Data acquisition software for the CMS strip tracker

R Bainbridge[1], G Baulieu[2], S Bel[3], J Cole[4], N Cripps[1], C Delaere[3], A C Assis Jesus[5], F Drouhin[3,6], J Fulcher[1], A Giassi[7], K Gill[3] D Giordano[8], L Gross[9], K Hahn[10], S Mersi[11], L Mirabito[3], M Nikolic[12], V Radicci[1] S Tkaczyk[13] and M Wingham[1]

[1] Blackett Laboratory, Imperial College London, SW7 2BW, UK
[2] Institut de Physique Nucleaire de Lyon, France
[3] CERN, Geneva 1201, Switzerland
[4] Rutherford Appleton Laboratory, Didcot, UK
[5] Universidade do Estado do Rio de Janeiro, Brazil
[6] Universite de Haute Alsace, Mulhouse, France
[7] INFN, Sezione di Pisa, Italy
[8] INFN, Università di Bari, Italy
[9] Institut Pluridisciplinaire Hubert Curien, Strasbourg, France
[10] Massachusetts Institute of Technology, USA
[11] INFN, Università di Firenze, Italy
[12] University of California, Santa Barbara, USA
[13] Fermi National Accelerator Laboratory, Batavia, Illinois, USA

E-mail: Robert.Bainbridge@cern.ch, Laurent.Mirabito@cern.ch

Abstract.
The CMS silicon strip tracker, providing a sensitive area of approximately 200 m^2 and comprising 10 million readout channels, has recently been completed at the tracker integration facility at CERN. The strip tracker community is currently working to develop and integrate the online and offline software frameworks, known as XDAQ and CMSSW respectively, for the purposes of data acquisition and detector commissioning and monitoring. Recent developments have seen the integration of many new services and tools within the online data acquisition system, such as event building, online distributed analysis, an online monitoring framework, and data storage management. We review the various software components that comprise the strip tracker data acquisition system, the software architectures used for stand-alone and global data-taking modes. Our experiences in commissioning and operating one of the largest ever silicon micro-strip tracking systems are also reviewed.

1. Introduction

The CMS silicon strip tracker is unprecedented in terms of its size and complexity, providing a sensitive area of approximately 200 m^2 and comprising 10^7 readout channels. The strip tracker control and readout systems, along with components of the CMS event builder, all of which were extensively tested at the Tracker Integration Facility (TIF), are described below.

1.1. The strip tracker data acquisition systems

Figure 1 shows a schematic of the control and readout systems. The control system [1] comprises 350 control rings that start and end at the off-detector Front-End Controller (FEC) boards and is responsible for distributing a 40 MHz clock, Level-1 triggers and control commands to the front-end electronics. The signals are transmitted optically from the FECs to front-end digital opto-hybrids via digital links, and then electrically via token rings of Communication and Control Units (CCUs) to the front-end electronics.

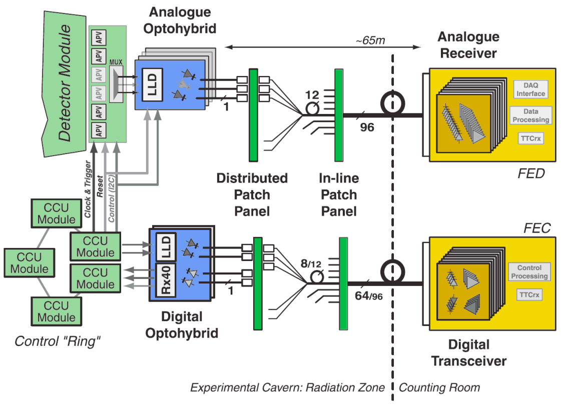

Figure 1. Strip tracker control and readout.

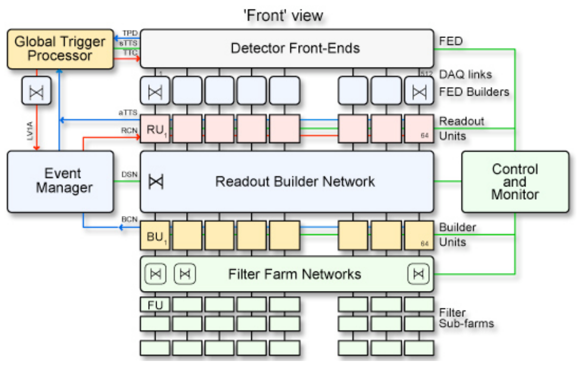

Figure 2. The CMS event builder.

The readout system is built around a custom front-end ASIC known as the APV25 chip [2], an analogue optical link system [3] and an off-detector Front-End Driver (FED) digitizing and processing board [4]. The system comprises 76000 APV25 chips, 38000 optical fibres (each transmitting data from a pair of APV25 chips) and 440 FEDs. The APV25 chip samples, amplifies, buffers and processes signals from 128 channels of a silicon strip sensor at the LHC collision frequency of 40 MHz. On receipt of a Level-1 trigger, pulse height and bunch-crossing information from pairs of APV25 chips are multiplexed onto a single line and converted to optical signals, which are transmitted via analogue fibres to the off-detector FED boards housed in VME crates. The FEDs digitize and sparsify the pulse height data (*i.e*, "zero-suppress" by comparison with thresholds) from up to 96 pairs of APV25 chips and format the data to construct a data packet. When using the global trigger and DAQ systems, the FEDs forward their data packets to the CMS event builder (EVB) and online computing farm via an S-Link connection [5]. Stand-alone operation of the strip tracker involves data being read out via the VME backplane to local computing resources.

The CMS trigger system [6] (not shown in Figure 1) is based around the Global Trigger Controller (GTC) and other ancilliary cards that receive and process trigger information from the calorimetry and muon systems and issue Level-1 trigger decisions to all sub-detectors (according to trigger rules that minimize deadtime in the various readout systems). All trigger and clock signals are distributed from the GTC to the various sub-detector control and readout components via the Timing, Trigger and Control (TTC) system [7]. Additionally, the Local Trigger Controller (LTC) can be used to operate a sub-detector individually in a stand-alone data-taking mode for debugging and commissioning purposes. The LTC can operate up to four *partitions* independently, each comprising a collection of hardware components that share the same trigger source. This granularity fits naturally with the partitioning of the strip tracker sub-detector, which is divided into inner and outer barrel regions and two end-cap regions. Global running of the CMS experiment comprises all sub-detector partitions (up to 32) using the GTC as the primary trigger source.

1.2. The CMS event builder

The event data enter the event builder (EVB) [8] as a set of data packets from ∼700 FEDs within CMS (440 of which are used by the strip tracker readout system). The task of the EVB, shown schematically in Figure 2, is to collect all these packets, assemble them into a complete event, and then forward the event to a processing node on the computing farm. The average event size for CMS is expected to be ∼1 MB, meaning that, for a L1 trigger rate of 100 kHz, the event builder network must be able to handle an aggregate throughput of 100 GB/s. This requires a highly parallelized system.

The event builder uses two stages of switch networks. The first stage comprises 64 FED builders, each performing a first pass of data concentration by taking data from up to 8 source cards (typically receiving data from pairs of FEDs) and building *super fragments* (from the FED data packets) using firmware via a small 8x8 Myrinet switch network connected to *Readout Units* (RU). In the second stage, complete events are built using the Readout Builder Network, which comprises 64 *Builder Units* (BU) receiving super fragments (of the same event) from the 64 RUs via a large 64x64 switch network via Gigabit ethernet. The BUs buffer and forward complete events to 2048 *Filter Unit* (FU) processes that are hosted on the computing farm. The FU processes perform online event filtering and monitoring.

1.3. The tracker integration facility

Construction and assembly of the strip tracker sub-detector have recently finished at the TIF at CERN, which provided an environmentally controlled clean-room with sufficient services and infrastructure to cable, power and operate as much as 1.6 million readout channels (approximately one sixth of the complete system) at any given time. This constitutes one of the largest silicon micro-strip tracking systems ever operated. The facility also provided a large cooling capacity, sufficient to perform cold tests down to -15°C. The sub-detector was assembled, commissioned and operated over a period of several months during 2007. Large samples of cosmic ray data were accumulated and analysis of the data has been performed to validate the detector performance and provide preliminary alignment parameters prior to the detector commissioning phase of CMS (without beam). The TIF has allowed the tracker community to maximize its experience in operating the detector and its service infrastructure, which is of vital importance, as expedient checkout and commissioning of the sub-detector at Point 5 is necessary to minimize any delays during the start-up phase of the experiment.

2. Software architecture and components

The strip tracker data acquisition (DAQ) software performs two primary functions. The first is to configure, control and monitor the readout system during global running, when the central trigger and DAQ systems described in Section 1 are used. The second is to provide commissioning procedures that configure, calibrate and synchronize the readout system, as described in Section 4. The majority of these procedures are performed in stand-alone mode using local computing resources, as beam is not needed. Some procedures, however, do require beam and also a Level-1 trigger, hence global running using the central DAQ services is necessary. This requirement affects the software design in order to be used transparently in both global and stand-alone modes. The scale of the strip tracker sub-detector has a significant impact on the software architecture, as a highly distributed system is required to handle the huge data volumes generated by the sub-detector. The input data rate to the global computing farm from the strip tracker is about 0.8 MB/event at a trigger rate of 100 kHz during nominal operation. Significant progress has been made recently in the development of the software infrastructure and tools for the DAQ system that satisfy the above requirements, which are reviewed below.

2.1. Software frameworks

Many services required by the DAQ software, such as distributed event building and online monitoring, are provided by the two software frameworks developed for and used throughout the CMS experiment, called XDAQ and CMSSW.

XDAQ [9] is the online software framework for the CMS DAQ systems and provides a core set of services and tools, including: a fast communication protocol for peer-to-peer messaging between processes registered with the framework; a slower communication protocol for configuration of the framework processes; a finite-state machine schema; event builder applications; and memory management tools. The DAQ software is based on XDAQ applications that control the trigger system, hardware configuration, event building and data analysis. The simple communication tools provided by the XDAQ framework allow to automate run sequences and acquisition loops. This feature is used by the strip tracker DAQ software to define automated detector commissioning procedures.

CMSSW is the offline software framework providing event reconstruction, physics analysis, High-Level Trigger (HLT) algorithms, calibration and monitoring, and is based around the CMS event data model (EDM) [10]. All non-event data are stored in the CMS conditions database, which is accessible within the framework via the *event setup* interface [11].

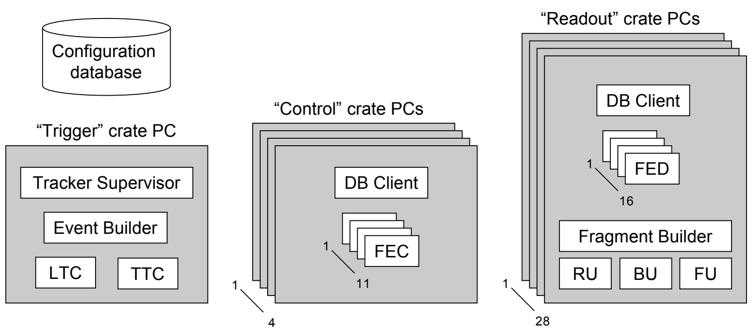

Figure 3. Schematic of the strip tracker data acquisition software, showing the various XDAQ applications (white boxes) that are hosted on the PC controllers of VME crates (grey boxes).

2.2. Software architecture

When operating the strip tracker in stand-alone mode, all DAQ processes are hosted by local computing resources, namely the PC controllers of the VME crates that house the off-detector electronics (FEDs, FECs, trigger cards, etc), as shown in Figure 3. The raw data from the FEDs are accessed via the VME backplane and *super fragments* are built (at the level of a VME crate) from the FED data buffers using a tracker-specific *fragment builder* application. Standard tools developed within the XDAQ framework are then used to build complete events from these fragments. Event building is distributed across several crate PCs (using Gigabit ethernet) in order to allow parallel event processing. The software used to handle the readout system is replicated at the level of a "readout" crate PC and comprises the following XDAQ applications:

- one *FED supervisor* per FED (up to 16 per VME crate) that controls and configures the FEDs and handles data capture, processing and readout;
- one *fragment builder* per VME crate, which receives data from up to 16 FED supervisors of a crate and builds a super fragment;
- each crate controller PC also hosts RU, BU and FU applications, identical to those described in Section 1.2, which are used to build complete events from super fragments. The FU is configured to perform online analysis and monitoring of the event data.

Additionally, each "control" VME crate that houses FEC cards (typically 11 per crate) has a PC hosting *FEC supervisors* (one per FEC) that are used to configure and control all front-end devices via the FECs. All readout and control crate PCs also host one *database client* application, which provides access to the configuration database. Additional applications hosted on the "trigger" crate PC include: one *LTC supervisor* and *TTC supervisor* to define and control trigger patterns when operating in stand-alone mode; one *event builder manager*, which handles trigger tokens from the LTC supervisor and controls the event building; and one top-level *tracker supervisor* application that communicates with all supervisor processes and steers the DAQ loops and commissioning procedures.

Online event processing is performed within the CMSSW software framework using the Filter Unit application. When running in stand-alone mode, the CMSSW framework is typically configured to use software modules that unpack the FED buffer data and perform calibrations and monitoring. Although these modules exist within the offline software framework, they can be used in an online context using the architecture described above to provide real-time feedback on the detector operation and performance.

When the strip tracker sub-detector is operated in conjunction with the whole of CMS (using the global trigger and DAQ systems), the software architecture remains largely unchanged, although in this case data from the FEDs are routed via their S-Link connections to the global online computing farm at the Level-1 trigger rate of 100 kHz. In this global mode of operation the trigger-related supervisors become redundant and are used only for monitoring purposes. All other supervisor applications continue to be used to configure, control and monitor the hardware components, whilst the event building tools are used to handle *spy channel* data from the FEDs, which are again read out via the VME backplane and processed using the local computing resources. This spy mode is expected to provide event data at the Hz level and will be used for monitoring purposes [12].

2.3. Motivation for chosen software architecture

There are many advantages to developing within the CMSSW framework. The framework, when used in an online context with the event builder, allows distributed data analysis, which is an essential feature when considering the huge data volumes generated by the strip tracker detector. In addition, many useful services are available within the offline framework, such as access to the CMS geometry and conditions databases and the Data Quality Monitoring framework [13], which provides distributed monitoring and is heavily used by the commissioning procedures.

Importantly, the chosen design is also sufficiently flexible to allow transparent use of both stand-alone and global operating modes, due to the (essentially) identical software architecture and tools. This is important for detector commissioning, as the global trigger and DAQ are required for some procedures and the achievable trigger rates and available computing resources are far superior to those available when operating in stand-alone mode.

3. Software components

Recent developments to major components of the DAQ software are reviewed below.

3.1. Configuration database

The online *configuration database* provides the database management system for the strip tracker sub-detector and is closely integrated with all components of the strip tracker DAQ software. It stores all parameters that are downloaded to hardware registers within the (local) trigger, control and readout systems that are required to bring the sub-detector into an operational state suitable for physics data-taking. All calibration constants concerning the hardware components of the readout system are also stored in this database.

The offline *conditions database* stores all information needed for event reconstruction, except the event data itself. A large sub-set of configuration and calibration data stored in the configuration database are also considered to be conditions data used by reconstruction. Examples include the calibration constants for various detector and hardware elements, such as gain and noise. This necessitates an Online-to-Offline (O2O) transfer tool that copies data from the configuration database to the conditions database. The conditions database is fully integrated with the CMSSW software framework, which guarantees synchronicity between the conditions data and the event data using a mechanism that defines an *Interval Of Validity* (IOV) range of time for all conditions data with respect to the event data.

The configuration database uses a relational model and is based on Oracle^{TM} technology. The model is a representation of the configurable hardware components within the control and readout systems. The database is queried using embedded procedures that are wrapped by a C++ API, which handles the configuration data for the end-user. The model used is defined as follows: configuration data are assigned to a *partition*, which is defined to be a set of devices (within either the control or readout system) that share a common trigger source; all parameters within a partition are *versioned* in order to keep a complete history of all changes to parameter values; a *state* is used to correlate all parameter versions within a partition; a *run* is defined to be a period of time for which a given state was used. All of the above information is stored in a dedicated *run table*, which is used by the O2O transfer tool in order to define IOV ranges used by the conditions database. In this way, consistency between the configuration and conditions databases, and the conditions data and event data, is guaranteed.

The parameter set that fully defines all register settings of the front-end chips within the control system comprises almost 2 million values. The parameter set for the readout system is much larger, due to the calibration constants required by the FED at the level of a detector strip (such as pedestals and noise). Each FED requires 40,000 parameters, resulting in a compressed data volume of 150 kB. The total data volume describing the complete 440 FED system comes to 65 MB per version. The total amount of data expected during one year of data taking is estimated to be as much as 50 GB.

A performance target was never formally specified for the configuration database, but the underlaying assumption is that the retrieval of configurations from the database and download to the hardware devices should not contribute significantly to the initialization period of the hardware and software systems at the start of a run. The configuration database has been used extensively during the tracker integration period of the last eight months. Performance problems were highlighted during its use, which largely concerned the amount of data being stored and the heavy use of the versioning system. Upload and download times are presently around a few minutes for the entire tracker system, and are expected to decrease significantly to around a minute in the near future, with developments focusing on further optimizing the versioning system and the development of several caching mechanisms.

3.2. Run control and configuration
The Run Control and Monitoring System (RCMS) has been developed to configure, control and monitor the DAQ systems of CMS during data-taking. The tool is based on web applications and interfaces implemented using Java Servlet technology, AJAX and JSP [14].

From its web interface, users can select and load a specific DAQ configuration and then trigger transitions between various states (*e.g,* from "Halted" to "Configured" to "Running"). RCMS provides the framework in which system-specific *function managers* (FM) can be defined. The FM is a top-level finite state machine that controls and steers all the finite state machines built into the low-level XDAQ applications. Selecting a state transition (*e.g,* "Configure") within the RCMS web interface triggers the FM to send the corresponding command to all the XDAQ applications under its control and checks that they reach the desired state. In addition, the RCMS steers the management of the XDAQ processes and applications by interacting with the *Job Control* service, which creates, monitors and destroys these processes on request. RCMS was used throughout the detector integration period at the TIF. Various function managers were developed to optimize the strip tracker configuration performances.

3.3. Data storage and monitoring
When operating the strip tracker in stand-alone mode using local computing resources, event building is performed as described in Section 2.2. The event data stream (including any reconstructed objects) is forwarded to the CMS Storage Manager application, which is one of the last components in the data handling chain [15]. This application performs two primary functions. The first is to provide the mechanism for writing the event data that is received from the event builder applications to disk. The second is to act as an event and histogram server for calibration and monitoring purposes. The Storage Manager is required to handle high bandwidths (up to 100 MB/s throughput), which is achieved by using an optimized, intermediate data format that is later converted to the CMS EDM format at the Tier-0 centre during the first-pass event reconstruction.

The Storage Manager was used by the strip tracker DAQ system for the first time during the tracker integration period at the TIF to stream data to local storage. The data were converted from the intermediatory format into the EDM format and reconstruction performed. The data were then archived using a central data storage service, registered with the standard CMS Data Bookeeping System and transferred to remote tier centres for analysis [16].

The Storage Manager was also configured to act as an event server, providing a real-time data stream for monitoring purposes, both at CERN and remotely at the Fermilab Remote Operations Centre [17]. Recent developments also include software for monitoring the hardware systems via scalars encoded within the raw event data stream and the monitoring of configuration and runtime errors from the hardware systems using the Error Diagnostic System [12].

3.4. Error diagnostic system
The strip tracker is a hugely complex detector with many potential sources of error conditions. The need for an Error Diagnostic System (EDS) arose very early in the development cycle of the strip tracker DAQ software, when it became clear that the scope for error conditions with such a complex system frequently leads to a point where it is impossible to identify quickly and easily the underlying problem. The EDS is the strip tracker solution to handling both configuration and runtime error conditions. The tool provides immediate diagnostic information to the end-user. The tool is eventually expected to provide failure analysis and automated recovery procedures.

A dedicated CMS *errors database* has been developed in order to record all errors, hardware failures and abnormal detector states that could occur during data-taking. This information is also considered to be conditions data required by the offline reconstruction and a condensed set of parameters will be propagated to the conditions database using the O2O transfer tool.

The EDS has been developed within the XDAQ framework and it fully integrated with all components of the strip tracker DAQ software. The central component is the *error dispatcher*, which collects and processes error messages from XDAQ applications (which typically communicate with hardware components). Each error dispatcher has various communication possibilities available, including:

- with other error dispatchers (to allow flexibility and scalability);
- with monitoring frameworks or commication protocols;
- file I/O, the dedicated errors database and logging applications (such as a custom application called LogReader that is distributed with the EDS software).

Communication between error dispatchers is arranged according to a hierarchical tree structure, which typically mimics the logical structure of the hardware systems. In this way, pre-processing of error conditions is possible at all levels within the system. *Action scenarios* can be embedded within the error dispatcher that are triggered on receipt of a known error condition, a simple example being the reconfiguration of a device following problematic initialization. Repeated failures will result in error conditions being propagated to higher level dispatchers, as the underlying problem may span across multiple device types or groups.

The EDS was used during the tracker integration period at the TIF. The experience with a large-scale system highlighted some performance issues, in particular concerning the display of error messages by the logger applications. This bottleneck effectively throttled the entire system, so the two aspects of error handling and visualization were decoupled by streaming all messages to an intermediate file, resulting in improved performance. The large-scale tests performed at the TIF, using as much as 15% of the final system, demonstrated that the error diagnostic system framework is now stable and well integrated with the strip tracker DAQ system.

Future goals involve developing the failure analysis and automatic recovery procedures, which relies on experience of running the detector.

4. Detector commissioning procedures

Commissioning procedures are required to bring the detector into an operational state suitable for physics data-taking [18]. These procedures will be heavily used during the integration and start-up phases of the experiment to validate and optimize the operational functionality and performance of the sub-detector. During the operational phase, the procedures will be performed with varying frequencies between periods of data-taking (for example, during a beam fill) in order to guarantee optimal detector performance and calibration during the subsequent period of data-taking. Some commissioning procedures are highlighted below.

- The sub-detector partitions (up to four) are automatically determined using hardware scans that detect (and check) all front-end and off-detector devices that share a common trigger source. Additionally, the connectivity maps between all front-end devices and power supply units and off-detector FED boards are automatically established. These essential tools will be heavily used during the check-out and commissioning phases.
- The biasing of the 15000 strip sensors will be tuned using an automated procedure that will perform high voltage scans in order to determine the sensor depletion voltages. The configurations of all front-end APV25 readout chips are individually tuned in order to optimize signal amplification and shaping (important to achieve optimum signal-to-noise ratios and for bunch crossing identification) as well as various bias and gain settings. Several thousands of laser driver ASICs within the optical link system are also individually tuned and calibrated to achieve optimum dynamic range usage and gain matching across the entire link system. Digitization and sparsification of the analogue data, performed by the off-detector FED boards, are also tuned, which require calibration constants for each detector strip. A sub-set of these constants are also used by the track reconstruction software during offline analysis of the event data.
- The sub-detector is internally synchronized to guarantee that all front-end modules receive in-phase clock and trigger signals via the control system [19]. The timing of the complete strip tracker system is then globally adjusted to be synchronous with LHC collisions (or cosmic muon triggers, as during early detector commissioning) and other sub-detectors, using a track-based analysis. This procedure requires high precision, as the detector signal is attenuated by 4% per nanosecond timing misalignment for the nominal operating modes.

The DAQ software for the strip tracker provides the implementation for the various commissioning procedures, which comprise dedicated DAQ "loops" that determine optimized configuration parameters and calibration constants from reconstructed calibration pulses, timing delay curves, dynamic range curves and other features constructed from the APV25 data stream Each DAQ loop is defined by a run comprising bursts of triggers separated by periods when the trigger is inhibited. For each cycle, buffered FED data packets are read out and the configuration of at least one device is changed via the control system. The configuration of several devices of

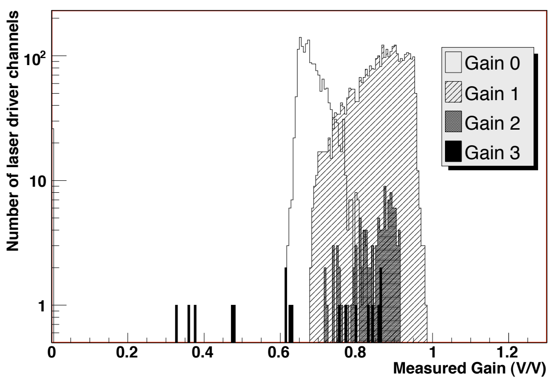

Figure 4. Measured gain [V_{out}/V_{in}] for ~6000 optical links operated at -10°C. Each link was configured to use its optimal gain setting and its operating gain measured.

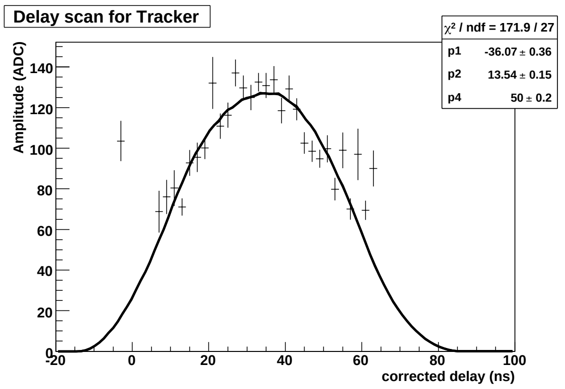

Figure 5. Signal amplitude of hits associated with cosmic ray tracks as a function of time. The curve was used to synchronize to the passage of cosmic rays through the tracker.

the same type are usually tuned in parallel. Trigger bursts provide data samples obtained using different hardware configurations and runs of several bursts are used to reconstruct the appropriate curves, which are then in turn used to extract the optimized configurations and/or calibrations constants for the hardware devices. These values are then uploaded to a configuration database and provide the configuration data for the subsequent commissioning procedure or physics run. The histogram-based analyses for by the commissioning procedures are implemented within the CMSSW framework and heavily use the Data Quality Monitoring framework. Any information required by the analysis during the DAQ loops (*e.g,* an identifier for the device being configured and its configuration) is encoded within the data stream itself. In this way, the analysis is entirely data-driven and no database access is required. The procedures described above were used to commission a system comprising 1.6 million individual readout channels. Some results from the commissioning procedures are highlighted below.

One procedure is designed to tune the Linear Laser Driver (LLD) component of the optical link system. The LLD is a custom ASIC that is located on the front-end module and performs electrical-to-optical signal conversion at the level of a pair of APV25 chips, which are read out by one FED input channel. Each LLD has four different gain settings (0, 1, 2 and 3) and is configured to operate using the setting that results in a measured gain closest to a value of 0.8 [V_{out}/V_{in}].[1] A system at the TIF comprising ~6000 links and cooled to the nominal operating temperature of -10°C was tuned in this way. The resulting distributions, one for each group of devices sharing the same gain setting, are shown in Figure 4. The distributions are in agreement with simulation [20].

Another procedure synchronizes the strip tracker sub-detector with LHC collisions, using a track-based analysis. Correcting the time alignment of the strip tracker includes compensating for the global trigger latency, timing delays in the strip tracker control system and the time-of-flight of particles. The objective is to be optimally synchronized with LHC collisions in order to maximize hit reconstruction efficiency, whilst minimizing the number of fake hits from adjacent bunch crossings. The scan in time is performed in two steps. First, a coarse scan is performed in steps of 25 ns, which allows the correct bunch crossing to be located. Then, a *fine delay scan* in steps of 1 ns is performed per detector layer. The aim is to achieve synchronization with a precision of better than 1 ns. The fine delay scan procedure was exercised at the TIF in order to

[1] V_{out}/V_{in} is used to parameterize the link gain. A value of 0.8 corresponds to digitized signal in the FED with a dynamic range of 3.2 minimum ionizing particles in 300 μm silicon.

synchronize to the passage of comsic rays through the detector (using scintillators and trigger logic specific to tests at the TIF). The measurement was performed using both the tracker inner and outer barrel partitions. Figure 5 plots the signal amplitude of hits as a function of time. The pulse shape maximum is located at 33.7 ns with a precision of 0.5 ns.

Once commissioned, the DAQ system was used to operate the strip tracker and take events triggered by cosmic rays passing through the sub-detector. The DAQ software proved to be stable during the data-taking period at the TIF and demonstrated good performances, with trigger rates around a few Hz (dependant on the scale of the system) and data rates of up to 10 MB/s (essentially limited by the maximum VME backplane throughput). An event sample and data volume of five million and 15 Terabytes, respectively, were archived to tape.

5. Conclusions

The scale of the CMS strip tracker sub-detector has a significant impact on the design of the data acquisition software, which must be highly distributed in order to handle the expected data volumes. Recent tests with large-scale systems at the tracker integration facility have demonstrated that the DAQ software is scalable and several online applications concerning event building, data storage, run control, databases, error diagnostics and online monitoring were extensively tested and optimized. Various commissioning procedures that are used to configure, calibrate and synchronize the readout system were also tested. Experiences have shown that a highly distributed architecture and automated procedures for detector commissioning are essential design features for data acquisition software of the present generation of detector systems, typically comprising tens of millions of readout channels.

References

[1] Drouhin F *et al* 2004 *CERN-CMS-CR-2004-032*
[2] French M *et al* 2001 *Nucl. Instr. Methods Phys. Res.* A **466/2** 359
[3] Troska J *et al* 2002 *Nucl. Sci. Symp.* **1** 233
[4] Coughlan J *et al* 2002 *CERN-LHCC-2002-034* 296
[5] Van der Bij H C *et al* 1997 *IEEE Trans. Nucl. Sci* **44/3** 398
[6] Valera J 2002 *CERN-CMS-NOTE-2002-033*
[7] Troska J *et al* 2006 *IEEE Trans. Nucl. Sci.* **53/3** 834
[8] Pieri M *et al* 2006 "CMS DAQ event builder based on gigabit ethernet", *Proceedings for Computing in High Energy and Nuclear Physics*, Mumbai, India
[9] Kozlovszky M *et al* 2004 *Nucl. Instr. Methods Phys. Res.* A **534** 125-129
[10] Jones C D *et al* 2006 "The new CMS event data model and framework", *Proceedings for Computing in High Energy and Nuclear Physics*, Mumbai, India
[11] Jones C D *et al* 2006 "Access to non-event data for CMS", *Proceedings for Computing in High Energy and Nuclear Physics*, Mumbai, India
[12] Mersi S *et al* 2007 "Real-time analysis of the operational state of the CMS strip tracker readout system", *these proceedings*
[13] Leonidopoulos C *et al* 2006 "Physics and data quality monitoring at CMS", *Proceedings for Computing in High Energy and Nuclear Physics*, Mumbai, India
[14] Oh A *et al* 2007 "The run control and monitoring system of the CMS experiment", *these proceedings*
[15] Sexton-Kennedy E *et al* 2007 "The CMS storage manager and data flow in the high level DAQ", *these proceedings*
[16] De Filippis N *et al* 2007 "Real-time dataflow and workflow with the CMS tracker data", *these proceedings*
[17] Giordano D *et al* 2007 "Data quality monitoring and visualization for the CMS silicon strip tracker", *these proceedings*
[18] Bainbridge R *et al* 2006 *CERN-LHCC-2007-006* 419
[19] Gill K *et al* 2003 *CERN-LHCC-2003-055* 289
[20] Dris S *et al* 2006 *CERN-CMS-NOTE-2006-145*