JUSTUS-LIEBIG-
UNIVERSITÄT
GIESSEN

Bachelor Thesis

Multiparameter Analysis of the Belle II Pixeldetector's Data

Using Principal Components Analysis and Neural Networks

Stephanie Käs

Experimental Physics II

Belle II

Contact: Stephanie.kaes@physik.uni-giessen.de
Titlepage graphics taken from *typesetters.se, wikipedia.org, uni-giessen.de* and *belle2.org.*
Printing: Schreibwaren Baehr Marburg

First printing, July 2019

Justus-Liebig Universität Giessen
II. Physikalisches Institut
FB07

Bachelor Thesis

Multiparameter Analysis of the Belle II Pixeldetector's Data

Using Principal Components Analysis and Neural Networks

Multiparameter Analyse der Daten des Belle II Pixeldetektors mit Principal Components Analysis und neuronalen Netzen

Author: Stephanie Käs
Supervisor: PD Dr. Jens Sören Lange AR
2^{nd} reader: Dr. Hans-Georg Zaunick AR

July 2019

Abstract/Zusammenfassung

Recently, Katharina Dort finished her thesis on the use of self-organising maps and feed forward networks for classification processes, in the analysis of data from Belle II's pixel vertex detector. This thesis continues her work by examining correlations between the charge-related and size-related pixel cluster properties. To find correlations, principal components analysis was applied to a data set of antideuterons and beam background. Additionally, first attempts in the analysis of cluster shapes were made, observing that all clusters are either orientated in 45° direction or parallel to a module's lateral borders. More than half of them can be grouped into single pixel clusters, rectangular, or quadratic shapes. The PCA-transformed data set was used to train a SOM. A comparison of the classification results to a SOM trained with the original data set lead to the result, that PCA does not necessarily improve the outcome. Lastly, use of a SOM was made to differentiate more than one particle from background, for the first time.

Im Mai 2019 veröffentlichte Katharina Dort ihre Ergebnisse über den Nutzen von neuronalen Netzen für die Separierung von Daten des Belle II Pixel-Vertex-Detektors (PXD). Im Folgenden sollen ihre Studien weitergeführt werden. Der Fokus liegt auf der Analyse von Zusammenhängen zwischen den messbaren Eigenschaften von Pixel-Clustern. Zur Untersuchung eines Datensatzes aus Antideuteronen und Beam Background wurden sowohl eine Hauptkomponentenanalyse (PCA) als auch selbstorganisierende Karten (SOMs) verwendet. Die Analyse ergab unter anderem, dass Cluster - abhängig von ihrer Orientierung - in vier Gruppen eingeteilt werden können. Neben den Ein-Pixel-Clustern zeigt die Hälfte der Cluster eine quadratische oder rechteckige Form. Alle anderen Cluster sind entweder parallel zu den Kanten des jeweiligen Moduls des PXD ausgerichtet oder ihre Pixel gruppieren sich entlang eines 45°-Winkels. Der PCA-transformierte Datensatz wurde zum Training von SOMs verwendet. Das Ziel bestand darin, Signale vom Untergrund zu trennen. Bei dem Vergleich mit nicht transformierten Daten ergab sich, dass die PCA nicht zwangsläufig zur Verbesserung der Trennung von Signal und Beam Background führt. Zuletzt wurde zum ersten Mal eine selbstorganisierende Karte verwendet, um mehr als drei Teilchen vom Beam Background zu separieren.

Contents

Chapter 1

Introduction

A lot has been achieved within the last 20 years of research in particle physics. To state an example, the spectroscopy of bound states of elementary particles led to the discovery of so-called exotic states. Exotic states do not fit into predictions made by physical models that have been accepted as true for a long time. One of those theories is the standard model of particle physics, which describes interactions between elementary particles. New theories do not claim to replace what has been proven to be successful for a long time, but they try to explain observations that go beyond the physics included in their predecessors. To verify those theories, experiments like Belle II are created. Located in Japan, Belle II is an international cooperation of research institutes from all around the world. Common to all participants is the one big goal: the search for new physics beyond the standard model.

This thesis is devoted to the analysis of data coming from the Pixel Vertex Detector (PXD). It is one of Belle II's subdetectors and placed around the collision point of particles. This position enables the PXD to detect particles that will not reach outer detector parts. These *Highly Ionising Particles* (HIPS) are characterised by their high energy loss along their way through the detector's material. Among them, magnetic monopoles and exotic states are expected. For the future detection of exotic states, a proper understanding of the parameters measured in the PXD is needed. For this reason, the aim of this thesis is to discover correlations between different properties of pixel clusters. Any measured data set includes signals coming from decay products of colliding particles as well as background noise. To differentiate between intentional signals and background, artificial intelligence is applied. In this field of study, important achievements have been made by KATHARINA DORT (Ref. [1]). Therefore, this thesis can be seen as direct successor of her work[1].

The particles that have been used in this study are antideuterons, negatively charged pions and tetraquarks. Especially tetraquarks are of interest as the discoveries of exotic states holding more than three quarks verify different theories that have been introduced since the 1960's. The existence of tetraquarks based on quarkonia was confirmed by the discovery of X(3872) at Belle II in 2003. In the last years, upcoming theories on tetraquarks led to the search for tetraquarks that do not contain quarconia. One of those states is $\Upsilon(3882)$, which resembles the antideuteron in charge. The "tetraquarks" in this thesis are simplified versions of $\Upsilon(3882)$. If $\Upsilon(3882)$ is discovered one day, theories like the one from KARLINER and ROSNER (Ref. [2]) can be verified.

The present study will start with a short introduction of the physics background including the standard model of particle physics and exotic states. In the same chapter, a short overview of the Belle II experiment and its subdetectors is given. Chapter 3 focuses on the methods of data analysis and introduces the reader to principal components analysis (PCA) and self-organising neural networks (SOMs). The research part of this thesis is separated into three projects. The first project focuses on the analysis of a simulated data set of antideuterons and beam background. The computer-based simulation did create data similar to the one that would have been generated by real particles passing the PXD. To find correlations between the simulated data properties, principal components analysis has been applied. The results will be discussed in detail in chapter 4.

[1] Her thesis can be found on *fb07-indico.physik.uni-giessen.de:8080/wiki/index.php/Category:Publications.*

The PXD is made from modules consisting of matrices of pixels. If an ionising particle passes the PXD, pixels will get activated. These activated pixels can be grouped into clusters. Each cluster belongs to a particle that has passed the PXD. The second project, presented in chapter 5, focuses on investigations concerning the clusters' spatial orientation. It provides first indications for tendencies of shapes among the clusters. Cluster shapes are interesting for research as they include information of the special interaction of different particles with the PXD.

In analogy to Dort's work, SOMs are used in chapter 6 to differentiate signals from background. This project is split into two parts. Firstly, the antideuterons and beam background set, that has been pre-transformed by PCA, will be used to train a SOM. The resulting maps will be compared to maps that have been trained with data without former PC-transformation. Secondly, an attempt is made to differentiate antideuterons, negatively charged pions and tetraquarks, from background signals. The thesis closes by a summary and discussion of usability of the results. Lastly, a short outlook on other methods that might be useful for the analysis of PXD clusters is given.

Chapter 2

Physics and Experimental Background

2.1 The Standard Model of Particle Physics

The standard model of particle physics (SM) lists all so-far observed elementary particles. At the present time four fundamental interactions are known: gravitation, weak, strong and electromagnetic (EM) interaction. The SM only includes three fundamental interactions. Gravitation is not included as its hypothetical mediating particle still remains to be determined. Nevertheless, the standard model of particle physics is confirmed by several experiments and compatible with the special theory of relativity and quantum mechanics. The elementary particles are often listed in a table as seen in 2.1. Depending on their spin they can be grouped in fermions or bosons. Mediation particles responsible for interactions between particles, are always bosons. Fermions are characterised by half-integered spin, whilst bosons always show whole-numbered spins. Additionally, fermions subgroup in leptons and quarks depending on their charge. These subgroups split into three generations. Within a generation, mass increases from left to right. For example, electrons are lighter than muons and tauons, and up-quarks lighter than charm- and top-quarks. Each lepton generation includes charged and non-charged particles, called neutrinos. The most important fact is that for each fermion an anti-fermion with opposing charges exists.

Table 2.1: The standard model of particle physics

Fermions	1	2	3	Interaction		Gauge Bosons	Interaction
Leptons	ν_e	ν_μ	ν_τ	weak		γ	EM
	e	μ	τ	EM, weak		g	strong
Quarks	u	c	t	EM, weak, strong		Z^0, W^+, W^-	weak
	d	s	b	EM, weak, strong			

Mediation particles (also: *gauge bosons*) transfer energy and momentum between particles [1, p.5]. For example, the proton consists of three quarks (uud) and therefore it is expected that the sum of quark masses is equivalent to the total proton mass in rest. This is not the case and hence other particles exist that carry energy and momentum. Some gauge bosons carry mass like Z^0, W^+ and W^-, others (γ and g) are massless, but they do carry momentum. Mediation particles are photons (γ) for electromagnetic interactions, gluon (g) for strong interaction and Z^0, W^+, W^- for weak interaction. The strength of the interaction depends on a coupling constant, that is – unlike its name implies - not constant, but depends on momentum transfer. One must point out that particles can only participate in interactions if they carry the respective charge. An exception from this rule is the Higgs-Boson that is responsible for all elementary particles' masses.

Observable matter always consists of combinations of fermions. Whereas leptons can be observed as single particles, quarks always pair in groups. This is a phenomenon of strong interaction, called colour confinement. With strong interaction a new type of charge, called colour, is included in the SM. Colours are red (r), green (g) and blue(b), supplemented by their anti-colours. Neutral colour charge is white and can be created by combining either colour and its respective anti-colour or *rbg* ($\bar{r}\bar{b}\bar{g}$). An absolute precondition for matter formed out of quarks (hadronic matter), is that the total colour charge combines to white.

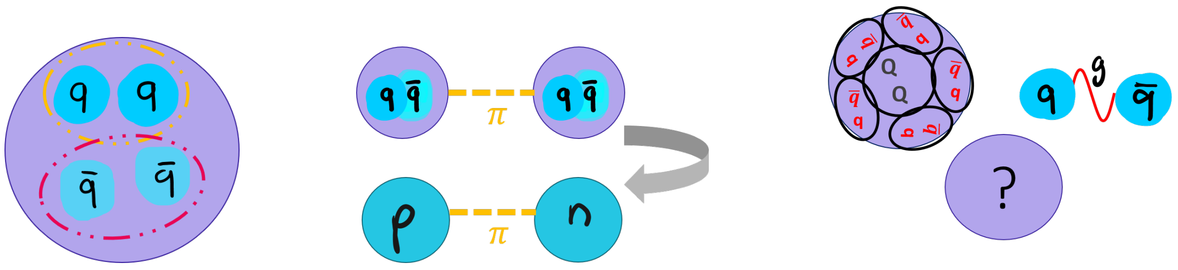
Hadronic matter can be divided into mesons and baryons. As classical models understand, mesons include two quarks and baryons host three quarks. Modern experiments like Belle II will show that multi-quark states like tetraquarks are also possible. Fundamentally the idea of multi-quarks states is not new and can be found in Gell-Mann's paper *A schematic model of baryons and mesons* published in 1964 [3]. Multiquark states will be explained in more depth in the following, since a part of this thesis covers methods for data separation of tetraquarks from background.

As the SM does not include gravitation, it cannot be a complete description of particle physics. In addition to gravitation, several other phenomena like dark matter and dark energy [1, p.8] affirm that an expansion of the SM is needed. Most of the parameters in the SM like masses of quarks and leptons as well as the coupling constants are not explained by the SM, but simply taken from experimental results. Experiments like Belle II are designed to confirm theoretical models, that try to explain physics beyond the standard model and explore new physics going beyond those theories.

2.2 Exotics: Multiquark States

As mentioned above, Gell-Mann's theory of quark states includes multiquark states. Multiquark states like tetra- and pentaquarks tend to have a very short mean lifetime ($< 10^{-21}$ s) [4, p.68] accordingly, they cannot be seen as tracks in detectors. To prove the existence of multiquark states, decay products are measured and compared with already known decay reactions. Plotting the invariant mass versus rates, can help to find stable states as they appear as peaks in the plotted curve. In this way, it has been possible to verify exotic particles like X(3872) [5, p.68].

One should begin by stating that there is no proper definition of *exotic states*. A possible way to examine exotics is to compare an unknown state to known *quarkonium states*. The name *quarkonia* is used for bonded states of quark and corresponding antiquark. Analog to positronium, a quarkonium has excited states that can be visualised in a term diagram (also: *Grotrain diagram*). If an unknown particle does not fit into predicted states of quarkonia, it is called *exotic*. Even though it is possible to give a probability for the quark composition of exotic states, the nature of exotic states still remains uncertain. Many theories have been discussed in the last year, including *compact* and *diquark-diquark tetraquarks*, *hadro-charmoniums*, *molecule-like states*, *hybrid states* and *glueballs* [5, 6]. It is highly probable that different exotics can have different structures. Thus, judging whether a theory has to be proven wrong is very hard, as the risk remains that a particle fitting into a certain model exists, but has not as yet been found.



(a) Classic tetraquark

(b) Molecule-like

(c) Others

Figure 2.1: Schematic drawing of some possible natures of multiquarks [7] (based on [8], [6]). (a) respresents the *classic tetraquark* state consisting of two mesons. (b) Tetraquarks might be *molecule-like* states bound together by exchange particles. This model is created in analogy to the deuteron (turquoise)). (c) Possible other natures of multiquarks like *hadro-charmonium* (left) and *hybrid* states (right).

2.2.1 Tetraquarks Including Quarkonia

Example: The X(3872)

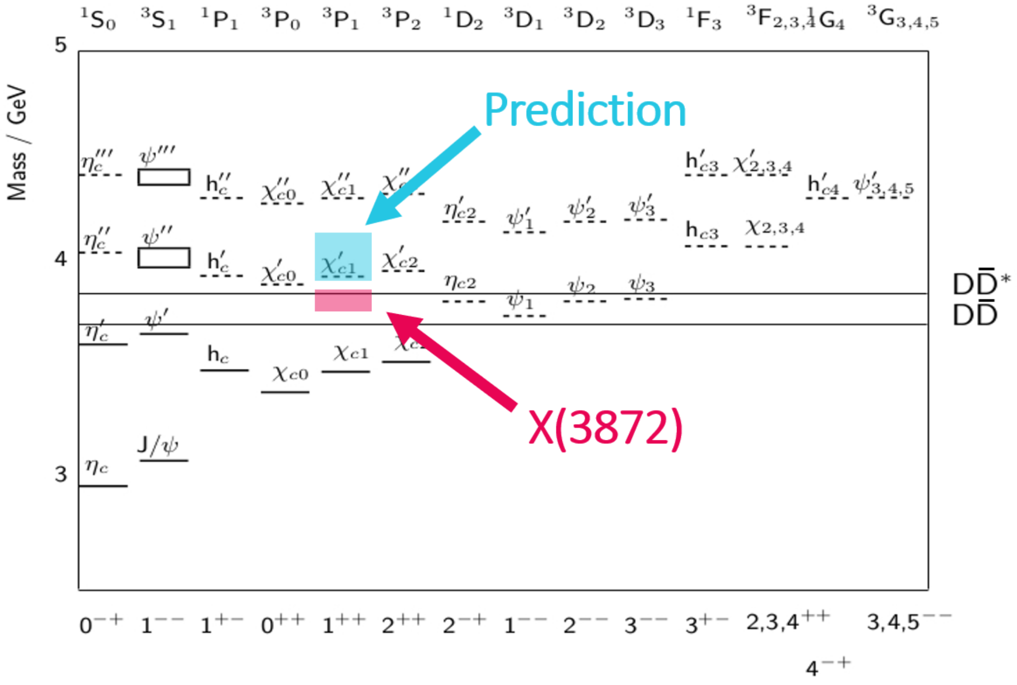
X(3872) is produced in decays of $B^{\pm}$ and was first discovered in Belle. It is considered as being an exotic tetraquark state. Its nature still remains unknown, but *charmonium* appears to be one of its decay products. Charmonium can be classified as quarkonium. Its Grotrian diagram can be seen in figure 2.2. The y-axis shows the energy levels, and on the x-axis, the corresponding quantum numbers J^{PC} (lower x-axis) and $^{2S+1}\mathrm{L}_J$ (upper x-axis) are depicted. Both axis labels J^{PC} and $^{2S+1}\mathrm{L}_J$ are included as they contain the same information due to the following relations between the quantum numbers J, L, S, P and C:

- $J = L + S$
- $P = -1^{L+1}$
- $C = -1^{L+S}$

Different excited states of charmonium are represented by short lines. Normal lines correspond to states that have been theoretically predicted **and** experimentally confirmed. Dashed lines represent states that have not been observed in experiments. Little rectangles belong to states which are unstable. The two horizontal straights in the centre represent threshold energies. Any state with energies higher than the threshold energy is able to decay into the particles listed at the edge. In the case of charmonium, these decay products are $D\bar{D}$ and $D\bar{D}^*$. Since particles are able to decay in strong interaction, when their masses are higher than the given threshold, they are named *unstable*. While measuring energy and momentum of the decay particles, X(3872) appeared as a sharp peak in the *invariant mass*

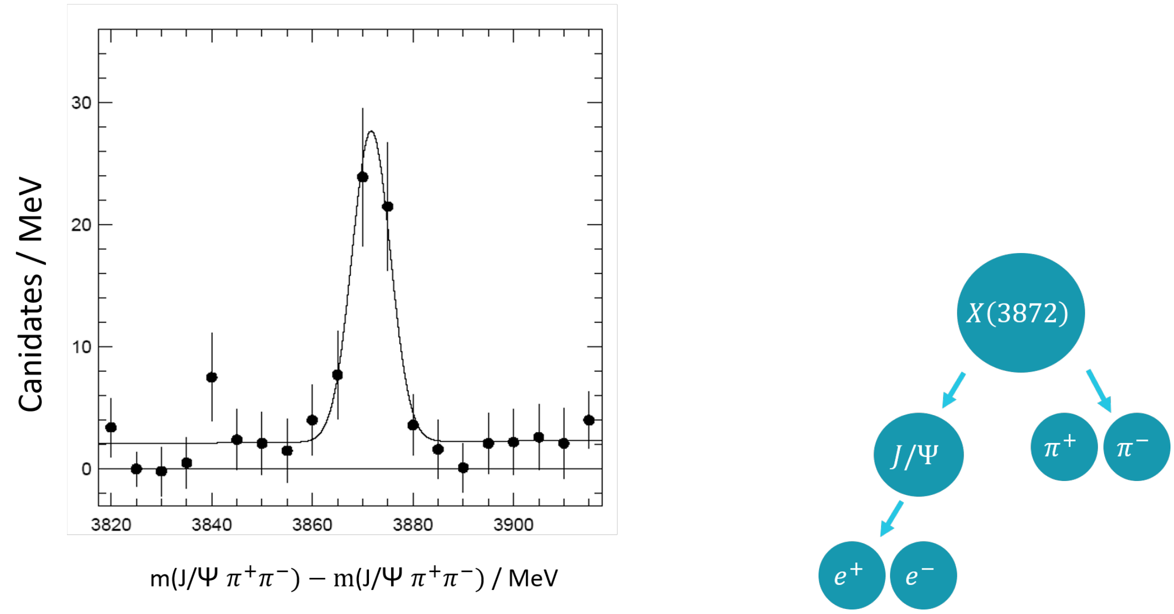
$$m = \frac{1}{c}\sqrt{\frac{E^2}{c^2} - \boldsymbol{P}^2} \tag{2.1}$$

distribution of decay products. After the mass of X(3872) was reconstructed, no matching charmonium state was found. The best matching predicted charmonium state was 50 MeV lower in mass than X(3872). As time went by, 7 independent experiments confirmed its mass of 3.872 GeV. X(3872) decays in an electron-positron pair and a pair of pions (figure 2.3).



$^1S_0$ $^3S_1$ $^1P_1$ $^3P_0$ $^3P_1$ $^3P_2$ $^1D_2$ $^3D_1$ $^3D_2$ $^3D_3$ $^1F_3$ $^3F_{2,3,4}$ $^1G_4$ $^3G_{3,4,5}$
Mass / GeV
5
4
3
Prediction
X(3872)
η_c''' ψ''' h_c'' χ_{c0}'' χ_{c1}'' χ_{c2}''
η_{c2}' ψ_1' ψ_2' ψ_3' h_{c3}' $\chi_{2,3,4}'$ h_{c4}' $\psi_{3,4,5}'$
η_c'' ψ'' h_c' χ_{c0}' χ_{c1}' χ_{c2}'
η_{c2} ψ_1 ψ_2 ψ_3 h_{c3} $\chi_{2,3,4}$
η_c' ψ' h_c χ_{c0} χ_{c1} χ_{c2}
η_c J/ψ
$D\bar{D}^*$
$D\bar{D}$
0^{-+} 1^{--} 1^{+-} 0^{++} 1^{++} 2^{++} 2^{-+} 1^{--} 2^{--} 3^{--} 3^{+-} $2,3,4^{++}$ $3,4,5^{--}$
4^{-+}

Figure 2.2: Grotrian diagram of charmonium. The y-axis represents the energy of the charmonium states. The y-axis represents the sprectroscopical numbering scheme of the quantum numbers of the different excited states. Picture taken from [8].



(a) (b)

Figure 2.3: (b) Peak caused by X(3872) in invariant mass observed at Belle in 2003 [8]. (a) Decay scheme of X(3872) [7].

2.2.2 Tetraquarks Excluding Quarkonia

Most of the tetraquarks that have been observed so far are made of a heavy quarkonium $Q\bar{Q}$ and two light quarks q_i and q_j. Karliner and Rosner [2] presented their hypothesis on QQq_iq_j-states, focusing on a stable $bb\bar{u}\bar{d}$ tetraquark. Their work has been motivated by the discovery of stable cc-baryons at LHCb. They predicted that the $bb\bar{u}\bar{d}$ has a mass of $10\,389 \pm 12\,\text{MeV}$ with $J^P = 1+$. Strong and electromagnetic decays are therefore suppressed due to parity and charge conservation laws. Weak decays are possible. It has been considered stable under strong and electromagnetic interaction, as its mass lies $> 100\,\text{MeV}$ underneath its decay threshold into two B-mesons.

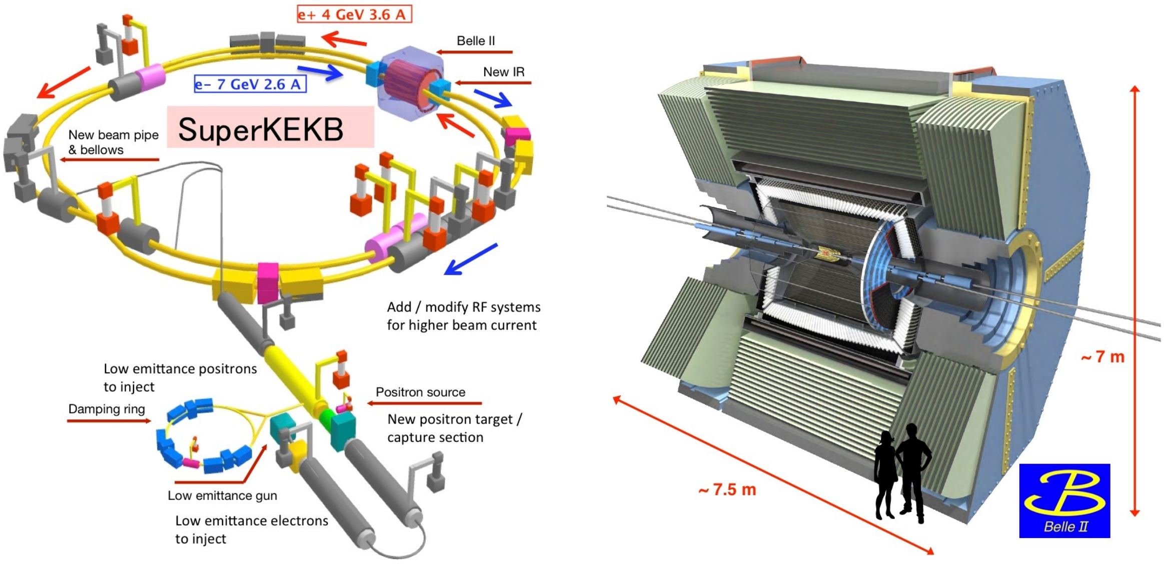
Additionally, to $bb\bar{u}\bar{d}$, the mass of a $cc\bar{u}\bar{d}$-state was predicted to be 3882 ± 12 MeV [2]. This particle is called $\Upsilon(3882)$. The mass of $\Upsilon(3882)$ lies 7 MeV higher than its $D^0\bar{D}^{*+}$ treshold and 148 MeV higher than its its $D^0\bar{D}^+\gamma$ threshold[1]. Therefore Karliner and Rosner labelled it as unstable. It has to be stressed that the $D^0\bar{D}^{*+}$-threshold is included in the interval of the predicted error. Therefore the possibility exists that $\Upsilon(3882)$'s real mass is actually lower and therefore it might be stable. The authors suspect that $\Upsilon(3882)$ shows violations of isospin, such as the ones discovered in decays of X(3872).

2.3 The Belle II Experiment

All simulations in this thesis are done for the Belle II experiment in Tsukuba, Japan. Belle II is an experiment using the electron-positron collider and accelerator rings of SuperKEKB. SuperKEKB belongs to the High Energy Accelerator Research Organisation (KEK) and is the successor of KEKB. With $2.11 \cdot 10^{34}\,\frac{1}{cm^2\,s}$ [9] Belle II is planned to reach the world record for collisions with the highest instantaneus luminosity. The aim of the Belle II experiment is to search for new physics like multiple Higgs, dark matter decays or exotic states. In contrast to other experiments, it concentrates on precise

[1] The parity and charge of $J^P = 1+$ forbids $\Upsilon(3882)$ to decay strongly or electromagnetically into D^0D^+. The hadronic decay channel of lowest energy is $D^0\bar{D}^{*+}$.

measurements of rare processes instead of high energies [1, p.12].The following section gives a short overview of the Belle II detector. Most of the information is taken from Gessler's dissertation [10, p.33-47] who referred to the official *Belle II Technical Design Report* [11]. Other sources are explicitly marked.


e+ 4 GeV 3.6 A
Belle II
New IR
e- 7 GeV 2.6 A
SuperKEKB
New beam pipe & bellows
Add / modify RF systems for higher beam current
Low emittance positrons to inject
Positron source
Damping ring
New positron target / capture section
Low emittance gun
Low emittance electrons to inject
~ 7 m
~ 7.5 m
Belle II

(a) SuperKEKB

(b) Belle II Detector

Figure 2.4: (a) Schematic drawing SuperKEKB and its accelerator rings (b) Virtual view into the Belle II detector. The PXD is the innermost detector, directly surrounding the interaction point. Pictures taken from *www.belle2.org* [12]

2.3.1 The SuperKEKB Accelerator

The SuperKEKB accelerator consists of two rings with a circumference of 3.016 km each. Electrons are accelerated to 7 GeV and stored in the high-energy ring (HER), whilst positrons are only brought to energies up to 4 GeV. They are stored in the low-energy ring (LER). Particles are grouped in 2506 bunches per ring with 10^{11} particles each. Due to the energy difference, SuperKEKB is an asymmetric collider. As electrons and positrons are accelerated by the the same linear accelerator, the acceleration has to alternate pulse wise. The frequency of 50 Hz leads to both an almost constant luminosity and noise that increases the background. Electrons are emitted by radiofrequency photocathode electron guns, and then boosted to their final energy in the linear accelerator. To create positrons, a thermionic radiofrequency electron gun is used. This gun generates 3.3 GeV electrons that produce bremsstrahlung photons. Decay of these photons lead to electron-positron pair production. The positrons are accelerated stepwise before injecting them into the LER. Passing the damping ring before accelerating the positrons to 4 GeV, reduces their emittance.

> The beam energies were chosen such, that in the collisions, mainly B-mesons were produced, which is the reason why the facility is also known as a B factory.
> (*www.belle2.org* [12])

The reachable center-of-mass (CMS) energy is 10.58 GeV [1, p.15] which corresponds to $\Upsilon(4s)$'s mass. The $\Upsilon(4s)$-resonance almost exclusively decays in B-mesons. As SuperKEKB is also called a "B-meson factory" it stands to reason that the CMS-energy is chosen as $\Upsilon(4s)$'s mass. A special feature of the SuperKEKB accelerator is the nanobeam scheme. Compared to SuperKEKB's predecessor KEKB the colliding beams are highly compressed in a vertical direction. At the interaction point, the ratio of both beams is in nanometer range and therefore very small. This compression leads to a 40-times higher luminosity than in KEKB's Belle experiment.

2.3.2 The Belle II Detector

The Belle II detector is placed around the interaction point to detect particles resulting from the collision. Belle II is optimised for tracking and energy measurement, as well as particle identification. Due to its geometry, Belle II does not have a 360° acceptance in the laboratory frame. The azimuthal plane covers angles ϕ of 360°, whereas the polar angle θ has an acceptance between 17° and 150°. The forward direction corresponds to θ-values of zero. Belle II can be split into the forward region ($1° < \theta < 30°$), barrel region ($30° < \theta < 125°$) and the backward region ($125° < \theta < 150°$). Belle II's subdetectors are arranged in a stratified order from the interaction point outwards: PXD, SVD, CDC, Top, ARICH, ECL, solenoid coil and KLM. As the pixel detector (PXD) has the highest relevance for this thesis, it will be discussed in the end.

Vertex Detector

The vertex detector (VXD) is a semiconductor detector. Consisting of six silicon detector layers in a cask-like structure, the VXD allows precise reconstruction of decay vertices. It is mostly used to reconstruct B-meson decay vertices. The PXD comprises the inner two layers of the VXD. The outer four layers are formed by the so-called SVD (silicon vertex detector). It consists of 187 sensors in either trapezoidal or rectangular shape. The modules are built out of slightly overlapping sensors that prevent track-loss through inactive gaps in the detector regions.

Central Drift Chamber

The central drift chamber (CDC) surrounds the vertex detector. It covers a cylindrical ring-shaped region with radii from 1.6 to 11.3 cm and a length of 2.4 m. The whole chamber is filled with a helium-ethane-gas mixture (1:1). As the CDC is a wired chamber, it holds 9 superlayers of sense wires made from gold-plated tungsten and aluminium field wires. The lattice structure of the 14 336 sense wires enables detection of an event's coordinates. Since the CDC is surrounded by a magnetic field of 1.5 T, charged particles ionise the gas atoms along a helical trajectory. Moving along the electric field in the chamber, freed electrons are able to produce a signal in the sensor wires. As the drift velocity of electrons is known and the time between signals is measurable, it is possible to calculate the actual position of the particle in the CDC. Following on from that, the helical trajectory of the particle can be reconstructed. This enebales to calculate the particle's momentum out of the trajectory's radius. As the energy loss per distance dE/dx depending on the momentum highly differs between different particle types, the CDC can be used to identify particles.

Cherencov Detectors for Particle Identification

Ring-imaging (RICH) and imaging time-of-propagation (iTOP or TOP) Cherenkov-detectors are used for particle identification. Cherenkov light is created by charged particles that move through a dielectrical medium with a velocity higher than the phase velocity of light. Atoms along the particle's trajectory are polarised and emit electromagnetic radiation that can be detected. The Cherenkov light behaves like a Mach-cone and can be seen on a screen. If the distance between light-source and screen as well as the medium's refraction index n, is known and the radius measured, it is possible to calculate the particle's velocity. In this case, use is made of the fact that the opening angle of the light cone depends on the velocity.

The iTOP consists of sixteen transparent silicia quartz bars that are attached barrel-like around the CDC. iTOP detectors are underpinned by the same principles as DIRC (detection of internally reflected Cherenkov light detectors). Cherenkov light in DIRCs is totally reflected in the detector's material in such a way that it is forced to pass the whole detector without leaving it. The principle can be compared to fibre optic cables. At the end of the detector the light is guided into sensors consisting of photomultipliers. The other end is secured by mirrors so that light can only leave the quartz bars in one direction. iTOPs are used together with the CDC to allow to draw precise conclusions about momentum, coordinates and angle at which a particle hits the detector. Another type of RICH detectors used in Belle II is the Aerogel RICH (ARICH). Particles move through a box filled with aerogel. Aerogel radiators are arranged as pairs in a sandwich structure with free space in-between. The second radiator's refraction index is chosen in a way that a light cone with a bigger angle is created. In this way, it is possible to increase the resolution of the ring, that has to be detected by photo detectors. ARICHs are used at the forward end-cap region.

Both detectors offer the opportunity to differentiate charged kaons from charged pions by exploiting the mass difference. For a given momentum, kaons are heavier and emit cherenkov light in a wider cone than pions. Concerning the iTOP, kaons need less reflections to pass the material and are therefore faster than pions.

The Electromagnetic Calorimeter

The main tasks of the electromagnetic calorimenter are to detect photons created from decay products of B-mesons and measuring the luminosity. It is also used in combination with the KLM to detect K_L^0. The electromagnetic calorimeter consists of 8736 (CsI(TI))-crystals. Scintillation light is created by particles passing the crystals and can be detected by photdiodes attached to the crystals [1]. As hadronic showers leave broader tracks than electric showers, it is possible to differentiate between photons and neutrons.

The K_L^0 and Muon Detector

The K_L^0 and muon detector (KLM) is placed in the forward and backward end-cap as well as in the barrel region. It consists of 14 iron plates. Each of the 47 mm thick plates absorb kaons. Muons are not absorbed in the plates, but they lead to hadronic showers. To detect these showers, different detector types are used. In the barrel region, the gaps between the plates are filled with two RPCs (resistive plate chambers) separated by gas. When an electric field between two RPCs is applied, charged particles that pass through the region lead to gas ionisation. Thus, a conductive channel between the electrodes appears. The charge flow causes a discharge that can be registered as a signal. As the RPCs have a long dead-time they are inefficient in regions with increased background. Therefore, in the end-caps organic scintillator strips are used instead of RPCs. Passing particles lead to scintillation light, which photomultipliers amplify.

The Pixel Vertex Detector

The PXD surrounds the interaction point. Due to the proximity to the interaction point, it is possible to detect decay vertices of charged particles with a short lifetime such as B-mesons.

To detect events, a double-layer of silicon detectors is used. Both are arranged in a barrel-like structure, the first at 14 mm in radial direction, and the second one at 22 mm. The PXD consists of 40 modules. Each sensor module is built of 768×250 pixels as well as ASICs for data read-out. A schematic drawing of a PXD ladder including a module is depicted in figure 2.5.

Each module can be seen as a plane formed by a vector in $\boldsymbol{u}$-direction and another vector in $\boldsymbol{v}$-direction. Pixels can therefore be interpreted as a point of (u,v)-coordinates, with a measureable *pixel charge*. In this thesis pixels grouped together are refered to as *clusters* and interpreted as response to the same particle passing the PXD. Thus, properties are analysed for clusters. These properties can be the cluster's overall number of pixels (*"size"*), the length in $\boldsymbol{u}$-direction and $\boldsymbol{v}$-direction (*"size in u"*, *"size in v"*) and the sum of the charge (*"charge"*). Additionally the minimal charge of a pixel in the cluster (*"minimum charge"*) and the maximal charge (*"seed"*) are analysed. For each cluster one can also reconstruct the layer and ladder of the module in which it is embedded, to reconstruct the position of the cluster.

As the track density in horizontal direction close to the interaction point is higher, the pixels need to be smaller (50 µm ×55 µm) to enable precise measurement. Any other pixels are 50 µm × 85 µm in size. With 75 µm all pixels have equal height[2]. The pixels are made from depleted field-effect transistors (DEPFET). In general DEPFET pixels can be seen as a special form of metal-oxide-semiconductor field-effect transistors (MOSFETs).

A pMOSFET is built from an n-doped silicon wafer. Source and drain are implanted as p-doped regions. As the contact between the p and n region is a neutral zone, no charge flows. To create a channel, through which charge flow can be regulated, a gate electrode is applied between source and drain. To enlarge the resistance, the gate is isolated from the n-doped region by a metal-oxide layer.

[2] J. S. Lange, personal communication, July 7, 2019

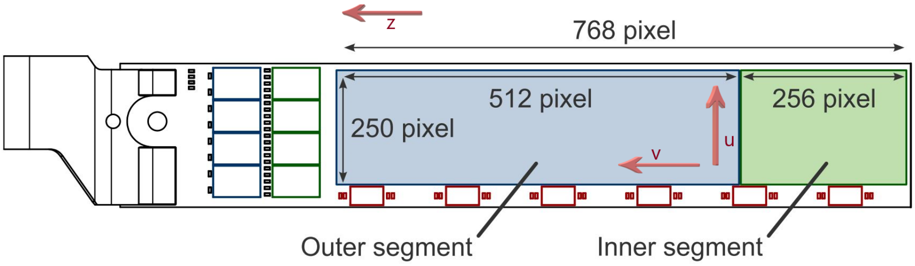
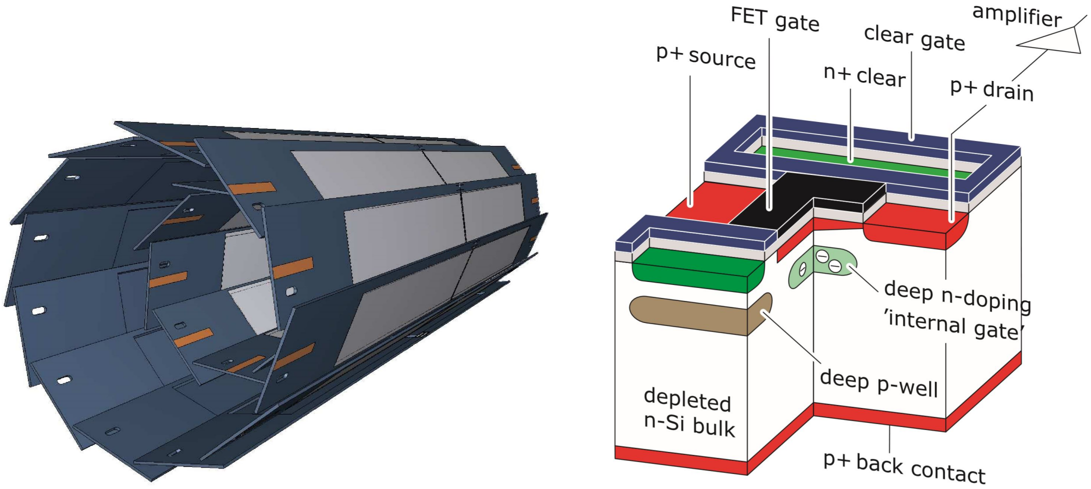


Figure 2.5: Schematic drawing of a PXD ladder. Each ladder is built of an inner and an outer module. The plane containing the pixels is described by the coordinates $\boldsymbol{u}$ and $\boldsymbol{v}$. The $\boldsymbol{u}$-xis is equivalent to the direction vector of the polar angle ϕ, whereas the local $\boldsymbol{v}$-axis is equivalent to the global $\boldsymbol{z}$-axis. Original picture taken from [13, p. 53].



(a) PXD (b) DEPFET pixel

Figure 2.6: (a) Schematic drawing of the PXD sensors. Light grey regions belong to matrices of DEPFET pixels. Each sensor ladder hosts DEPFET pixel matrices. The windmill structure prevents non-sensitibity at the transition region between sensor ladders. (b) Schematic drawing of a DEPFET pixel. Both pictures taken from [11, p.79].

When a negative voltage between source and gate is applied, the concentration of holes increases between source and drain. This so called conductive channel enables charged particles to flow from source to drain. By changing the source-gate voltage, the charge flow can be regulated as the channel changes in size.

In DEPFETs, an additional contact (bulk) is implanted that is connected to the wafer's back (back-contact). A high negative voltage on the bulk causes positive charges to be aspirated. If ionising particles pass the n-doped region and create electron-hole-pairs, the holes immediately drift to the bulk. The part underneath the gate is extremely high n-doped in DEPFETs. If electrons are created due to ionisation caused by passing particles, they are collected in a potential minimum, called the internal-gate. The small capacitance of the sensor has the advantage that data can be taken with a reduced noise level [11, p.79]. The holes that appeared in pair production, together with the aforementioned electrons, gather at the back contact. After the read-out process is finished, the electrons have to be removed from the internal gate. That is done by applying a highly positive voltage to an additional n-doped gate next to the so-called clear gate. This process "clears" the internal gate from signal and thermally generated electrons [11, p.79]. To avoid electrons storing at the clear contact instead of the internal gate during measuring processes, it is surrounded by a p-doped region.

2.3.3 Background Signals

Being so close to the interaction point, the PXD data is overlapped by a high amount of background. When talking about background, two major types have to be distinguished

- luminosity dependent and
- beam induced background.

Luminosity Dependent Background

Luminosity dependend background is caused by reactions of colliding electrons and positrons such as annihilation followed by pair production. The highest background source is the production of two photons which for their part produce another electron-positron pair each. Electron-positron scattering does not always lead to pair production within the PXD. In terms of *radiative Bhabha scattering* $n \in \mathbb{N}$ photons are produced under small angles. Their secondary particles might create signals in other parts of the detector. If one of the secondary particles changes its direction after scattering, it or its decay products can be seen in the PXD. In further discussions luminosity background will be neglected as a discussion of its effect will go beyond the scope of this thesis.

Beam-Induced Background

Beam background is not induced from collisisons between electrons and positrons, but from collisions inside the beam pipes[3]. Three major effects will be explained in the following:

- Intra-beam scattering and Touschek effect
- Beam-gas scattering
- Synchrotron radiation

The beams travelling trough the SuperKEKB rings have to be renewed periodically as they have limited lifetimes. The *beam-lifetime* is limited due to particles leaving the beam's trajectory or accelerator's spatial or momentum acceptance. Such particles are able to create showers when colliding with the beampipe wall or a beam-mask. Showers in the region around the interaction point are registered by the Belle II subdetectors and overlay intentional signals as background noise. One should note that background particles are able to pass the detectors multiple times as back-scattering is possible. Additionally, charged particles can be bent by the magnetic field in such a way that they pass the same location more than one time. These particles are called *curlers* [13, p.168]. Another observable effect is intra-bunch scattering of two electrons (or two positrons) within one beam bunch. These collisions are caused by oscillations of the particles in the beam pipe, vertical to their direction of propagation. The results are hadronic or electromagnetic showers visible in the PXD. If particles collide under small angles, the effect is called *intra-beam scattering.* Collisions under big angles lead to the *Touschek effect.* The energy exchange is then high enough to cause particles to fly off the bunch. Therefore, the Touschek effect leads to a shortened beam-lifetime. The Touschek effect is proportional to the squared number of particles per bunch [13, p.164]. For Belle II, it is especially important, as it increases with decreasing square of the beamsize and acceleration energies.

> Therefore, at SuperKEKB, the nano-beam scheme leads to a large Touschek scattering rate about 20 times larger for the LER and 28 times larger for the HER compared to KEKB. (Moll[p.164][13].)

Effects of Touschek-created electrons and positrons coming from LER can be seen especially in the forward region, at $\theta \approx 0°$ and $\phi \approx 172°$ to equal amounts on both layers. Effects concerning both PXD layers are common for showers. The location dependency is caused by the asymmetry of the collider [13, p.167]. HER effects are seemingly mirrored, but no proper conclusion could have been made due to low statistics in Moll's simulations. According to Moll, the dominant source of particles causing Touschek effect is the LER beam [13, p.168].

[3] The information in this abstract is taken from Moll's dissertation [13, p.160 and p.173]. Any other pages used from Moll's thesis are marked explicitly.

Although the beam pipes are aimed to be under vacuum conditions, there is never an ideal vacuum. Therefore collsions of particles with gas molecules - mostly H_2 and CO - are possible. This effect is called *beam-gas-scattering*. Like the Touschek effect, it is likely to cause particles to leave their trajectory or the acceptanance of the accellerator. If particles are back-scattered, the effect is caused by *elastic Coulomb scattering*. *Inelastic Bremsstrahlung* effects are also observable. Beam-gas scattering background is more intense in the outer layer of the PXD [13, p.177] and does not show any preferences in the θ-plane. Compared to Touschek effect, the amount of background signals caused by beam-gas scattering is small and therefore negligible.

Additionally, other background effects can be measured. For example, the storage of electrons and positrons in rings causes *synchrotron radiation*. Charged particles changing their direction of propagation are accelerated and emit electromagnetic radiation. This is the case for electrons and positrons whose trajectories are bent by the magnetic field. Due to the energy loss, the total energy of charged particles in the accelerator is limited. Most of the synchrotron radiation background is caused by HER beams [13, p.201]. Although the effect is not expected to reach the interaction point, the PXD is sligthly affected by this effect. Simulations have shown that the emitted photons are able to free charges in the PXD modules by photoeffect. This leads to the response of a very small amount of pixels in the PXD, which are forming tiny clusters [13, p.203].

> It is found that the dominant background is the two-photon QED process accounting for almost 70 % of the total PXD background. The second largest background is synchrotron radiation, closely followed by radiative Bhabha scattering. (Moll, [13, p.237])

2.3.4 Belle II Software Framefork

The Belle Software Framework 2 (`basf2`) is a data analysis and reconstruction environment based on C++ and Root I/O [9] that uses Python as an interface [1, p.32]. It hosts a data storage (`DataStore`) and `modules` that are processed one after another on `paths`. `basf2` is able to simulate Monte Carlo particle interactions with the whole detector using the simulation software Geant4. In this thesis no particle collisions have been simulated. Instead, the `basf2 module ParticleGun` has been used to simulate single particles that are shot from the interaction point into the detector. The simulated data sets can be processed in `basf2` like any real data set [1, p.32]. For PXD, data this means that the information measured in the pixel detector (charge, pixel position, layer, ladder) is reconstructed and clusters of pixels lying next to each other are created by the `Clusterizer module`. These clusters are then interpreted as marks left by one particular particle.

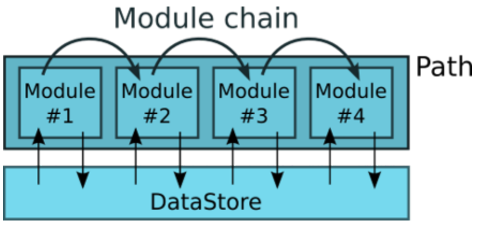


Figure 2.7: Schema of a module chain in the `basf2` environment. `Modules` are loaded into a `path` and processed depending on their order. Data is loaded from and stored in `DataStore`. Original picture taken from [9].

Clusterising Pixels in the PXD

As aforementioned, pixels are grouped into *clusters* by the `Clusterizer module`. The `Clusterizer module` module starts with a pixel in the first row of the (u,v)-layer and checks if its charge ratio is over the noise treshold. If it is, pixels that are direct neighbours of the respective pixel are tested. All pixels that are in direct neighbourhood and above the threshold will be united into a cluster. It is possible that some pixels in the new cluster's neighbourhood already belong to other clusters (see figure 2.8). If this is the case, the clusters will be merged. In this way, it is ensured that one pixel does not contribute to more than one cluster. The clusterisation process ends when the last pixel in the last row is reached [13, p.95].

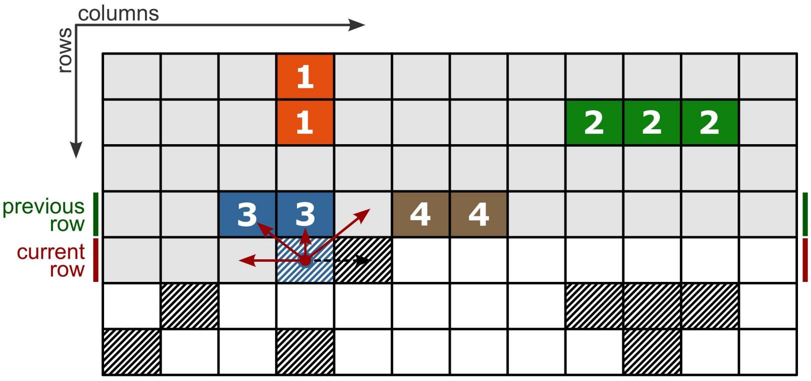


Figure 2.8: ”The PXD clustering method. Clusters 1, 2, 3 and 4 have already been found. The current pixel under investigation is added to cluster 3. The next pixel to the right will fill the gap between cluster 3 and 4. This will lead to clusters 3 and 4 being merged to a large cluster.” Picture and description taken from [13, p.95]

Chapter 3

Methods for Data Analysis

This chapter introduces *principal components analysis* and *artificial neural networks* as methods of data analysis used in this thesis. In both cases the focus is set on the comprehension needed to interpret the results presented in the next chapters. As the *principal components analysis* used in this thesis was self-implemented, an additional introduction to its mathematical background is given.

3.1 Principal Components Analysis

On the whole the idea behind principal component(s) analysis (PCA) is to approximate a number of statistical variables through linear combination of a smaller sized set of new variables. To visualise the principle, it is best to set an example. The measured data may have six different properties. This means that for each data point six different numbers are measured. Therefore, each data point can be seen as a vector in a six-dimensional space. The goal is to find a new coordinate system of lower dimension for the measured data, but to lose as little information as possible. The PCA uses the least squares method to find a straight. Fundamentally, a random straight is generated and a perpendicular drawn, that connects the straight to each data point. The straight's length is seen as the error. The idea is to minimise the sum of error squares. Additionally, it is required that the distance of data points along any straight are maximal. This leads to a maximisation of variance. After a straight is found, another straight is drawn so that both are at right angles to each other. Again, a minimisation of error squares and maximisation of variance follows. This procedure repeats iteratively until six straights are drawn. As these straights are orthogonal and normalised, they form the basis of a new six-dimensional room.

Mathematically, the steps mentioned above can be realised by calculating the *covariance matrix* or *correlation matrix* out of the data set. The eigenvectors of one of these matrices can be seen as columns of a transformation matrix. It is used to transform the measured data points into the new room. The new room's axes are called principal components. The percentage of each eigenvalue, to the total sum of eigenvalues represents how much information the accompanying principal component includes. If, for example, the first four dimensions already make up 90% of the total information, it might be useful to reduce the room's dimension to four. This can be done by keeping only the first four columns of the transformation matrix and therefore rotating the original data set into four-dimensional space.

3.1.1 Mathematical Basics

Definiton

PCA is a method of multivariate statistics. The following section will give a short overview about the mathematical basics, but does not claim to be complete, or perfectly mathematically correct. For further information like theorems and proofs see [14] and [15]. The following definition is taken from MARDIA's book *Multivariate Analysis* [14, p.214]. Note that in data analysis, $\boldsymbol{x}$ is usually a matrix of $n \times p$ variables with n being the number of individuals and p being the number of properties. It is required that n is much bigger than p [16, p.73].

> If $\boldsymbol{x}$ is a random vector with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, the the principal component transformation is the transformation
>
> $$\boldsymbol{x} \rightarrow \boldsymbol{y} = \boldsymbol{\Gamma}'(\boldsymbol{x} - \boldsymbol{\mu}) \tag{3.1}$$
>
> where $\boldsymbol{\Gamma}$ is orthogonal, $\boldsymbol{\Gamma}'\boldsymbol{\Sigma}\boldsymbol{\Gamma} = \boldsymbol{\Lambda}$ is diagonal and $\lambda_1 \geq \lambda_2 \geq .. \geq \lambda_p \geq 0$. The strict positivity of the eigenvalues λ_i is guaranteed if $\boldsymbol{\Sigma}$ is positive definite. [...] The ith principal component of $\boldsymbol{x}$ may be defined as the ith element of the vector $\boldsymbol{y}$, namely as
>
> $$y_i = \boldsymbol{\gamma}'_{(\boldsymbol{i})}(\boldsymbol{x} - \boldsymbol{\mu}). \tag{3.2}$$
>
> Here $\boldsymbol{\gamma}_{(\boldsymbol{i})}$ is the ith column of $\boldsymbol{\Gamma}$, and may be called the ith vector of principal components loadings. The function y_p may be called the last principal component of $\boldsymbol{x}$.

Further Definitions

To understand MARDIA's definition, terms like *covariance* - and *correlation matrix* have to be clarified. The following definitions are based on AHRENS' book *Mathematik* [17, p.1367-1374].

May $\{(x_i, y_i)|i = 1, ...n\}$ be a cloud of points and $\hat{x}$ and $\hat{y}$ the mean values.

- The *variance*

  $$var(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x})^2 \geq 0 \tag{3.3}$$

  is a measure of the spread of the variables x_i.

- The *empirical covariance coefficient*

  $$cov(x_i, y_i) = cov(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x})(y_i - \hat{y}) = \frac{1}{n}\sum_{i=1}^{n} x_i \cdot y_i - \hat{x}\hat{y} \tag{3.4}$$

  is an indication for correlations between $\boldsymbol{x}$ and $\boldsymbol{y}$. As $cov(\boldsymbol{x}, \boldsymbol{y})$ depends on the dimension of the space of the data set, it has to be normalised before interpreting it.

- The *empirical correlation coefficient*

  $$r(\boldsymbol{x}, \boldsymbol{y}) = cov(\boldsymbol{x}^*, \boldsymbol{y}^*) \tag{3.5}$$

  is a dimensionless, skale-invariant parameter with $x_i^* = \frac{x_i - \hat{x}}{\sqrt{var(x)}}$ and $y_i^* = \frac{y_i - \hat{y}}{\sqrt{var(y)}}$.

- A *maximal correlation* between $\boldsymbol{x}$ and $\boldsymbol{y}$ is equivalent to a linear correlation and occurs, if

  $$r(\boldsymbol{x}, \boldsymbol{y})^2 = 1. \tag{3.6}$$

- The data set is *uncorrelated*, if

  $$r(\boldsymbol{x}, \boldsymbol{y})^2 = 0. \tag{3.7}$$

  One must point out that *uncorrelated* should not be interpreted as missing "connection" between $\boldsymbol{x}$ and $\boldsymbol{y}$. It is still possible that $\boldsymbol{y}$ is a non-linear function of $\boldsymbol{x}$.

When considering m properties at once, it stands to reason that visualisations of any combinations of correlations should be made in matrix form.

- The *2-dimensional covariance matrix* is a $p \times p$-matrix that includes all varances and pair-wise covariances of a given data set. If $(\boldsymbol{x}, \boldsymbol{y})$ is two-dimensional, then the covariance matrix can be written as
$$\Sigma = \begin{pmatrix} var(\boldsymbol{x}) & cov(\boldsymbol{x}, \boldsymbol{y}) \\ cov(\boldsymbol{x}, \boldsymbol{y}) & var(\boldsymbol{y}) \end{pmatrix} \tag{3.8}$$

The data set used in this thesis is six-dimensional, thus a generalisation of the covariance matrix is needed for p-dimensional space. May $\boldsymbol{X}$ be a $m \times p$ matrix that can be written as $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, ...\boldsymbol{X}_p)$.

- The *m-dimensional covariance matrix*
$$\Sigma = \begin{pmatrix} var(\boldsymbol{X}_1) & cov(\boldsymbol{X}_1, \boldsymbol{X}_2) & \cdots & cov(\boldsymbol{X}_1, \boldsymbol{X}_m) \\ cov(\boldsymbol{X}_2, \boldsymbol{X}_1) & var(\boldsymbol{X}_2) & \cdots & cov(\boldsymbol{X}_2, \boldsymbol{X}_m) \\ \vdots & \vdots & \ddots & \vdots \\ cov(\boldsymbol{X}_m, \boldsymbol{X}_1) & cov(\boldsymbol{X}_m, \boldsymbol{X}_2) & \cdots & var(\boldsymbol{X}_m) \end{pmatrix} \tag{3.9}$$
is symmetrical and therefore positive semi-definite as the variance is always positive.

- The *n-dimensional correlation matrix*
$$R = \begin{pmatrix} r(\boldsymbol{X}_1, \boldsymbol{X}_1) & r(\boldsymbol{X}_1, \boldsymbol{X}_2) & \cdots & r(\boldsymbol{X}_1, \boldsymbol{X}_m) \\ r(\boldsymbol{X}_2, \boldsymbol{X}_1) & r(\boldsymbol{X}_2, \boldsymbol{X}_2) & \cdots & r(\boldsymbol{X}_2, \boldsymbol{X}_m) \\ \vdots & \vdots & \ddots & \vdots \\ r(\boldsymbol{X}_m, \boldsymbol{X}_1) & r(\boldsymbol{X}_m, \boldsymbol{X}_2) & \cdots & r(\boldsymbol{X}_m, \boldsymbol{X}_m) \end{pmatrix} \tag{3.10}$$
visualises the correlations between the components of $\boldsymbol{X}$. Note that $r(\boldsymbol{X}_i, \boldsymbol{X}_j) = 1$ for $i = j$.

Normalising

To prevent distortions, data should be normalised before calculating the covariance matrix. The normalisation in this thesis has been done using the python package *scikit* [18]. The following explanation is based on information taken from *scikit-learn.org* [19]. The class `sklearn.preprocessing.StandardScaler`(*copy=True, with_mean=True, with_std=True*) scales a data set $\boldsymbol{X}$ in the following way:
$$\boldsymbol{X}_{norm} = (\boldsymbol{X} - \boldsymbol{\mu})/\sigma \tag{3.11}$$
with σ being the standard deviation of the data set computed as
$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \hat{x})^2} \tag{3.12}$$
with the options

- `copy`: ”If False, try to avoid a copy and do inplace scaling instead. This is not guaranteed to always work inplace; e.g. if the data is not a NumPy array or scipy.sparse CSR matrix, a copy may still be returned.” [19]

- `with_mean`: If `False`, $\boldsymbol{\mu} := 0$.

- `with_std`: If `False`, $\boldsymbol{\sigma} := 1$

All three options are set `True` by default.

3.1.2 Choosing The New Coordinate System

Both $\boldsymbol{\Sigma}$ and $\boldsymbol{R}$ can be used for PCA. If $\boldsymbol{\Sigma}$ is used $\boldsymbol{X}$ should be normalised to prevent distortions. Firstly the eigenvalue problem needs to be solved. The eigenvalues of $\boldsymbol{\Sigma}$ written as rows of a matrix $\boldsymbol{\Gamma}$ enable the transformation of $\boldsymbol{\Sigma}$ into diagonal shape.

$$\boldsymbol{\Lambda} = \boldsymbol{\Gamma}^{\boldsymbol{T}}\boldsymbol{\Sigma}\boldsymbol{\Gamma} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_m \end{pmatrix} \tag{3.13}$$

Accordingly, the axes of the new coordinate system are the columns of Γ. They are called *principal components* and sorted by eigenvalues $\lambda_1 \geq \lambda_2 \geq .. \geq \lambda_m \geq 0$. Any cartesian vector $\boldsymbol{v_{kart}}$ can now be transformed into PC-space [20, p.85-89]:

$$\boldsymbol{v}_{PC} = \boldsymbol{\Gamma}\boldsymbol{v}_{kart} \Longleftrightarrow \boldsymbol{v}_{kart} = \boldsymbol{\Gamma}^T\boldsymbol{v}_{PC} \tag{3.14}$$

As $\boldsymbol{\Gamma}$ is orthogonal per construction, it does not change the length of the transformed vector [21, p.110].

If the variables highly correlate with each other, some variables contribute in a higher manner than others to the regression. It possible to reduce the dimension of the PC-space without losing much information. Still there is a risk to reject much information about a certain variable. Hence the values of eigenvectors should be compared to prevent unintentional loss of information. To calculate the number of dimensions needed to keep a certain amount of information, the eigenvalues are summed up both simple and cumulative.

$$sum_i = \frac{\lambda_i}{\sum_{i=1}^{m} \lambda_i} \tag{3.15}$$

$$cum_k = \sum_{j=1}^{k} sum_j \quad \text{with } k \leq m \tag{3.16}$$

There is no recipe for dimension reduction. However, some rules of thumb are taken from SEBER's book *Multivariate Observations* [15, p.175].

- Kaiser criterion: If the correlation matrix has been used, only eigenvalues greater than unity should be included.
- The cumulative sum of eigenvalues should reach at least 90%.

The author recommends to ”go for something in-between”. However it is helpful to visualise the percentage of variance explained by each eigenvalue in a *scree graph*. The visualisation helps to look for intervals in which the change of slope is small. In most cases it is appropriate to retain the components in the first interval.
After the new dimension k is chosen, a new matrix $\hat{\boldsymbol{\Gamma}}$ is defined. Its columns only contain the first k principal components. The reduced data set can be calculated with:

$$\boldsymbol{X}_{PC,k} = \hat{\boldsymbol{\Gamma}}\boldsymbol{X} \tag{3.17}$$

3.1.3 Interpretation of PCA Results

When the first two components include a high amount of information, it is useful to plot them for each data point as a *scatter diagram.* Such plots can give a first idea about the distribution of variables [14, p.227]. Additionally, an *h-plot* helps to find correlations between data properties.

> The lengths of the rays in the h-plots and the cosines of the angles between the rays will therefore be close to the respective standard derivations of the variables and the correlation coefficients. The axis of the plots are used only for constructing the plot and not for interpreting them. (SEBER, [15, p.211])

The h-plot might suggest a presence of several groups. Subsequently, it is advisable to name the groups depending on the main information they include. In this context, the entries of each eigenvector should be compared. The most important values in eigenvectors are those with values higher than 0.7 [15, p.191]. If possible values have absolute values higher than 0.3, they should not be ignored while interpreting [22, 122]. If one eigenvector value is highly positive and another one highly negative, the principal components represent the contrast between those variables [15, p.197]. Additionally, the correlation matrix should be examined. According to Jeffers [23] (cited in [15, p.192]) a large number of high values suggests a high degree of collinearity. Accordingly, very few basic properties of the system have been measured. Thus, it is suggested to measure new variables uncorrelated with the ones measured before.

3.1.4 Important Remarks

PCA presumes that any correlations in a given data set are linear. This will not usually be the case. If a data set generally distributes around a straight, but Euclidean distances between data points increase with slope, the data set can be logarithmised. Taking the natural logarithmic usually adapts a data set like this to normal distribution [22, p.119]. If the data set cannot be prepared so that it is linear, correlated distortions like the *horseshoe effect* (see fig.3.1) can be observed. For further details on this effect see [24]. The worst case scenario will be that relative distances are distorted and additionally the relative order of objects is incorrect [22, p.120]. Therefore, Leyer suggests to use PCA only with "relatively homogenous data sets" [22, p.122].

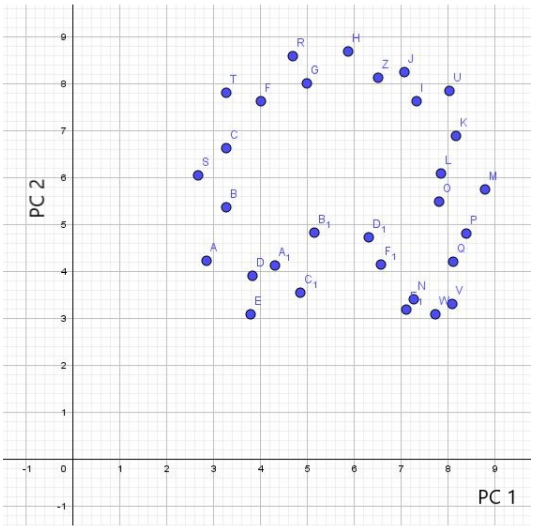


Figure 3.1: Example for horseshoe effect in PC-transformed data. The typical horseshoe-like bending is clearly visible. The letters next to the data points are of no interest for this thesis.

One must take into consideration that PCA is sensitive to outliners. This is due to the fact that the mean is used for calculationg the (co-)variance. The effect can be reduced by replacing the mean by the *median.*

The *median* of a sorted data set $x_{(1)}, x_{(2)}, ..., x_{(n)}$ is defined as

$$x_{med} = \begin{cases} x_{\frac{n+1}{2}} & \text{if } n \text{ is even} \\ \frac{1}{2}(x_{\frac{n}{2}} + x_{\frac{n+1}{2}}) & \text{if } n \text{ is uneven} \end{cases} \tag{3.18}$$

It is called *robust* as it is influenced less, the more a data point differs from the others [17, p.1351-1352].

3.2 Artificial Neural Networks

The mammalian brain consists of several different brain cells grouped in *neurons* and *glia cells*. Among these cells, neurons are said to be responsible for a transfer of information through electrical signals. Though meanwhile it is known that glia cells are essential for proper brain function, the attempts of constructing artificial brains only focus on neurons.
Biological neurons form networks. The connection point of two neurons is called the *synapse*. Depending on the signal and type of synapse, neurons fire chemical substances that can be either inhibitory or stimulating. Depending on the chemical substances, positive or negative electrical signals are transferred. The electrical charges are summed up; however, the neuron does not always "react". If a certain threshold is not crossed, the signal flux ends at the respective neuron.
ANNs try to copy the function of biological neurons. Different types of *artificial neural netorks* (ANNs) have been implemented, but the main idea of connecting neurons stays the same. Depending on the training of neural networks two main types have to be distinguished:

- Supervised learning, for example in feedforward neural networks (FFNNs).
- Unsupervised learning, like in self-organising maps (SOMs).

In this section only a brief overview of FFNNs and SOMs, with a focus on comprehension will be given. Further details will be omitted to prevent duplications with Katharina Dort's *Supporting Information* [25].

3.2.1 Feed Forward Neural Networks

FFNNs consist of different layers of neurons. Between two layers all neurons are connected. Each connection is weighted depending on how strongly the neurons should interact with each other. FFNNs learn by adapting the connection weights. This can be done by providing a training data set. The FFNN compares the difference of its own response with the supposed outcome of the training set and adapts its weights if necessary [26, p.67]. This principle is called error correction learning. FFNNs are of no further interest for this thesis.

3.2.2 Self-Organising Maps

The main differences between SOMs and FFNNs are

- SOMs do not use a training data set that predicts the output.
- Neurons in SOMs learn by competing with each other not by error correction.

The idea of imitating the mammalian brain is driven further by the adaption of *mapping*. Brain cells receive multidimensional input from several different organs. Observations showed that different brain regions respond to different signals. Similar signals are analysed in the same brain region. It is possible to visualise the functional brain regions as a two- or three-dimensional map (figure 3.2). High dimensional information of sensory perceptions from inside and outside the body is therefore mapped into lower dimensional space. Analogous SOMs are often implemented to create two dimensional maps of a data set. For this reason SOM algorithms are discretising dimension reduction methods and can be used for cluster analysis [27, p.452].

To obtain a map that clusters data of a particular topic in particular local regions, the neuronal network has to be trained. Any implementation of neural networks includes parameters that have to be selected before training. Depending on the implementation these can be the number of neurons, number of iterations, learning rate, neighbourhood function and its parameters and dimension of the plotted map. If parameters are not chosen correctly, results might either be inaccurate, over-fitted or show unexpected convergence behaviour [29, p.2].

Competitive Learning

A set of neurons is created with random weights. The algorithm starts by choosing a random input vector of the training set and comparing it to the weight vectors of all neurons. The winning neuron is

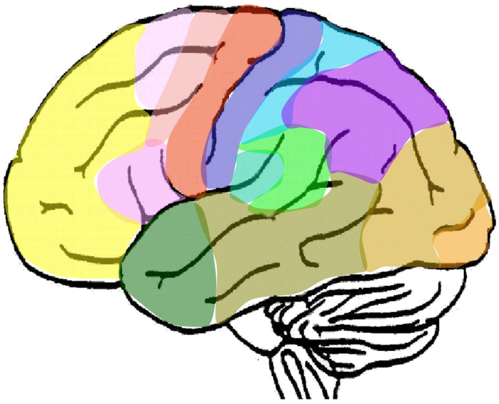


Figure 3.2: Schematic two-dimensional drawing of the human brain's functional areas [28]. Note that this picture only shows a lateral view of the brain.

the one whose weight vector has the highest similarity with the input vector. For reasonable results, it is necessary to assure that weight and input vectors have the same dimensionality [27, p.449]. For comparison of that weight and input vectors the *Euklidian Distance* is used in most implementations.

The next step is to update the winning neuron's weights so that its value is closer to the input vector's value. A neighbourhood function is chosen that affects other neurons surrounding the winning neuron. The number of neurons whose weights are changed can be altered by the parameter σ. If σ is chosen to be small, only the winning neuron is updated. Otherwise, all other neurons' weights will be changed depending on their distance to the winning neuron. Larger distances lead to smaller updates. The algorithm repeats those steps, choosing another input vector and altering weights until all input vectors have been compared to the network.

During the process, the movement of neurons' weight vectors changes the network's geometry. The final result will be a map with different coloured regions representing clusters of data.

In this thesis, SOMs are used to create maps that are able to distinguish particles from background. These pre-trained maps will be used in the future to analyse datasets created in Belle II measurements.

MiniSom Implementation

Several pre-implemented versions of self-organising maps exist that are ready to use. This thesis uses the Python MiniSom implementation that has already been used in [1]. The code is based on a tutorial by Giuseppe Vettigli that can be found on GitHub [30]. The hyperparameters of MiniSom that have to be set manually before training are: number of neurons, neighbourhood function, sigma, learning rate and the number of iterations. In this thesis, the parameters were set as shown in table 3.1. The neighbourhood function has four options: *Gaussian*, *Mexican hat*, *triangle* and *bubble*. The Gaussian function already proved to be best in the predecessor of this thesis [1].

Table 3.1: MiniSom hyperparameter values used in this thesis.

Parameter	Value
Number of neurons	10×10
Neighbourhood function	Gaussian
Sigma	7
Learning rate	0.01
Number of iterations	200 000

SOM Quality

A possible quality measure for classification is the *Receiver Operating Characteristic* (ROC) curve. The following explanation is an analogous translation of ERTEL's explanation [31, p.155]. A ROC curve is calculated to simplify the visualisation of the outcome and ensure comparability with similar studies. It is generated by plotting *sensitivity* and *specifity* against each other.

- *Sensitivity* measures the relative proportion of positive cases that are recognised properly.

$$P(\text{classified positive} \mid \text{positive}) = \frac{\mid \text{positive and classified positive} \mid}{\mid \text{positive} \mid} \tag{3.19}$$

- *Specifity* measures the relative proportion of negative cases that are recognised properly.

$$P(\text{classified negative} \mid \text{negative}) = \frac{\mid \text{negative and classified negative} \mid}{\mid \text{negative} \mid} \tag{3.20}$$

As in this thesis ROC curves will be applied to SOMs that classify between beam background and signal, one might interpret negative classification as *signal-like* or *efficiency* and positive classification as *background-like* or *background rejection.* This leads to the curve shown in figure 3.3.

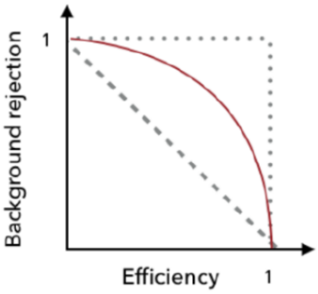


Figure 3.3: Exemplary representation of a ROC curve. The box function (dashed) can be interpreted as perfect classification, whereas the linear function (dotted) depicts the worst case. Real classification results will be somewhere in-between (red). Graphic taken with permission from [1, p.41].

To rate the quality of training, it is suggested to test different sized SOMs in questions of mapping precision, topology preservation and a combination of both. This can be done by calculating the *Best Matching Unit* (BMU), i.e. the weight vector whose Euclidean distance to the input vector is the smallest. In QUINTANA's paper, the following was done:

> 1) The mapping precision was measured using average quantization error between data vectors and their BMUs on the map.
> 2) The topological representation accuracy was measured as the percentage of data vectors for which the first- and second-BMUs are not adjacent units.
> 3) The average 'combined' error over all input vectors was calculated from the sum of quantization error [...] and topographic error [. . .]. (QUINTANA, [29, p.6])

These three points were not tested in this thesis, as implementing these is an extensive task. It is advisable to include them in further research on the use of SOMs for PXD data processing.

3.3 Summary

This chapter inroduced both principal components (PCA) analysis and self-organising neural networks. Principal components analysis allows the finding of correlations between properties of a data set. It can be understood as a transformation into a space where the axes are parallel to the directions that are containing the most information. By taking a smaller subset of axes in the PCA-space one can reduce the problem's dimension.

Self-organising-neural networks (SOMs) are used to map a high-dimensional problem onto lower dimensional space, while keeping the topography of the problem. This is done by adjusting the weights between nodes of the network after comparison with input vectors. The trained network will be able to differentiate between different data types as nodes in different parts of the map respond to input vectors of different sources. The success of the classification process can be visualised by ROC-curves.

Chapter 4

Project1: Multiparameter Analysis of Antideuterons

The discovery of new particles requires a deeper understanding of particle behaviour in the Belle II detector. In order to evaluate the measured data of the pixel detector correctly, it is necessary to improve the interpretation of pixel clusters. For this reason, the correlations between six cluster properties *charge*, *minimum charge*, *seed*, *size*, *size in u* and *size in v* will be examined in this chapter. Since it is possible to produce both antideuterons and multiquark states at SuperKEKB, antideuterons are chosen for data analysis in this section. At Belle II experiment, antideuterons are observed to take a further step towards understanding the principles of dark matter. As explained in DORT's thesis [1], cosmic ray antideuterons are produced in dark matter decays. This production is reconstructed at Belle II. An evaluation of simulations can be found in [1]. Additionally, antideuterons resemble the tetraquark state $\Upsilon(3882)$ in terms of electrical charge and therefore physical behaviour of a $\Upsilon(3882)$ can be approximated in first order by a modified antideuteron. In simulations, a simplified tetraquark can therefore be created by changing the mass of antideuterons and keeping any other properties.

The study of cluster property correlations will start with a short overview of the generation of the data set and afterwards plots of different cluster properties will be discussed. Thirdly, the results of a principal components analysis will be presented. The chapter closes by a short physical interpretation of the results.

4.1 Generation of Data Set

Other than in DORT's Master thesis [1] antideuteron events have been created with the `basf2 module ParticleGun`. The Python script responsible for the simulation is called *anti_deuterons.py*. Apart from `ParticleGun`, it loads several other `basf2 modules` including the `PXDClusterizer`. Several parameters have to be set in the script. Per call of *anti_deuterons.py*, 2000 antideuteron events with a uniform momentum distribution between 0.05 and 1 GeV, are created.

The flowchart (figure 4.1) visualises the data generation and analysis process of project 1. To generate the events of the antideuterons data set, the file has been called parallel 20 times to minimise runtime. The output files have been merged into *Antideuterons_run_all.root*. This file includes any measurable information of 40 000 events. The Python script *sim_to_clsprop_modinfo_dd.py* has been used to write cluster properties into the file *Beam_dd_cluster_sim.root*. The file *Beam_dd_cluster_sim.root* was then converted into a txt-file to simplify data processing with Python.
The table 4.1 is taken from DORT's Master thesis [1, p.36]. It shows the cluster properties used as input vectors for the neural networks in the thesis mentioned above. The same cluster properties have been used in the present case as input of the PCA. Any cluster property that has not been used is marked with a cross.

Table 4.1: Table and description taken from [1, p.37]. Note that some contents have been altered. ”Cluster properties used in this [and Dort's] thesis. The second column marks whether a property is already computed in the `Clusterizer`. The third column indicates which properties are chosen to be part of the input vector for the neural network[s].”

Property	Computed by `Clusterizer`	Part of input vector
Total cluster charge	✓	✓
Cluster seed charge	✓	✓
Cluster minimum charge	✗	✓
Total cluster size	✓	✓
Total cluster length	✗	✗
Cluster size in u	✓	✓
Cluster size in v	✓	✓
Pixel coordinates in u	✗	✗
Pixel coordinates in v	✗	✗
Cluster angle	✗	✗
Cluster eccentricity	✗	✗

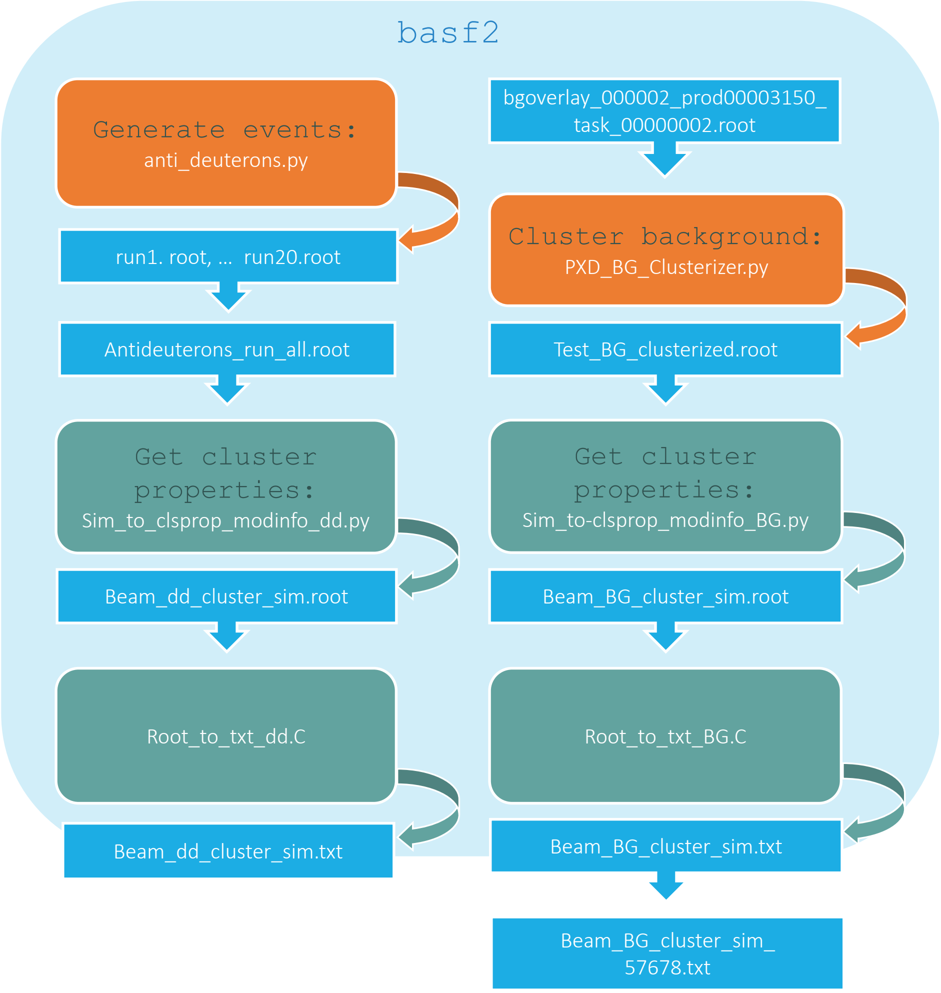


Figure 4.1: Flowchart of data generation. The antideuterons data set has been generated using *anti_deuterons.py*, whereas the background data is taken from KEKCC and had to be clusterised by *PXD_BG_Clusterizer.py* before further processing. Afterwards cluster-properties have been written into *Beam_dd_cluster_sim.root* and *Beam_BG_cluster_sim.root*.

In case of the beam background, the events have not been generated by `Particle Gun`. The background data set *bgoverlay_000002_prod00003150_ task_00000002.root* can be found on KEKCC as one of the simulated *Official beam background samples*. It is one of more than 300 files. Note that for optimal results all files should be used as an input which leads to very long runtimes due to the data size (several GB). As this thesis was not calculated on a mainframe computer, only one file has been chosen as beam background. For further processing clusterisation was required. *PXD_BG_Clusterizer.py* uses the `PXDClusterizer module` to write the clustered data into *Test_BG_clusterized.root*. All other steps are equal to the ones mentioned above. Both *Beam_dd_cluster_sim.root* and *Beam_BG_cluster_sim.root* consist of six columns. Each row represents a cluster with the properties *charge*, *minimum charge*, *seed*, *size*, *size in u* and *size in v*. To prevent distortions in either antideuteron or background direction, both txt-files need to have the same number of rows. After shortening *Beam_BG _cluster_sim.root* to 57678 rows the data sets *Beam_dd_cluster_sim.root* and *Beam_BG_cluster_sim_57678.root* are ready be read into the SOM or PCA files.

4.2 Antideuterons and Background Data Set

The antideuterons data set *Beam_dd_cluster_sim.txt* consists of 57678 clusters. For each cluster the properties *charge*, *minimum charge*, *seed*, *size*, *size in u* and *size in v* are listed. To gain a better understanding of the properties and their correlations, plots of different combinations will be compared in this section. In each plot antideuteron and background data are plotted on top of each other so that the background distribution might be overlapped in some figures. Some plots will be used in the next section to interpret the principal components by finding similarities between PCA results and the original data set.

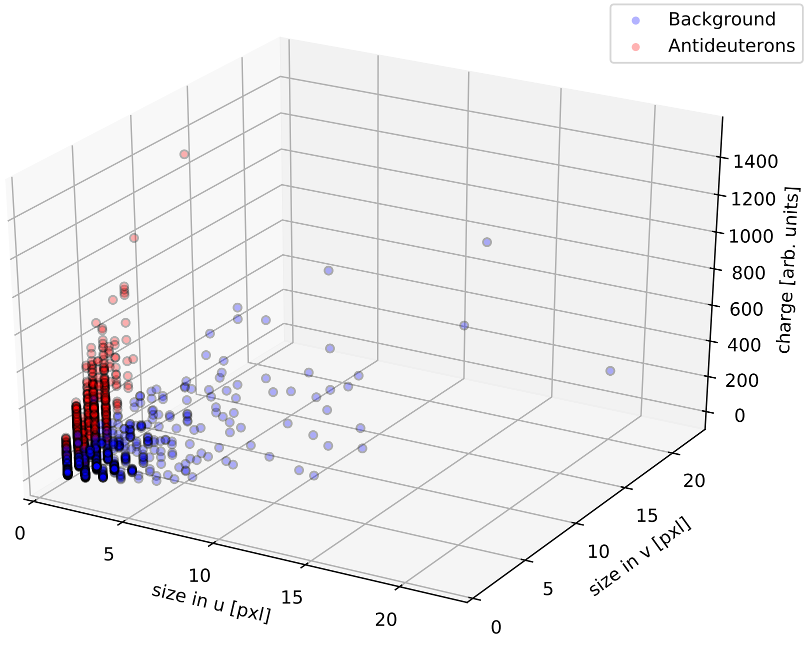


Figure 4.2: Cluster size in u and v depending on cluster charge (original data set). The antideuterons (red) and background clusters' (blue) sizes are enlarged in different directions due to the collider's asymmetrie. The antideuterons' boost in v-direction is cleary visible. In both cases cluster charge increases with cluster size.

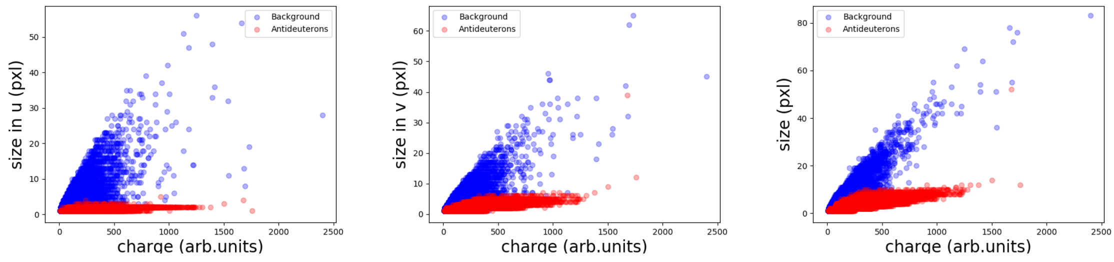
As the whole data set is a 115 356 × 6 matrix, it cannot be plotted three-dimensionally. Nevertheless, an attempt of visualising the data set can be made by plotting only two or three properties. By definition the information of minimum charge and seed is included in charge, as well as the information of size in u and v is contained in size. It can be speculated that size and charge are the most relevant properties. On the other hand, size does not include information about the spatial geometry of clusters. Figure 4.2 shows how cluster charge and the cluster's geometry in u and v are connected. Clusters belonging to antideuterons are significantly larger in v-direction than background clusters. In both cases it can be observed that cluster charge increases in a non-linear manner with cluster size. Note that this does not mean that small clusters are always low charged. The slope between size and charge seems to vary stronger for background than antideuterons. This can be explained by the fact that background clusters host a composition of clusters generated by different particles. antideuteron clusters apparently tend to keep a smaller size at high charges (Figure 4.3c). The correlation between charge and size in u or v seems to be almost the same irrespected of the choice of direction (Figure 4.3a+b). The slope in antideuterons' distribution shows a slightly stronger rise in v-direction. However, antideuterons and background clusters smaller than 5 pxl × 5 pxl predominate.

Figure 4.4 shows that the minimal charged pixel in an antideuteron's cluster (minimum charge) underlies great variations. The highest charges can be observed in single pixel clusters. In v-direction clusters with high minimum charges tend to be slightly longer than in u-direction due to the asymmetry of the collider. In contrast background clusters accumulate at lower charges. All clusters longer than 15 pxl in one direction show minimum charges lower than 25 (arbitrary units).

In a similar way, cluster size depending on seed differs for antideuterons and background. The background distribution shown in Figure 4.5 indicates high variations in seed with a tendency to accumulate around 30. Seeds between 200 and 250 are associated with overall cluster sizes smaller than 10. The deuterons data set shows a kindred distribution to the one plotted in Figure 4.4. In v-direction, seed charges higher than 50 prevail, whereas in u-direction, seed charges between 25 and 255 are evenly dispersed.

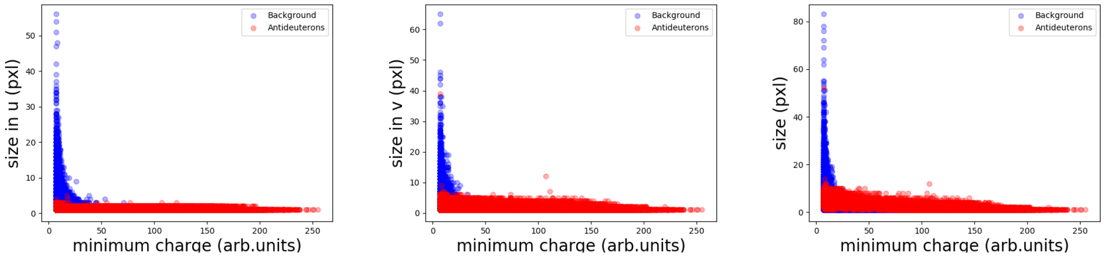
Figure 4.6 shows the three different charge-related properties charge, seed and minimum charge plotted in pairs against each other. Note that all three plots show a straigth line that splits the figure in two halfes. One half is filled with data points, the other is not. This line arises from the fact that in single pixel clusters seed, minimum charge and charge have the same values. Clusters with equally charged pixels can also contribute to the straigth line in 4.6b, but overall charge will vary depending on cluster size. This way the claw-like shape in 4.6a can be explained. Equally charged clusters show overall charges that are integer multiples of minimum charge. 4.6c shows that background clusters have higher general charges at lower seeds due to the possibility of overall bigger cluster sizes.

The antideuteron clusters show a smaller variety in cluster size than in background clusters as visible in figure 4.7.



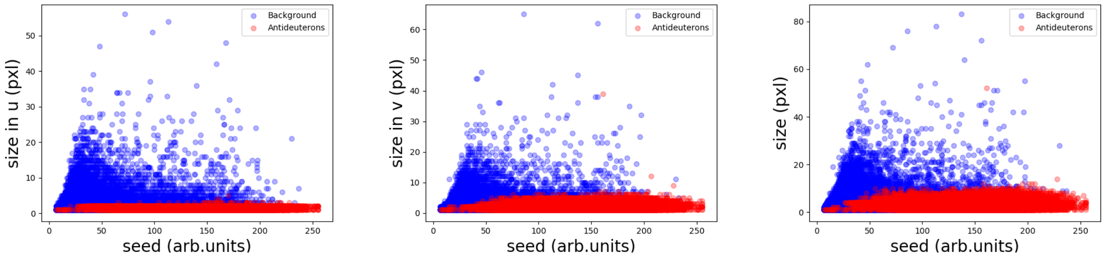
(a) (b) (c)

Figure 4.3: Distribution of overall cluster size in u (a), size in v (b) and overall cluster size (c) depending on charge. The correlations are seemingly very similar. The slope of the background clusters's distribution (blue) is generally higher than for antideuterons (red).



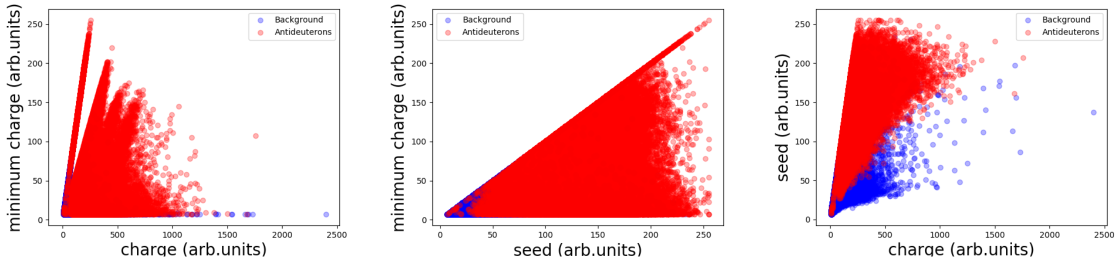
(a) (b) (c)

Figure 4.4: Size in u (a), size in v (b) and overall size (c) depending on minimum charge. For antideuterons (red) high minimum charges appear in clusters with lengths smaller than 10 pxl. Giant background clusters (blue) larger than 15 pxl in one direction accumulate around charges lower than 5 pxl.



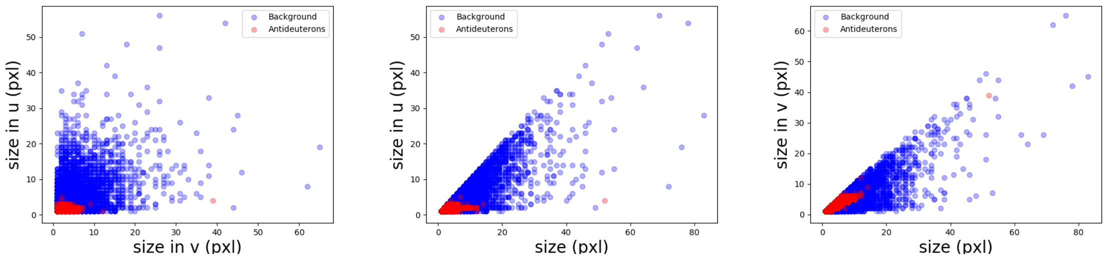
(a) (b) (c)

Figure 4.5: Size in u (a), size in v (b) and overall size (c) depending on seed. For antideuterons (red) most seed charges are higher than 50, whereas background seed charges prevail at charges smaller than 50.



(a) (b) (c)

Figure 4.6: Charge-related distributions plotted against each other. (a) Minimum charge depending on charge. A claw-like structure due to clusters of equally charged pixels is visible. Antideuteron (red) and background (blue) are perfectly overlapping. (b) Minimum charge plotted against seed. Note the straigth line coming from single pixel or equally charged clusters. (c) Seed depending on charge. Background clusters show a tendence for lower seed charges at higher general charges.



(a) (b) (c)

Figure 4.7: Size-related properties plotted against each other. (a) Size in u against size in v. Antideuteron clusters (red) show a tendency to be longer in v-direction. Background clusters (blue) seemingly have no prefered geometry.

4.3 PCA Results

In this section the results of the principal components analysis of the antideuterons and background data set is presented. The implementation of the PCA is based on a tutorial by Raschka [32] on *plot.ly*. Summed up 115 356 rows from *Beam_dd_cluster_sim.txt Beam_BG_cluster_sim.txt* have been read in as numpy array. As a first step in this analysis, the correlation matrix and its eigenvectors will be discussed. Although the goal of this analysis is to observe the relations of cluster properties to each other in more depth, an initial attempt will be made to reduce the dimension of the six-dimensional system to simplify further data processing.

4.3.1 Interpretation of Correlation Matrix

Table 4.2 gives the lower half of the correlation matrix. High correlations (> 0.7) can be found between the following tupels: (seed, charge), (size in u, size), (size in v, size). These correlations have been expected, as size in u and v affect the overall size per definition. The same goes for seed and minimum charge with regard to overall charge. As the minimum charge is smaller than the other cluster's charges, it is likely to contribute less to the overall charge. Furthermore, considerably smaller correlations ($0.4 - 0.5$) exist between the tupels (size, charge), (size in v, charge) and (size in u, size in v). Almost uncorrelated are (size, seed) and (size in v, seed). Since the collinearity between cluster properties is not very high in general, it is mathematically not required to extend the data set with further properties.

Table 4.2: Lower half of correlation matrix

Cluster property	Charge	Min. Charge	Seed	Size	Size in u	Size in v
Charge	1.					
Min. Charge	0.2233	1.				
Seed	0.7854	0.4771	1.			
Size	0.4617	-0.2882	0.0392	1.		
Size in u	0.1596	-0.2600	-0.1399	0.8044	1.	
Size in v	0.4144	-0.2091	0.0546	0.8414	0.4627	1.

4.3.2 Interpretation of Principal Components

The whole set of corresponding eigenvectors of the correlation matrix (e.g. the principal components) is given in table 4.3. One should mention that the data set has been scaled so that the eigenvectors have unit length. As mentioned in chapter 3 the entries of the eigenvectors qualify the contribution of each cluster property to the respective principal component.

Table 4.3: Eigenvectors of correlation matrix

Cluster property	PC1	PC2	PC3	PC4	PC5	PC6
Charge	-0.3570	-0.5135	0.2299	-0.6695	0.2713	-0.1848
Min. charge	0.1544	-0.5031	-0.0498	-0.1217	-0.7524	0.3738
Seed	-0.1121	-0.6500	-0.0929	0.6722	0.1737	-0.2725
Size	-0.5927	0.0924	-0.7892	-0.0769	-0.09923	0.0393
Size in u	-0.4626	0.2251	0.4042	0.1317	-0.5455	-0.5071
Size in v	-0.5205	0.0377	0.3870	0.2489	0.1510	0.7022

The first principal component is domintated by the weights of the cluster properties size, size in u and size in v as well as charge. The second principal component mostly represents the three charge related-properties. It should also be noted that all size-related properties are weighted with opposite sign in PC2. As the weights of size in v and overall size are negligibly small, this PC might be interpreted as a juxtaposition of charge-related properties and size in u. PC3 is highly correlated to size and seems to show the difference between size and the positively weighted parameters charge, size in v and size in u. PC4 is connected to charge and seed, contrasting it to size in u and v. PC5 shows highly negative weights of both minimum charge and size in u. PC6 is dominated by positive weights of size in v and minimum charge and negative weights for size in u and seed. It is not possible to make any certain statements concerning the overall result of the principal components analysis. However, one might try to simplify the observations made above by making some rough interpretations:

1. PC1 can be called a ”measure for size”. It also represents that higher overall charges are more probable in big clusters.

2. PC2 can be called a ”measure for charge”. It also includes the information that the size in u tends to be smaller for overall highly charged clusters.

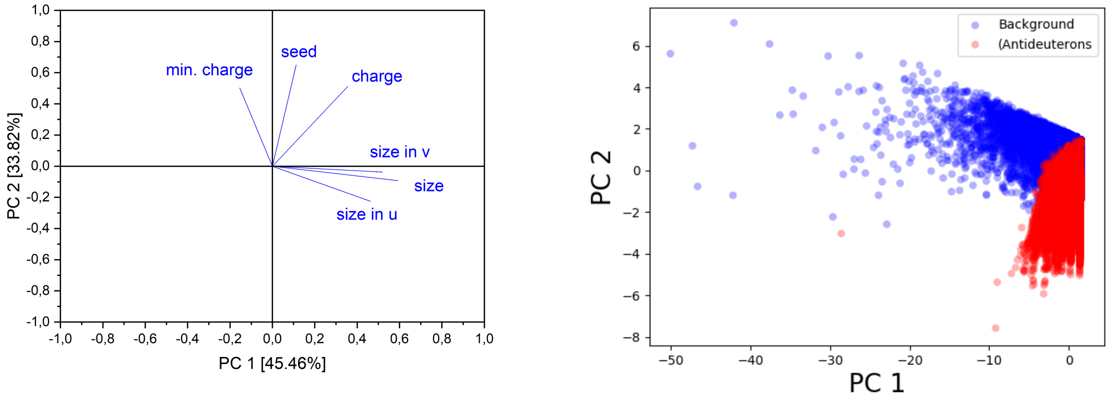
3. According to PC3, clusters that are small in overall size tend to be almost equal in u- and v-size. This is the case for single pixel and small squared clusters.

4. PC4 indicates that an overall highly charged cluster has smaller seed charges.

5. PC5 shows that small minimum charges can be found in short clusters in u-direction. If it is overall small-sized, a slight increase in charge is observable.

6. PC6 visualises that for long clusters in v-direction minimum charge increases while showing a decrease in u-size and seed.

The clusterisation of properties in the two groups ”size-related” and ”charge-related” is also visible in the h-plot (shown in figure 4.8 a). The overall cluster charge seems to have a stronger correlation to the size-related properties than the other charge-related properties. When plotting the transformed data points into two-dimensional PC-space, a spatial separation of antideuterons and background is visible. Although the origin of the distribution is the same for both of them, antideuterons distribute more strongly along the second principal component.



(a) H-plot

(b) Scatter graph

Figure 4.8: (a) The h-plot shows projections of all cluster properties onto the first two principal components. The length of vectors represents the weight. Two groups are visible: properties depending on charge (charge, min. charge, seed), properties depending on size (size, size in u, size in v). (b) Data points transformed onto the first two dimensions of the PC-space. Antideuterons (red) and beam background (blue) are distributed in different directions.

4.3.3 Dimensionality Reduction

As it was mentioned before, PCA can be used for dimension reduction, as not every principal component holds the same amount of information. The eigenvalues and their explained variances in percent are listed in table 4.5. The first column shows that only the first two eigenvalues have values higher than 1. The second column lists the explained variance of the eigenvalue in percent. It is interpreted as information included in the respective principal component. To evaluate the total amount of information included in a set of principal components, the cumulative sum of the second row has been calculated in the third column. The first two principal components already contain almost 80% of the total information. Applying the methods of dimensionality reduction introduced in capter 3 to the problem leads to the results presented in table 4.4. The mean of the results would lead to a reduction onto three dimensions. This way almost 90% of the information could be kept. Plotted into 3-dimensional space (figure 4.10) the PC-tranformed data set resembles of figure 4.2. The straight line of data points that has been visible in several other plots in the section above is also visible in figure 4.10. In comparison with figure 4.4 the data points seem to be more compressed with constant distances in bewteen. This might suggest that the transformation had a pretty similar effect on the single pixel clusters of the data set as taking the natural logarithm of each cluster property's values.

Table 4.4: Results of different attempts of dimensionality reduction

Rule	Reduced Dimension
Only eigenvalues greater than unity should be included.	2
The cumulative sum of eigenvalues should reach at least 90.%	4
Interpretation of scree graph.	2 or 4

The Use of High-dimensional Plots for Further Interpretation

The attempt of a 4-dimensional visualisation lead to figure 4.11 and figure 4.12. For both antideuterons and background, an arrow-head like shape is visible, as well as a straight line of data points at the edges. As the fourth dimension is hard to interpret, one might definitely choose three dimensions for further interpretation and data processing.

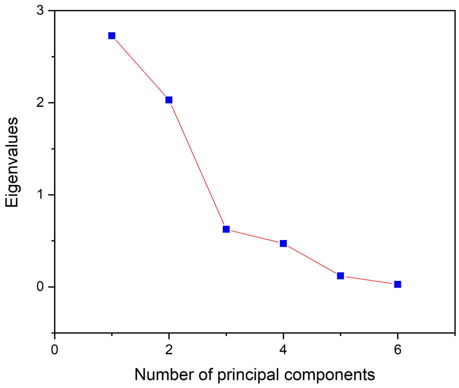
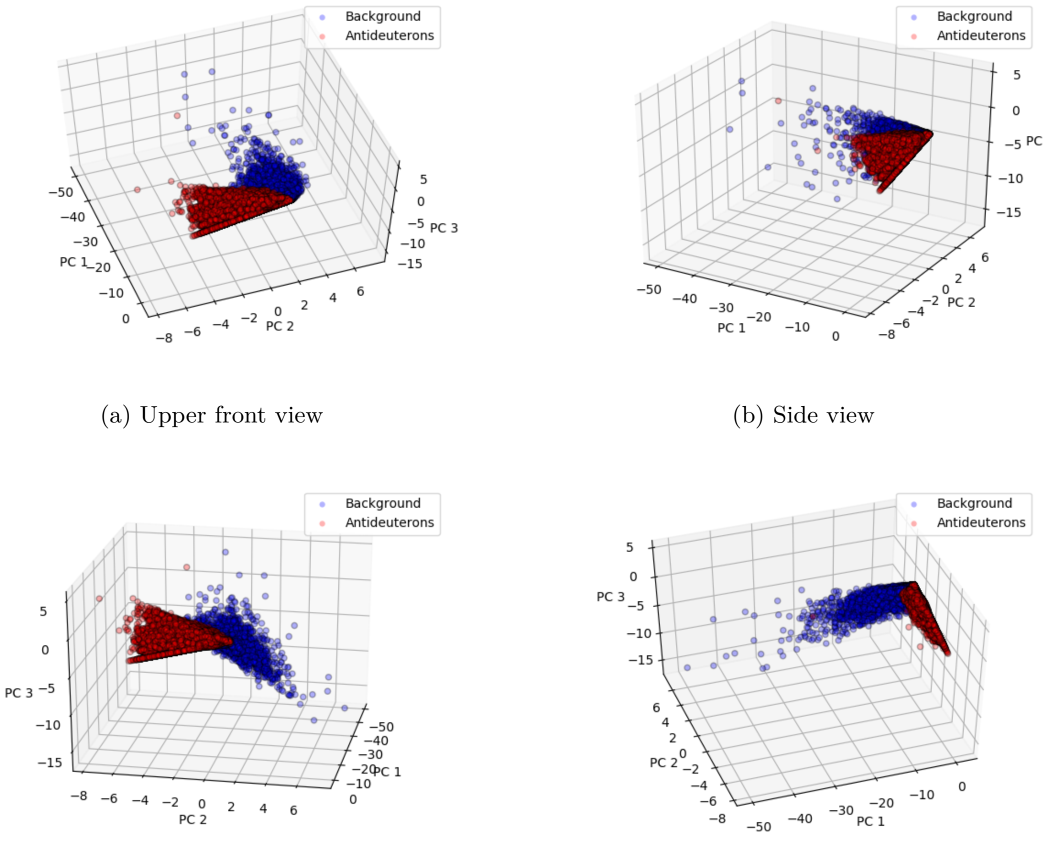


Figure 4.9: Scree graph, showing eigenvalues and corresponding principal components. Main changes in slope can be observed at PC3 and PC5.

Table 4.5: Eigenvalues and corresponding explained variances in percent

Eigenvalues	Sum [%]	Cum. Sum [%]
2.73	45.46	45.46
2.03	33.82	79.28
0.62	10.41	89.69
0.47	7.86	97.55
0.12	1.99	99.54
0.03	0.46	100



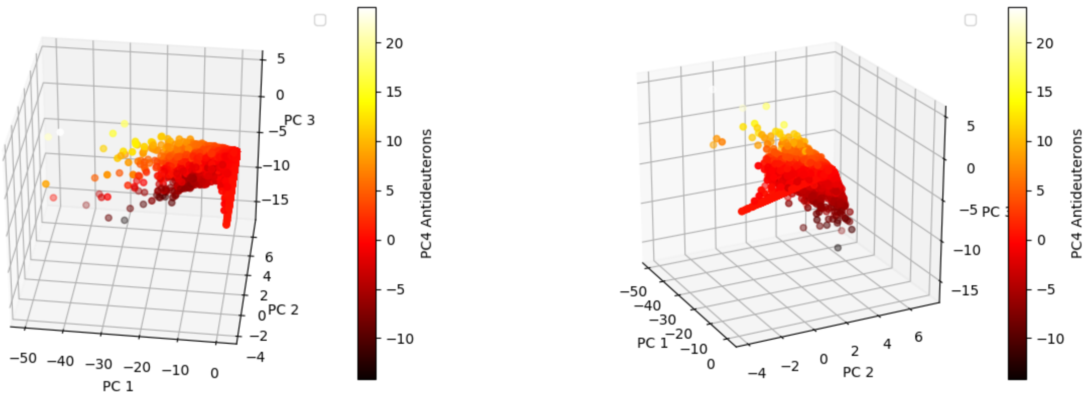
(a) Upper front view

(b) Side view

(c) Front view

(d) Lateral backside view

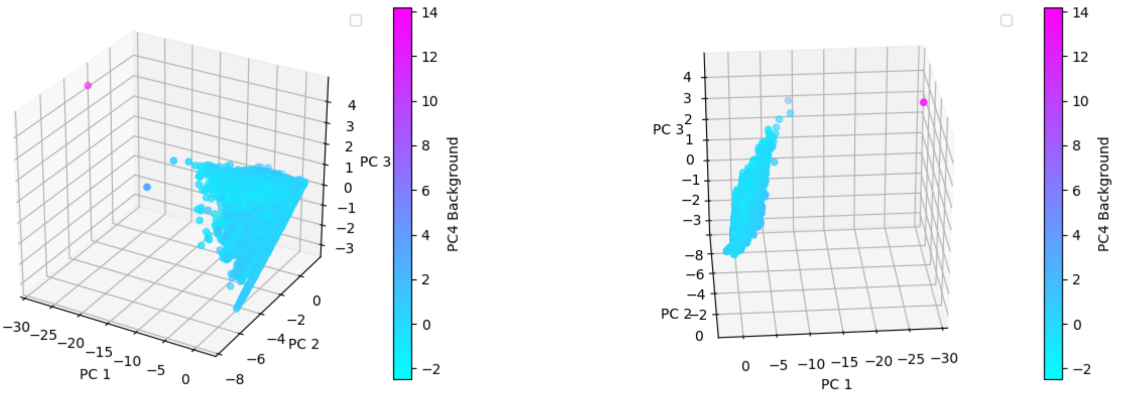
Figure 4.10: 3-dimensional plot of PC-transformed data shown from different angles. Antideuterons (red) and background (blue) seem to form two wings. The background-wing is longer along PC 1 and wider distributed in PC 3-direction. Note the straight line showing at the front coming from both antideuterons and background (visually overlapped).



(a)

(b)

Figure 4.11: 4-dimensional plot of PC-transformed antideuteron data (without background) seen from different angles. PC 4 increases at the centered ”tail” of the distribution. Note the straight line of data points on the left-side of the arrowhead-shaped structure.



(a)

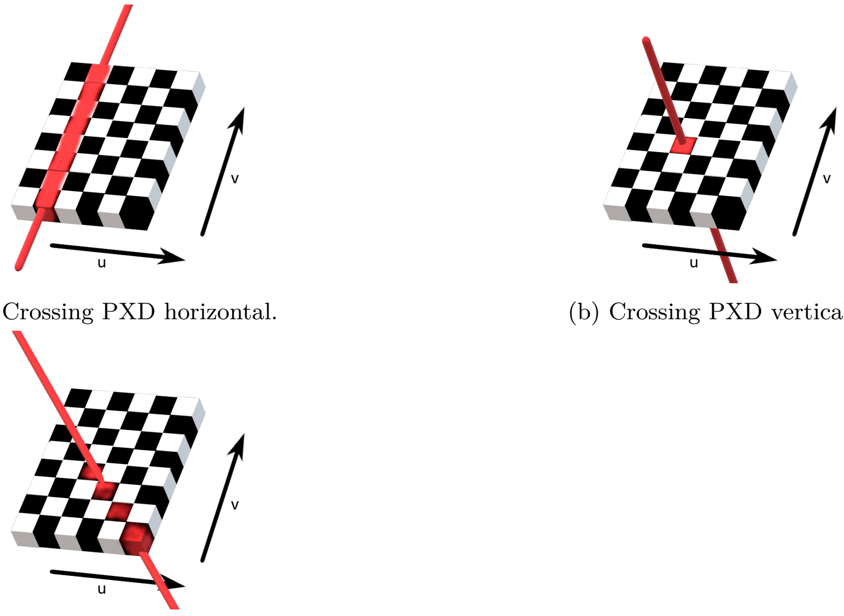
(b)

Figure 4.12: 4-dimensional plot of PC-transformed background data (without antideuterons) seen from different angles. PC 4 is seemingly evenly distributed due to an outlier (pink dot) that influences the PC4-axe’s scaling.

4.4 Physical Conclusion on Multiparameter Analysis

The interpretation of PCA results confirmed observations that had already been made by plotting the original cluster properties against each other in section 4.2. In the following abstract a physical hypothesis on the observations is proposed.

Cluster shapes depend on the particle's trajectory and charge. Some highly ionising particles are able to activate more than one pixel. In this case lower charged pixels can be observed around the pixel that the particle actually passed. Some examples for particles passing the PXD are given in figure 4.13. If a particle passes the PXD almost horizontally, the trend of leaving either narrow rectangular clusters in v or u direction or diagonal clusters figure 4.13a can be observed. If a particle passes the PXD vertically it activates only a single pixel. Due to particles passing the PXD at an angle additional cluster shapes can appear. On its way through the detector, the particle slows down. As described in the *Bethe-Bloch-equation* (see: [33]), particles lose more energy the slower they travel. The pixel in which the cluster deposits the most energy will be the seed. Other pixels that are merely touched might show lower pixel charges (see PC 5). Due to the collider's asymmetry, particles receive a boost in v-direction. Thus deuteron clusters have a tendency to be longer in v-direction. As single pixel clusters dominate for both antideuteron and background, the correlations concerning bigger clusters contribute less to the total amount of information. This explains why the first two principal components merely include any information about further cluster shape and charge correlations.



(a) Crossing PXD horizontal.

(b) Crossing PXD vertical.

(c) Passing PXD at an angle and in diagonal direction.

Figure 4.13: Schematic drawing of a particle passing PXD. The pixels are drawn as a chessboard. The particle's trajectory is depicted as a red line. Activated pixels are marked red. The number and geometry of an activated cluster depends on particle charge and crossing direction. A perfect horizontal crossing of the PXD (a) will not be observed due to the detector's geometry.

4.5 Summary, Facit and Outlook

In this chapter PCA was applied to a mix of the antideuterons and background data sets. Each of the two data sets mixed, consists of six different cluster properties, measured for 57 678 clusters. Both data sets have been plotted before applying PCA. They have been normalised to prevent scale-distortions, but did not show a normal distribution. Therefore, it was not expected that the PCA would lead to precise results. Other than expected, no conspicuous distortions like the horseshoe effect (see last chapter) were noticable. Additionally, the variables did not show very high values in the correlation so there was no need to extend the data set with further uncorrelated cluster properties. The highest correlations have been found within size-related and charge-related properties. These two are included in the first two principal components. Therefore, it is possible to reduce the data set to two dimensions. To maintain an appropriate level of precision it is advisable to choose three instead of two dimensions in further processing as then almost 90% of the information is kept. Attempts at physical interpretations of the results have been made, but the PCA results have been too abstract to draw a precise conclusion. Although it is visibe in figure 4.8b that antideuterons and background have a tendency to cluster in different directions, they show an overlap. Therefore they have not been separated properly. This was expected as PCA is not made for testing how similar two data sets are. Further interpretations an be found in project 3.

If a simlarity-check was meant to be done, Seber gave the advise to apply PCA on both data sets separately. Afterwards it is possible to test for similarities by comparing the first principal components [15, p. 201]. In further studies one might include more cluster properties such as *module*, *layer* and *ladder* or *cluster angle* into the PCA. When doing this, firstly it is important to check the correlation matrix, as there is no sense in transforming a huge data set of highly correlated properties. Seber advises in his book [15, p.200] to "reduce the number of variables to a smaller subset" before "carrying out a PCA". One could test by applying PCA one which variables contribute only in a small manner to the data set. These variables should be left out in the future.

An interesting topic for further studies could additionally be to test different versions of data processing with PCA. One could think of making a comparison between the PCA results given in this thesis and results of a PCA on a logarithmised data set. Taking the logarithm might change the data set into a distributon closer to the ideal normal distribution. Additionally, the mean in the PCA implementation should be replaced by median to minimise sensitivity for outliers. Alternatively, more stable PCA-implementations such as `RandomizedPCA` or `SparsePCA` included in Python's `Skikit-Learn`-package can be tested. To drive the PCA-methods further one could test hypotheses about principal components such as described in chapter 8 of Mardia's book [14, p. 233] and implement methods which are highly related to PCA such as *Correspondance Analysis* and *Allometry. Allometry* is often used in botanics and allows a measurement of size and shape [14, p.239].

In the following chapter cluster sizes, orientation and shapes will be studied in more depth to gain a further understanding of the data set and PCA results.

Chapter 5

Project 2: Cluster Shapes in the Antideuterons Data Set

In the last chapter, correlations between the six cluster properties have been discussed. One of the outcomes is that antideuteron cluster have a tendency to be wider in v-direction due to the collider's asymmetry. In this chapter a closer look will be taken onto the antideuteron clusters' shape. Any analysis concerning shape of background clusters will be skipped, as this would go beyond the scope of the discussion.

5.1 Processing of Data Set

The *Antideuterons_run_all.root* file that has been introduced in chapter 4 is used for cluster shape analysis in this chapter. When analysing cluster shapes, one has to include different cluster properties than the ones used in the last chapter. Therefore, data processing has to be altered. A cluster's shape depends on the coordinates of the single pixels which are forming a cluster. These coordinates cannot be computed by the `Clusterizer module` and are calculated by *Sim_to_clsprop_PCA_angle.py.* A full list of cluster properties used for shape analysis can be found in table 5.1.

Figure 5.1 visualises the steps of data processing. *Sim_to_clsprop_PCA_angle.py* prints the *pixel coordinates* as well as *size* and *charge* on the screen. The terminal command > is used to save the output in *deuteron.txt.* This file includes special characters such as ".", "[" and "]" that have to be removed before reading in the file as `numpy array` in *write_clusterprop_from_deuteron4.py.* Clusters are sorted by size and saved in *Clusters_lengthi.txt*, with *i* being the corresponding cluster size. As *write_clusterprop_from_ deuteron4.py* skips single pixel clusters, the smallest file includes clusters with size 2. Single pixel clusters are skipped due to the fact that they cannot have any special cluster shape. To analyse cluster shape, the Python skript *PCAPython_angle_function.py* loads all *Clusters_lengthi.txt* files line by line. Each line corresponds to a cluster and is therefore treated as separate data set. A PCA is done for the pixel coordinates in each line. The script calculates the angle between the original axes $(\boldsymbol{u}, \boldsymbol{v})$ and the first two principal components $(\boldsymbol{x}, \boldsymbol{y})$ and saves them together with the file number, the cluster number, *size* and *charge* in *File_clusterno_charge_size_ux_uy_vx_vy.txt.* Calculating angles between the axes gives the opportunity to learn more about the spatial distribution of the pixels.

A first set of rectangular clusters with a width of 1 pxl, is created by checking which clusters have pixel coordinates that are equal in $\boldsymbol{u}$ or $\boldsymbol{v}$ direction. Angles for these clusters are not computed as the correlation matrix cannot be calculated due to division by zero. Nevertheless, the principal components of these clusters have to be parallel to $\boldsymbol{u}$ or $\boldsymbol{v}$ because of the rectangular shape. Following on from that, the combination of angles $(\sphericalangle(\boldsymbol{u},\boldsymbol{x})\ ,\ \sphericalangle(\boldsymbol{u},\boldsymbol{y})\ ,\ \sphericalangle(\boldsymbol{v},\boldsymbol{x})\ ,\ \sphericalangle(\boldsymbol{v},\boldsymbol{y}))$ will probably be either (0°,90°, 90°, 0°) or (90°, 0°, 0°, 90°). The number of narrow rectangular clusters is saved together with the corresponding filename and cluster size in *File_rectangular-clusters_ percentage.txt.* A visualisation of this data set can be seen in figure 5.5 and will be explained in the next section.

[1] `Origin`® logo taken from *www.originlab.com.*

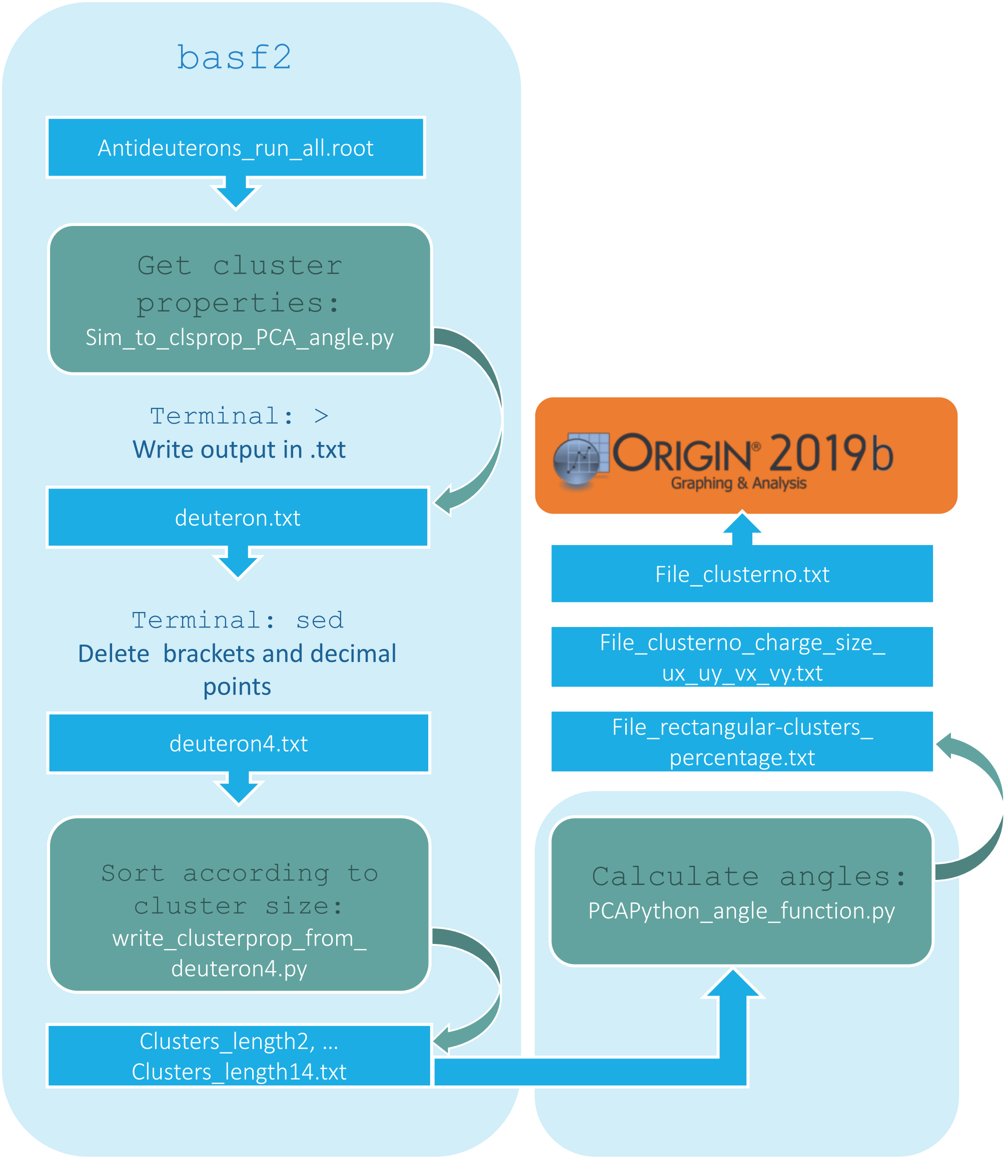

basf2
Antideuterons_run_all.root
Get cluster properties:
Sim_to_clsprop_PCA_angle.py
Terminal: >
Write output in .txt
deuteron.txt
Terminal: sed
Delete brackets and decimal points
deuteron4.txt
Sort according to cluster size:
write_clusterprop_from_deuteron4.py
Clusters_length2, ...
Clusters_length14.txt
ORIGIN® 2019b
Graphing & Analysis
File_clusterno.txt
File_clusterno_charge_size_ux_uy_vx_vy.txt
File_rectangular-clusters_percentage.txt
Calculate angles:
PCAPython_angle_function.py

Figure 5.1: Flowchart of data analysis in project 2. The *Antideuterons_run_all.root* file from project 1 is used to generate the new data set *deuteron.txt* that includes the *pixel coordinates* as well as the clusters' *size* and *charge*. All clusters with the same cluster *size* are then saved separatly in *Clusters_length2*, ... *Clusters_length14.txt*. These files are read into *PCAPython_angle_function.py* where angles between the original axes $(\boldsymbol{u}, \boldsymbol{v})$ and the first two principal components $(\boldsymbol{x}, \boldsymbol{y})$ are calculated. `Origin`® is used for graphical depictions[1].

Table 5.1: Original table and description taken from [1, p.37]. Note that some content has been altered. "Cluster properties used in this [and DORT's] thesis. The second column marks whether a property is already computed in the clusteriser. The third column indicates which properties are chosen to be part of the input vector" for PCA angular analysis.

Property	Computed by `Clusterizer`	Part of input vector
Total cluster charge	✓	✓
Cluster seed charge	✗	✗
Cluster minimum charge	✗	✗
Total cluster size	✓	✓
Total cluster length	✗	✗
Cluster size in u	✗	✗
Cluster size in v	✗	✗
Pixel coordinates in u	✗	✓
Pixel coordinates in v	✗	✓
Cluster angle	✗	✗
Cluster eccentricity	✗	✗

5.2 Analysis of Cluster Shapes

In this section the PCA method for angle computation is presented. The first part will give a short overview of the idea behind the method. Afterwards the results will be discussed.
Before looking at cluster shapes one should keep in mind that the antideuteron data set mostly consits of small-sized clusters. The reative frequency of cluster size is plotted in figure 5.2. Almost 90% of the clusters are smaller than 5 pxl. 35.7% of all 57 678 clusters consist of two pixels and only one cluster shows an overall size of 14 pxl. Clusters with sizes higher than 14 pxl have not been observed.

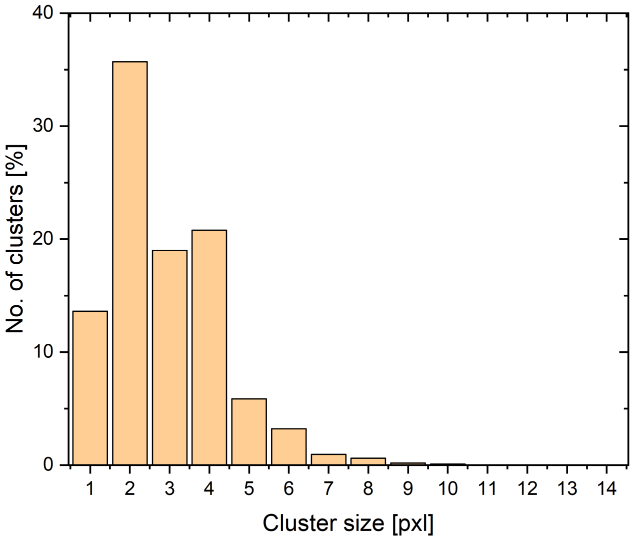


Figure 5.2: Cluster sizes and their relative frequency in the antideuteron data set. Most common are clusters with a general size of two. Clusters with sizes bigger than 14 pxl have not been observed.

5.2.1 Computing Cluster Angles with PCA

The idea behind the method is visualised in figure 5.3. Each pixel is treated as a point (u,v), with the u-coordinate representing its horizontal position, and the v-coordinate representing its vertical position in a module of the pixel detector. Each layer of a pixel detector's module represents an Euklidean two-dimensional space. The symmetry axes in perfectly diagonal clusters and the $\boldsymbol{u}$- and

$\boldsymbol{v}$-axis will therefore form a 45° angle. If PCA is applied to a perfectly diagonal cluster, the first principal component will be positioned alongside the symmetry axis, also in a 45° angle to the $\boldsymbol{u}$- or $\boldsymbol{v}$-axis. In the script, the axes are arbitrarily set as $\boldsymbol{u} = \binom{0}{1}$ and $\boldsymbol{v} = \binom{1}{0}$. The first principal component is referred to as $\boldsymbol{x}$- and the second principal component as $\boldsymbol{v}$-axis. The calculation of the angle uses the connection between the *dot product* and angle between two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$:

$$\boldsymbol{a} \cdot \boldsymbol{b} = |\boldsymbol{a}||\boldsymbol{b}|cos(\triangleleft(\boldsymbol{a}, \boldsymbol{b})) \tag{5.1}$$

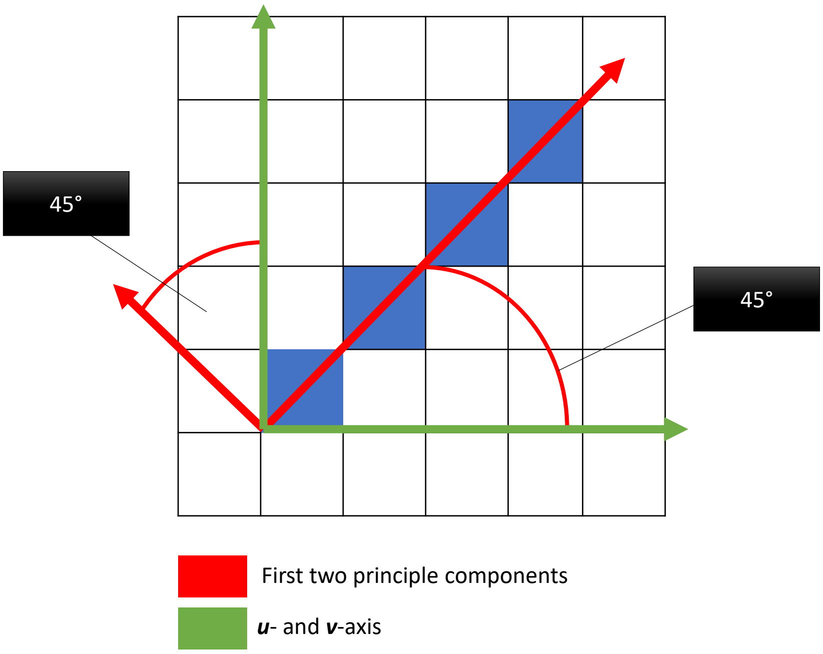


Figure 5.3: Visualisation of the calculation of angles in project 2. The blue squares represent pixels in the (u,v)-layer. The first two principal components of a data set consisting of pixel coordinates are computed. Afterwards the angles between the original coordinates (green) and the principal components (red) can be calculated.

As all vectors are known, a set of four angles can be calculated: $(\triangleleft(\boldsymbol{u}, \boldsymbol{x})$, $\triangleleft(\boldsymbol{u}, \boldsymbol{y})$, $\triangleleft(\boldsymbol{v}, \boldsymbol{x})$, $\triangleleft(\boldsymbol{v}, \boldsymbol{y}))$. The results of the analysis are listed in table 5.2 and depicted in figure 5.4. The table groups clusters according to their angular distribution into four groups:

- *Diagonal*: This group of clusters is characterised by a principal component that is parallel to the angle bisector of the $(\boldsymbol{u}$,$\boldsymbol{v})$-layer. If the cluster shape is not perfectly symmetrical, small deviations (≈ 1°) appear. This group of clusters can also be rectangular if their symmetry axis is equal to the angle bisector.

- *Possibly rectangular*: The combination of angles shows that the principal components are parallel to the $\boldsymbol{u}$- and $\boldsymbol{v}$-axis. It is highly probable that these clusters are rectangular with a width > 1 pxl. Additionally, it is possible that they are shaped irregulary with a tendency to group evenly distributed around an axis that is parallel to the $\boldsymbol{u}$- or $\boldsymbol{v}$-axis.

- *Rectangular*: This group of clusters has pixel coordinates that are either equal in u- or v-direction. Accordingly, they are rectangular with a width of 1 pxl. PCA cannot be applied to this set of clusters as the variance is equal to zero. If angles were computed, they would have been either (0°, 90°, 90°, 0°) or (90°, 0°, 0°, 90°).

- *Single pixel*: This cluster type consits of only one pixel and therefore has no shape apart from the pixel's shape itself.

It is notable that only six different combinations of angles have been observed. This should not be mistaken with only six different cluster shapes. Most of the pixels are distributed along the principal components, but this does not imply that there is a perfect symmetry in the distribution along those axes.

Table 5.2: Angles between original axes $(\boldsymbol{u}, \boldsymbol{v})$ and the first two principal components $(\boldsymbol{x}, \boldsymbol{y})$ calculated for antideuteron clusters. Clusters have been grouped according to their shape. The 6^{th} column shows how many clusters with the corresponding angle were found in the antideuterons data set. The 7^{th} column displays the percentage of those clusters with regard to the total amount of antideuteron clusters. Note that angles for single pixel clusters and rectangular clusters have not been calculated.

Shape	$\sphericalangle(\boldsymbol{u}, \boldsymbol{x})$ [°]	$\sphericalangle(\boldsymbol{u}, \boldsymbol{y})$ [°]	$\sphericalangle(\boldsymbol{v}, \boldsymbol{x})$ [°]	$\sphericalangle(\boldsymbol{v}, \boldsymbol{y})$ [°]	No. of clusters
Diagonal	135 45 135 135	45 45 135 45	45 45 45 135	45 135 135 135	5715 5428 938 551
Possibly rectangular	0 90	90 0	90 0	0 90	11712 1
Rectangular (width = 1 pxl)	-	-	-	-	25506
Single pixel	-	-	-	-	7827

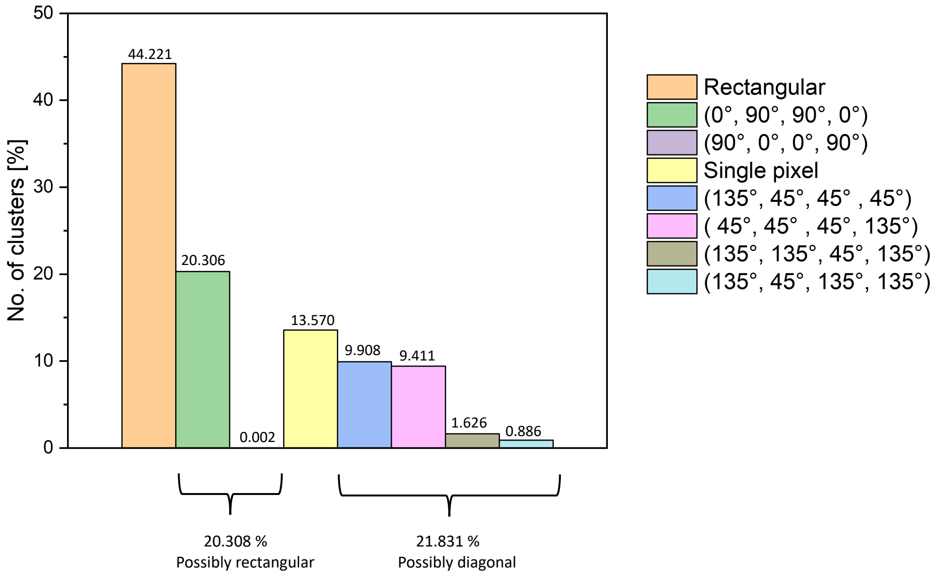


Figure 5.4: Relative frequency of angles between original axes and the first two PC.

The relative frequency of narrow rectangular clusters depending on cluster size can be see in figure

5.5. Among rectangular clusters size two is predominant. Clusters with sizes higher than 5 pxl have not been observed. After randomly plotting some clusters it has been notable that 9707 4 pxl sized square-shaped clusters appear in the "possibly rectangular" and "diagonal clusters"-set. They make up 16.83 % of the total data set. The incidence of those clusters in both angular distributions is explained by the fact that spatial information is evenly distributed in rectangular clusters. Among the 4 pxl sized clusters 89.32 % are square-shaped.

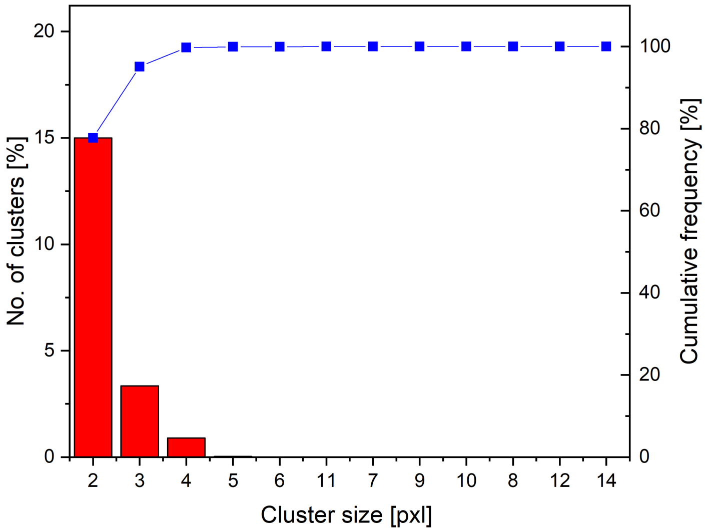
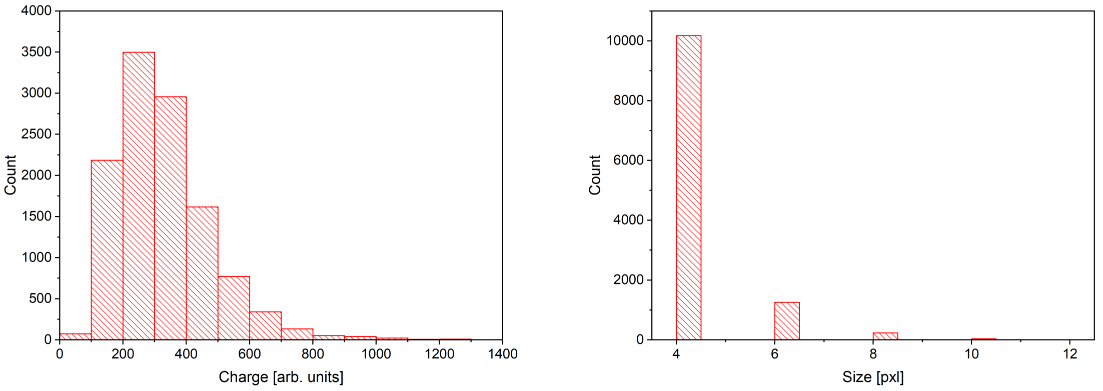


Figure 5.5: Relative frequency of rectangular clusters with a width of 1 pxl depending on cluster size. The cumulative frequency is depicted by a blue line.

For each combination of angles found in the data set histograms of size and charge have been created (figure 5.6 - 5.10). The first two histograms belonging to (0°, 90°, 90°, 0°)-clusters are presented in figure 5.6. More than 85% of all clusters within this angular distribution consist of four pixels. It is conspicuous that these clusters show only even numbers in size. It is likely that these clusters have a width of 2 pxl, because rectangular clusters broader than 3 pxl would be at least 9 pxl in size. Clusters like that are possible, but they do not dominate as seen in figure 5.2. One can definitely tell that square-shaped 4 pxl sized clusters have width 2. More than half of (0°, 90°, 90°, 0°)-clusters are charged between 100 and 500 (arbitrary units). As the combination of angles (90°, 0°, 0°, 90°) only exists once, no histogram has been created. With 889 (arbitrary units) this cluster is above-average charge and huge in size (10 pxl).

The biggest set of diagonal clusters shows the widest variety in cluster size. In figure 5.7 one can observe that clusters from sizes between 3 and 9 show the combination of angles (135°, 45°, 45°, 45°). One should point out that 4 pxl clusters are underrepresented compared to the data set shown in figure 5.6. Additionally, a 14 pxl cluster appeared which might be an outlier due to its size and high charge of 1501 (arbitrary units). All other clusters resemble the (0°, 90°, 90°, 0°)-clusters in charge distribution. The diagonal clusters set includes more clusters charged less than 200, than the possibly rectangular ones discussed above. Clusters with angles (45°, 45°, 45°, 135°) behave almost equally, despite the fact that no size 14 cluster appeared.

More differences can be seen in figure 5.9 and 5.10, as they do not show any 4 pxl clusters and a preference for clusters charged from 100 to 300. Most of these include 3 pxl, although sizes up to 9 pxl have been registered. One notable difference between clusters with angles (135°, 135°, 45°, 135°) and (135°, 45°, 135°, 135°) is that the for (135°, 135°, 45°, 135°)-angles more 5 pxl clusters are observable whereas the other data set shows higher amounts of 7 pxl clusters. A 12 pxl cluster with a high charge of 1381 arbitrary units is depicted with angles (135°, 45°, 135°, 135°).



(a) Charge histogram

(b) Size histogram

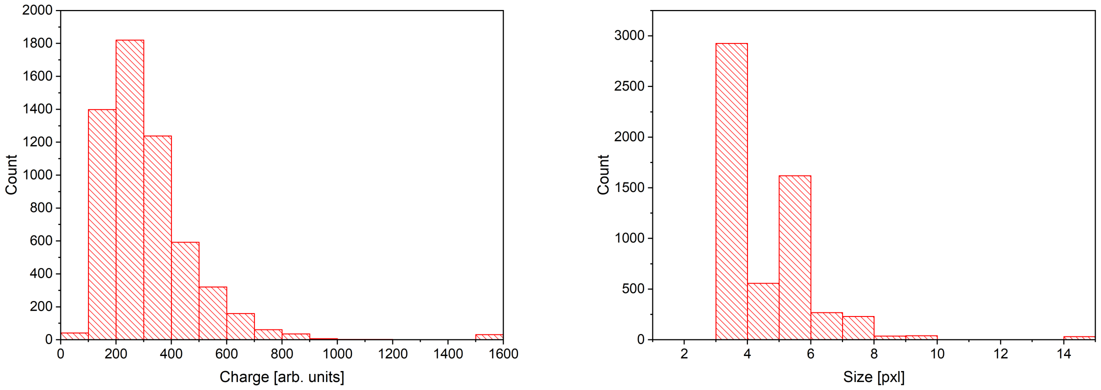
Figure 5.6: Histograms of charge and size for (0°, 90°, 90°, 0°)-clusters. Only even-numbered sizes have been observed. Square-shaped 4 pxl clusters dominate.

5.3 Summary and Facit

In this chapter cluster shapes have been analysed by grouping clusters depending on their spatial orientation. The first two axes of the cluster, which included most of the pixels, were calculated using PCA. Together with the dot product, it was possible to calculate the angle between the original axes and the principal components. This way, it has been possible to differentiate between clusters that are possibly rectangular and others that are possibly diagonal. The analysis of cluster shapes led to the following results:

- Most of the clusters are small in size (< 5 pxl). Clusters made of 4 pxl are dominant.
- 44.22 % narrow rectangular clusters of 2 to 4 pxl in size have been observed.
- 20.31 % possibly rectangular clusters mostly consisting of 4 pxl were depicted and
- 89.32 % of the 4 pxl clusters are squares.
- 21.83 % diagonal clusters (including rectangles along the angle bisector) have been found. They show a tendency to be made of 3 pxl and charged between 100 to 400 (arb. units).
- Diagnonal clusters with angles (135°, 45°, 135°, 135°) and (135°, 135°, 45°, 135°) are never made of 4 pxl.
- 13.57 % single pixel clusters have been registered.

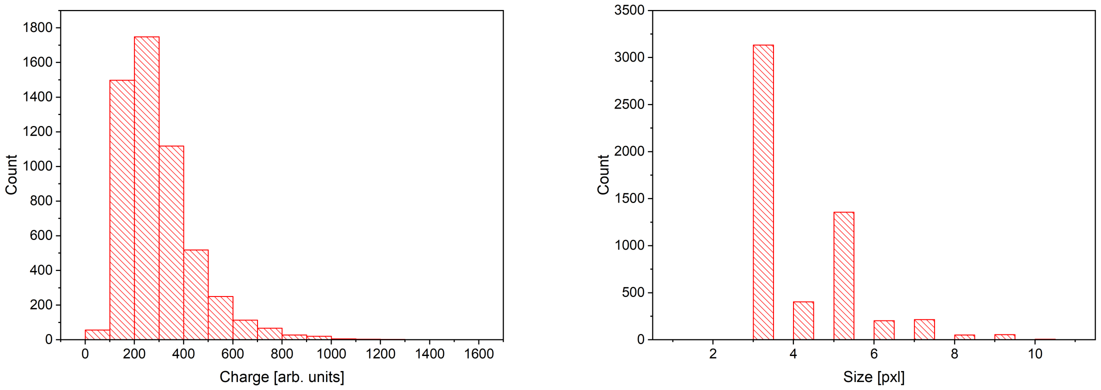
The PCA-method to find angles proved to be very inaccurate. To state an example, square-shaped clusters are included in both possibly rectangular and possibly diagonal clusters. One should mention that the classification was very rough. In further studies one should differentiate between accurate and rounded values of angles, as most clusters are not perfectly diagnonal or rectangular (see figure 5.11). Very precise calculations are needed, as mostly angles differ in the 4^{th} or 5^{th} significant number. Additionally, the results do not include any relative frequencies of precise cluster shapes. To detect relative frequencies of unknown cluster shapes, it is advisable to use deep learning algorithms for pattern recognition in the future.



(a) Charge histogram

(b) Size histogram

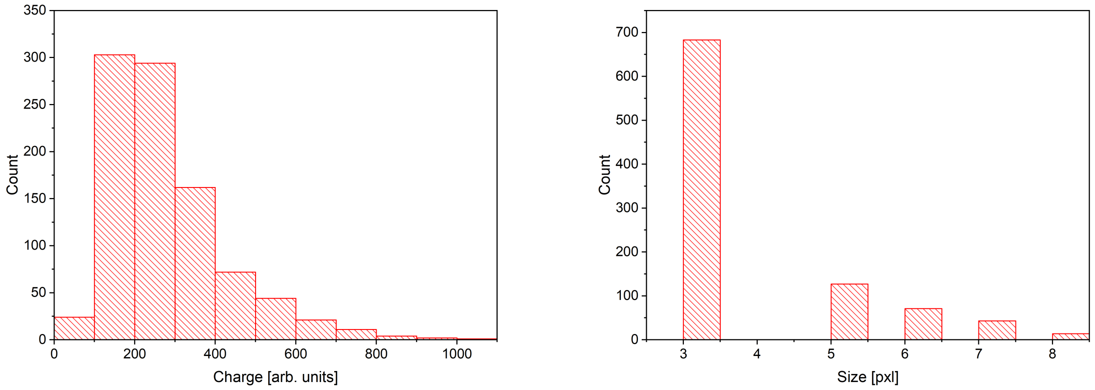
Figure 5.7: Histograms of charge and size for (135°, 45°, 45°, 135°)-clusters. Sizes from 3 to 10 pyl have been observed. Square-shaped 4 pxl clusters and 5 pxl-clusters dominate.



(a) Charge histogram

(b) Size histogram

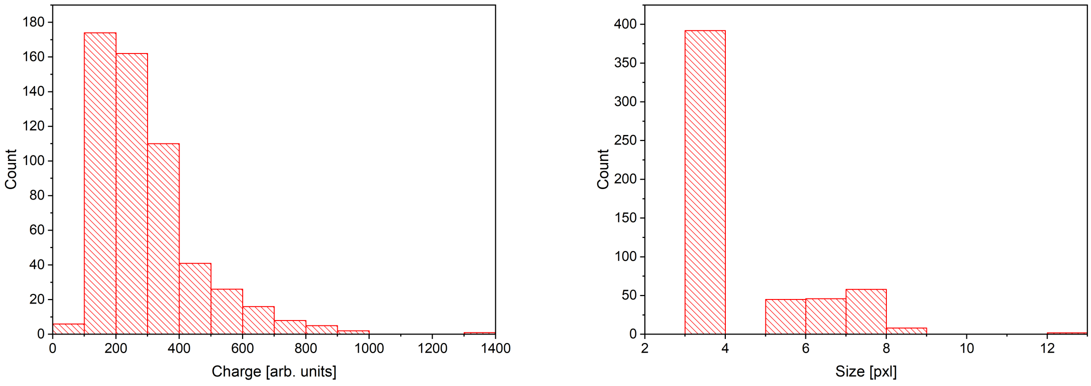
Figure 5.8: Histograms of charge and size for (45°, 45°, 45°, 135°)-clusters. Cluster with sizes from 3 to 9 pxl were found.



(a) Charge histogram

(b) Size histogram

Figure 5.9: Histograms of charge and size for (135°, 135°, 45°, 135°)-clusters. As visible in (b), no 4 pxl clusters are included in this data set.



(a) Charge histogram

(b) Size histogram

Figure 5.10: Histograms of charge and size for (135°, 45°, 135°, 135°)-clusters. As visible in (b), no 4 pxl clusters are included in this data set.

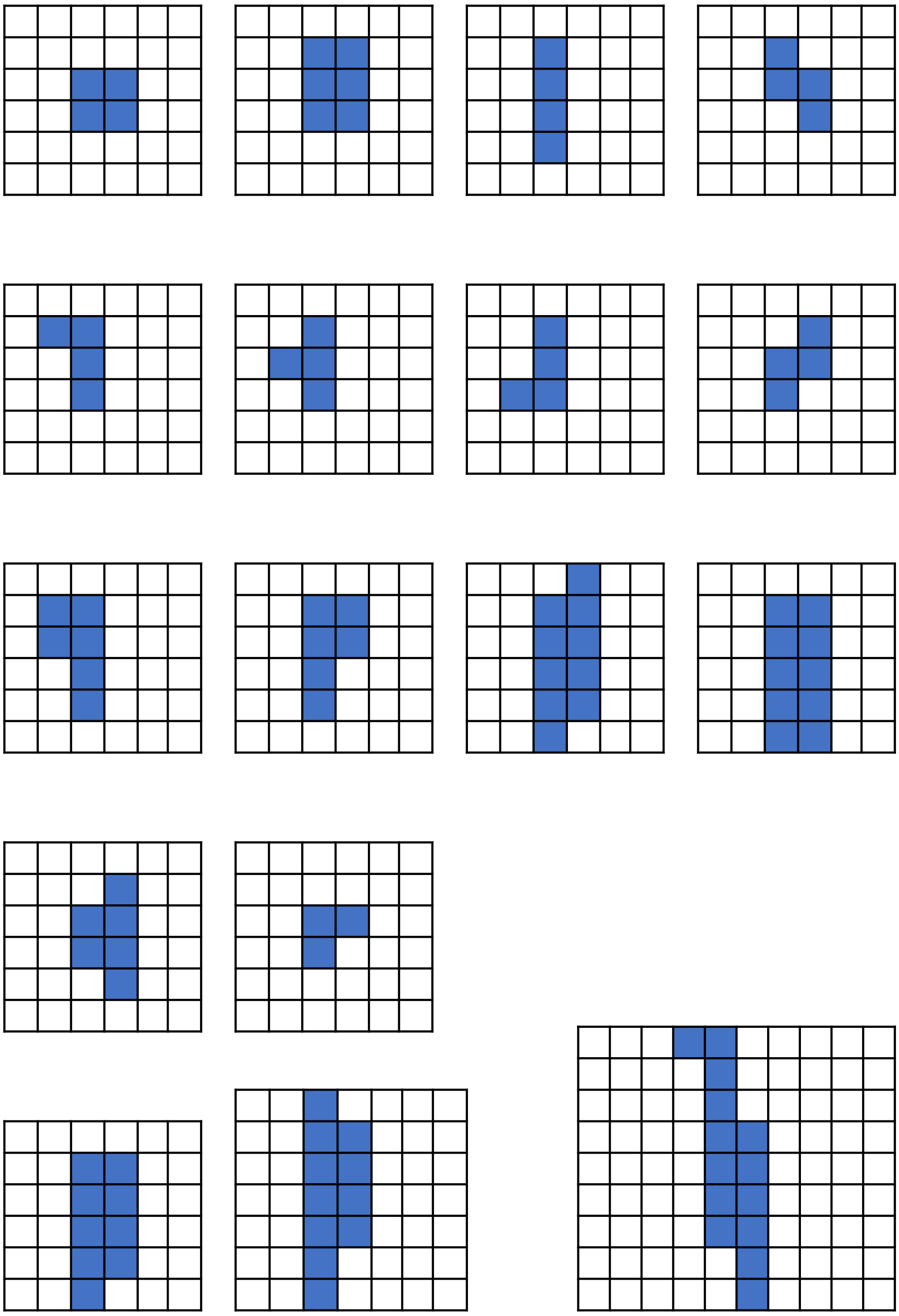


Figure 5.11: Randomly chosen cluster shapes that can be found in the the antideuterons data set.

Chapter 6

Project 3: Data Separation Using SOMs

In preparation for the processing of real data coming from Belle II, methods for separating different particles from background have to be obtained. To differentiate these particles from background, artifical intelligence can be used. Considerable progess in this field of work has been made by Katharina Dort. Her results using SOMs and FFNNs can be found in [1]. As this thesis can be seen as a direct successor of her work, the same implementation of SOMs is used to test if PCA improves the separation process. In the first section of this chapter the classifcation results of SOMs trained with and without PC-transformed antideuterons and background data, will be compared. Collisions in detectors such as Belle II lead to the production of several different particles, therefore studies including the separation of **multiple** signals from background are needed. The second section of this chapter aims to make a first attempt in separating mixed particles from background by using a combination of the antideuterons data set with π^- and a simplified version of tetraquarks. Both sections include a short conclusion at the end.

Note on SOM classification maps

The classification maps in this thesis show patches from blue to yellow. The maps were created by choosing the winning node for each input vector. The classification index stays beyond 0.5 (blue-green) if the winning node is more *background-like* and reaches values up to 1.0 (yellow) for *signal-like* (e.g. *antideuteron-like*) nodes [1, p.66]. Dark blue squares belong to nodes that either belong to background or failed to assign to background or signal.

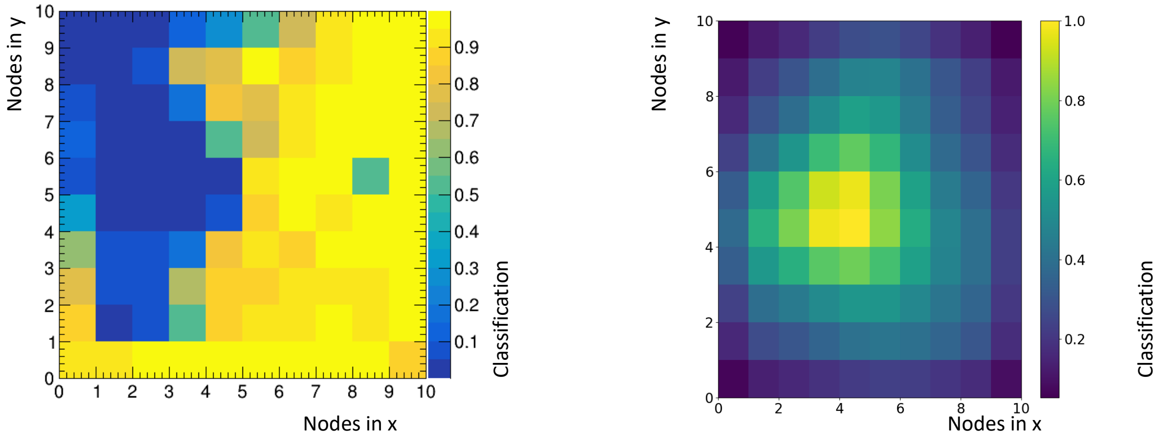
6.1 Separating Antideuterons from Background

6.1.1 Training with PCA-transformed Data

The PC-transformed data set of antideuterons and background presented in the last chapter has been used to train a *MiniSom*. The first 57 678 rows of the data set include the transformed antideuterons data, the following 57 678 rows are filled with transformed background data. All six columns have been read into the *MiniSom*-file *som_clsprop_with_pca.py* as a numpy array and were randomly shuffled before training. Shuffling is needed, as sorted data sets lead to distortions in the final network and therefore data separation will be impossible.

The training results are depicted in figure 6.1a. A large dark blue patch is visible, that spreads unevenly from the upper left to the centre. Lighter blue and turquoise squares accumulate on the outer left and upper left side of the map. All bluish nodes correspond to background-like nodes. A green node is placed in the yellow antideuteron region. This node did not match properly with any antideuteron-input. Ths node is left blank in the background winner map (figure 6.2a). Therefore, this node did not respond to background-input at all. The Euklidean distance map (figure 6.1b) reveals the fact, that nodes' weights in the left centre tend to differentiate higher from the input vectors than at the outer edges. This leads to the issue that data might not be classified as precisely in this region. The region is therefore interpreted as passage between background and antideuteron

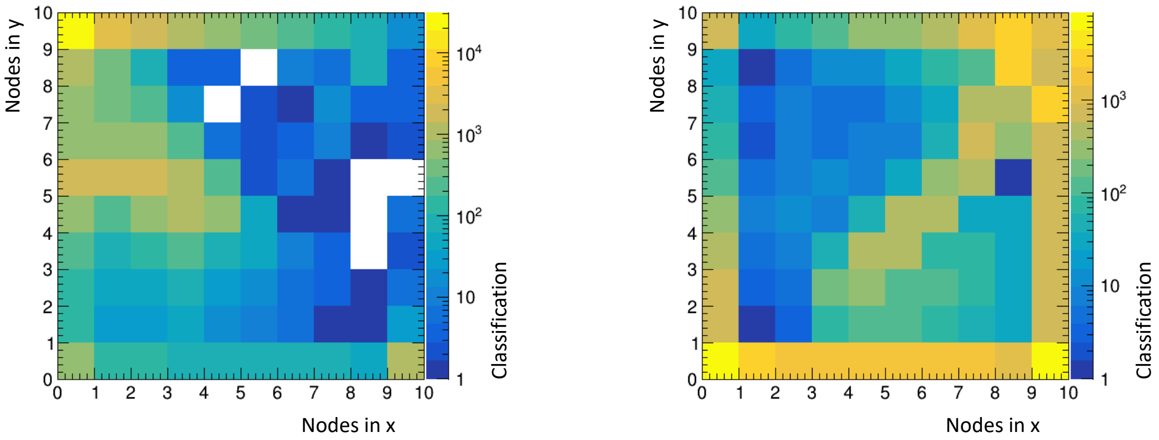
nodes. For a deeper understanding of the map one should look at the background and antideuteron winner nodes maps. Figure 6.2a shows the nodes which responded to background input. Yellow indicates a high response, whereas blue is connected to low responses. The winning background nodes seem to be placed at the outer edges of the map, especially around the left edge and the left side of the centre. Antideuteron winner nodes are displayed in figure 6.2b. All outer-edged nodes reply to antideuteron-input - also those who already reacted on backgroud. Nevertheless, one could state that most antideuteron-like nodes group at the outer right side, the upper right edge and lower edge of the map. On top of that, a narrow diagonal line in the centre is depicted.
If a cut should be made between the background-like and antideuteron-like nodes one might cut along the diagonal between nodes (0,0) and (10,10). With the implementation of ROC-curves that have already been used in Dort's thesis [1], cutting along diagonals was impossible. Instead a cut was made along the y-axis. When looking at figure 6.3b, which depicts the projection of the map onto the y-axis, one can see that deuteron-like nodes mostly group at the left edge. At the outer right edge of the y-axis another peak of background-like nodes is visble. Antideuteron-like nodes show worse spatial distributions on the map. They seem to be almost evenly distributed, with a slight tendency to dominate at the right side of the y-axis. The ROC-curve visible in figure 6.3a does not imply perfect classification results, but nevertheless one can say that the classification process was successful.



(a) Classification map

(b) Euklidean distance map

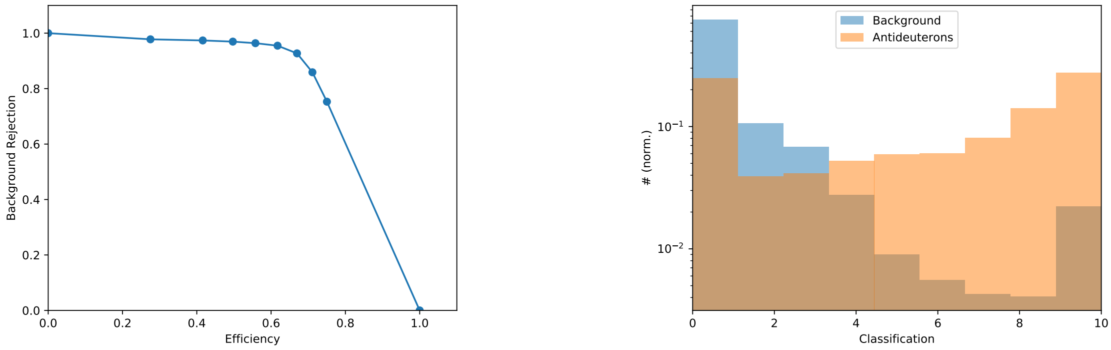
Figure 6.1: Classification (a) and Euklidean distance (b) map of SOM trained with PC-transformed data.



(a) Background winner nodes

(b) Antieuteron winner nodes

Figure 6.2: Background (a) and antideuteron (b) winner maps of SOM trained with PC-transformed data.



(a) ROC-curve

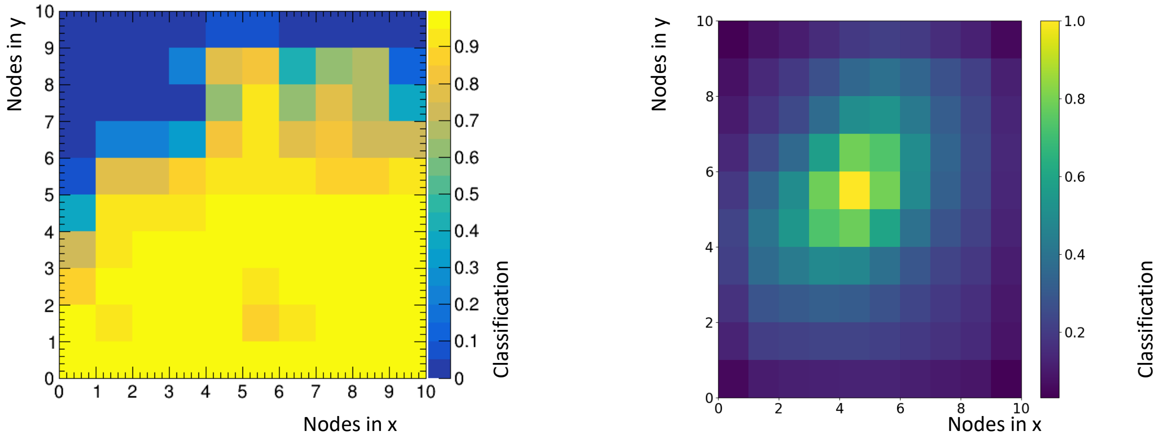
(b) Projection on y-axis

Figure 6.3: ROC-curve and projection of the map onto the y-axis of a SOM trained with PC-transformed data of antideuterons and beam background. (a) ROC-curve plotting background rejection against efficiency. The ROC-curve shows that the classification process was quite successful. (b) The projection of the map onto the y-axis shows how background (blue) and antideuteron (orange) were separated.

6.1.2 Training with Non PCA-transformed Data

When taking a look at the classification maps, the classification seemingly improves if the data set is not pre-transformed into PC-space. Although the classification map in figure 6.4a also shows dark blue patches in the upper left, the Euklidean distance map (figure 6.4b) depicts generally smaller distances between values of input vectors and node weights. Only a small transition region is visible in the centre. The background-like nodes cluster on top of the map (see figure 6.5a), while antideuteron-like nodes group at the lower outer edges of the map and in the centre.

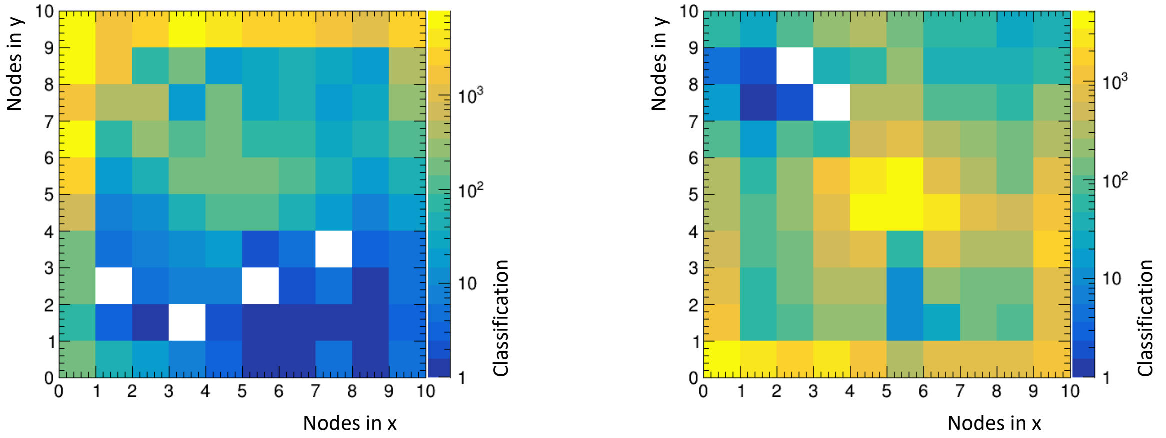
A cut between background-like and antideuteron-like signals was made along the y-axis, leading to the projection visible in figure 6.6b. This figure implies that the classification was worse than with PCA-transformed data. The ROC-curve shown in figure 6.6a confirms this statement. This result conflicts with the interpretation of the classification maps and Euklidean distance map, as these suggest that non-PCA transformed data leads to better classification results.



(a) Classification map

(b) Euklidean distance map

Figure 6.4: Classification (a) and Euklidean distance (b) map of SOM trained with non PCA-transformed data.



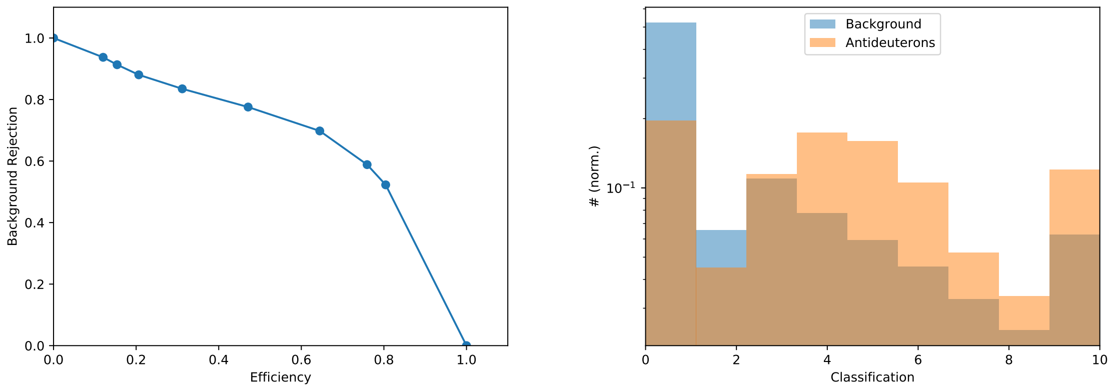
(a) Background winner nodes

(b) Antideuteron winner nodes

Figure 6.5: Background (a) and antideuteron (b) winner maps of SOM trained with non PCA-transformed data.

6.1.3 Conclusion

It is highly probable that PC-transformed data performs worse due to the impact of the mean on the correlation matrix. The mean is calulated over both background and antideuteron set, causing background data to become more antideuteron-like and vice versa. Although PCA is used to find groups or clusters in a given data set, it is unable to differentiate perfectly between background and antideuterons. As demonstrated by the overlap of antideuterons and background in most plots shown in the last chapter, the similarities are not negligible. This hypothesis contradicts the fact that the ROC-curve of the PCA-transformed data set suggests better classification. One should keep in mind that in both maps, cuts have been made along the y-axis. Results of the ROC-curves might therefore be without significance, as the spatial distribution of blue background-like nodes indicates that cuts need to be made along the plane diagonal for correct separation.



(a) ROC-curve

(b) Projection on y-axis

Figure 6.6: ROC-curve and projection of the map onto the y-axis of a SOM trained with antideuterons and beam background. No PC-transformation has been applied. (a) ROC-curve plotting background rejection against efficiency. The ROC-curve shows that the classification process was not ideal but neither without success. (b) The projection of the map onto the y-axis shows that background (blue) and antideuteron (orange) were not separated properly.

6.2 Separating Multiple Signals from Background

To prepare the separation of signal and background coming from real data sets, a mixed data set of antideuterons, negatively charged pions, tetraquarks and beam background is used in this section. The first subsection will be a short description of the generation of data. Afterwards, the results of the classification process using *MiniSom* will be presented in the second subsection.

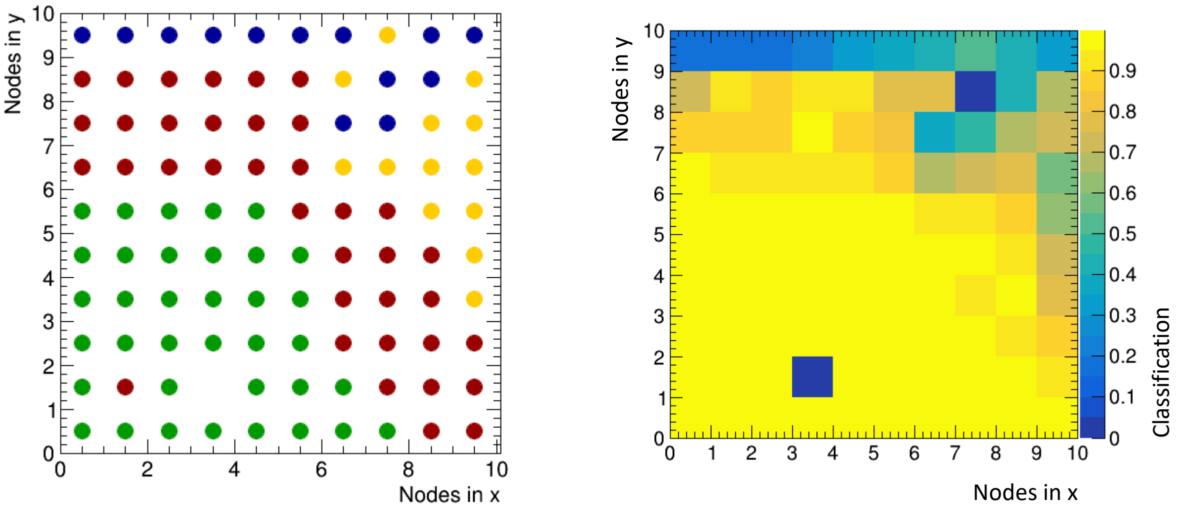
6.2.1 Generation of the Mixed Data Set

To ensure optimal training results four different data sets have been created:

- antideuterons
- pions
- simplified tetraquarks
- beam background

If the particles had been created as one data set and background as another, it would have been impossible to localise the individual particles' types in the classification map. This is due to the fact that the map would only differentiate between signal (antideuterons, pions, tetraquarks) and background.

The generation of the antideuterons data set has already been described in chapter 4. The output file *Beam_dd_cluster_sim.txt* was reused in this project. For negatively charged pions a data set was created analog to the antideuterons data set by changing the *Monte Carlo Particle Identification Number* from (-10001020 $\hat{=}$ antideuteron) to (-211 $\hat{=}$ π^-) in the files *anti_deuterons.py* and *Sim_to_clsprop_modinfo_dd.py*. All files were renamed to include *..._pi..* in their name instead of *..._dd...* The C-skript *Root_to_txt_dd.C* was changed to save the pion data set in the output file *Beam_pi_cluster_sim.txt*. As tetraquarks are not included in the *Monte Carlo Particle Set*, a simplified version was made by changing the antideuteron's mass to 3.882 GeV. As aforemetioned, this is justified by the fact that singular electrically charged tetraquark states like the $\Upsilon(3882)$ approximately resemble the antideuteron's physical behaviour. After changing the mass of the antideteron in the `basf2`-file `evt.pdl`, the tetraquark data set has been created in the exact same way as the antideuterons data set. It has been saved in *Beam_T_cluster_sim.txt*. The background file had to be altered, as this time it was needed to remove antideuteros and pions. Since tetraquarks are not included in the *Monte Carlo Particle Set* and accordingly also lacked in the beam background data set, there was no need to remove them. For removing antideuterons and negatively charged pions, the script *Sim_to_clsprop_modinfo_BG.py* had been revised.



(a)

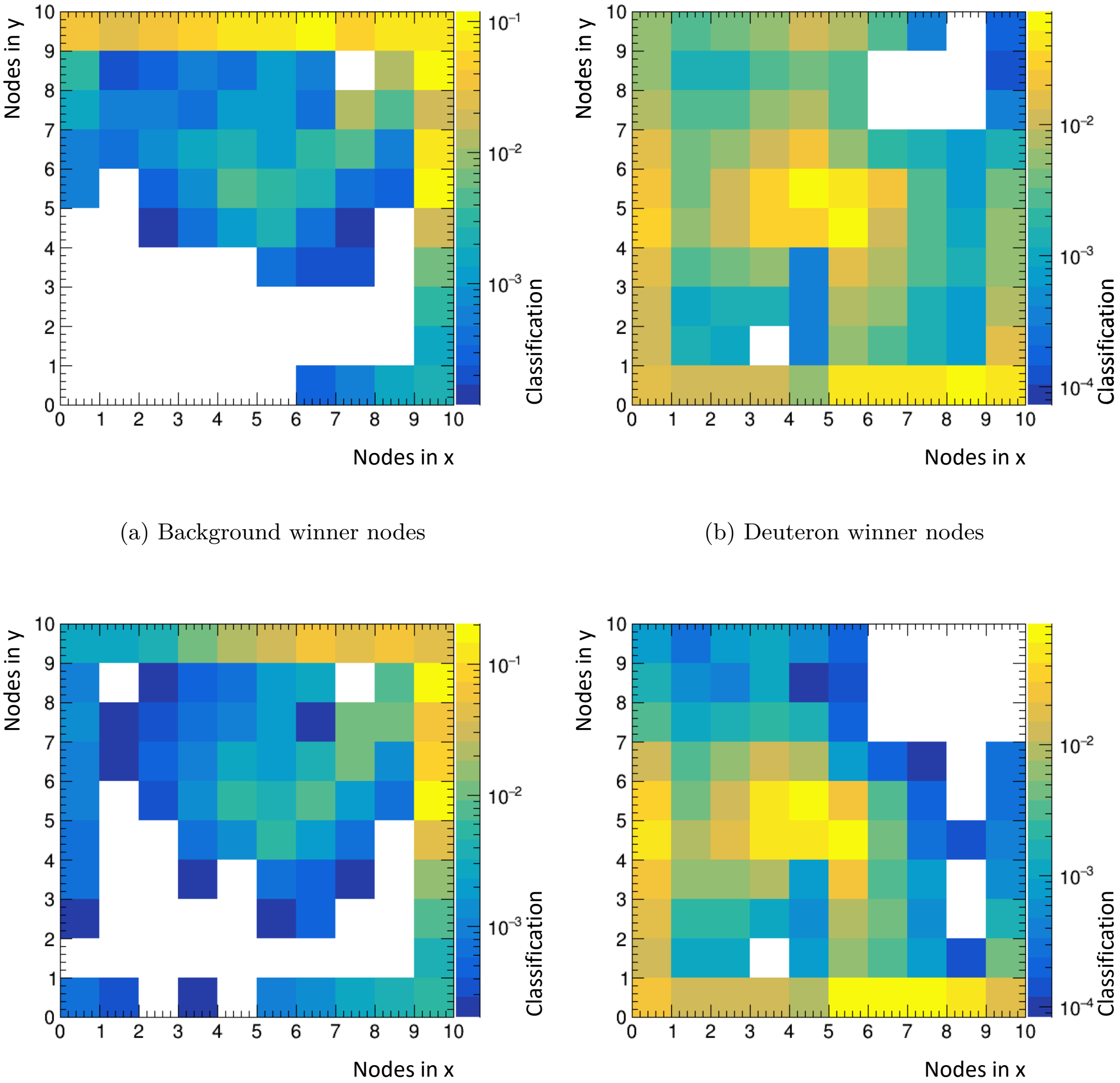
(b)

Figure 6.7: Classification maps of a SOM trained with antideuterons, π^-, tetraquarks and beam background. (a) The winning nodes with the highest resemblance to the respective particles plotted. Background-like (blue) and pion-like (yellow) nodes cluster at the outer right edge, whereas antideuteron-like (red) nodes are grouped in the transition region between tetraquark-like nodes (green) and the background-pion region. (b) Classification map showing the distribution of signal-like (yellow) and background-like nodes (blue). Signal-like nodes include antideuteron-, pion- and tetraquark-like nodes. The dark blue node in the lower centre is interpreted as neuron that neither responded to background nor signal.

6.2.2 Training with the Mixed Data Set

All four files have been read into *som_clsprop_without_pca.py*. The results of the classification process can be seen in figure 6.7. Figure 6.7b only depicts a differentiation between background (blue) and any other signals (yellow). A colour gradient between colours is only clear and precise if two classes are separated. To get more information about the signal distribution, the map has been complemented by a four-coloured version. It colours the neurons that have been most background-like blue, the ones that have been most deuteron-like red, pion-like yellow and tetraquark-like green. The map suggests that most neurons are tetraquark-like. Antideuteron-like signals can be found on the diagonal line that cuts the map in two halves. Pions are grouped on the upper left edge, mixing with background-like nodes that cluster on the upper edge. This is confirmed by the winner maps of the single groups (figure 6.8a-d). Blue colour corresponds to a low accordance between node and input, whereas yellow represents a high similarity. The topological distribution of pion and background nodes is quite similar, as both show lots of winning nodes (yellow) at the upper right edge of figure 6.8a and 6.8c. When taking a closer look at the winner maps of antideuteron and tetraquark (figure 6.8b+d), the same effect is notable. Antideuteron and tetraquark winning nodes are seemingly mostly the same, despite from the upper part of the map where nodes are more antideuteron-like. As tetraquarks and antideuterons only differ in mass, it stands to reason that the SOM has difficulties in differentiating them. The similarities between background and pion winner maps lead to the conclusion that the background file includes particles that share high similarities in cluster properties with π^-. It is possible that π^+ particles are responsible for this phenomenon.

The Euklidean distance map depicts that distances between input and neuron weights are overall small. Only two neurons in the middle of the map show higher distances, as this region is the transition region between the four groups.



(a) Background winner nodes

(b) Deuteron winner nodes

(c) Pion winner nodes

(d) Tetraquark winner nodes

Figure 6.8: Winner maps of the SOM trained with antideuterons, π^-, tetraquarks and beam background. The coloured squares represent the nodes thet responded to the input. Yellow represents a high accordance between input and node, blue stands for low accordances. Background-like (a) and pion-like (c) nodes show very similar topological distributions of winning neurons. The same effect is observable between antideuteron-like (b) and tetraquark-like (d) nodes.

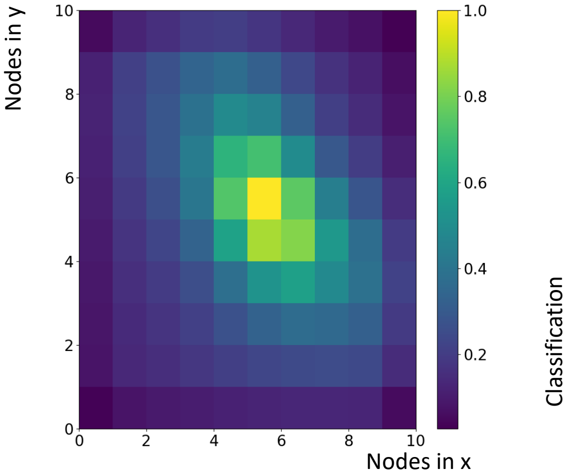
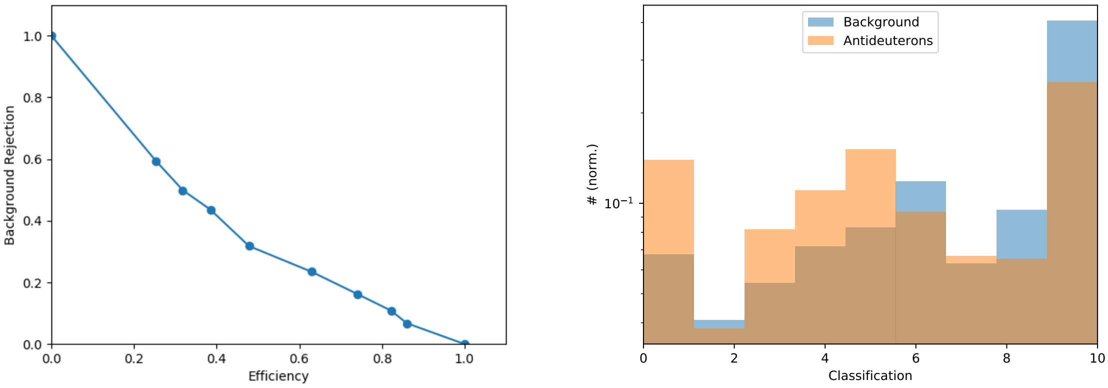


Figure 6.9: Euklidean distance map of the SOM trained with antideuterons, π^-, tetraquarks and beam background. Yellow corresponds to high distances, whereas blue is interpreted as low distance. The yellow patch in the middle is therefore interpreted as transition region between the three different signal types and background.



(a) ROC-curve

(b) Projection on y-axis

Figure 6.10: ROC-curve and projection of the map onto the y-axis of a SOM trained with antideuterons, π^-, tetraquarks and beam background. (a) ROC-curve plotting background rejection against efficiency. The ROC-curve shows the worst case possible in classification processes. (b) The projection of the map onto the y-axis shows that background (blue) and antideuteron (orange) were not separated properly.

6.2.3 Conclusion

Although the Euklidean distance map implies a good matching of weights and input, this can only be understood as a good result for separating signal from background. Within the signals, the separation of particle types was not overly successful. The winner maps show that particles with similar properties lead to a respondance of mostly the same nodes. If real data sets were used on this pre-trained SOM, no precise statement about the particle type would have been possible. Therefore, it is advisable to test other versions of SOMs that might lead to better results than the *MiniSom* implementation used in this thesis. Additionally increasing the number of cluster properties used as input variables will possibly lead to better classification results. In this case one should remember to use uncorrelated properties. To test the amount of correlation between cluster properties, PCA can be applied as shown in the first sections of this chapter.

Chapter 7

Summary and Outlook

This thesis can be seen as a preparation for future analysis of data measured in Belle II's PXD. A focus was set to the understanding of properties coming from simulated antideuterons. They are especially interesting as they resemble the predicted tetraquark $\Upsilon(3882)$ in charge. Any progess in the data analysis of signals coming from $\Upsilon(3882)$ is a further step into the discovery of non-quarkconia tetraquarks. To detect these HIPs, one has to use the PXD, as they might not reach the outer detector parts. The PXD is made from DEPFET-pixels. If an ionising particle passes the PXD it leads to a charge flux in the pixels. If a threshold is passed, pixels are activated. Neighboured pixels are grouped into clusters, as they have been activated by the same particle. Among these clusters, different properties like tion of pixel clusters. For this reason, the correlations between six cluster properties (*charge*, *minimum charge*, *seed*, *size*, *size in u* and *size in v*) can be measured.

In this thesis PCA has been applied to an antideuterons and background data set. It has been observed that the highest correlations can be found between *size*, *size in u* and *size in v* as well as *charge*, *minimum charge* and *seed*. This was expected, as these properties are related to each other per definition. Therefore, it is possible to reduce the dimension of the data set to three dimensions in further studies, without losing more than 11 % of the total information. The 3^{rd} principal component already implied that there might be clusters which are equal in length and width. This fact has been verified. A calculation of the spatial orientation of clusters was made using PCA. One of the outcomes is that 16.32 % of the clusters are quadratic and of four pixels in size. Among all clusters 2 pxl sized clusters dominate. About 90 % of all clusters are smaller than 5 pxl in size. Additionally 44.21 % of the clusters proved to be rectangular with a width of 1 pxl. Another 20 % are shaped in a way that they can be named approximately rectangular or diagonal.

The antideuterons and beam background data set was used as an input for a SOM. The classification and winner maps showed that the differentiation between background and signal was far from being ideal. When comparing the result to a SOM trained with non-PCA-transformed data it was notable that the classification and winner maps looked significantly better, but the ROC-curves implied a worse classification. This has been justified by the fact, that the cuts, that are needed for calculating the ROC-curves, should have been made along the map's diagonal instead of the y-axis. Therefore, the conclusion was made, that PCA did not improve classification.

Another outcome of this thesis is, that the *MiniSom* implementation of SOM should be replaced by a more sensitive version, as the SOM was not able to differentiate properly between three different particles and background. Additionally, one should enlarge the number of uncorrelated cluster properties used as input for training. Detailed suggestions for further data analysis on the PXD cluster properties can be found in the respective chapters. In the following a short summary will be given.

It is advisable to improve the PCA method by using more stable implementations and transforming the data set into an approximately normal distribution by taking the logarithm. Additionally, one should enlarge the number of uncorrelated cluster properties that are PC-transformed. Concerning the cluster shape analysis, one might try to use non-intelligent methods for finding patterns first. To state an example, one can try to examine local pixel configurations. As some of them can be transformed into each other by rotations, one should put them into congruence classes to prepare further analysis [34, p.26]. More details can be found in literature on *digital image processing.* For an introduction into image processing with Java, ref. [35] is advisable. Interesting tutorials on image processing using Python can be found at *pyimagesearch.com.* For example, a short tutorial on finding shapes using `OpenCV` claims to be a one-weekend introduction into the topic of computer vision [36]. As pattern recognition with deep-learning has become a broad field of study, it might be advisable to cooperate with research groups focusing on this subject to raise cluster shape analysis to a higher level.

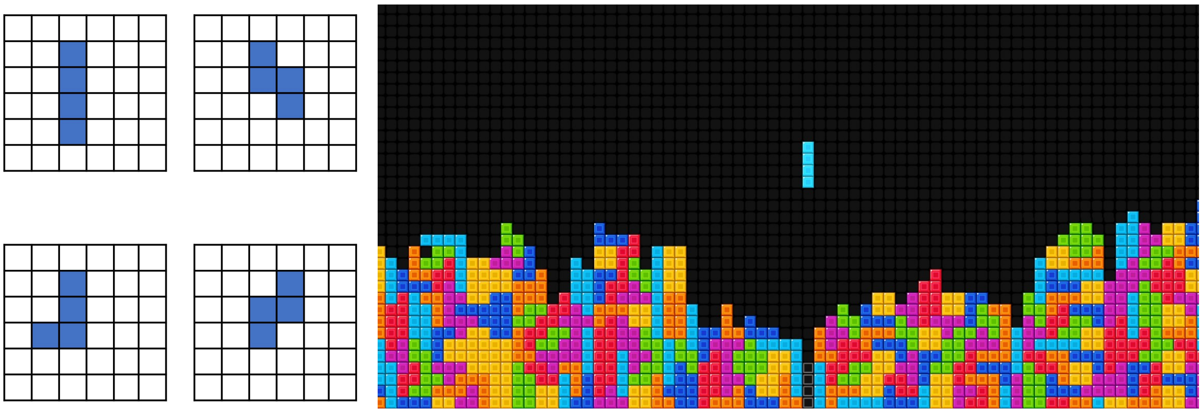
Aknowledgement

> We have to continually be jumping off cliffs and developing our wings on the way down.
> (KURT VONNEGUT, *If This Isn't Nice, What Is?: Advice for the Young*)

This thesis would have never been written without the support of the whole group of PD Dr. Jens Sören Lange. I want to thank all of you for your support. Klemens, who was setting up my laptop and two other computers and solving any of my technical problems, no matter how many hours he had to invest. Katharina who kindly provided any of her code and additionally helped to debug mine. I also want to thank Simon for sharing his office with me and constantly reminding me of thinking positive. Lastly, I am very grateful to Sören for supporting me during my "Wissenschaftliches Präsentieren"-seminar and bachelor thesis. I have never met a person explaining every topic with a patience that comes close to yours. I am also grateful to Prof. Dr. Eichner for lending me his books on multivariate analysis and inspiring discussions. Not to forget, I owe a great debt of gratitude to Katharina, Timo, Philipp and Sue for proof-reading my thesis as well as to Hans for being my second reader. My dear thank goes to Silke Kühn, my former math and physics teacher. She was to one to encourage me to change my study subject from biomedical science to physics.

Of course, I want to thank my family and childhood friends for the support that they gave me though all of the years. Thank you all, for your support, understanding and love in any stressful and frustrating parts of the last years. A special thanks goes to Timo, without whom I wouldn't be where I am now. You are the shoulder that I can cry and rely on.
Giessen definitely offers a very familiar atmosphere and excellent supervision. After three years of studying physics, I am proud to finally finish my bachelor's thesis. Holding a degree in physics is not a personal achievement of mine, but an achievement of a whole cohort of students. I would have never been as successful without all of the people that became my dear friends in the last years. After all, we made it and more importantly, we made it together.

> You can't depend on your eyes when your imagination is out of focus.
>
> (Mark Twain, *A Connecticut Yankee in King Arthur's Court*)



(a) Randomly chosen cluster

(b) Popular video game [37]

Figure 7.1: The pictures shown in this thesis are products of scientific research and are in no way intended to represent any real individual, company, product, or event, unless otherwise noted.

Declaration of Authorship/ Selbstständigkeitserklärung

In the following the author declares, that any of the work presented is the author's intellectual property if not otherwise indicated.

Hiermit versichere ich, die vorgelegte Thesis selbstständig und ohne unerlaubte fremde Hilfe und nur mit den Hilfen angefertigt zu haben, die ich in der Thesis angegeben habe. Alle Textstellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen sind, und alle Angaben, die auf mündlichen Auskünften beruhen, sind als solche kenntlich gemacht. Bei den von mir durchgeführten und in der Thesis erwähnten Untersuchungen habe ich die Grundsätze guter wissenschaftlicher Praxis, wie sie in der "Satzung der Justus-Liebig-Universität zur Sicherung guter wissenschaftlicher Praxis" niedergelegt sind, eingehalten. Gemäß § 25 Abs. 6 der Allgemeinen Bestimmungen für modularisierte Studiengänge dulde ich eine Überprüfung der Thesis mittels Anti-Plagiatssoftware.

....

Datum Unterschrift

List of Figures

List of Tables

Bibliography

[1] K. Dort. Search for Highly Ionizing Particles with the Pixel Detector in the Belle 2 Experiment, 2019.

[2] M. Karliner and J. L. Rosner. Discovery of doubly-charmed Ξ_{cc} baryon implies a stable $(b\bar{u}\bar{d})$ tetraquark.

[3] M. Gell-Mann. A schematic model of baryons and mesons. In *Murray Gell-Mann*, pages 151–152. World Scientific, feb 2010.

[4] G. Wolschin. Den Multiquarks auf Der Spur. *Spektrum der Wissenschaft 9.16*, 2016.

[5] J. S. Lange. XYZ states: Experimental observation of new narrow states with heavy quarks, 2013.

[6] E. Braaten. Seminar: Charmed Exotics. 2009.

[7] S. Käs. Exotics: Heavy penta- and tetraquarks. 2019.

[8] J. S. Lange. X, Y und Z, 2014.

[9] D. Y. Kim. The Software Library of the Belle 2 Experiment. Jul 2014.

[10] T. Geßler. *Development of FPGA-Based Algorithms for the Data Acquisition of the Belle 2 Pixel Detector*. PhD thesis, 2015.

[11] T. Abe et al. Belle 2 Technical Design Report.

[12] Unknown. Super KEKB and Belle 2, 2019.

[13] A. Moll. Comprehensive study of the background for the Pixel Vertex Detector at Belle 2. Juli 2015.

[14] K. V. Mardia et al. *Multivariate Analysis*. Propability and Mathematical Statistics. A Series of Monographs and Textbooks. Academic Press (London) Ltd., 1979.

[15] G. A. F. Seber. *Multivariate Oberservations*. Probability and Mathematical Statistics. John Wiley & Sons, Inc., 1984.

[16] K. Adachi. *Matrix-Based Introduction to Multivariate Data Analysis*. Springer Nature Singapore Pte Ltd., 2016, corrected publication 2018.

[17] T. Ahrens et al. *Mathematik*. Springer Spektrum, 3rd edition, 2015.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[19] Scikit learn developers. sklearn.preprocessing.StandardScaler. standardscaler.

[20] M. Otto. *Rechenmethoden für Studierende der Physik im ersten Jahr*. Spektrum Akademischer Verlag Heidelberg, 2011.

[21] C. B. Lang and N. Pucker. *Mathematische Methoden in der Physik*. Spektrum Akademischer Verlag Heidelberg, 2nd edition, 2010.

[22] Ilona Leyer and Karsten Wesche. *Multivariate Statistik in der Ökologie. Eine Einführung*. Springer-Verlag Berlin Heidelberg, 2007, corrected publication 2008.

[23] J. N. R. Jeffers. Two Case Studies in the Application of Principal Component Analysis. *Applied Statistics*, 16(3):225, 1967.

[24] S. Holmes P. Diaconis, S. Goel. Horseshoes In Multidimensional Scaling And Local Kernel Methods. *The Annals of Applied Statistics*, 2(No. 3, 777–807):777–807, 2008.

[25] K. Dort. Search for Highly Ionizing Particles with the Pixel Detector in the Belle 2 Experiment, 2019.

[26] T. Rashid. *Neuronale Netze selbst programmieren: Ein verständlicher Einstieg mit Python*. Heidelberg, 1. auflage edition, 2017.

[27] C. C. Aggarwal. Neural Networks and Deep Learning : A Textbook, 2018.

[28] R. Westermann. Vorlesung: Neuroanatomie. 2016.

[29] Quintana-Duque et al. Phase registration improves classification and clustering of cycles based on self-organizing maps, 2015.

[30] G. Vettigli. Minisom.

[31] W. Ertel. *Grundkurs Künstliche Intelligenz*. Springer Fachmedien Wiesbaden, 2013.

[32] S. Raschka. Principal Component Analysis in Python.

[33] W. R. Leo. *Techniques for Nuclear and Particle Physics Experiments*. Springer Berlin Heidelberg, 1994.

[34] Joachim Ohser. Angewandte bildverarbeitung und bildanalyse : Methoden, konzepte und algorithmen in der optotechnik, optischen messtechnik und industriellen qualitätskontrolle : mit 121 bildern, 147 beispielen und 35 aufgaben, 2018.

[35] Wilhelm Burger and Mark James Burge. *Digitale Bildverarbeitung*. Springer Berlin Heidelberg, 2015.

[36] A. Rosenbrock. Finding Shapes in Images using Python and OpenCV.

[37] Tetris Colorful Blocks Cubes HD Wallpapers.