Univerzita Palackého v Olomouci
Přírodovědecká fakulta
Katedra experimentální fyziky

DIPLOMOVÁ PRÁCE

Rychlá simulace šíření čerenkovských fotonů optickou částí detektoru AFP

Vypracovala:	**Bc. Kateřina Jiráková**
Studijní program:	N1701 Fyzika
Studijní obor:	Aplikovaná fyzika
Forma studia:	Prezenční
Vedoucí diplomové práce:	Mgr. Tomáš Sýkora, Ph.D.
Termín odevzdání práce:	květen 2017

Prohlášení

Prohlašuji, že jsem předloženou diplomovou práci vypracovala samostatně pod vedením Mgr. Tomáše Sýkory, Ph.D. a že jsem použila zdrojů, které cituji a uvádím v seznamu použitých pramenů.

V Olomouci dne 22. května 2017

.....

Bc. Kateřina Jiráková

Poděkování

Chtěla bych poděkovat Mgr. Tomáši Sýkorovi, Ph.D. za jeho rady, odborné vedení a kontrolu této práce.
Děkuji Mgr. Liboru Nožkovi, Ph.D. za to, že mě naučil používat simulaci v Geantu, abych si mohla sama generovat data pro neuronové sítě. Jsem mu také vděčná za jeho ochotu implementovat všechna rozšíření, která jsem potřebovala k hladkému průběhu své práce.
Všem členům skupiny částicové fyziky SLO UPOL také patří můj dík za jejich podporu, cenné rady a náměty.
Děkuji také poskytovatelům obou grantů, z nichž byl podporován můj výzkum, a to Nadačnímu fondu UP a internímu grantu Univerzity Palackého (IGA_PrF_2017_005).

A v neposlední řadě patří poděkování mé rodině a přátelům, kteří mně po celou dobu studia podporovali jak slovem, tak i mnoha skutky a jen s jejich pomocí jsem se dostala tak daleko. Moc vám za to děkuji!

Bibliografická identifikace

Jméno a příjmení autora	Bc. Kateřina Jiráková
Název práce	Rychlá simulace šíření čerenkovských fotonů optickou částí detektoru AFP
Typ práce	Diplomová
Pracoviště	Katedra experimentální fyziky
Vedoucí práce	Mgr. Tomáš Sýkora, Ph.D.
Rok obhajoby práce	2017
Abstrakt	Detektor Time-of-Flight (ToF) AFP, který byl čerstvě nainstalován na urychlovač LHC v CERNu, se zabývá měřením času příletu protonů rozptýlených pod malými úhly při srážkách na LHC. Optická část ToF AFP se skládá ze 16 tyčinek z taveného křemene ve tvaru písmene "L", tzv. LQbarů. Osm LQbarů má navíc funkční vylepšení zvané "taper". Průchod vysoce relativistického protonu tyčinkami v nich způsobí vznik čerenkovského záření, které se dále šíří tyčinkou totálními odrazy až do speciálního mnohokanálového fotonásobiče. Cílem této práce bylo vytvořit počítačový model popisující průchod fotonů záření LQbary, a to jak s taperem, tak bez něj, s ohledem na požadavek vysoké časové přesnosti. Svá řešení porovnám s již existující, tzv. plnou simulací, v rozšířeném nástroji částicových fyziků zvaném GEANT4.
Klíčová slova	LHC, AFP, ToF, detektor času letu, čerenkovská simulace, čerenkovské záření, difrakční proton, urychlovač, částice, LQbar, neuronová síť, perceptron
Počet stran	79
Počet příloh	2
Jazyk	anglický

Bibliographical identification

Autor's first name and surname	Bc. Kateřina Jiráková
Title	Fast simulation of Cherenkov photons transmission through the optical part of the AFP detector
Type of thesis	Master
Department	Department of Experimental Physics
Supervisor	Mgr. Tomáš Sýkora, Ph.D.
The year of presentation	2017
Abstract	The main goal of the Time-of-Flight (ToF) AFP detector which has been recently mounted at the LHC accelerator in CERN is to measure arrival time of protons scattered by small angles during collisions at the LHC. The optical part of the ToF AFP consists of 16 "L-shaped" bars made of fused silica called LQbars. Eight of them have a special functional improvement called "taper". Passage of highly relativistic proton through bars causes them to emit Cherenkov radiation that travels through the bar by total reflections into the special multichannel photomultiplier. The aim of this thesis was to implement computer model describing passage of photons of radiation through LQbars with and without taper with regard to the requirement of high time precision. My solutions will be verified with existing so-called full simulation in the widespread tool of particle physicists called GEANT4.
Keywords	LHC, AFP, ToF, time of flight detector, Cherenkov simulation, Cherenkov radiation, diffractive proton, accelerator, particle, LQbar, neural network, perceptron
Number of pages	79
Number of appendices	2
Language	English

Contents

Symbols, Notation and Acronyms

θ_{CH}	the characteristic angle under which the Cherenkov radiation is emitted (Cherenkov angle)
θ_{C}	critical angle of the incident ray
n	refractive index
λ	wavelength of light
β	relativistic factor; $\beta = \frac{v}{c}$, where v is velocity and c is the speed of light
μ	parameter of the Poisson distribution
LQbar or bar	"L - shaped" fused silica bar (16 of them form optical part of the ToF detector)
train	four LQbars in the row
AFP	forward detector called Atlas Forward Proton
ToF	Time of Flight detector
ANN	artificial neural network
NN	neural network
PMT	photomultiplier
MCP-PMT	micro-channel plate photomultiplier
ID	identification number

Introduction

Detector AFP or unabbreviated **A**tlas **F**orward **P**roton is one of four forward detectors of the ATLAS experiment which is among largest apparatus at the accelerator LHC (Large Hadron Collider) in CERN. AFP's stations are placed 205 m and 217 m in both directions from ATLAS interaction point (IP). Positions of forward detectors mounted in the forward area of ATLAS are schematically illustrated in the Figure 1.

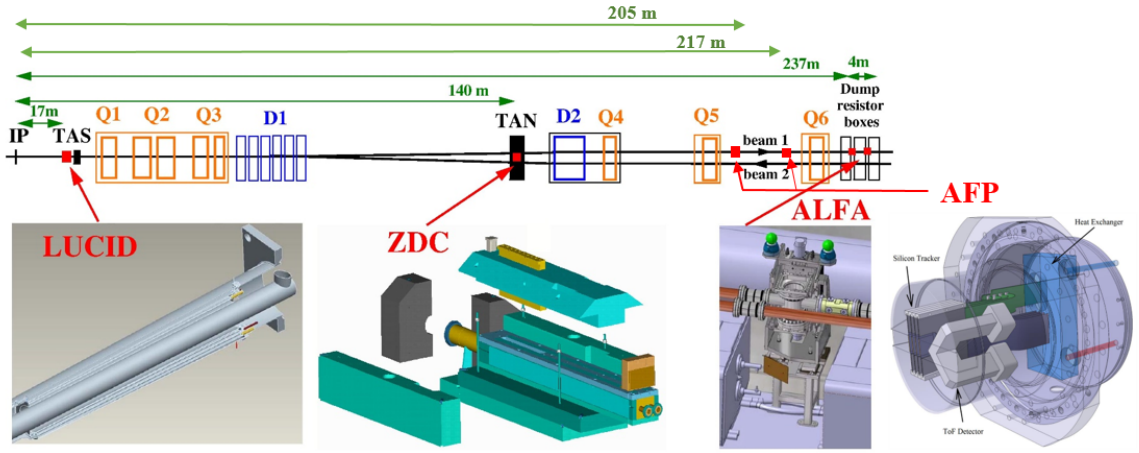


Figure 1: ATLAS forward detectors. Position of each is marked by the small red rectangle [8].

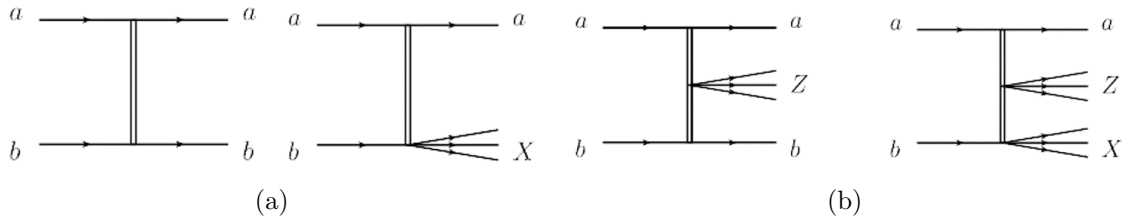
The closest forward detector to the ATLAS IP is LUCID (Luminosity measurements Using Cherenkov Integrating Detector). It uses aluminium tubes filled with gas (C_4F_{10}) as Cherenkov radiators and is dedicated to monitor LHC's instantaneous luminosity [26]. The second closest detector is the ZDC (Zero Degree Calorimeter). It detects neutral forward particles at zero degrees with respect to the beam line using quartz fibres which generate Cherenkov light. ZDC serves for heavy ion studies and it is able to count neutral particles and measure their energy [26].

The detector ALFA (Absolute Luminosity For Atlas) consists of scintillating fibres as well. They enable detection of protons scattered during proton-proton elastic and diffractive interactions in the IP and these measurements are used for determination of absolute luminosity and total cross section of pp interactions.

The AFP's objective is very similar to the ALFA's one: to observe protons that are scattered during collisions by a very small angle (max hundreds of μrad). Such protons originate from so-called *diffractive processes* (Figure 2) in which one or both protons survive the collision and continue moving very close to the beam so that they cannot be measured in the ATLAS detector but a dedicated forward detector is needed.

Products of proton dissociations (labelled by the letter X in Figure 2) are detected in the ATLAS IP.

During a diffractive event a proton emits a virtual colourless object called *pomeron* (marked by double line in the Feynman diagrams in the Figure 2). Some of events (not all) measurable by the AFP are visualised in the following Feynman diagrams.



(a) (b)

Figure 2: (a) Elastic event (on the left) and single diffractive dissociation on the right. (b) Central diffraction with one or two protons surviving [25].

Other processes, the AFP aims to measure, are exclusive production of dijet and a two-photon production of W pair (Figure 3).

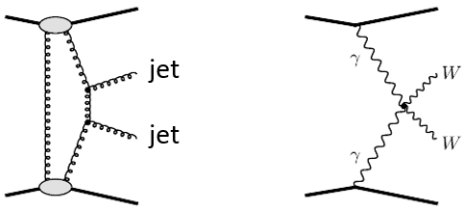


Figure 3: On the left there is an exclusive production of dijet where the "curly" lines represent gluons and the "bulb" stands for more complicated process. On the right there is a two-photon production of W pair [16].

The AFP consists of two detector parts: the silicon tracker (pixel detector) and the Time-of-Flight (ToF) detector. Both of them are mounted in a so-called Roman Pot (Figure 4), a device enabling precise movement in very close distances to the main beam of protons.

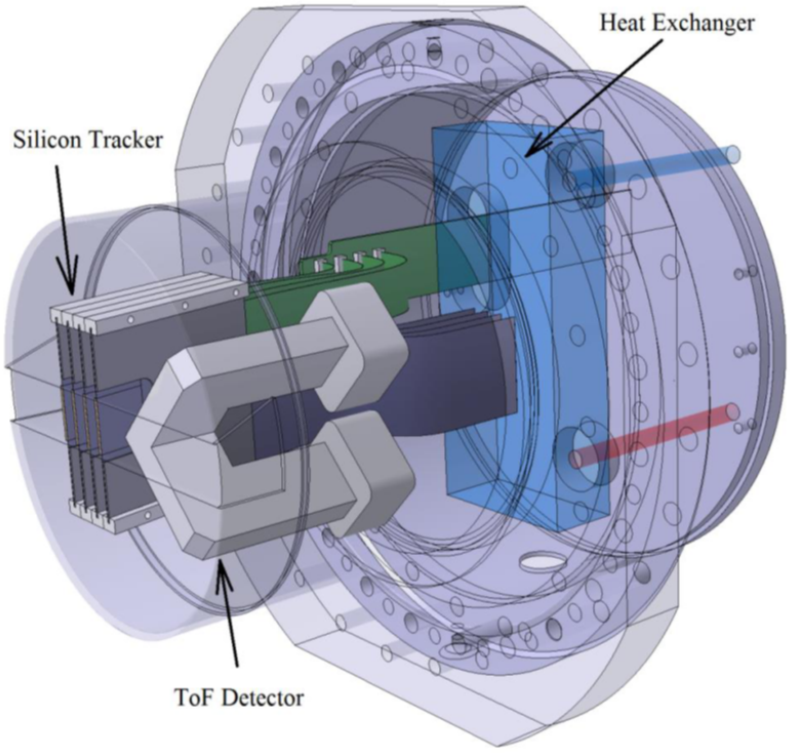


Figure 4: The Roman Pot with the AFP detector inside [16].

This thesis focuses mainly on matters connected with ToF, so the pixel detector is not discussed. Details of its function and construction may be found in [16]. In what follows we will focus on the ToF its purpose within the AFP and principle of its function.

The LHC proton beam is not a homogeneous and continuous flux of protons. Protons are rather organised into groups called *bunches*. In the interaction points along the LHC (such as the ATLAS IP) these bunches are crossed (so-called *bunch-crossing*) and protons collide with each other. Interactions within one bunch-crossing do not occur in a single infinitely small region (Figure 5), but the area is rather smeared in the z-axis that is along the accelerator (usually several centimetres). This phenomenon is called *in-time pile-up*. The main objective of ToF is to suppress such a pile-up.

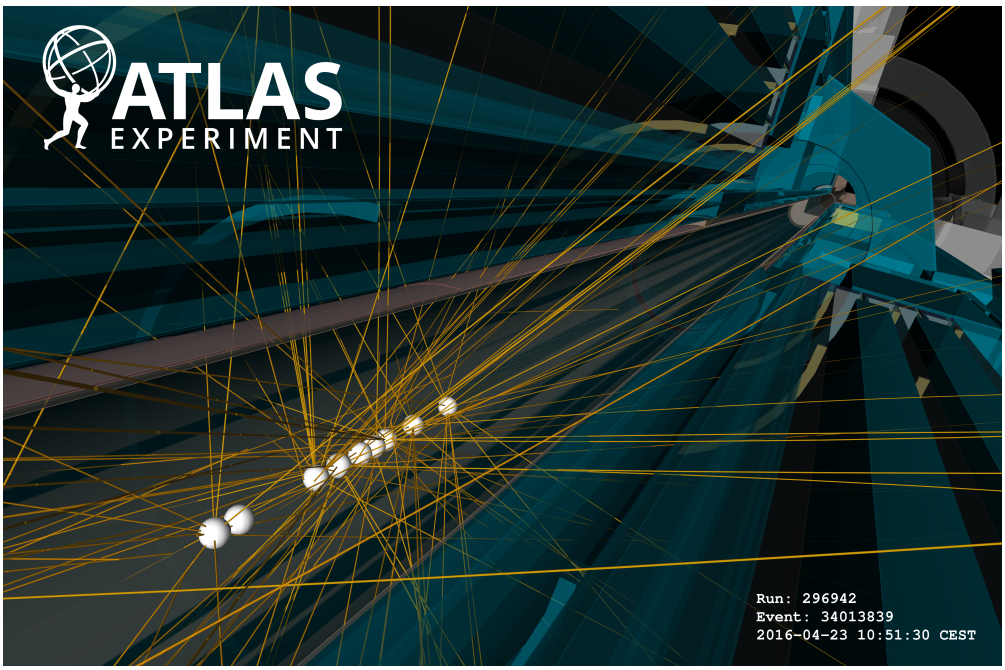


Figure 5: An interaction region within the ATLAS detector. Credits: ATLAS Experiment © 2016 CERN

Each "ball" in the Figure 5 stands for a single *pp* interaction vertex. In the Figure we can see 6 events occurring in the ATLAS IP, although in the high luminosity runs there are actually many more of them (tens - hundreds).

There is one more phenomenon, a smearing in time - an *out-of-time pile-up* which is formed by collisions occurring in previous or subsequent bunch-crossing than the collision of interest [21]. Frequency of bunch-crossings on the LHC is set on 25 ns. It puts a condition on detectors and their read-out electronics to be able to read fast enough to differentiate from what bunch-crossing detected products originate.

From now we focus only on the in-time pile up. It causes that it is hard to differentiate proton interactions or their products from each other within a bunch-crossing. In the context of diffractive physics we need to be able to differentiate if protons measured by the AFP originate from the same vertex and thus are among the diffractive events of interest or came from different vertices and therefore form a background. Both situations are depicted in the Figure 6.

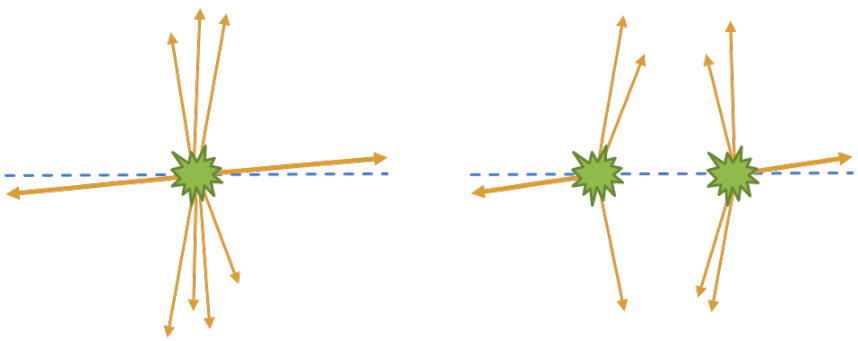


Figure 6: In the left there is a diffractive event with two protons and two jets emitted whereas in the right there are two background vertices. Courtesy of Ladislav Chytka.

The idea of the ToF functionality is schematically pictured in the Figure 7:

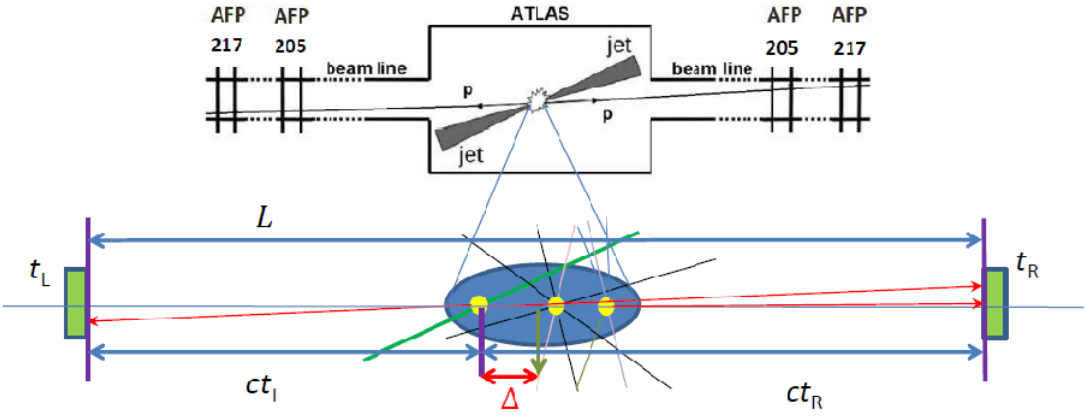


Figure 7: The upper figure shows an interaction region within the ATLAS detector. The lower picture zooms in the interaction and shows the pile-up: three interactions are happeaning in the same time. (Upper figure [24], lower figure courtesy of Tomas Sykora.)

In the ATLAS IP (circled by the blue colour in the lower part of the picture) there occurs three interactions. One of them is a diffractive event of interest with two surviving protons (red lines in the lower part of the figure) and two jets (green line). The ToF detector (purple line in front of the green rectangle) measures the time of arrival of both protons on the left and right side from the ATLAS, t_L and t_R. These times are measured by local clocks that are synchronised with respect to the reference clock which is again synchronised with bunch-crossings. So the time of interaction is well defined. The distance L between both AFP stations is known so by calculating the length of each proton trajectory, $c \cdot t_L$ and $c \cdot t_R$, we obtain the length difference Δ. Its magnitude gives evidence if protons came from the same vertex or not.

This mechanism will work only if the time resolution of the ToF detector is sufficiently high. The goal was to reach approximately 10 ps which poses in terms of proton trajectory length 3 mm ($= 3 \cdot 10^8 [ms^{-1}] \cdot 10 \cdot 10^{-12} [s]$). This practically means that it would be possible (if the 10 ps is reached) to measure the position of the interaction point with precision of $\frac{3}{\sqrt{2}}$ mm.

The detection of protons starts in 16 "L-shaped" bars made of fused silica, a Cherenkov active medium. The Cherenkov radiation is emitted when the scattered proton passes the bars. The theory of Cherenkov light is partially covered in the first chapter. The details of bars construction and electronics parts of ToF are in details discussed in Chapter 2 of this thesis.

For measurements of ToF a fast and accurate simulation of time distribution is needed. Creating such a simulation is the subject of my thesis; two methods to achieve it were adapted.

The first one follows the logic presented in [18] whose basis was purely geometrical. This approach is presented in Chapter 4. Nevertheless, this model did not describe the testing data generated by the full simulation - *Geant4* well. Thus the second approach was invented. It is based on usage of artificial neural network (ANN) whose fundamental principles is devoted Chapter 3. The idea was to use ANN for the regression of the Cherenkov photon trajectory length distribution. The implementation of the Fast Cherenkov simulation using neural networks is discussed in Chapter 5.

The Appendix A includes a data sheet of suprasil UVL, a fused silica from which some parts of LQbars have been manufactured. The Appendix B comprises a list of files of the Fast Cherenkov simulation as well as other contents of the CD-ROM enclosed to the printed version of this thesis.

Chapter 1

Cherenkov Radiation and its Production in LQBars

1.1 Basic Theory

The Cherenkov effect is analogical to the well-know sonic boom effect that occurs when an aircraft or a bullet for instance exceeds the speed of sound in the air [11]. This causes creation of a sonic shock wave that propagates in a characteristic cone (Figure 1.1). When the aircraft passes through the points 4 to 0 it creates sound waves. These waves add together to create a plain wave that propagates in the direction of arrows.

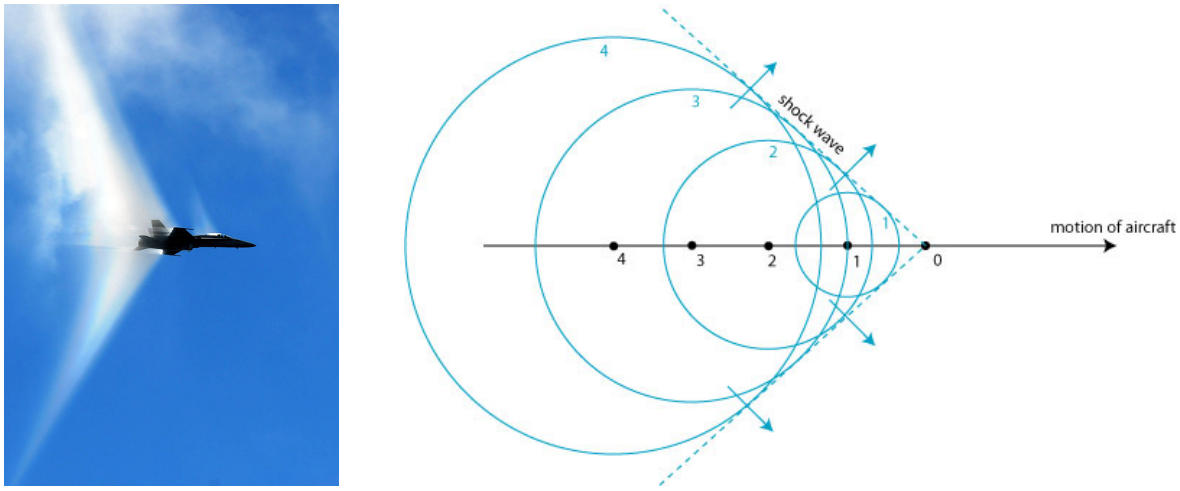


Figure 1.1: (a) An aircraft is creating a sonic boom [2]. (b) A scheme of soundwave propagation [5].

Cherenkov radiation occurs when a velocity v_{particle} of a charged particle exceeds in some medium the speed of light in that medium v_{light}. So these inequalities hold:

$$v_{\text{particle}} > v_{\text{light}} \Rightarrow \tag{1.1}$$

$$\Rightarrow \beta c > \frac{c}{n} \Rightarrow \tag{1.2}$$

$$\Rightarrow \beta > \frac{1}{n}\,, \tag{1.3}$$

where $\beta(= \frac{v_{\text{particle}}}{c})$ is a relativistic factor of incidental particle and n is a refractive index of the given medium. We see that if refractive index is fixed there exist a minimal value

of β, say $\beta_{\text{threshold}}$, for which the particle will emit Cherenkov radiation:

$$\beta_{\text{threshold}} = \frac{1}{n_{\text{threshold}}}. \tag{1.4}$$

To be able to detect this radiation the medium must be transparent for light such as ice, water, glass, atmosphere etc.

Cherenkov radiation has an electromagnetic nature and therefore can be generated only by a charged particle in a dielectric medium. The reason for its generation is following: When a charged particle moves through a dielectric medium it polarises its atoms, so the electron gas surrounding the atomic nuclei excites to the higher energetic states. Thus electrons are displaced from the nuclei and dipoles are formed. Then these dipoles collapse and emit electromagnetic radiation. In case the initial particle moves slowly ($v_{\text{particle}} < v_{\text{light}}$) the field of dipoles is perfectly symmetrical with respect to the position of the particle and the resultant radiation neutralises (Figure 1.2(a)). Whereas if the particle is highly relativistic ($v_{\text{particle}} > v_{\text{light}}$) delayed polarisation is observed. The dipoles organise into a cone (Figure 1.2(b)) and emit radiation perpendicularly to this cone surface.

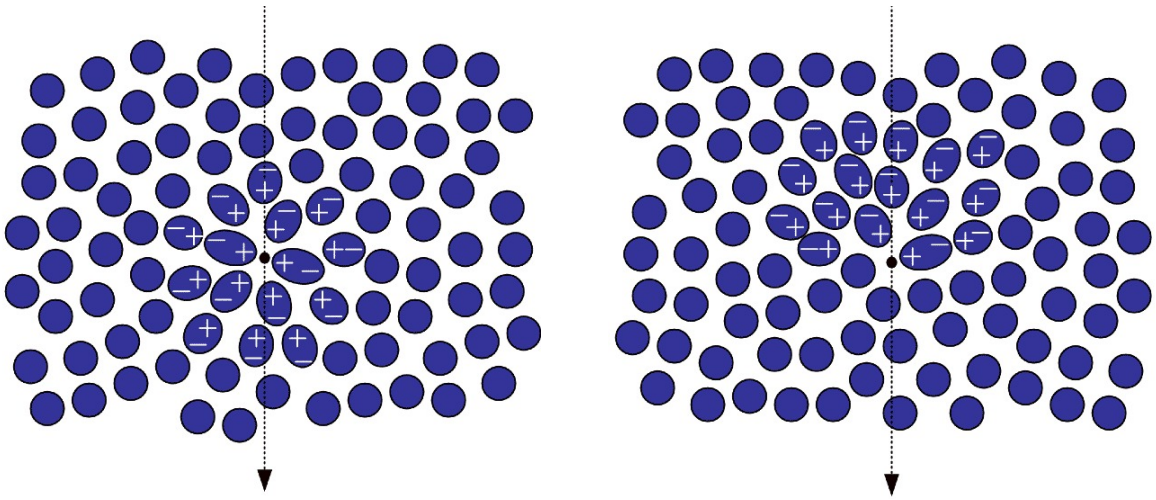


Figure 1.2: Polarisation of dielectric medium by a charged particle which is slower (a) or faster (b) than the speed of light in that medium [1].

Spectrum of wavelengths of Cherenkov radiation is given by the equation (1.6), so the maximum count of photons belongs to near ultraviolet part. Photons are generated in a cone (Figure 1.3) defined by a Cherenkov angle θ_{CH} which is described by an equation [18]:

$$\cos\theta_{\text{CH}} = \frac{1}{\beta n}. \tag{1.5}$$

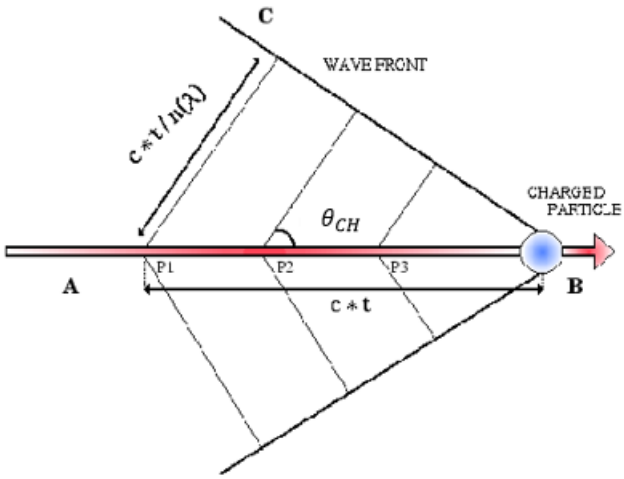


Figure 1.3: The characteristic cone of the Cherenkov radiation [4].

It is worth stressing that n is a function of wavelength of light λ; $n = n(\lambda)$. Therefore θ_{CH} is different for photons with different wavelength.

The amount of emitted Cherenkov photons $\mathrm{d}N$ per unit interval of distance $\mathrm{d}x$ and per unit energy $\mathrm{d}E$ is described by Frank-Tamm formula that takes form [3]:

$$\frac{\mathrm{d}^2N}{\mathrm{d}x\mathrm{d}E} = \frac{\alpha^2 Z^2}{r_e m_e c^2}\left(1 - \frac{1}{\beta^2 n^2(E)}\right) , \tag{1.6}$$

$$\frac{\mathrm{d}^2N}{\mathrm{d}x\mathrm{d}E} \sim 370 Z^2 \left[\frac{photons}{eV \cdot cm}\right]\left(1 - \frac{1}{\beta^2 n^2(E)}\right). \tag{1.7}$$

Z^2 is a multiplicity of particle's charge (a charge of particle q equals eZ), α is the fine-structure constant ($= \frac{1}{137}$), r_e and m_e are classic radius and mass of an electron. We use natural units, so e equals one. When applying equation (1.5) we can simplify the relation and get:

$$\frac{\mathrm{d}^2N}{\mathrm{d}x\mathrm{d}E} = 370 Z^2 \sin^2\theta_{\mathrm{CH}}. \tag{1.8}$$

For the description of photon's production the Poisson distribution may be used whose probability mass function P of discrete random variable X is given by:

$$P(X = k) = \frac{\mu^k e^{-\mu}}{k!}, \tag{1.9}$$

where μ is a parameter and $k = 0, 1, 2, \ldots$ Poisson distribution expresses the number of occurrences of certain event in a fixed space or time if the average rate of occurrences is defined and events occur independently of time. For the Cherenkov radiation production these conditions are satisfied. As the parameter we take [18]

$$\mu = \frac{\mathrm{d}N}{\mathrm{d}x} \cdot StepLength \tag{1.10}$$

where $StepLength$ is the length of proton trajectory in the Cherenkov active material. After inserting equation (1.8) we obtain:

$$\mu = 370 \cdot Z^2 \sin^2\theta_{\mathrm{CH}} \cdot \mathrm{d}E \cdot StepLength. \tag{1.11}$$

1.2 Cherenkov Radiation Propagation in an LQBar

As mentioned earlier the optical part of the AFP ToF detector is made of "L - shaped" fussed silica bars called *LQbars* where the Cherenkov radiation is emitted. Its transmission through LQbars can be viewed in the approximation of geometrical optics. Photons are generated to form a surface of a characteristic cone and each photon can be substituted by a line. Inside an LQbar the radiation is mainly transferred by *total internal reflections* from sides of the LQbar (see Figure 1.4).

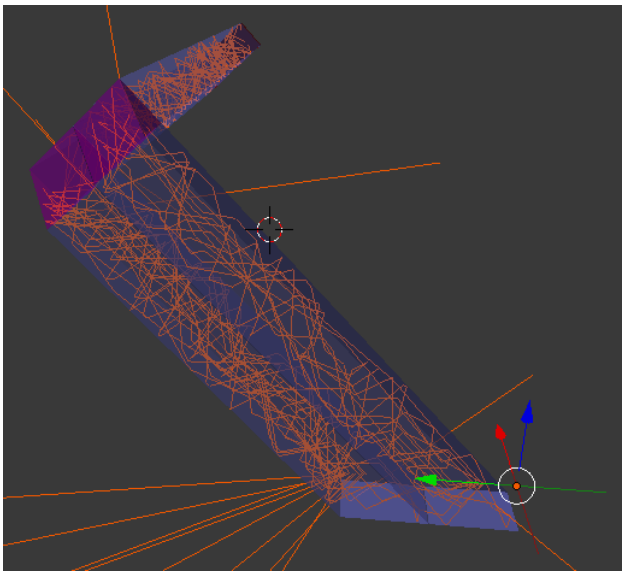


Figure 1.4: Visualisation of the Cherenkov photons (orange lines) in two LQbars. The incidental proton comes in the direction of the green base vector. (Courtesy of Tomas Komarek).

This phenomenon is in accordance with the Snell's law which characterises the behaviour of a light ray on a border between two enviroments with different refractive indices n_1 and n_2.

$$n_1 \sin\theta_1 = n_2 \sin\theta_2. \tag{1.12}$$

Angles θ_1 and θ_2 are taken with respect to the normal of boundary. When the angle of incident ray exceeds certain critical value θ_C the ray does not get to the second environment at all, but is reflected back to the material from which the ray came (Figure 1.5).

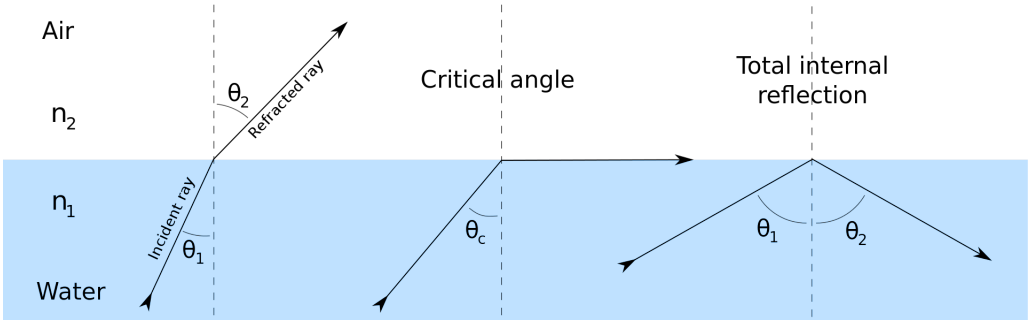


Figure 1.5: Behaviour of the incident ray on the boundary between two environments [19].

For the critical angle we can express from equation (1.12) following relationship

$$\sin\theta_{\mathrm{C}} = \frac{n_2}{n_1}. \tag{1.13}$$

If we consider the transit from glass to the vacuum and when we take into account that refractive index of the air n_2 is approximately 1 there can be derived a relationship between Cherenkov angle θ_{CH} and critical angle θ_{C}. We express from relations (1.5) and (1.13) $\frac{1}{n_1}$ and put them equal:

$$\sin\theta_{\mathrm{C}} = \beta \cdot \cos\theta_{\mathrm{CH}} \ , \tag{1.14}$$

$$\theta_{\mathrm{C}} = \arcsin(\cos\theta_{\mathrm{CH}}) \ . \tag{1.15}$$

For highly relativistic particles the factor $\beta \sim 1$.

Chapter 2

The AFP ToF Detector

2.0.1 Construction of the Optical Part of the AFP ToF

The optical part of the ToF detector is composed of 16 "L-shaped" bars made of suprasil (synthetic fused silica) called *LQbars*. They are bound in a holder to be in direct contact with the photomultiplier (Figure 2.1).

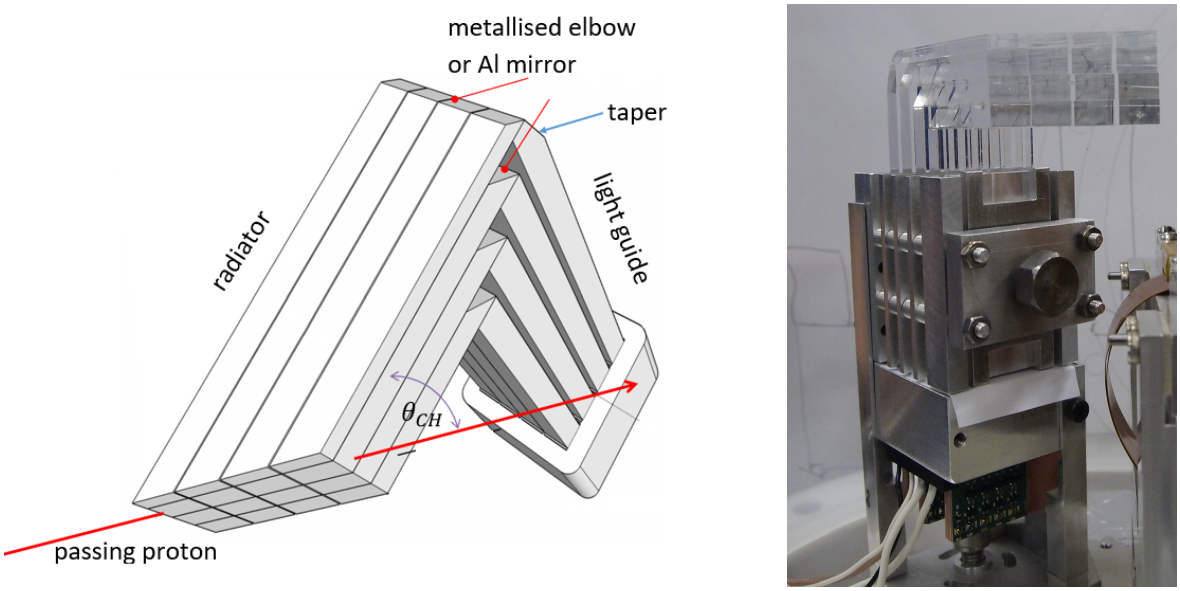


Figure 2.1: (a) Sketch of the optical part of ToF. Proton passes through the second train. (b) A photo of the optical part of ToF mounted in the holder.

The set of bars in the horizontal row is called a *train*. There are obviously 4 trains; train 1 is the most upper one and the train 4 is the lowest one. When the scattered proton comes through horizontal part of the bars so-called *radiator* it causes creation of the Cherenkov radiation. The light than travels through the bars mainly via total internal reflections to the vertical part of the bars called *light guide* which is glued to the radiator. Under the light guide there is a photomultiplier. Each LQbar has also a metallised elbow covered with aluminium that has a function of a mirror [22].

Incoming protons enter ToF practically parallel (deviations are few hundreds of μrad) to the edge of bars as visible from the Figure 2.1. The edge of every bar is cut under Cherenkov angle θ_{CH} which is for fused silica $\sim 48°$. This cut causes that vast

majority of emitted Cherenkov photons go straightly with minimal number of total reflections to the metallised elbow and to the photomultiplier.

The design of bars is such that radiators in a train are not off the same length. The proton first enters the longest one and continues to the shortest one. This has the effect that photons created along the whole proton path emitted in the same direction arrive to the photomultiplier in the same time (time *compensation*).

When photons do not satisfy the condition of total reflection or due to probability of reflectance from the boundary between bar and the air they escape from the bar. It may happen that they get to some other bar where they will reflect and such photons are detected in different bar than they came from. This is called *optical crosstalk* and has negative influence on the time resolution of bars [20]. Another type of crosstalk is also present, an *electronic* crosstalk, connected with the inner arrangement of the photomultiplier. The resolution of single bar is under 30 ps [23]. When having 4 bars in a row (a train) the resolution is ideally $\sim$ 15 ps if bars are uncorrelated [23].

For the train 1 and later train 2 there was developed a certain improvement in design called *taper angle* [22]. It is another cut situated just under the metallised elbow (Figure 2.2(a)). The purpose of the taper is to straighten the photon trajectories and thus detect them in the photomultiplier in the shortest possible time (Figure 2.2(b)). Another reason is to be able to detect photons that does not fall on the elbow to reflect but just under it. For many of them the condition of total reflection is not satisfied and therefore they are lost.

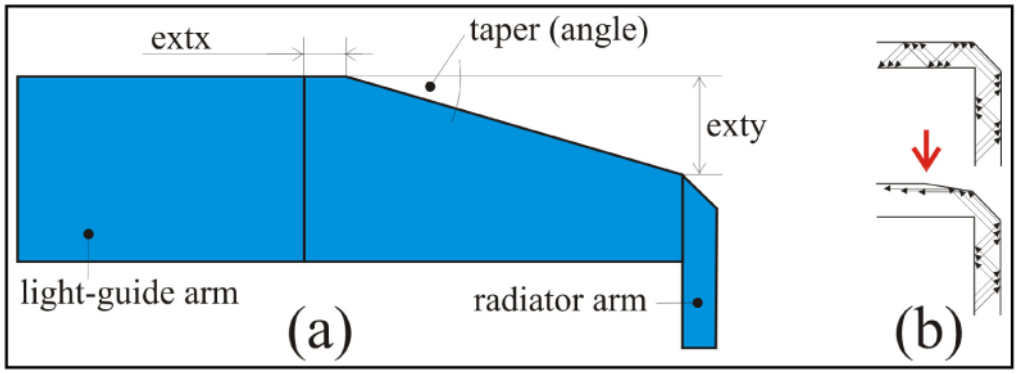


Figure 2.2: A scheme of taper [22]

The train 1 has one extra feature; its radiator's width is only 2 mm to decrease the possibility of multiple proton hits.

2.0.2 Electronics Part of the AFP ToF

The simplified scheme of the AFP ToF prototype is introduced in this subsection (see Figure 2.3).

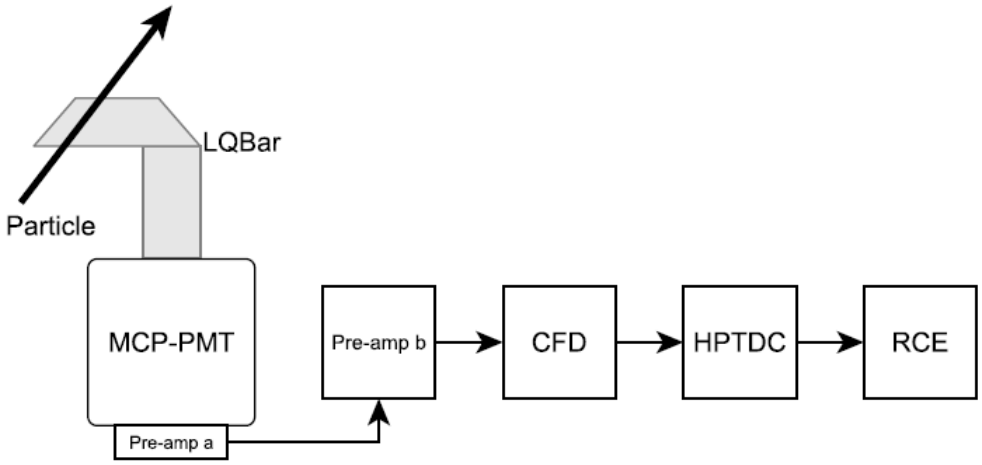


Figure 2.3: The electronic modules of the AFP ToF [20].

Cherenkov photons from the light guide get into the photomultiplier. We use the miniPLANACON multi-channel plate photomultiplier (MCP-PMT) by Photonis company[1]. The light is transformed here into an electrical pulse from whose falling edge the time information can be derived. The statistics of incoming photons influences the amplitude of the signal not its shape [16].

The output signal is amplified in two steps by preamplifiers (pre-amp "a" and "b"). Thus we get *raw signal* whose amplitude is equivalent to the number of generated photoelectrons. The next step is the CDF (constant fraction discriminator) an electronic device determining when the signal came (the time of leading edge of the signal) regardless of its amplitude. This is crucial for precise timing, because the signal coming from the MCP-PMT does not preserves amplitude but the shape of the signal. In case the time was determined when the signal reached some threshold value, resulting time would vary for signals with same shape but different amplitude which is undesirable (Figure 2.4). Whereas CFD fixes the constant fraction of the signal's peak causing the time being independent from maximal amplitude of the signal.

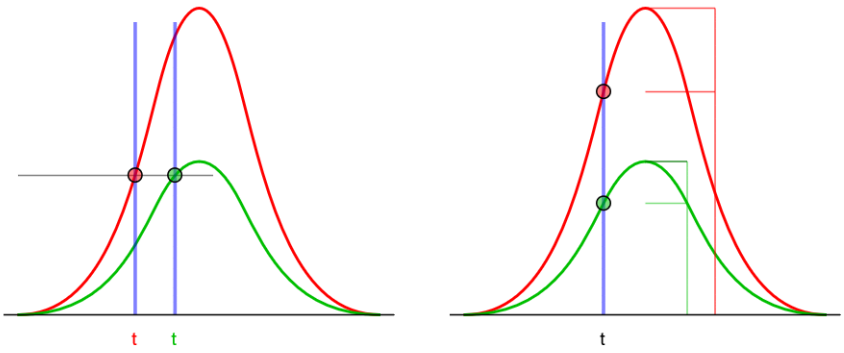


Figure 2.4: Difference between threshold-based and constant fraction based determination of time [10].

[1]Data sheet available online: https://www.photonis.com/uploads/datasheet/pd/Mini-PLANACON-4x4-datasheet.pdf

The HPTDC (High Precision Time to Digital Converter) ensures the transfer of time recording into the digital form. The last part is the RCE a system for the data acquisition.

Chapter 3

Artificial Neural Network

In this thesis we consider two options how to simulate motion of the Cherenkov radiation in the LQbars. One of them is geometrically based and will be discussed in detail in Chapter 4. This chapter is devoted to the theory of the second approach which is rather different and consists of using artificial neural network (ANN) based fitting of dedicated histograms.

3.1 General Description of Function

The artificial neural network (ANN or just NN) is a computational algorithm or model whose function is based on *artificial neurons* - simple units loosely resembling function of brain neurons and their axons. As in the brain so in the ANN neurons form a structure with large number of connections between themselves; by training (or calibrating) are these connections strengthened or weakened to get proper output from the ANN. The model is created in the way it may learn itself rather then being preprogrammed for some specific purpose. ANNs are non-linear processing systems used for vast range of tasks and can be trained for accomplishing several issues using sample data.

Neurons are organised into certain patterns serving different purposes. The basic conception of artificial neuron function is visualised in the Figure 3.1. There are five input values $x_1,...,x_5$ into a neuron while each connection has its given weight $w_1,...,w_5$. This set of weights is updated during the process of learning. The first step of the neuron is to calculate *weighted sum* ζ:

$$\zeta = \sum_{i=1}^{n=5} w_i x_i \ . \tag{3.1}$$

The second step is than to insert the value of the weighted sum into the given *activation function* $\sigma(\zeta)$:

$$y = \sigma(\zeta) \tag{3.2}$$

to get an output value y of the neuron. There are several types of activation functions and will be mentioned later.

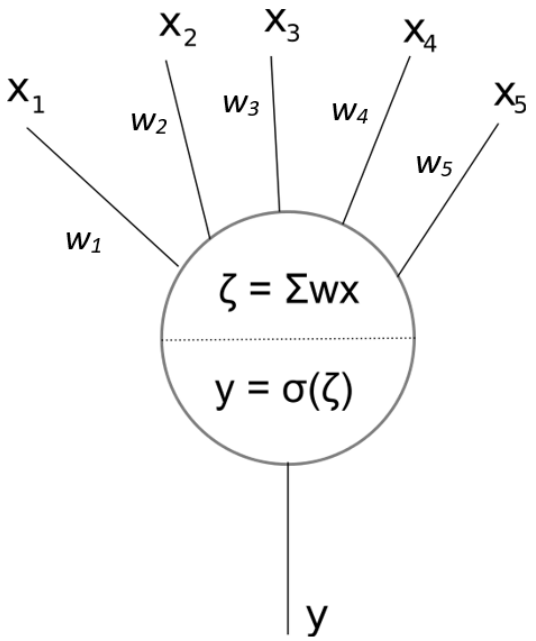


Figure 3.1: Ilustration of neuron's function [6].

The general diagram of ANN function is depicted in the Figure 3.2. It principally consists of three layers: *input layer* containing as many neurons as is the number of input variables (in the Figure 3.2 there are 4 input neurons marked by circles), one or more *hidden layers* (here it is shown 1 hidden layer consisting of 5 neurons) and an *output layer* (made of 1 neuron). ANNs tend to have for each degree of system freedom one input neuron. The same holds for the output of the ANN.

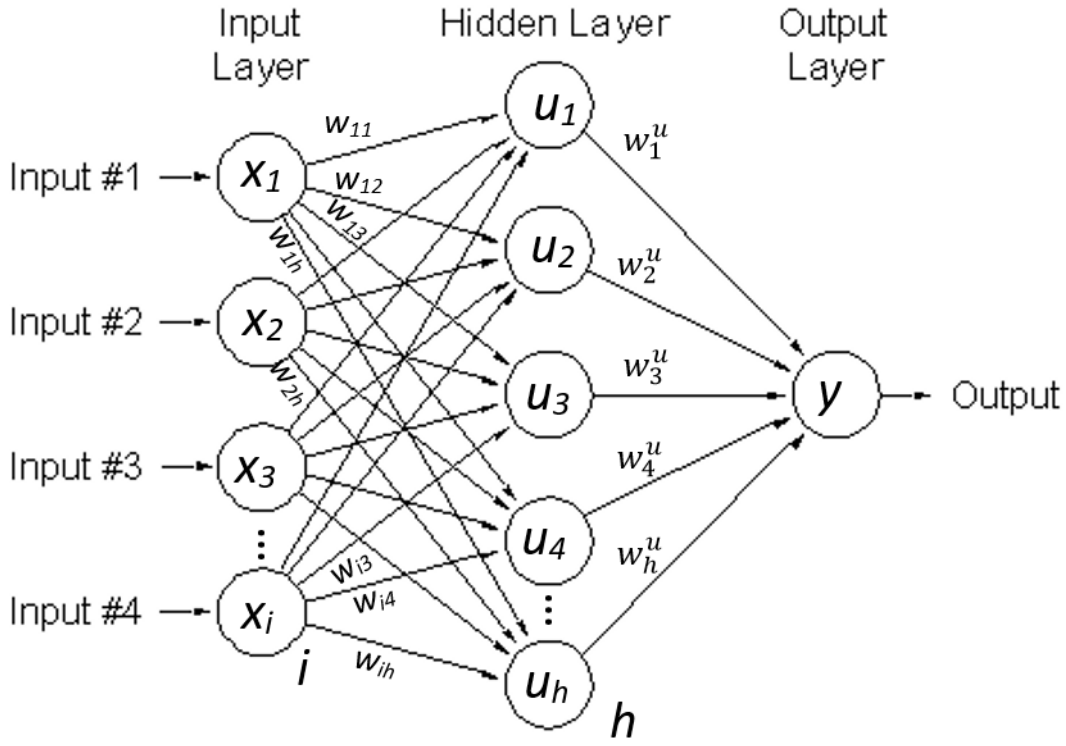


Figure 3.2: Chart of an ANN architecture [9].

The principle of the ANN function is following: There is a set of input variables $x_1, ..., x_i$ that describe a given issue and they are somehow connected with the desired output variable y or the set of variables $y_1, ..., y_m$. One input neuron stands for one input variable into the ANN. All neurons in the input layer are connected with all neurons in the first hidden layer (their number is h). Every such artificial synapse has its weight: $w_{11}, w_{12}, w_{13}, ..., w_{ih}$. Each neuron in the hidden layer calculates a weighted sum and inserts its result into the activation function as described in equations (3.1) and (3.2). This value (let us call it $u_1, ..., u_h$) is an output of each neuron in the hidden layer. All neurons in the hidden layer have weighted synapses (w_1^u, ..., w_h^u in the case of one output) to a neuron or neurons in the output layer that function in the same manner as described above. The output neuron contains weighted sum from values of all neurons in the hidden layer. By inserting the result into the equation (3.2) we get the output value y of the ANN. This was the description of so-called *feed-forward neural network*, the simplest NN containing only forward paths and no feedback. That means neurons in one layer never get values and weights from the following layers but only from the previous layers.

Generally, ANNs are used to manage two types of issues: *regression* and *classification*. The first issue is connected with description of behaviour of a continuous variable whose values are arbitrary. Classification task is to correctly predict values of a variable which can reach only from 0 to 1.

Now we render two statements justifying the usage of NNs in both upper cases. It is possible to prove that the structure with one hidden layer containing finite number of neurons can approximate any continuous function (*Universal approximation theorem*). The theorem was proved by George Cybenko in 1989 for sigmoid activation functions [15]. Another theorem that applies to ANN states: If ANN is trained with output variable that has values only 0 and 1, the approximated function of inputs X is the probability of output = 1, knowing X [7].

The ANN can be trained with the set of data where for each input the output value is specified. There are two stages that the ANN has to accomplish: *training* and *testing*. During training the algorithm goes through some part of the data and calibrates weight of each synapse. After that it tests using the rest of the data how successful the training was. This process is repeated many times; each repetition is referred to as an *epoch*.

3.2 ANN Implementation in the ROOT Framework

The ROOT is a scientific framework used mainly in the field of particle physics. It is used as an C/C++ interpreter and compiler. ROOT incorporates several C++ libraries for handling graphs, histograms, large data samples and also a library with implemented feed-forward ANN called *multi layer perceptron*. The library is represented by files `TMultiLayerPerceptron.h` and `TMultiLayerPerceptron.cxx`.
There are several types of constructor, here there is discussed only the one used during the work on the Fast Cherenkov simulation:

```
TMultiLayerPerceptron::TMultiLayerPerceptron (const char *layout,
TTree *data, const char *training = "Entry$%2==0",
const char *test = "", TNeuron::ENeuronType type = TNeuron::kSigmoid,
```

`const char *extF = "", const char *extD = "")`

The structure of the ANN form in particular input and output variables, number of hidden layers and number of neurons in these layers is defined in a string `layout` with specific structure. E.g.: `"vtxx,vtxy,vtxz,vtxphi,vtxtheta:20:@trlen"` means that expressions before the colon constitutes 5 input variables divided by commas. The number in the middle means one hidden layer containing 20 neurons. There can by of course more hidden layers; it is always necessary to divide them by a colon. The last character `@trlen` stands for an output variable that has to be normalised by the ANN. If `@` is omitted no normalisation is done. Another possibility is to finish the whole expression with `!` i.e. `"vtxx,vtxy,vtxz,vtxphi,vtxtheta:20:var1!"` to force the usage of Softmax activation function in the output layer only.

Values of input and output variables are stated in a `TTree` called `data`, a ROOT structure for storing data. In strings `training` and `test` one can specify what data will be used for the training and for testing of the NN. The specification is done via characteristic expressions using the class `TTreeFormula`. The formula `Entry$%2==0` means that half of the events will be taken for training and the second half for testing.

Expression `TNeuron::ENeuronType type = TNeuron::kSigmoid` allows to choose which activation function (in ROOT manual often referred to as *neuron type*) will be used. In the class `TNeuron` there are implemented several types like:

- `kSigmoid`

  is defined as:

  $$f(x) = \frac{1}{1 + e^{-x}} \tag{3.3}$$

  and its range is $(0; 1)$ (Figure 3.3). It is set default in the constructor.

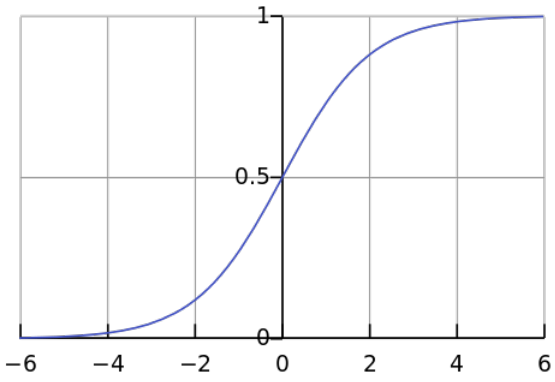


Figure 3.3: The sigmoid function [14].

- `kLinear`
- `kTanh`
- `kGauss`
- `kSoftmax`

  is used when the ANN should deal with classification task and there are more output variables.

- `kExternal`

  is defined by the user.

Before the training can start it is possible to choose a training method by calling function: `void TMultiLayerPerceptron::SetLearningMethod (TMultiLayerPerceptron::ELearningMethod method)` . There is a choice between several implemented methods:

- `kStochastic`

  turned out to markedly slow down the learning process.

- `kBatch`

- `kSteepestDescent`

- `kRibierePolak`

- `kFletcherReeves`

- `kBFGS`

  or unabbreviated the Broyden, Fletcher, Goldfarb, Shanno method is defaultly set and gives the best results.

The training itself is switched on by calling function
`void TMultiLayerPerceptron::Train(Int_t nEpoch, Option_t *option = "text", Double_t minE = 0)`.

The first argument says how many times the NN is going through the data set to learn from it, i.e. how many epochs is set. The most useful options (divided by a dash) are:

- `"text"`

  that is a text output on the screen. It describes quadratic error of training and testing.

- `"graph"`

  causes the graph of the error being plotted. It may look like the one in the Figure 3.4

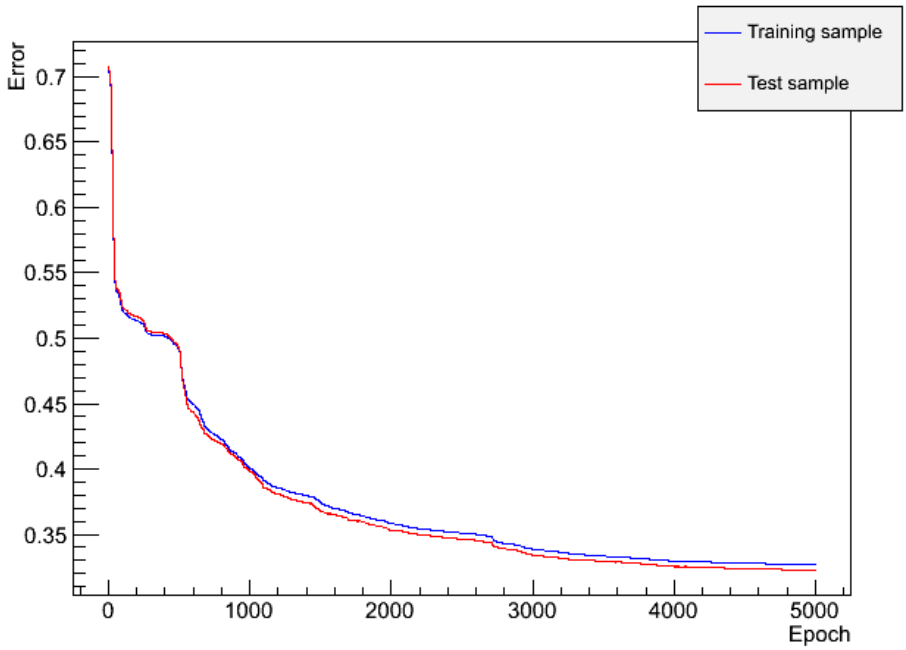


Figure 3.4: The plot of evolution of the error during the training.

- `"update=10"`

  symbolises that values of the error are updated and printed on the screen after 10 training epochs.

For instance the method `Train` could be used like this:
`p_mlp->Train(ntrain, "text,graph,update=10");` .

After the training the weights of artificial synapses can be dumped into a text file by calling method `Bool_t TMultiLayerPerceptron::DumpWeights(Option_t *filename = "weights.txt")`. This file can be used when the user decides to train the NN again. In such a case before calling `Train` it should be called method `Bool_t TMultiLayerPerceptron::LoadWeights(Option_t *filename = "weights.txt")`. The loading of weights to initialise the learning process should noticeably improve the result.

The mathematical expression of the the trained ANN network is generally a very large formula and for further use it should be exported into a standalone code. This is done by calling function:
`void Export (Option_t* filename = "NNfunction", Option_t* language = "C++")` which is a member of `TMultiLayerPerceptron` class. The programming language Python is also supported by typing option `"python"`. The function `Export` creates a class called `NNfunction` which contains a function `double nnFunction::Value(int index,double in0, double in1, ..., double in`$_i$`)`. The variable `index` expresses the numerical order of the output neurons from the ANN, so if the user wants to know the value of the first output neuron he puts `index` equal to 0. Variables `in0, ..., in`$_i$ are the input variables.

The resulting NN analysis can be visualised. It is enabled by class `TMLPAnalyzer` whose constructor is:

`TMLPAnalyzer::TMLPAnalyzer (TMultiLayerPerceptron net)` .

Firstly, the analysis must be initialised using function `void TMLPAnalyzer::GatherInformations ( )`. This method collects useful information about the network. Secondly, it is recommended to call `void TMLPAnalyzer::CheckNetwork ( )` to get information about the NN in the terminal (Figure 3.5). The most useful is the lower part of the Figure where we see the influence of each variable on the training. It helps to judge to which variables is the ANN sensitive and which of them are useful.

```
Epoch: 1970 learn=0.250335 test=0.251141
Epoch: 1980 learn=0.250197 test=0.250915
Epoch: 1990 learn=0.250111 test=0.250875
Epoch: 1999 learn=0.249984 test=0.250836
Training done.
20_20hi_5000ep_random_shorter_clas/nnFunc_20_20hi_2000ep_weight.h
 and 20_20hi_5000ep_random_shorter_clas/nnFunc_20_20hi_2000ep_wei
ght.cxx created.
Network with structure: vtxx,vtxy,vtxz,vtxphi,vtxtheta:20:20:shor
ter
inputs with low values in the differences plot may not be needed
vtxx -> 0.0154542 +/- 0.0750606
vtxy -> 0.00776529 +/- 0.0467341
vtxz -> 0.00129165 +/- 0.00749154
vtxphi -> 0.138413 +/- 0.294797
vtxtheta -> 0.00366569 +/- 0.0229456
```

Figure 3.5: The terminal output from the ANN training.

This may be visualised graphically by function `void TMLPAnalyzer::DrawDInputs ( )` . It uses the test data sample to perform small variation of the input to the NN and draws the distribution of influence of the NN output (top left part of the Figure 3.6). Such plot helps with the optimisation of the NN.

To plot the network structure the function `void TMultiLayerPerceptron::Draw ( )` is called. Neurons are marked by blue circles and width of synapses are proportional to the weights (top right part of the Figure 3.6).

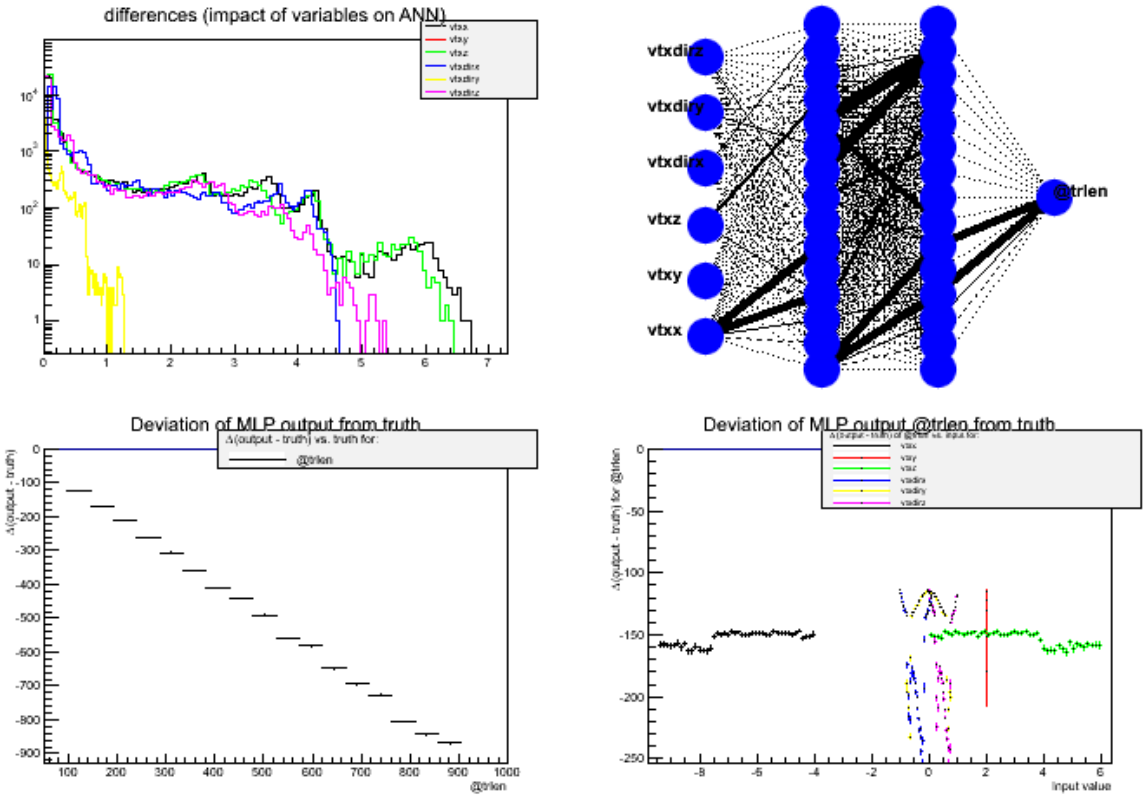


Figure 3.6: A graphical output of the ANN training.

Detailed information is given in the ROOT manual [7] and in two example macros [12] one solving a regression issue and the second dealing with classification.

Chapter 4

Geometrical Solution of Fast Cherenkov Simulation

Since GEANT4 treats each photon individually which is rather time consuming it became useful to implement a Fast Cherenkov simulation algorithm. The major idea of such simulation is to approximate each photon in terms of geometrical optics - by an effective trajectory through the LQbar. The issue is firstly treated purely geometrically. Secondly, effects like absorption of photon in the bar or on the metallised elbow, etc. are taken into account.

4.1 Basic Concept

The first step of Fast Cherenkov simulation is to calculate the length of photon trajectory in the bar. However this photon path is non-trivial as the photon reflects from the sides of the bar in 3-dimensional space. The main idea is to straighten its trajectory, take into account space angles under which the photon was radiated and calculate the trajectory length.

When the photon is generated it must be tested whether it conforms to conditions of total reflection in the bar. Therefore angles under which the photon was radiated with respect to the bar are proposed. Let us consider an element of the bar depicted in the Figure 4.1.

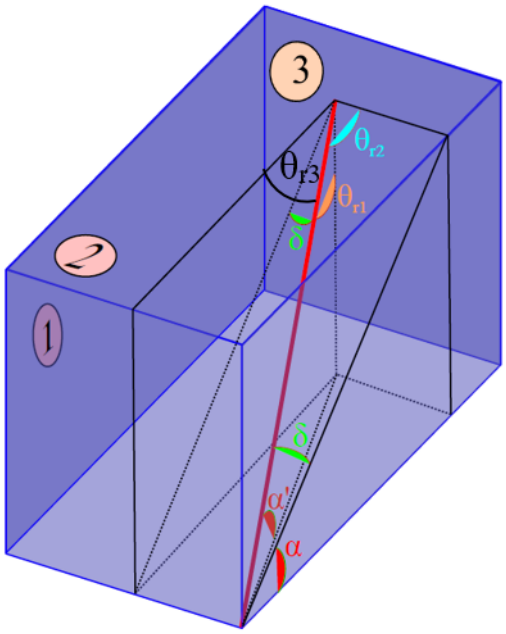


Figure 4.1: Part of the bar with photon (red line) emitted in the bottom right corner reflecting from plains number 1 and 2 [18].

The photon (red line) originates from the lower corner and reflects from upper plain (number 2) and side plain (number 1) in the upper corner. The radiation of the photon is described by two angles, α (between plain 2 and the diagonal of plain 1) and δ (between photon trajectory and the diagonal of plain 1). Notice that the diagonal of plain 1 forms perpendicular projection of photon trajectory to the side plain. Angles θ_{r1} and θ_{r2} are reflection angles from plains 2 and 1 respectively. They are defined as angles between the photon trajectory and the normal to the reflection plain.

If the photon should reflect, it has to follow these conditions of total reflection:

$$\theta_{r1} > \theta_C \ , \tag{4.1}$$
$$\theta_{r2} > \theta_C \ . \tag{4.2}$$

The probability, expressed from the Fresnel equations [13], that the photon reflects even if its angle of incidence on the boundary is smaller than the critical angle was neglected.

If we express cosines of reflection angles in terms of angles α and δ

$$\cos\theta_{r1} = \sin\alpha\cos\delta \ , \tag{4.3}$$
$$\cos\theta_{r2} = \sin\delta \ , \tag{4.4}$$

conditions in equations (4.1) and (4.2) take form:

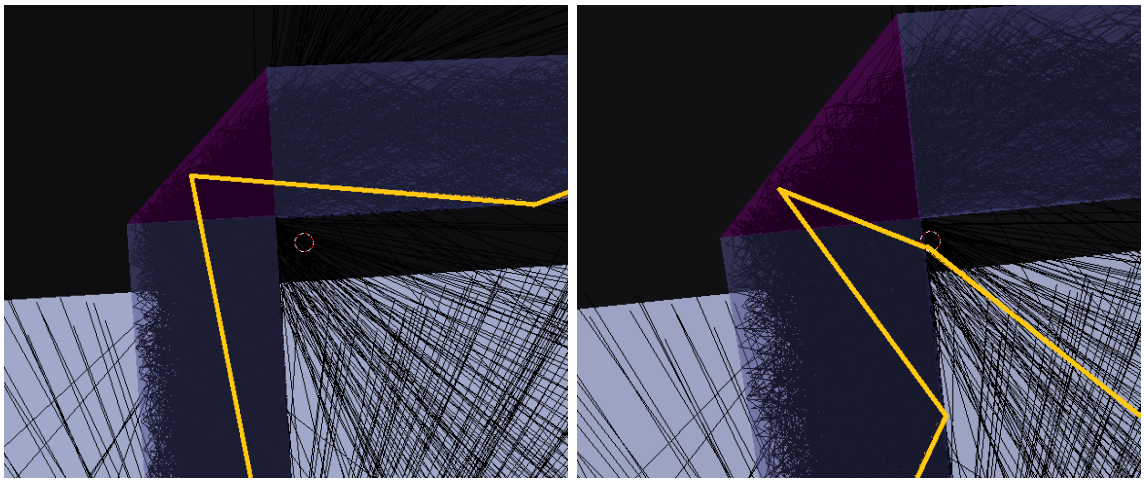
$$\cos\theta_{r1} < \sin\alpha\cos\delta \ , \tag{4.5}$$
$$\cos\theta_{r2} < \sin\delta \ , \tag{4.6}$$

for α, $\delta \in (0; \frac{\pi}{2})$ rad. The inequalities are opposite because the function cosine is in this interval decreasing whereas sine is increasing.

4.2 Description of Geometrical Principles of Photon Motion in the Bar

In this section there is considered a bar without taper. Three basic situations of photon motion in the bar are discussed.

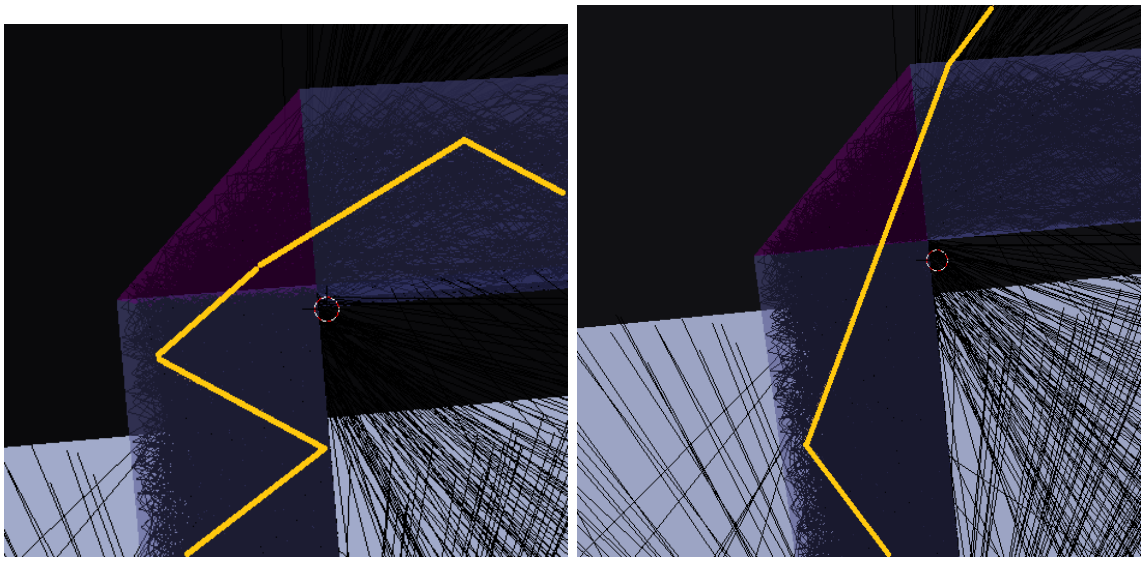
As mentioned earlier, edges of bars are cut under the Cherenkov angle 48 °. This causes large number of photons going straightly or with minimum reflections on the metallised elbow to deflect there to the photomultiplier. Firstly, this option is treated. We will call it *option 1 pass* in case photon reflects on the mirror to the photomultiplier or *option 1 not pass* when the photon is reflected back to the direction it came from (Figures 4.2(a) and (b)).



(a) (b)

Figure 4.2: Part of the bar with Al mirror is depicted. A photon deflects on the mirror to the PMT (a) (option 1 pass) or it reflects back (b) (option 1 not pass). Courtesy of Tomas Komarek.

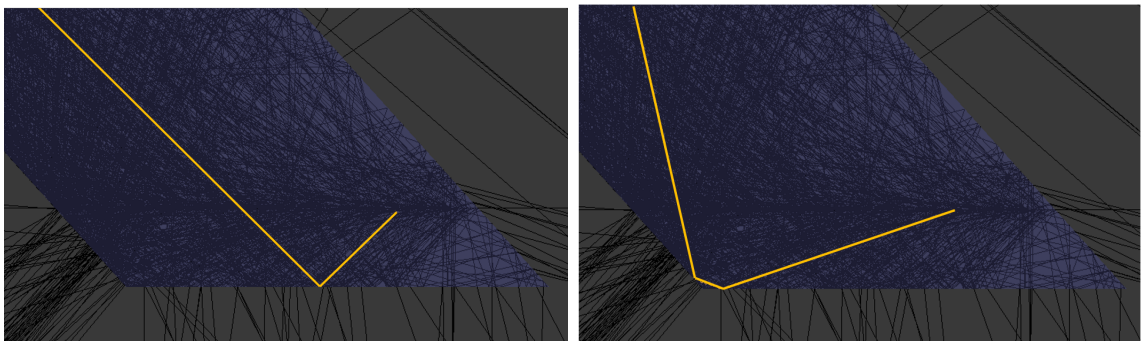
It may happen that the photon was emitted under such an angle that it misses the metallised mirror and goes straightly to the light guide (the second part of the bar) where it reflects from the bar, let's call it *option 2 pass* (Figure 4.3(a)) or flees from the bar, *option 2 not pass* (Figure 4.3(b)).



(a) (b)

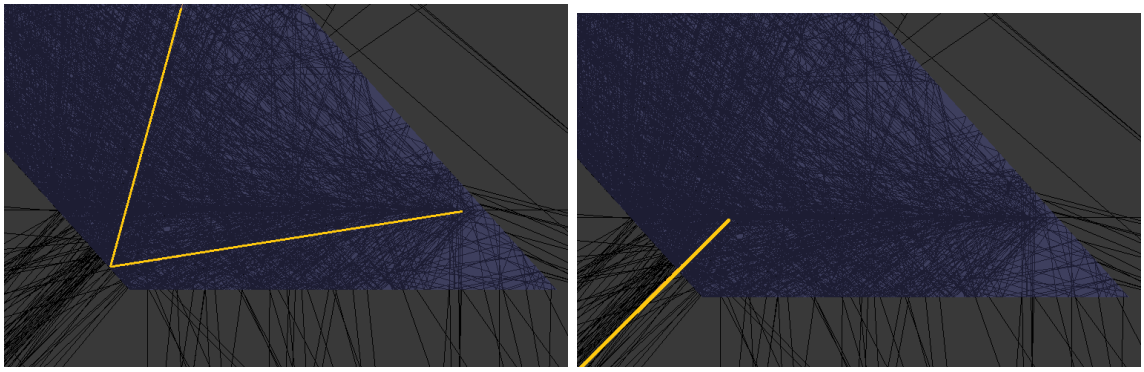
Figure 4.3: (a) Option 2 pass, (b) option 2 not pass. Courtesy of Tomas Komarek.

At last there are photons emitted in such a way they are heading not to the mirror but oppositely to the cut edge of the bar (*option 3*) (Figure 4.4(a)). This is the most complex issue as there are more possibilities where the photon may be reflected (Figure 4.4(b), Figure 4.5(a) and (b)).



(a) (b)

Figure 4.4: (a) Option 3 pass, (b) option 3 pass in another way. Courtesy of Tomas Komarek.



(a) (b)

Figure 4.5: (a) Option 3 pass, (b) option 3 not pass. Courtesy of Tomas Komarek.

The goal of the fast simulation is to create algorithm to differentiate between these options and to calculate the length of photon trajectories. Each option's solution is now treated separately in devoted subsection.

4.2.1 Option 1

From geometrical drawings it is possible to derive that angles α and δ under which the photon was emitted (Figure 4.1) are not changed for the particle that reflects from the mirror to the light guide. This key finding makes place for considerable simplification of the whole issue because it does not matter whether the bar is curved or flat which is of course simpler.

Henceforth we will consider only side view of the bar. In the following it is presented how to calculate trajectory length L of the photon in the bar.

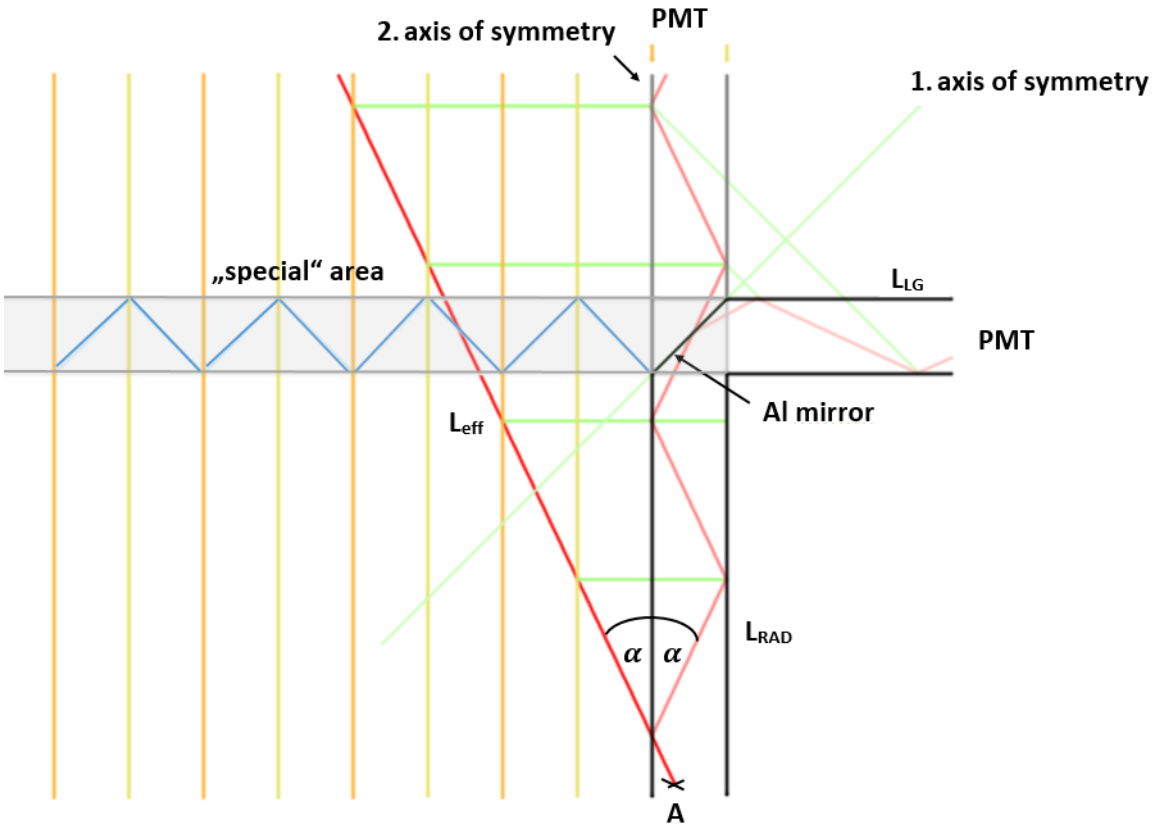


Figure 4.6: A sketch exlaining the solution of the option 1. Courtesy of Tomas Komarek.

In the Figure 4.6 there is a side view of an LQbar that was straightened so the mirror covered by Al poses an axis of symmetry. In this bar in the point A there was radiated a photon (red line) under an angle α. This photon reflects from the mirror and gets to the photomultiplier (PMT). According to the law of reflection, the angle α is not changed when photon reflects from the sides of the bar. The left side of the bar may be understood as a second axis of symmetry. So the trajectory of photon can be straightened and calculated:

$$L_{\text{eff}} = \frac{|A\ \ PMT|}{\cos\alpha}\ , \tag{4.7}$$

where $|A\ \ PMT|$ is the distance from the point of photon's emission to the PMT. This distance is calculated simply: we add lengths of radiator and light guide ($L_{\mathrm{RAD}} + L_{\mathrm{LG}}$) and subtract distance of the point A from the beginning of the bar.

The motion of photons takes place in the space so using the Figure 4.1 it can be derived the desired trajectory length [18]:

$$L = \frac{|A\ \ PMT|}{\cos\alpha \cdot \cos\delta}\ . \tag{4.8}$$

When implementing the code one must be capable of deciding whether the photon:

1. falls on the mirror
2. reflects from the mirror to the PMT and not back to the direction it came from (Figure 4.2(b)).

In this, the Figure 4.6, may be helpful. If the photon is to fall on the mirror, it must cross the blue line within the "special" area. The blue line represents mirroring of Al mirror over the second axis of symmetry. In case, the photon for instance crossed the blue line but then the orange line entering the neighbouring gray cell of "special" area, it would be reflected back to the radiator.

The way of calculating the length L presented above is not the only possible one. Another easy mean is to establish a coordinate system like the one in the Figure 4.7 bellow.

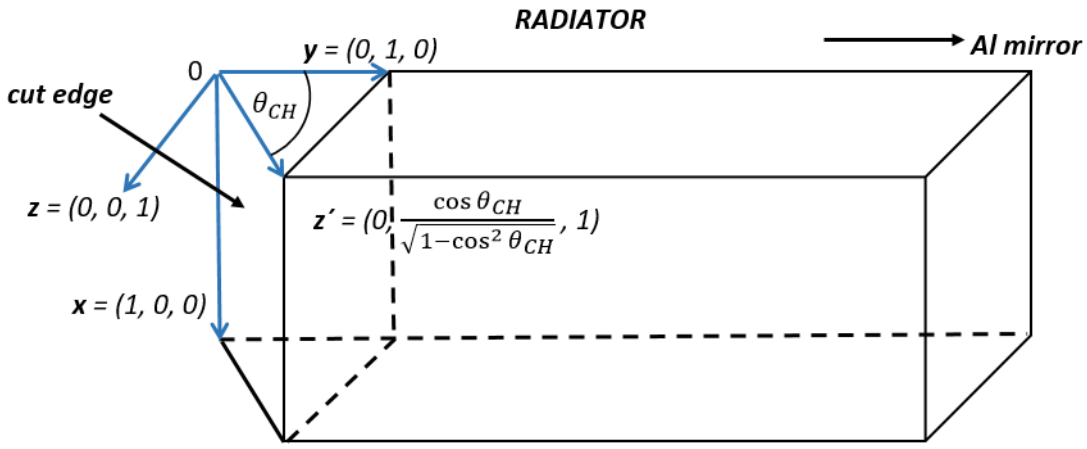


Figure 4.7: Coordinate system used for calculating of the length L.

In such a system we specify coordinates of point $A = [A_{\mathrm{x}},\ A_{\mathrm{y}},\ A_{\mathrm{z}}]$ and then by means of analytical geometry we calculate coordinates of the point $B = [B_{\mathrm{x}},\ B_{\mathrm{y}},\ B_{\mathrm{z}}]$. Point B forms the intersection between the line of photon (characterised by the point A and its direction vector $\mathbf{S}$) and the line parallel to the z-axis in the distance $L_{\mathrm{RAD}} + L_{\mathrm{LG}}$ (position of PMT) from the beginning. The relation for the length L is:

$$L = \sqrt{(B_{\mathrm{x}} - A_{\mathrm{x}})^2 + (L_{\mathrm{RAD}} + L_{\mathrm{LG}} - A_{\mathrm{y}})^2 + (B_{\mathrm{z}} - A_{\mathrm{z}})^2}\ . \tag{4.9}$$

Both relations (4.8) and (4.9) were implemented and provide same results. The second possibility is especially useful when angles α and δ are not known and have to be calculated separately.

4.2.2 Option 2

For this option one more condition of total reflection must be derived because the photon falls on the side of the light guide under different angle than α. Basically we need to set new angle of incidence of photon trajectory on the x-z plane (labelled by number 3 in the Figure 4.1). Let us call this angle θ_{r3}. Any photon that misses the Al mirror and gets into the PMT has to follow this condition:

$$\theta_{r3} > \theta_C \ . \tag{4.10}$$

Or it may be expressed as function of angles α and δ:

$$\cos\alpha \cos\delta > \cos\theta_C \ . \tag{4.11}$$

The principle of trajectory length calculation is introduced in the following diagram.

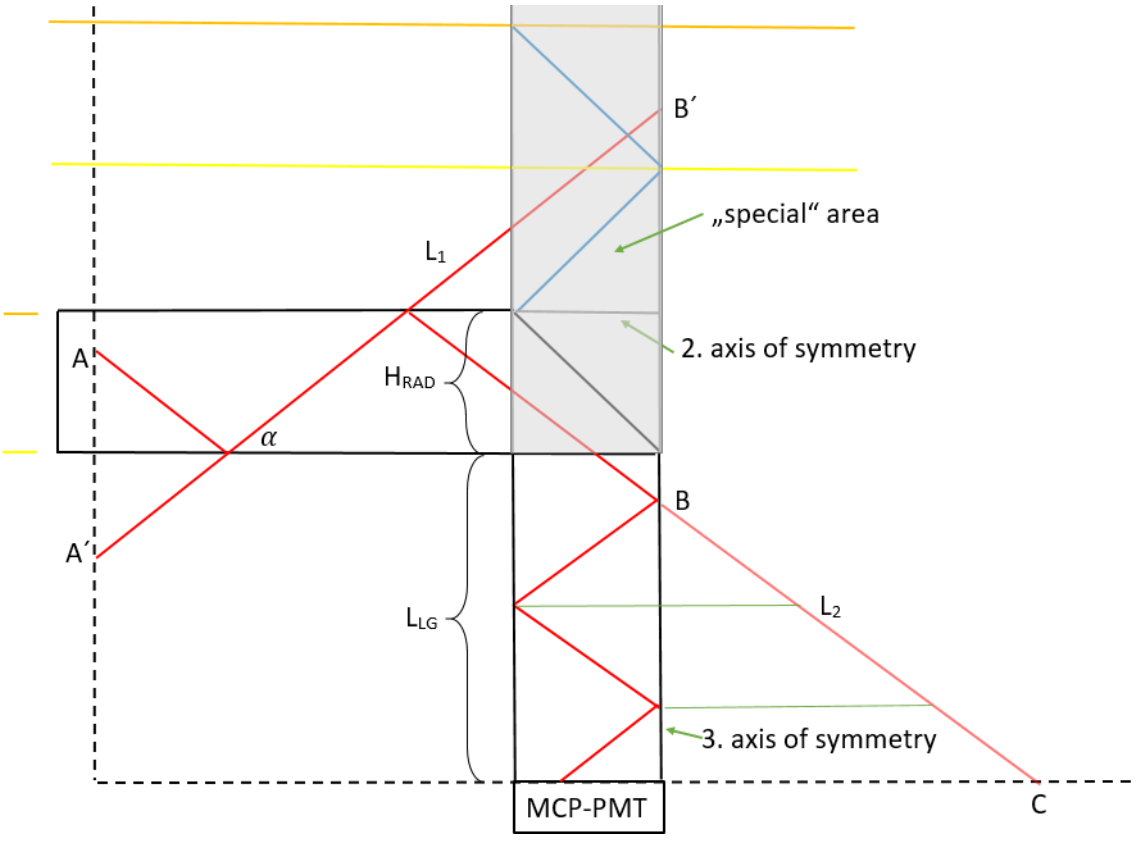


Figure 4.8: A sketch to the exlanation of the solution for the option 2.

This option is recognized in such a way that unlike the option 1, the photon in the "special" area (gray rectangle) does not cross the blue line (representing the Al mirror) and it crosses the yellow line (meaning the right side of the bar.)

The calculation of L has 2 steps. The first one pays close resemblance to the one presented within the option 1. We find coordinates of the point B' that is the point B mirrored through the second axis of symmetry. In the point B the photon hits the side of the light guide. Having these coordinates we can calculate length L_1.

The second step is to calculate coordinates of the point C where the straightened photon trajectory intersects the line parallel to the y-axis in distance $L_{\mathrm{LG}} + H_{\mathrm{RAD}}$, where H_{RAD} is hight of the radiator. The length of the second part of the trajectory L_2 is:

$$L_2 = \sqrt{(L_{\mathrm{LG}} + H_{\mathrm{RAD}} - B_{\mathrm{x}})^2 + (C_{\mathrm{y}} - B_{\mathrm{y}})^2 + (C_{\mathrm{z}} - B_{\mathrm{z}})^2} \ . \tag{4.12}$$

The whole length is an addition of both lengths:

$$L = L_1 + L_2 \ . \tag{4.13}$$

4.2.3 Option 3

Without loss of generality it is presented only solution of the situation depicted in the Figure 4.4(b) (the figure "a" is only a special case of "b"). Firstly, the condition of the total reflection on the cut edge has to be found which is done in the very same manner as described within the options 1 and 2. The normal to the cut edge is expressed and by means of analytic geometry the angle τ between the photon trajectory and the normal is calculated. This angle has to be greater as the critical angle θ_{C}.

Henceforth it is considered only upper view of the bar because only in this view the cut edge is clearly visible. The principle of solution is evident from the following sketch.

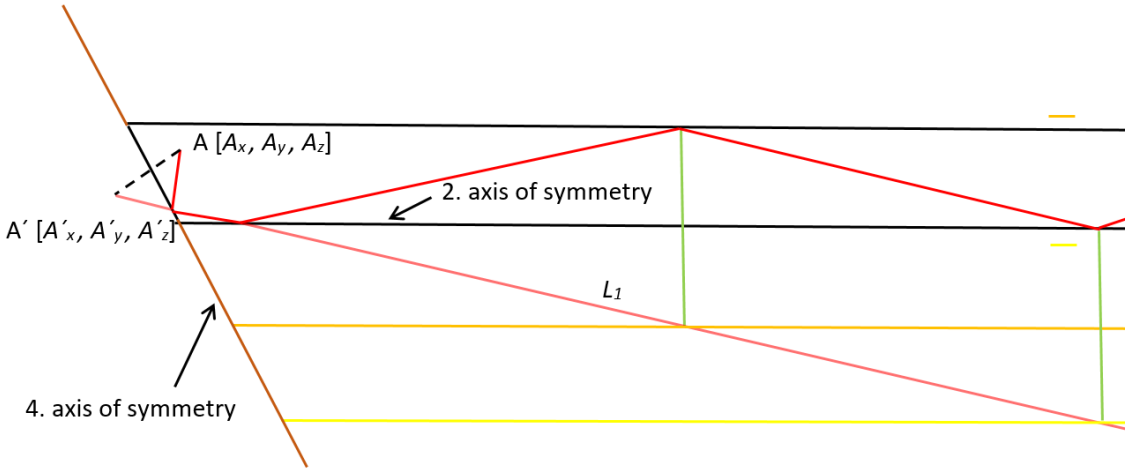


Figure 4.9: A sketch to the exlanation of the solution for the option 3.

The fourth axis of symmetry is introduced: the cut edge of the bar. Through this axis the point A is mirrored into point A'. The new coordinates may be calculated in the following way. The general equations of the edge and line segment $|A\ A'|$ are:

$$B_{\mathrm{y}} - \frac{\cos\theta_{\mathrm{CH}}}{\sqrt{1-\cos^2\theta_{\mathrm{CH}}}} \cdot B_{\mathrm{z}} = 0 \ , \tag{4.14}$$

$$\frac{\cos\theta_{\mathrm{CH}}}{\sqrt{1-\cos^2\theta_{\mathrm{CH}}}} \cdot B_{\mathrm{y}} + B_{\mathrm{z}} - \frac{\cos\theta_{\mathrm{CH}}}{\sqrt{1-\cos^2\theta_{\mathrm{CH}}}} \cdot A_{\mathrm{y}} - A_{\mathrm{z}} = 0 \ . \tag{4.15}$$

The solution of this set of equations are two coordinates of the point $B = [B_x,\ B_y,\ B_z]$ that is in the middle between A and A'. That means: $y = \frac{A_y + A'_y}{2}$ and $z = \frac{A_z + A'_z}{2}$.

$$B_y = A_y \cos^2\theta_{CH} + A_z \cos\theta_{CH} \sin\theta_{CH} \ , \tag{4.16}$$
$$B_z = A_y \cos\theta_{CH} \sin\theta_{CH} + A_z \sin^2\theta_{CH} \ . \tag{4.17}$$

Combining these relations together we obtain mirrored coordinates A'_y and A'_z:

$$A'_y = 2A_y \cos^2\theta_{CH} - A_y + 2A_z \cos\theta_{CH} \sin\theta_{CH} \ , \tag{4.18}$$
$$A'_z = 2A_y \cos\theta_{CH} \sin\theta_{CH} - A_z + 2A_z \sin^2\theta_{CH} \ . \tag{4.19}$$

So the point A' becomes new initial point and the photon has also different direction vector. Now the issue is treated as option 1 or 2.

4.2.4 Comparison with the Full Simulation in GEANT4

Because of verification of proper functionality there was generated a data sample in Geant4 containing all necessary information about photons i.e. coordinates of their initial position and direction angles. This data was inserted into the code of Cherenkov simulation and results were compared.

Into the tested code were also implemented these effects:

1. reflectivity on the Al mirror (90%),
2. transmission of fused silica to the light which is 99% for wavelength typical for the Cherenkov radiation (this information was obtain from the data sheet of suprasil, the material of the bars, included in the Appendix A),
3. attenuation in the glue that connects radiator and light guide. Its value is a function of wavelength and the implementation was retaken from Geant4.

Firstly, only options 1 and 2 and separately 3 were verified (Figures 4.10 and 4.11).

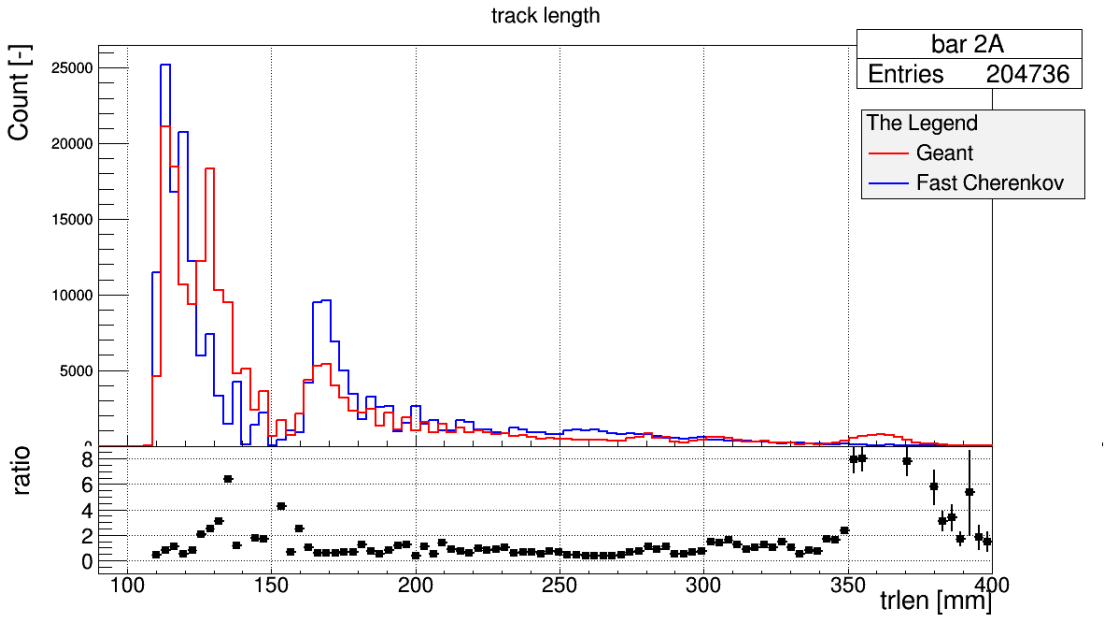


Figure 4.10: The comparison of length histograms between the Fast Cherenkov code and Geant data. Only options 1 and 2 are considered.

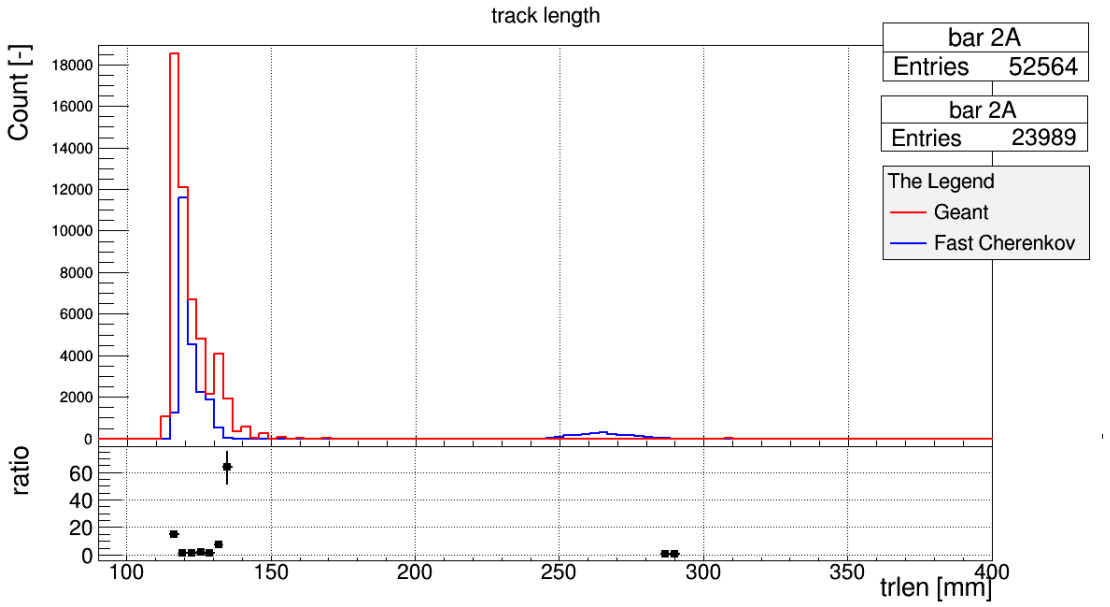


Figure 4.11: The comparison of length histograms between the Fast Cherenkov code and Geant data. Only option 3 is considered.

Secondly, all three options were put together (Figure 4.12).

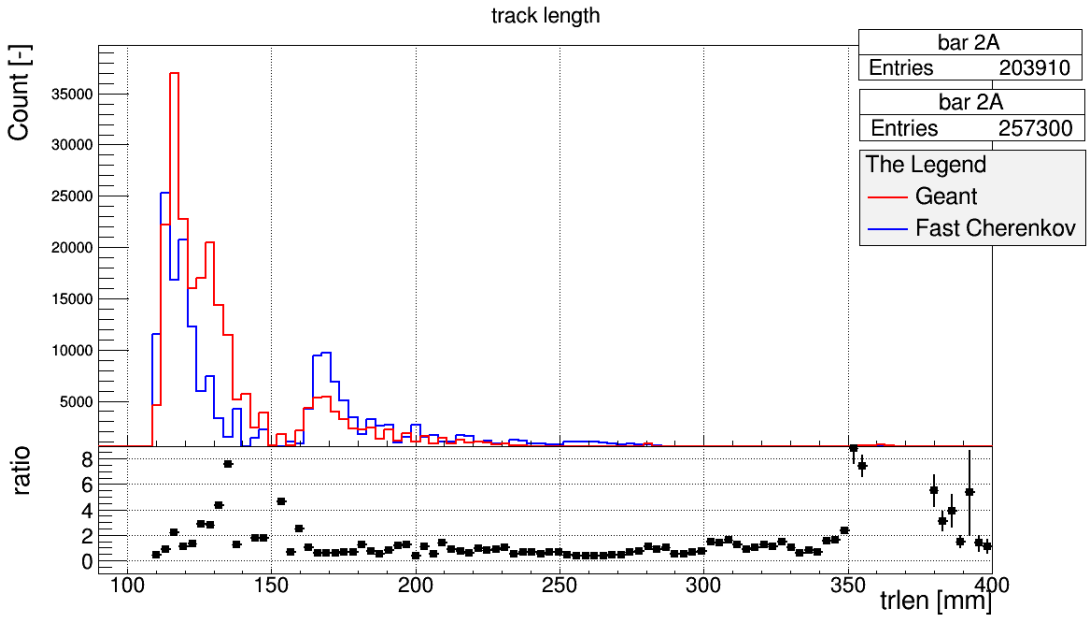


Figure 4.12: The comparison of length histograms between the Fast Cherenkov code and Geant data. All options are considered.

It is obvious that the tested code does not describe the Geant data very well. The leading edge is in a good agreement, however in the region of length (130 - 140) mm the code throws away huge number of particles so the second peak is completely missing.

In order to disclose the reason, there were done drawings by hand of few photons. It was discovered that some photons are treated as different options by the code and by Geant. In some cases the drawings agreed with the result Geant rendered.

When implementing more complex possibilities in the option 3 the problem has risen that photons did not fulfil the condition of total reflection and were lost. In the Geant data, however, these photons came through. This phenomenon was most probably caused by an error in the code. Despite considerable efford, the error was not found yet.

Chapter 5

Implementation of the Fast Cherenkov Simulation Using the ANN

In this chapter there are discussed matters concerning details of the training of neural networks, the choice of input variables and training samples preparation. The whole process of Fast Cherenkov Simulation implementation is described as well as validation of the output with full simulation in Geant4.

5.1 The Strategy

The objective of the AFP ToF is to measure the time when the proton entered the LQbars. The global time of each bunch-crossing t_0 is known, so the only task is to simulate what time t each photon needs to pass through the bar.

The first step to achieve this is to calculate each photon's trajectory length L. The idea is to use the neural network for regression of the histograms of trajectory length generated by Geant4. Examples of such histograms for bars with and without taper respectively are shown in the Figures 5.1 and 5.2.

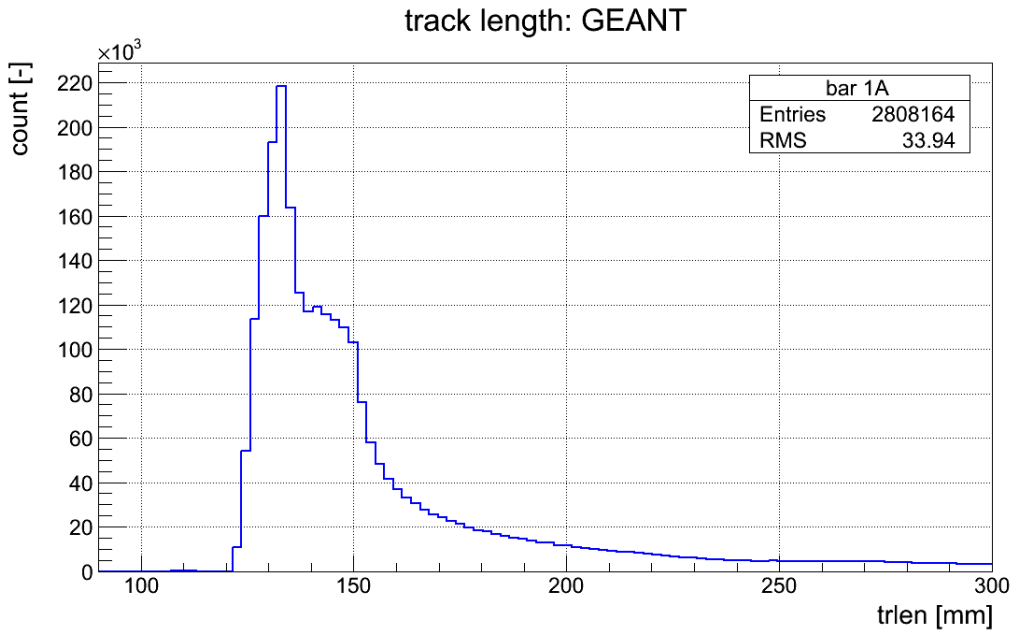


Figure 5.1: Histogram of photon trajectory length for the bar 1A (with taper).

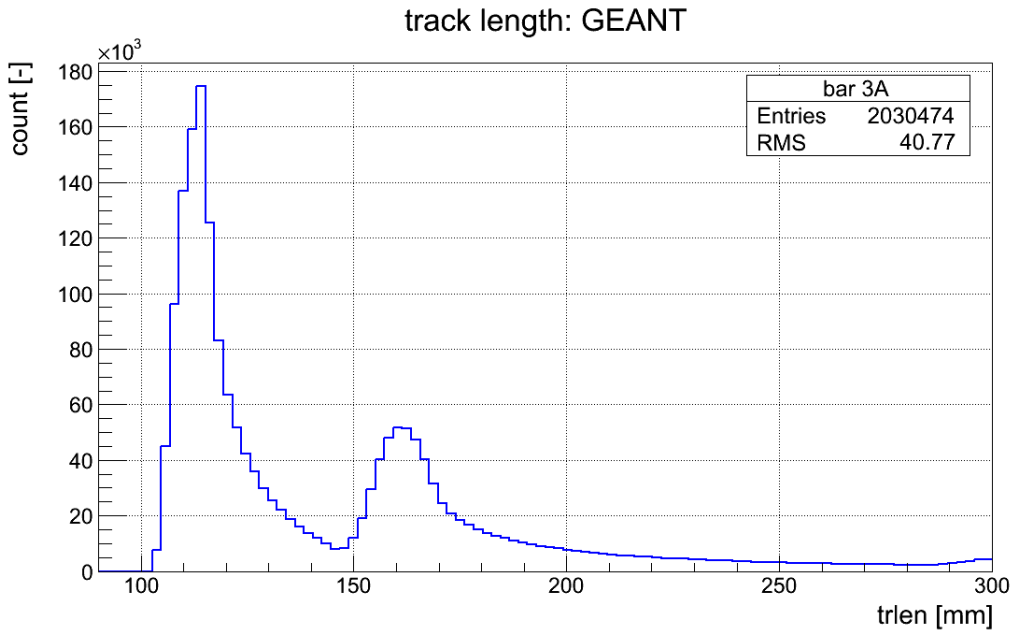


Figure 5.2: Histogram of photon trajectory length for the bar 3A (without taper).

The second step is to calculate probability of absorption p_{abs} [18] of the photon in the bar:

$$p_{\text{abs}} = 1 - e^{-\mu_{\text{abs}}(E)\cdot L} \ , \tag{5.1}$$

where μ_{abs} stands for the attenuation coefficient of the material of bars and it is a function of energy ($\mu_{\text{abs}} = \mu_{\text{abs}}(E)$).

In the third step the time t_{photon} needed by the photon to pass through the bar may be expressed:

$$t_{\text{photon}} = \frac{L}{v_{\text{c}}} = \frac{n(\lambda)}{c} \cdot L \ . \tag{5.2}$$

Nonetheless, for purpose of comparison with Geant data it is necessary to calculate time of flight t_0 of the proton from some chosen plain (yellow plain in the Figure 5.3) until it radiates give photon. In Geant the trigger plain (blue plain) is shifted with respect to the yellow plain by the time approximately $t_{\text{trig}} = 25$ ps. So the final time plotted in histograms takes form:

$$t = t_{\text{photon}} + t_0 + t_{\text{trig}} \ . \tag{5.3}$$

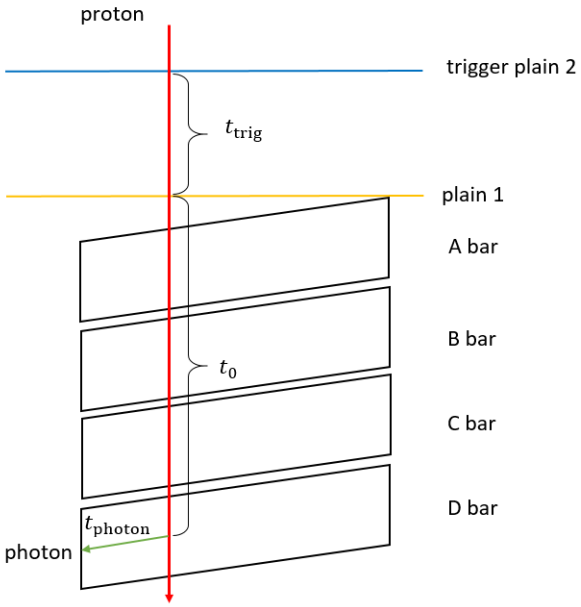


Figure 5.3: Sketch explaining time corrections.

5.2 Architecture and Training of the NN

5.2.1 Input Data and Variables

The data source for the training of neural network was prepared in Geant4. The Geant implementation allows for various settings like the number of bars placed in the holder or select which part of radiator is hit with incoming protons. For the purpose of the Fast Cherenkov Simulation preparation only one selected LQbar was simulated having protons randomly distributed in a square window spanning 5 mm from its edge (Figure 5.4).

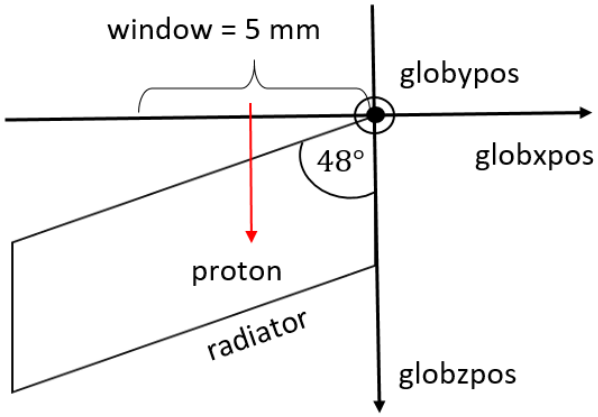


Figure 5.4: Global coordinates and the window for protons.

The reason for such position is that the best resolution for the bars, observed during testing campaigns, was just at the edge of bars compared to further distances of the window from the cut edge [23].

The number of generated protons was chosen different for LQbars with and without taper mainly for the fact that these bars have different proportions in *globypos* coordinate. So for LQbars with taper we generated 30 000 protons and for LQbars without taper only 10 000 protons were sufficient with regard to the simpler design of the bar. The energy of protons was set to 7 TeV.

Geant4 dumps the information about all protons and emitted photons into a ROOT file which is comprised of four `TTrees` called: `PhotonTrack, Sensor, EventInfo, RunInfo`. In the Table 5.1 there is introduced a list of the most important variables of these TTrees and their meaning. All variables of such a TTree can be found in a ROOT file called `tofsim_set14_1A_new_moved_random_1000Evt.root` (generated for the bar 1A) on the enclosed CD-ROM.

Variable	`TTree` name Unit	Meaning
	`PhotonTrack`	
`trackid`	-	Identification number (ID) of the generated photon (or of its track).
`vtxx`	*mm*	Position coordinates of the photon's emission in the local
`vtxy`	*mm*	coordinate system of the bar (Figure 5.5).
`vtxz`	*mm*	
`vtxdirx`	*mm*	Direction coordinates of photon in the local coordinate
`vtxdiry`	*mm*	system of the bar.
`vtxdirz`	*mm*	
`vtxwl`	*nm*	Wavelength of the emitted photon.
	`Sensor`	
`eventid`	-	ID of the event. Photons originating from one event have same `eventid`.
`trackid`	-	ID of the photon (or of its track).
`barid`	-	ID of the bar from which the photon fell on the PMT. ID's layout is specified in the Figure 5.6.
`vtxbarid`	-	ID of the bar where the photon was emitted.
`time`	*ns*	Arrival time of the photon into the PMT with respect to a trigger plain which is situated before A bars.
`trlen`	*mm*	Length of the photon trajectory.
`wl`	*nm*	Wavelength of the photon and its value equals to the `vtxwl` for the given photon.
`vtxphi`	*rad*	Angle under which the photon was radiated in the plane *globypos* - *globxpos* (Figure 5.7).
`vtxtheta`	*rad*	Cherenkov angle of the photon. It is measured from the axis *globzpos* (Figure 5.7).
	`EventInfo`	
`gpxpos`	*mm*	Global position coordinates of the gun of the event, x
`gpypos`	*mm*	y
`gpzpos`	*mm*	z
`gptheta`	*rad*	Angle between the direction vector of proton and the axis *globzpos*.
`gpphi`	*rad*	Angle of projection of the proton's trajectory into in the plane *globypos* - *globxpos* For the purpose of training of NN both angles are set zero, so proton's directions are parallel to the cut edge.

Table 5.1: Some of variables used in the `TTree` and their explanation.

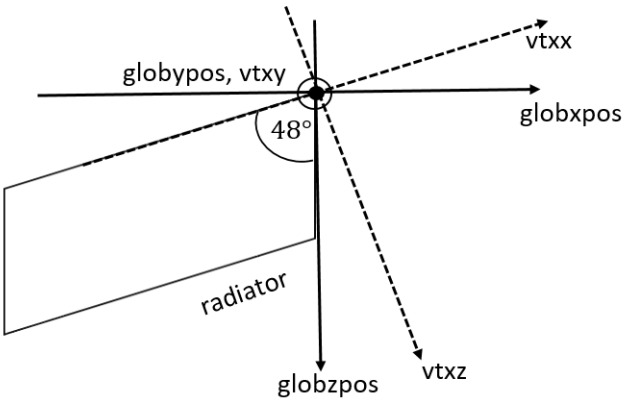


Figure 5.5: Local and global coordinate system.

	D	C	B	A
Train 1	1D ~ 44	1C ~ 43	1B ~ 42	1A ~ 41
Train 2	2D ~ 34	2C ~ 33	2B ~ 32	2A ~ 31
Train 3	3D ~ 24	3C ~ 23	3B ~ 22	3A ~ 21
Train 4	4D ~ 14	4C ~ 13	4B ~ 12	4A ~ 11

Figure 5.6: Maping of the LQbars to `barids`. This figure shows the front view of the ToF like the one in the Figure 2.1(b).

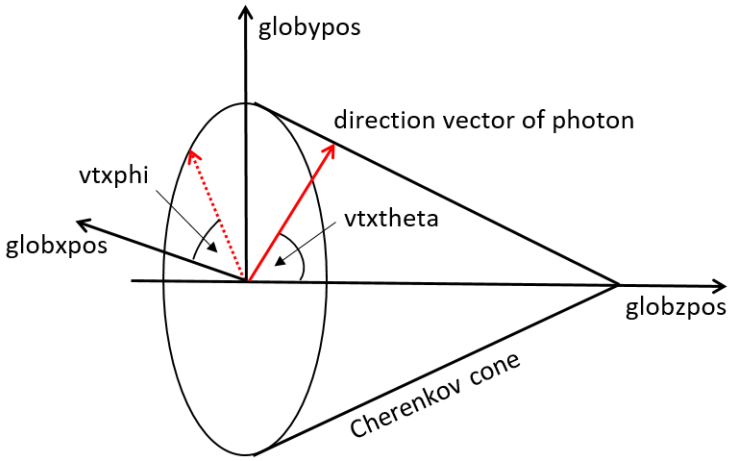


Figure 5.7: Description of the meaning of angles `vtxtheta` and `vtxphi`. Photon track is substituted by a red full arrow and the projection of the track by dotted red arrow. In this case, the original proton is parallel to the `globzpos` axis.

5.2.2 Training of the ANN

Variables relevant for the training of the ANN were, after several tests, set these: `vtxx, vtxy, vtxz, vtxphi, vtxtheta`. For the purpose of training the original ROOT file has to be recreated because these variables are placed in two separate trees. For the training we also use only photons that passed the whole bar and got to the PMT (our sensor). An example of such an input `TTree` is enclosed on the CD-ROM under the name `inputTree_1A_new_moved_random_1000Evt.root`. Into the input Tree there must be also added information about the photon track length `trlen`.

Several layouts of the ANN were tried e.g. 1 hidden layer containing from 20 to 50 neurons or 2 hidden layers comprising from 20 to 30 neurons each. Also the more complicated structures containing 3 or 4 hidden layers with 20 neurons each were tested. It was discovered that the best agreement between Geant data and the output of the NN is reached with 2 hidden layers each made of 25 neurons with sigmoid activation function (`kSigmoid`) and default training method (`kBFGS`). The final structure is in the Figure 5.8.

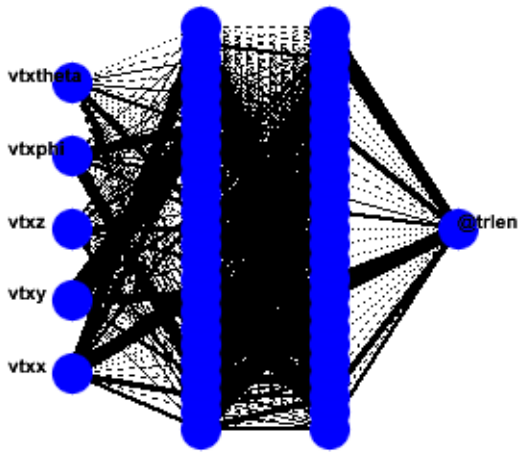


Figure 5.8: The final structure of the ANN.

Continuous testing also revealed that it is convenient to train the NN twice. At the end of the first training the weights of synapses are dumped into a special text file (for details see the Section 3.2 of this thesis). At the beginning of the second training this file is loaded and weights are used to initialise the training. This procedure enables lowering of training error and ensures the proper placement of the leading edge of length histograms.

5.3 Correlation of Variables

The choice of input variables is crucial for the training process and the results obtained from exported code representing the trained NN. At the end of the training the algorithm optionally analyses the influence of each variable on the output. This information should be carefully considered as it helps to get rid of useless variables that burden the network.

There are two features of special importance that have major effect on the ANN's training and results: Firstly, at least one of the input variables must be correlated

to the output. Otherwise the trained NN will not provide good results. In our case this variable is the angle `vtxphi` and its dependence is shown in the Figure 5.9. The importance of this angle is also evidenced by the text output of ANN training which shows that `vtxphi` has bigger influence on the resulting length than other variables.

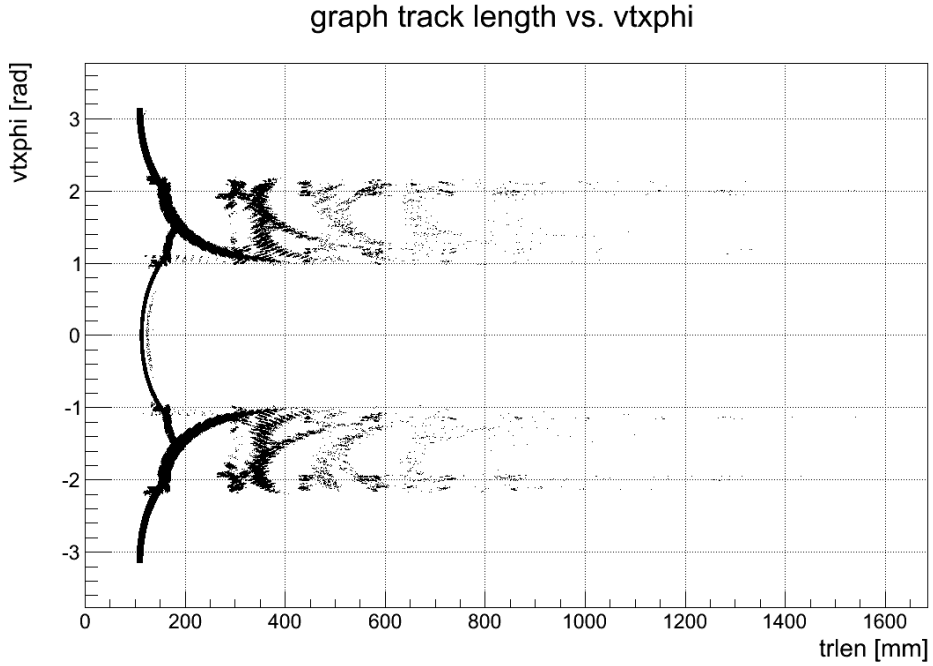


Figure 5.9: Correlation between the angle `vtxphi` and the photon trajectory length `trlen`. (Source Geant4.)

This structure was at first introduced in the article [22] and it was called *wings*. In the central part `vtxphi` $\in$ (-1; 1) rad there are photons going straightly on the mirror and to the PMT; this part will be called *mid wing*. The upper part of the plot, `vtxphi` $\in$ (1; 3.2) rad, is called *up wing* and the lower part (`vtxphi` $\in$ (-1; - 3.2) rad) is *down wing*. The biggest number of photons is situated at low lengths (Figure 5.10).

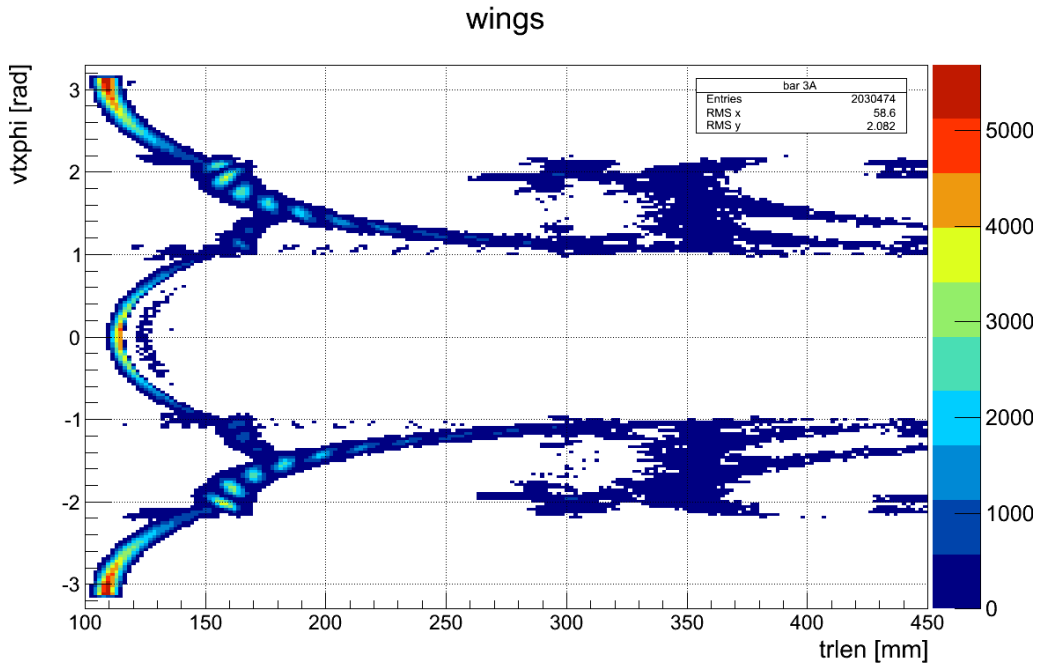


Figure 5.10: Histogram of wings.

Secondly, the correlation between the input and output variable has to be unambiguous. When we look back on the Figures 5.9 and 5.10 we see that this condition does not hold especially for up and down wings. Therefore a set of cuts is proposed to clean the training data sample.

5.3.1 Cuts

The exact formula of cuts differs from LQbar to LQbar; bear in mind that each bar has different length of the radiator because of the effect of compensation of photon's arrival time. Also the fine structure of wings is more complex for bars with taper. For these reasons only general principles of cutting of data are discussed.

Bars without taper require only in most cases cuts on long trajectories. It was discovered that only cca 8 % of all photons have `trlen` $>$ 300 mm. Moreover these photons form the major part of unwelcome ambiguities. After the cut the up wing of the bar 3A may look like in the Figures 5.11 and 5.12.

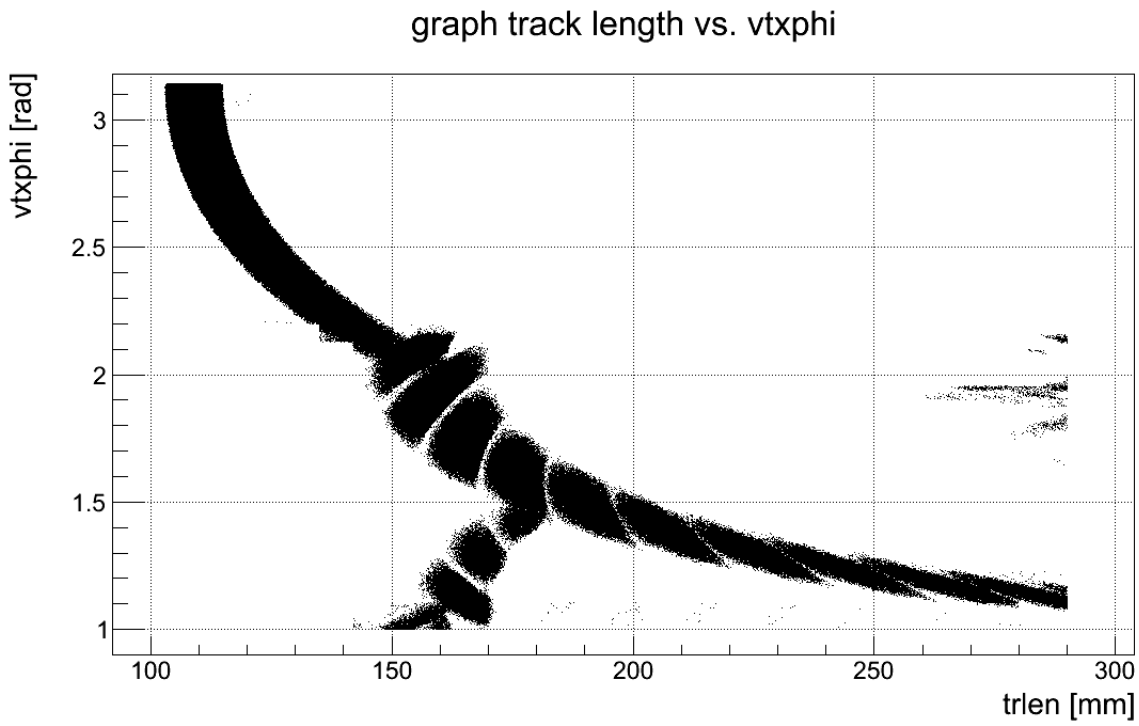


Figure 5.11: Plot of the up wing of bar 3A (without taper) after the cut.

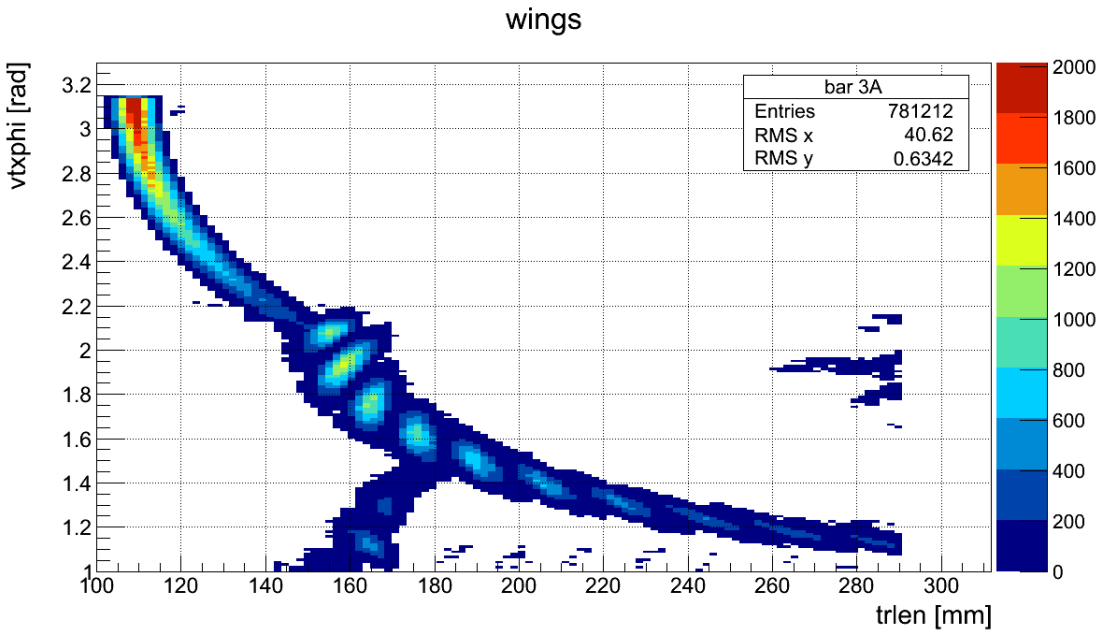


Figure 5.12: Histogram of the up wing of bar 3A after the cut.

The lower region (down wing) has very similar cuts. The mid wing i.e. the region `vtxphi` $\in$ (-1; 1) rad requires only cuts for particles with longer trajectories: In the case of bar 3A all lengths should be smaller than 156 mm. Photons with greater length deviate from the main "curve" (Figure 5.13) of the mid wing.

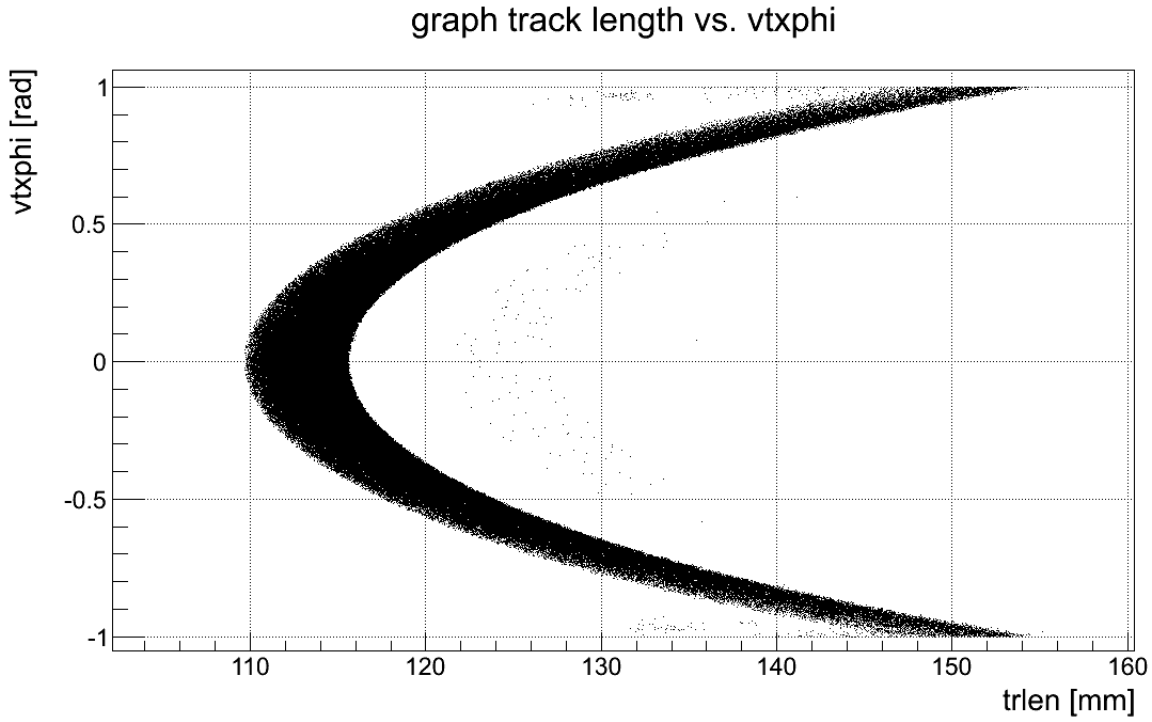


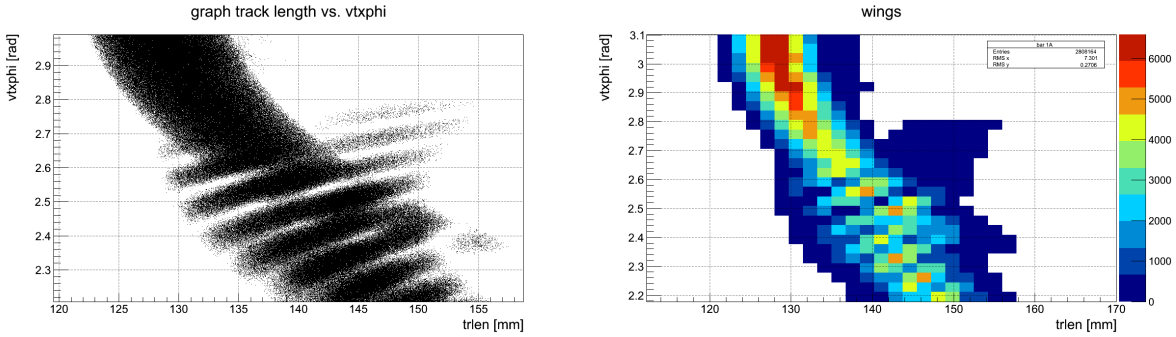
Figure 5.13: Graph of the mid wing of the bar 3A after the cut.

Bars with taper with respect to bars without taper evince complex fine structure of the wings and therefore have more ambiguities. After several test the best results were obtained when each of the three wings is separated into two or three parts. Spans of these intervals along with the cuts on trajectory length for bars 1A and 1C are presented in the Table 5.2. It is convenient to train different NN for each of these intervals with one exception mentioned later.

1A		
name of the interval	`vtxphi` $\in$	`trlen`
sector 1 up wing	(2.58; 3.2)	< 144
sector 2 up wing	(1.; 2.58)	< 325
sector A mid wing	(0.5; 1.)	< 159
centre mid wing	$(-0.5; 0.5)$	< 146
sector B mid wing	$(-1.; -0.5)$	< 159
sector 2 down wing	$(-2.58; -1.)$	< 325
sector 1 down wing	$(-3.2; -2.58)$	< 144
1C		
sector 1 up wing	(2.6; 3.2)	> 105 && < 144
sector 2 up wing	(1.; 2.6)	> 105 && < 275
sector A mid wing	(0.5; 1.)	< 145
centre mid wing	$(-0.5; 0.5)$	< 132
sector B mid wing	$(-1.; -0.5)$	< 144
sector 2 down wing	$(-2.6; -1.)$	> 115 && < 275
sector 1 down wing	$(-3.2; -2.6)$	> 105 && < 131

Table 5.2: Intervals of the testing data and their cuts.

For instance, the reason for the separation of up wing into sectors 1 and 2 is apparent from the Figure 5.14(a) and (b). There we can see rapid change of the structure of the data: In the sector 1 the distribution of the data is random whereas in the sector 2 photons create certain "stripes" which are more fine than "clusters" in the Figure 5.11.



(a) (b)

Figure 5.14: Detail of the up wing, boundary between sector 1 and 2.

For the data from sector 1 it is therefore convenient to be trained separately from sector 2. Nevertheless, the whole sector 2 is made of "stripes" (Figure 5.15) which spoils the training. So for the testing of the data from sector 2 it is better to train sector 2 together with sector 1, but use such NN only for data from sector 2.

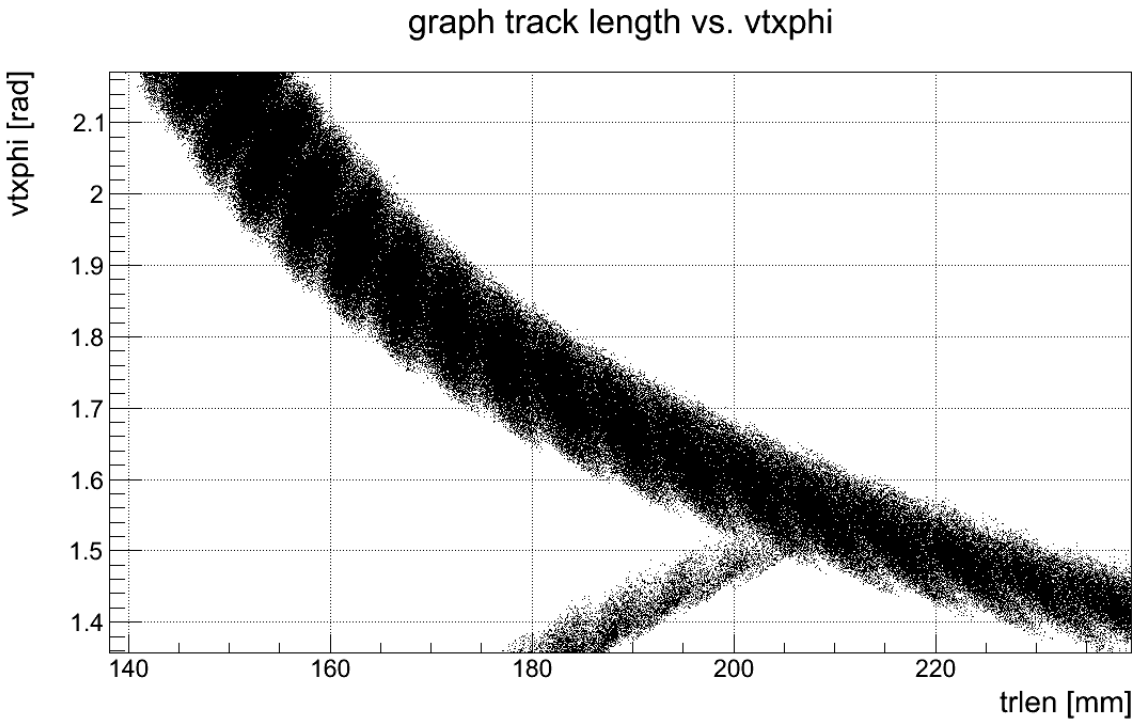
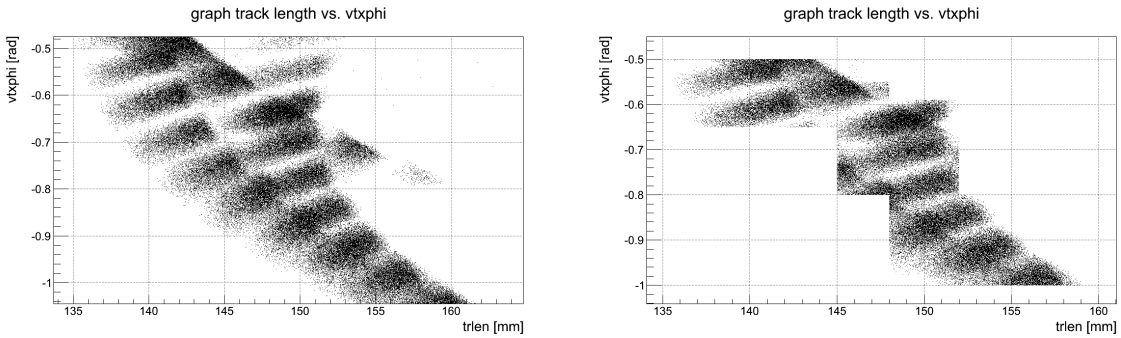


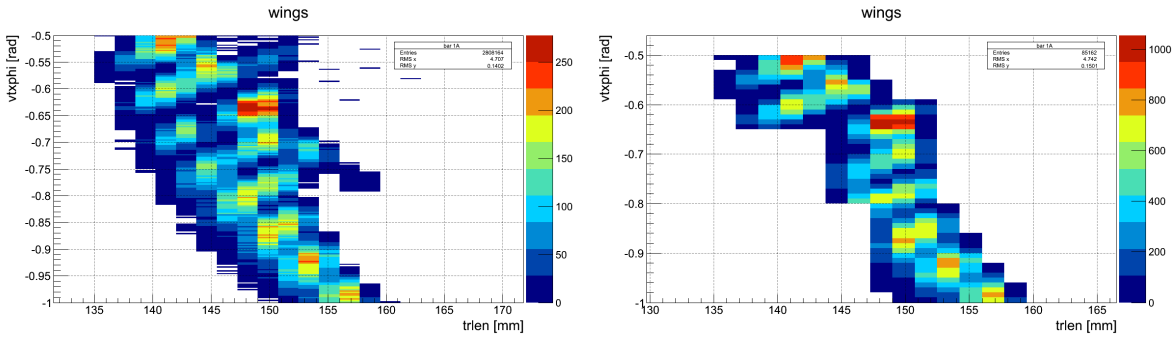
Figure 5.15: Fine "stripe" structure of the sector 2 up wing.

Conditions on length proposed in the Table 5.2 are only very rough. Here is given an example of the data from sector B mid wing before the cut (Figures 5.16(a)) and after the cut (Figure 5.16(b)).



(a) (b)

Figure 5.16: Graph of the sector B mid wing before the cut (a) and after the cut (b).



(a) (b)

Figure 5.17: Histogram of the sector B mid wing before the cut (a) and after the cut (b).

Obviously these cuts are done with respect to places with largest counts of photons (Figure 5.17(a) and (b)) bearing in mind the natural structure of the data and omitting ambiguities.

Very similar cuts are done for sector A; they are just a bit moved in the length and `vtxphi`.

5.4 Results for Single LQBar

This section is devoted to the presentation of results of training for single bar without and with taper. To do so all NNs have to work together to simulate the whole span of the angle `vtxphi`. In order to demonstrate the correctness of the NN's function the comparison between the trained NN and Geant data is done here.

For the purpose of testing of trained neural networks, it is convenient to generate new data sample in Geant instead of using the same sample for training of NN and for testing. This prevents of "overfitting" the NN.

5.4.1 Results for the LQBar without Taper

Here we provide results for bar 3A because only this bar has been trained for reasons provided later.

To judge how good is the agreement between NN and Geant the data from Geant were inserted into the NN and histograms of the photon track length for both Geant and NN were created (Figure 5.18). A ratio histogram of their division was also found useful.

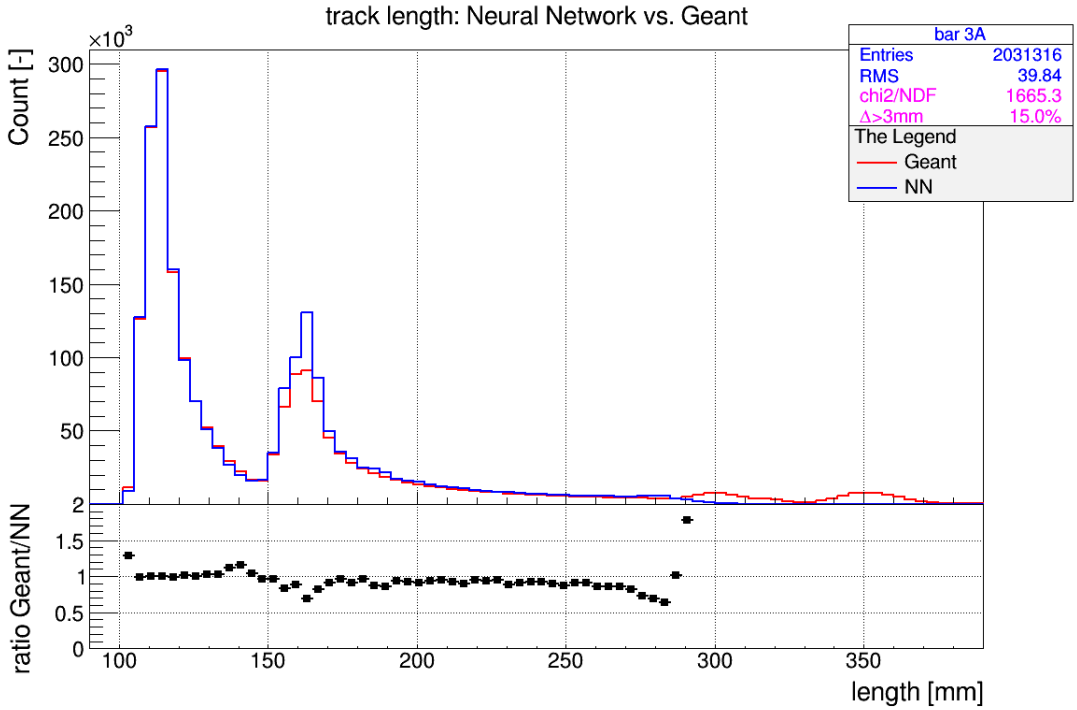


Figure 5.18: Comparison of photon track length given by Geant (red histogram) and calculated by NN (blue histogram).

We see that the NN describes the data almost perfectly. The secondary peak is a little overestimated by the ANN, but that is not so important as the first peak enters sooner to the PMT to give information of the presence of the proton.

In the table in the upper right hand corner there is information about statistics. The last line, $\Delta > 3$ mm, says how many percent of photons had track length difference (track length Geant - track length NN) bigger than 3 mm. It is also very meaningful to create a histogram of such difference (Figure 5.19).

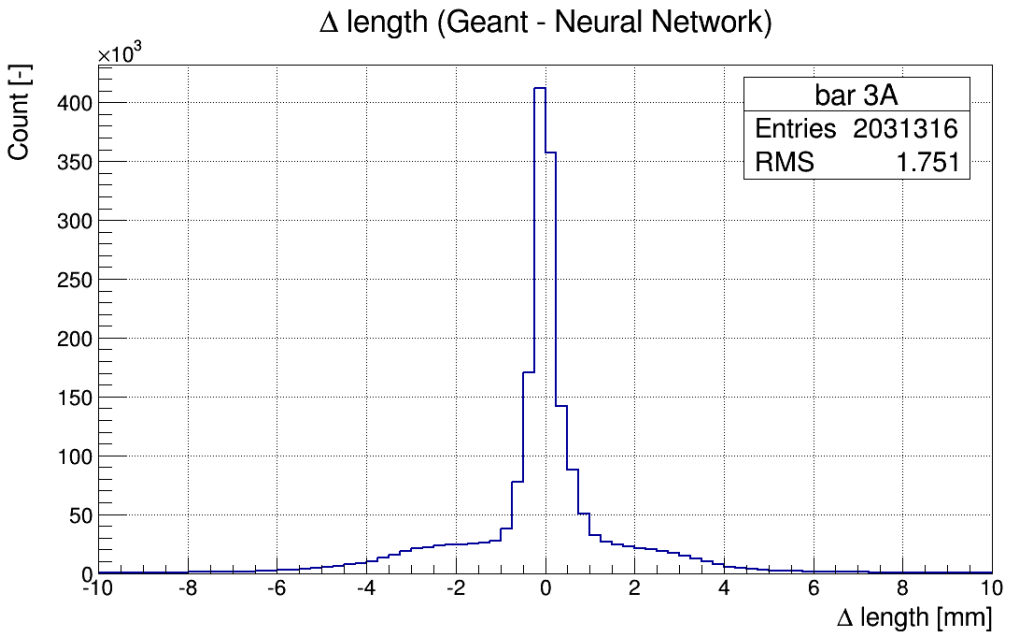


Figure 5.19: Histogram of photon track difference: track length Geant minus track length NN.

This histogram shows that vast majority of photons has correctly calculated track length. The RMS is also sufficiently small in comparison with the desired resolution of 10 ps which poses 3 mm of proton trajectory length.

5.4.2 Results for the LQBar with Taper

In this subsection we present results for bar 1A. The same histograms as in the previous subsection for track length (Figure 5.20) and its difference (Figure 5.21) were created.

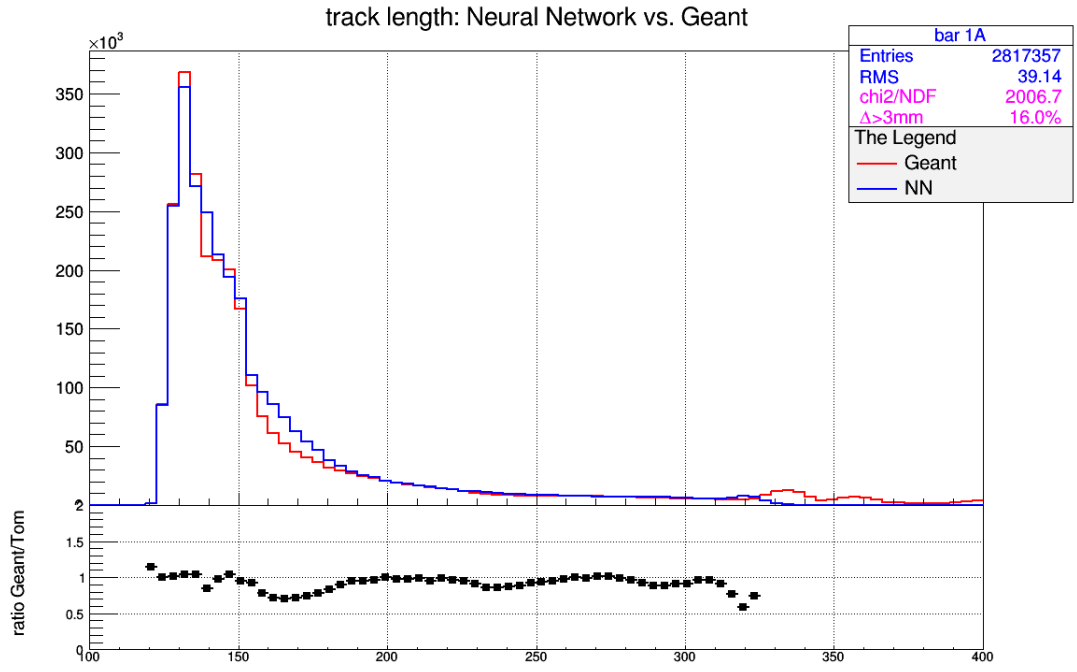


Figure 5.20: Comparison of photon track length given by Geant and calculated by NN.

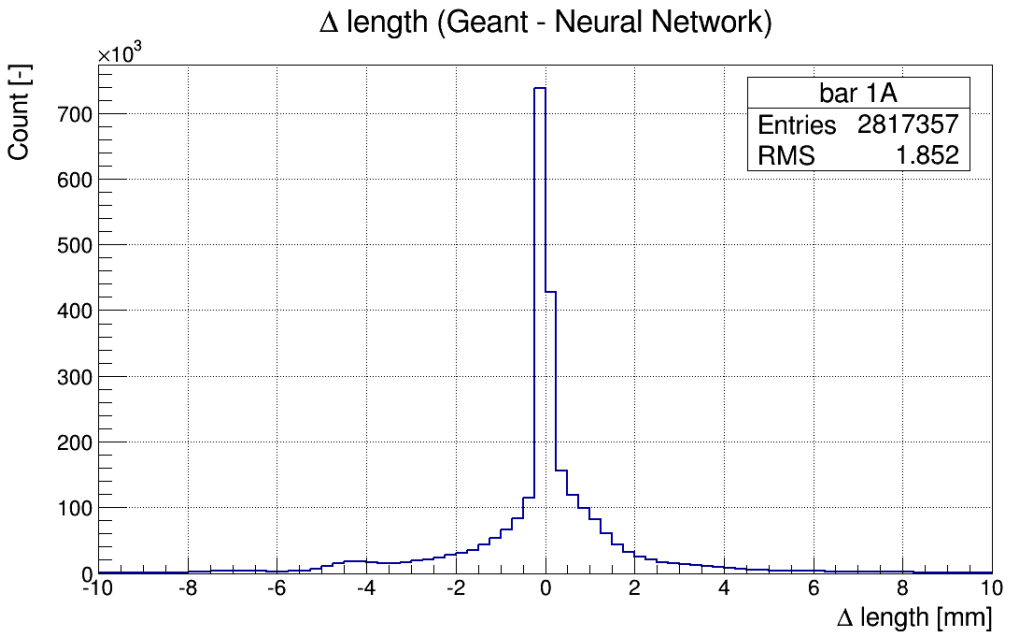


Figure 5.21: Histogram of photon track difference: track length Geant minus track length NN.

Although the simulation of this bar was more complicated (7 NNs were used for each interval of `vtxphi`), the training has good results. Notice that due to the effect of taper the secondary peak in the track length histogram is merged into the primary one.

5.5 Implementation of the Trains

It was found out that for the simulation of the whole train it is not necessary to train dedicated ANN for each bar. LQbars without taper differ from each other only by the length of radiator and light guide. These bars are in trains 3 and 4 which have only different width of radiators (the dimension in `vtxy` coordinate), 5 mm and 5.5 mm respectively. Therefore, it is possible to find a constant by which the track length histograms of bars are moved from each other.

Trains without taper have two peaks which have to be moved separately as same as the minimum between them. For example the distance between primary peaks of trains 3 and 4 is approximately 5.7 mm. The exact values of shifts are provided in the code of Fast Cherenkov enclosed on the CD in the library called `routines_network` in functions `move_train34` and `move_train12`. Thus it is possible to use the NN trained only for bar 3A to all bars in trains 3 and 4.

Shapes of track length histograms for trains with taper differ from each other more than these histograms for trains without taper. It is due to the train 1 has different width (2 mm) and the slope of the taper than train 2 that is 4 mm wide. These circumstances cause changes of the shape of the track length histogram within one train and the presence of the tertiary peak in the train 2. In order to describe the data sufficiently new ANN for the bar 1C has been trained. So to summarise: the NN for 1A is used to approximate bar 1B, 2A and 2B when necessary shift is applied. The NN for 1C approximates 1D, 2C and 2D.

We provide an example of shift of 1A to 1B (Figure 5.22) and 1A to 2A (Figure 5.23).

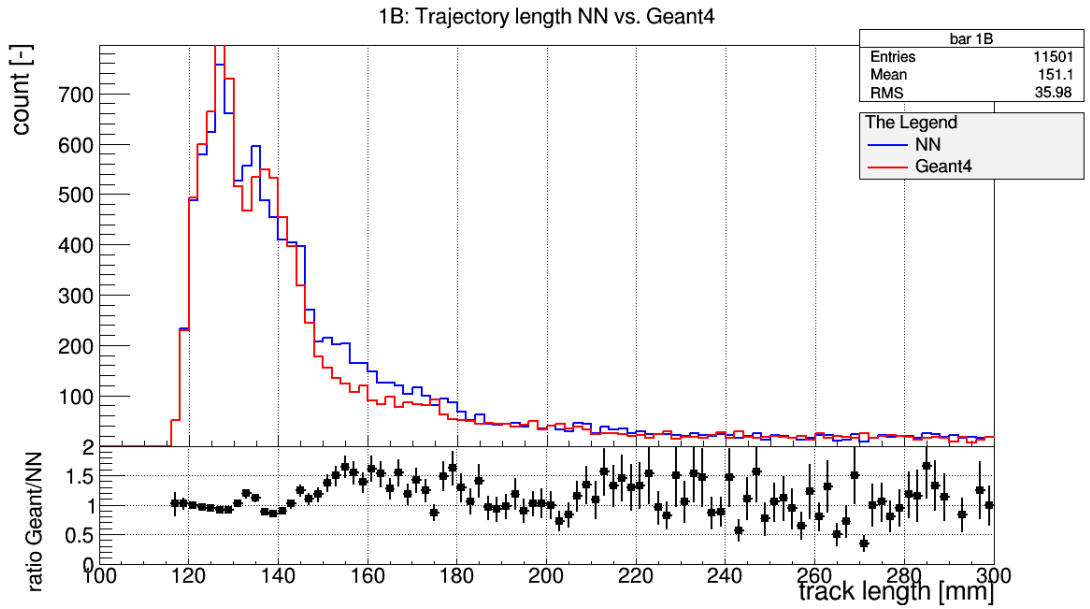


Figure 5.22: Histogram of photon track length from Geant for the LQbar 1B (red histogram) and track length calculated using the NN for bar 1A (blue histogram) with the appropriate shift applied.

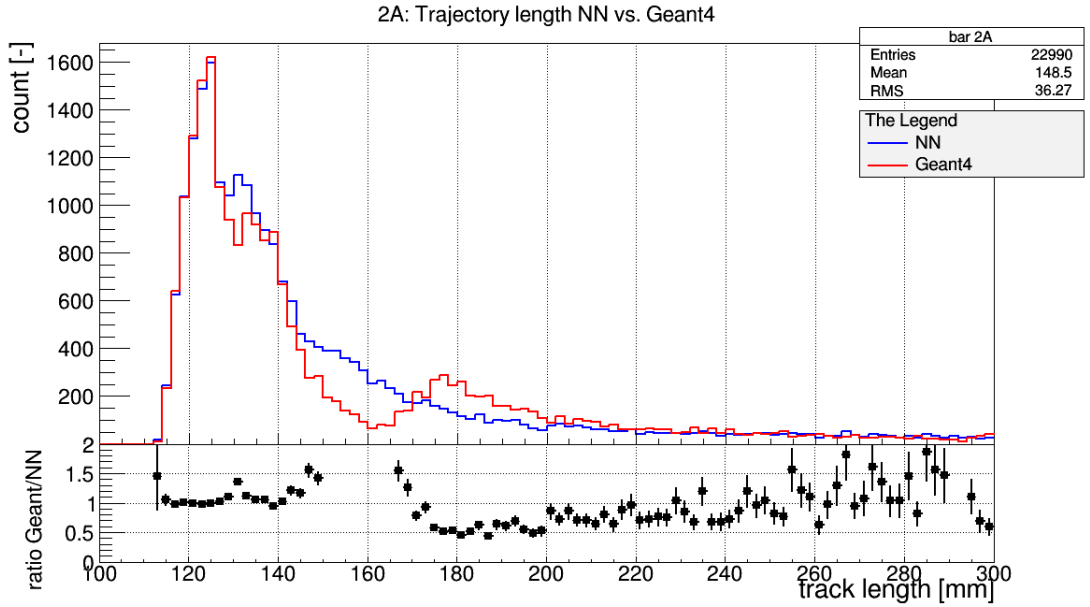


Figure 5.23: Histogram of photon track length from Geant for the LQbar 2A (red histogram) and track length calculated using the NN for bar 1A (blue histogram) with the appropriate shift applied.

The tertiary peak is omitted in this approximation as its importance is very low in this phase of simulation of the ToF detector. It is obvious that this shifting describes the data from the full simulation well.

5.5.1 Influence of the Proton's Smearing

In real operation of the LHC the protons measured by the AFP have non-zero *smearing* angle `gptheta` with respect to the *globzpos* - *globypos* plane. Magnitude of such an angle was considered 300 μrad. A corresponding data sample for the train 1 was created in order to find out the influence of the smearing on the NN. Result for the bar 1A is given in the Figure 5.24.

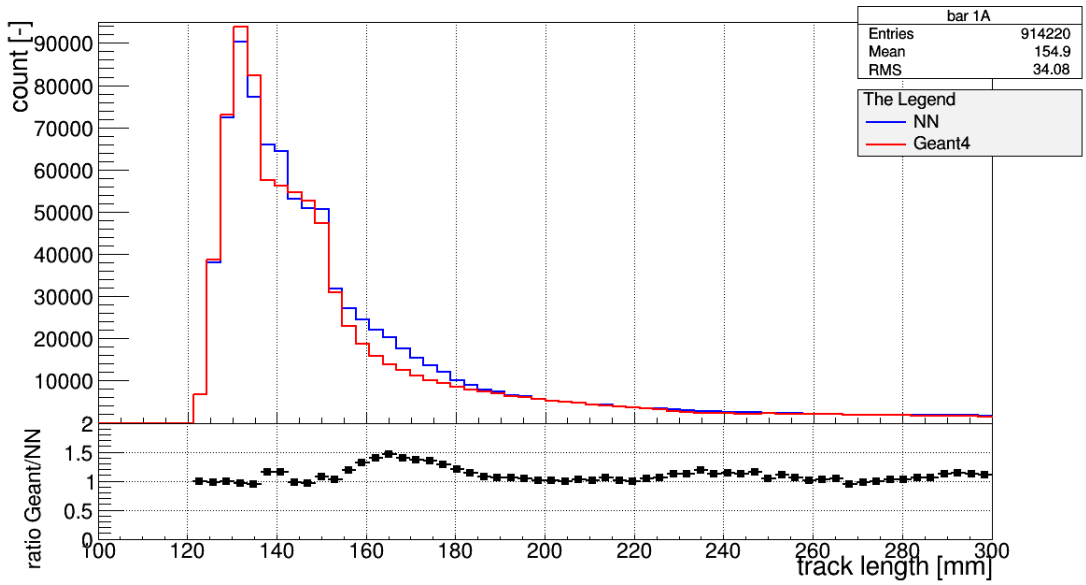


Figure 5.24: Histograms of photon track length for the LQbar 1A created using Geant data with non-zero smearing angle.

The effect of smearing is not noticeable which makes place for significant simplification of description of proton's paths through bars during the implementation of the Cherenkov photon's generator.

5.6 Implementation of the Generator of Cherenkov Photons

For purpose of final use of the Fast Cherenkov simulation it is crucial to implement algorithm for generation of Cherenkov photons. This procedure has several steps which are discussed within this section. The logics of the generator's function follows the way of implementation in Geant.

Functions for Cherenkov photon's generation are implemented in the library `routines_generator` and they are called by the function `Generator` of the `fast_cherenkov` library. This function is responsible for defining single proton's track along which photons are generated (by calling function `ProtonGenerator`). Coordinates and angles of photons along with other important variables are dumped into vectors (function `generator_photon`). It is iterated through these vectors and using trained NNs dedicated histogram for given bar is filled (function `network`). A simplified block diagram of the Fast Cherenkov code is shown in the Figure 5.25.

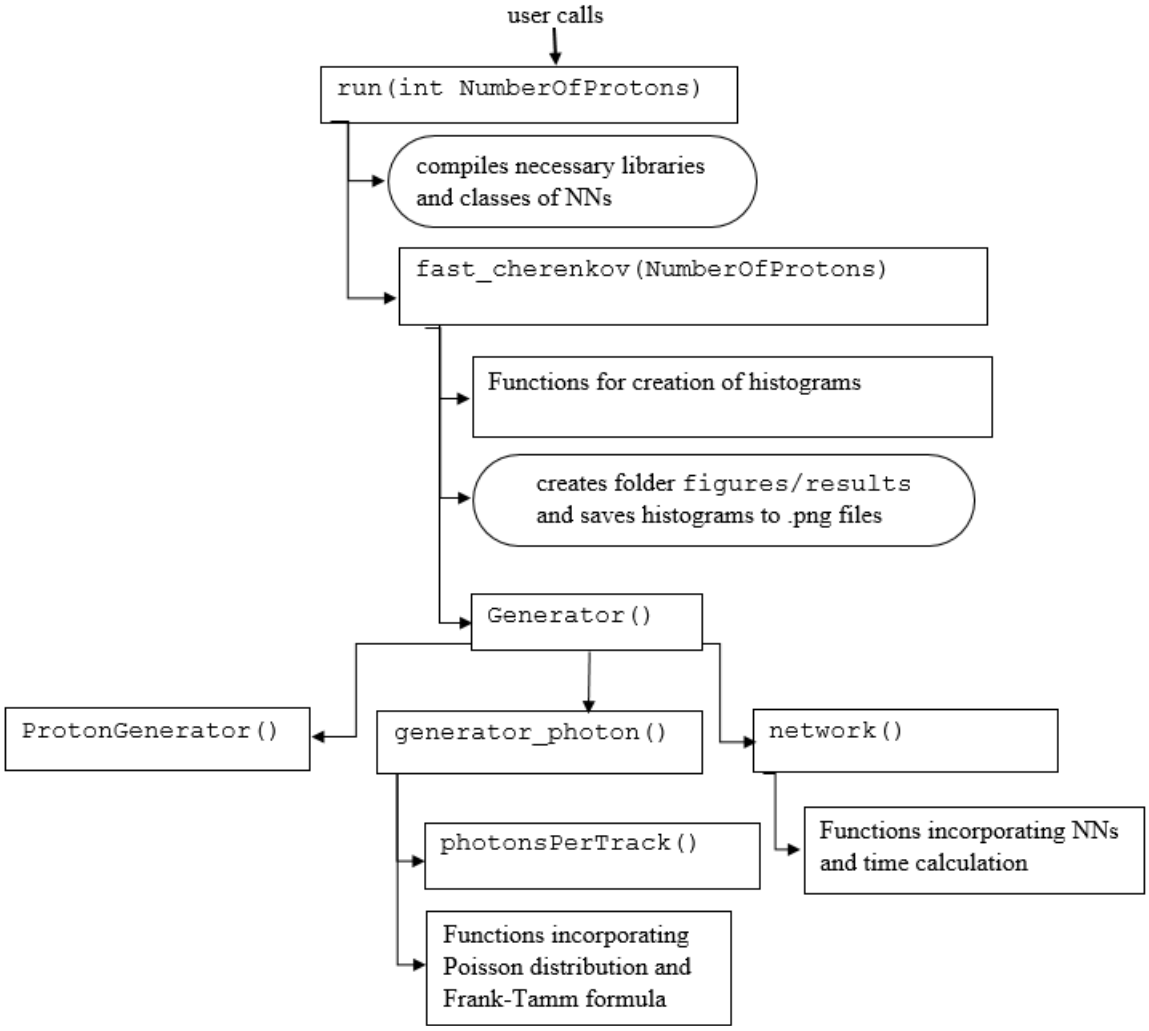


Figure 5.25: Simplified block diagram of the Fast Cherenkov simulation.

There are several steps that the generator has to perform:

- **Generation of the point where protons enter the bar**

  The position coordinates of protons while hitting the surface of the bar are generated by function `ProtonGenerator`. The NNs are trained only for protons coming in a 5 mm large window, see Figure 5.4. So the range of local x coordinate generated by this function, `vtxx` (Figure 5.5), is from 0 to $\frac{5}{\cos(42\,^\circ)}$ (length of the projection of the window in the side of the bar). In order to fill every train the range of the *globypos* coordinate spans from 0 to the value of hight of all four trains (between each train there is a space 0.1 mm wide). The *globypos* coordinate carries thus information which train was hit. However, in order to fill NNs with y coordinates they have been trained for local y coordinates of each train, `vtxy`, has to be generated separately. Ranges of `vtxy` for each train are given in the Table 5.3.

train	span of the interval of $vtxy$ [mm]
1	0 - 2
2	0 - 2
3	0 - 5
4	0 - 5

Table 5.3: Spans of generated y-coordinates for each train.

The direction vector of every generated proton is parallel to the cut edge of the bar. In local coordinates the vector is: $(-1; 0; \frac{1}{\tan(42\,^\circ)})$. This simplification is possible due to smearing of proton's direction has no effect on track length histograms (see Subsection 5.5.1).

- **Calculation of the mean number of photons per track**

The implementation of this calculation closely follows the theory described in the Chapter 2 of this thesis. The mean number of photons follows from the Poisson distribution whose parameter μ is obtained from the equation (1.11). The expression $\mathrm{d}E$ was substituted by the difference of maximal and minimal wavelength of photons that pass through the bar according to Geant ($654\,nm - 200\,nm = 454\,nm = 4.54 \cdot 10^{-5}\,cm$).

The value of the parameter *StepLength* was set different for each train so that the mean value of generated photons per proton track is by few procent higher than the value of photons that pass in Geant. This is necessary because unlike Geant the Fast Cherenkov simulation does not have any other mechanism to calculate probability whether the photons passes the bar or not except the effect of attenuation. The total number of generated photons in the generator may differ from the value in Geant because other effect play role, e.g. optical crosstalk whose influence is mentioned later.

Refractive index in the Frank-Tamm formula (equation (1.7)) is obtained by a random generation of wavelength which lies in the above interval. Then it is transformed into the refractive index. The transformation is based on fitting of the data given in the data sheet for suprasil UVL (enclosed in the Appendix A). Parameters of the fit are shown in the table in the Figure 5.26 and the fitting function is implemented in the function `RefractiveIndexFunction`.

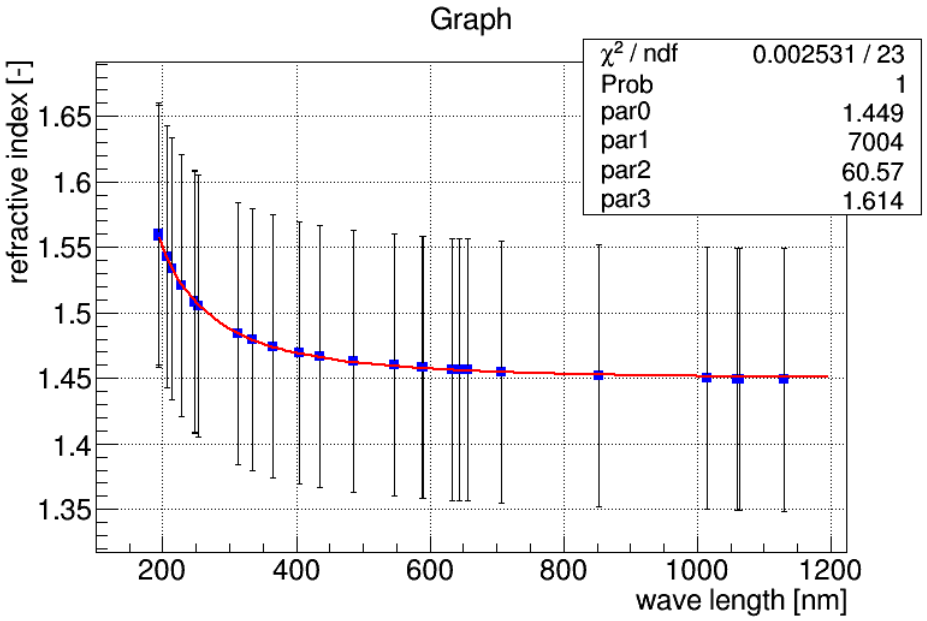


Figure 5.26: Dependece of photon's wavelength on the refractive index of suprasil UVL. False error bars were inserted to help ROOT in finding the best fit parameters.

LQbars are made of several types of suprasil; the UVL was chosen as an approximation of them. It is expected that potential variations have minor impact.

- **Generation of direction angles of photon**

It was discovered that the generator used together with NNs provides better results when values of angles `vtxtheta` and `vtxphi` are not generated uniformly, but their evolution copies shapes from Geant for photons that passed the bar. The kinematic region shaped according to Geant is in the Figure 5.27.

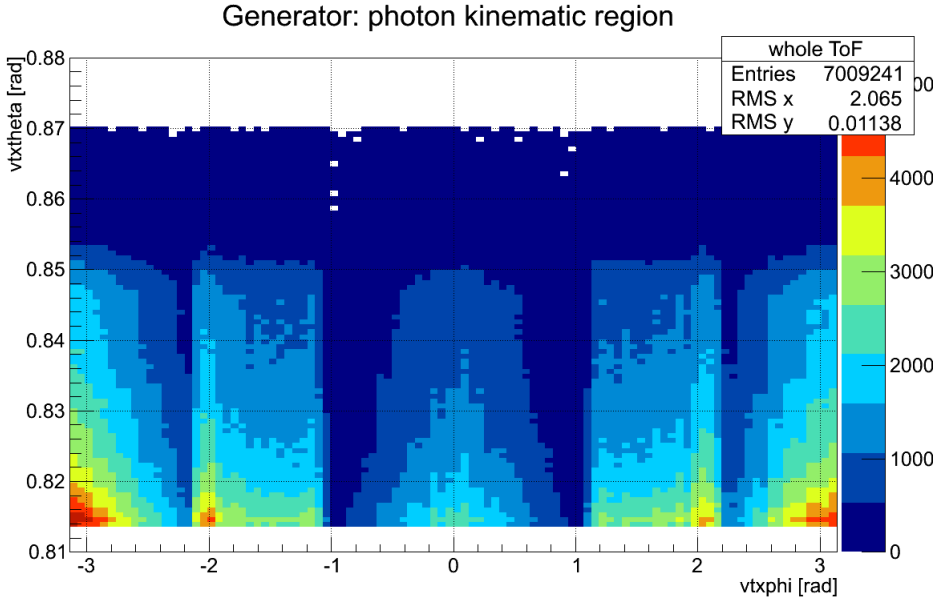


Figure 5.27: Kinematic region of angles `vtxtheta` and `vtxphi`.

This has been achieved by saving bin's values of normalised histograms of wavelength because it depends on refractive index and on the Cherenkov angle (equation 1.5)) and of `vtxphi`. These values are found in functions `generator_wl` and `generator_phi` of the library `routines_generator`. Because these histograms are normalised they have meaning of the probability, so using the generator of random numbers it is possible to shape the original uniform distribution of values of angles.

5.7 Results for the Whole ToF

In this chapter there are presented results for all bars in the ToF detector. In the function `network` of the `routines_network` library there is a code that distinguishes between bars and calculates corresponding track length and arrival time. The generator of Cherenkov photons is used as the source of data for neural networks and their output is compared with Geant data sample created for whole ToF.

5.7.1 Implementation of Attenuation

Before the track length histograms of bars are introduced it has to be explained how the attenuation is implemented. The relation for the probability of photon's absorption in the bar is found in the equation (5.1). The length L is obtained from the NN and for the attenuation coefficient applies relation:

$$\mu_{\text{abs}} = \frac{1}{L_{\text{abs}}} \ , \tag{5.4}$$

where $L_{\text{abs}} = L_{\text{abs}}(\lambda)$ is the attenuation length which is a function of wavelength. Its evolution is typical for the given material and its measurement for fused silica was published in the article [17]. According to the results from that article a plot and fit has been made (Figure 5.28).

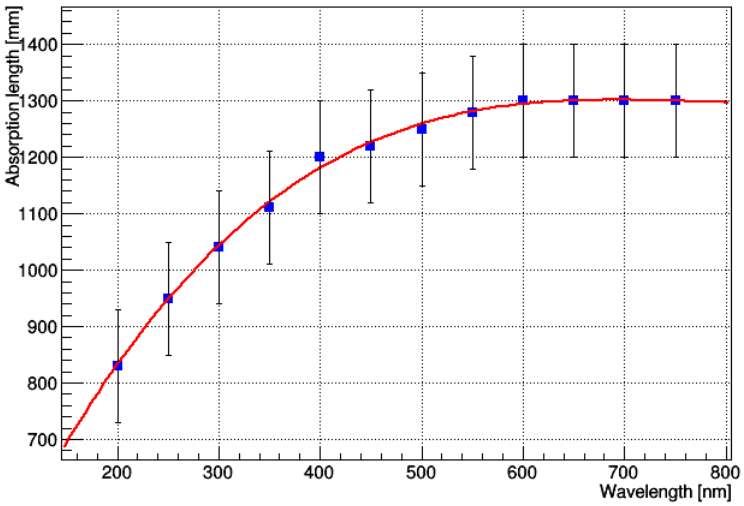


Figure 5.28: Dependence of photon's wavelength on attenuation length of suprasil. False error bars were inserted to help ROOT in finding the best fit parameters.

The fit function is polynomial of the third order and fit parameters are implemented in the function `likelihoodAbs` of the library `routines_network`.

5.7.2 Track Length Histograms

Track length histograms for one chosen bar of each train are presented now, because the shape of histograms within train is very much the same. The effect of attenuation is implemented as described above. In Geant there was prepared a data sample which comprises 8000 protons (only 7 178 protons actually passed through the bars) and this sample was compared to the output of the Fast Cherenkov simulation fed by its generator of photons.

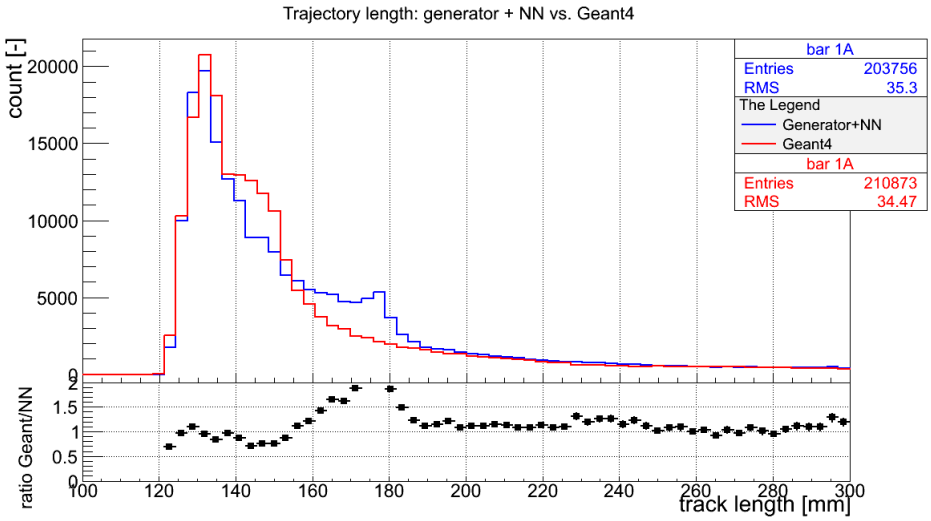


Figure 5.29: Track length histogram for bar 1A.

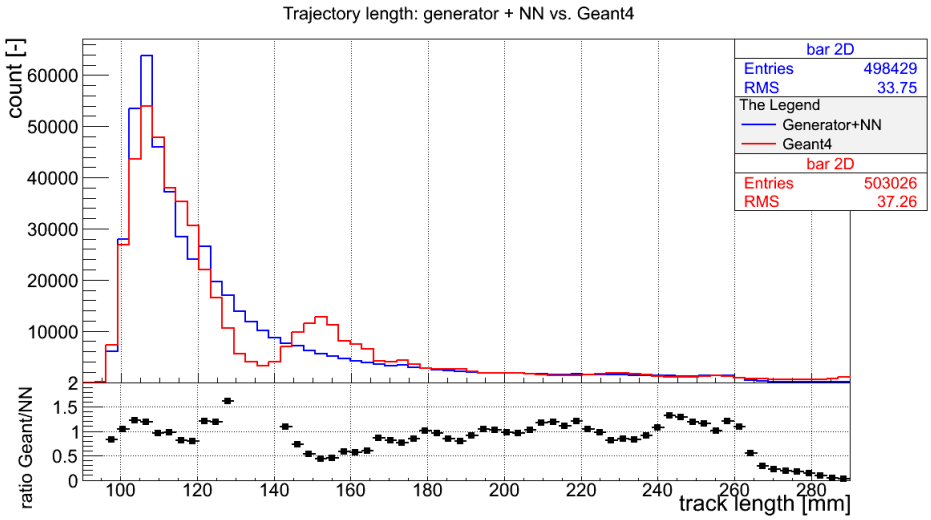


Figure 5.30: Track length histogram for bar 2D.

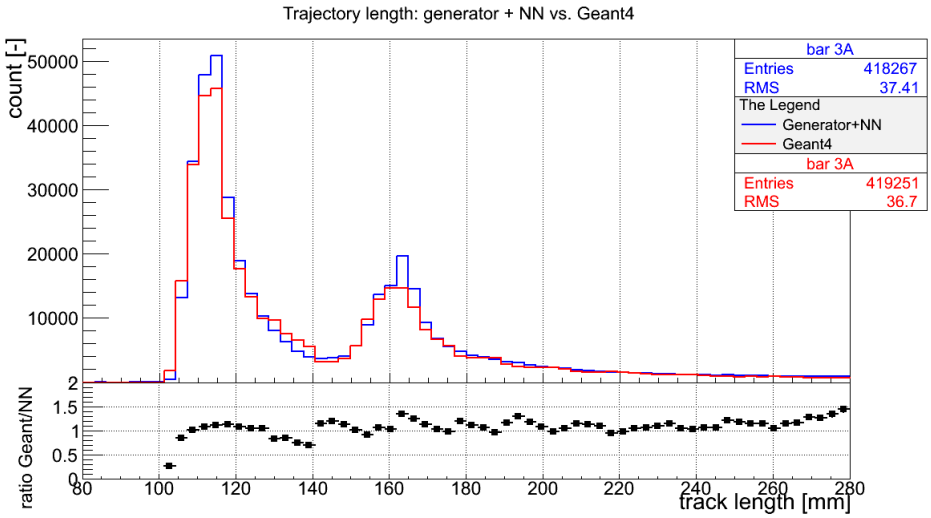


Figure 5.31: Track length histogram for bar 3A.

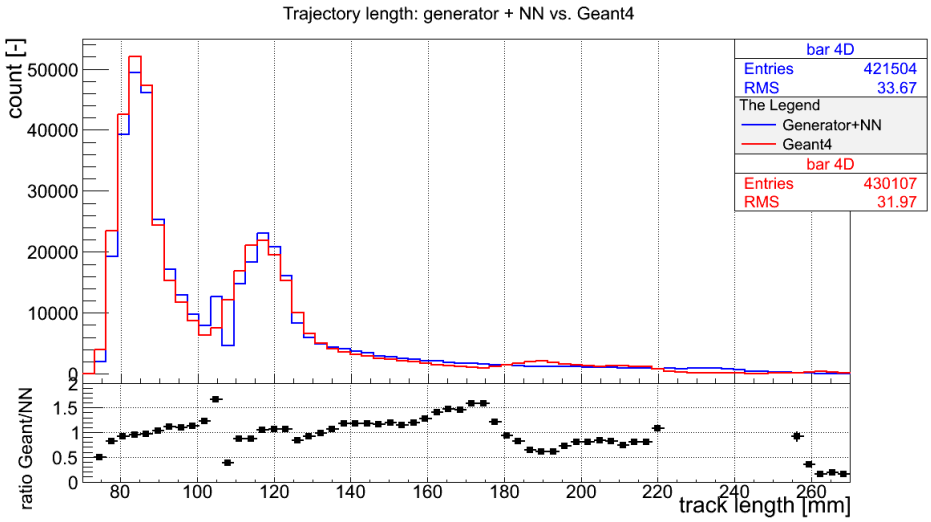


Figure 5.32: Track length histogram for bar 4D.

Especially bars of trains 3 and 4 are in very good agreement with Geant. However, bars in trains 1 and 2 are still sufficiently well described.

5.7.3 Influence of Optical Crosstalk

All the previous comparisons between the Fast simulation and Geant were done for photons that originated in one bar and got to the PMT from the same bar, i.e. the optical crosstalk between bars was neglected. Here we present histograms of Geant data without any filtration. To evaluate the impact of crosstalk bars 1A and 1B and 3A and 3B may be compared with each other as A bars are not influenced by the crosstalk.

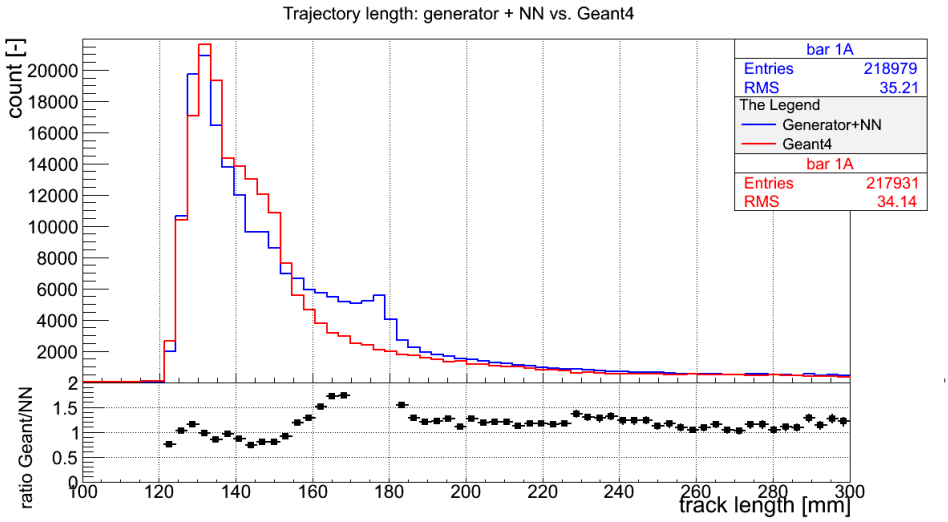


Figure 5.33: Comparison between Fast simulatin and Geant data with crosstalk for bar 1A.

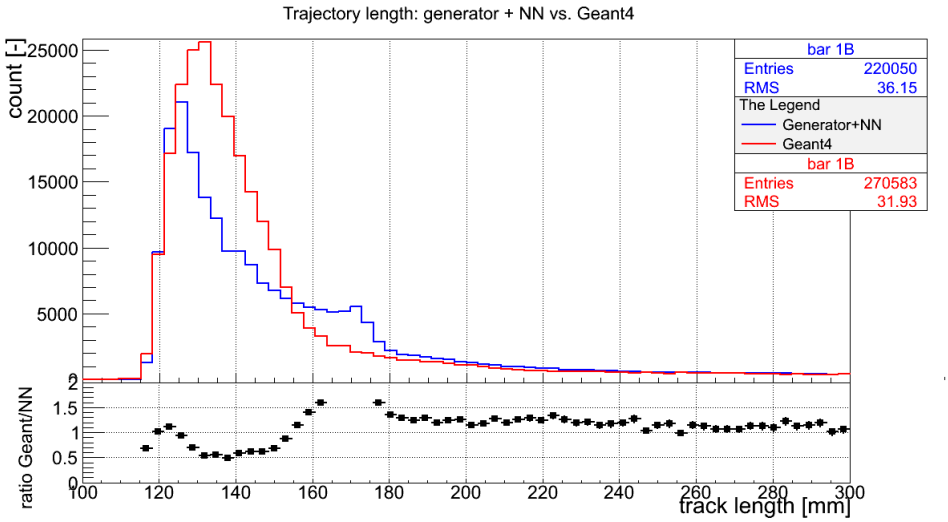


Figure 5.34: Comparison between Fast simulatin and Geant data with crosstalk for bar 1B.

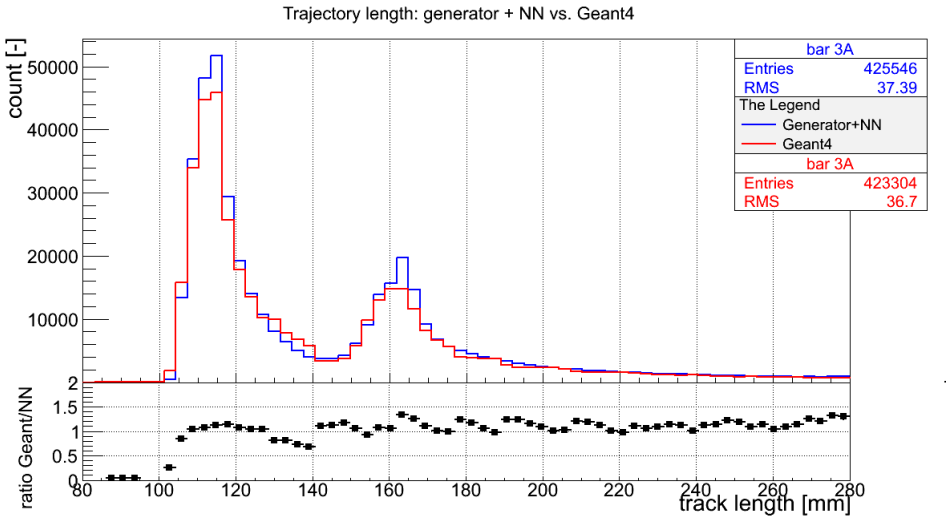


Figure 5.35: Comparison between Fast simulatin and Geant data with crosstalk for bar 3A.

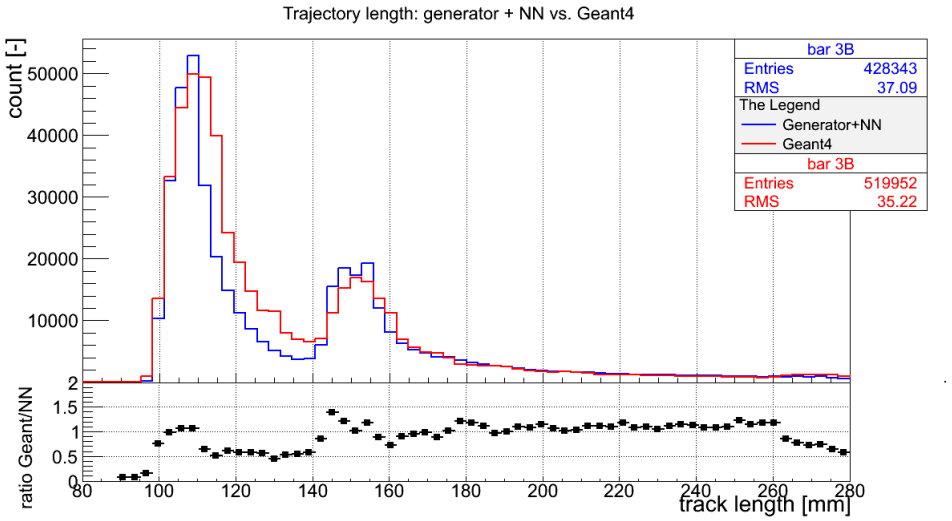


Figure 5.36: Comparison between Fast simulatin and Geant data with crosstalk for bar 3B.

The impact of crosstalk lies in increase of photon's count and in change of the histograms' shape. Actual version of the Fast Cherenkov simulation does not contain this effect.

5.7.4 Arrival Time Histograms

The arrival time was calculated obtained using relation (5.2) and (5.3). Histograms for the same bars like in the Subsection 5.7.2. are provided.

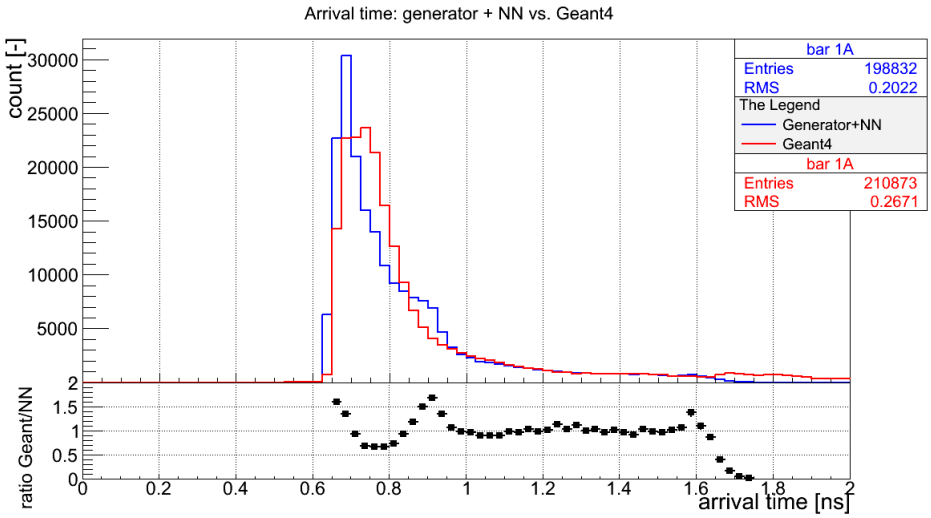


Figure 5.37: Arrival time histograms for bar 1A.

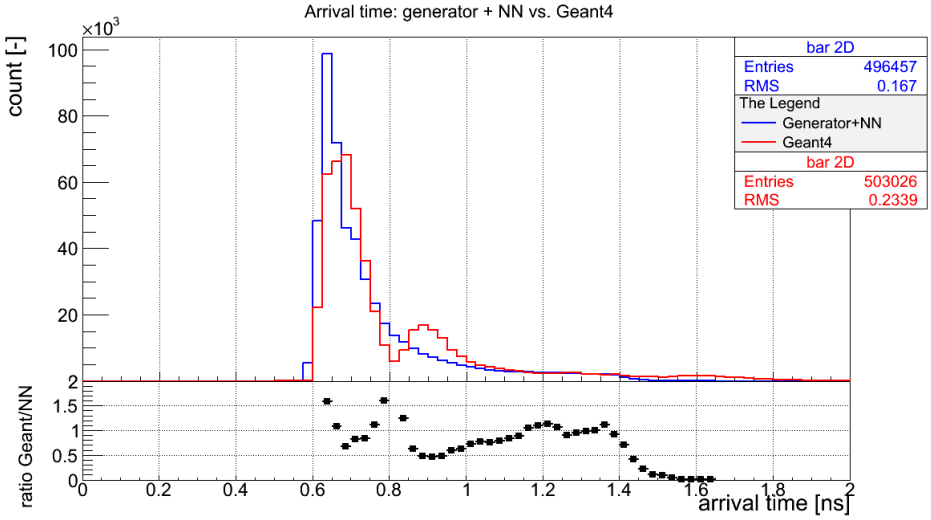


Figure 5.38: Arrival time histograms for bar 2D.

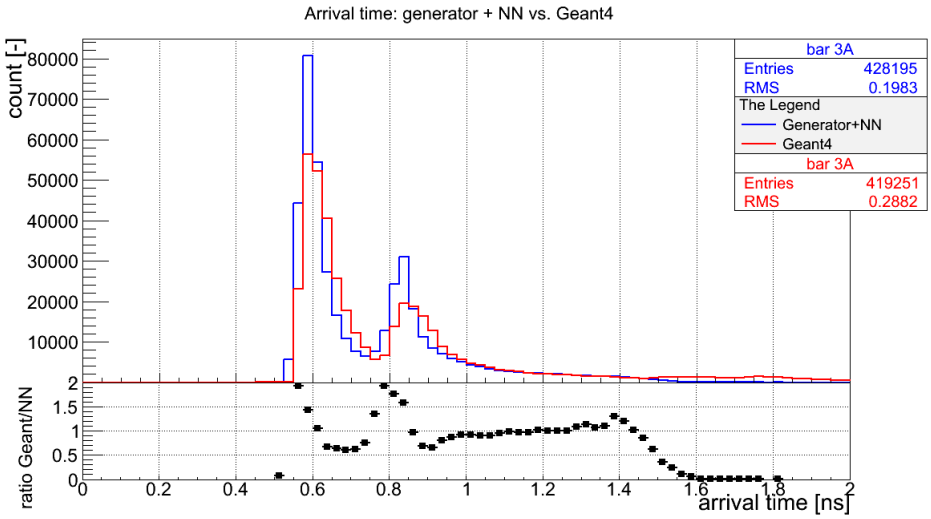


Figure 5.39: Arrival time histograms for bar 3A.

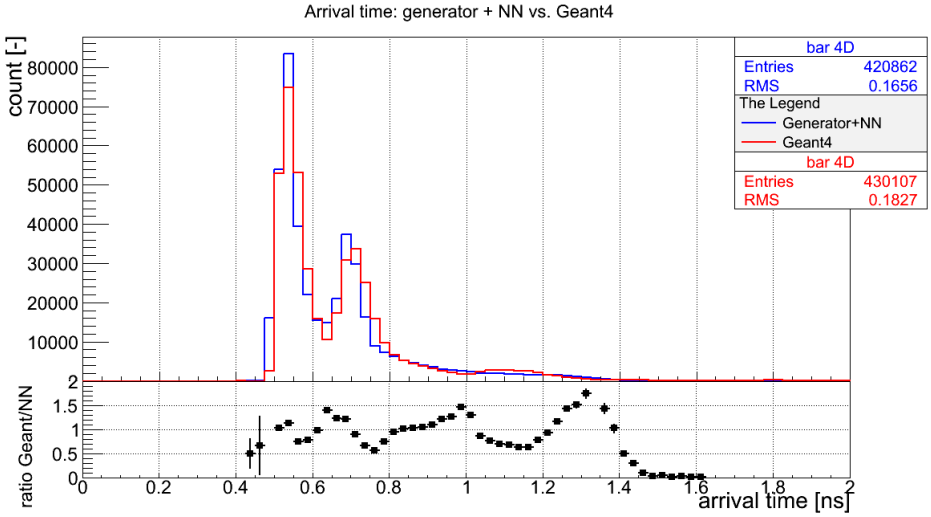


Figure 5.40: Arrival time histograms for bar 4D.

Conclusion

The goal of this master thesis is an implementation of the code for the transmission of Cherenkov radiation through LQbars with and without taper and comparison of results with the output of the full simulation in Geant4.

The first strategy of implementation was based on principles of analytic geometry. The trajectory of photon through the LQbar was straightened and its length calculated. The algorithm differentiated between three scenarios of photon's transmission: photon reflects from the metallised mirror to the PMT or misses the mirror but still gets to the PMT and photon reflects from the cut edge (Figures 4.2(a) - 4.4(b)). To determine whether the photon transmits through the bar or leaves it the fulfilment of conditions of total internal reflection was tested. In case these conditions were not met the particle was considered lost. The likelihood of photon's reflectance on the border between silica and air when the angle of incidence is smaller than the critical angle was neglected.

Into the final track length histograms there were implemented effects of reflectivity on the Al mirror, transmission of the fused silica and attenuation in the glue connecting radiator and light guide. Resulting histogram for all three scenarios is in the Figure 4.12. The disagreement between Geant data and geometrical model is considerable. We assume that this might have negative impact on quality of the pulse from the PMT. The reason of this disagreement is most probably the error in the code and the neglecting of probability of photon's reflectance when critical angle is not reached could also play role.

The second strategy of implementation of the Fast Cherenov code is based on the employment of neural networks for regression of track length histograms. Usage of NNs is supported by the Universal Approximation Theorem (see Section 3.1). The continuous function approximated by the NN is obtained from the dependence of track length on the angle `vtxphi`. When appropriate cuts are applied a piecewise smooth function arise.

To evaluate the correct function of the trained neural network a Geant data sample was compared with the output of the NN for the very same data. The results for bar with and without taper respectively are shown in Figures 5.20 and 5.18. These plots show that NNs describe the Geant data well. NNs were trained only for three chosen bars; the others were approximated by the shift of the track length histogram by some constant.

An important part of the Fast Cherenkov simulation is the generator of Cherenkov photons that supplies neural networks with input data. To verify the correct functioning of the generator we generated protons in the same volume of bars like in the Geant data sample and compared both (Figures 5.29 - 5.32). In these histograms there is implemented the effect of attenuation of light in the fused silica. The track length

histograms were subsequently converted into arrival time distributions (Figures 5.37 - 5.40).

It has to be mentioned that during a real operation of the ToF detector there is an influence of light leakage between bars. Its impact is visible from the Figures 5.33 - 5.36. The aim of the presented Fast simulation was not to describe this effect.

The main difference between the Fast simulation and Geant lies in the statistics of photons and in case of some bars deformation of the shape. Photon's statistics influences only the amplitude of the output pulse of the PMT. The usage of the CFD ensures that the time of rising edge of signals with higher and smaller amplitude is set the same. Concerning the shape of the distributions it is still very similar to the Geant, so we expect that the response of the PMT should be also similar. To quantify the impact on the overall time resolution more precise model of MCP-PMT would be needed.

Finally, the CPU time consumption of Fast Cherenkov and Geant was evaluated. Fast simulation is 2.6 times faster than Geant for the same data sample. It is worth stressing that this value depends on the machine used for running, too. This result also makes the Fast Cherenkov simulation being an suitable alternative to the full simulation in Geant4.

Bibliography

[1] Web page of Methods of Experimental Physics Laboratory at the University of Minnesota, in: Generating Cerenkov Radiation. http://mxp.physics.umn.edu/s04/projects/s04cherenkov/theory.htm.

[2] Web page of Pinterest, in: Breaking sound barrier. https://cz.pinterest.com/nycon02/breaking-sound-barrier/.

[3] Web page of The Spectrum of Riemannium, in: Tamm-Frank formula. https://thespectrumofriemannium.wordpress.com/tag/tamm-frank-formula/.

[4] Web page of High Energy Physics Group at Universidad Complutense de Madrid, in: Cherenkov Radiation. http://www.gae.ucm.es/ emma/docs/tesina/node4.html, 2001.

[5] Web page of Skulls in the Stars, in: Reversing optical "shockwaves" using metamaterials. https://skullsinthestars.com/2009/11/20/reversing-optical-shockwaves-using-metamaterials/, 2009.

[6] WikiBook Artificial Neural Networks. 2013, available online: https://en.wikibooks.org/wiki/Artificial_Ne- ural_Networks/Print_Version.

[7] ROOT User's Guide, in: Fitting. 2014, available online: https://root.cern.ch/root/htmldoc/guides/users-guide/ROOTUsersGuideA4.pdf.

[8] Web page of Forward detectors of the ATLAS experiment, in: The ATLAS Forward Detector project. http://atlas-project-lumi-fphys.web.cern.ch/atlas-project-lumi-fphys/images/forward_detectors.jpg, 2014.

[9] Web page of ResearchGate. https://www.researchgate.net/figure/273768094_fig2-_Figure-4-A-hypothetical-example-of-Multilayer-Perceptron-Network, 2014.

[10] Web page of Wikipedia online encyclopedia, in: Constant Fraction Discriminator. https://en.wikipedia.org/wiki/Constant_fraction_discriminator, 2014.

[11] Web pages of Stanford University. http://large.stanford.edu/courses/2014/ph241/a-laeian2/, 2014.

[12] Web page ROOT Tutorials, in: Neural Networks. 2015, available online: https://root.cern.ch/root/html/tutorials/mlp/.

[13] Web page of Wikipedia online encyclopedia, in: Fresnel equation. https://en.wikipedia.org/wiki/Fresnel_equations, 2017.

[14] Web page of Wikipedia online encyclopedia, in: Sigmoid Function. https://en.wikipedia.org/wiki/Sigmoid_function, 2017.

[15] Web page of Wikipedia online encyclopedia, in: Universal Approximation Theorem. https://en.wikipedia.org/wiki/Universal_approximation_theorem, 2017.

[16] ADAMCZYK, L.; et al: *Technical Design Report for the ATLAS Forward Proton Detector*. CERN, available from the CDS web page: https://cds.cern.ch/record/2017378/files/ATLAS-TDR-024.pdf.

[17] ALBROW, M. G.; et al: Quartz Cherenkov counters for fast timing: QUARTIC. *Journal of Instrumentation*, volume 7, 10, 2012: page 10027, available online: http://stacks.iop.org/1748-0221/7/i=10/a=P10027.

[18] DYNDAL, M.; SHOEFFEL, L.; SYKORA, T.: *Fast Cherenkov model of optical photons generation and transportation*. CERN, 2017, available from the CDS web page: https://cds.cern.ch/record/2241362/files/ATL-SOFT-PUB-2017-001.pdf.

[19] JOSELL7: https://commons.wikimedia.org/w/index.php?curid=21670922, 2014, wikipedia online encyclopedia.

[20] KOMÁREK, T.: *Simulation of optical part of high-energy particle time-of-flight detector and comparison with data*. Master thesis, Palacky University Olomouc, Faculty of Experimental Physics, Olomouc, 2016 [cit. 2017-03-30], available online: http://theses.cz/id/zts9l9/.

[21] MARSHALL, Z.: Simulation of Pile-up in the ATLAS Experiment. *Journal of Physics: Conference Series*, volume 513, 2, 2014: page 022024, available online: http://stacks.iop.org/1742-6596/513/i=2/a=022024.

[22] NOZKA, L.; et al: Design of Cherenkov bars for the optical part of the time-of-flight detector in Geant4. *Opt. Express*, volume 22, 23, Nov 2014: pages 28984–28996, doi:10.1364/OE.22.028984, available online: http://www.opticsexpress.org/abstract.cfm?URI=oe-22-23-28984.

[23] NOZKA, L.; et al: Construction of the optical part of a time-of-flight detector prototype for the AFP detector. *Opt. Express*, volume 24, 24, Nov 2016: pages 27951–27960, doi:10.1364/OE.24.027951, available online: http://www.opticsexpress.org/abstract.cfm?URI=oe-24-24-27951.

[24] TRZEBINSKI, M.: Probing the BFKL Pomeron with Future ATLAS Forward Detectors. In *Proceedings, International School on High Energy Physics : Workshop on High Energy Physics in the near Future. (LISHEP 2013): Rio de Janeiro, Brazil, March 17-24, 2013*, available online: https://inspirehep.net/record/1233876/files/arXiv:1305.3996.pdf.

[25] TRZEBINSKI, M.; CHWASTOWSKI, J.; ROYON, C.: *Study of QCD and Diffraction with the ATLAS detector at the LHC*. PhD thesis, Cracow, INP and U. Paris-Sud 11, Dept. Phys., Orsay, 2013, available online: http://cds.cern.ch/record/1616661.

[26] VALENTINETTI, S.: *Luminosity measurements with the LUCID detector in the ATLAS experiment*. PhD thesis, Bologne U., 2011, available online: https://cds.cern.ch/record/2094394.

Appendix A

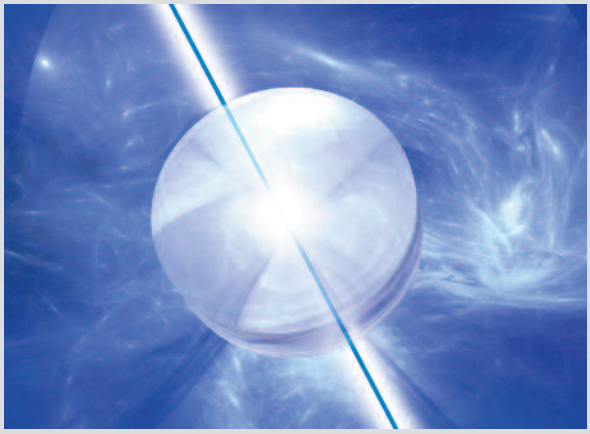
Heraeus

Suprasil® UVL

Suprasil® UVL synthetic fused silica is manufactured using a patented, environmentally friendly process resulting in a glass of exceptional purity and excellent visual quality. It is a very homogeneous synthetic fused silica glass for deep UV optical applications.

Suprasil® UVL is chlorine-free resulting in outstanding laser damage resistance due to the reduced tendency to form E' centres.

Suprasil® UVL is free of bubbles and inclusions and due to its ultra-high purity, has exceptional optical transmission in the deep ultraviolet and visible, with a useful range from below 180 nm through to 2000 nm.



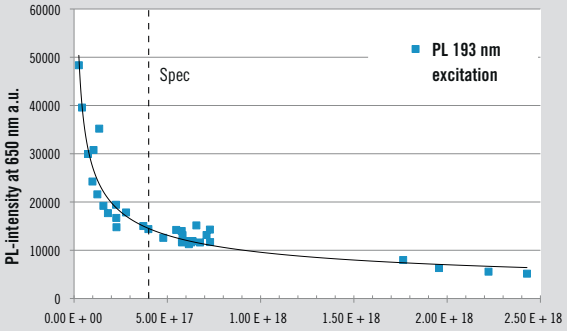
Suprasil® UVL

Refractive Index Homogeneity[1]	≤ 10 ppm – lower values upon request
Striae	
Visible striae	3 Directions Free
ISO 10110-4	Class 5 in Functional Direction
MIL-G-174B	A in Functional Direction
Birefringence / Residual Strain[1]	
(Typical values)	≤ 5 nm/cm
Bubbles	
Bubble class (DIN 58927)	0
Maximum number of inclusions[2]	0
Fluorescence (254 nm excitations)	None

1) Stress induced birefringence and optical homogeneity are valid for 80 % of the diameter of an ingot or for 90 % diameter of a machined component.

2) Bubbles and Inclusion with ø ≤ 80 µm are not counted. Inclusion free down to 10 µm upon request.

Photoluminescence @ 193 nm



Measurement parameters: λ_{Exc} = 193 nm; Rep. Rate = 200 Hz; P = 50 mJ
H_2 specification: ≥ $4 \cdot 10^{17}$ molecules / cm^3 in CA

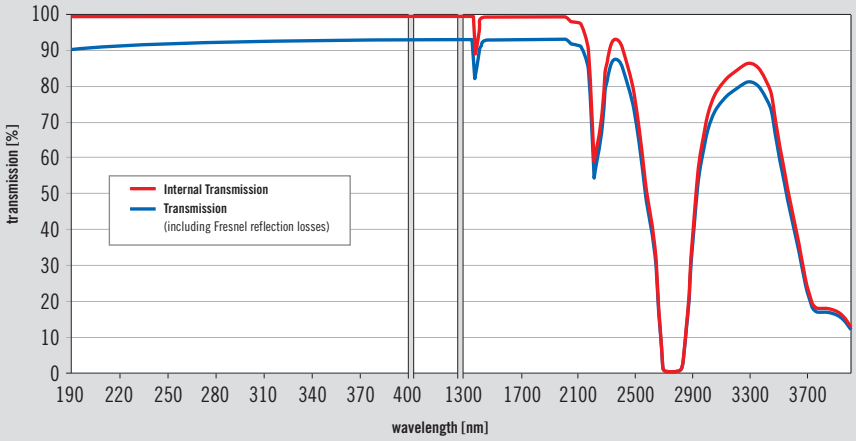
Transmission – Typical Internal Transmission (10 mm path length)

	λ = 193.4 nm	λ = 248 nm
Suprasil® UVL	> 99 %	> 99.5 %

Heraeus

Typical Transmission Spectrum

Sample thickness: 10 mm



Typical Chemical Analysis

Typical trace elements in ppb	Al	Ca	Co	Cr	Cu	Fe	K	Li	Mg
Suprasil® UVL	<10	<10	<10	<10	<10	<10	<10	<10	<10

Typical trace elements in ppb	Mn	Na	Ti	V	Zn	Zr	in ppm	Cl	OH
Suprasil® UVL	<10	<10	<10	<10	<10	<10		<0.15	800-1200
								(limits of detection)	

Other Properties

Abbe number: 67.8
Density: 2.2 g/cm^3
Hardness (Mohs scale): 6-7

Thermal Properties

Strain point*: 950°C
Annealing point*: 1100°C
Softening point*: 1710°C
Coefficient of thermal expansion:
(Average K^{-1} 0-600°C) 0.54 x 10^{-6}

*Note that these values may vary, depending on the thermal history of the glass.

Refractive Index & Thermal Coefficient (at 20°C & 1 bar / 760 mm Hg)

				n t	n s	n r	n c	n c'	n He-Ne	n D
Wavelength (nm)	1128.95	1064.00	1060.00	1013.98	852.11	706.52	656.27	643.85	632.80	589.29
Refractive Index (n)	1.44887	1.44963	1.44968	1.45024	1.45247	1.45515	1.45637	1.45670	1.45702	1.45840
Thermal Coefficient (dn/dT (ppm/K))	9.6	9.6	9.6	9.6	9.7	9.9	9.9	10.0	10.0	10.1

	n d	n e	n F'	n g	n h	n l				n KrF
Wavelength (nm)	587.56	546.07	486.13	435.83	404.66	365.01	334.24	312.66	253.73	248.30
Refractive Index (n)	1.45846	1.46008	1.46313	1.46669	1.46962	1.47454	1.47975	1.48447	1.50547	1.50838
Thermal Coefficient (dn/dT (ppm/K))	10.1	10.2	10.4	10.6	10.8	11.2	11.6	12.0	13.9	14.2

						n ArF		
Wavelength (nm)	248.00	228.87	214.51	206.27	194.23	193.40	193.00	184.95
Refractive Index (n)	1.50855	1.52109	1.53365	1.54259	1.55884	1.56014	1.56077	1.57495
Thermal Coefficient (dn/dT (ppm/K))	14.2	15.5	17.0	18.1	20.3	20.5	20.6	22.7

Germany

Heraeus Quarzglas GmbH & Co. KG
Optics
Quarzstraße 8
63450 Hanau
Phone +49 (6181) 35-62 85
Fax +49 (6181) 35-62 70
sales.hqs.optics.de@heraeus.com

China

Heraeus ShinEtsu Quartz (China), Inc.
QianJiang Tower
20th Floor, Room A
No. 971 Dongfang Road
200122 Shanghai
Phone +86 (21) 68672266-809
Fax +86 (21) 68751434
sales.hqs.optics.cn@heraeus.com

USA

Heraeus Quartz America, LLC.
Optics
100 Heraeus Blvd.
Buford, GA 30518
Phone +1 (678) 714-4350
Fax +1 (678) 714-4355
sales.hqs.optics.us@heraeus.com

UK

Heraeus Quarzglas GB
4 Tannery House
Tannery Lane
Send, Woking
Surrey GU23 7EF
Phone +44 (1483) 213323
Fax +44 (1483) 213329
sales.hqs.optics.uk@heraeus.com

www.optics.heraeus-quarzglas.com

HQS-SO Suprasil UVL 05.2011/V1.1_E/Lunis

Appendix B

Contents of enclosed CD

- folder `fast_cherenkov_simulation`:
  - `commonhdr_local.h`
  - `fast_cherenkov.c`
  - `fast_cherenkov.h`
  - `README.txt`
  - `routines_generator.c`
  - `routines_generator.h`
  - `routines_network.c`
  - `routines_network.h`
  - `run.c`
  - `25_25hi_5000ep_1A_new_moved_random_centre_midwing/`
  - `25_25hi_5000ep_1A_new_moved_random_sectorA_midwing/`
  - `25_25hi_5000ep_1A_new_moved_random_sectorB_midwing/`
  - `25_25hi_5000ep_1A_new_moved_random_trlen_325_vtxphi_1_downwing/`
  - `25_25hi_5000ep_1A_new_moved_random_trlen_325_vtxphi_1_upwing/`
  - `25_25hi_5000ep_1A_new_moved_random_vtxphi_2_58_sector1_downwing/`
  - `25_25hi_5000ep_1A_new_moved_random_vtxphi_2_58_sector1_upwing/`
  - `25_25hi_5000ep_1C_new_moved_random_centre_midwing/`
  - `25_25hi_5000ep_1C_new_moved_random_sectorA_midwing/`
  - `25_25hi_5000ep_1C_new_moved_random_sectorB_midwing/`
  - `25_25hi_5000ep_1C_new_moved_random_trlen_275_vtxphi_1_downwing/`
  - `25_25hi_5000ep_1C_new_moved_random_trlen_275_vtxphi_1_upwing/`
  - `25_25hi_5000ep_1C_new_moved_random_vtxphi_2_6_sector1_downwing/`
  - `25_25hi_5000ep_1C_new_moved_random_vtxphi_2_6_sector1_upwing/`
  - `25_25hi_5000ep_3A_new_moved_random_midwing/`
  - `25_25hi_5000ep_3A_new_moved_random_trlen_290_vtxphi_1_downwing/`
  - `25_25hi_5000ep_3A_new_moved_random_trlen_290_vtxphi_1_upwing/`
- `inputTree_1A_new_moved_random_1000Evt.root`

- tofsim_set14_1A_new_moved_random_1000Evt.root
- jirakovaK_diploma_thesis_printVersion.pdf
- jirakovaK_diploma_thesis_versionWithReferences.pdf