

Online-Analysis of Hits in the Belle-II Pixeldetector for Separation of Slow Pions from Background

S.Baehr, O.Sander, M.Heck, C.Pulvermacher, M.Feindt and J.Becker

ITIV, Karlsruhe Institute of Technology, Engesserstrasse 5, D-76131 Karlsruhe, Germany

E-mail: steffen.baehr@kit.edu

Abstract. The impending Upgrade of the Belle experiment is expected to increase the generated data set by a factor of 50. This means that for the planned pixel detector, which is the closest to the interaction point, the data rates are going to increase to up to 28 Gbit/s. Combined with data generated by the other detectors, this rate is too big to efficiently send out to offline processing. In order to reduce the data rates online data reduction schemes, in which background is detected and rejected, are going to be employed. In this paper, an approach for efficient online data reduction for the planned pixel detector of Belle-II is presented. Its central part is the NeuroBayes algorithm, which is based on multivariate analysis. It allows the identification of signal and background by analyzing clusters of hits in the pixel detector on FPGAs. The algorithm is leveraging the fact that hits of signal particles can have very different characteristics, compared to background, when passing through the pixel detector. The applicability and advantages in performance are shown through the D^* decay. In Belle-II, these decays produce pions with such a small transversal momentum, that they barely escape the pixel detector itself. In a common approach like an extrapolation of tracks from outer detectors to RoIs, these pions are simply lost, since they do not reach all necessary layers of the detector. However, cluster analysis is able to identify and separate these pions from the background, thus keeping their data. For that characteristics of corresponding hits, like the total amount of charge deposited in the pixels, are used for separation. The capability for effective data reduction is underlined by a background reduction of at least 90% and signal efficiency of 95%, for slow pions. An implementation of the algorithm for usage on Virtex-6 FPGAs that are used at the pixel detector was performed. It is shown that the resulting implementation succeeds in replicating the efficiency of the algorithm, implemented in software while throughputs that suffice hard real-time constraints, set by the read-out system of Belle-II, are achieved and efficient use of the resources present on the FPGA is made.

1. Introduction

Envisaged luminosities for SuperKEKB are expected to generate extremely high hit rates for the detectors close to the beam pipe [1]. This is especially true for the *pixel detector* (PXD) of the Belle-II experiment, which is located closest to the interaction point. The generated data rates are estimated to reach up to 28 Gbit/s. However underlying DAQ system cannot transmit the data to offline processing with a sufficient rate. To solve this problem online data reduction is used close to the PXD.

The primary mechanism to reduce data in the pixel detector is based on extrapolation of hits in the outer detector layers to areas, so called *Regions of Interest* (RoI), inside the PXD. Only the data of active pixels in these areas is kept. This way most of the interesting particle hits, called signal, are kept while less interesting background is suppressed. However, this method leads to



the loss of all interesting particles that are not even reaching the outer layers.

One important process in Belle-II is the decay of a B meson into a D^* , an orbitally excited D meson. As the B mesons are always produced in pairs, it is useful to do a full reconstruction of the decay products of one B meson thus fixing the four-momentum of the other B. This greatly simplifies reconstruction of the B from the so called signal side. Because of the large branching fraction for the production, it is vital that the D^* is correctly reconstructed. However D^* can decay into so called slow pions, earning its name from having very low energy. They have such a low transversal momentum, that outer layers of the detectors are not reached. In this case, RoI selection suppresses them. That is not acceptable for the Belle-II experiment since reconstruction efficiency would be greatly decreased.

To solve this issue, an alternative approach is used. A machine learning algorithm, the NeuroBayes, is executed online. It has the ability to predict whether a cluster of hits in the PXD was due to a slow pion or background. This way pions can be saved, as they would have been lost otherwise. To match Belle-II data reduction requirements, the algorithm has to be implemented on FPGAs close to the pixel detector.

The NeuroBayes was designed to run on PCs. It has to be shown that porting the algorithm onto FPGAs can achieve efficient separation of slow pions from background. Additionally the throughput of the PXD has to be matched to avoid any overflow and loss of important data. Since available FPGAs in the PXD DAQ are already used for other tasks, the resource demand has to be sufficiently small to allow smooth integration.

This paper is organized in the following way. In Section II, a description of the PXD, RoI selection and an estimation of the envisaged data rates are given. Section III concentrates on the slow pion rescue mechanism. It encompasses the architecture used on FPGA and overview of the NeuroBayes algorithm. Results of the implementation are shown in Section IV. A conclusion and outlook are given in Section V.

2. Related Work

The H1-Level 2 Trigger of the HERA accelerator used neural networks to improve the suppression of the background rate and increase of the signal efficiency [3]. At first it was deployed on dedicated ASICs. In the following trigger upgrade FPGAs were used. It showed the FPGA's capability to host highly parallel and pipelined designs while meeting the hard timing requirements [4].

Neural networks were also proposed for the z-vertex trigger for Belle-II [5]. Its goal is to improve background suppression by estimating the z-vertex more accurately. Data from the central drift chamber is used as an input for the network to make an estimation. The network is planned to be implemented on an FPGA in order to meet throughput requirements.

Both approaches show that machine learning algorithms can be used online to improve background suppression. Hard realtime constraints were met by using FPGAs. The NeuroBayes algorithm, used for the slow pion rescue, is based on neural networks. However, it was originally designed to improve particle identification. For that it uses custom preprocessing algorithms.

3. Belle-II Detector Context

3.1. DEPFET Pixel Detector DAQ

The PXD of Belle-II represents the two innermost detector layers and is part of the *vertex detector* [2] (VXD). It is built based on DEPFET technology [6]. The DEPFET pixels are arranged in 768x250 matrices located on separate modules called half ladder. These modules are arranged in two cylindrical layers around the interaction point. Charge deposited by particles is digitized by the *DEPFET Current Digitizer ASIC* [7] (DCD). They are digitized into 8-bit ADC and then sent to the *Data Handling Processor* [8] (DHP). At the DHP zero suppression, common mode noise and pedestal fluctuation corrections are performed. The processed data

is then passed on to the *Data Handling Hybrid* [9] (DHH), in which data of 5 half ladders is concentrated. Additionally clusters of adjacent hits in the pixel matrices are build. Clusters are then passed on to the *Online Selection Nodes* (ONSEN) [10]. In the ONSSEN the data is matched with RoIs and passed to offline storage. Tracks for RoI selection are delivered by the *data concentrator* (DatCon).

The total amount of pixels in the PXD is at 7.68 million. Meanwhile a worst case occupancy of 3% can occur. As a result the generated data is going to reach about 1 MByte/event. However the ONSSEN has an output data rate of about 100 kByte/event, resulting in a required data reduction of 90%.

3.2. RoI Selection

Data reduction in the PXD is achieved by definition of so called RoI. They define areas in the PXD that are suspected to contain pixels that were hit by interesting particles. Data is only kept for pixels located inside these areas. RoIs are defined by extrapolation of tracks to the PXD. These tracks are constructed using hits in the four outer layers of the VXD, called the *silicon vertex detector* (SVD). As a result particles need to have a certain transversal momentum to reconstruct tracks. This means that at least three hits have to be present at the SVD. Particles not reaching the outer layers are considered as background. This way substantial data reduction is achieved.

3.3. Slow Pions in the Pixel Detector

The D^* decay is important for correct reconstruction of events in Belle-II. One possible product of this decay are pions. They can have varying transversal momenta, shown in Fig. 1(a). Most of the D^* decays below 60 MeV/c include pions, making them even more important for reconstruction. These pions are additionally called slow, earning their name from the low momentum. Due to the low momentum there is a chance for these pions to not reach the outer layers. To investigate that Fig. 1(b) is presented. It shows the layers of the VXD that are reached by pions with a transversal momentum smaller 80 MeV/c. Considering a momentum of 60 MeV/c or less the third layer of the SVD is often times not reached. Less momentum leads to even less pions reaching that three layers. This is important since particles need to reach least three SVD Layers in the for RoI mechanism. As a result a slow pion rescue mechanism is needed to avoid loss of data.

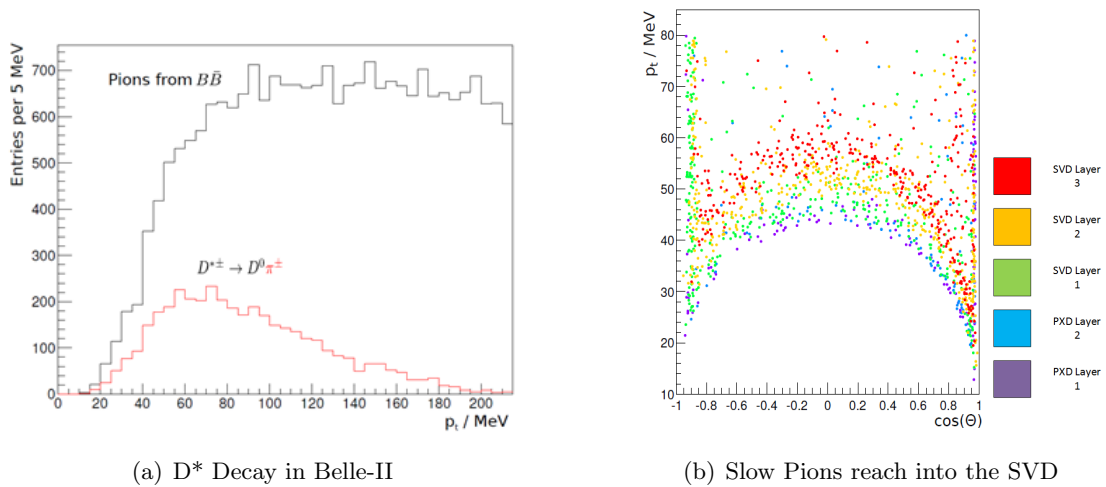


Figure 1

3.4. Slow Pion Rescue in the PXD DAQ

Considering the PXD DAQ system [13] only two viable options exist for the slow pion rescue to be executed on. These are the DHH and the ONSSEN. Both provide FPGAs for hosting the rescue mechanism and are still close enough to the PXD. However the ONSSEN is already tasked with matching of pixel clusters to calculated RoIs. Implementation here could lead to tough integration, since most of the resources are already in use. The DHH on the other hand has around 50% CLBs and 30% DSPs available. For this reason it is chosen to host the slow pion rescue. An overview of the resulting system is depicted in Fig. 2.

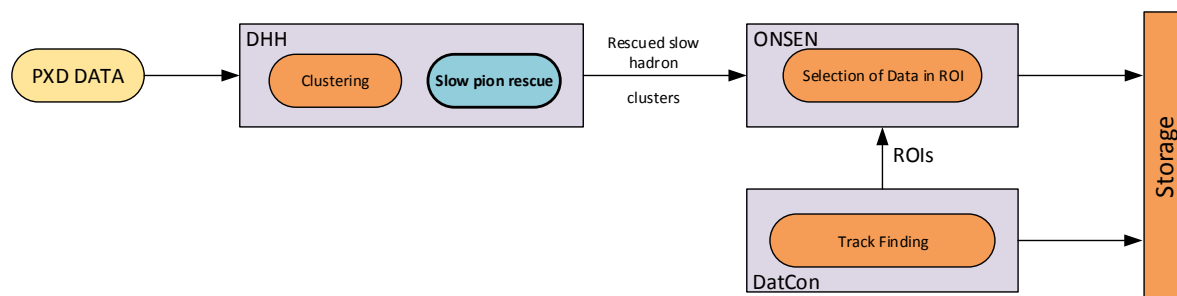


Figure 2: Slow Pion Rescue in the PXD DAQ

4. Slow Pion Rescue Mechanism

4.1. Slow Pion Identification

Slow pions can be identified by using data from pixels clusters, they passed through. The available data consists of the ADC for all pixels in a cluster, the layer and the positions of active pixels in a half ladder. The most indicative characteristic is the digitized charge. Distribution of charges deposited by particles in pixels of the PXD are depicted in Fig. 3 for different momenta. Here four classes of particles are shown, the most important ones being pions and electrons, that are seen as background. The occurrence of pions are indicated by the red arrow with pion label. The area below the red horizontal line at about a cluster seed charge of 50 represents the occurrence of electrons. It is observable, that pions typically deposit much more charge in than electrons. Consequently slow pions can be separated from other particles, by introducing a cut-off for the read out charge from pixels. This method was applied to simulation with the help of basf2. The result was that by introducing this threshold about 50% of the simulated pions could separated correctly from background.

4.2. NeuroBayes Machine Learning Algorithm

Motivated by the possibility of separating hits caused by slow pions from background seen in Section 4.1, more advanced algorithms can be used to achieve high signal efficiencies. One such algorithm is the NeuroBayes[11], which is based on multivariate analysis and was developed for usage as a scientific tool in high energy physics.

The general flow of usage of the NeuroBayes is shown in Fig. 4. It consists of two main parts, they are the Teacher and the Expert. The Teacher is used for generating a prediction model, called the expertise. This model is used for predicting a class to a given set of input data. The data used for the Teacher is typically historic, either taken from a real scenario or simulation. Meanwhile training is conducted supervised. The model is then used by the NeuroBayes Expert. The Expert's task is to predict the correct class for the data given at the input. Its output is a

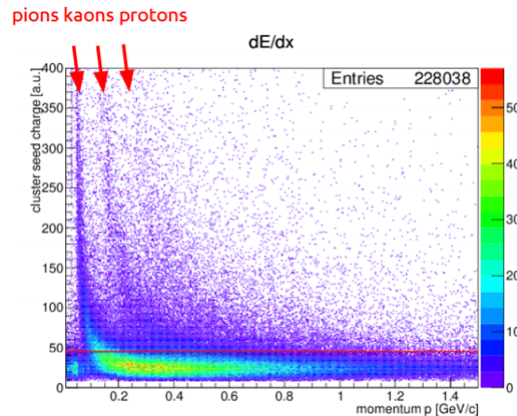


Figure 3: Cluster Seed Charge over Momentum of Particles in the PXD

probability density function representing the algorithms trust into its classification decision. In our case it is the probability for an analyzed particle cluster being a slow pion. The probability is then typically mapped onto a binary value representing the classification decision.

In the slow pion rescue, only the expert is going to be executed on FPGAs. The teacher is used offline beforehand. Here historic data corresponds to simulated clusters of hits in the PXD. They were generated with the help of the basf2 simulation framework[12]. While current data are pixel clusters passed on from the PXD to the expert.

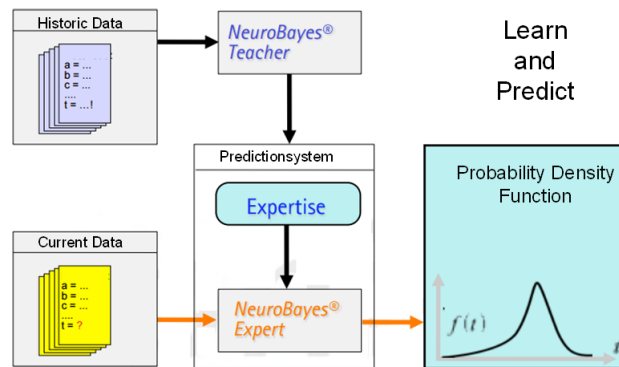


Figure 4: Structural Overview of the NeuroBayes Algorithm

4.3. Architecture on FPGAs

The architecture of the slow pion rescue is depicted in Fig. 5. It consists of three major components, the protocol handling, the feature extraction and the NeuroBayes Expert algorithm. All parts are connected with each other in a pipelined way. This is the result of the required throughput at about 1 processed cluster per clock cycle.

4.3.1. Protocol Handling The protocol handling's task is straightforward, as it decodes data packets of pixel data produced by the clustering on the DHH. Its main purpose is to decouple both mechanisms, as the same packets can be passed on to the ONSSEN without necessarily being processed by the slow pion rescue. This allowed for easier integration into the DHH.



Figure 5: Architecture Overview of the Slow Pion Rescue

4.3.2. Feature Extraction The feature extraction transforms data from the PXD into a more suitable representation, before it can be used by the NeuroBayes Expert. For optimal performance of the algorithm so called features are defined. These are separate data streams, which are preprocessed independently. A feature has a distinct impact on the prediction made by the algorithm, however, they can still be correlated with each other in some way. For the slow pion rescue, 8 features have been found out to be reasonable. These are computed at the feature extraction before being passed on the Expert. They are listed in the following:

- Sum of all pixel charges in a cluster
- Maximum pixel charge of a cluster
- Minimum pixel charge of a cluster
- Layer of the PXD containing the cluster
- Length of cluster in z-direction
- Length of cluster in $r\text{-}\phi$ -direction
- Total length of cluster
- Number of pixels in a cluster

4.3.3. NeuroBayes Expert The NeuroBayes Expert operates on multiple parallel input data streams and can be partitioned into pipelined processing steps. Each input data stream is corresponding to one of the predefined features. The first processing step of the Expert is called binning and it is performed separately and in parallel for each input stream. The main component of this step is the bin, which is an interval in the range of possible values a single feature can assume. Bins do not overlap and are bounded by an upper and lower limit. Binning assigns the current value of an input stream to a bin by checking whether it is within the bin's predefined interval. The assigned bin is then mapped onto a weight, which essentially represents the influence of the selected bin on the prediction. In case that, the value of a feature is between the limits of a bin, an interpolation of the weight is performed.

After the preprocessing all calculated weights are multiplied with a predefined vector. Each entry in the vector contains a value that represents the importance of a feature compared to the others. This accounts for one feature having more significance on the prediction than others. The result of this multiplication is the probability density function.

The last step in processing of the Expert is the cut. Here the computed value of the probability density function is compared to a predefined threshold. If the value is above the threshold, a 1 is returned indicating that this cluster was probably produced by a slow pion. Otherwise a 0 is returned for background. The signal efficiency and background rejection rates vary with the selected threshold value.

As this algorithm was originally written in FORTRAN and developed for usage on PCs all of the mentioned processing components were implemented by hand in VHDL. Fortunately most of the processing steps can be broken down into simple arithmetics i.e. additions, multiplications.

4.4. Performance and Resource Demand

The slow pion rescue was implemented on a Virtex6 VLX75T, which is used at the DHH[14]. Due to the pipelined architecture, throughput is directly corresponding to the achievable clock frequency. Making use of the FPGA's capability to cascade inputs and outputs of DSPs, high clock frequencies are achieved. As depicted in Table.1 350 MHz are achieved. Hereby the critical path was the signal propagation of the used shift registers in vector multiplications. The clusters of hits delivered by the DHH, enter with a frequency of 200 MHz, that means that there's still some cushion for the future. The latency is however dependent on features used for the NeuroBayes Expert. Each additional feature results in an increase of the latency by at least 1 clock cycle. For using 8 parallel input streams the achieved latency is at 12 clock cycles. Regarding the resource demand, the FPGA's DSPs are best used with one input at 25 bits and the other at 18, as one DSP slice is sufficient for each feature. However when increasing either input's bitwidth by 1, at least another additional DSP slice is needed and latency increased to 7 cycles. Fortunately signal identification degrades only slightly, when bitwidth of the data is bounded to these values. The difference of the output probability density function compared to a reference implementation in software is negligible. The total number of CLBs used is at around 2% of the available CLBs.

Throughput	Latency	DSPs	CLBs	Clock frequency
1 Cluster/s	13 cycles	9	2%	350 MHz

Table 1: Performance and Resource Demand of Slow Pion Rescue

4.5. Identification Efficiency

The capabilities of algorithms to identify desired particles is measured with 2 metrics, these are signal efficiency and background rejection. In this case signal efficiency represents the algorithms ability to correctly identify pions out of a given cluster of hits. On the other hand background rejection represents the algorithms ability to correctly identify a cluster of hits that was not produced by a pion, as background. Fig.6 shows the signal efficiency on the x-axis over the achieved background rejection, on the y-axis. Overall 5 classes of pions with different momentum are depicted with different colored lines. The momentum range is from 15 to 65 MeV, these pions are expected to not reach outer layers. It can be seen that overall an overall background rejection of at least 88% can be achieved, this is in case the highest signal efficiency is selected. The implementation behaves the best when momentum is between 25 and 55 MeV. For the targeted Background Rejection Rate of 90%, at least 95% Signal Efficiency is achieved.

5. Conclusion and Outlook

In this paper we showed, that using the NeuroBayes algorithm on FPGAs is a suitable solution for identifying certain particles types, using the data from the Belle-II PXD. Under usage of simulation data generated with the help of basf2, it was shown that the implementation can achieve signal efficiency of at least 95% with a background rejection of 90%. Both are sufficient to match the data reduction requirements set by the Belle-II experiment. Additionally the implementation is suitable to be integrated into Belle-II. The resource demand is small enough to allow for integration with the other components to be used on the DHH. To achieve the strict throughput requirements set by the DAQ, a pipelined architecture is used. Not only is it fulfilling the requirements, but it can reach even higher throughputs than demanded. Future work is going to focus on further increasing the implementation's signal efficiency and allowing for easy adaptation in case of changes in the PXD.

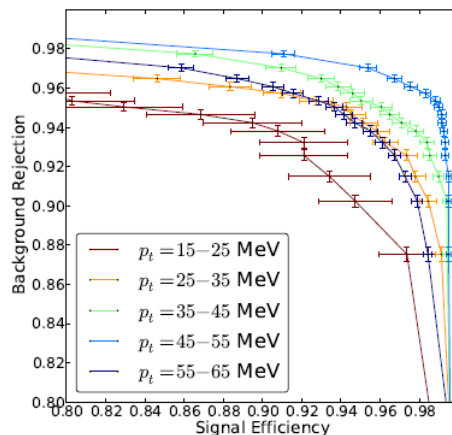


Figure 6: Signal Efficiency over Background Rejection

6. Acknowledgments

The authors would like to acknowledge the support and thank the Belle-II collaboration and Blue Yonder GmbH.

References

- [1] T. Abe et al., Belle II Technical Design Report, ArXiv e-prints, Nov. 2010.
- [2] S. Furlotov, The belle ii pixel vertex detector, Journal of Instrumentation, vol. 7, no. 01, p. C01014, 2012.
- [3] J. Mock et al, "Artificial Neural Networks as a Second-Level Trigger at the H1 Experiment - Performance Analysis and Results", in Proc. New Computing Techniques in Physics Research VI, Heraklion, Greece, Apr. 1999, pp. 8594.
- [4] A. Gruber et al, "A neural network architecture for the second level trigger in the H1-experiment at the electron proton collider HERA", arXiv:1406.3319 [physics.ins-det]
- [5] S. Skambraks et al, "A z -Vertex Trigger for Belle II", arXiv:1406.3319 [physics.ins-det]
- [6] J.Kemmer, G.Lutz. New detector concepts Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 253, no. 3, pp. 365 -377, 1987
- [7] J. Knopf, P. Fischer, C. Kreidl, and I. Peric, A 256 channel 8-bit current digitizer asic for the belle-ii pxd" Journal of Instrumentation, vol. 6, no. 01, p. C01085, 2011.
- [8] M. Lemarenko et al., "The Data Handling Processor for the Belle II Pixel Vertex Detector: efficiency optimization", 2011 JINST 7 C01069.
- [9] Levit, D. ;Konorov, I. ; Zhuravlev, B. ; Paul, S. ; Gessler, T. ; Lange, S. ; Munchow, D. ; Spruck, B. ; Kuuehn, W. ; Jingzhou Zhao ; Zhen'An Liu ; "Data acquisition system of the DEPFET detector for the Belle II experiment"; Nuclear Science, IEEE Transactions on, pp. 1-5, 2013.
- [10] T. Geler, W. Kuehn, Jens S. Lange, Zhen'An Liu, David Mnchow, B. Spruck, J. Zhao, "The ONSSEN Data Reduction System for the Belle II Pixel Detector", arXiv:1406.4028 [physics.ins-det]
- [11] M. Feindt; F. Keller; M. Kreps; T. Kuhr; S. Neubauer, D. Zander, A. Zupanc "A Hierarchical NeuroBayes-based Algorithm for Full Reconstruction of B Mesons at B Factories" arXiv:1102.3876 [hep-ex]
- [12] A.Moll, The Software Framework of the Belle II Experiment, J.Phys.: Conf. Ser. 331, 032024 (2011).
- [13] B. Spruck et al., The Belle II pixel detector data acquisition and reduction system Nuclear Science, IEEE Transactions on, vol. 60, no. 5, pp. 37093713, 2013.
- [14] D. Levit, I. Konorov, D. Greenwald and S. Paul, FPGA Based Data Read-Out System of the Belle 2 Pixel Detector, arXiv:1406.3864 [physics.ins-det]