Neural Triggering System Operating on High Resolution Calorimetry Information

A. dos Anjos, R.C. Torres, J.M. Seixas, B.C. Ferreira, T.C. Xavier [a]

[a]Signal Processing Laboratory/COPPE/Federal University of Rio de Janeiro

This paper presents a electron/jet discriminator system for for operating at ATLAS/LVL2. In order to handle the high data dimensionality, the RoIs are organized in the form of concentric ring sums, so that both signal compression and improved performance can be achieved. The ring information is fed into a feed forward neural network. This implementation resulted on a 97% electrons detection efficiency for a false alarm of 3%. The full discrimination chain could still be executed in less than 500 μs.

1. INTRODUCTION

The ATLAS experiment at CERN [1] is looking for interesting new physics generated by LHC proton-proton interactions at very high luminosity ($10^{34} cm^2/s$). Its research program includes a search for the Higgs boson, super-symmetry and other new phenomena. The detector is composed of specialized sub-detectors to register the properties of the decaying particles: an inner detector inside a magnetic field of 2 T measuring trajectories, a Calorimeter to measure energy and finally a muon spectrometer.

The LHC will provide bunch crossings at a rate of 40 MHz. It is the job of a online triggering system [2], to separate trivial well-understood physics from the exciting new phenomena that ATLAS wants to study. This system is being built in three-levels. The First-Level Trigger (LVL1) is directly connected to the detector front-end electronics of the calorimeter and muon detectors. Fast algorithms and energy adders implemented in custom hardware are used for LVL1 event selection. This trigger level also defines Regions of Interest (RoIs) in the detector where interesting physics signatures were found. Event data of accepted events are sent out into the Data Acquisition system (DAQ) via read-out drives (RODs) and are made available to the High-Level Triggers (HLT), i.e. the Second Level Trigger (LVL2) and the Event Filter (EF), through ∼1,600 read-out buffers (ROBs). The LVL1 trigger has to cope with the high input bandwidth of the experiment (40 MHz), being design to have a maximum output rate of 75 kHz, upgradeable to 100 kHz.

The RoIs found by LVL1 are used as seeds for the Second-Level Trigger (LVL2) [3]. By only looking at data in LVL1 RoIs, it is possible to reduce the amount of data transferred into the LVL2 processors to less than 2% of the total event data (∼1.3 MB) and achieve further background rejection. LVL2 selection algorithms request data from variable numbers of RoIs, typically 1 or 2. A RoI spans on average 18 ROBs when located in the calorimeter section, but only a maximum of 3 ROBs if LVL1 triggered on muon candidates. If an event is accepted by LVL2, a detailed summary of the processing, the LVL2 Result, is appended to the event stream and used by the Event Filter to proceed with the analysis. The last trigger level is the Event Filter (EF). After a LVL2 accept, the full event data is assembled by special computing nodes and redirected to specialized processing farms, where more elaborate filtering and monitoring algorithms are used.

1.1. Requirements for the LVL2 Trigger

At LVL2, the total average processing time per event is expected to be ∼10 ms [2]. This number is devised by considering the LVL1 output rate, and a LVL2 processing farm with a capacity equivalent to 1,000 CPUs at 4 GHz. For this configuration, each node should deliver a trigger decision rate of ∼100 Hz, requiring an input bandwidth of 2.6 MB/s.

To achieve these requirements, it is expected that data discrimination algorithms reject as soon as possible uninteresting physics to leave most of the available processing budget for interesting events.

Both the LVL2 and EF use offline software components for doing event selection. A thin interface, the Steering Controller (SC) [4], binds the offline ATHENA/GAUDI [5] software environment to the HLT framework. Slightly different implementations of the SC are available for LVL2 and EF. Event selection happens in LVL2 in multiple, concurrent threads of execution, while the EF is process based.

In both cases, multiple algorithms are scheduled on a per-event basis by a common steering software. It manages the execution order of algorithms based on the seed received, i.e., in LVL2, it uses the RoI information provided by LVL1 result, while in EF, the LVL2 Result.

1.2. The e/γ Trigger

When high-p_T electron candidates are detected by LVL1, LVL2 starts its processing sequence trying to confirm the object taking into consideration the full detector granularity. The LVL2's calorimeter algorithm, called T2Calo, job consists of evaluating a more precise interaction point using the calorimeter full granularity and, based on that, extract features of the cluster that may lead to a reasonable discrimination accuracy. This algorithm is followed by a hypothesis making algorithm which takes the extracted features from the target cluster and, with a simple set of cuts is able to distinguish interesting objects with better accuracy. High-p_T electrons are part many interesting Higgs signatures, being one of the most important objects to be detected by ATLAS. Based on the LVL1 configuration and available discrimination information, a strong jet background is expected to contaminate the electron trigger. Approximately, for every 25,000 objects tagged as electrons by LVL1, only 1 will truly be an electron.

The job of the LVL2 e/γ calorimeter algorithm is to remove, as much as possible, the jet contamination and make sure events with true high-p_T electrons are kept and forwarded to the EF for preliminary reconstruction.

2. Neural Networks for ATLAS/LVL2 e/γ detection

Neural networks have been investigated and deployed in other High-Energy Physics experiments [6–8] with great success. They represent a good alternative for the classical Bayesian approach, still keeping very good timing figures. The advantages of their use are:

- robustness: neural networks can be easily retrained to take into account dead-channels;
- maintainability: neural algorithms are available on a variety of platforms;
- speed: since most of the computation can be vectorized, commodity computer architectures can have very fast implementations;
- discrimination performance: by looking at the multi-dimensional input space proposed by most of calorimetry, neural networks can propose better adjusted separation paths.

At this work, we distinguish the two phases of algorithm processing in a modern experiment:

1. Feature Extraction: a phase where the high-dimensionality input data space is compressed into a set of highly discriminant set of features;
2. Hypothesis Making: the phase following the feature extraction, where a decision algorithm inputs the features calculated previously to formulate a decision about the nature of the object being analyzed.

The before-mentioned T2Calo algorithm is an example of a Feature Extractor. Neural networks can be deployed just after to achieve better discrimination efficiency than today's set of simple cuts. Another approach, also investigated in this work, is to propose an alternative feature extraction method and use that as input for the neural discriminator.

2.1. Topological "Ring-Like" strategy

By looking at the way objects normally interact with a calorimeter, it is possible to devise an algorithm that better preserves the features of the available RoI-based input. At this work we propose that, for each ATLAS calorimeter layer, a set of squared-concentric rings are calculated by summing the cells around the central interaction point on that layer. Because the granularity changes radically between the different layers, the number of rings will be different for each layer. The width of the rings is determined by the standard layer granularity. Once the rings geometries are calculated, a sum of the cells that fall on those is produced and accumulated.

Taking into consideration the standard cell sizes on these subdetectors, and an RoI size of 0.4 by 0.4 in $\eta \times \phi$, around 100 sums are produced for each object tagged as electron (by LVL1) to be analyzed. In average, every RoI consisted of ~1,300 cells spread among different calorimeter layers. Figure 1 displays the interaction of an electron with the various calorimeter layers. The left part represents the e.m. section and the right part, the hadronic section.

These sums, or simply *rings*, are fed into a normalization system that equalizes the difference in energy of the input objects and amplifies, in a controlled way, the discriminatory variables which are known to be off-center. This is how it goes: for the first ring, its value is value is divided by the total energy deposited on the layer. The second ring has its value divided by the total energy on the layer *minus* the energy deposited on the first ring. This continues recursively, until some limit on the layer, or the algorithm would be amplifying too much the peripherical noise in the RoI. From this point on, the ring values are divided by a constant.

The set of normalized rings is fed to a feed-forward ("back-propagation") neural discriminator with 100 inputs, 5 hidden neurons and a single output neuron.

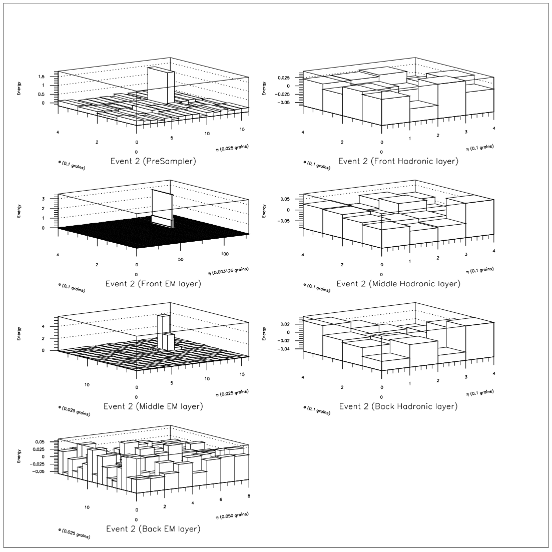

Figure 1. An electron interacting with the ATLAS calorimeter.

3. Results

A set 22,000 electrons from 20 and 30 GeV single electron samples, or derived from simulations of $H \rightarrow ZZ \rightarrow 4e^-$ and $H \rightarrow ZZ \rightarrow 2e^- + 2\mu$ was chosen to represent the electron class. Around 7,000 Jets from simple di-jet simulations, that would pass through a realistic LVL1 filter composed the jet class. Because of the nature of these simulations, a broad energy range was covered. These data were passed by the ring extraction system and normalized. Half of the input was used for training the neural network system and the half for testing the discriminator performance.

For comparison purposes the same data was passed through an up-to-date implementation of T2Calo. It's output was fed into various discriminators, normalizing it with respect to its average and standard deviation if necessary, to adapt the values to the discriminator needs. Figure 2 shows comparative results between all the techniques. As it is possible to devise, the system that compresses less the data, i.e. the proposed topological ring-making procedure, presents the best discrimination results, outperforming T2Calo roughly by a factor of 8, for the same detection efficiency of

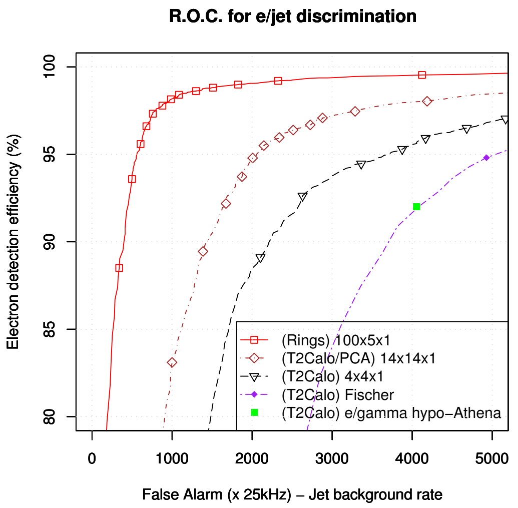

Figure 2. Comparative results between ring-extraction and T2Calo using various discrimination techniques.

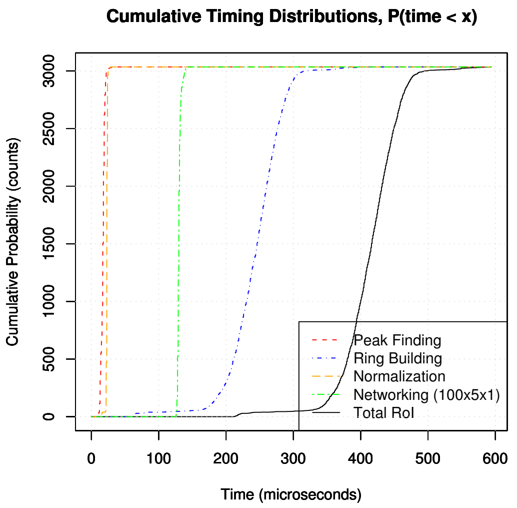

Figure 3. Timing figures for the the *neural-ringer* feature-extraction and discriminator system.

~92%. This figure also shows comparative results for T2Calo's output applied to a Fischer discriminator or using principal component analysis and neural networks.

Figure 3 show the timing performance of the *neural-ringer* discriminator, after basic cell pre-processing from raw data. The figure depicts a series of cumulative distributions for each of the processing phases: ring-making, normalization and neural discrimination and total timings. The C++ code was compiled with GCC, with optimization turned on and ran on a single processor Pentium-IV with 512 MB of RAM and a clock of 2.4 GHz. Taking into consideration the time budget of 10 ms, the average performance of ~450 μs shows adequate for LVL2. No figures of T2Calo were available for comparison as of today.

4. Data Relevance

Once the system base performance was established, optimization can take place. At this work, optimization was carried out by understanding the relevance[9] of every ring value to the discrimination process and by proposing dimensionality cuts based on the importance of every variable. The relevance was estimated in the following way:

$$R_i = \frac{1}{N}\sum_{j=1}^{N}\left[\text{output}(\overrightarrow{x_j}) - \text{output}(\overrightarrow{x_j}\,|_{x_{j,i}=\bar{x}_i})\right]^2 \qquad (1)$$

This technique was deployed for the features available just before the neural discriminator input, after ring extraction and normalization. Although the process is completely blind to the ring location, it is interesting to note that, for the neural discriminator in question, the most relevant information were located around the hottest calorimeter cell in each layer, what confirmed specialist analysis. Figure 4 shows the same as in Figure 2, but includes new discriminators which were trained to take into consideration a subset of the original system. The variables from the orig-

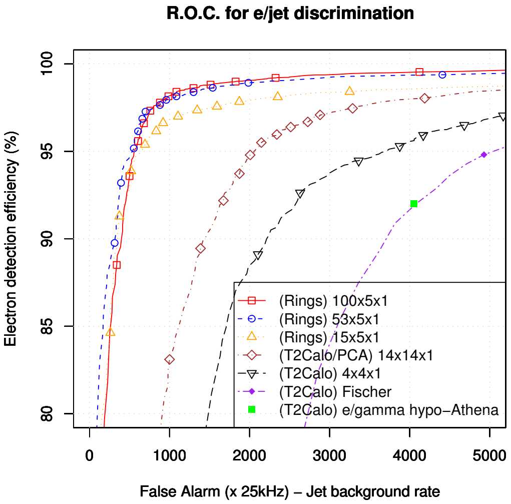

Figure 4. Comparative results showing the relative drop in efficiency caused by the suppression of rings.

inal input space were selected based on the relevance of each variable, by using a fixed threshold. Time performance figures show an average processing speed of ∼300μs for the system with 53 inputs.

5. The DSP Alternative

In digital signal processing applications, one can use any kind of digital device. However, some consideration must be taken into account before choosing the right device. For instance, a general purpose processor (PC), although fast and easy to program, is expensive and demand too much power. A FPGA (*field programmable gate array*) [10] is fast, compact, but very complex to program for high complexity problems.

In digital signal processing, there are algorithms which are very common, like multiply and accumulate, circular buffer access and strong iterative processes. In order to efficiently execute those operations, an special digital device was developed and named DSP (digital signal processors) [11]. The DSP exploits inherent features of the digital signal processing in order to achieve high execution rates in fewer clock cycles. In Figure 5, one can see some of the main differences of an general use processor and the DSP. While general purpose microprocessors have only one ALU with one single bus for transporting instruction and data (Von Neuman architecture [12]), the DSP has, in addition to the ALU, a hardware implemented multiplier, all that connected to an internal memory connected with multiple buses (Harvard architecture (Von Neuman architecture [12])). In addition, the DSP has other independent devices for I/O and efficient data access, in order to maintain the computational units focused only on the data processing.

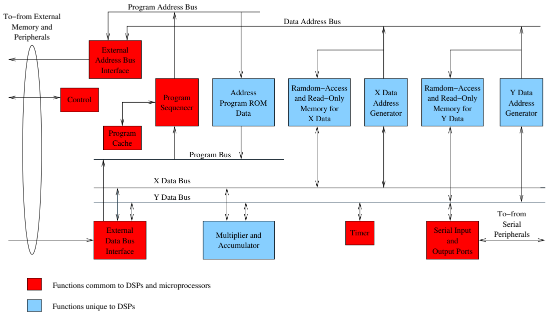
Program Address Bus
Data Address Bus
To–from External Memory and Peripherals
External Address Bus Interface
Control
Program Sequencer
Program Cache
Address Program ROM Data
Ramdom–Access and Read–Only Memory for X Data
X Data Address Generator
Ramdom–Access and Read–Only Memory for Y Data
Y Data Address Generator
Program Bus
X Data Bus
Y Data Bus
External Data Bus Interface
Multiplier and Accumulator
Timer
Serial Input and Output Ports
To–from Serial Peripherals
Functions commom to DSPs and microprocessors
Functions unique to DSPs

Figure 5. General overview of the DSP inner structure.

6. Results Using DSP

The DSP chosen was a floating point, 32-bit, SHARC ADSP-21160 from Analog Devices with 100 MHz clock, 4 Mbits of internal memory and duplicated computational units for SIMD [13] operation. For the proposed electron/jet discrimination problem, performed by a neural network with 100 rings as input signal, an execution time of 4.692 ± 1.108 *ms* per event was achieved.

In Figure 6, is presented the cumulative time distribution functions for each phase of the discrimination process. One can note that the rings

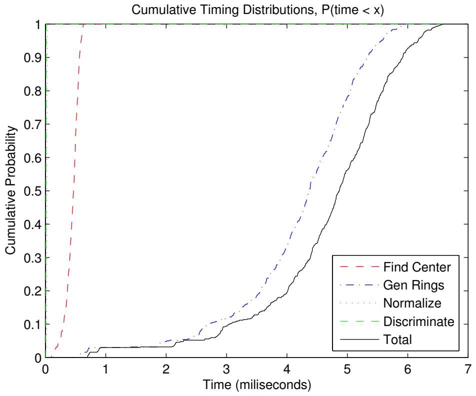

Figure 6. Cumulative time distribution function for each part of the algorithm.

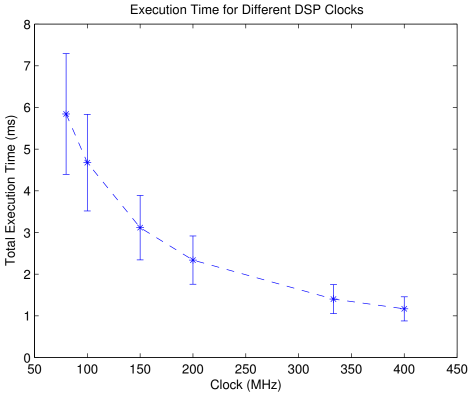

Figure 7. Total execution time for different clock cycles for SHARC family DSPs.

generation step is the most time consuming, due to the high conditional code used to generate the rings, which takes little advantages from the DSP inner structure. Although, the discrimination step, due to the high amount of inner product operations, was capable of fully exploit the DSP features, being able to perform the pattern recognition task in only 10.429 ± 0.465 μs per event, while a Pentium 4 @ 2.8 GHz performed the same task in approximately 125 μs. Finally, the execution time can be further reduced by simply choosing a DSP with higher clock cycle within the same family, for full compatibility. In Figure 7, is presented the total execution time for different clock cycles for SHARC family DSPs, which show the full scalability of these devices. The execution time was obtained by dividing the number of instructions executed in the algorithm (since this family executes every instruction in one single clock cycle) by the DSP's clock cycle.

7. Conclusions

At this work we have proposed a simple discrimination system based on a topological mapping and neural networks, to be deployed at the Second-Level Trigger of the ATLAS experiment. Such a system presents a factor of 8 better physics performance and a very low time budget ($\sim$450μs), taking into consideration the experiment's requirements. A mechanism to define the relevance of every topological component calculated could be deployed to control robustness and speed in very simple terms. A factor of $\sim$2 reduction in the number of rings to be calculated speeded up the system roughly by a 35%.

The implementation of the same algorithm, still in C++, on a DSP running at 100 MHz was carried out. It showed that further reduction on the timings required for discrimination could be achieved ($\sim$10:1) using this platform while keeping a easily maintainable system.

REFERENCES

1. A. Collaboration, ATLAS: Technical Proposal for a general-purpose pp experiment at the Large Hadron Collider at CERN (, 1994).
2. A. Trigger and D.A. Collaboration, ATLAS High-Level Triggers, DAQ and DCS Technical Design Report (, 2003).
3. A. dos Anjos et al., IEEE Transactions on Nuclear Sciences 3 (2004), Part III.
4. W.W. et al., IEEE Transactions on Nuclear

Sciences 3 (2004), Part III.

5. Atlas offline computing.
6. C. Kiesling et al., New Computing Techniques in Physics Research III (1994).
7. C. Lindsey, B. Denby and T. Lindblad, Artificial neural networks in high energy physics, internet: http://neuralnets.web.-cern.ch/NeuralNets/nnwInHep.html.
8. R. Bock, J. Carter and I. Legrand, CERN preprint 11 (1994).
9. A. Gruber et al., New Computing Techniques in Physics Research (1994).
10. S. Brown and Z. Vranesic, Fundamentals of Digital Logic with VHDL Design (McGraw-Hill, 2000).
11. J.H.M. Clellan, R.W. Schafer and M.A. Yoder, DSP First: A Multimedia Approach (Prentice Hall, 1998).
12. J.G. Ackenhusen, Real-Time Signal Processing (Prentice Hall, 1999).
13. Analog Devices, ADSP-21160: SHARC DSP Hardware Reference, , 2 ed., 2002.