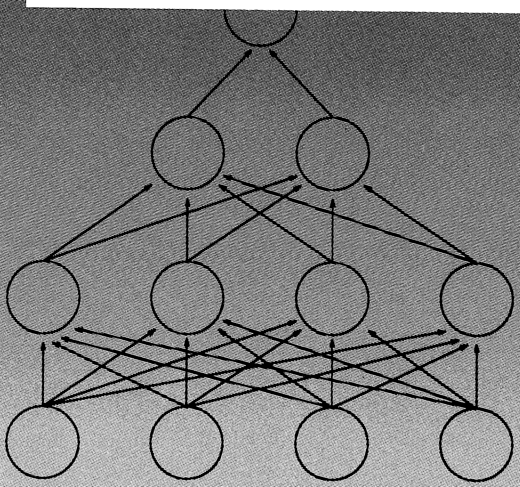
~~EX - 10448~~
~~SW 9549~~

A Measurement of the B^0-$\bar{B}^0$ Mixing Parameter at LEP Using a Neural Network

CERN LIBRARIES, GENEVA
CM-P00068770



Thesis-1995-Los

Martin Ernst Los

A Measurement of the $B^0\bar{B}^0$ Mixing Parameter at LEP Using a Neural Network

ACADEMISCH PROEFSCHRIFT

TER VERKRIJGING VAN DE GRAAD VAN DOCTOR
AAN DE UNIVERSITEIT VAN AMSTERDAM,
OP GEZAG VAN DE RECTOR MAGNIFICUS
PROF. DR. P.W.M. DE MEIJER
TEN OVERSTAAN VAN EEN DOOR HET COLLEGE
VAN DEKANEN INGESTELDE COMMISSIE
IN HET OPENBAAR TE VERDEDIGEN
IN DE AULA DER UNIVERSITEIT
OP MAANDAG 27 NOVEMBER 1995 TE 12:00 UUR

door

Martin Ernst Los

geboren te Heemskerk

#213107

Promotor: *Prof. Dr. A.N. Diddens*
Co-promotor: *Dr. J.J.M. Timmermans*

Overige leden: *Prof. Dr. J.J. Engelen*
Prof. Dr. Ir. F.C. Erné
Prof. Dr. K.J.F. Gaemers
Prof. Dr. Ir. F.C.A. Groen
Prof. Dr. R.H.P. Kleiss

Faculteit der Wiskunde, Informatica, Natuurkunde en Sterrenkunde

NIKHEF FOM

The work described in this thesis is part of the research program of 'het Nationaal Instituut voor Kernfysica en Hoge-Energie Fysica (NIKHEF-H)' in Amsterdam. The author was financially supported by 'de Stichting voor Fundamenteel Onderzoek der Materie (FOM)'.

Aan mijn ouders

Contents

Introduction **1**

1 Theory **3**
- 1.1 The Standard Model 3
- 1.2 Heavy Flavour Physics 5
  - 1.2.1 B^0-$\bar{B}^0$ Mixing 6

2 Neural Networks **11**
- 2.1 Neural Network Concepts 11
- 2.2 Learning 13
  - 2.2.1 Coincidence Learning 13
  - 2.2.2 Performance Learning 15
  - 2.2.3 Competitive Learning 15

3 Back-propagation in Feed-forward Neural Networks **17**
- 3.1 Architecture and Operation 17
- 3.2 Capabilities of a Feed-forward Network 20
- 3.3 Practical Issues using a Back-propagation Network 21
  - 3.3.1 Network Architecture 21
  - 3.3.2 Setting the Training Parameters 22
  - 3.3.3 Achieving Generalisation 23
- 3.4 High Energy Physics Applications 23
  - 3.4.1 Quark-Gluon Separation 23
  - 3.4.2 Noise Suppression 24

4 The DELPHI Detector **25**
- 4.1 Tracking Detectors 25
- 4.2 Calorimetry 27
- 4.3 Particle Identification 28
- 4.4 Other Sub-detectors 29
- 4.5 Data Taking and Data Handling 29

5 The Inner Detector **31**
- 5.1 The Jet Chamber 31
- 5.2 The Trigger Layers 32
- 5.3 The ID-OD Trigger 32
- 5.4 The Inner Detector Calibration Database 35
- 5.5 Z Measurement with the Trigger Layers 36
  - 5.5.1 Results 36

6 Determination of the B^0-$\bar{B}^0$ Mixing Parameter **39**
6.1 Muon Identification . . . 39
6.2 Electron Identification . . . 42
6.3 Event Selection and Lepton Identification . . . 45
6.4 Single Lepton Composition . . . 45
6.5 Dilepton Selection and Composition . . . 48
6.6 The Neural Network Classifier . . . 52
6.7 Extraction of χ . . . 59
6.7.1 Systematic Errors . . . 61
6.8 Final Result, Comparisons and Conclusions . . . 61

A The Back-propagation Algorithm **63**

B The Neural Network Simulator NNC **65**
B.1 Installation and Running . . . 65
B.2 Data Files, extension .nnd . . . 65
B.3 Parameter File, extension .par . . . 66
B.3.1 Operation Type . . . 66
B.3.2 Stop File, nn.stp . . . 66
B.3.3 Structure File, extension .str . . . 67
B.3.4 Saved Network File, extension .net . . . 69
B.3.5 Response file, extension .ntp . . . 69

C A Partially Connected Network **71**
C.1 The Toy Problem . . . 71
C.2 Learning and Performance . . . 74
C.3 Computational Cost . . . 78
C.4 Concluding Remarks . . . 78

D DBPIT **79**
D.1 Overview of Operation . . . 79
D.2 Output of DBPIT . . . 80

References **83**

List of Figures **89**

List of Tables **91**

Index **93**

Summary **95**

Samenvatting **97**

Acknowledgements **99**

Curriculum Vitae **101**

Introduction

High energy physics studies subatomic particles and their interactions in an effort to understand the basic laws of our universe. The basic forces in nature are classified into four types of increasing strength: gravitational, weak, electromagnetic and strong interactions. Our present knowledge of the latter three interactions is incorporated in a theory which is called the Standard Model.

Within the Standard Model there are three classes of fundamental particles: the leptons, the quarks and the bosons. The leptons and quarks form the basic building blocks of all matter. Their interactions are modelled by the exchange of bosons: the $W^{\pm}$ and Z^0 bosons, the photon and the gluons. These bosons are the mediators of the weak, electromagnetic and strong interactions, respectively.

Past experiments have revealed the structure of the atom with its nucleus and orbiting electrons. Present-day research has progressed to even smaller scales with the aid of large particle accelerators.

In the Large Electron Positron accelerator (LEP) at CERN, electrons and positrons are collided at a centre-of-mass energy around 91 GeV. At these energies there is an abundant resonant production of the Z^0 boson. A substantial fraction ($\approx 15\%$) of all Z^0 decay modes consists of decays into a beauty quark pair, which makes LEP an ideal environment for heavy quark physics. This thesis uses data recorded in 1991 and 1992 with the DELPHI detector, one of the four experiments at LEP.

The B^0 meson is a neutral short-lived particle that is a bound state between a beauty quark and a lighter quark. This thesis presents a measurement of the B^0-$\bar{B}^0$ mixing parameter, which is the probability that the B^0 meson transforms into its anti-particle the $\bar{B}^0$ before it decays. Within the Standard Model the rate of mixing depends on the top quark mass and on the elements V_{td} and V_{ts} of the Cabibbo-Kobayashi-Maskawa matrix.

In the analysis, Z^0 decays are selected in which both beauty quarks decay directly into a lepton. Pairs of leptons of the same charge signal the presence of B^0-$\bar{B}^0$ mixing.

Problems encountered in the diverse field of high energy physics often benefit from techniques from other disciplines. This thesis investigates the use of a neural network, an artificial intelligence technique. In conventional analyses the transverse momentum of the lepton is used to separate direct beauty quark decays from a background of other hadronic Z^0 decays. The neural network improves the separation by using additional information.

This thesis is organised as follows. Chapter 1 outlines the physics relevant to the measurement described in this thesis. After a brief introduction of the Standard Model and heavy flavour physics, the B^0-$\bar{B}^0$ mixing formalism is presented in detail.

Chapter 2 is a general introduction to neural networks. It covers basic neural network terminology and introduces the concept of learning.

Chapter 3 is dedicated to feed-forward neural networks, the type of neural network used in the analysis of this thesis. This chapter focusses on the capabilities and the practical issues operating a back-propagation neural network.

Chapter 4 introduces the DELPHI detector. It contains a short functional description of all its components and their performances. The remainder of the chapter describes the data taking and data handling in DELPHI.

Chapter 5 is devoted to the Inner Detector, the inner wire chamber of DELPHI. This chapter presents work performed by the author as a member of the Inner Detector group. The Inner Detector calibration database is described and the internal z resolution of the detector is discussed.

In chapter 6 the measurement of the B^0-$\bar{B}^0$ mixing parameter is presented. A comparison is made between a conventional p_t analysis and the neural network analysis. The results and their prospects are discussed.

Appendix A presents the derivation of the back-propagation algorithm introduced in chapter 3. Appendix B is a manual for the neural network simulator developed by the author and applied in this thesis. Appendix C investigates some properties of a partially connected back-propagation network. Appendix D describes DBPIT, the monitoring program for the off-line calibration database of the Inner Detector.

Chapter 1

Theory

This chapter outlines the physics relevant to the measurement described in this thesis.

First, the Standard Model is briefly described. It combines the Glashow-Weinberg-Salam model[1] of electroweak interactions and quantum chromodynamics[2], the theory of strong interactions.

The following section discusses heavy flavour physics. Heavy quarks are abundantly produced at LEP. An overview is given of the current precision measurements in the heavy quark sector.

The last section of this chapter introduces B^0-$\bar{B}^0$ mixing. Its mathematical formalism within the framework of the Standard Model is presented. The theory is linked to the measurement of the B^0-$\bar{B}^0$ mixing parameter χ using dileptons.

1.1 The Standard Model

The *Standard Model* describes the interactions between fundamental particles and is based on the gauge group $\mathrm{SU}_C(3)\otimes\mathrm{SU}_L(2)\otimes\mathrm{U}_Y(1)$.

$\mathrm{SU}_C(3)$ corresponds to the strong interaction of coloured quarks and gluons and is called *quantum chromodynamics* (QCD). The weak interactions, mediated by the vector bosons, are described in the $\mathrm{SU}_L(2)$ part of the Standard Model. $\mathrm{U}_Y(1)$ represents the *quantum electrodynamics* (QED) part, the interaction of the photon with the other fundamental particles.

In its present form, the Standard Model describes a world consisting of spin $\frac{1}{2}$ *fermions* that interact through exchange of *bosons* of integer spin. The following paragraphs summarise the fundamental particles and their properties.

Quarks and Leptons: the spin $\frac{1}{2}$ fermions

	Quarks		Leptons	
Generation 1	u	d	ν_e	e
Generation 2	c	s	ν_μ	μ
Generation 3	t	b	ν_τ	τ
Charge Q_f	2/3	−1/3	0	−1
Weak Isospin I_3^f	1/2	−1/2	1/2	−1/2

Table 1.1: *The fundamental left handed spin $\frac{1}{2}$ fermions.*

Table 1.1 shows that the fermions logically group together in 'generations' (or 'families'), consisting of two quark flavours and two leptons. Members of a higher generation have the same quantum numbers[1] as the corresponding member of the first generation, only their mass is higher.

The leptons and quarks are assigned to doublet (left handed) and singlet (right handed) representations of $SU_L(2)$. Table 1.1 only displays the left handed fermions. The right handed fermions, which are not sensitive to charged current weak interactions, have $I_3^f=0$. Right handed neutrino's do not exist in the Standard Model. Leptons don't take part in the strong interaction, because they don't carry a colour charge.

Force carriers: bosons with integral spin

Coupling	Boson(s)	Spin
Strong	8 massless gluons $g_1, \ldots, g_8$	1
Weak	3 massive vector bosons Z^0, W^-, W^+	1
Electromagnetic	1 massless photon γ	1
Higgs	1 massive Higgs H	0

Table 1.2: *The fundamental couplings.*

Table 1.2 summarises the situation for the bosons. $SU_C(3)$ has 8 generators (8 *gluons*), $SU_L(2)$ has 3 weak isospin generators (Z^0, W^+, W^-) and $U_Y(1)$ has the hyper-charge generator Y (the *photon*). The hyper-charge is for all particles assigned such that $Q_f = I_3^f + Y/2$.

The neutral spin-0 *Higgs boson*[4], which has not been found yet, plays a special role. One speaks of *spontaneous symmetry breaking* when the fundamental equations of a system possess a symmetry that is not displayed by the ground state.

The introduction of additional scalar fields, called Higgs fields, offer a way to generate masses for the $W^\pm$ and Z bosons through spontaneous symmetry breaking. After that, the $W^\pm$ and the Z^0 acquire mass and $SU_C(3)\otimes U_Y(1)$ is left as an exact gauge symmetry with the gluons and the photon massless.

Both the electromagnetic neutral current and the weak neutral current couple to the fermions. Therefore, the *electroweak mixing angle* θ_W was introduced for $SU_L(2)\otimes U_Y(1)$. This angle θ_W is defined by the following relation between the W mass and the Z mass:

$$\sin^2\theta_W \equiv 1 - \frac{M_W^2}{M_Z^2} \tag{1.1}$$

The couplings of the fundamental fermions to the W's, the Z^0 and the photon γ can be written in terms of the fermion charge Q_f, θ_W and the weak isospin I_3^f as follows:

- $W^\pm f\bar{f}$-coupling: $ie\gamma_\mu(1-\gamma_5)\frac{1}{2\sqrt{2}\sin\theta_W}$
- $Z^0 f\bar{f}$-coupling: $ie\gamma_\mu(v_f - a_f\gamma_5)$
- $\gamma f\bar{f}$-coupling: $-ieQ_f\gamma_\mu$

The $\gamma_\mu(\mu = 0,1,2,3)$ are the Dirac gamma-matrices and $\gamma_5 \equiv i\gamma_0\gamma_1\gamma_2\gamma_3$. The parameters a_f and v_f are called the *axial* and *vector* couplings, they are given by:

$$a_f = \frac{I_3^f}{2\sin\theta_W\cos\theta_W} \qquad v_f = \frac{I_3^f - 2Q_f\sin^2\theta_W}{2\sin\theta_W\cos\theta_W} \tag{1.2}$$

---

[1] Q_f in units of the electron charge magnitude $e = 1.60217733(49)$ x 10^{-19} C and I_3^f in units of the reduced Planck constant $\hbar \equiv h/2\pi = 1.05457266(63)$ x 10^{-34} Js, values taken from[3]

Originally, the weak interaction as it was first observed in the β-decay $n \rightarrow pe^-\bar{\nu}_e$ was thought to be point-like. All quarks and leptons have the same overall weak coupling strength given by the Fermi constant G_F. In the $\mathrm{SU}_C(3)\otimes\mathrm{U}_Y(1)$ of the the Standard Model, G_F relates to M_W and $\sin^2\theta_W$ as:

$$G_F = \frac{\hbar^2 e^2}{4\epsilon_0\sqrt{2}M_W^2 c^2 \sin^2\theta_W} \tag{1.3}$$

were c is the speed of light in vacuum and ϵ_0 the permittivity of free space[2].

The weak interactions, contrary to the strong and electromagnetic interactions, violate flavour conservation in the quark sector. One has to assume that the weak interaction quark eigenstates are mixtures of the actual flavour (mass) eigenstates. By definition the up-type quarks (u,c,t) are chosen to be unmixed and the mixing in the others is described by the *Cabibbo-Kobayashi-Maskawa* (CKM[5]) matrix V operating on the mass eigenstates of the down-type quarks (d,s,b):

$$\begin{pmatrix} d' \\ s' \\ b' \end{pmatrix} = V \begin{pmatrix} d \\ s \\ b \end{pmatrix} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} \begin{pmatrix} d \\ s \\ b \end{pmatrix} \tag{1.4}$$

The elements of this 3x3 unitary matrix define the strengths of the couplings between up-type and down-type quarks for flavour changing charged currents. Flavour changing neutral currents are not present and in the lepton sector flavour mixing would only occur if the neutrino's are not massless.

In order to account for all observed types of interactions, one also needs a massless spin-2 *graviton*, the mediator of gravity. Up till now, gravity has not been incorporated successfully in the Standard Model and has to be treated as a separate case using Einstein's theory of general relativity.

Other aspects of the standard model are also unsatisfactory. Many parameters are undefined without any explanation about their values: the quark and lepton masses, $\sin^2\theta_W$, the CKM matrix elements, the Higgs mass and the strengths of the gauge couplings themselves.

1.2 Heavy Flavour Physics

Heavy flavour physics[6,7] comprises all studies of properties of the c(harm) and the b(eauty) quarks. LEP is well suited for these studies because both $c\bar{c}$ and $b\bar{b}$ are abundantly produced around the Z^0 peak ($\Gamma_{c\bar{c}}/\Gamma_{hadrons} \approx 18\%$ and $\Gamma_{b\bar{b}}/\Gamma_{hadrons} \approx 22\%$). Various analyses are in progress that determine one or more of the free parameters in the Standard Model more and more precisely. The most important are:

- Cross section determinations.

  The cross sections are sensitive to a_f, v_f, G_F and M_Z. $\Gamma(Z \rightarrow b\bar{b})/\Gamma(Z \rightarrow hadrons)$ is via electroweak vertex corrections in m_t^2/m_W^2 dependent on the top mass.

- Higgs search.

  Since the Higgs boson, if produced at LEP, mainly decays in b quark pairs (the Higgs couples to fermions with a strength proportional to their mass), an efficient b tagging facilitates the Higgs search.

- Lifetime measurements.

  Measurements of the lifetimes of D and B mesons give values for their corresponding elements in the CKM matrix.

- Forward-backward asymmetries.

  The forward-backward charge asymmetry is proportional to the product of vector and axial couplings, so it determines $\sin^2\theta_W$.

---

[2] $c = 299792458\ ms^{-1}$ (defined) and $\epsilon_0\mu_0 = 1/c^2$ with $\mu_0 = 4\pi$ x $10^{-7}\ NA^{-2}$, values taken from[3]

- Mixing studies of K^0-$\bar{K}^0$ and B^0-$\bar{B}^0$.

  Precise measurements of the mixing parameter put constraints on the CKM matrix elements V_{td}, V_{ts} and on m_t, the mass of the top quark. Mixing takes place in both the K^0-$\bar{K}^0$ system and the B^0-$\bar{B}^0$ system, the latter will be the topic of the next subsection.

The basis of all these analyses is the actual ***heavy quark tagging***. Because the c and the b are much heavier than the other quarks, they can be identified through the decay properties of their (heavy) mesons and baryons. Some methods to tag heavy quarks are for example:

- High p_t lepton tagging.

  Leptons from B decay have a significantly higher transverse momentum p_t relative to the hadron direction than do leptons from other sources.

- Using dependence on shape variables.

  For example, a B enriched sample can be created using the directed sphericity $\bar{S}_Q$ of a set of Q tracks in a jet:

  $$\bar{S}_Q = \frac{\sum_Q p_t^2}{\sum_Q |p|^2} \tag{1.5}$$

  $\bar{S}_Q$ follows a different distribution for B decays than for light quark decays, because the B decays more isotropically than the longitudinal fragmentation of hadrons in light quark jets.

- Secondary vertex reconstruction.

  Events with a large distance between the primary and secondary vertex contain a heavy meson because they live longer.

- Through decay of the J/Ψ.

  By identifying a B meson decay into a charmonium state through observation of a J/ψ into a lepton anti-lepton pair.

1.2.1 B^0-$\bar{B}^0$ Mixing

Neutral B mesons, originating from the fragmentation of b quarks at LEP, are either a $\bar{b}d$ (B_d^0) or a $\bar{b}s$ (B_s^0) bound state. Before these mesons decay oscillations may occur: a B^0 transforms into its anti-particle $\bar{B}^0$ and vice versa, see figure 1.1 for B_d^0-$\bar{B}_d^0$. This phenomenon is called B^0-$\bar{B}^0$ *mixing*: the weak interaction connects the $|B^0\rangle$ and the $|\bar{B}^0\rangle$ state and the CP eigenstates $|B_{1,2}\rangle$ of the complete Hamiltonian are linear combinations of the flavour eigenstates, i.e.

$$|B_{1,2}\rangle = \frac{|B^0\rangle \pm |\bar{B}^0\rangle}{\sqrt{2}} \tag{1.6}$$

B_1 and B_2 have a mass difference ΔM, which introduces a time-dependent phase difference between their wave functions and thereby an oscillation between the B^0 and $\bar{B}^0$ components with a period proportional to $1/\Delta M$. This is in complete analogy to the well-known oscillations in the neutral kaon system, which were observed for the first time in the 1950's[8].

B^0-$\bar{B}^0$ mixing has been first observed experimentally in 1987, by the UA1[9] collaboration in proton-antiproton collisions and by the ARGUS[10] and CLEO[11] collaborations in e^+e^- scattering at the energy of the $\Upsilon(4s)$.

In 1964[12] it was observed that the K_L decays a small fraction of the time into $\pi^+\pi^-$. This was the first evidence for *CP violation*: the weak interaction is not invariant under a simultaneous Charge conjugation and Parity (*CP*) operation. Consequently, the two neutral kaon mass eigenstates K_L and K_S are not precisely CP eigenstates. The same is true for the eigenstates $|B_1\rangle$ and $|B_2\rangle$.

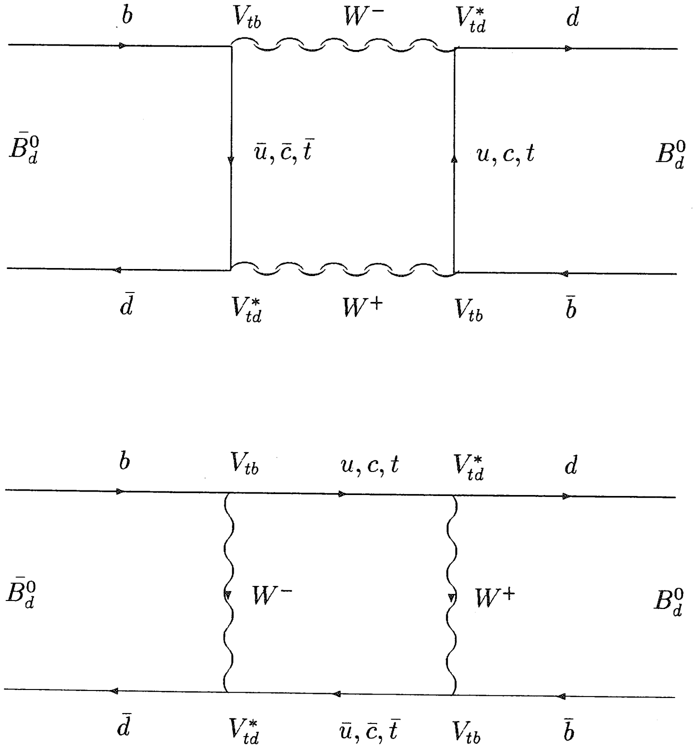


Figure 1.1: *The Feynman box diagrams responsible for B^0_d-$\bar{B}^0_d$ mixing. The dominant contribution comes from the top quark exchange and therefore only the CKM elements of the top quark have been indicated at the vertices.*

In the following mixing in the B^0-$\bar{B}^0$ system, with the presence of CP violation, will be described in more detail.

In general, a stable free particle of mass M at rest has a wave function $\Psi(t)$ which is proportional to e^{-iMt}. This becomes $\Psi(t) \propto e^{-i(M-i\Gamma/2)t}$ for an unstable particle with a 'width' Γ and a lifetime $\tau = 1/\Gamma$.

For coupled systems, like B^0-$\bar{B}^0$, M and Γ become 2x2 complex Hermitian matrices with positive eigenvalues. The Hamiltonian $\mathcal{H}$ of this coupled system takes the form:

$$\mathcal{H}\begin{pmatrix} B^0 \\ \bar{B}^0 \end{pmatrix} = \begin{pmatrix} M - \frac{i}{2}\Gamma & M_{12} - \frac{i}{2}\Gamma_{12} \\ M^*_{12} - \frac{i}{2}\Gamma^*_{12} & M - \frac{i}{2}\Gamma \end{pmatrix}\begin{pmatrix} B^0 \\ \bar{B}^0 \end{pmatrix} \tag{1.7}$$

The universal invariance of a simultaneous CP and Time reversal (*CPT*) operation requires the diagonal elements of $\mathcal{H}$ to be equal, thus B^0 and $\bar{B}^0$ have the same mass M and width Γ. The off-diagonal elements represent the B^0-$\bar{B}^0$ mixing and they are only equal if CP violation is neglected. After diagonalising $\mathcal{H}$, its eigenstates can be written in the form:

$$|B_{1,2}\rangle = \frac{(1+\epsilon)|B^0\rangle \pm (1-\epsilon)|\bar{B}^0\rangle}{\sqrt{2(1+|\epsilon|^2)}} \tag{1.8}$$

with

$$\eta \equiv \frac{1-\epsilon}{1+\epsilon} = \sqrt{\frac{M_{12}^* - \frac{i}{2}\Gamma_{12}^*}{M_{12} - \frac{i}{2}\Gamma_{12}}} \tag{1.9}$$

The parameter ϵ quantifies the CP violation. If $\epsilon = 0$ in equation (1.8) then $|B_1\rangle$ and $|B_2\rangle$ would be exactly CP eigenstates like in equation 1.6.

The eigenvalues of $\mathcal{H}$ corresponding to the states $|B_{1,2}\rangle$ are:

$$H_{1,2} = M_{1,2} - \frac{i}{2}\Gamma_{1,2} = M - \frac{i}{2}\Gamma \pm Q \tag{1.10}$$

with

$$Q = \sqrt{(M_{12}^* - \frac{i}{2}\Gamma_{12}^*)(M_{12} - \frac{i}{2}\Gamma_{12})} \tag{1.11}$$

and their mass difference ΔM and width difference $\Delta\Gamma$ is:

$$\Delta M = M_1 - M_2 = 2 \text{ Re } Q \tag{1.12}$$

$$\Delta\Gamma = \Gamma_1 - \Gamma_2 = -4 \text{ Im } Q \tag{1.13}$$

B^0-$\bar{B}^0$ oscillations are caused by the different time evolution of the mass eigenstates $|B_1\rangle$ and $|B_2\rangle$. Starting at $t = 0$ from a pure B^0 state, gives at a time t:

$$|B(t)\rangle = \frac{\sqrt{1+|\epsilon|^2}}{\sqrt{2}(1+\epsilon)}(e^{-i(M_1 - \frac{i}{2}\Gamma_1)t}|B_1\rangle + e^{-i(M_2 - \frac{i}{2}\Gamma_2)t}|B_2\rangle) \tag{1.14}$$

$|B(t)\rangle$ can be written as a superposition of $|B^0\rangle$ and $|\bar{B}^0\rangle$ using equation (1.8). The coefficients are the transition amplitudes $A(B^0 \to B^0)$ and $A(B^0 \to \bar{B}^0)$; they are:

$$A(B^0 \to B^0) = \frac{1}{2}(e^{-iM_1t}e^{-\frac{1}{2}\Gamma_1 t} + e^{-iM_2t}e^{-\frac{1}{2}\Gamma_2 t}) \tag{1.15}$$

$$A(B^0 \to \bar{B}^0) = \left(\frac{1-\epsilon}{1+\epsilon}\right)\frac{1}{2}(e^{-iM_1t}e^{-\frac{1}{2}\Gamma_1 t} - e^{-iM_2t}e^{-\frac{1}{2}\Gamma_2 t}) \tag{1.16}$$

The ratio r of the total (amplitudes squared and integrated over time) probabilities is:

$$r = \frac{P(B^0 \to \bar{B}^0)}{P(B^0 \to B^0)} = \frac{\int_0^\infty |A(B^0 \to \bar{B}^0)|^2 dt}{\int_0^\infty |A(B^0 \to B^0)|^2 dt} \tag{1.17}$$

Using equations (1.15) and (1.16) and doing the integrals gives:

$$r = \left|\frac{1-\epsilon}{1+\epsilon}\right|^2 \frac{x^2+y^2}{2+x^2-y^2} \tag{1.18}$$

where $x = \frac{\Delta M}{\Gamma} = \frac{M_1 - M_2}{\frac{1}{2}(\Gamma_1+\Gamma_2)}$ and $y = \frac{\Delta\Gamma}{2\Gamma} = \frac{\Gamma_1-\Gamma_2}{\Gamma_1+\Gamma_2}$.
Similarly, starting from $t = 0$ from a pure $|\bar{B}^0\rangle$ state, one would have obtained:

$$\bar{r} = \frac{P(\bar{B}^0 \to B^0)}{P(\bar{B}^0 \to \bar{B}^0)} = \left|\frac{1+\epsilon}{1-\epsilon}\right|^2 \frac{x^2+y^2}{2+x^2-y^2} \tag{1.19}$$

Therefore, the asymmetry

$$A = \frac{r - \bar{r}}{r + \bar{r}} = \frac{|\eta|^2 - |\eta|^{-2}}{|\eta|^2 + |\eta|^{-2}} \simeq -4 \text{ Re } \epsilon \tag{1.20}$$

is a measure of CP violation. The last step of (1.20) follows using:

$$|\eta|^2 = \left|\frac{1-\epsilon}{1+\epsilon}\right|^2 = 1 - \frac{4 \text{ Re } \epsilon}{|1+\epsilon|^2} \tag{1.21}$$

It will now be shown that CP violation effects are small. From analysis of the box diagrams of figure 1.1 in the limit when all quarks are massless except for b and t, follows[13,14]:

$$M_{12} \propto m_t^2 (V_{tb} V_{td}^*)^2 \tag{1.22}$$

$$\Gamma_{12} \propto m_b^2 (V_{ub} V_{ud}^* + V_{cb} V_{cd}^*)^2 \overset{\text{(V unitary)}}{=} m_b^2 (V_{tb} V_{td}^*)^2 \tag{1.23}$$

The box calculations have two consequences. First of all, Γ_{12} and M_{12} have almost the same phase, because both are determined by $V_{tb} V_{td}^*$. In the limit that they have the same phase, ϵ is purely imaginary as seen from (1.9), thus Re ϵ is small and $|\eta|^2 \simeq 1$. In this limit equations (1.11), (1.12) and (1.13) become:

$$Q \simeq |M_{12}| - \frac{i}{2} |\Gamma_{12}| \tag{1.24}$$

$$\Delta M \simeq 2|M_{12}| \tag{1.25}$$

$$\Delta\Gamma \simeq 2|\Gamma_{12}| \tag{1.26}$$

Furthermore, $\Gamma_{12}/M_{12} \simeq \mathcal{O}(m_b^2/m_t^2)$. Because $m_b^2 \ll m_t^2$, $|\Gamma_{12}| \ll |M_{12}|$ and $\Delta\Gamma \ll \Delta M$, so y is very small compared to x. With the two consequences $y \ll x$ and $|\eta|^2 \simeq 1$ of the box calculations, it follows that a good approximation for B mesons is:

$$r = \frac{x^2}{2 + x^2} \tag{1.27}$$

Experimentally, the *flip probability* is often used to represent B^0-$\bar{B}^0$ mixing:

$$\chi \equiv \frac{P(B^0 \to \bar{B}^0)}{P(B^0 \to B^0) + P(B^0 \to \bar{B}^0)} = \frac{r}{1+r} = \frac{x^2}{2(1+x^2)} \tag{1.28}$$

This is the B^0-$\bar{B}^0$ *mixing parameter* χ that is measured in this thesis.

χ ($0 \leq \chi \leq \frac{1}{2}$) is determined by observing a convenient final state in a sample where a B^0-$\bar{B}^0$ pair is produced. The analysis of this thesis uses the *dilepton* state, in which there are two leptons, each coming from a different B meson. An excess of like sign dilepton events is an indication for mixing, because with no mixing b–semi-leptonic decays always result in unlike sign dilepton events ($\bar{B}^0 \to l^-$ and $B^0 \to l^+$, where the lepton l is either an electron, a muon or a tau).

χ is calculated by determining the ratio R, which is defined as the number of like sign dileptons $N(l^\pm l^\pm)$ over the total number of dileptons, $N(l^\pm l^\mp) + N(l^\pm l^\pm)$.

CP violation effects are ignored ($\epsilon = 0$), because as shown before, they are small. If CP violation effects could not have been neglected, (1.20) is an asymmetry between the like-sign pairs l^+l^+ and l^-l^-, namely

$$A = \frac{r - \bar{r}}{r + \bar{r}} = \frac{N(l^+l^+) - N(l^-l^-)}{N(l^+l^+) + N(l^-l^-)} \tag{1.29}$$

Ignoring CP and explicitly writing all possible sign combinations, it follows from (1.28) that χ, r and R are related according to:

$$R = \frac{N(l^+l^+) + N(l^-l^-)}{N(l^\pm l^\mp) + N(l^\pm l^\pm)} = 2\chi(1-\chi) = \frac{2r}{(1+r)^2} \tag{1.30}$$

When R has been measured, (1.30) therefore gives:

$$\chi = \frac{1-\sqrt{1-2R}}{2} \tag{1.31}$$

Chapter 2

Neural Networks

Until the late 1980's essentially all information processing applications used a single basic approach: programmed computing. This involves devising an algorithmic procedure for solving the problem and coding it correctly in software.

Sometimes however, a precise algorithm is not known or is very time consuming; then a new approach to information processing that does not require rule development is the solution. Such an approach is offered by neural networks. They are parallel, distributed and adaptive systems that develop information processing capabilities after exposure to an environment.

The general function of a neural network is often similar to that of a subroutine (input → processing → output) so it can be embedded inside a programmed computing environment. However, neural networks are not meant to be immediately used as 'black boxes'; first a sufficient level of understanding should be reached as well as care taken during the development phase.

This chapter is a concise introduction to neural networks and serves as background information to the more practical treatment of the next chapter, which focusses on the type of the neural network used in this thesis. Those already familiar with 'neural terminology' and learning concepts might skip this chapter.

2.1 Neural Network Concepts

A *neural network* is a parallel information processing structure consisting of a set of nodes with a set of directed line segments between them. Figure 2.1 shows the organisation of a generic *node*, also referred to as a *processing element*, *unit* or (formal) *neuron*.
In general, a node has the following properties:

- The directed line segments are *connections*. Each connection is an instantaneous uni-directional signal conduction path. The strength of a connection is represented by a number, often called the *weight* of that specific connection.

- Each node receives a number of signals $x_1, \cdots, x_N$ via its incoming connections. A node has a *transfer function* that uses only its *local memory* and incoming signals to compute its output signal y. During this process its local memory may be updated also.

- Nodes are connected to other nodes and/or to the outside world. If they are receiving input from outside the network they are called *input nodes* and if they output their signal to the outside world they are called *output nodes*. The other nodes are called *hidden nodes* since they function entirely within the network.

Nodes can be connected in any way to form a neural network with a specific *architecture*. Any neural network can be divided in a number of disjoint *layers* in which the nodes have the same transfer function. Generally, one layer is reserved for the input nodes and another one for the output nodes.

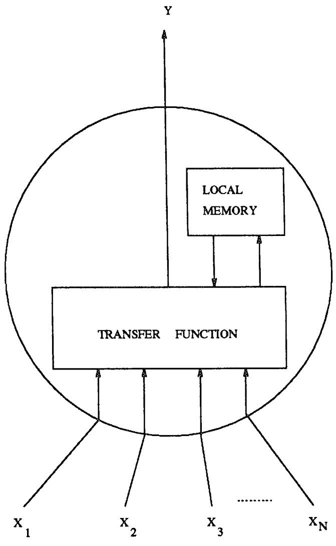


Figure 2.1: *A generic node.*

Information enters the network via the input nodes and leaves it via the output nodes. Nodes are numbered according to a convenient scheme, such as left to right, bottom layer to top layer.

Figure 2.2 displays a small example neural network architecture, consisting of three layers. Arrows indicate the direction in which the connections operate and the connection strength between a node with index j and a node with index i is designated as w_{ij}. When describing the information flow in a neural network, a target based approach is used. This means that each node uses the connections that impinge on it as 'come from' links rather than that they are treated as 'goto' links by their sending nodes.

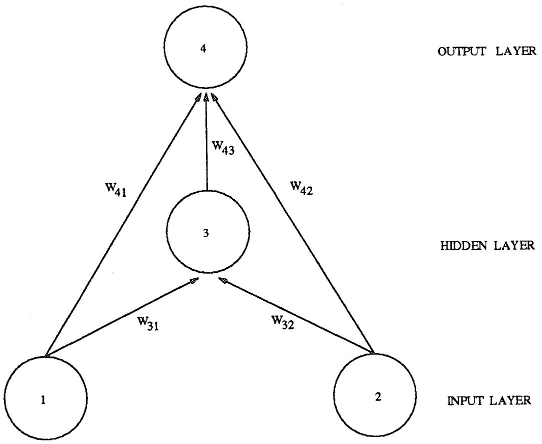


Figure 2.2: *A simple neural network architecture with three layers.*

2.2 Learning

Learning is modifying a system in such a way that it better achieves a stated goal. For a neural network that means that the weights between the nodes should be modified. Although there are many different ways to implement learning behaviour, only three learning strategies will be discussed here. They (or a mixture of them) are widely used in different neural network architectures.
The main learning strategies are:

- *Coincidence Learning.*

  Weight changes take place in response to events within a node that occur simultaneously, for example an incoming line and its output are both high.

- *Performance Learning.*

  Learning laws in this category try to find a set of weights that minimise or maximise a performance (cost) function defined for the network.

- *Competitive Learning.*

  A competitive process involving some or all nodes selects nodes which are 'winners' and are thus allowed to modify their weights (or modify them in a different way from the 'losers').

Apart from a learning strategy the adaptation of the network always takes place according to a *training regime*, which is the way in which the environment is presented to the network during the learning process. There are three different training regimes:

- *Supervised Training.*

  Supervised training is the most common training regime. Repeatedly a large set of input-output pairs is presented to the network, which consists of example inputs together with their desired or *target* network response. This set is called the *training set.* During the training the network forms a reduced internal representation that is able to map the input on the output space.

- *Graded Training.*

  Although similar to supervised training, the network only receives a score or grade that tells it how well it has done over a sequence of multiple inputs. This is useful if the correct output is not known for every input.

- *Self Organisation.*

  A totally different philosophy in which the network is required to extract the features of the probability distribution of the data presented to the network.

The above mentioned learning strategies will each be described in more detail in separate paragraphs below in the context of their usual training regime.

2.2.1 Coincidence Learning

Coincidence learning has been introduced by the Canadian psychologist Donald Hebb. In his 1949 book *The Organisation of Behaviour*[15], he poses that if a specific input to a neuron helps to let that neuron emit a pulse, the efficacy of that input will be increased to help that happen again in the future. An appealing concept because it implies behavioural reward even at the cellular level.

Hebb's learning concept is the basis for the *Hebb learning law* for neural networks, which operates in the context of an architectural element called the *linear associator*. In figure 2.3, a linear associator layer consisting of M nodes is displayed. Its input is a vector $\vec{x} = (x_1, x_2, x_3, \dots, x_N)^T$ drawn from R^N, coming from the outside world or from other nodes in the network. The transfer function of a node in the linear associator computes a weighted sum of its inputs, so the output vector $\vec{y} = (y_1, y_2, y_3, \dots, y_M)^T$ at the top is derived from $\vec{x}$ by means of the product:

$$\vec{y} = W\vec{x} \tag{2.1}$$

where $W(=w_{ij})$ is an $M\mathrm{x}N$ weight matrix with rows $\vec{w}_1^T, \vec{w}_2^T, \cdots, \vec{w}_M^T$, which are the weight vectors of the individual nodes.

The basic idea is that the linear associator should learn, under a supervised training regime, the correct input-output pairs $(\vec{x}_1, \vec{t}_1), (\vec{x}_2, \vec{t}_2), \cdots, (\vec{x}_L, \vec{t}_L)$ that form the training set. After that, input of one of the training vectors $\vec{x}_p$ should give an $\vec{y}$ equal to $\vec{t}_p$, but moreover the input of any vector close to $\vec{x}_p$ should give an output vector close to $\vec{t}_p$. This is achieved when the weights are updated with the following form of the Hebb learning law:

$$w_{ij}^{new} = w_{ij}^{old} + t_{pi}x_{pj} \tag{2.2}$$

x_{pj} and t_{pi} are respectively the j^{th} and i^{th} component of the vectors $\vec{x}_p$ and $\vec{t}_p$. The w_{ij}'s play the role of the efficacy of the input to neuron i coming from neuron j. All w_{ij}'s start off at zero, therefore the efficacy w_{ij} is only altered if both x_{pj} and t_{pi} are not equal to zero.

The Hebb learning law can of course be implemented in parallel for all nodes in the whole linear associator layer, therefore in terms of the whole weight matrix:

$$W^{new} = W^{old} + \vec{t}_p\vec{x}_p^T \tag{2.3}$$

As the pairs $(\vec{x}_p, \vec{t}_p)$ are entered, W is built up by their outer product sum. The linear associator can store at most N pairs of vectors if one requires it to map each $\vec{x}_p$ exactly on its target $\vec{t}_p$. Note however that this full capacity is only reached if the vectors that have to be stored are orthogonal. If they are not, one makes an error attempting to reconstruct $\vec{t}_p$ using the product of formula 2.1.

If the vectors are not orthogonal, a mean-square error measure can be introduced, which leads to a general method to find the best W. This is the Widrow-Hoff learning law, which will be discussed in the next paragraph about performance learning.

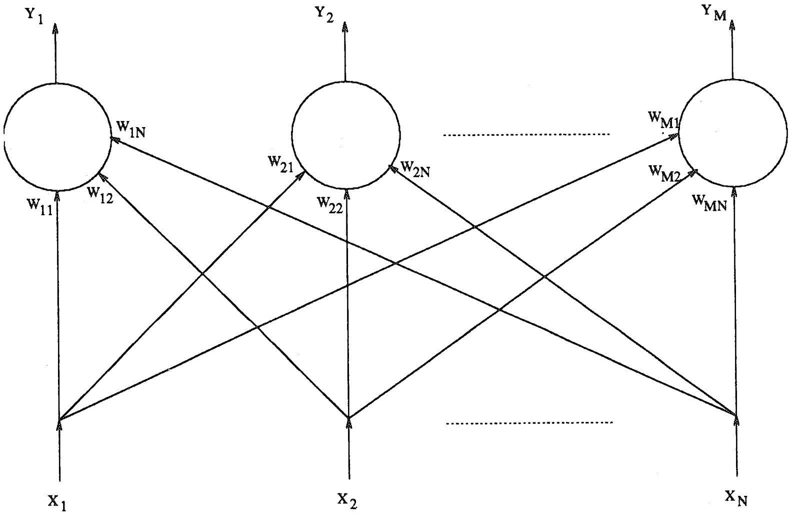


Figure 2.3: *The linear associator.*

2.2.2 Performance Learning

All forms of performance learning use an error measure that indicates how well the network performs during its training. A very common goal is the *mean-square error measure* criterion, which quantifies the difference between current behaviour of (nodes in) the network and their desired behaviour. In *Widrow-Hoff learning*[16], the performance is for each node i defined as the mean-square error measure $F(\vec{w}_i)$ for a large number (L) of training pairs:

$$F(\vec{w}_i) \equiv \lim_{L\to\infty} \frac{1}{L}\sum_{p=1}^{L}(y_{pi} - t_{pi})^2 \tag{2.4}$$

where $y_{pi} = \vec{w}_i \cdot \vec{x}_p$ is the response of the node when fed with the training vector $\vec{x}_p$, and t_{pi} its desired output.

The gradient $-\nabla_{\vec{w}_i}F(\vec{w}_i)$ is the direction in which $F(\vec{w}_i)$ decreases at the fastest possible rate. Using definition (2.4) and $\delta_{pi} \equiv (y_{pi} - t_{pi})$, this gradient is:

$$\begin{aligned} -\nabla_{\vec{w}_i}F(\vec{w}_i) &= -\lim_{L\to\infty} \frac{1}{L}\sum_{p=1}^{L} 2(y_{pi} - t_{pi})\nabla_{\vec{w}_i}y_{pi} \\ &= -\lim_{L\to\infty} \frac{1}{L}\sum_{p=1}^{L} 2\delta_{pi}\vec{x}_p = -2 < \delta_{pi}\vec{x}_p > \end{aligned} \tag{2.5}$$

To find the direction in which $F(\vec{w}_i)$ decreases, a large number of $\delta_{pi}\vec{x}_p$ vectors has to be averaged. The idea of Widrow and Hoff was to use each $\delta_{pi}\vec{x}_p$ as a correction to $\vec{w}_i$ to minimise $F(\vec{w}_i)$; they proposed:

$$\vec{w}_i^{new} = \vec{w}_i^{old} - \eta\delta_{pi}\vec{x}_p \tag{2.6}$$

where η is a small positive constant, which has to be tuned by hand and is called the step-size or *learning rate*. Typically one uses $0.01 \leq \eta \leq 10.0$, but every different problem requires another η that has to be varied during the training process. If η is too small, convergence of the weight vector will take too long and if η is too big, there will be no convergence and the vector keeps oscillating around the minimum.

Formula (2.6), the Widrow-Hoff learning rule, is also known as the least-mean-square (LMS) learning rule or the *delta-rule* for linear associator networks. In order to fully explore the capabilities of the linear associator, one often adds a *bias input* to the network. This means that the input vectors $\vec{x}$ each artificially get an extra coordinate equal to one. If these augmented vectors are then subjected to the linear transformation $W\vec{x}$ then the associator layer can implement all *affine transformations*: arbitrary rotation, scaling and translation operations.

It is not useful to add a second linear associator layer, since a sequence of two affine transformations can always be implemented by a single other affine transformation. Adding more layers is only useful if the nodes have a non-linear transfer function. This accesses a whole new range of powerful networks, which can also implement non-linear mappings. In the next chapter about feed-forward networks their capabilities will be described in detail.

2.2.3 Competitive Learning

Competitive learning laws are inspired by nature's principle of 'survival of the fittest': only winning nodes are allowed to learn. In this section a specific type of competition learning known as Kohonen learning will be explained. It is a good example of the use of the self organisation training regime. Self organisation implies that it is not needed to supply the network with the correct responses $\vec{t}_p$. The weights in the network will represent the (sub)structures in the input space, thus the training set only consists of data vectors $\vec{x}_p$.

The *Kohonen learning* law[17] operates in a substrate known as a Kohonen layer. Like the linear associator of figure 2.3, it consists of M nodes each receiving an input vector of length N.

However, it functions in a different way since each node calculates the quantity:

$$I_i = D(\vec{x}_p, \vec{w}_i)$$

where D is any distance measuring function, for example the Euclidean distance $D(\vec{x}_p, \vec{w}_i) = ||\vec{x}_p - \vec{w}_i||$. After that, a competition takes place between the nodes to find out which node has its weight vector $\vec{w}_i$ the closest to the currently fed $\vec{x}_p$.

The idea is that the weights of the Kohonen layer should form an accurate and compact representation for the probability density function of the data. This is very useful, because all information that can be known about the data is contained in its probability density function $\rho(\vec{x_1}, \vec{x_2}, \ldots, \vec{x_p})$. All the $\vec{w}_i$ vectors should arrange themselves so that they are approximately equiprobable, in a nearest neighbour sense, with respect to the data vectors $\vec{x}_p$.

Moving a weight vector $\vec{w}_i$ of the winning node towards a data vector $\vec{x}_p$ can be achieved by the basic Kohonen learning law

$$\vec{w}_i^{new} = \vec{w}_i^{old} + \eta(\vec{x}_p - \vec{w}_i^{old}) \tag{2.7}$$

In order to make this work in practice, two major problems have to be overcome.

First, if ρ contains disconnected regions in which the weight vectors can get stuck, there will be under-represented regions of ρ. Two common methods exist that solve this problem. In the first one, called ***radial sprouting***, all weight vectors start off as the zero vector and the inputs $\vec{x}_p$ are multiplied with a small positive number β. During learning, β is slowly increased to 1, so the weight vectors are slowly peeled off from the origin, as they follow the $\vec{x}_p$'s. The second solution to solve the problem of under-represented regions is to add uniformly distributed noise vectors to the data. First the amplitude of the noise vectors is very high and as training proceeds their level gets lowered.

Apart from the problem of under-represented regions, the Kohonen learning law also does not produce an equiprobable positioning of the $\vec{w}_i$ vectors: their density tends to be too low where ρ is small and too high where ρ is large. The solution is to build a 'conscience' into each node, which keeps track of the fraction of time that that node wins the competition. This is called the *conscience mechanism*[18]: nodes with a frequency of winning the distance competition below (or above) the desired equiprobable level $1/M$ are favoured (or disfavoured) to win the weight modification competition.

Chapter 3

Back-propagation in Feed-forward Neural Networks

Back-propagation, the learning strategy for an important class of neural networks, was first introduced in 1974 by Paul Werbos[19]. For more than ten years the algorithm remained obscure and unappreciated. It was in 1985 due to David Rumelhart and the other members of the PDP group[20] that back-propagation became available to a large audience in usable form. After that, the technique was there to stay and many applications quickly emerged.

In this chapter, first the architecture and operation of the standard feed-forward back-propagation network will be described. Next, the capabilities of these networks will be shown and practical issues will be addressed that are important in their design and training stages. Finally, examples of applications in high energy physics will be given.

3.1 Architecture and Operation

The architecture of a standard *feed-forward neural network* comprises two or more layers with only connections between subsequent layers. In order to simplify the discussion the example network of figure 3.1 that has only three layers will be used here. The nodes in its input layer are labelled k, the nodes in the hidden layer are labelled j and the output nodes are labelled i.

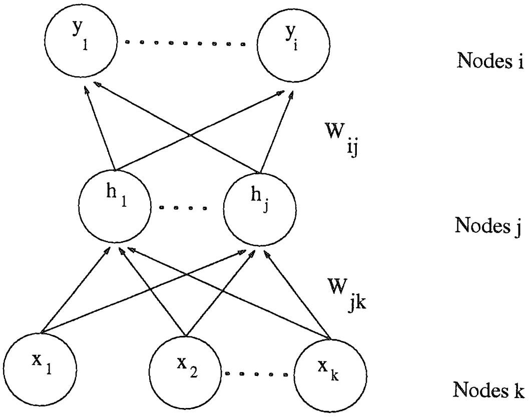


Figure 3.1: *A feed-forward neural network with three layers.*

In a feed-forward neural network, information flows from the bottom to the top. The input to the network (see figure 3.1) are the values x_k, which directly are the outputs of the nodes in the first layer. Each node in the next layer uses the outputs of the previous layer and the weights between the two layers to compute its output.

Like in the linear associator, introduced in the previous chapter, each node calculates the weighted sum of its inputs. This value is called the *activity* a of the node. In addition, the output of each node is bounded by a function g.

In figure 3.1 the nodes in the hidden layer compute:

$$\begin{aligned} a_j &= \textstyle\sum_k w_{jk} x_k = \vec{w_j} \cdot \vec{x} \\ h_j &= g(a_j) \end{aligned} \tag{3.1}$$

After that, the nodes in the output layer similarly compute their activities and outputs using the outputs of the previous layer and the weights w_{ij}:

$$\begin{aligned} a_i &= \textstyle\sum_j w_{ij} h_j = \vec{w_i} \cdot \vec{h} \\ y_i &= g(a_i) \end{aligned} \tag{3.2}$$

g is called an *activation* or *threshold* function; its purpose is to limit the response of a node to a certain range of values.

In the simplest case each node behaves like a ***binary threshold neuron*** and g is a step function: if the activity is above some threshold θ the output of the node becomes 1 otherwise 0. Unfortunately, it is not possible to find a learning algorithm for a network with binary threshold neurons that has more than two layers. An intuitive learning rule does exist if there are only two layers. A two layer linear associator with binary threshold neurons is called a *perceptron*[21]; however it has a limited usefulness and will not be considered any further here.

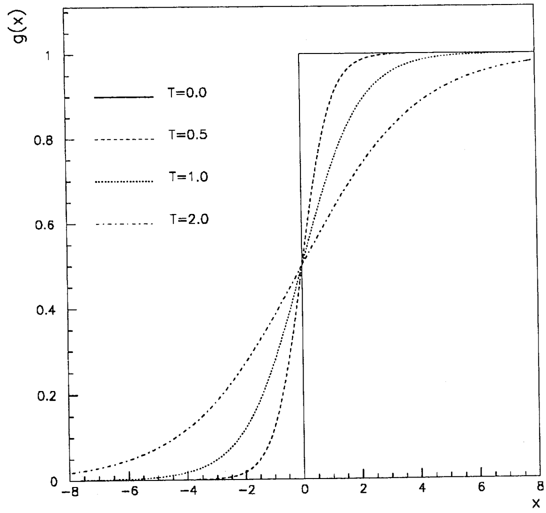


Figure 3.2: *The sigmoid function* $g(x) = 1/(1 + e^{-x/T})$ *for* $T = 0, T = 0.5, T = 1.0$ *and* $T = 2.0$.

In order to devise a learning rule a continuous non-decreasing differentiable function replacing the step function is necessary. A common choice is the *sigmoid* function:

$$g(a) = \frac{1}{1+e^{-(a-\theta)/T}} = \frac{1}{2} + \frac{1}{2}\tanh((a-\theta)/T) \tag{3.3}$$

The output of $g(a)$ is a real number between 0.0 and 1.0. The sigmoid function acts as a soft threshold function: the network is given an effective '*temperature*' T that is the inverse slope of the sigmoid (see figure 3.2).

After introducing $g(a)$, the gradient descent procedure introduced in the previous chapter can be generalised to feed-forward architectures with one or more hidden layers. The resulting recipe that minimises the mean-square-error over the training set is called the *generalised delta-rule*[20] or the *back-propagation algorithm*, a derivation of which is found in appendix A.

Back-propagation consists of two passes. In the first pass an input vector $\vec{x}$ proceeds through the network and generates a certain output vector $\vec{y}$. Next, in the backward pass, the difference between $\vec{y}$ and desired output $\vec{t}$ generates an error signal that is propagated back through the network to teach it to come closer to the desired output. This is achieved by applying the following steps:

```
1. Initialise all the weights with small random positive or negative values: $w_{ij}^{ini}$ and $w_{jk}^{ini}$.
2. REPEAT 3-6
3.     Take the next training pair $(\vec{x},\vec{t})$.
4.     Calculate $\vec{h}$ and the network output $\vec{y}$ according to formulae (3.1) and (3.2).
5.     Calculate for each node in the output layer the error signal $\delta_i = y_i - t_i$.
6.     Update the weights with learning rate $\eta$.
       Output nodes: $w_{ij}^{new} = w_{ij}^{old} - \eta\delta_i h_j$
       Hidden nodes: $w_{jk}^{new} = w_{jk}^{old} - \eta\sum_i w_{ij}\delta_i g'(a_j)x_k$
7. UNTIL the $w_{ij}$ have converged or the desired performance level is reached.
```

The back-propagation algorithm minimises the mean square error:

$$E(\mathbf{w}) = \frac{1}{2}\sum_p\sum_i (y_{pi} - t_{pi})^2 \tag{3.4}$$

where p runs over all training pairs $(\vec{x}_p,\vec{t}_p)$. At any time, $E(\mathbf{w})$ is a function of all the weights $\mathbf{w}$ of the network. The values $E(\mathbf{w})$ constitute the *error surface* of the network.

Many variations exist on the basic back-propagation algorithm. A very common extension is the addition of a *momentum term* to the weight update equations. The idea is to provide each weight with some inertia with strength α to avoid oscillations around the minimum:

$$\Delta w_{ij} = -\eta\delta_i h_j + \alpha\Delta^{old}w_{ij} \tag{3.5}$$

$$\Delta w_{jk} = -\eta\sum_i w_{ij}\delta_i g'(a_j)x_k + \alpha\Delta^{old}w_{jk} \tag{3.6}$$

More drastic changes to achieve faster convergence usually involve the introduction of individual learning rates. Manipulation of individual learning rates enables a more efficient exploration of the error surface and therefore yields a faster convergence. Examples of these self adapting back-propagation algorithms are the SAB[22], SuperSAB[23] and delta-bar-delta[24] methods.

A different approach is the incremental neural network technique of Timur Ash[25]. This method starts with a small number of hidden nodes and adds nodes in response to changes in the error level as measured using test data.

The use of back-propagation is not limited to simple architectures with layer to next-layer connections. For example, it can be applied to architectures in which connections skip layers (see figure 2.2 in chapter 2), recurrent architectures[26] and higher order architectures[27].

3.2 Capabilities of a Feed-forward Network

Like other multivariate classifiers, a feed-forward neural network exploits the correlations among variables describing the objects to discriminate. It is clear that this is a much more powerful approach than simply projecting each of the variables on their axis, because then their correlation is lost and the individual distributions will usually overlap.

In the simplest type of classifier, the *Fisher discriminant*[28,29], one searches for another projection axis, that is a linear combination of the variables, to reduce the amount of overlap. Fisher discriminant analysis can not be outperformed if the distributions of the individual variables are Gaussian and have the same correlation matrices, i.e. if they have the same shape and just distinct centroids. If the distributions have different correlation matrices, one can introduce a more general *Gaussian classifier*, that can not be outperformed if distributions are Gaussian[29].

If the distributions are not Gaussian, the Gaussian classifier does not achieve the best possible separation. That is where back-propagation neural networks come to help: they can form a non-linear discriminating function that is able to separate also non-Gaussian distributions. In general the nodes in the hidden layer(s) act as feature detectors in the sense that each of them 'sees' two different regions in the multi-dimensional space and divides them using *hyper-planes*.

A two-dimensional example of this mechanism is shown in figure 3.3 with two distributions A and B, which are clearly not Gaussian. Dashed lines in this figure indicate the three hyper-planes (in two dimensions they are lines), that correspond to three hidden nodes together forming a curve to separate the distributions. A more detailed and numerically explicit example of this mechanism is described in appendix C.

There is mathematical justification, although not completely satisfactory up till now, that a feed-forward neural network can implement a very wide range of discriminating functions. After David Hilbert presented in the beginning of this century 23 problems for future mathematicians to solve[30], Andrei Kolmogorov and Vladimir Arnol'd started a friendly mathematical battle in which they tried to solve the 13th problem. Kolmogorov won[31] and his victory and later improvements from others[32–34] led to the theorem below, which is relevant to the capabilities of feed-forward neural networks.

Kolmogorov's Mapping Neural Network Existence Theorem[34] Given any continuous function $f : [0,1]^n \rightarrow R^m, f(\vec{x}) = \vec{y}$, f can be implemented **exactly** by a three-layer feed-forward neural network having n nodes in the first ($\vec{x}$ input) layer, $(2n+1)$ nodes in the middle layer, and m nodes in the top ($\vec{y}$ output) layer.

Kolmogorov's feed-forward network does *not* have nodes with a sigmoid activation function. The existence theorem was proved for semi-linear nodes in the hidden layer and highly non-linear nodes in the output layer.[1]

Although the theorem indicates that the quest for approximations of functions by a feed-forward network is theoretically justified, it is not constructive: it only says such a three layer network must exist, but not how to find it. Specific examples of the transfer functions for the nodes in the hidden and output layers have not been found yet.

Kolmogov's theorem was a first step, but the next theorem below shows that a *back-propagation* network is also able to implement any function of practical interest.

Back-propagation Theorem[35] Given any $\epsilon > 0$ and any function $f : [0,1]^n \rightarrow R^m$ for which each of its coordinate functions is square-integrable on the unit cube[2], there is a three-layer **back-propagation** neural network that can approximate f to within ϵ mean-square-error accuracy.[3]

---

[1] The actual requirements were: hidden nodes j calculate $h_j = \sum_{k=1}^{n} \lambda_k \Psi(x_k + j\epsilon) + j$, where Ψ is monotonically increasing, continuous and real. λ is real and ϵ a rational number with $0 < \epsilon \leq \delta$ with $\delta > 0$ an arbitrary chosen constant. Output nodes calculate $y_i = \sum_{j=1}^{2n+1} g_i(h_j)$, the g_i's are real, continuous and highly non-linear.

[2] A function $g(\vec{x})$ is said to be square-integrable on the unit cube if $\int_{[0,1]^n} |g(\vec{x})|^2 d\vec{x}$ exists.

[3] $\epsilon \equiv \int_{[0,1]^n} |f(\vec{x}) - \tilde{f}(\vec{x})|^2 d\vec{x}$ with $\tilde{f}(\vec{x})$ an approximation of $f(\vec{x})$.

Note that this theorem only says that for any $f : [0,1]^n \rightarrow R^m$, there exists a set of weights for a back-propagation network that implement it. This does not imply that the back-propagation algorithm or any other known learning algorithm is able to develop these specific weights.

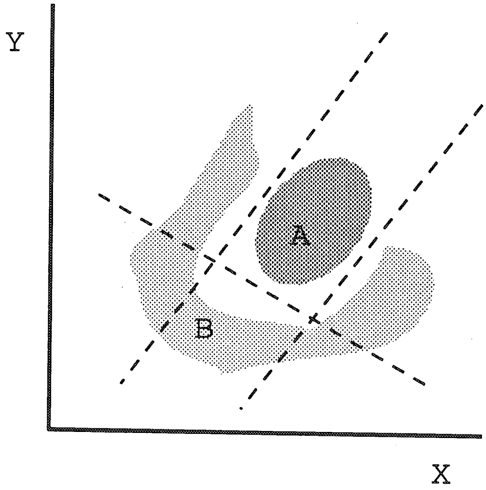


Figure 3.3: *Separation of two distributions A(x,y) and B(x,y).*

3.3 Practical Issues using a Back-propagation Network

When developing a back-propagation neural network, one has to choose an architecture and a set of initial parameters for the training. Most of these choices are a matter of experience and depend on the problem at hand. This section contains guidelines for the design and training of back-propagation networks.

3.3.1 Network Architecture

Number of layers
The back-propagation theorem indicated that three layers is always enough, and in practice that is often true. However, sometimes it is necessary to use four layers or even more. This is when too many hidden nodes would be required to solve the problem with only one hidden layer, whereas an adequate solution can be obtained with a smaller network size by using more than three layers.

Number of input nodes
The number of input nodes is equal to the number of distinctive and sensitive variables chosen to solve the problem. Often this number can be decreased significantly by combining basic variables to form a 'smart' variable, which is known to be more sensitive to the problem.

The number of output nodes

The function of the output layer is to represent the networks response on anything fed into the input layer. So, if the network is used to discriminate between between two predetermined classes of events, A and B, one output node is needed (binary coding). If the output of that node is less than 0.5, the event presented at the input layers is assigned to class A, otherwise to class B. For N classes, the number of output nodes should be at least equal to the number of bits required for the binary representation of N. In practice, one chooses the number of output nodes equal to N.

Number of hidden nodes

The hidden nodes implement separation planes in the multi-dimensional space of the input variables. Choosing more hidden nodes increases the number of paths to the global minima of the problem and decreases the chance of getting stuck in local minima during the training. On the other hand increasing the number of hidden nodes also increases the dimensionality of the weight space thereby giving rise to a lower generalisation performance (see section 3.3.3). The normal approach is to start out with many hidden nodes and decrease their number if the performance of the network is still satisfactory. For many problems the number of hidden nodes is in between half and two times the number of input nodes.

3.3.2 Setting the Training Parameters

A number of items are important for the learning behaviour of the network. For standard back-propagation learning (with the momentum extension) they are:

- Scaling of the input variables.

  The network learns more easily if the all inputs are scaled O(1). The optimal weights between the input and the (first) hidden layer will then be of the same order of magnitude as in the rest of the network. Moreover, if one of the inputs would be significantly larger than the others the network will concentrate on it too much and take a very long time to learn the others.

- The learning rate η.

  The optimal learning rate η for a node scales like 1/(its *fan-in*) or stronger[36]. The fan-in to a node is the number of nodes connecting to that node in the forward direction. In back-propagation where there is only one learning rate for all nodes in the network the initial learning rate can be safely chosen of the order of 1/(average fan-in of nodes in the network). As learning proceeds, η is usually decreased (*weight decay*) to allow for a less coarse search for (and near) the minimum.

- The strength α of the momentum term.

  In practice all values between 0 and 1 are used. Since the momentum term is supposed to damp out oscillations near the minimum, its value is increased as training proceeds.

- The temperature T.

  The higher the temperature, the less likely it is for the network to end up in a stable state. If the temperature is too low however, the network will be unable to escape local minima. In practice $T = 1.0$ is often a convenient and good choice.

- The initial weights.

  In back-propagation the weight changes $\Delta w_{jk} = w_{jk}^{new} - w_{jk}^{old}$ are proportional to $g'(a)$. The initial changes to the weights should be large, this means $g'(a)$ should be large and the initial summed inputs should be small. As a rule of thumb choose the initial weights $w_{ij}^{ini}, w_{jk}^{ini} \approx$ 0.1/(maximum fan-in).

3.3.3 Achieving Generalisation

The upper limit of generalisation performance for a classification problem is given by the *Bayes limit*[37], which is the minimum overlap of the multidimensional distributions. For simple problems with overlapping Gaussian distributions, the Bayes limit is easy to calculate and it has been demonstrated that back-propagation networks generalise to values very close to this limit[38]. For 'real' problems with higher dimensionality it is often not feasible to calculate the Bayes limit.

In practice the network *generalisation performance* depends mainly on the ratio N_w/N_p, where N_w is the total number of weights in the network and N_p the total number of training patterns. N_w is a measure for the amount of information that can be stored in the network. If N_p is too small, the training set can be completely stored in the network and the network has found a useless solution to the problem only valid for that specific training set. Feed-forward networks with one hidden layer have a generalisation error of order $\mathcal{O}(N_w/N_p)$[39]. A rule of thumb is that one should at least use 10 times more training sets than there are weights in the network. It is anyway advisable to use the biggest training set available, that is independent of the set where the network will be applied to.

To improve the generalisation performance, it also sometimes helps to add random noise to the training vectors, which smears them. Another method is to train, after initial training with the whole set, only with sets whose outputs are close to the midpoint value of the activation function. These are the difficult cases that are close to the border between classes.

Another phenomenon that has to be dealt with is *over-training*. This sometimes happens in the very last phase of the training when the performance on an independent data sample actually goes down, while the error on the training set still (slowly) decreases. The reason is probably that, because of the finite size of the training set, the network starts focusing on details in the training set in its attempt to fit the data better and better. There are two ways to attack over-training. First of all, one can stop training when the error on the training set starts decreasing very slowly, because usually a satisfying level of performance has then been reached anyway. This, of course, has to be checked afterwards on an independent data set: the *test set*. Another method is to keep track of the error on the test set during training on the training set; when the error on the former goes up, the training is stopped.

3.4 High Energy Physics Applications

High energy physics contains many interesting feature recognition problems in both off-line analysis and on-line event selection. Obvious examples are heavy quark tagging[40–43], Higgs tagging[44], mass reconstruction[45], tracking[46], and many other situations in which the separation of signal from background is essential. There are many established neural network solutions to these problems around. Many results have been achieved which are superior to conventional approaches[47–49] and when their performance was equivalent to conventional approaches their solution was found to be stable[50].

In this section an existing implementation of quark-gluon separation with a feed-forward neural network will be described in some more detail, and a second example, a possible implementation of noise suppression will be presented.

3.4.1 Quark-Gluon Separation

Tests of QCD often require the knowledge of the parton origin of a hadronic jet. For example, the distinction between quark and gluon jets provides a measurement of the gluon-gluon and quark-gluon coupling in multi-jet events.

Hadronic jets originating from quarks have on average a larger energy than those originating from gluons, which is exploited in the standard quark-jet tags. For heavy flavours the presence of a lepton from semi-leptonic decay indicates a quark jet. The neural network approach, described below, offers a tag that doesn't rely on the jet energy but exploits the fragmentation difference between quarks and gluons[49].

The data available to the network were simulated e^+e^- events at 29 GeV/c and 92 GeV/c centre-of-mass energy, produced by the three different Monte Carlo generators JETSET[51], ARIADNE[52] and HERWIG[53]. All of these were forced 3-jet events with the angles between the jets greater than 40 degrees and clustered with LUCLUS[51].

The network used 16 variables: the four-momenta $(\vec{p}, E)$ of the four leading tracks in the jet. It was trained on a single jet basis using a three layered feed-forward back-propagation network with 16 input nodes, 10 hidden nodes and 1 output node. Gluons were expected to give an output of 1.0 and quarks an output of 0.0. If the output was greater than 0.5 the jet was interpreted as a gluon jet otherwise it was interpreted as a quark jet.

The resulting network classifies gluon and quark jets $(85 \pm 1)\%$ of the time correctly. The performance of the network was found to be stable for all three Monte Carlo generators at both centre-of-mass energies.

This is a spectacular result, because selecting the jet with the smallest energy as the gluon jet typically yields 65% identification rate[54] and even the best more elaborate schemes have a 70-75% performance[55].

For this problem it was possible to calculate an estimate for the Bayes limit using the total energy and momentum of the jet and reducing the accuracy of the kinematical variables. After dividing the kinematical ranges into 100x100 bins, the integration indicated ≈88% maximum performance. The neural network prediction is just a few percent below this theoretical limit.

3.4.2 Noise Suppression

Feed-forward back-propagation networks don't always have to be provided with example inputs together with their desired outputs during training. When *auto-associative learning* is applied, the network is taught to perform the identity function: the input to the network should be reproduced as good as possible in the output layer.

Typically the number of hidden nodes is less than the number of output nodes. The network encodes general properties of the input data as efficient as possible into the nodes of the hidden layer. The smaller the number of hidden nodes, the less detail can be kept by the non-linear dimensional reduction carried out in the step between input and hidden layer. The hidden nodes act as feature detectors, they keep what they can parametrise, the rest like random noise they discard.

Auto-associative feed-forward networks have been used already with success in digital signal processing applications[56,57] outside of high energy physics. A possible implementation in high energy physics could be the removal of noise from signals of the strips of a Multi Wire Proportional Chamber (MWPC) and other situations that could benefit from a non-standard low pass filter.

Chapter 4

The DELPHI Detector

The DELPHI[58] experiment is one of the four experiments at the LEP collider facility at CERN. It has been designed as a general purpose detector for the study of e^+e^- collisions. For the last five years, LEP has been operated around the Z^0 peak to obtain high precision measurements of Standard Model parameters.

The various detector components of DELPHI, which surround the interaction point like layers of an onion, combine precise track reconstruction, calorimetry and particle identification, to make a full reconstruction of the event possible.

In this chapter the different parts of the DELPHI detector (shown in figure 4.1) will be briefly discussed. A short functional description of each component will be given as well as performance information relevant to their operation in 1992.

In the following, the (longitudinal) z-coordinate is defined as the coordinate along the beam axis, $0 < \theta < \pi$ as the polar angle and $0 < \phi < 2\pi$ as the azimuthal angle in the transverse xy-plane. Around the polar angles 40° and 140°, space was needed for cabling and support structures of the various sub-detectors. These areas are the separation between the *barrel* and *forward* (or end cap) region of the detector.

One of the sub-detectors, the Inner Detector, will be discussed in more detail in chapter 5.

4.1 Tracking Detectors

The *tracking detectors* of DELPHI handle the track reconstruction of charged particles. They all reside within the 2.75 meter radius of DELPHI's super conducting magnet: a solenoid 6.8 meters long. The magnet's very uniform axial field with a strength of 1.2 Tesla allows precise measurement of the momenta of charged tracks.

Together, the barrel tracking detectors cover the polar angles between 25° and 155° and yield a reconstruction efficiency of practically 100%. The combined average momentum resolution $\delta p/p$ ranges from 0.6 to 1.3 $10^{-3}p[GeV]$, depending on θ.

The following sections describe DELPHI's tracking detectors starting from the centre of the detector towards larger radii.

Vertex Detector (VD)

The VD consists of three cylindrical layers of silicon micro-strip detectors, which immediately surround the 6 cm radius beam pipe. The layers cover all polar angles between 44° and 136° and reside at radii of 6.3 cm, 9.0 cm, and 11.0 cm. The VD provides $r\phi$ coordinates of charged tracks close to the interaction point. This improves the reconstruction of the primary vertex and of secondary vertices from heavy flavour decays.

In 1992 the VD had an intrinsic point resolution for isolated tracks in the transverse plane of 8 μm. The track separation is better than 100 μm and the impact parameter resolution σ_{ip} depends on the transverse momentum p_t and can be parametrised as $\sigma_{ip} = (0.069/p_t[GeV/c] + 0.024)\ mm$[59].

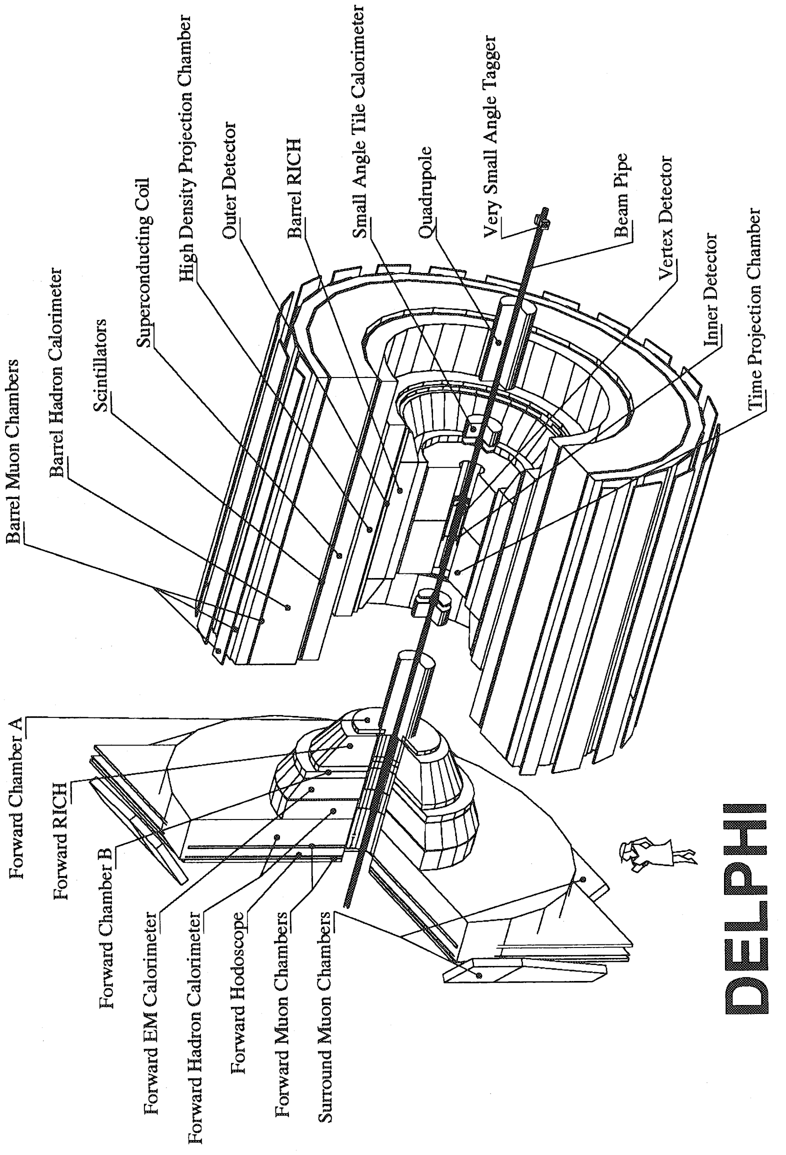


Figure 4.1: *Perspective view of the DELPHI detector.*

Inner Detector (ID)

The ID is a cylindrical wire chamber covering polar angles between 29° and 151° with an inner radius of 12 *cm* and outer radius of 28 *cm*. It comprises a jet chamber section providing 24 $r\phi$-coordinates, surrounded by five Multi Wire Proportional Chambers, which give both $r\phi$- and longitudinal z-coordinates. Together with the Outer Detector, the ID provides a fast first level trigger.

The resolution figures for the ID are $\sigma_{r\phi} = 90\ \mu m$ for the jet chamber and $\sigma_z < 1.0\ mm$ for the proportional chambers. The ID will be more elaborately discussed in chapter 5.

Time Projection Chamber (TPC)

The TPC is DELPHI's central tracking detector and provides the major part of the three dimensional spatial information. It has an inner radius of 0.35 *m* and and outer radius of 1.11 *m*. With its length of 2.60 *m*, it covers polar angles between 20° and 160°.

The large drift volume of the TPC is read out by MWPC's at the end caps and is divided in two parts by a wall at $z = 0$. The cathodes of the MWPC's do the reconstruction of $r\phi$-coordinates and drift times yield the z-coordinates. The anode wires measure $\mathrm{d}E/\mathrm{d}x$: the energy loss through ionisation per unit of length. $\mathrm{d}E/\mathrm{d}x$ behaves differently for different particles of the same momentum and is therefore used for particle identification, in particular for electron identification (see section 6.2).

The TPC's drift volume is operated at normal (atmospheric) pressure. By operating at atmospheric pressure however, some momentum and $d\mathrm{E}/dx$ resolution is lost; however the former is improved by combining all tracking detectors.

The TPC achieves a single point $r\phi$-resolution in the range 180-250 μm depending on ϕ and on z, and a z-resolution of about 1.0 *mm*.

Outer Detector (OD)

The OD consists of five layers of drift tubes, covers the polar angles between 42° and 138°, and is situated between 1.98 *m* radius and 2.06 *m* radius.

In $r\phi$, the OD provides fast trigger information in conjunction with the ID (see section 5.3 for the ID-OD trigger).

Furthermore, the OD adds an accurate $r\phi$ measurement (about 100 μm) at large distance from the interaction point, which improves the momentum resolution of fast tracks. In the z measurement, only three layers of the OD are employed; they give a fast but coarse ($\approx$4 *cm*) point, used in the trigger.

Forward Chamber A (FCA) and *Forward Chamber B* (FCB)

The FCA and FCB are positioned perpendicular to the beam axis in the end cap region of DELPHI; the former directly after the TPC and the latter between the forward RICH and the FEMC.

They are wire chambers yielding resolutions of about 300 μm in the xy-plane and measure the momentum and angle for particles produced at $\theta < 35°$. Furthermore, the forward chambers are a first level forward trigger and complete the particle trajectories before and after the two RICH counters.

4.2 Calorimetry

Calorimeters measure energies of photons, electrons and other neutral and charged particles. These particles interact with the material of the calorimeters and thereby induce a shower of particles of which energy and position are measured. The characteristics of DELPHI's electromagnetic and hadron calorimeters are described in the following paragraphs.

High-density Projection Chamber **(HPC)**

The HPC is the main electromagnetic calorimeter covering polar angles between 42° and 138°. It uses a lead converter structure for shower development interleaved with nine gas sampling volumes. An electric field extracts the resulting ionisation electrons onto a proportional wire chamber, which provides three dimensional reconstruction of electromagnetic showers. The energy resolution per electron of the HPC has been measured in $Z^0 \to e^+e^-\gamma$[60] and amounts to $(25/\sqrt{E[GeV/c]} + 6.3)\%$.

Electron identification in DELPHI relies mainly on the HPC. Identification is based on the comparison between the observed longitudinal shower profile and that of an electron of the same energy. More on electron identification can be found in section 6.2.

Forward Electro-magnetic Calorimeter **(FEMC)**

The FEMC is the electromagnetic calorimeter for the forward direction ($143.5° < \theta < 170°$ and $10° < \theta < 36.5°$). It consists of counters of homogeneous lead glass blocks equipped with one-stage photo-multipliers (triodes). The energy resolution of the FEMC was measured to be[60]:

$$\left(\frac{\sigma_E}{E}\right)_{FEMC} = \sqrt{\left(\frac{20}{\sqrt{E[GeV]}} + 4.7\right)^2 + \left(\frac{20}{E[GeV]}\right)^2}\ \%$$

Hadron Calorimeter **(HAC or HCAL)**

The HAC covers the full solid angle $10° < \theta < 170°$ (barrel + end cap) and is a sampling gas detector incorporated in the magnet yoke, which is used as absorber. It detects both neutral and charged hadronic particles and measures the total hadronic energy.

During the 1992 run the achieved energy resolution σ_E/E was about $(120/\sqrt{E[GeV]})\%$.

4.3 Particle Identification

DELPHI uses various types of information to identify particles in an event:

- Velocity measurements, see the Ring Imaging Cherenkov Counters below.
- Energy loss measurements in the TPC.
- Total (electromagnetic) energy in the calorimeters.
- Momenta of charged particles derived from their curvature in the magnetic field in all tracking detectors.
- Penetrating power: the particle in question reached the Muon Chambers, see below.

This quantitative information is combined to form a consistent 'four-vector' reconstruction of an event. Two groups of particles though, escape direct detection.

First, the short lived particles that decay before they leave the beam pipe. They can only be identified by their decay products and good vertex reconstruction by the VD.

Secondly, the neutrino's, which can not be measured directly because of their very meager interaction with matter. Their presence is deduced from missing energy and momentum in the event.

Ring Imaging Cherenkov Counters **(RICH)**

A charged particle causes emission of a light cone when it traverses a medium with a velocity which is greater than the speed of light in that medium; this is called the *Cherenkov effect*. The opening angle of the emitted light cone depends on the velocity of the particle.

DELPHI's RICH counters, which are unique at LEP, make use of this principle in both the barrel RICH and the forward RICH. They contain two types of radiators: a liquid radiator and a gas radiator. Cherenkov photons, formed in the gas radiator, are focussed by parabolic mirrors into a drift tube with MWPC read-out; the liquid gas radiator uses proximity focussing. More information about the RICH can be found in[58,61].

The RICH is used to separate protons, pions and kaons over a large momentum range, making it unique among the LEP detectors. See references[62–64] for recent studies that depend on the RICH.

***Barrel Muon Chambers* and *Forward Muon Chambers* (MUB and MUF)**

The muon chambers are a set of drift chambers giving three dimensional information about a traversing particle. Charged particles have to penetrate the HAC over its full depth to get here, so they are very likely to be muons.

In the MUB ($52^\circ < \theta < 128^\circ$) there are three sets of chambers with each set consisting of two layers of chambers. The first set lies within the slots in the iron of the HAC about 1 meter deep and the other two are just outside of the iron. Muons are tagged by demanding the presence of two reconstructed space points on the tracks of traversing particles for each set. Measurement of the angle of the charged particle separates real muons from background events. The resolutions of the MUB measured using $\mu^+\mu^-$ events extrapolated to the muon chambers are $\sigma_{r\phi} = 4\ mm$ and $\sigma_z = 25\ mm$.

The MUF handle the muon identification for the forward region, they cover the polar angles $9^\circ < \theta < 43^\circ$ and $137^\circ < \theta < 171^\circ$. The resolution figures for the MUF: $\sigma_x = \sigma_y = 3\ mm$, have been determined using so-called *halo muons*. These are muons created at the collimators far from the interaction point; they enter DELPHI parallel to the beam axis, accompanying the beams.

Detailed information about the muon identification can be found in section 6.1.

4.4 Other Sub-detectors

Luminosity detectors detect Bhabha events: $e^+e^- \rightarrow e^+e^-$. Using the known cross section for this process, they can then be used to measure/monitor the luminosity. In DELPHI these are the *Small Angle Tagger* (SAT) and *Very Small Angle Tagger* (VSAT). They distinguish photons from charged particles for tracks in $2.5^\circ < \theta < 7.7^\circ$ and $0.3^\circ < \theta < 0.4^\circ$ respectively.

Finally, the barrel and the end cap region are equipped with scintillation counters, namely the *Time-Of-Flight Counters* (TOF) and *Forward Hodoscope* (HOF). They are used to separate background events (cosmics) from real events at the trigger level.

4.5 Data Taking and Data Handling

This section describes the way experimental high energy physics is practised using the DELPHI detector; from the moment the LEP beams collide and possibly produce an interesting event, to the time the data is ready to be analysed.

Collision → Triggers → Raw Data

Four (or eight) bunches of electrons and positrons, which are accelerated by radio-frequency cavities, bent by dipole magnets and prevented to disperse by quadrupole magnets, circulate the LEP ring in opposite directions. This results in a *bunch crossing* (BCO) every 22 (or 11) μs. At the four points along the ring where the LEP detectors L3[65], ALEPH[66], OPAL[67] and DELPHI are situated, they are focussed by quadrupole magnets such that a **collision** can take place. At each BCO, a signal is sent from LEP to activate the read-out systems of the LEP detectors.

In order to discriminate between real events and background processes, DELPHI uses an hierarchical filter procedure embodying four ***trigger*** levels[68–70]: T1, T2, T3 and T4. The first two triggers T1 and T2 are implemented in hardware and are synchronous to the BCO. T3 and T4, implemented in the data acquisition system from the 1992 run on, are software filters.

The *first level trigger* (T1), studies the significance of incoming signals within 3 μs and therefore must be implemented in hardware. It comprises several sub-triggers for the various detectors and produces a global characterisation of the event. Only minimal information such as energy or transverse momentum thresholds, satisfying certain requirements in terms of multiplicity, is considered. No correlation between the different sub-detectors is introduced at this level. If an event is uninteresting, the TPC and the RICH-counters close. T1 combines all sub-trigger information and reduces the 'event' rate from the 45 (90) kHz BCO rate to about 1 kHz.

The *second level trigger* (T2), which is also implemented in hardware, makes a decision within 35 μs. It combines the trigger signals arising from the different sub-detectors. Because T2 has to wait for the data of the large drift counters (TPC and HPC), several BCO's are lost. The 1 kHz event rate, which persists after T1 is now reduced to about 3-5 Hz.

The *third level trigger* (T3) has to respond within 50 ms. A more accurate characterisation of an event is now possible with a software approach, such as a simple pattern recognition technique. T3 was implemented in 1992 with the aim of keeping the data logging rate in the vicinity of 2 Hz. After the third level trigger, the event rate is reduced to 2-3 Hz.

Finally, the *fourth level trigger* (T4) was devised in late 1992 with the aim to disentangle background triggers from physics candidates in real time. In practice T4 tags the events and rejects those regarded as empty.

Only events that have passed at least T2 are ever written to tape. These data are called ***raw data*** and consist of the information as it is directly delivered by the read-out electronics, such as drift times, charges and hit patterns.

DST → DSTFIX → N-Tuple

The raw data are treated by DELPHI's general analysis program DELANA[71] to create *Data Summary Tapes* (**DST**'s). DELANA has software modules for every sub-detector of DELPHI to perform local pattern recognition, which finds valid track elements and energy clusters.
After that, DELANA reconstructs the event by linking the individual track elements and energy clusters together to form complete tracks. It hereby relies on an updated DELPHI detector description and calibration database to provide calibration and alignment constants for each sub-detector.

After study of the DST's, it often turns out that the alignment/calibration of the detector is not completely correct and that adjustments are necessary at the single sub-detector level. Most of these inaccuracies are solved by directly adapting the DST, performing a so-called **DSTFIX**; then it is not necessary to process the bulky raw data again.

The (fixed) DST can be the starting point for the physics analysis, but many people prefer to condense the data even more, for example because their analysis doesn't need the information of all sub-detectors or only needs a limited set of DST derived variables. Programs that can read the DST, like PHDST[72], are used to produce mini-DST's and **N-tuples**, which contain a limited set of physics variables.

At NIKHEF a general purpose package has been developed called NIKLIB based on PHDST, which allows intuitive embedding of user modules and produces portable N-tuples. This package has been used to create the N-tuples that were the basis of the analysis presented in this thesis.

Analysis

After the data is available in a handy form, like a mini-DST or an N-Tuple, the analysis can begin. In order to understand the DELPHI data, simulated data produced by DELPHI's simulation program DELSIM[73] has to be studied. Simulated events have gone through the same reconstruction chain as real events, but contain extra information which reveals the true nature (momenta, particle type etc.) of all tracks. Consequently, one can check efficiencies of the reconstruction, consistency of applied cuts and many other issues.

Chapter 5

The Inner Detector

The *Inner Detector* (ID, a transverse view is shown in figure 5.1) immediately surrounds the Vertex Detector and consists of two major parts: the Jet Chamber and the Trigger Layers, which both provide a track trigger. The Jet Chamber delivers accurate points in the $r\phi$-plane and the Trigger Layers rz- and $r\phi$-coordinates. As a whole the ID is a fast single track trigger that is also used in combination with the Outer Detector (OD).

The Jet Chamber is segmented azimuthally into 24 sectors. Each sector is a separate drift chamber with an electric field arranged in such a way that drifting electrons have a constant angular velocity towards the centrally located anode plane. The drift time information of (up to) 24 anode wires gives an accurate $r\phi$ track element at about 17 cm from the interaction point.

The Trigger Layers are situated outside the Jet Chamber. They consist of five layers of cylindrical Multi Wire Proportional Chambers (MWPC's[74]). Cathode strips run perpendicular to the anode wires and form loops around the detector. A signal along an anode wire induces a charge on several cathode strips. The charge distribution on the cathodes is analysed to obtain z-coordinates. The rϕ information from the five layers of anode wires is used to resolve left-right ambiguities present in the Jet Chamber.

The following sections elaborate on the ID parts and the ID-OD trigger. After that, the ID calibration database and the determination of the internal z resolution will be described in more detail.

5.1 The Jet Chamber

The *Jet Chamber* occupies the region between 12.0 and 23.0 cm radius and consists of 24 wedge-shaped drift sectors, each containing 24 signal wires. In the gap between Jet Chamber and Trigger Layers, a mirror is installed at every fourth sector to allow laser calibration of the Jet Chamber.

The electric field in a drift sector increases linearly in the radial direction, which causes a linear dependency of the drift velocity on the distance to the interaction point. As a result, tracks originating from the vertex region give a coincident signal on all sense wires. This feature is used in the pattern recognition and in the trigger.

Because the Jet Chamber only delivers drift time information of a track, it is impossible to tell at which side of the anode plane the particle passed: every track has an associated 'mirror track'. This left-right ambiguity is resolved by the rϕ information provided by the Trigger Layers, or in a later stage by the reconstructed track segment in the TPC.

For the Jet Chamber a gas mixture consisting mainly of CO_2 (94.7%) was chosen as drift gas because of its small Lorentz angle and low diffusion coefficient. Furthermore, small amounts of isobuthane (4.6%) and alcohol (0.7%) were added, which make the detector more stable. The Jet chamber is operated at a constant pressure of 996.1 ± 0.2 $mbar$.

The drift times are measured using LTD's: Time to Digital Converters (TDC's) capable of registering multiple hits at one wire (input channel). The LTD's have a time resolution of 2.2 ns.

In practice, response time spread, charge collection, diffusion and noise cause the chamber to have a resolution of about 90 μm per point.

5.2 The Trigger Layers

The *Trigger Layers* are five layers of cylindrical MWPC's with cathode strip read-out and occupy the region between 23.0 and 28.0 cm radius and -40.0 to $+40.0$ cm in z.

Each layer (see figure 5.2) contains 192 anode wires (20 μm diameter W/Au) and 192 cathode strips. Signals on the cathodes are read out from 192 pickup strips, which are parted from the cathodes by a 100 μm Kapton layer. The anode wires are separated by field wires (70 μm diameter Cu/Be) to form 9 mm wide drift cells. The remainder of the layer is a 2 mm thick cylinder wall consisting of support and shielding material.

The drift gas employed in the Trigger Layers is Ar/CO_2 (70%/30%), which gives a good cell separation in a perpendicular magnetic field[75].

The charges on both the anode wires and the pickup strips are sampled by 8-bit Flash Analog to Digital Converters (FADC's[76]). One FADC handles 96 channels, therefore four FADC's are needed to cover the 192 anodes and 192 cathodes of each trigger layer. The FADC's deliver 32 samples of the charge pulse at 15 MHz with values in the 0-255 range, which are summed to a total pulse height for both anodes and cathodes.

A non-zero pulse height on the anodes produces an $r\phi$-trigger bit. If the pattern formed by the $r\phi$-trigger bits in different layers indicate a valid track, this track (and possibly track elements from the TPC or OD) is used to solve the $r\phi$ ambiguity of the Jet Chamber.

The cathodes carry signals induced by charges on the anode wires. The induced charge originating from one anode wire is spread out over 2-7 cathode strips. First, a cluster algorithm disentangles the individual induced charges. After that, calculating the centre of a charge cluster gives the z-coordinate.

The determination of the internal z resolution using the Trigger Layers is described in detail in section 5.5.

5.3 The ID-OD Trigger

This section describes the *ID-OD track trigger* with emphasis on the ID trigger and how its output is combined with the OD. Already at the first level trigger, trigger information of the ID is combined with that of the OD. At the second level the ID outputs the same information while the OD uses z information to confirm its $r\phi$ information.

The Jet Chamber returns 24 trigger bits in $r\phi$, one for each sector, the Trigger Layers similarly return 24 $r\phi$ trigger bits for each 'sector' plus 48 trigger bits in rz. Two hardware modules implement the ID trigger: the JET Trigger Module (JETTM[77]) and the LAYer Trigger Module (LAYTM[78]). In this section only the LAYTM will be discussed in some detail, more information about the JETTM can be found in reference[50].

LAYTM handles both the $r\phi$ and the rz triggering, which are implemented on two identical electronic boards. The two different functions can be jumper selected.

The LAYTM $r\phi$-trigger is implemented by 12 trigger modules each containing 16 Track Finder (TF) chips; a single trigger module handles two neighbouring sectors. A TF-chip monitors eleven anode wires centred around an anode wire of the middle (third) trigger layer and looks for valid hit patterns: the patterns are displayed in Figure 5.3. If a TF-chip lies at a sector boundary, neighbouring wires are taken into account.

Three trigger modules implement the LAYTM rz-trigger, where each module handles 16 super-strips in 5 layers. A super-strip consists of 4 adjacent cathode strips whose pulse heights are summed. Such a super-strip may or may not contain a pulse height and like in $r\phi$ the TF-chips search for valid hit patterns in rz.

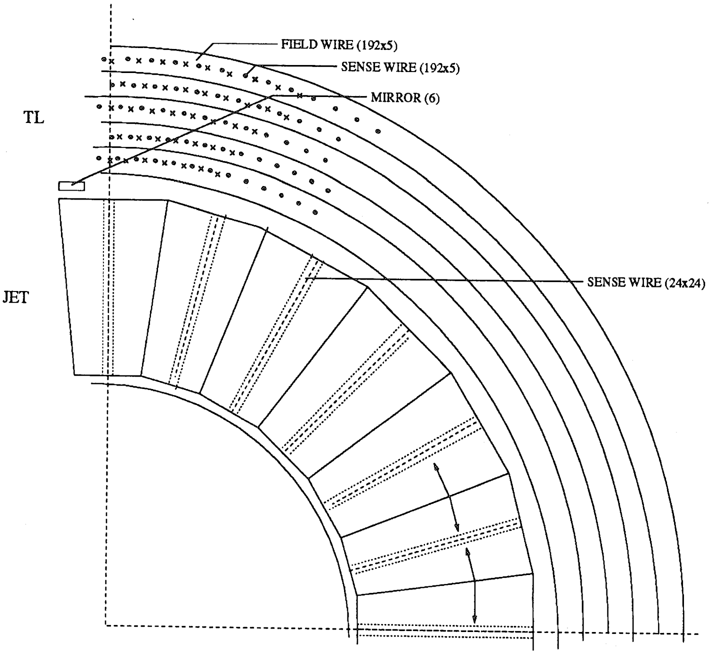


Figure 5.1: *Transverse view of the first Inner Detector quadrant.*

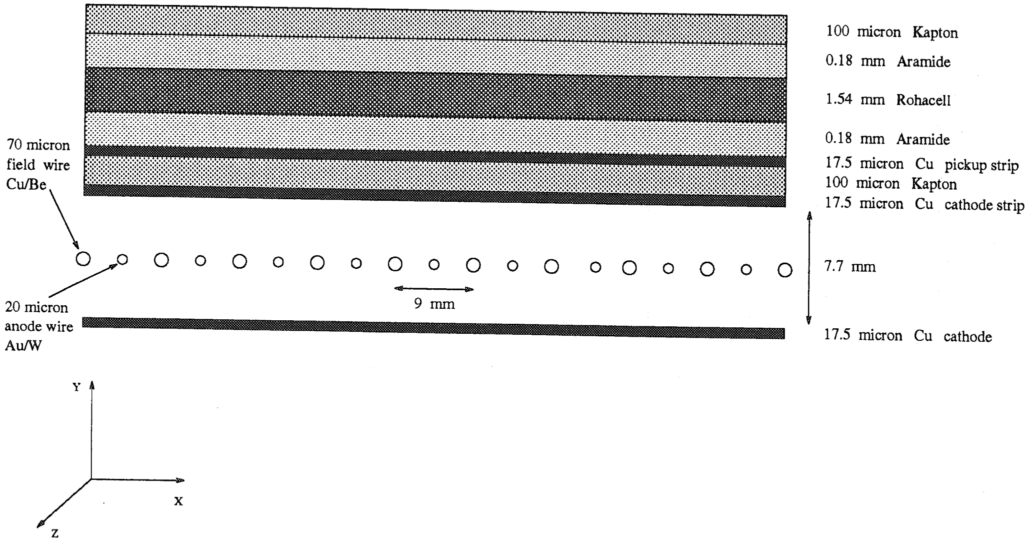


Figure 5.2: *A Trigger Layer in detail.*

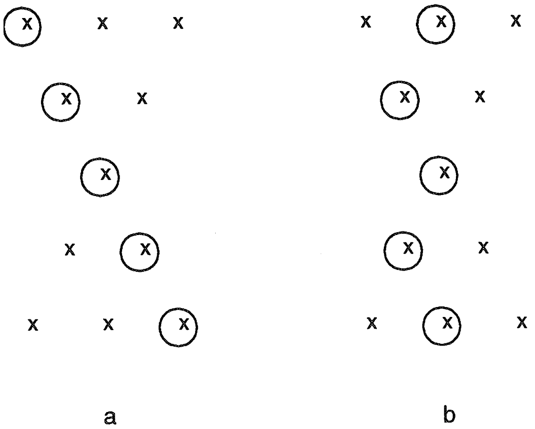


Figure 5.3: *The two valid hit patterns in a TF-chip region, a diagonal pattern (a) and a straight pattern (b); the other two valid patterns are their mirror images.*

The following summarises the operation of the ID-OD trigger:

- JETTM checks for valid tracks in each sector.

  The JETTM sets a trigger bit for a sector when at least 12 out of the 24 wires receive a signal within a time window of about 200 *ns*. The JETTM modules output a total of 24 trigger bits, one for each sector.

- LAYTM checks for valid tracks in each sector.

  For the $r\phi$-trigger, LAYTM looks for (one of the four) valid track patterns, in which at least 3 out of 5 layers must have a hit. If there is a valid track somewhere in the TF-chips of a sector (logical OR) then the $r\phi$-trigger bit of this sector is set. The LAYTM modules output a total of 24 $r\phi$-trigger bits.

  For the rz-trigger, LAYTM similarly looks for valid rz-patterns in the super-strips of the five layers and a rz-trigger bit is set when at least 3 out of 5 layers in a super-strip are hit. In total there are 48 rz-trigger bits.

- Trigger information from ID and OD is combined.

  Valid ID tracks in $r\phi$ are found by a sector by sector logical AND of JETTM and LAYTM. The first level trigger of the ID consists of the OR of all 24 ID bits, the z bits are not used.

  ID tracks are ANDed with the OD, but only for the six sectors that correspond to the holes in the TPC because of its segmentation in $r\phi$.

  The ID-OD correlation in $r\phi$ is (ID track in sector N) AND (OD track in sector N+a), with $a = -1$ or 0 or +1. For the correlation in z, rz-patterns in the ID are combined with rz-patterns in the Outer Detector.

After the above combined first and second level trigger, the third level trigger of the ID operates on the remaining tracks only in $r\phi$. The central third level trigger, data transfer and other overhead leave 20 *ms* for the third level trigger decision of the ID. This turned out to be long enough to implement the third level trigger algorithm in software. It has a high efficiency ($98.5 \pm 0.8\%$) for tracks with transverse momentum $p_t > 2.0\ GeV/c$ and rejects tracks not coming from the interaction region.

Detailed information about the ID third level trigger can be found in reference[79].

5.4 The Inner Detector Calibration Database

The Inner Detector calibration needs about 1 permille sensitivity of the relevant parameters, because this corresponds to about 25 μm variation on a track in the Jet Chamber, its most sensitive component.

All parameters are recorded during the data-taking in the on-line database and updated at the moment their change influences the calibration by more than 1 permille. After a maximum of one day, the on-line updates are imported into the off-line calibration database.

The ID calibration database comprises the following parameters and their sensitivities :

Gas Information, 1 *mbar* or 1 *vol* % sensitivity.

The atmospheric pressure (PRESA), the Jet Chamber pressure (PRESJ), and volume percentages of the gas mixture in the Jet Chamber: nominally at 94.7% CO_2, 4.6% isobuthane and 0.7% alcohol.

Furthermore, the volume percentages of the gas mixture in the Trigger Layers: 70.0% Argon and 30.0% CO_2.

Temperatures (TEMP), 0.3° *C* sensitivity.

Ten temperatures measured by probes located at various places around the Inner Detector. During normal operation these temperatures read in the order of 27 degrees centigrade.

Jet Chamber and Trigger Layers High Voltages, 2 *V* sensitivity.

The high voltages of the 24 sectors of the Jet Chamber (JET HV, nominally at 2224 *V*) and the high voltages of the 5 planes of the Trigger Layers (TL HV at 1600, 1610, 1620, 1630 and 1640 *V* respectively).

Jet Chamber Field Shaping Voltages, 3 *V* sensitivity.

These are the 24 field shaping voltages (JETFS 1-24, nominally at 5500 *V*) and the voltage applied on the inner cylinder (VCYL at 2750 *V*). For each sector in the Jet Chamber, JETFS feeds a resistor chain running along the edges which produces an isochronous drift field. The actual field shaping voltages are monitored by means of a digital voltmeter measuring the voltage over a 1 in 1000 resistor divider.

All calibration parameters are stored in so-called GASD(ata) and SLOW (control) fields of the calibration tree of the off-line database[80]. The records from the ID on-line database are checked during data taking by the monitor programme DBMON. It only performs simple monitoring, because the load of on-line processing jobs must be kept low.

It is very important that the information on the off-line database is correct, because DELANA uses the database for the track reconstruction. In particular errors have to be corrected before DELANA's main (re)processings, which produce the most recent batches of tapes for physics analyses.

The ID off-line calibration database is checked regularly by the monitor programme DBPIT, which is described in detail in appendix D.

5.5 Z Measurement with the Trigger Layers

A traversing particle ionises the drift gas in the Trigger Layers, which is followed by the forming of an electron avalanche. The avalanche forms the signal on the anode wire and the signal induces a charge spread out over a cluster of typically 2-7 adjacent cathode strips.

In general, the z-coordinate of a particle traversing a MWPC is obtained by determining the centre of the charge distribution on the cathodes[81,82]. In the Trigger Layers, cathode charge centres are calculated with the modified Centre-Of-Gravity (COG) algorithm.

The *COG algorithm* computes a weighted sum of the strip charges in a cluster to find the centre of the induced charge pattern. If one adds a threshold for strip charges, the modified COG[81] of a cluster of n_c adjacent induced charges A_i at positions Z_i becomes:

$$\bar{Z} = \frac{\sum_{i=1}^{n_c}(A_i - B)Z_i}{\sum_{i=1}^{n_c}(A_i - B)} \tag{5.1}$$

for $(A_i - B) > 0$, where:

$$B = b\sum_{i=1}^{n_c} A_i \tag{5.2}$$

and b is a bias level. The bias level and the number of contributing strips n_c must be suitably chosen to reduce the effects of pickup and electronic noise. Wrong choices of n_c can lead to unnecessary big systematic errors[83].

The modified COG delivered satisfying results in the prototype test of summer 1987[84], where a z resolution of 125 ± 15 μm was obtained with small bias level (~2%) utilising 6 significant strips in the COG for tracks with incident angles between ±4 degrees.

Tracks in the ID have in the z projection incident angles varying from 0 to 60 degrees. As shown by Hajduk[85], *Landau fluctuations*, which are fluctuations in the anode (and thus cathode) charge distribution, become important if the incident angles are bigger than about 20 degrees.

In 1989, a study of the internal z resolution was performed for tracks from a pion test beam with an incident angle of about 30 degrees[86]. This measurement indicated that the overall z resolution was about 0.9 *mm*, but the quality of the data was far from optimal. There were electronic pickup signals on the strips and only planes 2,4 and 5 were partly connected because only one FADC was available at the time. Nevertheless, it was suspected that it would be difficult to reach the overall design resolution of about 0.5 *mm*.

During the real data taking it indeed became clear that tracks in hadronic Z^0 events had internal z resolutions of about 1 *mm*. Although this 1 *mm* was at that time the best z-measurement available in DELPHI, it was still somewhat disappointing.

In 1991 and 1992, various attempts have been undertaken to improve the resolution, but invariably one found internal single track resolutions no better than 700 μm averaged over all angles.

In the next section the result of one of these attempts is described, using isolated tracks coming from 1992 lepton-pair events.

5.5.1 Results

The following choices were made for single tracks participating in the measurement of the internal z resolution:

- Pulse heights were only accepted if there were more than 6 FADC samples available. FADC overflows, if present, were corrected with a parabolic fit around the top.
- $b = 0.02$ in the COG calculation.
- The induced charge cluster must be 3-7 strips wide.

- At least 4 out of 5 planes must be hit. The difference between a data point and the fit through the other points forms the residual distribution.

Figure 5.4 displays the obtained z residual distribution; a double Gaussian fit is applied to make better quality tracks (the ones with lower incident angles) visible. The width of the main Gaussian indicates an average z resolution of 800 μm. Figure 5.5 shows how the residuals depend on the polar angle θ. An incident angle of 0 degrees corresponds to a polar angle of 90 degrees.

Finally, figure 5.6 displays the z resolution as a function of the polar angle θ.

There could be various reasons why the measured z resolution is worse than the prototype promised.

First of all, the prototype was flat and the real detector is not. Hardware imperfections, in particular a slight distortion of the ID cylinder, therefore have a significant influence on the z reconstruction.

In addition, varying noise conditions probably yielded non-optimal calibration constants for the cathodes, which are for each cathode strip parametrised as a gain value and an average noise level (pedestal).

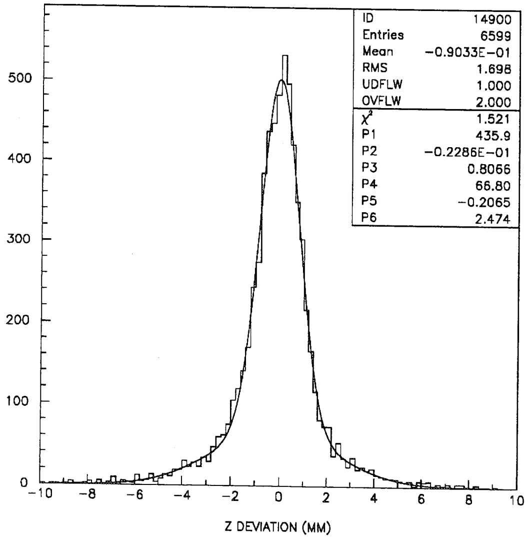


Figure 5.4: *Z residual distribution for* $b = 0.02$.

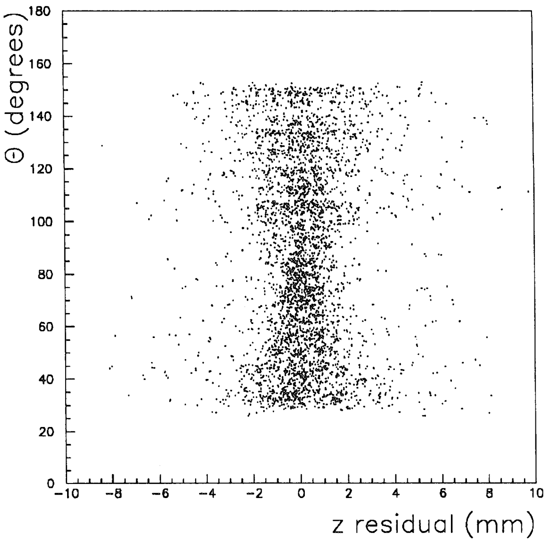


Figure 5.5: *The polar angle θ (degrees) versus the z residuals (mm).*

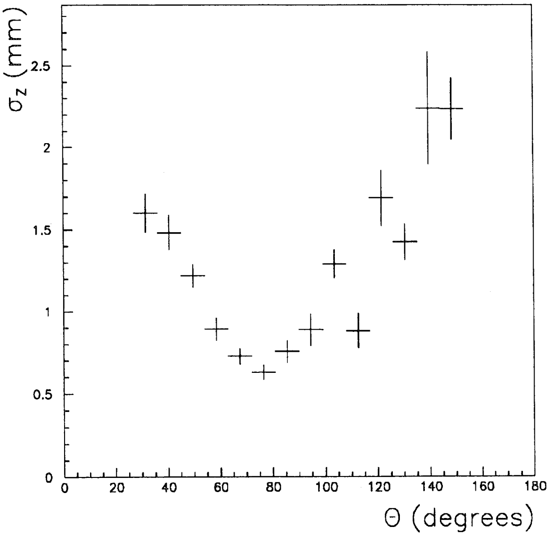


Figure 5.6: *Z resolution (mm) as a function of the polar angle θ (degrees).*

Chapter 6

Determination of the B^0-$\bar{B}^0$ Mixing Parameter

This chapter contains the determination of the B^0-$\bar{B}^0$ mixing parameter χ, using a dilepton sample extracted from data taken by DELPHI in 1992.

The first two sections elaborate on lepton identification in DELPHI. To identify muons the package MUFLAG was used and for electrons the package ELPROB; both packages will be introduced.

The next section describes the composition of the simulation and data samples, which formed the starting point for the extraction of the mixing parameter χ.

After that, a neural network classifier is introduced, which is used to suppress the decays that are background for the mixing measurement. The performance of nn^{min}, the variable constructed by the neural network, is compared to that of the common minimum transverse momentum p_t^{min} in the dilepton.

Finally, χ is determined exploiting p_t^{min} or nn^{min} as a separating variable. This chapter ends with a discussion of the obtained results.

6.1 Muon Identification

DELPHI's muon chambers in the barrel (MUB) and in the forward region (MUF) are employed to identify muons. In a mixing measurement one is primarily interested in muons originating from heavy quarks. The background consists of particles that penetrate the iron of the hadron calorimeter and give a signal in the muon chambers:

- 'Punch-through'.

  If a hadron interacts in the iron of the calorimeter, secondary particles like pions, kaons and protons are produced. If they leave the iron, they can give a signal in the muon chambers and are then misidentified as muons.

- 'Sail-through'.

  This refers to hadrons that have traversed the calorimeter without interacting and are misidentified as muons in the muon chambers.

- Muons from light hadron decays.

  Light hadrons which decay in flight into muons (e.g. $K^+ \rightarrow \pi^0\mu^+\nu_\mu$), which are detected in the muon chambers.

Backgrounds from the above sources are reduced by extrapolating tracks from the tracking detectors to the muon chambers and associating them to the measured hits.

In addition, punch-through is substantially reduced by requiring at least one hit in the outer layers of the muon chambers. The outer layers are layers 4-7 for the MUB and layers 3-4 for the MUF.

MUFLAG[87] is the standard muon identification package in DELPHI. In the MUFLAG environment, all tracks which are potential muons are identified and re-fitted (with MUCFIX[88]) under conditions suitable for a specific tag philosophy.

All muon information available to MUFLAG comes originally from the DELANA module EMMASS[89] that fits tracks to muon chamber hits. MUFLAG compares measured coordinates ($R\phi$ and z for MUB, x and y for MUF) to the equivalent coordinates of the extrapolated track, plus the polar and azimuthal directions Θ and Φ.

EMMASS delivers chi-squared's, which are a measure of the association quality, and hit patterns of the associated muon chamber layers. They are a global chi-squared of fit, χ^2_{global}, and χ^2_{ex} that expresses the contribution to the fit from the extrapolation. χ^2_{global} is build up from χ^2_{ex} and χ^2_{muc}, the contribution to the fit from the muon chamber hit points only.

In addition, MUFLAG constructs chi-squared's that are used as tagging variables for the barrel. In the following formulae the subscript 'muc' refers to the muon chamber measurement points, 'fit' to the internally fitted muon chamber track, and 'ex' to the extrapolated track.
First, there is the extrapolated azimuthal chi-squared:

$$\chi^2_{ex\ azth} \equiv \left(\frac{R\phi_{fit} - R\phi_{ex}}{\sigma^{R\phi}_{ex}}\right)^2 + \left(\frac{\Phi_{fit} - \Phi_{ex}}{\sigma^{\Phi}_{ex}}\right)^2 \tag{6.1}$$

Analogously, the extrapolated polar chi-squared is defined by:

$$\chi^2_{ex\ polar} \equiv \left(\frac{z_{fit} - z_{ex}}{\sigma^{z}_{ex}}\right)^2 + \left(\frac{\Theta_{fit} - \Theta_{ex}}{\sigma^{\Theta}_{ex}}\right)^2 \tag{6.2}$$

Finally, the global azimuthal chi-squared, defined by:

$$\chi^2_{global\ azth} \equiv \chi^2_{ex\ azth} + \sum_{layers} \left(\frac{R\phi_{fit} - R\phi_{muc}}{\sigma^{R\phi}_{muc}}\right)^2 \tag{6.3}$$

where the sum runs over the layers with associated hits.

In the re-fit performed by MUCFIX, bad hits are dropped according to a χ^2_{hit}, which for the MUB is given by:

$$\chi^2_{hit} \equiv \left(\frac{R\phi_{fit} - R\phi_{muc}}{\sigma^{R\phi}_{muc}}\right)^2 + \left(\frac{z_{fit} - z_{muc}}{\sigma^{z}_{muc}}\right)^2 \tag{6.4}$$

and for the MUF it is:

$$\chi^2_{hit} \equiv \left(\frac{x_{fit} - x_{muc}}{\sigma^{x}_{muc}}\right)^2 + \left(\frac{y_{fit} - y_{muc}}{\sigma^{y}_{muc}}\right)^2 \tag{6.5}$$

Three muon tags have been implemented in MUFLAG. By cutting on the chi-squared's with increasing severity a *loose*, a *standard* and a *tight* muon identification level is achieved. For the loose and standard selections the last term in equation (6.4) is zero, because they don't use the z information in the refit.

Table 6.1 summarises the cuts for each identification level and table 6.2 shows their identification efficiencies and misidentification probabilities in hadronic events.

Tag	MUB	MUF
Loose	$\chi^2_{ex\ azth}$ p.d.f. ≤ 7 $\chi^2_{global\ azth}$ p.d.f. ≤ 7 $\chi^2_{ex\ polar}$ p.d.f. ≤ 10 $\chi^2_{hit} > 10$	χ^2_{global} p.d.f. ≤ 7 $\chi^2_{hit} > 20$
Standard	≥ 1 hit in outer layers $\chi^2_{ex\ azth}$ p.d.f. ≤ 5 $\chi^2_{global\ azth}$ p.d.f. ≤ 5 $\chi^2_{ex\ polar}$ p.d.f. ≤ 8 $\chi^2_{hit} > 10$	≥ 1 hit in outer layers χ^2_{global} p.d.f. ≤ 5 if $z > 0$ χ^2_{global} p.d.f. ≤ 6 if $z < 0$ χ^2_{ex} p.d.f. ≤ 5 $\chi^2_{hit} > 20$
Tight	≥ 1 hit in outer layers χ^2_{global} p.d.f. ≤ 2.5 $\chi^2_{hit} > 8$	≥ 1 hit in outer layers χ^2_{global} p.d.f. ≤ 2.0 $\chi^2_{hit} > 10$

Table 6.1: *The cuts for the loose, standard and tight muon identification.*

Tag	Efficiency (%)		Misid. Probability (%)	
	MUB	MUF	MUB	MUF
Loose	90.3 ± 0.8	(87 ± 1)	1.18 ± 0.04	(1.52 ± 0.06)
Standard	81 ± 1	(82 ± 2)	0.70 ± 0.03	(0.90 ± 0.05)
Tight	66 ± 2	(63 ± 2)	0.47 ± 0.03	(0.53 ± 0.05)

Table 6.2: *MUFLAG: efficiencies and misidentification probabilities for the loose, standard and tight muon identification.*

6.2 Electron Identification

Information coming from the TPC, the RICH and the HPC is used for electron identification in DELPHI. The HPC response is a vital component of any possible electron identification scheme, therefore identification of electrons is limited to the HPC angular coverage $42° < \theta < 138°$. A possible electron tag has the following variables at its disposal:

- $\mathrm{d}E/\mathrm{d}x$

  $\mathrm{d}E/\mathrm{d}x$ is the energy loss per unit of length of a particle measured in the TPC. For a reliable value it is required that at least 20 wires out of 192 TPC wires contribute to the $\mathrm{d}E/\mathrm{d}x$ measurement. The mean value of $\mathrm{d}E/\mathrm{d}x$ of which the scale is fixed such that a minimum ionising particle (*mip*) has a value of 1.0, is bigger for electrons than for hadrons and muons.

  $\mathrm{d}E/\mathrm{d}x$ is always used in combination with other electron tagging variables since it divides candidate tracks into two sets: the 'signal', which is not very pure yet, and a background containing very little electrons. For that purpose it is save to apply the cut $\mathrm{d}E/\mathrm{d}x > 1.3$ *mip*.

- RICH tag.

  Electrons always radiate photons under the saturated angle. One can search the RICH for particles with a momentum between 0.7 and 2.1 *GeV* (muon gas threshold) whether there are photons in the saturated angle.

- Shower position: Δz and $\Delta\phi$.

  $\Delta z = z_{ex} - z_{HPC}$ is a measure for the quality of the track-shower association. It is the difference between the z position of the shower in the HPC and the one predicted from the track extrapolation. $\Delta\phi = \phi_{ex} - \phi_{HPC}$ is the similar quantity for the azimuthal association; ϕ_{HPC} is the direction of the shower axis.

  There are two reasons why the track-shower association is not perfect. First, if the energy deposition in the HPC is too small, then the shower is lost because it can not be reconstructed by DELANA. This effect is only important for tracks below 5 *GeV*. Secondly, if a hard photon is radiated then the electron deviates from its original trajectory and the actual impact point in the HPC doesn't match the one predicted by the track extrapolation anymore. The loose association criteria of DELANA prevent these candidates to be rejected at an early stage.

- Shower profile.

  A match between the fraction of the total energy collected by the nine longitudinal layers of the HPC and the expectation for an electron having the same energy as the one measured. The quality of the match is expressed in a chi-squared of the form:

  $$\chi^2_{shower} \equiv \sum_i (F_i - < F_i >)^2/\sigma_i^2 \tag{6.6}$$

  where the sum runs over the nine layers of the HPC, F_i is the observed energy fraction and $< F_i >$ and σ_i are the mean and the root mean square of the distribution of the expected energy fraction. Their values are determined as a function of the electron energy using pure electron samples from $Z^0 \rightarrow e^+e^-(\gamma)$ and $\tau^- \rightarrow e^-\bar{\nu}_e\nu_\tau$. In the energy parametrisation HPC 'cracks' in θ are also taken into account. On 1992 data the efficiency to select Compton electrons using the shower profile was found to be $(86 \pm 1.3)\%$[90].

- E/p

  The ratio of the energy released by the particle in the HPC and its reconstructed momentum from the tracking detectors. In an ideal world electrons loose all their energy inside the HPC and hadrons deposit most of their energy in the hadron calorimeter. Selection of electrons is based on the variable η:

  $$\eta \equiv (E/p - < E/p >)/\sigma \tag{6.7}$$

where the dependence of the mean value $< E/p >$ and standard deviation σ on momentum and shower energy are tuned using the above mentioned pure electron samples.

The correction of $< E/p >$ involves removing the effect of matter before the HPC: an uncorrected $< E/p >$ decreases as the track momentum measured by tracking detectors is smaller. After tuning, a simple cut of $E/p > 0.5$ can be applied which retains almost all electrons and rejects about 60% of the pions.

In an electron tagged sample, the following backgrounds are expected:

- Misidentified hadrons.

  A hadron can simulate an electron by producing a π^0 when interacting with the HPC lead. Furthermore, a hadron can overlap with a true photon shower in the HPC.

- *Photon conversions* ($\gamma \rightarrow e^+e^-$).

  These are electrons produced from the conversion of photons in the detector material in front of the HPC. This material corresponds to about 0.7 radiation lengths, of which about 0.14 is before the TPC. If the conversion occurs after the TPC no valid track is reconstructed, but the isolated shower may overlap with another shower.

- Electrons from light hadron decays.

  If light hadrons decay in flight into electrons (e.g. $K^- \rightarrow \pi^0 e^- \bar{\nu_e}$), they may give a signal in the HPC.

Most of the background is reduced by as good a match as possible between the HPC shower and the electron candidate. Photon conversions are suppressed by a V^0 search, in which one tries to reconstruct the mass of a possible e^+e^--pair.

ELPROB[90] is an electron identification package using various combinations of the above mentioned tagging variables, which it converts to probabilities. It includes extensive HPC fixing, which comprises fine tuning of the shower variables and E/p.

In order to be examined by ELPROB, electron candidates must have a minimum momentum of 1.0 GeV/c and have deposited more than 0.3 GeV in the HPC. The tagging variables of ELPROB are:

- P($\mathrm{d}E/\mathrm{d}x$), the probability from the $\mathrm{d}E/\mathrm{d}x$ measurement of the TPC.
- P(E/p), the probability from the value of E/p.
- P(Δz) and P($\Delta\phi$), the probabilities reflecting the quality of the shower position match.
- P(sh), the probability derived from a fit to the longitudinal shower profile.
- The RICH tag, which counts the number of photons in the saturated angle.

ELPROB also constructs a combined probability P(HPC) from the HPC variables P(Δz), P(sh), P($\Delta\phi$) and P(E/p); it is only filled if $E/p > 0.5$.

In analogy to MUFLAG, ELPROB divides electron candidates into three categories: loose, standard and tight.

Table 6.3 shows the efficiencies and misidentification probabilities of the ELPROB tags in hadronic events. A distinction is made between the case in which no reliable $\mathrm{d}E/\mathrm{d}x$ is available and in which there is. The former ELPROB cuts are displayed in table 6.4 and the latter in table 6.5.

Tag	Efficiency (%)	Misid. Probability (%)
Loose	85	7
Standard	65	0.6
Tight	45	0.1

Table 6.3: *ELPROB: efficiencies and misidentification probabilities for the loose, standard and tight electron identification. The errors on these values are in the order of 10-15%.*

Tag					
Loose	P(HPC)> 0.01	OR	RICH>= 2.0	OR	P(HPC)> 0.05 RICH>= 1.0
Standard	P(Δz)> 0.05 P($\Delta\phi$)> 0.02 P(sh)> 0.02 P(E/p)> 0.05	OR	P(HPC)> 0.10 RICH> 1.0		
Tight	P(HPC)> 0.25	OR	P(HPC)> 0.15 RICH> 2.0		

Table 6.4: *The cuts of the loose, standard and tight electron identification if no dE/dx information is available.*

Tag					
Loose	P(dE/dx)> 0.02 P(HPC)> 0.001	OR	P(dE/dx)> 0.05 RICH>= 1.0	OR	$p_e <= 5.0\ GeV/c$ P(dE/dx)> 0.20
Standard	dE/dx > 1.3 P(Δz)> 0.05 P($\Delta\phi$)> 0.02 P(sh)> 0.02 P(E/p)> 0.05	OR	P(dE/dx)> 0.10 RICH> 1.0		
Tight	P(dE/dx)> 0.10 P(HPC)> 0.10	OR	P(dE/dx)> 0.10 RICH> 2.0		

Table 6.5: *The cuts of the loose, standard and tight electron identification; dE/dx information is available.*

6.3 Event Selection and Lepton Identification

Hadronic decays $Z^0 \to q\bar{q}$ were tagged by using the following criteria:

- Events had to contain at least 7 charged tracks. This removes the events of the type $Z^0 \to \tau^+\tau^-$ and thereby the lepton background from direct tau decays.
- The total visible energy $E_{charged} + E_{neutral}$ had to be greater than 0.30 E_{cm}, where E_{cm} is the centre-of-mass energy of LEP. $E_{charged}$ is the energy from charged particles detected in the tracking chambers and $E_{neutral}$ the energy from the neutrals detected in the calorimeters.

The selection of hadronic events yielded about 256000 events for the 1991 data and about 720000 for the 1992 data. In the event reconstruction both charged and neutral tracks were taken into account. Charged tracks had to satisfy the following criteria:

- Track momentum $p > 0.2\ GeV/c$.
- Relative error on the momentum $\sigma_p/p < 100\%$.
- Track length $> 30\ cm$.
- z and $r\phi$ projection of the tracks' impact parameter less than 10 respectively 5 cm.
- Angle θ with the beam direction, $|\cos\theta| < 0.93$, TPC acceptance.

Neutral tracks had to satisfy:

- $0.8 < E_{HPC} < 30.0\ GeV/c$.
- $0.4 < E_{EMF} < 30.0\ GeV/c$.
- minimum 3 layers in HPC hit.
- $\sigma_E/E < 100\%$ for both HPC and EMF.
- $|\cos\theta| < 0.98$.

Leptons were identified by applying the standard tag, both for the electrons and the muons. Because the ELPROB tags were not available for the 1991 data, only muons were tagged for 1991. A pre-selective cut was made on the lepton momenta at 3 GeV/c in order to remove the kinematical region which is most heavily dominated by background events. Furthermore, a V^0 search was applied to reject electron candidates coming from converted photons.

Two samples of simulated $Z^0 \to q\bar{q}$ decays, corresponding to the 1991 and 1992 detector setups were used for the analysis. They were generated with the LUND Parton Shower model in the JETSET 7.3 program[51], passed through the full detector simulation and processed with the same event reconstruction as the data.

6.4 Single Lepton Composition

The selections described in the previous section are applied to obtain hadronic jets containing a lepton. If a jet had more than one lepton, the lepton with the highest transverse momentum with respect to the jet direction was chosen.

The *transverse momentum* p_t was computed by first removing the track from the jet it belongs to and after that recomputing the jet direction. A p_t calculated by first excluding the lepton is called p_t^{out}, if the lepton is included it is called p_t^{in}. From now on, the variable p_t is a p_t^{out}, unless otherwise stated.

Table 6.6 shows the decay fractions, before and after a cut of $p_t > 1.0$ GeV/c, for single electrons and muons in the 1992 simulation; charge conjugate decays are implied where appropriate. This table corresponds to about 29000 single electrons and 36000 single muons, 10000 electrons and 16000 muons remained after $p_t > 1.0$ GeV/c. The most important electron backgrounds are misidentified pions and photon conversions which were not recognised by the V^0 search. Prominent muon backgrounds are misidentified pions and kaons.

In a mixing measurement it is convenient to assign single jets containing a lepton to one of the following categories:

- PB (Primary b)

  Primary semi-leptonic decays of b-hadrons: $b \to l^-$. This also includes $b \to \tau^- \to l^-$.
- SC (Secondary c)

  Cascade decays from the charm daughter of a b-parent: $b \to c \to l^+$.
- $\overline{\text{SC}}$ (Secondary $\bar{c}$)

  Cascade decays of a charm state from the W^- in the b-decay, denoted $b \to (\bar{c}s) \to l^-$ or shortly as $b \to \bar{c} \to l^-$.
- PC (Primary c)

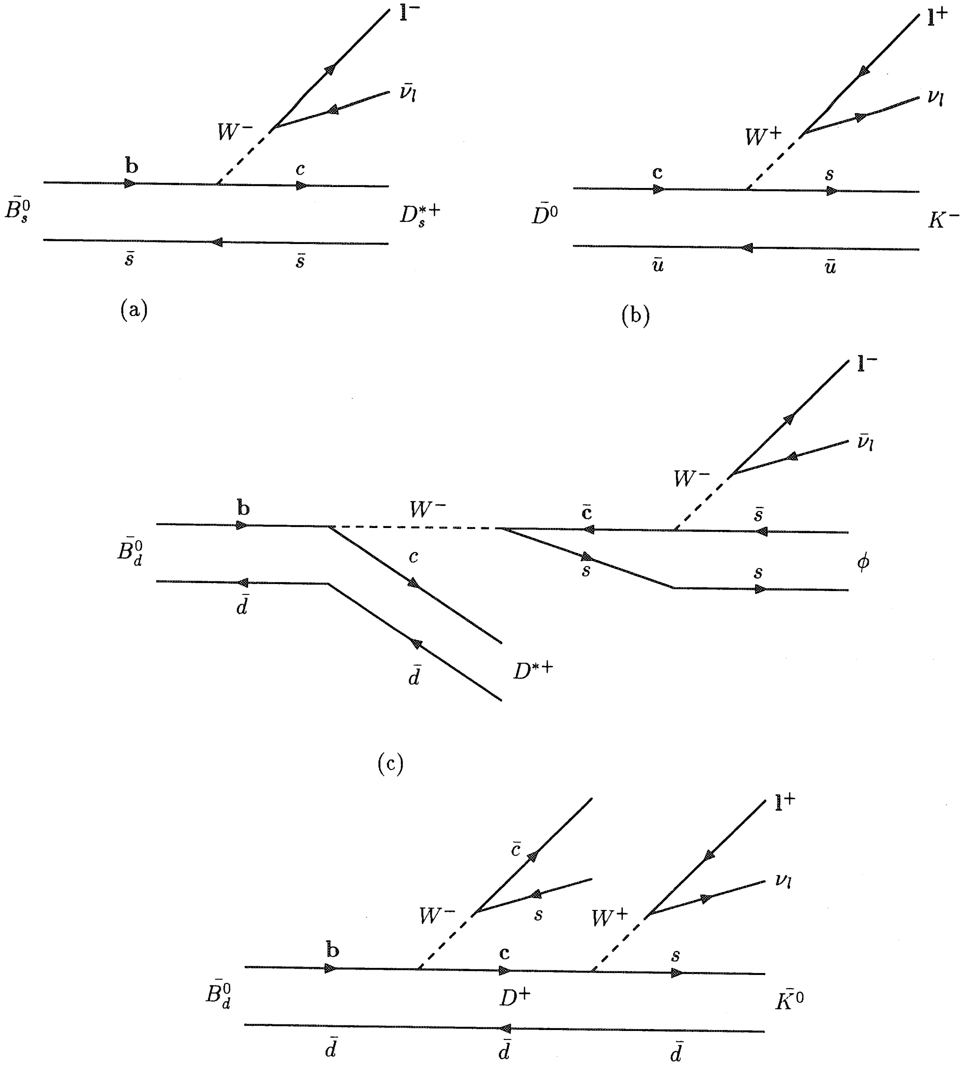
  Semi-leptonic decays of charm states produced in $Z^0 \to c\bar{c}$, denoted $c \to l^+$.
- BCK (Background)

  The remaining decays: hadrons misidentified as leptons, leptons from π and K decays and from J/Ψ. Furthermore, electrons from converted photons or *Dalitz decays* ($\pi^0 \to e^+e^-\gamma$ or $\pi^0 \to e^+e^-e^+e^-$).

Figure 6.1 displays examples of Feynman diagrams of semi-leptonic decays belonging to the categories PB, PC, $\overline{\text{SC}}$ and SC respectively.

Lepton source	Electrons		Muons	
	No p_t cut	$p_t > 1.0$ GeV/c	No p_t cut	$p_t > 1.0$ GeV/c
$b \to l^-$	25	58	33	61
$b \to \tau^- \to l^-$	1	1	1	1
$b \to c \to l^+$	9	8	14	10
$b \to \bar{c} \to l^-$	<1	<1	<1	<1
$b \to J/\Psi \to l^\pm$	<1	<1	<1	<1
$c \to l^+$	12	9	17	12
light hadron $\to l^\pm$	3	4	7	2
converted photons	10	3	-	-
misid. pions	32	17	17	7
misid. kaons	4	2	9	6
misid. protons	2	<1	<1	<1
others	<1	<1	<1	<1

Table 6.6: *The contribution (in %) of various single lepton sources in the 1992 simulation, before and after* $p_t > 1.0$ GeV/c.



(a) (b) (c) (d)

Figure 6.1: *Explicit examples of decays belonging to the classes PB, PC, $\overline{SC}$ and SC, respectively. (a) $\bar{B}^0_s \rightarrow l^-$. (b) $\bar{D}^0 \rightarrow l^+$, with the $\bar{D}^0$ directly produced in $Z^0 \rightarrow D^0\bar{D}^0$. (c) $\bar{B}^0_d \rightarrow (\bar{c}s) \rightarrow l^-$. (d) $\bar{B}^0_d \rightarrow D^+ \rightarrow l^+$.*

6.5 Dilepton Selection and Composition

Dileptons, selected for the measurement of χ, must have two back-to-back leptons. To ensure that the leptons are from different b-hadron decays, the angle between the two leptons was required to be bigger than 60 degrees. If more than two jets were found, the pair with leptons with the highest transverse momenta was considered.

Table 6.7 shows the number of like sign, unlike sign and total dileptons for the 1991 and 1992 samples used in this analysis. The 1991 simulation corresponds to about 473000 hadronic events and the 1992 simulation to about 630000 hadronic events. The 1991 dilepton statistics is much lower because only dimuons are tagged, the reconstruction efficiency is lower and the number of hadronic events collected in 1991 is lower.

	1991			1992		
	Like sign	Unlike sign	Total	Like sign	Unlike sign	Total
Simulation	127	221	348	1392	1752	3144
Data	86	170	256	1323	1872	3195

Table 6.7: *The number of like sign, unlike sign and total dileptons for the 1991 and 1992 simulation and data samples.*

In total absence of background, it was shown (equation 1.31 in section 1.2.1) that R, the ratio of the like sign over the the total number of dileptons, relates to χ as

$$\chi = \frac{1-\sqrt{1-2R}}{2} \tag{6.8}$$

In practice it is impossible to purify the dilepton sample to that point and not to loose an unacceptable amount of efficiency. One has to handle several dilepton categories, whose sensitivity to the mixing parameter differs. The following dilepton categories have to be taken into account:

- PB-PB, PB-$\overline{\text{SC}}$ and $\overline{\text{SC}}$-$\overline{\text{SC}}$

  These dileptons form the signal of the mixing measurement and have a like sign probability of $2\chi(1-\chi)$. The $\overline{\text{SC}}$ component is small and therefore the PB-PB category will in the following include the PB-$\overline{\text{SC}}$ and $\overline{\text{SC}}$-$\overline{\text{SC}}$ dileptons. PB-PB contributes to the total like sign fraction with a fraction designated as f_{11}.

- SC-SC

  These dileptons have also a like sign probability of $2\chi(1-\chi)$ and their fraction is indicated by f_{22}.

- PB-SC and $\overline{\text{SC}}$-SC

  These dileptons have a like sign probability of $\chi^2+(1-\chi)^2$ and their fraction is written as f_{12}.

- PC-PC

  Not sensitive to mixing, because they are unlike sign only. Their fraction is indicated by f_{33}.

- Background.

  All other dileptons involving a lepton not originating from the sources mentioned above. They do not contribute equally to the like and unlike sign samples as some memory of the original quark remains, but with a *background charge correlation* quantified by ξ.

From the simulation it was determined that

$$\xi = 0.45 \pm 0.02 \tag{6.9}$$

independent of p_t cuts.

The background dileptons are divided into the categories PB-BCK, SC-BCK, PC-BCK and BCK-BCK with fractions f_{14}, f_{24}, f_{34} and f_{44}, respectively.

Table 6.8 displays the contribution of the different dilepton categories to the total like sign fraction. Consequently, R relates to χ as:

$$R = 2(f_{11} - f_{12} + f_{22})\chi - 2(f_{11} - f_{12} + f_{22})\chi^2 + (f_{14} - f_{24} + f_{34} + f_{44})\xi + f_{12} + f_{24} \tag{6.10}$$

	PB or $\overline{SC}$	SC	PC	BCK
PB or $\overline{SC}$	$2\chi(1-\chi)f_{11}$	$(\chi^2 + (1-\chi)^2)f_{12}$	–	$f_{14}\xi$
SC		$2\chi(1-\chi)f_{22}$	–	$f_{24}(1-\xi)$
PC	–	–	0	$f_{34}\xi$
BCK				$f_{44}\xi$

Table 6.8: *Contributions to the like sign fraction from the different dilepton categories.*

The variable p_t^{min} is defined as the smallest of the two p_t's in a dilepton event. The contributions of the various dilepton categories to the total dilepton sample, before and after $p_t^{min} > 1.0$ *GeV/c* are shown in table 6.9 for the 1991 simulation and in table 6.10 for the 1992 simulation.

The 1991 sample composition differs from the 1992 sample composition. The 1991 dileptons contain more high p_t background. This results in a significantly bigger contribution of the PB-BCK category after $p_t^{min} > 1.0$ *GeV/c*. The BCK jets are misidentified pions and kaons but also leptons from light hadron decays. In 1991 the track quality is lower due to association problems.

Figures 6.2(a)-(d) compare the 1992 simulation to the 1992 data. They show the distributions of the lepton variables p, p_t, p_t^{min} and p_t^{max} in the 1992 dilepton sample. The variable p_t^{max} is defined as the largest of the two p_t's in the dilepton. The figures indicate that the distributions of the simulation and the data agree well.

Dilepton	No p_t cut	$p_t^{min} > 1.0$ GeV/c
	$\mu\mu$	$\mu\mu$
PB-PB	26.7±2.4	42.6±4.2
PB-SC	25.6±2.3	26.5±3.8
SC-SC	3.2±1.0	1.5±1.0
PC-PC	10.3±1.6	5.1±1.9
PB-BCK	15.5±1.9	16.2±3.2
SC-BCK	2.6±0.9	0.7±0.7
PC-BCK	8.3±1.5	5.9±2.0
BCK-BCK	7.8±1.4	1.5±1.0

Table 6.9: *Dimuon fractions in the 1991 simulation, before and after* $p_t^{min} > 1.0$ GeV/c.

Dilepton	No p_t cut				$p_t^{min} > 1.0$ GeV/c			
	ee	$e\mu$	$\mu\mu$	all	ee	$e\mu$	$\mu\mu$	all
PB-PB	20.3±1.6	24.3±1.1	26.1±1.3	24.1±0.8	62.0±4.3	64.1±2.7	60.9±2.8	62.5±1.8
PB-SC	15.0±1.4	19.5±1.1	25.1±1.3	20.5±0.7	16.3±3.3	14.4±1.9	23.5±2.5	18.3±1.4
SC-SC	3.3±0.7	3.3±0.5	5.0±0.7	3.9±0.3	3.1±1.5	2.5±0.9	1.4±0.7	2.1±0.5
PC-PC	6.1±1.0	6.9±0.7	10.2±0.9	7.8±0.5	3.1±1.5	2.1±0.8	2.4±0.9	2.4±0.6
PB-BCK	18.3±1.5	17.0±1.0	12.8±1.0	15.8±0.7	10.1±2.7	12.9±1.9	7.5±1.5	10.3±1.1
SC-BCK	6.4±1.0	7.4±0.7	5.8±0.7	6.6±0.4	0.8±0.8	1.8±0.7	0.7±0.5	1.2±0.4
PC-BCK	8.2±1.1	8.1±0.7	7.2±0.8	7.8±0.5	1.6±1.1	1.5±0.7	3.4±1.1	2.3±0.5
BCK-BCK	22.4±1.6	13.6±0.9	7.8±0.8	13.5±0.6	3.1±1.5	0.6±0.4	0.3±0.3	0.9±0.4

Table 6.10: *Dilepton fractions in the 1992 simulation, before and after* $p_t^{min} > 1.0$ GeV/c.

Stellingen

behorende bij het proefschrift

"A Measurement of the B^0-$\bar{B}^0$ Mixing Parameter at LEP Using a Neural Network"

1. Het is af te raden een neuraal netwerk eenvoudigweg alle mogelijk gevoelige variabelen van een classificatieprobleem aan te bieden. Hoewel theoretisch de beste separatie bereikt kan worden, maakt de vereiste grootte van de trainingset en de benodigde rekentijd voor de training dit onpraktisch.

   Dit proefschrift.

2. De klassieke mechanica is niet alleen een limietgeval van de quantummechanica, maar is ook noodzakelijk voor haar formulering. Dit is niet het geval voor de modellen van De Broglie/Bohm en Everett/DeWitt. Deze modellen zijn echter filosofisch onbevredigend.

   John Bell, "Against Measurement", Physics World, August 1990

3. Het gebruik van Internet als een gedistribueerde elektronische bibliotheek maakt deel uit van de informatie revolutie. De huidige ontwikkeling heeft twee nadelen. Ten eerste wordt de vooruitgang tot nu toe gestimuleerd door technologie in plaats van door de markt. Verder bevat de gemiddelde WWW pagina te veel informatie over informatie.

4. Het Rivest-Shamir-Adleman (RSA) publieke sleutel systeem is gebaseerd op het feit dat het moeilijk is een getal te factoriseren dat het produkt is van twee grote priemgetallen. Snellere hardware maakt cryptografie met RSA veiliger in plaats van onveiliger. Overigens zou een onverwachte doorbraak in factoriserings-algorithmes onmiddellijk de veiligheid van vele commerciële soft- en hardware produkten en van elektronische transacties in groot gevaar brengen.

5. De Polymerase Chain Reaction (PCR) methode voor de verdubbeling van een specifiek stuk DNA is zeer gevoelig en specifiek; tevens hoeft men voor het gebruik van deze methode slechts de exacte DNA sequenties die beide zijden van het stuk flankeren te kennen. PCR heeft vele medische en forensische toepassingen, en heeft bovendien 'Jurassic Park' dichterbij gebracht dan ooit.

   Raúl J. Cano and Monica K. Borucki, "Revival and identification of bacterial spores in 25- to 40-million-year-old Dominican amber", Science, vol. 268, May 1995

6. Het stemt tot enige ongerustheid dat slechts een kleine groep wetenschappers voor onze beschaving waakt door systematisch asteroïden te klassificeren die mogelijk in de buurt van de aarde komen. Een inslag zoals die van de 50 m diameter Tunguska asteroïde in 1908, die het effect van een grote atoombom had, komt eens in de 300 jaar ergens op aarde voor. Inslagen door grotere asteroïden (> 100 m diameter) zijn veel zeldzamer, maar zij zijn wel het enige natuurverschijnsel dat een bedreiging vormt voor de gehele mensheid.

   David Morrison, "Target: Earth!", Astronomy, October 1995

7. Elementaire logica is niet eenvoudig. Met behulp van de basis axioma's $(p \vee p) \supset p$, $p \supset (p \vee q)$, $(p \vee q) \supset (q \vee p)$ en $(p \supset q) \supset ((r \vee p) \supset (r \vee q))$ kan men het volgende theorema afleiden:

   $((p \supset q) \supset ((r \supset s) \supset t)) \supset ((u \supset ((r \supset s) \supset t)) \supset ((p \supset u) \supset (s \supset t)))$

   Ernest Nagel and James R. Newman, "Gödels proof", ISBN 0-415-04040-X

8. Gedurende droog weer in Amsterdam kunnen contactlenzendragers zich tegen stof weren door de volgende regels in acht te nemen, die alle met de tram verband houden:

   (a) Vermijd zoveel mogelijk het fietsen door straten met tramrails.

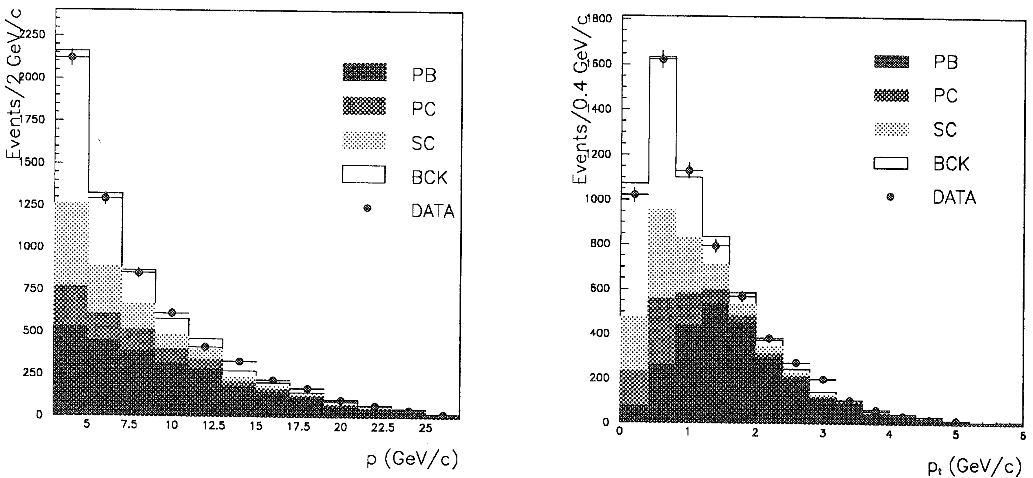
   (b) Wanneer er een tram passeert sluit de ogen gedeeltelijk om de werkzame doorsnede voor stof te verminderen.

   (c) Bij het verlaten van de tram *nooit* tegen de rijrichting in kijken.

9. Vijfennegentig procent van de vergiftigingen met dodelijke afloop door paddestoelen wordt veroorzaakt door groene of kleverige knol-amanieten die voor champignons worden aangezien. Niet alle eetbare padddestoelen zijn onder alle condities eetbaar; zo is de kale inktzwam bij gelijktijdig gebruik van alcohol giftig.

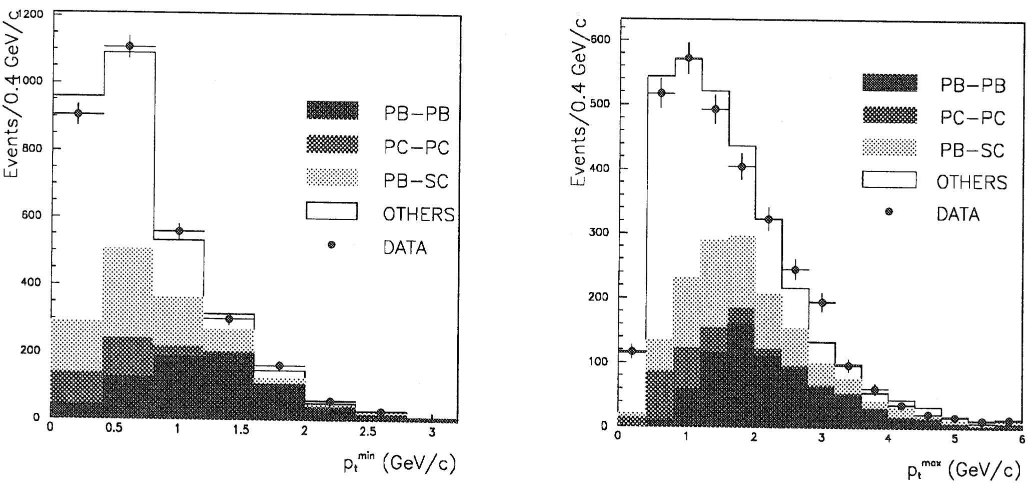
   Till R. Lohmeyer, "Paddestoelen", ISBN 9064100314

Martin Ernst Los, Amsterdam, 27 november 1995



(a) Lepton momentum p.

(b) Lepton transverse momentum p_t



(c) Smallest transverse momentum p_t^{min}.

(d) Largest transverse momentum p_t^{max}.

Figure 6.2: *Momenta and transverse momenta distributions in the dilepton sample for the 1992 data (dots) and simulation (histograms). The contributions of the different (di)lepton classes are shown.*

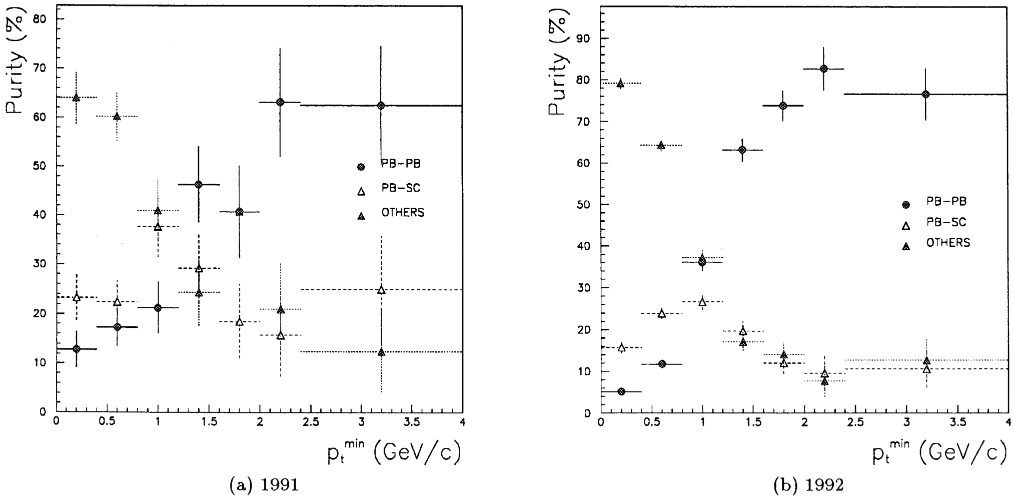
6.6 The Neural Network Classifier

In a mixing measurement using dileptons, the standard method to purify the sample is to exploit a variable composed from the lepton momenta from both jets. Examples of variables used in the past are:

- p_t^{min}, by ALEPH[91], DELPHI[92] and L3[94]
- $p_{dil} = \sqrt{\frac{1}{16}|\mathbf{p}_1 \times \mathbf{p}_2| + p_t^2}$, by DELPHI[93]
- $p_{comb}^{min} = \sqrt{(p/10)^2 + p_t^2}$, by OPAL[95]

It was observed that all the above variables have practically the same separation power. Therefore, the most recent measurements employ p_t^{min}, the simplest one.

In order to remove a major part of the background, transverse momentum cuts must be fairly high, $p_t^{min} > 1.0\ GeV/c$ is a common choice, to ensure a workable PB-PB purity. Most backgrounds are already significantly reduced with a lower cut, it is the trade-off between PB and SC at the single jet level which determines the optimal p_t^{min} cut. Figure 6.3 shows dilepton composition as a function of p_t^{min} for the components PB-PB, PB-SC and the other backgrounds; 1991 contains significantly more background.



(a) 1991 (b) 1992

Figure 6.3: *Dilepton sample composition for the 1991 and 1992 simulation as a function of p_t^{min}. The contribution PB-SC, which dilutes the χ sensitivity is shown separately.*

Furthermore, the secondary decays contribute significantly to the systematic error of a mixing measurement through the uncertainty in the branching ratio BR($b \to c \to l$).

With the above arguments in mind, it was decided to develop a neural network to suppress the SC decays and keep the PB decays on a single jet basis. The aim was to construct another variable that is used in the same manner as p_t^{min}, but separates PB-PB and PB-SC better and of course still suppresses the other background.

An earlier study[96] using a neural network on 1991 data showed that it was possible to separate $b \to c \to \mu^+$ jets from $b \to \mu^-$ without directly using the p_t of the lepton. This neural network classifier used 14 kinematical variables, amongst them the momenta components p_x, p_y and p_z of the four leading tracks in the jet. Although the muon track is likely to be among these four leading tracks, the network did not know which (p_x,p_y,p_z)-set it processed belonged to the lepton. Unfortunately, this network was not powerful enough to use it in a mixing measurement. This was due to the high p_t background in the 1991 data in combination with the poor separation power of what was analysed to be an 'average p_t' over the four leading tracks.

After telling the network what track was the muon, and after studying the way the network correlated the momentum components, it was found that it actually constructed the p_t of the lepton and mainly used that for the separation. Its separation power was about equal to using p_t directly, because too much effort went into constructing the p_t from the individual momentum components; the other variables were not optimally exploited. Furthermore, the high dimensional input space required a large training set, a condition which could barely be satisfied.

Therefore, it was decided to devise a much smaller network to which the momentum and transverse momentum of the lepton are provided directly. The above studies led to the following choice of 4 input variables:

1. The total jet momentum p_{jet}.

   The neural network does not employ this variable for the separation, but for the scaling of the other variables.

2. The momentum p of the lepton.

   Primary leptons from beauty decay have on average larger momenta.

3. The transverse momentum p_t of the lepton.

   This is the standard variable to enhance the PB purity in a lepton tagged sample, because direct beauty decays have on average larger transverse momenta than the others.

4. $|Q_{jet}|$, the absolute value of the *jet charge* $Q_{jet} \in [-1,1]$, calculated by:

   $$Q_{jet} \equiv \frac{\sum_i P_i q_i}{\sum_i P_i} \tag{6.11}$$

   where the index i runs over all charged tracks in the jet, P_i is the momentum of the track projected on the sphericity axis of the event and q_i (± 1) the charge of the track. The *sphericity axis* plays the role of the event axis and is the first eigenvector of the sphericity tensor, defined by:

   $$S^{\alpha\beta} = \frac{\sum_i p_i^\alpha p_i^\beta}{\sum_i |\mathbf{p}_i|^2} \tag{6.12}$$

   where i runs over all tracks in the event, p_i are the track momenta and $\alpha, \beta = 1,2,3$ correspond to the x y and z components. For more information about the sphericity axis see[97].

   The jet charges for secondary decays, kaon/pion decays and misidentified hadrons are on average small. For primary charm decays, $|Q_{jet}|$ is on average larger and it is on average the largest for primary beauty decays. Figure 6.4 shows $|Q_{jet}|$ for $c \to l$ (dashed) and for the other backgrounds (solid); figures 6.5(e) and 6.5(f) show $|Q_{jet}|$ for $b \to l$ and $b \to c \to l$.

The training set consisted of about 6000 single jets with a lepton, taken from special 1992 simulation samples: 1500 $b \to \mu$, 1500 $b \to c \to \mu$, 1500 $b \to e$ and 1500 $b \to c \to e$. Figure 6.5 shows the distributions of the training variables of the training set. $b \to l$ jets were given a target value of 0.05 and $b \to c \to l$ jets a target value of 0.95.

During training the performance of the network was constantly monitored using an independent test set with about the same composition as the training set. The training was stopped when the error on the test set didn't change any more. Table 6.11 summarises the parameters of the training.

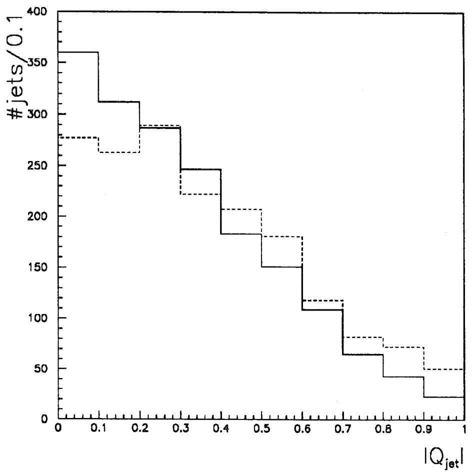
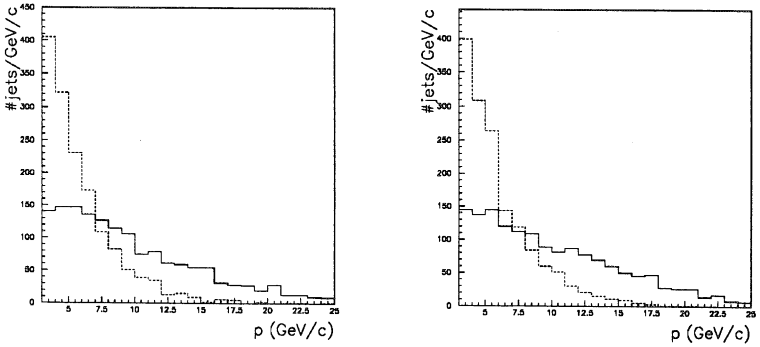


Figure 6.4: *$|Q_{jet}|$ for $c \rightarrow l^+$ (dashed) and for the other backgrounds (solid).*

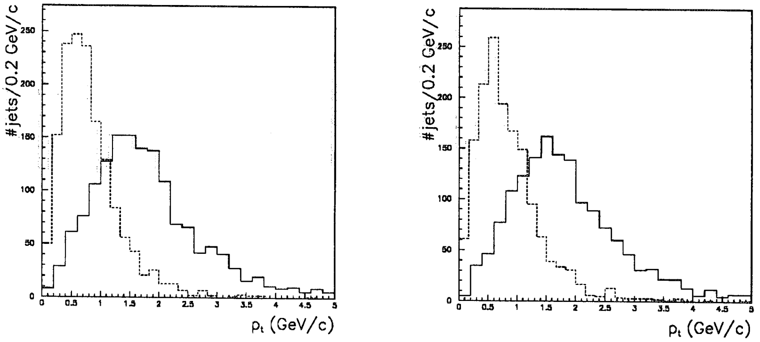
Parameter	Setting
initial learning rate η	0.03
decay rate of η	5%
lower bound of η	0.001
initial moment strength α	0.5
increase rate of α	1%
upper bound of α	0.9
temperature	1.0
initial weights	$[-0.1, 0.1]$
iterations	400

Table 6.11: *Settings of the neural network training parameters.*



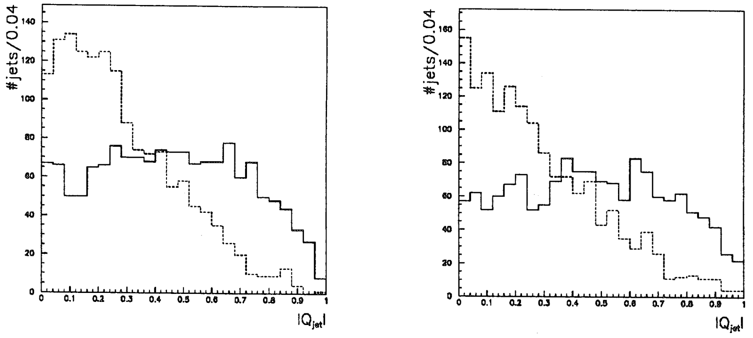
(a) p electrons.

(b) p muons.



(c) p_t electrons.

(d) p_t muons.



(e) $|Q_{jet}|$ electrons.

(f) $|Q_{jet}|$ muons.

Figure 6.5: *The distributions of the momentum p, the transverse momentum p_t and the absolute value of the jet charge $|Q_{jet}|$ for $b \to l^-$ (solid) and $b \to c \to l^+$ (dashed). These samples formed the training sets for the neural network.*

The performance of the neural network classifier in terms of its ability to separate event classes is quantified with the *purity* and the *efficiency*. After tagging, the purity of the sample with respect to events of some class X, is defined by:

$$\text{Purity(X)} \equiv \frac{\text{Number of class X events in the tagged sample}}{\text{Total number of events in the tagged sample}} \tag{6.13}$$

The efficiency for selecting events of class X is defined by:

$$\text{Efficiency(X)} \equiv \frac{\text{Number of class X events in the tagged sample}}{\text{Number of class X events presented to the classifier}} \tag{6.14}$$

The output distribution of the neural network on the independent test set is shown in figure 6.6. The $b \rightarrow l$ jets are on the left and the $b \rightarrow c \rightarrow l$ jets are on the right.

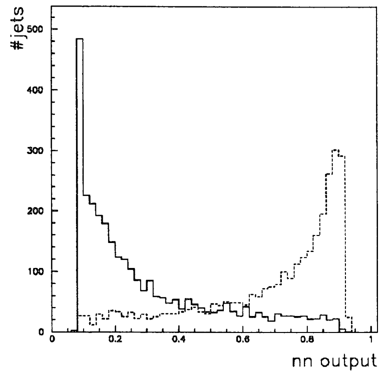


Figure 6.6: *Neural network output distribution for $b \rightarrow l^-$ (solid) and $b \rightarrow c \rightarrow l^+$ (dashed).*

The performance of the neural network on the test set is quantified in terms of Efficiency(PB) and Purity(PB) as a function of a cut in the neural network output (*nn output*). Figure 6.7 compares the neural network performance on the test set to that of the p_t. The neural network curve lies above the p_t curve. A cut of $p_t > 1.0\ GeV/c$ corresponds to a neural network cut of *nn output* < 0.5; at which point the neural network offers more than 6% improvement in purity at the same efficiency.

The network can now be applied to the dilepton samples. The jet that contains the lepton with the highest p_t is labelled jet 1, the other jet is labelled jet 2. Figures 6.8(a)-(d) compare the output distribution of the neural network applied to jet 1 and jet 2, for the 1991 and 1992 simulation and data. In both cases there is agreement between the output distributions of the simulation and the data.

It seems that the 1991 data contains more PB jets then the 1992 data. The network mistakes the BCK jets with higher p_t's for PB jets: they invoke lower values of *nnout*.

If the network is to be used the same way as p_t^{min}, the neural network cut has to be applied separately to both jets of each dilepton event. In the following this variable will be designated by nn^{min}. Contrary to p_t, the neural network output is *smaller* when a jet is more likely to be a PB jet. Therefore, nn^{min} is defined by:

$$nn^{min} = min(1.0 - \textit{nn output}\text{ for jet }1, 1.0 - \textit{nn output}\text{ for jet }2) \tag{6.15}$$

in order to facilitate the comparison of the p_t^{min} analysis with the neural network analysis.

The result of requiring $nn^{min} > 0.5$ on the 1991 and 1992 simulation is summarised in table 6.12. This table has to be compared with tables 6.9 and 6.10, the corresponding tables for the sample composition after the p_t^{min} cut. The network enhances PB-PB and suppresses PB-SC and SC-SC better. PC-PC is less well rejected because PC has a higher average jet charge than the other backgrounds. This is no problem because PC-PC is not sensitive to mixing.

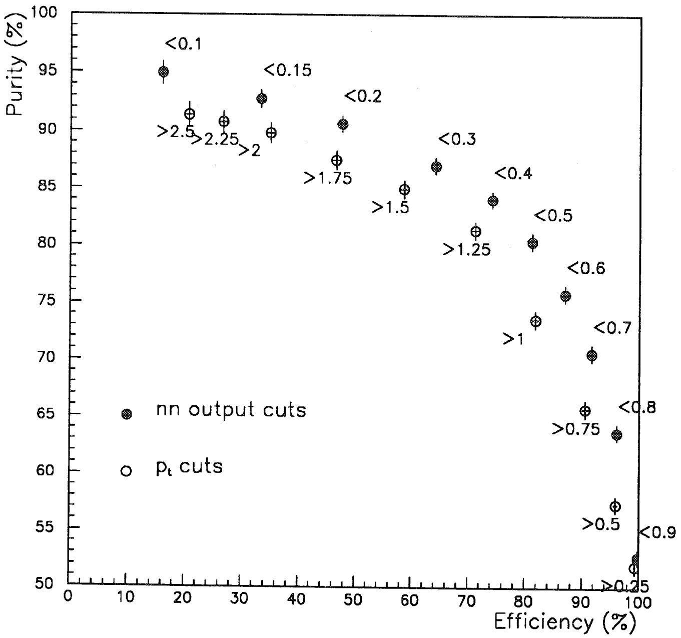
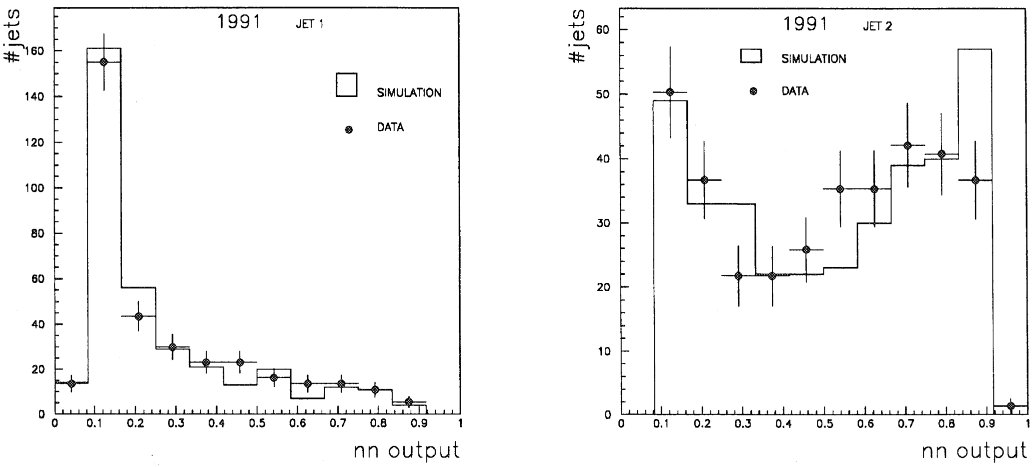


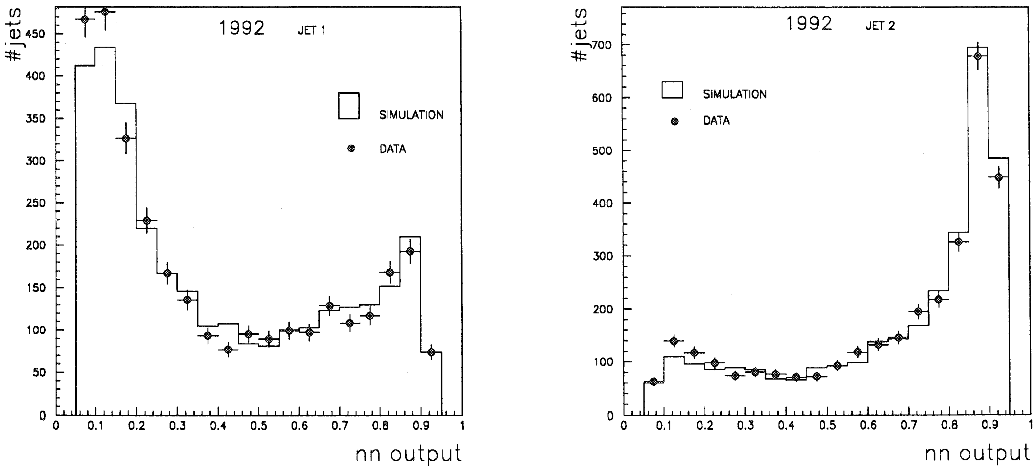
Figure 6.7: *Efficiency versus purity for the test set to select $b \rightarrow l^-$ jets applying neural network cuts or p_t cuts. The test set consists of about 6000 $b \rightarrow l^-$ jets and 6000 $b \rightarrow c \rightarrow l^+$ jets.*

Dilepton	1991	1992			
	$\mu\mu$	ee	$e\mu$	$\mu\mu$	all
PB-PB	46.0±4.1	66.7±4.5	64.8±2.8	67.0±2.8	66.0±1.8
PB-SC	25.0±3.6	12.6±3.2	14.3±2.0	17.0±2.2	15.1±1.4
SC-SC	0.6±0.6	1.8±1.3	1.0±0.6	0.4±0.4	0.9±0.4
PC-PC	6.8±2.1	3.6±1.8	3.3±1.0	4.6±1.2	3.9±0.7
PB-BCK	15.5±3.0	9.9±2.8	12.3±1.9	7.4±1.6	9.9±1.1
SC-BCK	0.7±0.7	1.8±1.3	1.3±0.7	0.4±0.4	1.0±0.4
PC-BCK	4.1±1.6	0.9±0.9	1.7±0.7	2.8±1.0	2.0±0.5
BCK-BCK	1.4±0.9	2.7±1.5	1.3±0.7	0.4±0.4	1.2±0.4

Table 6.12: *Dilepton composition of the 1991 and 1992 simulation after the neural network cut* $nn^{min} > 0.5$.



(a) 1991 jet 1. (b) 1991 jet 2.



(c) 1992 jet 1. (d) 1992 jet 2.

Figure 6.8: *The output distribution of the neural network on single jets for the simulation and the data.*

6.7 Extraction of χ

A first consistency check is the mixing parameter that can be directly calculated from the simulation. If one simply determines R for the PB-PB category, then R and χ are related according to equation (6.8). In the 1991 and 1992 simulation there are 207 like sign and 643 unlike sign PB-PB dileptons and one finds:

$$\chi_{sim} = (14.2 \pm 1.0)\% \tag{6.16}$$

This value is consistent with $\chi = 13.4\%$, the value which was put in the simulation.

The *counting method*, uses the full information contained in the simulation. It works by solving equation (6.10) after a cut that enhances the PB-PB component so that the sensitivity of R to χ is increased. For 1991 this goal can not be achieved: even at high p_t (see figure 6.3(a)) too much background remains.

If one relies on the sample composition after $p_t^{min} > 1.0\ GeV/c$ given in table 6.10 with ξ quoted in (6.9), then solving (6.10) yields for the 1992 data:

$$\chi_{1992\ counting\ p_t} = (9.0 \pm 2.8)\% \tag{6.17}$$

The counting method can also be applied to the data after the neural network cut $nn^{min} > 0.5$ (sample composition in table 6.12). Similarly, the neural network does not give a result for 1991 data. For the 1992 data one finds:

$$\chi_{1992\ counting\ nn} = (8.5 \pm 2.5)\% \tag{6.18}$$

The two results of the counting method are compatible with each other. The statistical error in these and the following results is composed of two parts: that due to the number of events and that due to the statistical errors in the fractions calculated with a finite simulation sample. Both contributions are comparable.

Finally the data sample of 1992 was used to calculate R as a function of p_t^{min} and as a function of nn^{min}. These distributions were compared with those predicted by the simulation as a function of χ, and the χ value determined using a least squares fit. This procedure was not followed for the 1991 data because the result is very sensitive to small changes in the fractions.

The ξ value given in (6.9) was not directly used in the fit, instead ξ was parametrised as a function of p_t^{min} per bin and as a function of nn^{min} per bin. Another possibility is to leave ξ as a free parameter in the fit.

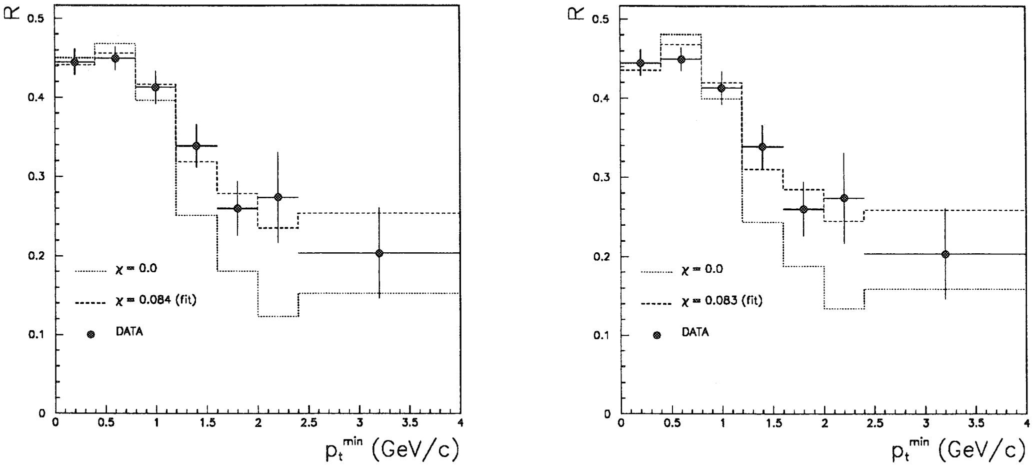
The result of the fit was found to be stable with respect to changes in the binning. Furthermore, the counting method and the fit method applied to the simulation gave $\chi \simeq 14\%$, consistent with the input value, both for p_t^{min} and nn^{min}.

Table 6.13 summarises the results of the fits that were performed. Figures 6.9(a)-(d) show the corresponding $R(p_t^{min})$ and $R(nn^{min})$ plots. In each case, the result of the fit is shown together with the curve for no mixing ($\chi = 0.0$).

The fit results for χ are consistent with one another and with the results of the counting method. The value of ξ obtained after leaving it as a free parameter in the fit is also consistent with the one determined from simulation.

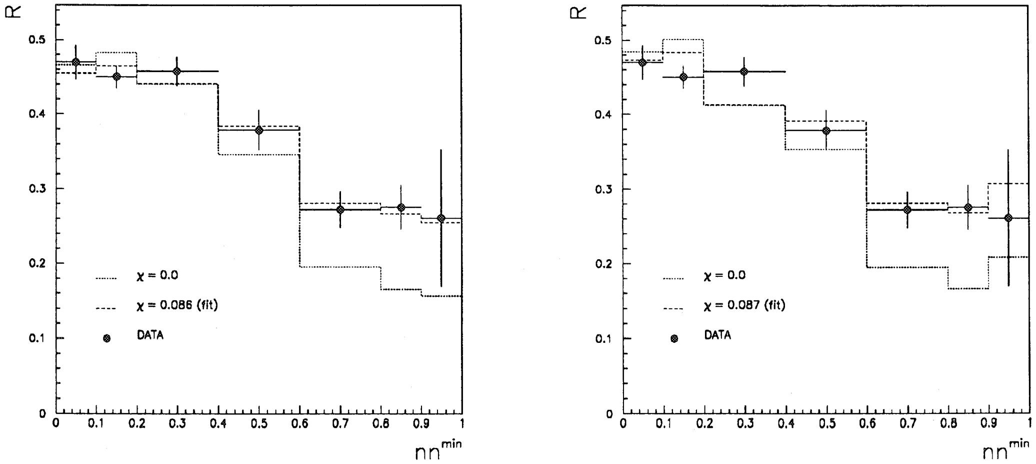
	$R(p_t^{min})$ fit	$R(nn^{min})$ fit
ξ free	$\chi = 8.4 \pm 2.4$ figure 6.9(a) $\xi = 0.44 \pm 0.04$	$\chi = 8.6 \pm 2.3$ figure 6.9(c) $\xi = 0.44 \pm 0.04$
ξ parametrised	$\chi = 8.3 \pm 2.4$ figure 6.9(b) $\xi(p_t^{min})$	$\chi = 8.7 \pm 2.3$ figure 6.9(d) $\xi(nn^{min})$

Table 6.13: *The results for χ (in percent) of the various fits for 1992.*



(a) The ratio R as a function of p_t^{min}.
ξ is left as a free parameter in the fit.

(b) The ratio R as a function of p_t^{min}.
ξ is parametrised as a function of p_t^{min}.



(c) The ratio R as a function of nn^{min}.
ξ is left as a free parameter in the fit.

(d) The ratio R as a function of nn^{min}.
ξ is parametrised as a function of nn^{min}.

Figure 6.9: *Plots of $R(p_t^{min})$ and $R(nn^{min})$ belonging to the various fits. The data points are shown together with the fitted values and their corresponding χ (dashed line). Also shown in each case are the values corresponding to $\chi = 0.0$ (dotted line).*

6.7.1 Systematic Errors

There are several sources of systematic errors intrinsic to the simulation used to estimate the background. The most important ones are shown in table 6.14, together with their effects on χ.

Most sources are estimated from DELPHI's 1994 analysis[92]. The contribution of the neural network to the systematic error is found by studying the effect of ±5% changes to the individual input variables p_{jet}, p, p_t and $|Q_{jet}|$. For all variables the effect on the neural network output was found to be less than 0.004, which has almost no effect on χ.

The branching ratios Br($b \to l$) and Br($b \to c \to l$) are correlated. Their ratio largely dominates the systematic error of the mixing measurement. The total systematic error 0.4% was obtained by adding the contributions in quadrature.

The differences between the models of semi-leptonic decay also contribute to the systematic error. These models predict the lepton momentum distribution in the rest frame of the B meson.

In DELPHI's 1994 analysis the ACCMM[98] and IGSW[99] models were considered. ACCMM predicts the inclusive $b \to l$ spectrum, while the IGSW spectrum depends on the relative production rates of D, D^* and D^{**}. In the DELPHI analysis a total systematic error of 0.4% was assigned to account for model dependencies. In the measurement described in this thesis, the same model error is assumed.

Source	Relative variation	Effect on χ
Br($b \to l$), Br($b \to c \to l$)[92]	±4%, ±5%	∓0.4%
Br($b \to \bar{c} \to l$)[92]	±50%	<0.05%
Br($b \to \tau$)[100]	± 23%	<0.05%
Br($c \to l$)[92]	±10%	<0.05%
Hadron misid.[92]	±10%	<0.05%
Fragmentation function[92]	$\epsilon_b = 0.005 - 0.007$	0.1%
Neural Network	±5%	<0.05%
Total		0.4%
Semi-leptonic decay model		0.4%

Table 6.14: *Contributions to the systematic uncertainty in the measurement of χ. Variations given in percent are relative to the values in the simulation.*

6.8 Final Result, Comparisons and Conclusions

The B^0-$\bar{B}^0$ mixing parameter χ has been determined using two different methods: a common p_t^{min} based analysis and an analysis using the variable nn^{min} constructed with a neural network.

A comparison has been made between using p_t^{min} or nn^{min} to separate the signal from the background. The variable nn^{min} substantially enhances the signal to background ratio as shown in figure 6.7 and the comparison of table 6.12 with table 6.10.

For both methods, χ has been measured with the counting method and with a fit to R, the ratio of like sign over unlike sign dileptons, as function of the separating variable. The fits have been performed with the background charge correlation ξ left free or with ξ parametrised as a function of the separating variable. The main results are listed in table 6.15.

The two methods are not independent, they use the same data sample. As the final result is taken:

$$\chi_{nn\,fit,\,\xi\,free} = (8.6 \pm 2.3(stat) \pm 0.4(sys) \pm 0.4(model))\% \tag{6.19}$$

From the correlation matrix in the fit, it was verified that the correlation between χ and ξ is small.

The statistical error of the result is dominant. This is due to the limited statistics of the simulation sample used to estimate the composition of the data and the limited amount of events in the data sample.

Table 6.16 compares the measurement of this thesis with χ measured by other experiments. The results of this work can not compete with those other experiments not withstanding the good performance of the neural network method. This is due to the smaller data sample and the smaller simulation sample used to calculate the fractions. The result of this thesis is not independent of the result of DELPHI[92] and therefore can not be used in a final averaging of all results.

A possible improvement of the neural network method lies in using four outputs: one for PB, one for SC, one for PC and one for the other background. Then an optimal separation might be achieved between the single jet classes PB, SC, PC and BCK.

Method	χ ± (stat) (%)		
	Counting	Fit	
		ξ free	ξ parametrised
p_t^{min}	9.0 ± 2.8	8.4 ± 2.4	8.3 ± 2.4
nn^{min}	8.5 ± 2.5	8.6 ± 2.3	8.7 ± 2.3

Table 6.15: *Summary of the results of the two different methods to determine χ.*

Experiment	χ value (%)	Dilepton Statistics
This thesis	8.6 ± 2.3 (stat) ±0.4 (sys) ±0.4 (model)	673 ee, 1414 $e\mu$, 1108 $\mu\mu$
DELPHI[92]	12.1 ± 1.6(stat) ±0.4(sys) ±0.4(model)	658 ee, 2175 $e\mu$, 2349 $\mu\mu$
DELPHI[101]	14.4 ± 1.4(stat) $^{+1.7}_{-1.1}$(sys)	46497 events high p_t $\mu - Q_{jet}^{opp}$
L3[94]	12.3 ± 1.2(stat) ±0.8(sys+model)	216 ee, 766 $e\mu$, 857 $\mu\mu$ ($p_t^{min} > 1.0\ GeV/c$)
OPAL[95]	14.3 ± 2.2(stat) ±0.7(sys)	882 ee, 1787 $e\mu$, 1163 $\mu\mu$
ALEPH[91]	11.3 ± 1.5(stat) ±0.8(sys) ±0.7(model)	710 dileptons ($p_t > 1.25\ GeV/c$)

Table 6.16: *Comparison with other results. The χ values are in percent.*

Appendix A

The Back-propagation Algorithm

For all patterns p in the training set and each output node i, the difference δ_{pi} between its target value t_{pi} and its output y_{pi} is defined by:

$$\delta_{pi} = (y_{pi} - t_{pi}) \tag{A.1}$$

At any moment a global performance measure of the network is the summed squared error:

$$E = \sum_p E_p = \sum_p \frac{1}{2} \sum_i \delta_{pi}^2 \tag{A.2}$$

All nodes calculate their activity by a weighted sum of their incoming lines, which are the outputs y_{pj} of nodes in the previous layer. For an output node i, the activity a_{pi} is:

$$a_{pi} = \sum_j w_{ij} y_{pj} \tag{A.3}$$

where w_{ij} is the weight between node i and node j.

Nodes apply a threshold function g to their activity that determines their final output. Thus, for an output node i:

$$y_{pi} = g(a_{pi}) \tag{A.4}$$

To decrease the mean squared error (A.2), the weights between the nodes i in the output layer and nodes j in the previous layer are changed applying a gradient descent with proportional constant η:

$$\Delta_p w_{ij} = -\eta \frac{\partial E_p}{\partial w_{ij}} \tag{A.5}$$

Using the chain rule in (A.2) with relations (A.1), (A.3) and (A.4), one finds:

$$\frac{\partial E_p}{\partial w_{ij}} = \frac{\partial E_p}{\partial \delta_{pi}} \frac{\partial \delta_{pi}}{\partial y_{pi}} \frac{\partial y_{pi}}{\partial a_{pi}} \frac{\partial a_{pi}}{\partial w_{ij}} = \delta_{pi} g'(a_{pi}) y_{pj} \tag{A.6}$$

If i does not refer to a node in the output layer, the error signal δ_{pi} in (A.6) becomes (A.1) propagated down to the lower layer in question using the weights of the higher levels. So, the back propagation update rule is:

$$\Delta_p w_{ij} = -\eta \delta_{pi} g'(a_{pi}) y_{pj} \tag{A.7}$$

with

$$\delta_{pi} = \begin{cases} (y_{pi} - t_{pi}) & \text{if } i \text{ output node} \\ \sum_k w_{ik} \delta_{pk} & \text{otherwise} \end{cases} \tag{A.8}$$

Appendix B

The Neural Network Simulator NNC

This appendix is a short manual for the neural network simulator NNC. The simulator is written in the programming language C and implements the back propagation learning scheme[20] for arbitrary connected feed forward networks with up to four layers.

B.1 Installation and Running

To install NNC (on UNIX systems), undertake the following:

1. Copy the standard version 4.50 of the simulator from */nikhefh/user/i14/neuralnets/nnc/nn450* to your directory. If you don't have access to the NIKHEFH computer, contact the author of this thesis to get a copy.

   NNC consists of the source code *nn.c* and the example input files *xor.par, xor.nnd, xort.nnd* and *xor.str*.
2. Compile and link with *cc nn.c -o nnc -lm -O*, which creates the optimised executable *nnc*.
3. Test the neural network with *nnc xor*, which solves the exclusive-or (XOR) problem (see section B.2) represented by the example files.

When NNC has finished the following output files have been created:

- *xor.net*, containing the weights that have been developed in the network during the training phase.
- *xor.ntp*, containing the response on the test set.

Using the XOR problem as an example, the formats of the input and output files will be described in the following sections.

B.2 Data Files, extension .nnd

The data files of NNC are the training set and the test set. Their format starts with the size of the set followed by that number of data blocks and ends with an end-of-file identifier.

Each data block starts with two (integer) identifiers, followed by the input variables and their target value(s) in floating point format. In a high energy physics application the identifiers can for example be the run and event number of the data in this block.

The XOR problem represented by the example files is a 2-bit parity problem which has the truth table shown in table B.1. The training set for the XOR problem is displayed explicitly in table B.2.

	0	1
0	0	1
1	1	0

Table B.1: *Truth table for XOR.*

```
4
1   1
0.0 0.0 0.0
1   2
1.0 0.0 1.0
1   3
0.0 1.0 1.0
1   4
1.0 1.0 0.0
99999.8888
```

Table B.2: *The training set xor.nnd.*

B.3 Parameter File, extension .par

The parameter file steers the operation of the neural network package. It defines the network topology, the learning parameters and the filenames for input and output.

An example parameter file for the XOR problem is shown in table B.3 and contains a short explanation of each parameter. For some parameters a more elaborate description is given in the sections below.

B.3.1 Operation Type

The variable TYPE in table B.3 can have values in the range 0-3 and controls the general operation of NNC.

If TYPE=0, the standard training mode of NNC is selected. First, the weights are initialised with random values in the range bounded by LWEIGHTS and RWEIGHTS. Then, the training set is loaded into memory and training takes place until the number of iterations exceeds ITERS or the error on the test set is smaller than ERROR. During the training the current values of the errors and learning parameters are printed every EVERY iterations. After that, the weights of the network and the response of this network on the test set is saved.

If TYPE=1 then the weights are not initialised at random, but a previously saved network is loaded and its response on the indicated test set is saved.

If TYPE=2, the network behaves like TYPE=0, apart from the last step, i.e. the response on the test set is not saved. This is useful if one doesn't want to overwrite an existing response file or waste disk-space writing an unnecessary one, because training will be resumed at a later moment.

TYPE=3 implements the continuation of previously stopped training. It proceeds like TYPE=0, but instead of randomising the weights, they are loaded from the saved network file indicated in the parameter file.

B.3.2 Stop File, nn.stp

The user has to have the possibility to stop the training at any moment in order to continue at a later time or to abort a non-successful training session.

```
XOR-problem             ;name of your session
-----GEOMETRY-------------------------------------------------
NINPUT         2        ;#input nodes
NHIDDEN        2        ;#nodes in second layer
NHIDDEN2       0        ;#nodes in (optional) third layer
NOUTPUT        1        ;#nodes in output layer
STRFILE        xor.str ;structure file, see section B.3.3
-----NN-PARAMETERS--------------------------------------------
ACTIVATION     0        ;use continuous sigmoid, keep at 0
TEMPER         1.0      ;slope of sigmoid or temperature
BIASOUT        1.0      ;bias on hidden and output nodes
LEARN          0.9      ;learning rate
DLEARN         0.0      ;decay of LEARN in %
MINLEARN       0.0      ;minimum learning rate
MOMENT         0.8      ;moment strength
DMOMENT        0.0      ;decay of moment in %
MINMOM         0.0      ;moment minimum
LWEIGHTS      -1.0      ;lower bound initial random weights
HWEIGHTS       1.0      ;upper bound initial random weights
-----OPERATION------------------------------------------------
TYPE           0        ;operation type, see section B.3.1
BINARY         0        ;0=ASCII I/O (1=Binary I/O)
NNDFILE        xor.nnd ;training set, see section B.2
NSETS          4        ;size of training set
CHUNK          4        ;size of training set kept in memory
NETFILE        xor.net ;saved network, see section B.3.4
NTPFILE        xor.ntp ;network response, see section B.3.5
ALLINF         0        ;keep at 0 (obsolete)
INFFILE        dum.inf ;dummy filename (obsolete)
TSTFILE        xort.nnd ;test set, see section B.2
NSETST         4        ;size of test set
CHUNKT         4        ;size of test set kept in memory
ERROR          0.1      ;stop at this error for test set
MINBPERR       0.00     ;minimum update error
ITERS          10000    ;maximum #iterations
EVERY          10       ;print status every 10 iterations
FSTOP          0        ;forced stop flag, see section B.3.2
```

Table B.3: *Parameter file for the XOR problem.*

If the FSTOP flag in the parameter file is set to 1 then training is stopped the moment a file named *nn.stp* is created in the current directory.

B.3.3 Structure File, extension .str

The structure file contains commands to remove connections or add connections starting from the standard feed forward connected network defined in the parameter file.

The structure file uses the following conventions for network topologies:

- Layers are numbered bottom to top, starting with (input) layer number 1.
- Nodes in one layer are numbered left to right, starting with node number 1.

- A specific node is designated by a layer number and a node number, separated by a comma. The bias node is defined as 0,0 (layer 0, node 0) and is by default connected to all nodes but the ones in the input layer.
- A specific connection is designated by the node the connection feeds into, followed by the node where it comes from. Hence in forward connected networks, the node with the higher layer number comes first.
- Layered network topologies are referred to as *#nodes in layer 1 : #nodes in layer 2 : ...* For example, a network with 10 input nodes, 7 nodes in the second layer, 3 nodes in the third and 1 output layer has a 10:7:3:1 topology.

For example, figure B.1 shows the transition of the standard 2:2:1 topology used for XOR into a smaller 2:1:1 topology with extra connections (a standard 2:1:1 can not solve XOR anymore). This transition requires that node 2,2 must be removed/isolated. This means all connections to and from 2,2 must be deleted; the connection from bias node 0,0, from 1,1, from 1,2 and the connection to 3,1. Furthermore, nodes 1,1 and 1,2 of the input layer also have to be directly connected to the output layer node 3,1.

The structure file for NNC which implements these changes is shown in table B.4.

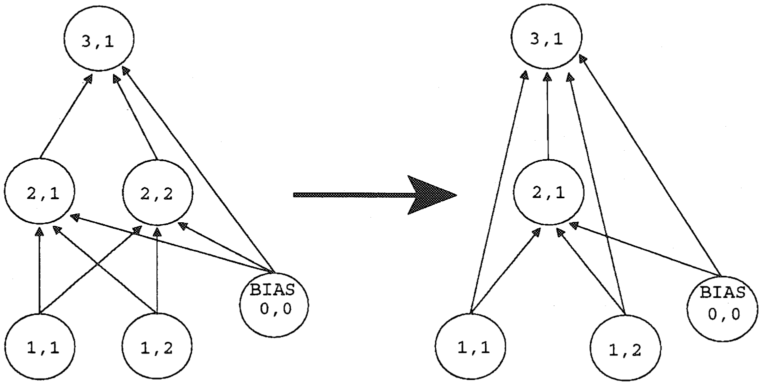


Figure B.1: *Changing a standard 2:2:1 topology in a 2:1:1 topology.*

```
d 2,2 0,0
d 2,2 1,1
d 2,2 1,2
d 3,1 2,2
c 3,1 1,1
c 3,1 1,2
```

Table B.4: *Structure file to make 2:1:1 topology.*

B.3.4 Saved Network File, extension .net

NNC saves (the weights of) a trained network in the file with the filename typed after NETFILE in the parameter file.

Each line of the saved network file contains a weight and the previous change in that weight. They are obtained from the last iteration of the back propagation algorithm over the training set. Internally NNC assigns an absolute node number to each node, simply going left to right, top to bottom (the bias neuron has absolute number 0).
The saved network file of the 2:2:1 XOR is shown in table B.5.

```
(  3,   0):    +1.985811   +0.000701
(  3,   1):    -5.755750   -0.000776
(  3,   2):    -7.420996   -0.000175
(  4,   0):    +4.997249   +0.000571
(  4,   1):    -3.457936   -0.005114
(  4,   2):    -3.558328   -0.004011
(  5,   0):    -3.069536   -0.005481
(  5,   3):    -7.858620   -0.001609
(  5,   4):    +7.106174   +0.004393
```

Table B.5: *Saved network file for 2:2:1 XOR.*

B.3.5 Response file, extension .ntp

The response file summarises the network response on the test set. Table B.6 shows the network response file for the XOR problem with the parameter settings of table B.3. The first line contains the size of the test set and the number of nodes in the output layer. After that for each data block of the test set, the identifiers, the target value and the response of the network.

```
4         1
1         1 0.000000 0.051119
1         2 1.000000 0.931141
1         3 1.000000 0.933403
1         4 0.000000 0.096507
```

Table B.6: *Response file for XOR.*

Appendix C

A Partially Connected Network

Fully connected feed-forward neural networks with many input variables typically have in the order of a thousand connections in total. Not only from a computational point of view it is advisable to minimise the total number of connections in the network. A reduced number of connections decreases the number of local minima of the problem and enables the same generalisation performance with a smaller training set.

Reducing the number of connections or '*pruning*' of the network can be done in a brute-force way. First, train a network with an ample number of hidden nodes till it has the desired performance. This performance, for example, is defined by an efficiency versus purity curve if one discriminates between two classes of events. After that, decrease the number of nodes until training shows a clear drop in performance: the network has become to small to implement a reasonable separation. It is also common to remove weak connections between nodes after the training, followed by a more finely tuned training of the remaining connections.

Instead of using the above methods it is more efficient to start with less connections by choosing an architecture that represents the problem. The input layer can sometimes be divided into groups, consisting of variables that logically group together, giving an input consisting of non-overlapping *receptive fields*. Partially connected networks enable to rule out unwanted correlations between the input variables.

The properties of a small partially connected network are studied here by applying it to a 'toy problem'.

C.1 The Toy Problem

The toy problem consists of the situation in which one wants to separate $\pi\pi$ events from πp and pp. It is assumed that events are characterised by only two variables per particle: the momentum (component) p and the energy of the particle, E.

The momenta p are generated from a uniform distribution in the range -2 to +2 GeV/c and E is calculated taking into account the p and π masses. Figure C.1 displays p versus E for generated pions and protons. The line E_1 separates these two classes for the negative momenta and E_2 for the positive ones.

First, a single pion-proton classifier was constructed that implements the separation lines E_1 and E_2. This network is shown in figure C.2 and requires a 2:2:1 architecture, i.e. 2 input nodes, 2 nodes in the second (or hidden) layer and 1 output node. Hidden node 1 is associated to E_1 and hidden node 2 to E_2. In addition one needs a bias node, which gives thresholds to the nodes in the hidden and output layer so that separation lines not going through the origin can also be formed.

The network was trained with the standard back-propagation[20] algorithm and the final connection strengths are also displayed in figure C.2. Each node first computes a weighted sum a over incoming connections. After that the sigmoid transfer function $\frac{1}{2}(\tanh(a/T) + 1)$ with $T = 1.0$ was applied to restrict the output of every node to [0,1].

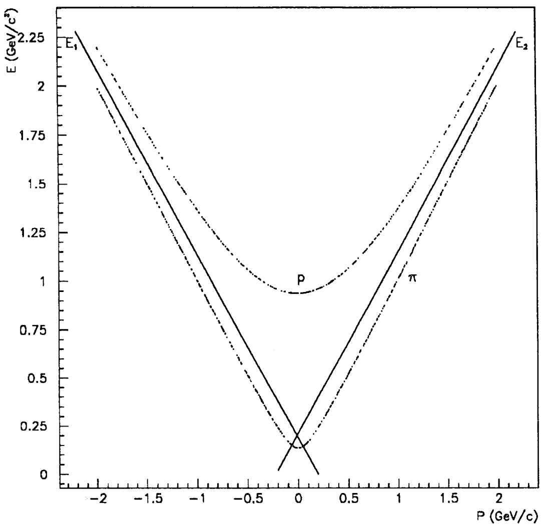


Figure C.1: *p versus E for the generated protons(p) and pions(π).*

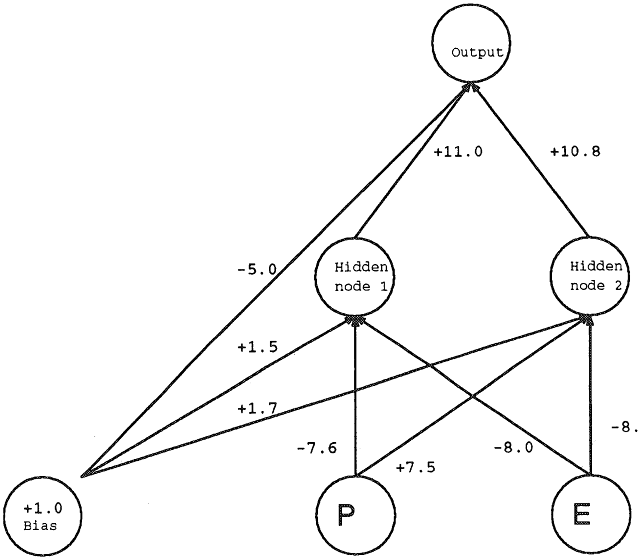


Figure C.2: *The small 2:2:1 network to separate protons and pions.*

The decision point of a single node lies at 0.5 (the midpoint of its output), which is when $a = 0$. Hidden node 1 implements $E_1 = -0.95p+0.19$ and hidden node 2 implements $E_2 = +0.94p+0.21$. The output node combines the effect of E_1 and E_2 to implement one separating curve. For a non-zero finite temperature T, the effective decision curve at the output node bends upward in the low momentum region.

By gluing two single pion-proton networks together one obtains a network with a 4:4:1 architecture. This network will be used to solve the toy problem. Two possible architectures are obvious: the fully connected architecture and a partially connected network, where the inputs are logically divided into two fields with each a (p, E)-pair. Figure C.3 shows these fully and partially connected architectures.

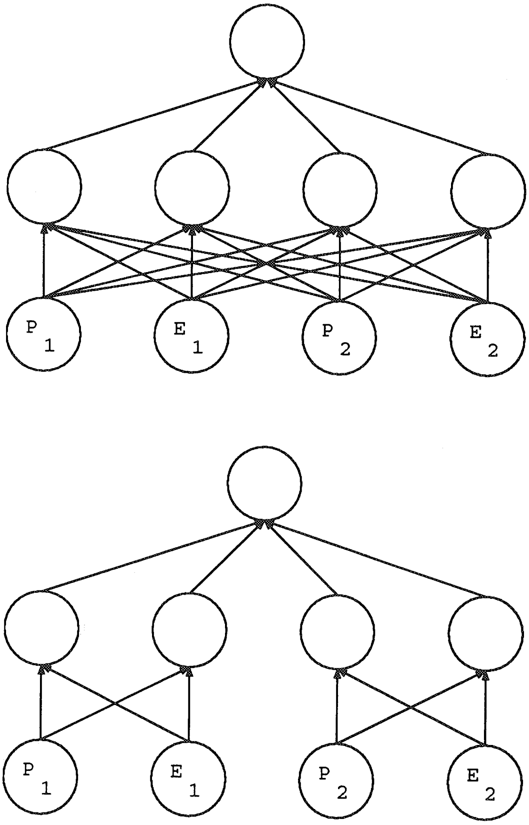


Figure C.3: *Architectures of the fully (top) and partially (bottom) connected 4:4:1 networks.*

A comparison of the behaviour of the two different networks involves two aspects. The following section first will discuss the differences between learning behaviour and performance of the two networks. Secondly, the advantage of a partially connected network in terms of computational cost will be considered.

C.2 Learning and Performance

In order to make a direct comparison between the two networks both were trained with the same sample, the same learning parameters and for the same number (1400) of *epochs*. An epoch is one sweep over the whole training set. The target values were 0.95 for $\pi\pi$ and 0.05 for the other combinations, and about 500 events were generated for each of these two classes.

First, the learning behaviour of both networks was studied by monitoring their error as a function of the number epochs. This error is defined as the sum of the squared differences between the target and the network output for each event in the training set, normalised to the total number of events.

It was found that the error during the learning behaved differently for the two networks, as is shown in figures C.4(a) and C.4(b). The error of the partially connected network decreases smoothly but sudden drops occur in the error development of the fully connected one. This can be explained by the presence of local minima, introduced by the unnecessary connections.

The response to an independent test set of the fully connected network is shown in figure C.4(c) and for the partially connected network in figure C.4(d). The output distribution of the partially connected network is more peaked around the target values.

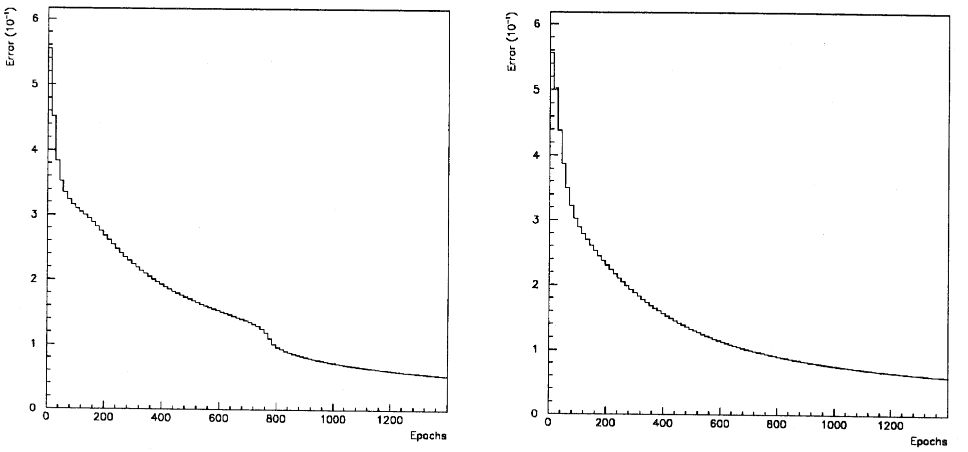
The partially connected network combines the strategy of two single pion-proton networks. In figure C.6, the connection strengths in the partially connected network are shown. Connections of the bias node to the nodes in the second layer and to the output node are not displayed, their strengths are displayed inside the nodes. If the first two nodes in the second layer are exchanged, one recognises two single networks each similar to figure C.2. Exchanging the first two hidden nodes and/or the last two leaves the logic intact.

It is expected that the connection strengths in the fully connected network reveal that the problem can be solved with less connections. In the fully connected network there are equivalent solutions for all possible ways to connect 2 pairs of hidden nodes to give the functionality of the partially connected network. The connection strengths in the fully connected network are displayed in figure C.5. There are indeed 8 big weights (absolute values of around 7) with the connectivity of the partially connected network. Four connections are weak with absolute values around 0.5 and the other four are very near to zero.

In many applications one would like to divide the problem into sub-problems, each being solved by an individual neural network. Later, these networks should be combined to one network, which will be trained further to obtain the final classifier.

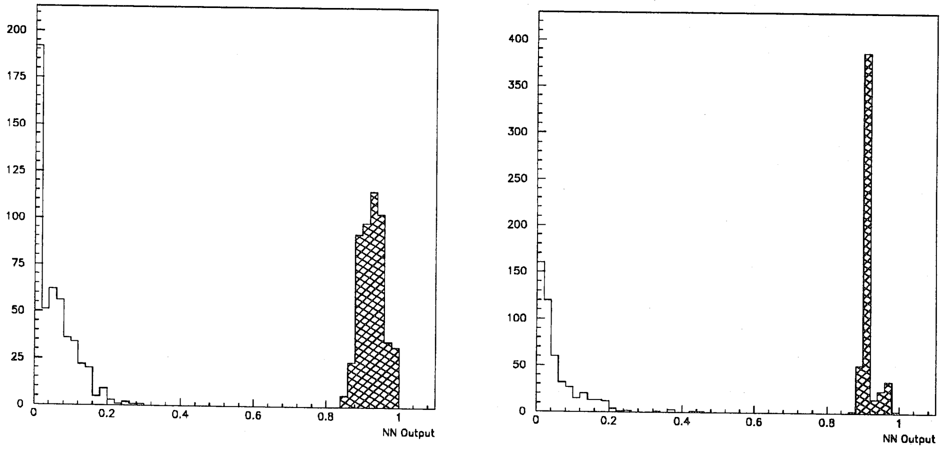
The advantages of such a multi-modular approach[105] are obvious. The solutions for the sub-problems are optimal, because they are implemented by dedicated small networks. Furthermore, after accomodating the small networks into one big network the training of the complete network starts off at a desired point in the multi-dimensional space.

The effect of this approach can be already demonstrated for the partially connected network of the toy problem. Before its training was started, the connection strengths (including biases) from input nodes to the hidden nodes were set equal to the ones in figure C.2. The others start off with small random values. Figure C.7 shows the error evolution during the training of this network. The same error level is reached a lot quicker than before. The connection strengths that have developed are shown in figure C.8.



(a) Error versus number of epochs for the fully connected network.

(b) Error versus number of epochs for the partially connected network.



(c) Output distribution fully connected network, $\pi\pi$ entries are hatched.

(d) Output distribution partially connected network, $\pi\pi$ entries are hatched.

Figure C.4: *Comparison between the partially and fully connected network.*

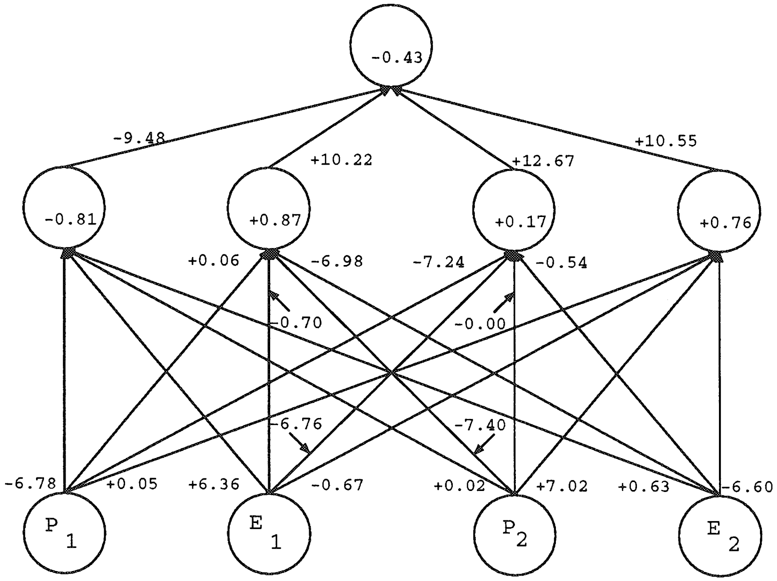


Figure C.5: *The connection strengths of the fully connected network.*

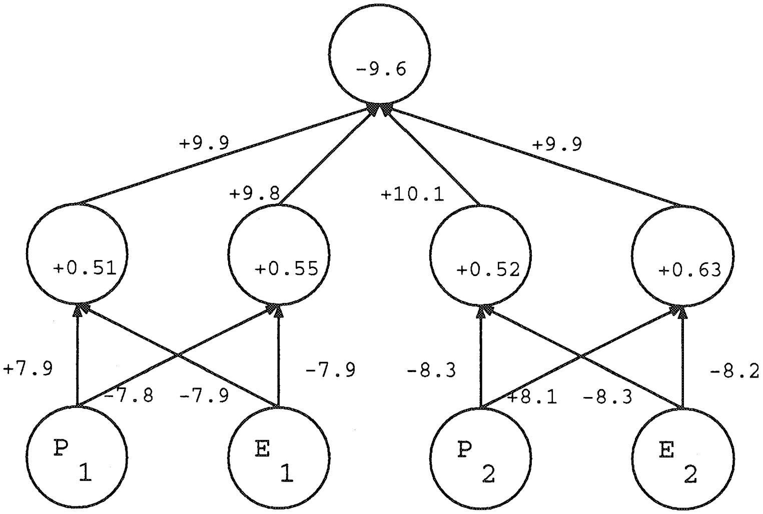


Figure C.6: *The connection strengths of the partially connected network.*

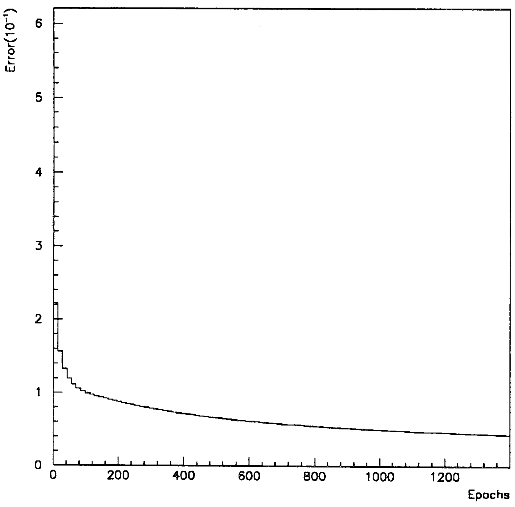


Figure C.7: *Error development of the network starting off with pre-trained fields.*

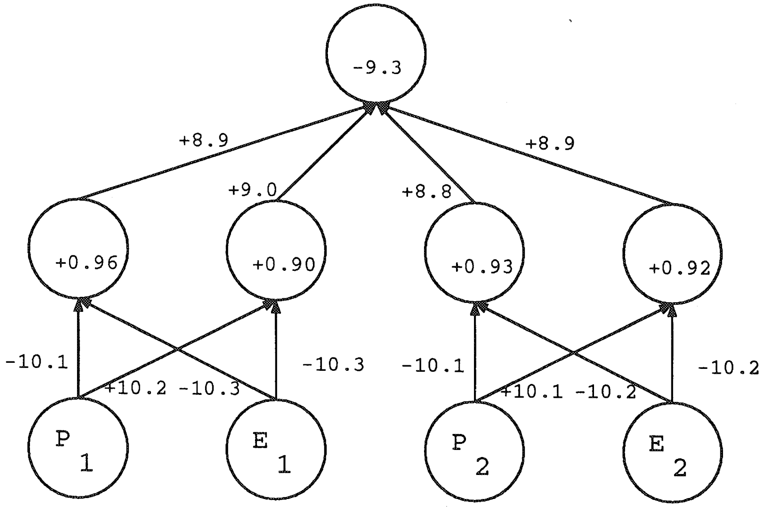


Figure C.8: *Final connection strengths of the network with pre-trained fields.*

C.3 Computational Cost

In the following, one considers the computational cost of a feed-forward network consisting of N input nodes and $L_1, L_2, \ldots$ nodes in the subsequent layers. A bias node counts as an extra node in all layers except the output (see figure C.2).

At single node level, a time T_X is spent to multiply and add the values of X incoming connections, followed by a time S for applying the sigmoid function giving its output. The total computing cost $C_{F(ully)}$ for the recall of a fully connected network is (using $T_X \approx XT_1$):

$$\text{(C.1)} \qquad \begin{aligned} C_F &= (T_N L_1 + T_{L_1} L_2 + T_{L_2} L_3 + \cdots) + S(L_1 + L_2 + L_3 + \cdots) \\ &\approx T_1(N L_1 + L_1 L_2 + L_2 L_3 + \cdots) + S(L_1 + L_2 + L_3 + \cdots) \end{aligned}$$

Assume that Q nodes of the input layer can be put in M different fields of the same size. For such a partially connected network with Q nodes equally distributed over M fields and $N - Q$ nodes still fully connected, the recall cost $C_{P(artially)}$ becomes:

$$\text{(C.2)} \qquad \begin{aligned} C_P &= (T_{N-Q} L_1 + M T_{Q/M} + T_{L_1} L_2 + T_{L_2} L_3 + \cdots) + S(L_1 + L_2 + L_3 + \cdots) \\ &\approx T_1((N-Q) L_1 + Q + L_1 L_2 + L_2 L_3 + \cdots) + S(L_1 + L_2 + L_3 + \cdots) \end{aligned}$$

From (C.1) and (C.2) one finds that the cost difference is:

$$\text{(C.3)} \qquad \Delta C = C_F - C_P = T_1 Q (L_1 - 1)$$

The back propagation algorithm consists of two steps. In the first step a recall is issued to get the errors at the output layer and store the outputs of all nodes for future reference. This gives a cost difference proportional to ΔC. The second step comprises a propagation of the errors from the output layer down to the input layer, which is from a computational point of view a recall in the other direction. This step involves the computation of the derivative of the sigmoid, the calculation of the weight change and its storage for calculating the next weight change.

Anyway, this second step leads to formulae like (C.1) and (C.2) with also a cost difference proportional to ΔC, because T_1 and S can be redefined to incorporate the extra computations, which are all at single node level.

The real-time speed difference is of course dependent on the actual implementation of the back-propagation algorithm. In this study a home-made package (see appendix B) written in the programming language C and running on an HP 9000/720 was used. The partially connected network was in real-time about 1.3 times faster during training under the same circumstances.

C.4 Concluding Remarks

A small partially connected neural network was studied and compared to its fully connected equivalent. By applying both networks to a toy problem, the following advantages of the partially connected architecture were observed:

- During learning, the error for the partially connected network decreases more smoothly than for the fully connected network.
- The partially connected architecture implements a more intuitive and better solution.
- Operating a partially connected network requires less CPU time.

To summarise, there is no reason not to use a partially connected neural network when the input can be grouped into non correlated fields. Furthermore, it is advantageous to combe the use of a partially connected network with a multi-modular approach. For example in a study like[41], one could first train a network dealing with the leptons and later incorporate it in the total network.

Appendix D

DBPIT

DBPIT is a special tool that has been developed by the author of this thesis to monitor the updates on the off-line calibration database of the Inner Detector. It runs as a batch job at DELPHI's experimental site ('the pit') and has been in use since the second half of the 1991 run. Its main purpose is to quickly detect anomalies in the off-line records, so that they can be corrected before DELANA starts the off-line processing of the data.

DBPIT checks the ID records in the calibration database 2-3 times a week, on request of the responsible ID expert on shift. The standard database scanning using DBPIT involves inspecting both a logfile and a set of useful histograms. These histograms are interfaced with the PRESENTER package, that is also used to view histograms of other ID jobs running in the pit.

The use of DBPIT extends to providing long-term information on the calibration parameters. For this purpose, DBPIT produces an N-tuple that contains all parameters of the ID calibration database. These N-tuples have been used in the calibration of the ID and similarly in the calibration of the VD specifically comparing the movements of the VD with the temperatures measured at different points around the ID.

In the following subsections, the functioning of DBPIT is described, illustrated with some example histograms.

D.1 Overview of Operation

When DBPIT is started, the user first chooses a start date and an end date, which define the period that needs to be checked. Next, the appropriate database records are extracted from the off-line database in ASCII form using the CARGO[106] package and their precise structure[80] is checked. Finally, an N-tuple and histograms are produced. The N-tuple directly reflects the extracted database records and is sorted in time. Although the standard histograms are sufficient to check the calibration database under normal conditions, the N-tuple allows 'experts' to study any correlation between the ID calibration parameters.

During all DBPIT's stages of execution, messages of various kind are collected in a logfile. For any calibration parameter, the window of safe (normally occurring) values is user-defined. Because the high voltages of the ID don't read exactly 0.0 V, even when they are off, a second window defining the safe values for the HV off state was introduced in order to suppress warnings from DPBIT.

The safe windows are designated in a title file which is a sequence of lines with name, minimum, maximum and off-state maximum of the parameter. They can be changed before DBPIT is run to suppress warnings or to increase the sensitivity for producing them.

The next section describes the logfile and the histograms in more detail.

D.2 Output of DBPIT

DBPIT has two output streams. The first stream is the logfile in which messages are collected at any time during the operation of DBPIT. The second stream is a set of histograms. Together, the logfile and the histograms are sufficient to spot a wide variety of problems.

The logfile collects warnings and errors and is constantly updated during the execution of DBPIT. Table D.1(a) shows an example of a typical logfile. The first two lines summarise the number of updates found over the requested period and they are followed by all messages produced by DBPIT. From the left to the right the columns designate:

- DATE and TIME of the start of the period of validity in the form YYMMDD and HHMMSS.
- EDATE and ETIME: the date and time this particular record entered the off-line database.
- UPDATE: the name of the calibration parameter for which the warning occurred.
- VALUE of the parameter which caused the warning. This example logfile contains warnings for JETFS values that are higher than usual, and JETHV, PRESA and PRESJ values that are lower than usual.
- HV ON: the status of the ID High Voltage at DATE and TIME. If an 'F(alse)' is displayed in this column the Jet Chamber high voltages were off at that moment: the detector is in OFF or in STANDBY state and it is not taking any data.

The histograms produced by DBPIT are divided into three categories.

General histograms contain update frequencies for all calibration parameters and distributions of their values. Figure D.1(b) shows the frequency histogram for all updated parameters, the two peaks are JETFS for sector 17 and sector 24. These updates were identified to be fluctuations within 3.5 minutes and they are caused by noise readings of the probes that measure JETFS.

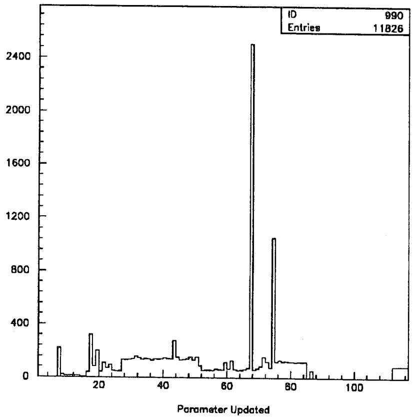
Other histograms contain parameter values as a function of time in the requested period. In DBPIT's histograms time is measured in days (from 01-01-1990) in order to facilitate scaling of the histograms for variable periods. Figure D.1(c), for example, shows the total update frequency as a function of time from day 638 (01-10-1991) till the end of the 1991 run (11-11-1991).

Finally, histograms are produced which are used to check the validation of updates. These histograms are distributions of the difference between consecutive updates of a parameter. An example of such a histogram is shown in figure D.1(d) for all temperature updates. There is a gap at $0.3° C$, which is the correct minimum difference between consecutive updates.

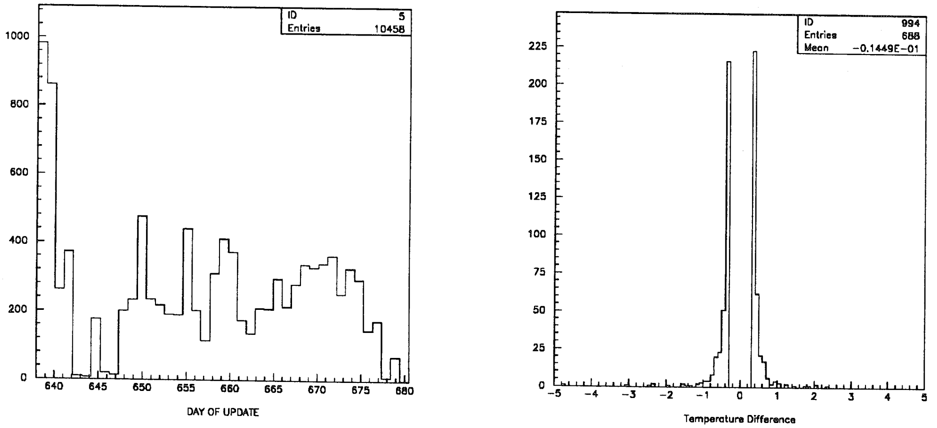
```
UPDATES 1054
PERIOD  911101:000228 TO 911104:235756

DATE    TIME    EDATE   ETIME   UPDATE   VALUE    HV ON
911101  083041  911101  110996  JETFS22  5562.34  F
911101  155436  911101  163167  JETHV5   2102.00
911101  160233  911101  163301  JETFS22  5560.51  F
911101  201543  911102  043950  JETFS22  5562.34
911102  051442  911103  010542  JETFS11  6283.74  F
911102  164224  911103  011274  JETFS21  6165.02
911103  005243  911103  011938  JETFS22  5566.01
911103  015347  911103  160283  JETFS22  5566.92  F
911103  024904  911103  160349  JETFS22  5563.26  F
911103  160711  911104  093802  JETFS22  5565.09  F
911103  182538  911104  094019  PRESA     969.74  F
911103  202026  911104  094298  PRESA     968.71  F
911103  213112  911104  094379  PRESA     967.64  F
911103  221359  911104  094434  PRESA     966.32  F
911103  231551  911104  094516  PRESJ     995.09  F
911103  234808  911104  094710  PRESA     965.19  F
911104  003323  911104  094955  JETFS22  5564.17
911104  012643  911104  095116  PRESA     964.06
911104  030517  911104  095225  JETFS21  5564.09
911104  040741  911104  095289  PRESJ     992.97
911104  042041  911104  095301  PRESA     963.03
911104  053638  911104  095363  PRESA     961.93
911104  054435  911104  095370  PRESJ     991.95
911104  072606  911104  095456  PRESA     963.01
SUCCESSFUL COMPLETION
```

(a) Example DBPIT log file.



(b) Update frequencies.



(c) Number of updates per day.

(d) Check of the update threshold for the temperatures.

Figure D.1: *DBPIT's logfile and example histograms.*

References

[1] S.L. Glashow, *"Partial-symmetries of weak interactions"*, Nucl. Phys. **22** (1961) 579;
S. Weinberg, *"A model of leptons"*, Phys. Rev. Lett. **19** (1967) 1264;
A. Salam, in *"Elementary particle theory: relativistic groups and analyticity"*, [Ed.] N. Svartholm , Almqvist and Wiksell, Stockholm (1968) 367.

[2] G. Altarelli, *"Partons in quantum chromodynamics"*, Phys. Rep. **81** (1982) 1-129;
W. Marciano and H. Pagels, *"Quantum chromodynamics"*, Phys. Rep. **36** (1978) 137-276.

[3] *Review of Particle Properties,* Phys. Rev. **D 50** (1994) 1173.

[4] P.W. Higgs, *"Broken Symmetries, massless particles and Gauge fields"*, Phys. Lett. **12** (1964) 132;
P.W. Higgs, *"Broken symmetries and the masses of Gauge bosons"*, Phys. Rev. Lett. **13** (1964) 508.

[5] N. Cabibbo, *"Unitary symmetry and leptonic decays"*, Phys. Rev. Lett. **10** (1963) 532;
M. Kobayashi and T. Maskawa, *"CP violation in the renormalizable theory of weak interactions"*, Prog. Theor. Phys. **49** (1973) 652.

[6] *"Z Physics at LEP"*, [Eds.] G. Altarelli, R. Kleiss and C. Verzegnassi, CERN Report CERN-89-08 **I** (1989) 10.

[7] E. Locci, *"B Physics at LEP: present and perspectives"*, Nucl. Instr. Meth. **A 333** (1993) 27-41.

[8] K. Lande *et al.*, *"Observation of long-lived neutral V particles"*, Phys. Rev. **103** (1956) 1901.

[9] C. Albajar *et al.* (UA1), *"Search for $B^0 - \bar{B}^0$ oscillations at the CERN proton-antiproton collider"*, Phys Lett. **B 186** (1987) 247.

[10] A. Albrecht *et al.* (ARGUS), *"Observation of $B^0\bar{B}^0$ Mixing"*, Phys. Lett. **B 192** (1987) 245-252.

[11] A. Bean *et al.* (CLEO), *"Limits on $B^0\bar{B}^0$ Mixing and τ_{B^0}/τ_{B^+}"*, Phys. Rev. Lett. **58** (1987) 183-186.

[12] J.H. Christenson, J.W. Cronin, V.L. Fitch and R. Turlay, *"Evidence for the 2π decay of the K_2^0 meson"*, Phys. Rev. Lett. **13** (1964) 138.

[13] G. Altarelli, *"Three lectures on flavour mixing"*, CERN TH 4896-87 (1987).

[14] P.D.B. Collins, A.D. Martin and E.J. Squires, *"Particle Physics and Cosmology"*, John Wiley & Sons, 1989, ISBN 0-471-60088-1.

[15] D. Hebb, *"The Organisation of Behaviour"*, Wiley, New York, 1949.

[16] B. Widrow, M.E. Hoff, *"Adaptive switching circuits"*, 1960 IRE WESOC Convention Record, 96-104, New York, 1960.

[17] T. Kohonen, *"Self-Organization and Associative Memory"*, Second Edition, Springer-Verlag, Berlin, 1988.

[18] D. Desieno, *"Adding a conscience to competitive learning"*, Proc. Int. Conf. on Neural Networks, **I**, 117-124, IEEE Press, New York, July 1988.

[19] P.J. Werbos, *"Beyond regression: New tools for prediction and analysis in the behavioural sciences"*, Doctoral Dissertation, Appl. Math., Harvard University, November 1974.

[20] D.E. Rumelhart, G.E. Hinton and R.J. Williams, *"Learning Internal Representations by Error Propagation"*, in D.E. Rumelhart and J.L. McClelland [Eds.], *"Parallel Distributed Processing: Explorations in the Microstructure of Cognition"*, **I**, MIT Press, Cambridge MA, 1986.

[21] M.L. Minsky and S. Papert, *"Perceptrons: an introduction to computational geometry"*, MIT Press, Cambridge 1969.

[22] M.R. Devos and G.A. Orban, *"Self Adaptive Backpropagation"*, Proc. NeuroNimes (1988).

[23] T. Tollenaere, *"SuperSAB: Fast Adaptive Back Propagation with Good Scaling Properties"*, Neural Networks **3** (1990) 561.

[24] R.A. Jacobs, *"Increased Rates of Convergence Through Learning Rate Adaption"*, Neural Networks **1** (1988) 295.

[25] T. Ash, *"Dynamic node creation in backpropagation networks"*, ICS Report 8901, Cog. Science Dept., University of California, San Diego, February 1989.

[26] F.J. Pineda, *"Recurrent backpropagation and the dynamical approach to adaptive neural computation"*, Neural Computation **1** (1989) 161-172.

[27] D.E. Rumelhart and J.L. McClelland, *"Parallel Distributed Processing: Explorations in the Microstructure of Cognition"*, **I** & **II**, MIT Press, Cambridge MA, 1986.

[28] R.A. Fisher, *"The use of multiple measurements in taxonomic problems"*, Annals Eugenics **7** (1936) 179.

[29] M. Kendall, A. Stuart and J.K. Ord, *"The Advanced Theory of Statistics"*, **3**, fourth edition, C. Griffin & Co. Ltd., London, 1976.

[30] D. Hilbert, *"Mathematical Problems"*, Lecture given at the International Congress of Mathematicians at Paris in 1900, Trans. M.W. Newson, Bull. Amer. Math. Soc. **8** (1902) 437-479.

[31] A.N. Kolmogorov, *"On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition [in Russian]"*, Dokl. Akad. Nauk USSR **114** (1957) 953-956.

[32] D.A. Sprecher, *"On the structure of continuous functions of several variables"*, Trans. Am. Math. Soc. **115** (1965) 340-355.

[33] G.G. Lorentz, *"The 13-th problem of Hilbert"*, Proc. of Symposia in Pure Math., Am. Math. Soc. **28** (1976) 419-430.

[34] R. Hecht-Nielsen, *"Kolmogorov's mapping neural network existence theorem"*, Proc. of the Int. Conf. on Neural Networks, **III**, 11-13, IEEE Press, New York, 1987.

[35] R. Hecht-Nielsen, *"Theory of the backpropagation neural network"*, Proc. of the Int. Joint Conf. on Neural Networks, **I**, 593-611, IEEE Press, New York, 1989.

[36] G. Tesauro and B. Janssens, *"Scaling Relationships in Back-propagation Learning"*, Complex Systems **2** (1988) 39.

[37] R. Duda and P.E. Hart, *"Pattern Classification and Scene Analysis"*, Wiley, New York, 1973.

[38] C. Peterson and E. Hartman, *"Explorations of the Mean Field Theory Learning Algorithm"*, Neural Networks **2** (1989) 475.

[39] E. Baum and D. Haussler, *"What Size Net Gives Valid Generalisation"*, Neural Computation **1** (1989) 151-160.

[40] Martin Los, Nicolo de Groot, *"B-Tagging in DELPHI with a feedforward neural network"*, Proceedings of the 1991 workshop 'Neural Networks: from Biology to High Energy Physics' [Eds.] O.Benhar, C. Bosio, P. Del Guidice, E. Tabet, 1992, ETS Editrice, Pisa, 459-466.

[41] P. Abreu *et al.* (DELPHI), *"Classification of the Hadronic Decays of the Z^0 into b and c Quark Pairs using a Neural Network"*, Phys. Lett. **B 295** (1992) 383.

[42] G. Cosmo, A. De Angelis, N. De Groot, P. Del Guidice, P. Eerola, J. Kalkkinen, L. Lyons, M. Los, E. Torassa, E. Vallazza, *"Classification of the Hadronic Decays of the Z^0 into b and c Quark Pairs using a Neural Network"*, DELPHI 92-67 Dallas PHYS 179 (1992).

[43] G. Cosmo, A. De Angelis, P. Eerola, J. Kalkkinen, L. Lyons, N. De Groot, M. Los, *"Using Neural Networks for the classification of the decays of Z^0 bosons"*, Computing in high energy physics: Proceedings Ed. by C. Verkerk and W. Wojcik, CERN 92-07 (1992) 665-668.

[44] A. Petrolini, *"Use of Neural Network Classifiers in Higgs search"*, DELPHI 91-43 PHYS 98 (1991).

[45] L. Lönnblad, C. Peterson, T. Rögnvaldsson, *"Mass Reconstruction with a Neural Network"*, Phys. Lett. **B 278** (1992) 181-186.

[46] M. Gyulassy, M. Harlander, *"Elastic Tracking and Neural Network Algorithms for Complex Pattern Recognition"*, Comp. Phys. Comm. **66** (1991) 31-46.

[47] N. de Groot, M. Los and S. Tzamarias, *"Pion-Kaon Separation with a Feed-Forward Neural Network"*, New Computing Techniques in Physics Research II, 1992.

[48] P. Ribarics, *"Neural Network Trigger in the H1 Experiment"*, Proceedings of the 1991 workshop 'Neural Networks: from Biology to High Energy Physics' [Eds.] O. Benhar, C. Bosio, P. Del Guidice, E. Tabet, 1992, ETS Editrice, Pisa, 383-410.

[49] L. Lönnblad, C. Peterson, T. Rögnvaldsson, *"Using neural networks to identify jets"*, Nucl. Phys. **B 349** (1991) 675.

[50] N. de Groot, *"The Determination of the Branching Ratios of the Z^0 Boson into Heavy Quarks Using Neural Networks"*, PhD thesis, University of Amsterdam, March 1993.

[51] T. Sjöstrand, *"The LUND Monte Carlo for jet fragmentation"*, Comp. Phys. Comm. **27** (1982) 243;

T. Sjöstrand, *"The LUND Monte Carlo for e^+e^- physics"*, Comp. Phys. Comm. **28** (1983) 229;

T. Sjöstrand and M. Bengtsson, *"The LUND Monte Carlo for jet fragmentation and e^+e^- physics: JETSET version 6.3: an update"*, Comp. Phys. Comm. **43** (1987) 367.

[52] L. Lönnblad, *"ARIADNE version 4: A program for simulation of QCD cascades implementing the color dipole model"*, Comp. Phys. Comm. **71** (1992) 15-31.

[53] I.G. Knowles, *"Spin correlations in parton-parton scattering"*, Nucl. Phys. **B 310** (1988) 571;

G. Marchesini and B.R. Webber, *"Monte Carlo simulation of general hard processes with coherent QCD radiation"*, Nucl. Phys. **B 310** (1988) 461.

[54] W. Bartel *et al.* (JADE), *"Particle Distribution in 3-Jet Events Produced by e^+e^- Annihilation"*, Z. Phys. **C 21** (1983) 37.

[55] Z. Fodor, *"Can we see the triple-gluon vertex already at 44 GeV?"*, Phys. Rev. **D 40** (1989) 3590;

Y.K. Kim *et al.* (AMY), *"Comparison of Quark and Gluon Jets produced in High-Energy e^+e^- Annihiliations"*, Phys. Rev. Lett. **63** (1989) 1772;

L. Jones, *"Towards a systematic jet classification"*, Phys. Rev. **D 42** (1990) 811.

[56] C.C. 'Casey' Klimasauskas, *"Neural Nets and Noise Filtering"*, Dr. Dobb's Journal, January 1989.

[57] B. Widrow and S.D. Stearns, *"Adaptive Signal Processing"*, Englewood Cliffs NJ, Prentice-Hall, 1985.

[58] P. Aarnio *et al.* (DELPHI), *"The DELPHI detector at LEP"*, Nucl. Instr. Meth. **A 303** (1991) 233.

[59] N. Bingefors *et al.*, *"The DELPHI Microvertex detector"*, Nucl. Instr. Meth. **A 328** (1993) 447.

[60] P. Checcia, *"Study of electromagnetic calorimeters with $Z^0 \to e^+e^-\gamma$ events"*, DELPHI 91-100 PHYS 145 (1991).

[61] P. Aarnio *et al.* (DELPHI), Technical Proposal, DELPHI 83-66/1 (1983);

P. Aarnio *et al.* (DELPHI), DELPHI Progress Report, DELPHI 84-60 (1984);

Stefan Haider, Chapters 3-6 in *"B_s^0 mixing at the Z^0 resonance determined with the RICH technique"*, PhD thesis, University of Leiden, June 1995.

[62] P. Abreu *et al.* (DELPHI), *"First measurement of the strange quark asymmetry at the Z^0 pole"*, Z. Phys. **C 67** (1995) 1-14.

[63] P. Abreu *et al.* (DELPHI), *"Charged kaon production in tau decays at LEP"*, Phys. Lett. **B 334** (1994) 435.

[64] P. Abreu *et al.* (DELPHI), *"Lifetime and production rate of beauty baryons from Z^0 decays"*, CERN-PPE/95-54 (1995).

[65] B. Adeva *et al.* (L3), *"The construction of the L3 detector at LEP"*, Nucl. Instr. Meth. **A 289** (1990) 35.

[66] D. Decamp *et al.* (ALEPH), *"ALEPH: a detector for electron-positron annihilations at LEP"*, Nucl. Instr. Meth. **A 294** (1990) 121.

[67] K. Ahmet *et al.* (OPAL), *"The OPAL detector at LEP"*, Nucl. Instr. Meth. **A 305** (1991) 275.

[68] L. Benetau *et al.* (DELPHI DAS Group), *"Architecture and Performance of the DELPHI Trigger System"*, DELPHI 92-162 DAS 135 (1992).

[69] J.A. Fuster, C. Lacasta, G. Valenti and J.A. Valls, *"The 1^{st} and 2^{nd} level Trigger User Manual; Summary of the 1992 Trigger Performance"*, DELPHI 93-42 DAS 144 (1993).

[70] B. Bouquet, Ph. Gavillet and J-Ph. Laugier, *"The Fourth Level Trigger of DELPHI"*, DELPHI 92-29 DAS 141 (1992).

[71] P. Aarnio *et al.* (DELPHI), *"DELPHI data analysis program (DELANA)"*, User's Guide, DELPHI 89-44 PROG 137 (1989).

[72] V. Perevozchikov and N. Smirnov, *"PHDST package description"*, DELPHI 91-85 PROG 176 (1991).

[73] P. Aarnio *et al.* (DELPHI), *"DELSIM Version 3.4, the DELPHI Event Generation and Detector Simulation"*, User's Guide, DELPHI 89-67 PROG 142 (1989);
P. Aarnio *et al.* (DELPHI), DELSIM Reference Manual, DELPHI 89-68 PROG 143 (1989).

[74] F. Sauli, *"Principles of operation of multiwire proportional and drift chambers"*, CERN 77-09 (1977).

[75] C. Baudoin, J. Timmermans and F. Udo, *"Test of a proportional counter with and without a magnetic field"*, NIKHEF-H internal report, July 1988.

[76] H. Schmid, *"Electronic Analog/Digital conversions"*, Litton Educational Publishing Inc., 1970.

[77] R. Brouwer, Jet Triggermodule (JETTM) F686A, NIKHEF-H Electronics Department.

[78] J. Weber, Layer Triggermodule (LAYTM) F686B, NIKHEF-H Electronics Department.

[79] Nicolo de Groot and Krzysztof Korcyl, *"The Inner Detector third level trigger"*, DELPHI 92-58 DAS 125 (1992).

[80] Yu. Belokopytov, G. Gopal, H. Klein, V. Perevozvhikov, W. Venus, *"The DELPHI database description: calibration & detector status file"*, DELPHI 93-4 PROG 194 DAS 138 (1993).

[81] G. Charpak *et al.*, *"Progress in high-accuracy proportional chambers"*, Nucl. Instr. Meth. **148** (1978) 471.
G. Charpak *et al.*, *"High-accuracy localization of minimum ionizing particles using the cathode-induced charge centre-of-gravity read-out"*, Nucl. Instr. Meth. **167** (1979) 455.

[82] I. Endo *et al.*, *"Systematic shifts of evaluated charge centroid for the cathode read-out Multiwire Proportional Chamber"*, Nucl. Instr. Meth. **188** (1981) 51-58.

[83] F. Piuz, R. Roosen and J. Timmermans, *"Evaluation of systematic errors in the avalanche localization along the wire with cathode strips read-out MWPC"*, Nucl. Instr. Meth. **196** (1982) 451-462.

[84] M.E. Bonapart, *"Test results of a MWPC with Cathode-Strip Read Out"*, Master thesis, University of Leiden, September 1987.

[85] Z. Hajduk, *"Cathode strips read-out in the Inner Detector"*, DELPHI 84-51 (1984).

[86] M.E. Los, *"Testdata analysis of the Trigger Layers of DELPHI's Inner Detector"*, Master thesis, University of Amsterdam, February 1990.

[87] H. De Boeck, G. Wilkinson, *"MUFLAG: A Framework for MUON Identification"*, DELPHI 93-14 PHYS 263 (1993).

[88] G. Wilkinson, P. Collins, *"MUCFIX - Refining the MUON Chamber Information on the DST"*, DELPHI 93-13 PHYS 262 (1993).

[89] Nigel Crosland, Guy Wilkinson and Peter Kluit, *"Muon Identification within DELPHI"*, DELPHI 92-17 PHYS 137 (1992).

[90] C.J. Kreuter, *"Longitudinal Shower Development in the DELPHI Electromagnetic Calorimeter HPC"*, Master thesis, University of Karlsruhe, IEKP-KA/93-9, June 1993.

[91] D. Buskulic *et al.* (ALEPH), *"Heavy flavour production and decay with prompt leptons in the ALEPH detector"*, Z. Phys. **C 62** (1994) 179-198.

[92] P. Abreu *et al.* (DELPHI), *"Measurement of the $B^0-\bar{B}^0$ mixing parameter at DELPHI"*, Phys. Lett. **B 332** (1994) 488-500.

[93] P. Abreu *et al.* (DELPHI), *"A study of $B^0-\bar{B}^0$ mixing using semileptonic decays of B hadrons produced from Z^0"*, Phys. Lett. **B 301** (1993) 145.

[94] M. Acciarri *et al.* (L3), *"Measurement of the $B^0-\bar{B}^0$ Mixing Parameter and the $Z \to b\bar{b}$ Forward-Backward Asymmetry"*, Phys. Lett. **B 335** (1994) 542-553.

[95] R. Akers *et al.* (OPAL), *"Measurements of $B^0-\bar{B}^0$ mixing, $\Gamma(Z^0 \to b\bar{b})/\Gamma(Z^0 \to hadrons)$ and semileptonic branching ratios for b-flavoured hadrons in hadronic Z^0 decays"*, Z. Phys. **C 60** (1993) 199-216.

[96] Martin Los, *"Classifying jet cascades with a muon using a neural network"*, International Journal of Neural Systems **3** (Supp. 1992) 321-330.

[97] T. Sjöstrand, *"PYTHIA 5.7 and JETSET 7.4, Physics and Manual"*, CERN-TH 7112/93 (1993) 273.

[98] G. Altarelli, N. Cabibbo, G. Corbò, L. Maiani and G. Martinelli, *"Leptonic Decay of Heavy Flavors: a theoretical update"*, Nucl. Phys. **B 208** (1982) 365.

[99] N. Isgur *et al.*, *"Semileptonic B and D decays in the quark model"*, Phys. Rev. **D 39** (1989) 799.

[100] D. Buskulic *et al.* (ALEPH), *"Measurement of the* $B \rightarrow \tau^- \bar{\nu}_\tau X$ *branching ratio"*, Phys. Lett. **B 298** (1993) 479-491.

[101] P. Abreu *et al.* (DELPHI), *"Measurement of the* $B^0 - \bar{B}^0$ *mixing using the average electric charge of hadron jets in* Z^0 *decays"*, Phys. Lett. **B 322** (1994) 459-472.

[102] Leslie Lamport, *"LaTeX: A Document Preparation System"*, ISBN 0-201-15790-X.

[103] R. Brun *et al.*, *"PAW - Physics Analysis Workstation"*, CERN Program Library Q121, long write-up, October 1989.

[104] J.A.M. Vermaseren, *"Axodraw"*, Comp. Phys. Comm. **83** (1994) 45-58.

[105] F. Fogelman Soulie, *"Multi-Modular Neural Networks Architectures for real world Applications"*, Proceedings of the 1991 workshop 'Neural Networks: from Biology to High Energy Physics' [Eds.] O. Benhar, C. Bosio, P. Del Guidice, E. Tabet, 1992, ETS Editrice, Pisa, 67-76.

[106] Yu. Belokopytov, V. Perevozchikov, *"CARGO Database Management Package"*, DELPHI 93-5 PROG 195 (1993).

This thesis was typeset with LaTeX[102]. Plots were made using the CERN package PAW[103] and incorporated into the LaTeX-document using the `psfig` package. Other figures were produced with Axodraw[104], using Postscript and `dvips` from Radical Eye Software, or with **`!Artworks`** from Computer Concepts on an Acorn Archimedes A440/I.

List of Figures

1.1 The Feynman box diagrams responsible for B_d^0-$\bar{B}_d^0$ mixing. The dominant contribution comes from the top quark exchange and therefore only the CKM elements of the top quark have been indicated at the vertices. 7

2.1 A generic node. 12
2.2 A simple neural network architecture with three layers. 12
2.3 The linear associator. 14

3.1 A feed-forward neural network with three layers. 17
3.2 The sigmoid function $g(x) = 1/(1 + e^{-x/T})$ for $T = 0, T = 0.5, T = 1.0$ and $T = 2.0$. . . 18
3.3 Separation of two distributions A(x,y) and B(x,y). 21

4.1 Perspective view of the DELPHI detector. 26

5.1 Transverse view of the first Inner Detector quadrant. 33
5.2 A Trigger Layer in detail. 33
5.3 The two valid hit patterns in a TF-chip region, a diagonal pattern (a) and a straight pattern (b); the other two valid patterns are their mirror images. 34
5.4 Z residual distribution for $b = 0.02$. 37
5.5 The polar angle θ (degrees) versus the z residuals (mm). 38
5.6 Z resolution (mm) as a function of the polar angle θ (degrees). 38

6.1 Explicit examples of decays belonging to the classes PB, PC, $\overline{\mathrm{SC}}$ and SC, respectively. *(a)* $\bar{B}_s^0 \to l^-$. *(b)* $\bar{D}^0 \to l^+$, with the $\bar{D}^0$ directly produced in $Z^0 \to D^0\bar{D}^0$. *(c)* $B_d^0 \to (\bar{c}s) \to l^-$. *(d)* $\bar{B}_d^0 \to D^+ \to l^+$. 47
6.2 Momenta and transverse momenta distributions in the dilepton sample for the 1992 data (dots) and simulation (histograms). The contributions of the different (di)lepton classes are shown. 51
6.3 Dilepton sample composition for the 1991 and 1992 simulation as a function of p_t^{min}. The contribution PB-SC, which dilutes the χ sensitivity is shown separately. 52
6.4 $|Q_{jet}|$ for $c \to l^+$ (dashed) and for the other backgrounds (solid). 54
6.5 The distributions of the momentum p, the transverse momentum p_t and the absolute value of the jet charge $|Q_{jet}|$ for $b \to l^-$ (solid) and $b \to c \to l^+$ (dashed). These samples formed the training sets for the neural network. 55
6.6 Neural network output distribution for $b \to l^-$ (solid) and $b \to c \to l^+$ (dashed). 56
6.7 Efficiency versus purity for the test set to select $b \to l^-$ jets applying neural network cuts or p_t cuts. The test set consists of about 6000 $b \to l^-$ jets and 6000 $b \to c \to l^+$ jets. 57
6.8 The output distribution of the neural network on single jets for the simulation and the data. 58
6.9 Plots of $R(p_t^{min})$ and $R(nn^{min})$ belonging to the various fits. The data points are shown together with the fitted values and their corresponding χ (dashed line). Also shown in each case are the values corresponding to $\chi = 0.0$ (dotted line). 60

B.1 Changing a standard 2:2:1 topology in a 2:1:1 topology. 68

C.1 p versus E for the generated protons(p) and pions(π). 72
C.2 The small 2:2:1 network to separate protons and pions. 72
C.3 Architectures of the fully (top) and partially (bottom) connected 4:4:1 networks. 73
C.4 Comparison between the partially and fully connected network. 75
C.5 The connection strengths of the fully connected network. 76
C.6 The connection strengths of the partially connected network. 76
C.7 Error development of the network starting off with pre-trained fields. 77
C.8 Final connection strengths of the network with pre-trained fields. 77

D.1 DBPIT's logfile and example histograms. 81

List of Tables

1.1 The fundamental left handed spin $\frac{1}{2}$ fermions. 3
1.2 The fundamental couplings. 4

6.1 The cuts for the loose, standard and tight muon identification. 41
6.2 MUFLAG: efficiencies and misidentification probabilities for the loose, standard and tight muon identification. 41
6.3 ELPROB: efficiencies and misidentification probabilities for the loose, standard and tight electron identification. The errors on these values are in the order of 10-15%. . . 44
6.4 The cuts of the loose, standard and tight electron identification if no $\mathrm{d}E/\mathrm{d}x$ information is available. 44
6.5 The cuts of the loose, standard and tight electron identification; $\mathrm{d}E/\mathrm{d}x$ information is available. 44
6.6 The contribution (in %) of various single lepton sources in the 1992 simulation, before and after $p_t > 1.0\ GeV/c$. 46
6.7 The number of like sign, unlike sign and total dileptons for the 1991 and 1992 simulation and data samples. 48
6.8 Contributions to the like sign fraction from the different dilepton categories. 49
6.9 Dimuon fractions in the 1991 simulation, before and after $p_t^{min} > 1.0\ GeV/c$. 50
6.10 Dilepton fractions in the 1992 simulation, before and after $p_t^{min} > 1.0\ GeV/c$. 50
6.11 Settings of the neural network training parameters. 54
6.12 Dilepton composition of the 1991 and 1992 simulation after the neural network cut $nn^{min} > 0.5$. 57
6.13 The results for χ (in percent) of the various fits for 1992. 59
6.14 Contributions to the systematic uncertainty in the measurement of χ. Variations given in percent are relative to the values in the simulation. 61
6.15 Summary of the results of the two different methods to determine χ. 62
6.16 Comparison with other results. The χ values are in percent. 62

B.1 Truth table for XOR. 66
B.2 The training set xor.nnd. 66
B.3 Parameter file for the XOR problem. 67
B.4 Structure file to make 2:1:1 topology. 68
B.5 Saved network file for 2:2:1 XOR. 69
B.6 Response file for XOR. 69

Index

α, *see* back-propagation momentum term
η, *see* learning rate
θ_W, *see* electroweak mixing angle
χ, *see* B^0-$\bar{B}^0$ mixing parameter χ
ξ, *see* background charge correlation

ACCMM, 61
activation function, 18
affine transformations, 15
axial coupling, 4

B^0 meson, 1
B^0-$\bar{B}^0$ mixing, 6
    parameter χ, 9
background, 46
    charge correlation, 48
back-propagation, 19
    derivation, 63
    momentum term, 19
    theorem, 20
Bayes limit, 23
BCK, *see* background
BCO, 29
bias input, 15
boson, 3
    Higgs, 4
    $W^{\pm}$, 4
    Z^0, 4

Cabibbo-Kobayashi-Maskawa matrix, 5
Cherenkov effect, 28
CKM matrix, *see* Cabibbo-Kobayashi-Maskawa matrix
classification
    efficiency, 56
    purity, 56
COG algorithm, 36
connection strength, *see* weight
conscience mechanism, 16
converted photons, *see* photon conversions
counting method, 59
CP violation, 6
CPT, 7

Dalitz decays, 46
DBPIT, 79
$\mathrm{d}E/\mathrm{d}x$, 42
DELANA, 30
DELPHI, 25
    barrel, 25
    forward, 25
    picture, 26
DELSIM, 30
delta-rule, 15
    generalised, 19
dilepton, 9
DST, 30
DSTFIX, 30

electron identification, 42
    loose, 44
    standard, 44
    tight, 44
electroweak mixing angle, 4
ELPROB, 43
EMMASS, 40

FADC, 32
FCA, 27
FCB, 27
FEMC, 28
fermion, 3
Fisher discriminant, 20
flip probability, 9
force carriers, 4

Gaussian classifier, 20
generalisation performance, 23
gluon, 4
graviton, 5

HAC, 28
HCAL, *see* HAC
heavy flavour physics, 5
heavy quark tagging, 6
Higgs, *see* boson Higgs
HOF, 29
HPC, 28
hyper-plane, 20

ID, 31
IGSW, 61

jet charge, 53
JETFS, 35
JET HV, 35
JETTM, 32

Kolmogorov's neural network existence theorem, 20

Landau fluctuations, 36
LAYTM, 32
learning, 13
    auto-associative, 24
    coincidence, 13
    competitive, 13
    Hebb, 13
    Kohonen, 15
    LMS, 15
    performance, 13
    rate, 15
    Widrow-Hoff, 15
lepton, 3
linear associator, 13
LTD, 31

mean-square error measure, 15
mip, *see* minimum ionising particle
minimum ionising particle, 42
MUB, 29
MUF, 29
MUFIX, 40
MUFLAG, 40
muon
  identification, 39
    loose, 41
    standard, 41
    tight, 41
  punch-through, 39
  sail-through, 39
MWPC, 24, 32

neural network, 11
  architecture, 11
  error surface, 19
  feed-forward, 17
  layers, 11
  output, *see* neural network response
  pruning, 71
  target, 13
  temperature, 19
neuron, *see* node
NIKLIB, 30
NNC, 65
nn^{min}, 56
nn output, 56
node, 11
  activity, 18
  binary threshold, 18
  fan-in to, 22
  hidden, 11
  input, 11
  local memory, 11
  output, 11
  picture, 12
  transfer function, 11

OD, 27

PB, 46
PC, 46
p_{comb}, 52
p_{dil}, 52
perceptron, 18
PHDST, 30
photon, 4
  conversions, 43
processing element, *see* node
PRESA, 35
PRESJ, 35
p_t, 45
p_t^{in}, 45
p_t^{min}, 49
p_t^{out}, 45

QCD, 3
QED, 3
Q_{jet} and $|Q_{jet}|$, *see* jet charge
quark, 3

R, 9
radial sprouting, 16
raw data, 30
receptive field, 71
RICH, 28

SAT, 29
$\underline{\mathrm{SC}}$, 46
$\overline{\mathrm{SC}}$, 46
self organisation, 13
sigmoid function, 19
sphericity axis, 53
spontaneous symmetry breaking, 4
Standard Model, 3
$\mathrm{SU}_C(3)$, 3
$\mathrm{SU}_L(3)$, 3

T1, T2, T3 and T4, *see* trigger levels
TDC, 31
TEMP, 35
test set, 23
TF-chip, 32
threshold function, *see* activation function
TL HV, 35
TOF, 29
TPC, 27
tracking detectors, 25
training, 13
  epoch, 74
  graded, 13
  over-, 23
  parameter settings, 22
  regime, 13
  self-organised, 13
  set, 13
  supervised, 13
transfer function, 11
transverse momentum, *see* p_t
trigger, 29
  ID-OD, 32
  levels (T1, T2, T3 and T4), 30

$\mathrm{U}_Y(1)$, 3
unit, *see* node

VCYL, 35
VD, 25
vector coupling, 4
VSAT, 29

$W^\pm$, *see* boson $W^\pm$
weight, 11
  decay, 22

Z^0, *see* boson Z^0

Summary

A Measurement of the B^0-$\bar{B}^0$ Mixing Parameter at LEP Using a Neural Network

In this thesis the B^0-$\bar{B}^0$ mixing parameter χ is measured. The data have been collected using the DELPHI detector at the electron-positron accelerator LEP at CERN in Geneva. At the LEP energy of about 91 GeV the Z^0 particle is produced. About 15 percent of the time the Z^0 decays into a $b\bar{b}$-pair, which makes LEP an ideal environment to study the properties of the heavy b quark.

The B^0 particle is a bound state between a b quark and a lighter quark. The electroweak part of the Standard Model predicts that the B^0 can transform before it decays into its anti-particle the $\bar{B}^0$, and vice versa. This phenomenon is called B^0-$\bar{B}^0$ mixing. The B^0-$\bar{B}^0$ mixing parameter χ represents the chance that the transformation occurs.

In this thesis, the signal for the measurement of χ consists of events in which there are two leptons in the final state. If both leptons directly originate from a b quark decay ($b \rightarrow l$), then their charge reflects the one of the b quark. Events with leptons of the same sign indicate the presence of B^0-$\bar{B}^0$ mixing.

Leptons originating from other decays constitute the background to the mixing measurement. The standard method to select direct leptons is to use the transverse momentum of the lepton, because direct leptons have on average a larger transverse momentum.

In the measurement of this thesis a back-propagation feed-forward neural network is employed to separate direct leptons from the background. The strength of neural networks lies in their ability to optimally exploit the correlation between variables. Hence, the neural network can construct a new and effective variable which could not have been constructed by hand.

Back-propagation networks are trained by example. In this measurement the examples consist of simulated decays. The network is trained with direct leptons and the difficult background of leptons from the secondary decay $b \rightarrow c \rightarrow l$.

The neural network variable achieves a better separation between the signal and the background then the transverse momentum. Using data recorded by DELPHI in 1992, one obtains for the mixing parameter $\chi = 8.6\% \pm 2.3\%(stat) \pm 0.6\%(sys)$. The statistical error dominates.

Samenvatting

Een Meting van de B^0-$\bar{B}^0$ Mengparameter bij LEP met behulp van een Neuraal Netwerk

In dit proefschrift wordt de B^0-$\bar{B}^0$ mengparameter χ gemeten. De meetgegevens zijn verzameld met behulp van de DELPHI detector bij de elektron-positron versneller LEP op het CERN te Genève.
In de LEP versneller wordt bij een energie van ongeveer 91 GeV het Z^0 deeltje geproduceerd. In een groot aantal gevallen ($\approx 15\%$) vervalt het Z^0 deeltje in een $b\bar{b}$-paar, zodat LEP uitermate geschikt is om eigenschappen van het zware b quark te bestuderen.

Het B^0 deeltje bestaat uit een b quark en een lichter quark. Het elektrozwakke gedeelte van het Standaard Model voorspelt dat de B^0 en zijn anti-deeltje de $\bar{B}^0$ in elkaar kunnen transformeren voordat ze vervallen. Dit proces wordt B^0-$\bar{B}^0$ menging genoemd en de B^0-$\bar{B}^0$ mengparameter χ vertegenwoordigt de kans dat dit plaatsvindt.

In dit proefschrift wordt het signaal voor de meting van χ gevormd door gebeurtenissen waarin zich twee leptonen (elektronen of muonen) in de eindtoestand bevinden. Als beide leptonen direkt van een b quark verval ($b \rightarrow l$) afkomstig zijn, reflecteert hun lading die van het b quark. Een gebeurtenis met twee gelijk geladen leptonen wijst op B^0-$\bar{B}^0$ menging.

Een achtergrond wordt gevormd door leptonen die via andere vervalswijzen geproduceerd worden. De standaard methode om direkte leptonen te selecteren maakt gebruik van de transversale impuls van het lepton, een variabele die gemiddeld groter is voor direkte leptonen.

In dit proefschrift wordt een voorwaarts verbonden neuraal netwerk gebruikt dat met behulp van achterwaartse foutpropagatie (*back-propagation*) geleerd wordt om direkte leptonen te scheiden van de achtergrond. De kracht van neurale netwerken is dat ze verbanden tussen variabelen optimaal kunnen benutten. Daardoor kan het netwerk een nieuwe effectieve variabele construeren, die met de hand niet gevonden had kunnen worden.

Een neuraal netwerk wordt getraind met behulp van voorbeelden. De voorbeelden zijn in deze meting gesimuleerde vervallen. Het network wordt getraind met direkte leptonen en een prominente achtergrond bestaande uit leptonen van het secundaire verval $b \rightarrow c \rightarrow l$.

Met behulp van de neurale netwerk variabele wordt een betere separatie tussen signaal en achtergrond bereikt dan met de transversale impuls. Gebruik makend van meetgegevens verzameld door DELPHI in 1992, wordt als resultaat gevonden $\chi = 8.6\% \pm 2.3\% \pm 0.6\%$, waarbij de eerste fout statistisch is en de tweede systematisch. De statistische fout domineert.

Acknowledgements

The experience of working together with scientists from different countries was very interesting. My gratitude goes to many people at CERN and NIKHEF of whom I can only mention a few.

At CERN, the atmosphere in the White Room contributed significantly to my creativity. This was definitely due to the interesting mix of PhD students whose stay somehow overlapped with mine: Erwin Agasi, Nichol Brummer, Nicolo de Groot, Stefan Haider, Ivar Siccama, (summer student) Wouter Schmitz and Hao Wei.

Happy and bad times for the Inner Detector I shared on-line with André Augustinus, Mark Dönszelmann, Zbyszek Hajduk, Krzysztof Korcyl and Jan Timmermans. Friendly help with database issues and computing I received from Yuri Belokopytov and Hans Klein.

The studies I did together with Nicolo the Groot in my first year at CERN were the starting point for the analysis presented in this thesis. It was very stimulating to work together with the other DELPHI neural network proponents: Alessandro de Angelis, Gabrielle Cosmo, Paola Eerola and Louis Lyons. I thank Lenick de Boeck, Peter Kluit, Marco Paganoni and Jan Timmermans for their practical help and comments at that time.

At NIKHEF-H, the members of the computer group provided perfect computing and network support. Especially I would like to mention René Wilhelm, who kept our calculating prodigy *freya* running at all times.

I thank all my DELPHI colleagues at NIKHEF-H. Dick Holthuizen provided the tape handling at SARA and I am grateful to Werner Ruckstuhl for developing and maintaining the indispensable analysis backbone NIKLIB. I'm very grateful to Ivar '.kumac' Siccama for all his practical help, philosophical discussions and sharing one of my biggest hobbies: film.

My special thanks go my promotor Bert Diddens and my co-promotor Jan Timmermans, for all their patience, help and careful reading of this thesis.

After work at CERN, Ujjwal Vikas provided the necessary diversions. Our discussions and adventurous voyages were and will be irreplaceable. Pratibha Vikas and Riswan Khan I thank for their hospitality during the various occasions I revisited CERN.

Life-lines to Amsterdam via e-mail were provided by: Erwin Blekkenhorst, Cor van Dusschoten, Kenneth Gilhuijs and Peter van de Ven. Furthermore, I thank Mike Nielsen at Philips Melbourne for many interesting discussions.

Back in Amsterdam, I am very grateful to my good friend and squash partner Marco Adriaanse for keeping me reasonably fit, especially in the last busy months.

My old university friends Martin van Aken, Kenneth Gilhuijs, Wim Peters and Peter van de Ven, I thank for their occasional but always stimulating company.

Finally, I would like to thank my family very much for all their love, encouragement and support over the last years.

Curriculum Vitae

De auteur van dit proefschrift werd geboren op 27 juni 1967 te Heemskerk. Na het behalen van zijn diploma Gymnasium β aan het Petrus Canisius College te Alkmaar, begon hij in 1985 met de studie natuur- en sterrenkunde aan de Universiteit van Amsterdam.

Na het propedeutisch examen in 1986, behaalde hij in 1990 het doctoraal examen informatica in de natuurkunde. Het afstudeerwerk werd verricht aan het 'Nationaal Instituut voor Kernfysica en Hoge Energie Fysica' (NIKHEF-H) te Amsterdam onder leiding van Dr. J.J.M. Timmermans. Dit leidde tot een scripte getiteld "Testdata analysis of the Trigger Layers of DELPHI's Inner Detector".

In 1990 begon de auteur, in dienst van de stichting Fundamenteel Onderzoek der Materie (FOM), zijn aanstelling als wetenschappelijk medewerker (OIO) verbonden aan NIKHEF-H.

In het kader van zijn onderzoek verbleef hij ruim twee jaar in het CERN te Geneve. Gedurende deze periode bezocht hij twee keer het prachtige eiland Elba voor de workshop 'Neural Networks: from Biology to High Energy Physics'.

Vervolgens keerde de auteur terug naar het NIKHEF-H om zijn onderzoek te finaliseren. De resultaten van het onderzoek dat hij in de bovenstaande periodes verrichtte zijn voor het merendeel vervat in dit proefschrift.