolution on the fraction of tracks assigned to it has also been studied. With only 10% of tracks assigned to the vertex, the resolution is almost 14 cm, with a 4 cm shift in addition.

Fig. 17. *Event z-vertex resolution of the CTD-SLT in comparison to the offline reconstruction.*

The rapid improvement in resolution to ~ 6 cm for 40% vertex-track events gets partially compensated in events with an even higher fraction of vertex tracks. This is mainly due to the fact these events tend to be dominated by low track multiplicities. The resolution of more than 12 cm for single-track events reduces to 4 – 5 cm if 9 tracks are assigned to the vertex.

Another interesting question is, how the z-vertex resolution varies with the z-position of the vertex. The resolution is observed to be basically independent of the z-vertex position in the range of ± 30 cm; further away from the nominal interaction point the rapidly falling event statistics prohibits a more detailed investigation. However, there is a systematic shift of the event vertex with the vertex position, which reaches up to ± 6 cm at $z = \pm 15$ cm. This effect is exactly what one expects from the straight line approximation, used in the CTD-SLT, of the measured z-by-timing time to distance ‘s-shape’.

5.7.3 Transverse Momentum Resolution

Another important parameter for the discrimination between interesting physics events and background, in particular for the online selection of heavy flavour event candidates, is the transverse momentum (p_T) of the track. In order to be able to select or reject events online based on the track p_T the reconstruction must again be reliable and precise, the resolution good and reproducible.

The dependence of the p_T -resolution on the track characteristics such as charge and T0 correction, has been studied using the same sample of 50000 single track $\mu^\pm$ Monte Carlo events. Fig. 18 shows the p_T -resolution for the different track charges. The overall p_T -resolution for negative tracks is $\frac{\Delta p_T}{p_T} \simeq 4.5\%$,

Fig. 18. Transverse momentum resolution for single track muon Monte Carlo events. The resolutions obtained from gaussian fits to the distributions for positively and negatively charged particles are given in the top right corner of each plot.

compared to which the offline reconstruction code [8] with full calibration and single wire corrections which is only a factor of 10 better. For positive tracks the resolution is $\frac{\Delta p_T}{p_T^2} \simeq 6.7\%$, which is slightly worse due to the CTD geometry and the approximations made in the CTD-SLT algorithm. For short tracks with only two or three vector hits length the resolution is found to degrade by almost a factor of 2 as a constrained circle fit is only just possible with 2 vector hits.

If the Lorentz angle is shifted slightly ($\Delta\beta$) from its true value the reconstructed p_T shows a small bias (less than 2%) which depends on the sign of the particle and linearly on $\Delta\beta$. The p_T resolution stays essentially unchanged for $|\Delta\beta| < 10^\circ$. The p_T -resolution also shows a rather strong dependence on the correct setting of the T0-correction while the central value is less sensitive and more stable for negative than for positive tracks.

Since the CTD-SLT p_T reconstruction algorithm assumes that all tracks originate from (0, 0) in (x, y) , i.e. that the beam crossing is at its nominal position, a deviation from this assumptions distorts the p_T resolution. For maximum (x, y) vertex shifts of 2 mm the p_T resolution can increase from 4 to 6%.

6 The Timing Performance of the CTD-SLT

6.1 Introduction

Speed and latency characterise the timing performance of a trigger system and must therefore be controlled and understood. They are often confused and interchanged, even though their meaning is very different, in parallel systems almost orthogonal. After a definition of the two terms, their history and

present status in the CTD-SLT are discussed. The CTD-SLT latency is compared to a toy-model simulation, which predicts the latency in several online running scenarios, and is finally compared to the CTD readout latency.

The speed (or rate) of a system is the average number of events that can be processed per unit time. The design figure for the SLT is 1 kHz, but due to careful tuning of the FLT filters the highest SLT input rate observed so far is close to 600 Hz. The speed is an average quantity describing the systems performance over a long period, eg a run.

The latency is the time between a GFLT accept signal and the time, when the component provides its trigger information to the GFLT. It is an event specific quantity and includes the times for data readout, transfer and processing. Dead time or waiting time and hence latency might be inherited from the previous event, which keeps a particular CTD-SLT stage busy. For the ZEUS-SLT the component's latency is supposed to be less than 10 – 16 msec in order to keep the DAQ dead time $< 1\%$.

Fig. 19. Analogy between a bucket chain and the CTD-SLT, illustrating the meaning of speed and latency.

The answer to the question why speed and latency do not contain the same information in parallel systems is illustrated in Fig. 19. The bucket chain is analogous to the CTD-SLT. Supposed one person, Mr. X, can only handle N buckets per minute, then even if everybody else could ‘process’ $10 \cdot N$ or $100 \cdot N$ buckets per minute, the entire chain could never deliver more than N buckets per minute. The effect here as well as in the CTD-SLT is, that the overall event rate in the system is entirely determined by the slowest step. In fact, the rate is the inverse of the average processing time on this stage of the system. The faster stages simply have to wait from time to time.

It might be helpful to consider two extreme cases in this context:

- Adding an arbitrary number of stages, all faster than Mr. X, does not affect the average number of events processed per time unit, the trigger rate remains constant. However such a measure would increase the total processing time for a particular bucket or event. In a similar way all, but Mr. X, might suddenly speed-up by a factor of 100. The overall event rate is again not

affected, but the event latency is vastly reduced.

- At the other extreme, a scenario might be constructed, where twice as many processing steps or people share the event processing task, where the processing on each step only takes half the time, it took before. The total processing time adds up to the same value as before, but the slowest step is now twice as fast. The event rate has been doubled without changing the event latency.

In reality a mixture of these two extreme cases occurs. Nevertheless tackling a low rate or a large latency require slightly different approaches.

6.2 *The CTD-SLT Speed and Latency*

The maximum event rate that the current CTD readout and trigger system can cope with has been estimated to be around 800 Hz using ZEUS test runs [19]. This result is also consistent with studies performed using the CTD-SLT playback system. However such extrapolations of rates from controlled data taking conditions to unfamiliar territory with unknown conditions and effects is a difficult task for a system as complex as the ZEUS trigger. As important, if not more so, has been the understanding of the factors contributing to the latency of the CTD-SLT and how to control them.

When running with FADC data for the first time in 1993, the CTD-SLT latency had a long tail, reaching well beyond 90 msec (Fig. 20). After improvements to the code and the data transfer structures the CTD-SLT latency in 1995 was reduced to ~ 24 msec. Only small changes were made for the 1996 data taking period, so that a similar distribution was expected and seen. The latency distribution in 1997 and the effect of changes made to the code are shown and discussed in Sec. 6.4 as they result from Monte Carlo studies described there.

A quantity, playing a crucial role for the latency, is the volume of track data to be transferred. The track data, as found in the 16 crates, have to be merged into one data set before the vertex finder can start its processing. In 1994 this task was done by sequential sending and merging of the track data from crate 16 to crate 15, 14 and so on. The data volume transferred per event resulting from this multiple sending procedure could add up to as much as 6000 words. Data transfer via the transputer links took up to ~ 13 msec for such events. In 1995 the track data from the crates 1 to 8 and 9 to 16 were merged in parallel, with amounts of data up to 2000 words per event taking a maximum of 4.4 msec to transfer. Finally in 1996 and 1997 the data of crate 1 to 4, 5 to 8, 9 to 12 and 13 to 16 are merged in parallel and sent to the vertex finder via the tree-structure of transputers shown in Fig. 8. Even though the actual data

Fig. 20. History of the CTD-SLT latency distribution. The CTD-SLT was used on the GSLT only after the latency was reduced below 25 msec.

volume dealt with is again reduced considerably, the fact that one additional send-process is required limits the improvement somewhat. The total track data volume transferred can be as large as 1250 words, which corresponds to a transfer time of ~ 2.8 msec.

A large CTD-SLT latency can cause other components to wait and fill up their input buffers which in turn leads to deadtime in the overall trigger and readout system. In order to avoid such a situation, the CTD-SLT latency has to be reduced even further which is a challenging task for the future.

6.3 Online Statistics

In order to gain insight into the timing behaviour of the transputer network an online statistics system was introduced in 1995. The average and maximum processing and the average data input waiting time in μsec along with the input and output volumes in words for the 4 CTD-SLT stages are determined by the online OCCAM code for regular physics data taking. A summary from a typical run in 1996 is shown in Table 1.

The essential characteristics are:

Process	ave time [μ sec]	max time [μ sec]	input 1 vol [words]	input 2 vol [words]	input 1 wait [μ sec]	input 2 wait [μ sec]
SEG	842	3584	21	-	1633	-
VHIT	784	3712	9	32	837	831
TRK	933	4416	33	33	1275	247
VTX	1116	6016	114	-	1363	-

Table 1

Online statistics: The average and maximum processing and waiting times in μ sec as well as the input data volumes in words for the the 4 major CTD-SLT processes are shown for a typical run in 1996. For VHIT, input1 and input2 refer to z-by-timing and segment data respectively; for TRK, input1 and input2 refer to the first and adjacent crate in the track finding mask.

- The average and maximum times and data volumes do not vary considerably from sector to sector and are therefore not shown separately. This is expected due to the average ϕ -symmetry of the ep -scattering events. However on the vector hit and track finder the crate-to-crate fluctuation is larger, as 25 MHz and 20 MHz transputers were mixed on these stages until October 1997, when almost all transputers were upgraded to 30 MHz.
- The maximum processing times are about 4.5-times as long as the average values, on the vertex finder even 6-times as long.
- The average input data volumes are consistent with the estimates given previously.
- The long input waiting times indicate that the CTD-SLT event rate was higher than the trigger input rate for this particular run, the system was not saturated.

Using the data volume measurements convoluted with the design specifications of link speeds, the maximum CTD-SLT latency can now be estimated (table 2). The maximum time taken to read out an event by the FADC-ROT is assumed to be ~ 3 msec (the average time has been measured to be slightly below 1 msec). The assumption is also made that the data of an event is processed immediately at every single CTD-SLT stage, no waiting time due to busy transputers is taken into account.

This resulting value of 24.3 msec is compatible with the falling edge of the 1995/96 latency distribution. It is therefore taken as an indication, that no additional unexpected effects slow down the transputer network and that the maximum latency is reproducible.

Processing Step	max. time [msec]
FADC ROT	3.000
FADC data transfer to segment finder	0.444
segment finder	3.584
segment data transfer to vector hit finder	0.220
vector hit finder	3.712
vector hit data transfer to track finder	0.110
track finder	4.416
track data merge	2.800
vertex finder	6.016
maximum CTD-SLT event latency	24.302

Table 2

Estimation of the maximum CTD-SLT event latency, based on studies of the single step processing times and data volumes.

6.4 Toy-Model Simulation of the CTD-SLT Latency

Apart from the position of the falling edge, the overall shape of the latency distribution with its central peak was not understood for a long time. The transputer processing time is essentially a linear function of the input data volume and all input data distributions fall monotonically. Hence one would naively expect the latency distribution to be monotonically falling too. However the CTD-SLT is a non-linear network. The main origin of the change is the combination of two crates to a track finding mask and the final merging of the track data. For example the track finder can only start the event processing if the vector hits from this and the adjacent crate are available. Suppose no segments are found in this crate. The segment and vector hit finder will be very fast. But the empty vector hit table from this crate will have to wait for the processing in the next crate to finish, i.e. up to 7 msec, before the track finding is started and some track data passed on to the merging process. At this stage similar waiting phases might occur again.

A toy-model [20] has been developed with the following features (see also Fig. 21):

- Random numbers are generated for the 16 crates according to the positive half of a gaussian of width 2.13. Values larger than 7 are set to 7. The resulting distribution has a mean of 1.7 and represents the sum of the times spent on the segment and vector hit finders per event and crate in msec.

Fig. 21. Timing distributions for the different steps in the CTD-SLT toy-model simulation.

- This number is scaled down, yielding a distribution of identical shape with mean 0.94, representing the FADC-ROT readout time in msec. This is consistent with the claim that the FADC readout can run at 1 kHz.
- The time spent on the track driver in msec is the sum of segment and vector hit finder times in this and the next crate, scaled by $\frac{4.3}{14.0}$. The mean of the resulting distribution, which already shows the changed shape, is 1.05. The number of found tracks is given by $INT(track\_time^{1.8} \times 0.35)$.
- 15 words are assumed to be stored for each track. The time when the track merging is finished, summing over all processing times and the track data transfer time via the 1.8 Mbytes/sec transputer link, is shown for crate 16 (shaded histogram) and crate 8 (dots). The shoulder at the high latency end is due to the limit of 40 tracks in the data transfer (since each track can be found twice and only up to 20 tracks are sent to the GSLT anyway).
- The time spent on the vertex finder is a quadratic function of the track data volume. The mean value of the distribution is 1.4, the maximum 6.5 msec.
- The data transfer time from the vertex finder to the GSLT is a linear function of the CTD-SLT output data volume, assuming 1.8 MBytes/sec transputer link speed.

Adding all these times together gives the final CTD-SLT latency distribution, which not only has a falling edge at 24 msec, but also reproduces the general shape of the real distribution. Given that the toy-model starts from a gaussian-

shaped input, which does not have the additional peak at low data volumes from cosmics, noise and events without tracks, the model appears to describe the online observed latency distribution surprisingly well. The model can serve as a useful tool in predicting the effects of changes to the CTD-DAQ and trigger system. As an example the following scenarios were studied during the development of the CTD-SLT in 1996:

scenario 1 The code and transputer network as in 1996.

scenario 2 The code runs only on FADC input data, an old z-by-timing data option has been removed. Instead of a minimum of 3 hits per cell now a minimum of 3 hit wires per cell are required for cell processing to start. This saves a lot of time since many noise cells are only unpacked but not processed. As soon as the maximum number of segments per cell (currently set to 3) is found, the processing of this cell is stopped immediately, all book-keeping and pointer management tasks are dropped. As soon as the maximum number of segments per crate (currently set to 6) is found, the processing of this event is stopped and the segment output sorting is started. None of these changes affect the output of the segment finder at all. Internal looping, data storage and book-keeping are optimised.

In addition the coding of the coordinate transformations used in the VHIT finder was modified and some calculations replaced by look-up tables.

scenario 3 as for scenario 2, but in addition the vertex finder processing parameters are set to be 0.85 msec (mean) and 2.0 msec (max), representing the performance of a faster CPU for this stage.

As can be seen from Fig. 22, to effect a significant reduction in latency both the changes to the segment finder and a faster vertex finder CPU are needed.

Fig. 22. CTD-SLT latency distribution for the 3 scenarios described in the text as obtained by the toy-model simulation.

In August 1997 the segment finder code was replaced by the faster version described in scenario 2 above. Given that the vector hit finder code remained

Fig. 23. CTD-SLT latency for two consecutive runs in 1997 with the standard code (dots) and the fast segment finder code (shaded histogram).

unchanged the latency reduction as expected according to the toy model simulation is slightly less than the difference between scenario 1 and scenario 2. Fig. 23 shows the latency distribution for two consecutive runs, taken in August 1997 under identical beam and rate conditions with the two different code versions. The implementation of the faster segment finder reduced the mean latency by ~ 1 msec to 10.7 msec while the maximum latency was reduced by 2.2 msec to 25.1 msec. In October 1997 all 25 MHz transputers were replaced by 30 MHz ones, reducing the latency tail by another 1 – 2 msec. These results demonstrate the usefulness of the toy model simulation and its predictive power. With it, the playback system and events of particular track

topology or custom-designed hit patterns in CTD cells a deep understanding of the CTD readout and trigger timing behaviour has been obtained.

7 Physics Applications

This section describes the use of CTD-SLT information for selection of events for physics analysis. First a brief account is given of the CTD-SLT data quality monitoring.

7.1 *The CTD-SLT Monitoring*

As demonstrated in the previous sections the quality and precision of the CTD-SLT reconstruction depends on CTD intrinsic parameters such as the Lorentz angle or T0 correction as well as on the status of the readout electronics, in particular the transputer network. In order to be able to use CTD-SLT information in online event selection its resolution and efficiency must be understood. In addition its quality has to be monitored and verified continuously during data taking.

For this purpose routines have been added to the general CTD monitoring system which runs test and calibration code on every 10th tape (one tape records ~ 2000 events) as they are written online. With this tool the data quality can be investigated, dead or ringing channels be identified and parameters like the Lorentz angle be determined. The CTD-SLT part of the monitoring produces histograms of all event and track parameters along with the z-vertex resolution in comparison to that from Third Level Trigger tracking and the CTD-SLT latency. The comparison to reference histograms reveals anomalies, so that corrective measures can be taken almost immediately. In addition to the online monitoring, the CTD-SLT simulation is run on the data offline. Comparison of the output tables from the online and offline processing entry by entry confirms the reproducibility of the trigger results, the track order as well as their parameters. Almost all parameters can be reproduced by the simulation to 99%. The agreement in transverse track momentum is only $\sim 95\%$ due to the different representations of floating point numbers on the transputers and in FORTRAN.

7.2 *Event Display*

A powerful tool for understanding how the CTD-SLT performs, particularly in comparison to the full offline tracking and how its information may be

used in physics event selection is the 3-dimensional version of the ZEUS event display LAZE (Look at ZEUS events). Figs 24 and 25 show 1995 and 1996 example events. The quality of the track reconstruction and increased discrimination power of the trigger, if track and CAL information is combined, are demonstrated:

- Figure 24 shows a candidate J/Ψ event from photoproduction, in which the J/Ψ is identified by its two-body decay to an e^+e^- pair. Only two final state particles are detected by the CTD and the CAL. The final state electron and positron deposit energy in the calorimeter (pyramid shaped objects) which are matched by a CTD-SLT track. For the identification of such events the tracking information plays obviously a crucial role since there is very little information from the calorimeter.
- Figure 25 shows the (r, ϕ) view of a 1996 neutral current DIS event with $Q^2 \sim 1000 \text{ GeV}^2$ and a high charged particle multiplicity. The scattered electron is identified by the short stiff track pointing towards the large electromagnetic calorimeter cluster in the top left-hand corner. The track finding ability of the CTD-SLT in a complex event is demonstrated.

These example events illustrate the overall efficiency and quality of the CTD-SLT track reconstruction. Event classification by eye is to a large extent based on the track topology and the matching behaviour of tracks and energy deposits in the calorimeter. The chosen events also give an impression on how the CTD-SLT information may be used in online event classification.

7.3 Physics Event Selection

As demonstrated by the ZEUS event display examples in the previous section tracking information either alone or in combination with other components is very valuable to the Global Second Level Trigger and increases its discrimination power significantly. From 1995 onwards all but the deep inelastic scattering¹⁴ physics filters used the CTD-SLT information in their event selection. Typical filter conditions for the different physics classes are given below.

Forward Muons Events with forward going muons are characterised by hits or track elements in the forward muon chambers, compatible energy deposits in the calorimeter and a good vertex in the CTD. The SLT requirements

¹⁴ The DIS filter does not use tracking information because of the very high efficiency of recognising the large electromagnetic energy deposit from the scattered beam electron in the calorimeter and the limited acceptance of the CTD at angles close to the beam line.

Fig. 24. 1995 J/Ψ in photoproduction candidate.

for such events are: the event passes the FMUON-SLT criteria; if a CTD-SLT event vertex is found it has to satisfy $|z_{vertex}| < 75$ cm. Events without tracks in the geometrical acceptance of the CTD-SLT are also accepted, but a large fraction of beam gas event and halo muons are rejected using this cut.

Soft Photoproduction (2 track decays of vector mesons). 2 track vector meson events are characterised by 2 offline tracks in the CTD and very little energy deposit in the calorimeter. As the vector mesons tend to follow the exchanged photon in the interaction, which in turn predominantly travels in incoming electron direction, the two decay products of vector mesons are almost always found in the rear direction of ZEUS. Apart from the two decay particles the event is empty. Due to the low energy deposits of the vector meson decay products it is very difficult to trigger on these events, in particular a calorimeter-only trigger is clearly not sufficient. These characteristics are reflected in the GSLT selection criteria: The number of CTD-SLT tracks has to be less than 4 (taking into account track splitting); if a CTD-SLT event vertex is found it has to satisfy $|z_{vertex}| < 60$ cm. If the tracks are at too shallow an angle for reconstruction these events are characterised by a small number of unmatched vector hits.

Hard Photoproduction Hard photoproduction events are characterised by a low $E - P_z$ due to the outgoing electron being lost down the rear beam pipe and a hadronic final state with large transverse energy, E_T . These events are simple to trigger on using the calorimeter only. However, without tracking

Fig. 25. 1996 neutral current event candidates in the (r, ϕ) view.

requirements the sample suffers from large proton beam gas background contaminations. Therefore the following GSLT selection criteria are applied: at least one track has to be found; a CTD-SLT vertex has to be found and it has to be reconstructed in the ± 90 cm range; the total $E - P_z$ at least 8 GeV; the transverse calorimeter energy excluding the first ring of cells around the FCAL beam hole has to be larger than 8 GeV; for events with $E - P_z$ larger than 12 GeV the energy deposits are not allowed to be concentrated in the FCAL, ie the ratio of P_z/E has to be less than 95%. This cuts is particularly powerful against beam gas contamination.

Heavy Flavour (Inelastic $J/\Psi \rightarrow e^+e^-$). Inelastic J/Ψ events in photoproduction, characterised by a low $E - P_z$ and a large track multiplicity with some tracks of high transverse momentum, are selected by: the requirement $P_z/E < 92\%$ is again imposed in order to reject beam gas background; $E - P_z > 4$ GeV is required for rejection of non- ep background; the sum of rear and barrel calorimeter EM energy has to be larger than 3 GeV. The e^+e^- decay mode of the J/Ψ ensures that the desired events pass this cut; since inelastic events have large track multiplicities the number of CTD-SLT tracks has to be larger than 2; at least one of these tracks has to have a transverse momentum of more than 1 GeV (typical for charm decays); at least two of these tracks have to have transverse momentum of more than 650 MeV.

High E_t Exotic Events This general filter selects all high E_t events while rejecting beam gas background using the following cuts: the calorimeter

transverse energy excluding cells in the first two FCAL rings has to be larger than 25 GeV; the calorimeter $E - P_z$ has to be larger than 15 GeV in order to reject non- ep background and some photoproduction events; the CTD-SLT beam gas flag must not be set. This flag indicates if there are many shallow tracks originating from vertices outside the CTD.

These physics filters demonstrate the event selection as well as the background rejection potential of tracking information on the SLT. Over the last three years most of the triggers using CTD-SLT information remained as in 1995. Only minor changes in cut values were made or additional requirements, mainly on the beam gas flag, were introduced.

8 HERA Upgrade

8.1 Introduction

In the year 2000 the HERA collider will be upgraded [21] to provide a significant increase in luminosity and to introduce polarised $e^\pm$ for the two general purpose ep scattering experiments. While most components of the ZEUS detector will remain unchanged, it has been decided to improve tracking by the introduction of a silicon micro-strip vertex detector (MVD) [22] and an enhancement of the forward tracking system by the addition of straw tube tracker (STT) [23]. To take full advantage of the increase in luminosity and the new components the second level track trigger needs to be widened in scope to become a global track trigger (GTT). In this section we present preliminary studies towards this goal: first a study of the improved resolutions to be gained by extending the present CTD-SLT algorithm to include MVD hits and second some comments on a single processor track SLT.

8.2 ZEUS Micro Vertex Detector

Apart from a considerable improvement in the measurement of charged particle tracks in general, the physics case for the ZEUS MVD has focused on heavy flavour identification and the search for new phenomena. The specifications and an $(r\phi)$ view of the MVD are given in Fig. 26.

The MVD consists of a barrel detector and forward wheels. The barrel detector is composed of ladders onto which the actual silicon detectors are mounted. The ladders are arranged in three radial layers with an asymmetric azimuthal distribution around the elliptical beampipe. The 4 wheels are orthogonal to

- angular coverage of $10^\circ - 170^\circ$
- three spatial measurements, in two projections each, per track
- $20\ \mu\text{m}$ intrinsic hit resolution
- impact parameter resolution of order $100\ \mu\text{m}$ at 90°
- noise occupancy $< 10^{-3}$
- hit efficiency $> 97\%$
- alignment accuracy $20\ \mu\text{m}$
- two-track separation $200\ \mu\text{m}$

Fig. 26. Layout of the ZEUS micro vertex detector in the r, ϕ view.

the beam line and consist of wedge shaped silicon detectors. The majority of the support structures (ladders, wheels and the overall support tube) will be made from a carbon fibre composite. Further details about the MVD and the current status of the project can be found in [22].

Several scenarios for including MVD information in the ZEUS trigger have been studied. First a standalone MVD trigger has been investigated in some detail [24]. Unfortunately neither the p_T nor the impact parameter resolution nor the estimated processing time met the necessary requirements. One major problem was the strong sensitivity of the processing time and the number of spurious tracks to the level of noise occupancy.

A second scenario [25; 18], also studied in some detail as a first step towards the GTT, is a track trigger at the second level using a combination of the CTD and the MVD data. It uses essentially the existing CTD-SLT algorithm but adds hits from the CTD stereo layers and the MVD. Starting a global tracking SLT with the current CTD-SLT approach seems very promising since the position of the MVD hits can be predicted to $150-1000\ \mu\text{m}$ via track extrapolation from the CTD into the MVD depending on Θ , ϕ and the details of the algorithm. Consequently pattern recognition in the MVD would become very simple and the sensitivity to the noise occupancy in the detector would be largely reduced. The precision on ϕ and the (x, y) and certainly the (r, z) vertex of tracks and the event would be almost entirely given by the MVD hits. The CTD hits would dominate the pattern recognition and the p_T determination.

These studies are based on a standalone Monte Carlo simulation of the MVD in combination with the ZEUS detector simulation. The simulation operates on single particle tracks as specified by their vertex and the particle's 4-momentum. The generation of hits in the MVD is realised in a 2-step approach [26]. Assuming a helix trajectory and a homogeneous B-field of $1.43\ \text{T}$ parallel to the z -axis, the intersection points of the tracks with the MVD

Fig. 27. *Example of the hit pattern in the CTD and the MVD for a track of $p_T = 0.50$ GeV. The open boxes represent hits currently used by the CTD-SLT while the black dots are CTD-hits on the track that the CTD-SLT could use. The stars are the MVD hits and the filled grey circles are hits from the CTD z-by-timing system (only shown in the $r-z$ view).*

volumes are determined. For this purpose, the MVD volumes are described as geometrical surfaces with an associated thickness (given in units of X_0) and material composition which were determined from the preliminary design parameters. For each intersection point, the material thickness traversed is calculated, and random multiple scattering angles are generated according to a distribution taking into account the non-Gaussian tails of the scattering angle distribution [27]. Intersections with Si volumes are smeared according to the intrinsic Si resolution for analogue readout (assumed to be $\mathcal{O}(10 \mu\text{m})$ for perpendicular incidence, degrading to $> 30 \mu\text{m}$ for small angles of incidence) and recorded as hits. Only hits in the barrel section of the MVD have been used for this study. The materials for the different components were taken to be: beampipe Al; active silicon layers Si+C; base plates C; cooling tubes Al+O+C. An example of a single particle hit pattern in the CTD and MVD together with a ‘track fit’ is shown in Fig. 27.

Note that this MVD simulation is independent of the full simulation of the rest of the detector. In particular, the multiple scattering “history” of a particle is inconsistent between the MVD and CTD. In addition the central beampipe section will probably be made of an Al-Be alloy. Together these effects give somewhat larger residuals in track fits and worse single particle resolutions in

the simulation than to be expected in reality.

From several different scenarios studied the following two are compared:

CTDSLT	The current version of the CTD-SLT. Up to 5 hits per track are used in the fast circle fit. The tracks in (r, ϕ) are forced to go through the origin $(0, 0)$. All z-information originates from the z-by-timing system only.
GTSLT	A modified algorithm of the CTD-SLT, running on a single CPU, includes up to 8 hits per segment with an additional ghost hit at $(0, 0)$. For long tracks the CTD stereo hits are included in the (r, z) straight line fit and the MVD hits are picked up during an iterative extrapolation into the MVD.

Table 3

Summary of resolutions on ϕ , p_T and distance of closest approach (DCA) for the CTD-SLT and GTSLT tracking scenarios. The FWHM values are obtained from fits of positive half gaussians to the corresponding distributions.

scenario	CTDSLT	GTSLT
Δz_{vtx}	8.8 cm	< 1 cm
$\Delta \phi$	1.32°	0.16°
$\Delta p_T/p_T^2$	5%	~ 1%
RMS of DCA_3	—	300 μm
FWHM of DCA_3 (fit)	—	216 μm
RMS of DCA_2	—	157 μm
FWHM of DCA_2 (fit)	—	91 μm
RMS of DCA_1	—	203 μm
FWHM of DCA_1 (fit)	—	94 μm
RMS of DCA_{vtx}	—	195 μm
FWHM of DCA_{vtx} (fit)	—	121 μm

The z-vertex resolution is improved by approximately one order of magnitude just from the CTD stereo data. The precise MVD z-information reduces this further to well below 1 cm depending on the topology and the details of the algorithm. The improvement in the ϕ resolution is mainly attributed to the 8 instead of 2 CTD hits per segment. For long tracks the increase in CTD hits also improves the p_T resolution to ~ 1%. For shorter tracks such precision can only be achieved using the MVD hits. Imposing the (x, y) -vertex constraint on the tracks improves the resolution and stabilises the pattern recognition part of the track SLT. The final fit would presumably be done without such a constraint as the heavy flavour tagging depends crucially on an unbiased reconstruction of the track distance of closest approach (DCA) to the vertex. In summary, the study indicates that the CTD-SLT algorithm can provide a track reconstruction with much higher precision than done at present. The

limitations are not intrinsic to the algorithm but rather a consequence of limited CPU power.

8.3 A Single-CPU track trigger.

The present CTD-SLT algorithm runs on a network of 130 transputers, performing parallel processing. This distribution of tasks onto many CPUs clearly has many advantages, resulting in 25 – 30 MHz CPUs being able to cope with track reconstruction at rates of up to 800 Hz. Nevertheless there are also disadvantages in such an architecture, for example the number of data packing, unpacking and transfer steps involved. As shown in Sec. 6.4 they result in the CTD-SLT latency distribution peaking at approximately 16 msec.

In order to gain some idea what the timing performance of the CTD-SLT or even a global track trigger would be on one single CPU the current CTD-SLT code was converted into C. The C code incorporates the general steps of the CTD-SLT algorithm. Only details such as the cut-offs on the number of segments found per cell or crate and the number of tracks found per crate etc. are not yet included. The C code was run on a 150 MHz SGI machine at the DESY computer centre with 1996 data as input. The resulting latency distribution is shown in Fig. 28 for neutral current DIS events.

The main features of this result are the latency tails of the distributions which reach up to ~ 25 msec and the mean values of 2.3 msec of the monotonic falling latency distribution. The maximum latencies are expected to be shorter after the data volume cut-offs are implemented, but the shape of the distributions and the drastically reduced mean latencies are very promising for a one-CPU tracking trigger and indicate the potential of a global tracking trigger. Given that ‘off the shelf’ CPUs rated at 500 MHz are already available and that processing speed will improve even more by the time the MVD starts data taking, this option appears both realistic and affordable. It is quite possible that a global track trigger could perform track reconstruction in shorter times than the current CTD-SLT with resulting dead time significantly reduced and the overall performance of the ZEUS DAQ improved.

To prepare for the GTT an extra 16 transputers were added to the CTD-SLT in 1998 (one for each sector) allowing CTD data sent to the SLT to be sent simultaneously to the GTT. Links from the 16 new transputers are connected to a VME crate from where they are sent to a PC farm using a Fast-Ethernet connection. The system has been operational since April 1999 and is being used initially for rate tests.

Fig. 28. Latency distribution for neutral current DIS events running on the one-CPU version of the CTD-SLT in C code on 1996 data.

9 Summary

In this paper we have described the design and performance of the second level trigger for the ZEUS central tracking detector. We have shown that a parallel processing software trigger provides a versatile solution, well matched to the demands of the high rate environment of the HERA ep collider. The trigger runs on a network of 130 INMOS T800 transputers and has achieved event rates approaching 800 Hz, not far from the design goal of 1 kHz. Parallelism is introduced both geometrically by splitting the data from the CTD into 16 independent sectors and functionally by splitting the algorithm into 4 major steps that run asynchronously. The trigger has been designed to achieve high track finding efficiency for high momentum charged particles and Monte Carlo studies have shown this to be nearly 100% for single tracks with $p_T > 400$ MeV/c and polar angles in the range $17^\circ < \theta < 162^\circ$. The resolution on p_T for such tracks is $\sigma_{p_T} = 0.05 p_T^2$. For events with low charged particle multiplicity the overall track finding efficiency is very high while in high multiplicity events the overall efficiency drops to around 50%, largely due to limits on the total volume of hit data that can be processed. Using a binning algorithm the resolution of the event vertex in z (coordinate along the beam line) is about 9 cm.

Apart from devising an efficient and fast algorithm, the main operational challenge has been to understand and control the latency of the CTD-SLT. This is a far from trivial problem in a tightly coupled parallel system which must handle large variations in both input data volumes and rates. A satisfactory solution has been obtained by very detailed studies of the data flows through the network and optimisation of each step in the algorithm for maximum speed. The development of online monitoring tools, offline simulations and a 'toy model' for the latency were vital in achieving this end.

The CTD-SLT is used for almost all physics event selections in the ZEUS second level trigger, the most important discriminants being the z_{vertex} position and the presence of 'good quality' tracks on the vertex. Finally studies have shown that the algorithm is robust and could be expanded to provide a global track trigger using data from the CTD and a proposed ZEUS microvertex detector. By the year 2000-2001 it is most likely that the extended algorithm could run on a single CPU, such is the advance in microprocessor technology since the CTD-SLT was designed 10 years ago.

Acknowledgements

We thank others involved in the development of the CTD-SLT, particularly John Lane for a first version of the segment finder, Sarah Pearson for early work on the tracking algorithm, Jon Butterworth for a first version of the offline simulation and Ian McArthur and Phillip Hallam-Baker for developing the prototype transputer system in Oxford. We thank the ZEUS trigger and DAQ experts, particularly Chris Youngman, Frank Cheblana, Leo Wiggers, Wesley Smith, Katsuo Tokushuku and Roberto Carlin for helpful criticism, advice and general support over the years. We also thank Chris Youngman and Uli Katz for help with the studies towards a future global track trigger. Arnulf Quadt gratefully acknowledges support by the Studienstiftung des deutschen Volkes and Christ Church College Oxford.

References

- [1] ZEUS Collaboration, Status Report 1993, ZEUS 93.
- [2] C. Youngman, 'The ZEUS Data Acquisition System', Proc. CHEP92 Annecy, Eds C. Verkerk & W. Wojcik, CERN (1992) 145.
- [3] W.H. Smith, K. Tokushuku & L. Wiggers, 'The ZEUS Trigger System', *ibid* 222; R. Carlin et al., 'The trigger of ZEUS, a flexible system for a high bunch crossing rate collider', Proc. 6th Conf. Instr. for e^+e^- Colliders, Novosibirsk 1996.

- [4] R. Carlin et al, 'Experience with the ZEUS Trigger System', Nucl. Phys. B (Proc. suppl.) **44** 430 (1995).
- [5] H.A.J.R. Uijterwaal, 'The ZEUS Global Second Level Trigger', Ph.D. Thesis, University of Amsterdam 1992.
- [6] D. Bailey et al., 'The ZEUS Third Level Trigger', Proc. CHEP92 Annecy, Eds C. Verkerk & W. Wojcik, CERN (1992) 171;
S. Polenz, 'A FDDI based processor farm for the ZEUS third level trigger', CHEP97, Berlin, to be published in the proceedings.
- [7] B. Foster et al., *Nucl. Instrum. Methods A* **338**, 254 (1994).
- [8] G.F. Hartner et al., 'VCTRAK (3.07): Offline Output Information', ZEUS-Note 97-064 (unpublished).
- [9] R. Hall-Wilton et al., 'The CTD Tracking Resolution', ZEUS-Note 99-027 (unpublished).
- [10] R.C.E. Devenish et al., Proc. CHEP90 Santa Fe, eds J. Lillberg & M. Othoudt, AIP Conf. Proc. 209 (1990) 155.
- [11] D.G. Cussans et al., *Nucl. Instrum. Methods A* **315**, 397 (1992).
- [12] D.S. Bailey et al., *Nucl. Instrum. Methods A* **396**, 320 (1997).
- [13] INMOS Limited; The Transputer Databook, 2nd edition: 1989.
- [14] INMOS Limited; OCCAM 2 Reference Manual. Prentice Hall 1988, ISBN 0-13-629312-3.
- [15] J. Zoll, ZEBRA - Reference Manual - FZ Sequential I/O, Program Library Q100, CERN 1991.
- [16] J. Butterworth et al., 'Simulation of the CTD-SLT', ZEUS-Note 91-072 (unpublished); D.Phil. thesis, University of Oxford 1992.
- [17] R.C.E. Devenish et al., 'Geometry and resolution of the CTD-SLT', ZEUS-Note 90-078 & erratum 90-078c (unpublished).
- [18] A. Quadt, MSc Thesis, University of Oxford 1997, OUNP-97-13.
- [19] H.A.J.R. Uijterwaal et al., Proc. CHEP95 Rio de Janeiro, eds R. Shellard & T.D. Nguyen, World Sci. (1996) 454.
- [20] A. Quadt et al., 'Simulation of parallel processing systems, the CTD, a case study', CHEP97 Berlin, to be published in the proceedings.
- [21] 'HERA Luminosity Upgrade', Report for Funding Agencies, ZEUS Collaboration, October 1997;
'ZEUS Interaction Region Upgrade', V. Noyes et al., ZEUS-Note 97-035
- [22] A microvertex detector for ZEUS, ZEUS-Note 97-006 (unpublished); ZEUS MVD web site <http://www-zeus.desy.de/~zeusmvd/MVD.html>.
- [23] A Straw Tube Tracker for ZEUS, ZEUS-Note 98-046 (unpublished).
- [24] S. Sampson, 'Charm tagging at the SLT with the MVD', ZEUS-Note 99-077 (unpublished).
- [25] A. Quadt, 'Running modes of the CTD-SLT and their impact on a global tracking SLT', ZEUS-Note 99-067 (unpublished).
- [26] U. Katz, Private communication; O. Strangfeld, Diplom Arbeit, University of Bonn 1999, BONN-IB-99-06.
- [27] J.R. Campbell, TRIUMPH preprint TRI-PP-95-34 (unpublished, 1995).