DISRUPTION PREDICTION AT JET

FEDERICO MILANI

Doctor of Philosophy

THE UNIVERSITY OF ASTON IN BIRMINGHAM

December 1998

Thesis summary

The sudden loss of the plasma magnetic confinement, known as disruption, is one of the major issue in a nuclear fusion machine as JET (Joint European Torus). Disruptions pose very serious problems to the safety of the machine. The energy stored in the plasma is released to the machine structure in few milliseconds resulting in forces that at JET reach several Mega Newtons. The problem is even more severe in the nuclear fusion power station where the forces are in the order of one hundred Mega Newtons.

The events that occur during a disruption are still not well understood even if some mechanisms that can lead to a disruption have been identified and can be used to predict them. Unfortunately it is always a combination of these events that generates a disruption and therefore it is not possible to use simple algorithms to predict it.

This thesis analyses the possibility of using neural network algorithms to predict plasma disruptions in real time. This involves the determination of plasma parameters every few milliseconds. A plasma boundary reconstruction algorithm, XLOC, has been developed in collaboration with Dr. D. O'Brien and Dr. J. Ellis capable of determining the plasma wall/distance every 2 milliseconds. The XLOC output has been used to develop a multilayer perceptron network to determine plasma parameters as ℓ_i and q_ψ with which a machine operational space has been experimentally defined. If the limits of this operational space are breached the disruption probability increases considerably.

Another approach for prediction disruptions is to use neural network classification methods to define the JET operational space. Two methods have been studied. The first method uses a multilayer perceptron network with softmax activation function for the output layer. This method can be used for classifying the input patterns in various classes. In this case the plasma input patterns have been divided between disrupting and safe patterns, giving the possibility of assigning a disruption probability to every plasma input pattern.

The second method determines the novelty of an input pattern by calculating the probability density distribution of successful plasma patterns that have been run at JET. The density distribution is represented as a mixture distribution, and its parameters are determined using the Expectation-Maximisation method. If the dataset, used to determine the distribution parameters, covers sufficiently well the machine operational space, then, the patterns flagged as novel can be regarded as patterns belonging to a disrupting plasma.

Together with these methods, a network has been designed to predict the vertical forces, that a disruption can cause, in order to avoid that too dangerous plasma configurations are run. This network can be run before the pulse using the pre-programmed plasma configuration or on line becoming a tool that allows to stop dangerous plasma configuration.

All these methods have been implemented in real time on a dual Pentium Pro based machine. The Disruption Prediction and Prevention System has shown that internal plasma parameters can be determined on-line with a good accuracy. Also the disruption detection algorithms showed promising results considering the fact that JET is an experimental machine where always new plasma configurations are tested trying to improve its performances.

Acknowledgements

This work has been carried out with the essential support of many people who helped me with their knowledge, their time and their friendship. It is my pleasure to express first my gratitude to:

my supervisor at Aston University Prof. Christopher M. Bishop, who inspired and directed my work with constant encouragement and patience;

Dr. Massimo Garribba, Dr. Stefano Puppin, Dr. David J. Campbell, Dr. Ben Tubbing and Dr. Morten Lennholm, who guided me and shared their experience with me;

Dr. Denis O'Brien and Dr. Julian Ellis for the fruitful collaboration for the development of the XLOC algorithm;

I also want to thank the following friends and colleagues for many fruitful discussions, their suggestions, and their criticisms:

Shakeib Ali Arshad, Maria Elena Angoletta, Enzo Bertolini, Tiziano Businaro, Marco Buzio, Peter Card, Susana Clement, Jean Christophe Desire, John Farthing, Geoff Fishpool, Mike Johnson, Paul Mc Cullen, Peter Noll, Valeria Riccardo, Fernanda Rimini, Loris Rossi, Gabriella Saibene, Filippo Sartori, Roberta Sartori, Geoge Sips, Robert Squibb, Arturo Tanga, John Wesson, Wolfang Zwingmann.

A special thanks to Stefania who constantly encouraged and supported me during this long period.

This work has been funded by a JET Fellowship grant awarded by the European Commission.

Table of Contents

List of Figures

List of tables

1. Introduction

1.1 Fusion Research

The world's dependence on fossil fuels as energy sources is a huge burden on our ecosystem. The problems associated with burning fossil fuels are too pressing to ignore. Degradation of the environment through pollution of the atmosphere, global warming, and depletion of the limited natural sources are reasons enough to look for sources that are non-polluting.

Solar, nuclear fission and nuclear fusion energy are possible alternatives. Solar energy has the advantage that it is unlimited and non-polluting, however the efficiency of current photo cells is less than 20 percent. Other problems are the energy storage for use during the night and the effect of large solar cell arrays on the environment.

Nuclear fission is a proven way of energy source even with the controversial use of breeder reactors. However there are the still unsolved problems of radioactive contamination from possible reactor accidents and from the basically unanswered question of nuclear waste disposal.

Nuclear fusion may be an almost unlimited supply of energy. Fusion has many advantages: the fuel for fusion is practically unlimited and the problem of radioactive contamination is less severe than in fission. A fusion reactor is an inherently safe system since any malfunction results in a rapid shutdown. The main fuel tritium is only slightly radioactive with a half-life of 12.4 years. The waste products of the fusion process are non-radioactive helium isotopes. However, the reactor itself becomes contaminated, but after 100 year the radioactivity level

will be comparable to a conventional thermal power station, provided that low activation materials are used for their construction. Overall, the advantages of fusion power at least justify its development as a long term alternative energy source.

1.1.1 Nuclear Fusion

The inexhaustible energy sources rely fundamentally on energy derived from nucleus of the atom. Large amounts of energy can be released when light nuclei are fused. In thermonuclear fusion reactions two light nuclei, carrying a positive charge, have to overcome their mutual repulsion, to form a heavier nucleus. Thus, sufficient energy has to be given to the nuclei so that they fuse together. The most favourable reaction is between the two isotopes of hydrogen deuterium (D) and tritium (T). When these two isotopes fuse together the mass deficit between the fuels and the reaction products is released in form of energy:

$$ {}_1^2D + {}_1^3T \rightarrow {}_2^4He + {}_0^1n + 17.59\ MeV \tag{1.1} $$

Deuterium is widely available in sea water and every litre contains about 33 mg of deuterium, which, according to the above equations, has the same energy as 360 litres of petrol. Tritium can be bred in a fusion reactor from the light metal lithium () using the neutron induced fission reaction:

$$ \begin{aligned} {}_3^6Li + {}_0^1n &\rightarrow {}_1^3T + {}_2^4He + 4.8\ MeV \\ {}_3^7Li + {}_0^1n &\rightarrow {}_1^3T + {}_2^4He + {}_0^1n - 2.5\ MeV \end{aligned} \tag{1.2} $$

A sufficiently high reaction rate requires a temperature of approximately 100 million K or equivalently 10 keV. To control plasmas at this temperature two different approaches has been developed: inertial confinement and magnetic confinement.

In inertial confinement a solid pellet is irradiated with intense, but very short, pulses of energy such as lasers, electrons and ion beams and the fusion conditions are created on implosion of the fuel before the material expands.

In the magnetic confinement scheme, hydrogen is heated to high enough temperatures so that the particles overcome the repulsive Coulomb potential. This hot ionised gas is called plasma. Magnetic confinement of this extremely hot ionised gas is necessary to sustain the

plasma, since any impurity insertion could cool down the plasma and reduce its confinement time.

Among the various possible magnetic configurations the most promising approach appears to be the Tokamak.

1.1.2 Tokamak

The Tokamak is the most widely and most successful device used to obtain thermonuclear fusion in a controlled way. It was designed in the early 1950's by A.D. Sakharov and I. E. Tamm at the Kurchatov Institute in Moscow. Its name is derived from the acronym of the Russian words "**TO**roidalnaya **KA**mera I **MA**gnitnaya **K**atushka", i.e. toroidal chamber and magnetic coil [1].

The magnetic configuration of a Tokamak can be seen in Figure 1.1.

Figure 1.1: Tokamak Magnetic Configuration, [2]

In the Tokamak the magnetic field lines are bent into the shape of a torus; the magnetic field consists of the combination of a toroidal field supplied by external toroidal coils and a poloidal field created by the toroidal plasma current itself.

The current in the plasma is induced by the current injected in the central solenoid. This acts like the primary winding of a transformer creating a flux change through the torus. The flux induces a toroidal electric field that drives the charged particles of the plasma. The toroidal current also heats the plasma through resistive heating.

Due to the radial decrease of the toroidal field the particles drift slowly towards the reactor wall. To obtain a stable plasma, it is therefore necessary to have a vertical poloidal field counteracting this drift. This field is generated by the poloidal field coil currents that determine the shape and the position of the plasma.

The combination of both fields produces a helical magnetic field geometry. The magnetic field lines form closed magnetic surfaces that do not intersect material surfaces. As a result the plasma particles move along the helical magnetic field lines in small gyrating orbits. This restricts their radial movement and provides particle confinement.

1.2 Joint European Torus (JET) Project [24], [25]

The JET Joint Undertaking is the world leading fusion research project supported by the European Atomic Energy Community (EURATOM), whose long term objective is the creation of safe environmentally sound prototype fusion reactors. At its heart is the Joint European Torus (JET), the largest Tokamak in the world.

The aim of the JET project is to study plasmas close to ignition conditions in order to investigate the feasibility of nuclear fusion as a source of energy.

JET started its operation in 1983 with deuterium plasmas; and the first D-T fuelling mixture was used in 1991 with only 10% tritium in the machine, resulting in an integrated fusion energy output of 2 MJ obtained over 2 seconds.

During the 1992-93 shutdown the JET Tokamak has undergone major changes. The new scientific programme aimed at divertor plasma configuration studies required the installation of four poloidal coils and a cryopump inside the vessel. This allows the development and study of more complex plasma configurations with favourable confinement properties and increased purity of the plasma.

The programme comprised the study of three different divertor configurations and a deuterium tritium campaign with 50-50 deuterium-tritium gas mixture. In November 1997 during the first period devoted to the Deuterium Tritium experimental campaign new world records have been established by JET: 21MJ of fusion energy; 16 MW of peak fusion power and a ratio of fusion power produced to net input power of 65%.

1.3 JET Structure [4], [23]

The main elements of the JET device can be summarised as follows:

- Mechanical structure
- Vacuum vessel
- Toroidal field coils circuit
- Transformer limb
- Inner and outer poloidal field coils circuit including divertor coils.

Figure 1.2: JET Apparatus.

The JET *vacuum vessel* is made up of eight sections or octants made entirely from Inconel. It has a D-shaped cross section to allow the creation of D-shaped elongated plasmas.

The iron *transformer limb* improves the magnetic coupling between the primary winding and the plasma current that can be seen as the secondary winding.

The *mechanical structure* is a metallic shell and weights about 460 tonnes. It surrounds the toroidal field coils and supports them against strong lateral forces resulting from the interaction with the poloidal field as well as with the toroidal field. It also restrains the vacuum vessel against electromagnetic forces.

The *toroidal field coils circuit* consists of 32 identical D-shaped, conventional copper wound, water cooled coils. The toroidal field coils create a toroidal magnetic field mainly responsible for the confinement of the plasma. The maximum field on the major radius of the torus is about 4.0 T.

Figure 1.3: Toroidal field coils

The *inner and outer poloidal field coils circuits* including *divertor coils* are used for the control of the plasma current, position and shape.

The poloidal field coils circuit is used to create a suitably configured poloidal magnetic field to control the plasma current and the plasma position and shape. The circuit is composed of 2×4 coils symmetric to the equatorial plane and the four new divertor coils installed during the 1992-1993 shutdown.

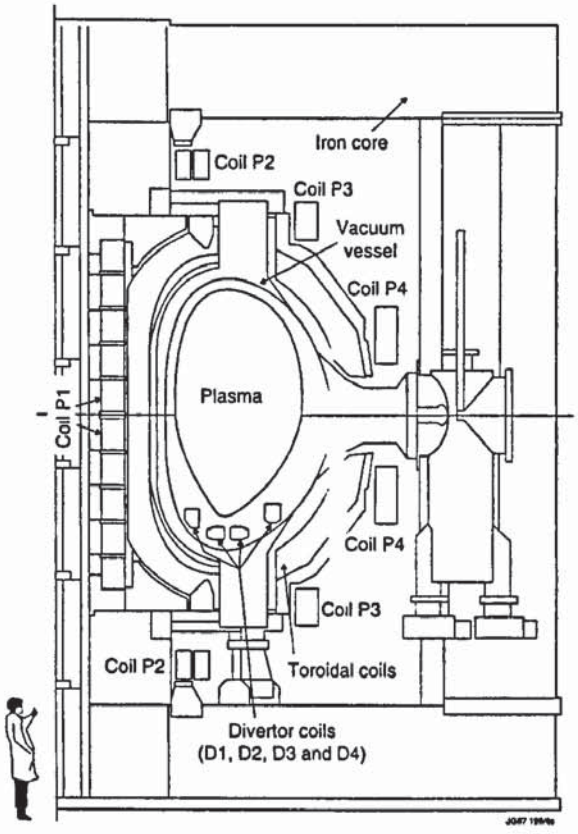

Figure 1.4: JET Poloidal Cross Section

All poloidal field coils are mounted outside the vessel except for the divertor coils.

The external poloidal field coils are usually called PF coils.

The PF coils can be grouped in different functional circuits such as

- *Transformer circuit*: P1_M is connected in series with P3_M. This circuit can basically be seen as the primary winding of the transformer that drives the plasma current.
- *Vertical field circuit:* P4_V generates essentially the uniform vertical field to compensate the radial hoop force of the plasma. Since these two coils are fed by two different amplifiers, it is possible to drive an imbalance current between the upper and lower coil to create a radial magnetic field to control the plasma vertical position.
- *Shaping field circuit:* P2_S and P3_S are connected in anti-series to create a quadrupolar shaping field for the generation of an elongated plasma. Whilst the current in P2_S has usually the same direction as the plasma current and therefore pulls the plasma outwards, the current in P3_S pushes the plasma inwards.

- *Radial field circuit:* P2_R and P3_R are series connected, whereby the current direction is anti-parallel in the upper and lower coils. These coils create a radial magnetic field used to control the vertical speed of the plasma. They are fed by the new fast radial field amplifier (FRFA).
- *The four divertor coils* are mounted on the inside bottom of the vessel. The introduction of these coils enables the creation of more complex plasma configurations with favourable confinement properties as well as very elongated plasmas. These configurations are known as X-point configurations. They will be described in the following section.

1.4 JET Heating Systems [24]

Besides ohmic heating, which becomes less efficient at higher temperatures because of a decrease in resistivity of the plasma, JET also has three systems of auxiliary heating. The Ion Cyclotron Radio Frequency (ICRF) heating system couples high power electromagnetic waves into the plasma with frequencies from 23 to 57 MHz. The charged plasma ions are accelerated by the electric field of the waves, then the plasma is heated up through collisions. Using this system, up to 22 MW of auxiliary heating power can be coupled into the plasma.

The Neutral Beam injection (NBI) injects into the plasma a beam of neutral particles. The atoms become ionised through collisions and transfer their kinetic energy to the plasma. With this method at JET hydrogen, deuterium, tritium or helium atoms can be injected either normal to the torus or tangentially. The total possible input power of the NBI system is 21 MW.

The Lower Hybrid system also couples electromagnetic waves into the plasma at a frequency of 3.7 GHz with a maximum power of 10 MW. This auxiliary heating system is not so much intended for heating, but to manipulate the electron velocity distribution resulting in a non-inductive current drive (LHCD). Several MA's of current can be driven by the LHCD system.

1.5 Principal JET Parameters

The principal parameters of the JET device are summarised in the following table.

Parameter	Design Value	Experimental Values
Plasma Major Radius, R_0	2.96 m	2.5 - 3.4 m
Plasma Minor Radius Horizontal, a	1.25 m	0.8 - 1.25 m
Plasma Minor Radius Vertical, b	2.1 m	0.8 - 2.1 m
Flat-Top Pulse Length	20 s	60 s
Toroidal Magnetic Field at R_0	3.45 T	4.0 T
Plasma Current (Limiter)	4.8 MA	7.1 MA
Plasma Current (X-Point)	-	6.1 MA
Neutral Beam Power	20 MW	21 MW
ICRH Power	15 MW	22 MW
LHCD Power	-	7.3 MW

Table 1.1: JET Principal Parameters

1.6 JET Operating Modes [24]

Under normal operating conditions the magnetic flux surfaces are nested inside each ot The edge of the plasma is defined by the magnetic surface that intersects the limiter. The magnetic field lines intersecting the walls of the chamber are those beyond the region bo by the limiters as shown in the diagram on the left. This is termed *limiter operation.*

The magnetic field configuration on JET can be modified by the shaping and the coils, so that one of the closed surfaces near the limiter is opened up to intersect vacuum vessel wall. In this configuration the magnetic separatrix is moved to w

vacuum chamber. This so called *x-point configuration* improved considerably the plasma confinement and purity.

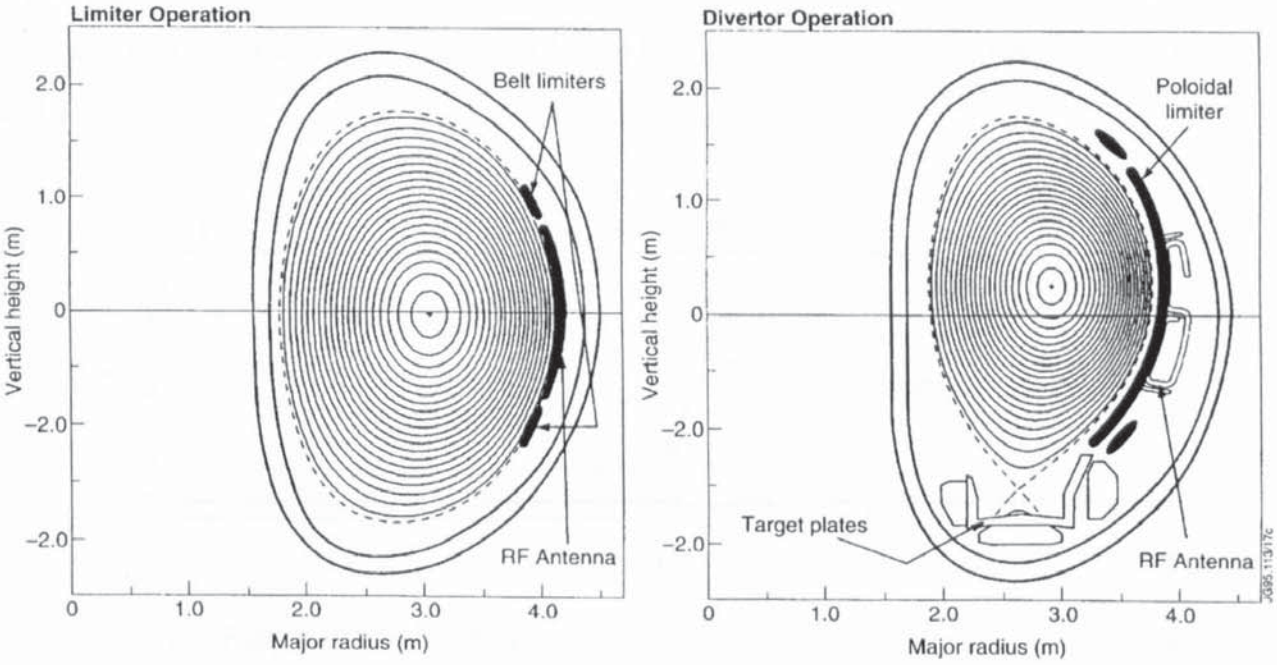

Figure 1.5: JET plasma operating modes: material limiter and X-point configurations.

The studies carried out during these years showed that the search of performance is usually stopped by the discovery of operational limits that leads to plasma disruptions when breached. During a plasma disruption the energy confinement is suddenly lost and all the plasma energy is transferred to the vessel and the machine components in few milliseconds, threatening the machine integrity itself.

Therefore, the study of plasma disruptions and how to avoid them plays an important role in the design of fusion reactors based on the tokamak concept.

1.7 Plasma Disruptions

The plasma disruption is a dramatic event in which the plasma confinement is suddenly destroyed. In a major disruption this is followed by a complete loss of the current as shown in Fig 1.6.

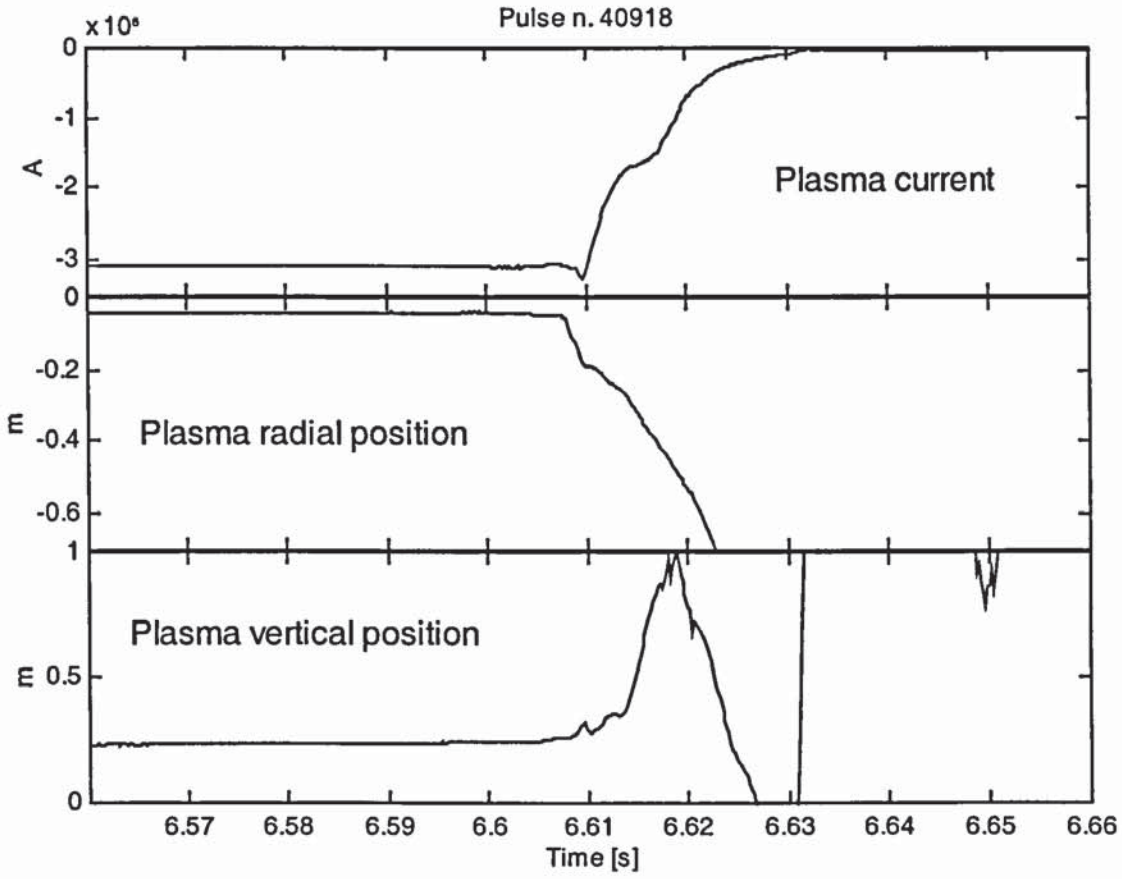

Figure 1.6: Plasma disruption. The sudden loss of the magnetic confinement is followed by the loss of the plasma current and position control. In few milliseconds the plasma moves towards the top inner wall releasing all its energy on the vacuum vessel structure.

Disruptions pose a serious problem for tokamak development. This firstly because they limit the range of operation in current and density, and secondly because their occurrence leads to large mechanical stresses and to intense heat loads.

The sequence of events that characterise a disruption can be summarised in four phases:

Pre-precursor phase: there is a change in the underlying conditions leading toward a more unstable configuration. This change is often clear, as in the increase in plasma density or in the radiated power. Sometimes however a disruption is precipitated without an identified change in the condition.

Precursor phase: when the underlying change reaches a critical point there is an onset of MHD instability. This is most obvious in the growth of the magnetic oscillations of an m=2 mode which had previously been saturated at low amplitude. The time during which the growth occurs is variable but is typically ~50 ms. Other low-m modes are also observed.

Fast phase: After the MHD instability has undergone considerable growth, a second critical point is reached at which much more rapid developments occur. The central plasma temperature collapses on a time scale typically of the order of milliseconds. A rapid flattening of the radial current profile is observed and this gives rise to a characteristic short negative voltage pulse, typically 10-100 times the applied positive voltage.

Quench phase: it is characterised by a sudden loss of the plasma energy, together with a very rapid drop in the electron temperature. The energy quench leaves the plasma in a highly resistive state due to the electron temperature drop. Since the current is initially hardly affected, the ohmic heating, I^2R, is enormously increased resulting in input power of the order of a gigawatt, this power is coming from the decay of the poloidal magnetic field energy. Finally the plasma current decays to zero transferring a substantial amount of the plasma current to the vacuum vessel leading to forces of hundreds of tonnes. The decay time depends upon the particular conditions and on the plasma position control but current decays as fast as 50MA s^{-1} occur.

Since there is at present no generally "complete" theory of plasma disruptions, the analysis of experimental results is mainly carried out in terms of the experimental operating conditions.

Four different types of disruptions can be identified: density limit, low q, β limit and vertical instability.

The density limit disruptions are the most studied disruptions since they pose a strong limit to the operation. The pre-precursor phase is determined by an increase of the density and a subsequent increase of the plasma radiated power. Unfortunately the density limit varies with

the plasma configuration and with the heating applied to the machine increasing when the plasma is heated by the additional heating systems.

The low q disruptions are much more well defined. The safety factor q, defined more precisely in chapter three, is the ratio between the toroidal and the poloidal rotations a magnetic field line does round the torus before to join up on itself. It has been proven that when q at the plasma surface is close to an integer number there could be the possibility of the arising of MHD instabilities with the growing of a mode that at the end locks resulting in a disruption. A strong limit has been found at q=2. When q at the surface is reaching the value at q=2 the formation of a mode is almost immediate and the subsequent disruption will occur in milliseconds.

The toroidal beta (β_T) is a basic MHD parameter measuring the confined pressure and, since there are MHD instabilities driven by pressure gradients, it is natural to expect that β_T will be subject to stability limitation. The high beta disruptions are related to an increase of the plasma pressure leading to MHD instability. It has been found experimentally that a β_T maximum value can be estimated. Using the empirical Troyon scaling it is possible to relate this β_T limit to the plasma current and the toroidal field:

$$\beta_{T,max} = g\frac{I_p}{aB_\Phi} \tag{1.3}$$

where I_p is the plasma current B_Φ is the toroidal field and a the plasma minor radius. The coefficient g is determined empirically. At JET it is very difficult to obtain the condition to reach the beta limit disruptions, since the plasma pressure usually does not reach high enough values to breach the β_T limit. Therefore the β_T limit disruptions will not be considered in this work.

The vertical instability disruptions are related to the fact that in order to increase the performance it has been discovered that high elongated plasmas are needed. These plasma configurations can be achieved by applying a quadrupolar field to the plasma. The field will generate a vertically unstable configuration. To stabilise these plasmas an ad hoc system has been implemented to control the vertical plasma speed. However, this system is limited in its capability of controlling the vertical speed and some times it fails due to its power limits. When this happens a vertical instability disruption occurs: the plasma vertical position is lost

and the plasma moves to the walls at a very high speed. It is possible to obtain in these cases plasma current decays up to 300 MA s^{-1}.

Disruption avoidance will be of critical importance for future fusion devices, such as the International Thermonuclear Experimental Reactor (ITER), for which the forces and heat loads caused by the disruption are expected to be 2 orders of magnitude larger than those in present day devices. In ITER, disruptions will threaten the structural integrity of the device, and their effects must be carefully evaluated in the machine design. A small number of disruptions at full operating parameters may ablate enough of the wall armour to require replacement of some areas. Thus, operation with a disruption rate much smaller than the disruptivity observed in present day machines will ultimately be required.

Two approaches to disruption prediction can be considered. The first is based on the detection of plasma instability, or precursors. For some types of disruption the precursors occur on a very short time scale of the instability growth rate, making it very difficult to take any effective action to suppress the instability and avoid a disruption.

The second approach, pursued in this study, is based on accurate mapping of the instability boundaries and avoidance of the unstable regions. The time resolution requirements for this scheme are much more relaxed, as the distance to the disruption boundary is estimated on the basis of the current state of the plasma, and the approach to the disruption is usually rather slow. However, traditional methods have not been successful in accurately mapping the disruption boundaries. This may be due to the apparent non linearity of the problem and the large number of plasma parameters that are required to describe the plasma state. Neural networks could provide a solution to this kind of problems. This work is devoted to the study and design of neural networks capable of mapping the complex nature of plasma disruptions.

1.8 Neural Networks for plasma disruption prediction

Neural networks offer a powerful set of tools for solving problems in pattern recognition, data processing, and non-linear control, which can be regarded as complementary to more conventional approach. Moreover, neural networks techniques are nowadays seen as fast non-linear methods that give the possibility of processing data in real-time every few milliseconds.

This work is focused primarily on the model known as the *multilayer perceptron*. It is part of the class of network models known as *feedforward* networks., which have been the subject of considerable research and application in recent years.

Neural networks represent a computational paradigm in which the solution to a problem is learned from a set of examples. The precise form of the transformation, from the input to the output space, is governed by a set of parameters called weights whose values can be determined on the basis of a set of examples of the required mapping. The process of determining these parameters is often called *learning phase.*

Neural networks has been chosen for mainly two reasons:

- They can be easily implemented in real time. The computational power needed to determine plasma parameters related to disruptions using the usual method does not allow a real time calculation. The neural networks give the possibility to recognise complex patterns in real time.
- The causes that generates a plasma disruption are so complex that it is not possible to use simple algorithms to determine when a disruption is going to occur. Therefore, the capability of neural network to catalogue events has been exploited to predict plasma disruptions.

The main disadvantage in using neural networks is the difficulty of determining the accuracy and reliability of the results. This usually leads to a certain scepticism on the use of neural networks for protection purposes as a disruption prevention system should be. Here, some methods are analysed in order to associate a confidence value to the results provided by the network. Mainly two methods have been used, both of them quantify the novelty of the input set compared to the data that has been used to train the network.

- Determination of the error bars on the neural network outputs. It is possible to evaluate the error bars on the output of a neural network using the information provided by the calculation of the Hessian matrix.
- Use of novelty detection techniques to determine if the parameters present certain anomalies which could lead to a plasma disruption.

1.9 The multilayer perceptron network

The multilayer perceptron network (MLP) is one of the most widely used class of neural networks due to its capability to represent any arbitrarily complex function between the multiple dimensional input and output spaces. Simpler techniques for representing non-linear mapping in multiple dimensional spaces as polynomial curve fitting or any kind of linear combination of fixed basis functions, can be considered as a sub-class of MLP networks. The multilayer perceptron networks are fairly simple algorithms to implement and not computationally demanding even if their training usually requires a long computing period off-line.

For this work MLP neural networks with two layers have been used for predicting plasma internal parameters (Chapter 3), vessel forces (Chapter 7), and for disruption classification (Chapter 5).

Mathematically MLPs can be represented as a linear combination of basis functions $z_j(\mathbf{x})$ also called hidden units, applied to the input vector $\mathbf{x}$:

$$y_k(\mathbf{x}) = \tilde{g}\left(\sum_{j=0}^{M} w_{kj}^{(2)} z_j(\mathbf{x})\right) \tag{1.4}$$

Here the hidden unit z_0 takes the fixed value 1 and allows a constant term in the expansion. The corresponding weight parameter w_{k0} is generally called bias.

The non linear function applied to the sum is used for the classification problem discussed in chapter 5. In the case of the prediction of plasma parameters or vessel forces the $\tilde{g}()$ is just the identity function.

As regard the hidden units, in a multilayer perceptron network they can be expressed in the form (1.4) in which the basis functions themselves contain adaptive parameters and are given by:

$$z_j(\mathbf{x}) = g\left(\sum_{i=0}^{d} w_{ji}^{(1)} x_i\right) \tag{1.5}$$

To allow the biases w_{j0} to be treated as the other weight parameters, an extra constant input $x_0=1$ has been introduced. If the weights w are fixed for all the z_j we are in the presence of a simple fixed basis function fitting.

The function $g(\cdot)$ is called activation function and must be non linear. A common choice for the activation function are the sigmoidal (S-shaped) functions and among these the tanh has been chosen:

$$g(a) = \tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}} \tag{1.6}$$

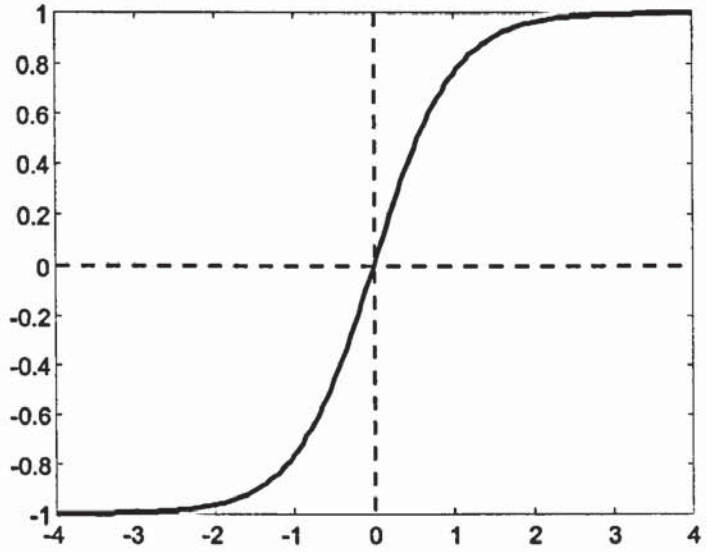

Figure 1.7: Plot of the tanh sigmoid function

This has the property, that its derivative can easily be expressed in terms of the function itself:

$$g'(a) = 1 - g(a)^2 \tag{1.7}$$

This property is very useful during the training process as we will see later.

A graphical representation of the network (Figure 1.8) can be used to understand the data flow. The inputs are multiplied by the first set of weights $w_{ji}^{(1)}$, the activation function is applied to sum to give the hidden functions as output (equation (1.5). Finally, these values are multiplied by the second set of weights $w_{kj}^{(2)}$, and the output activation function is applied to the sum to provide the output .

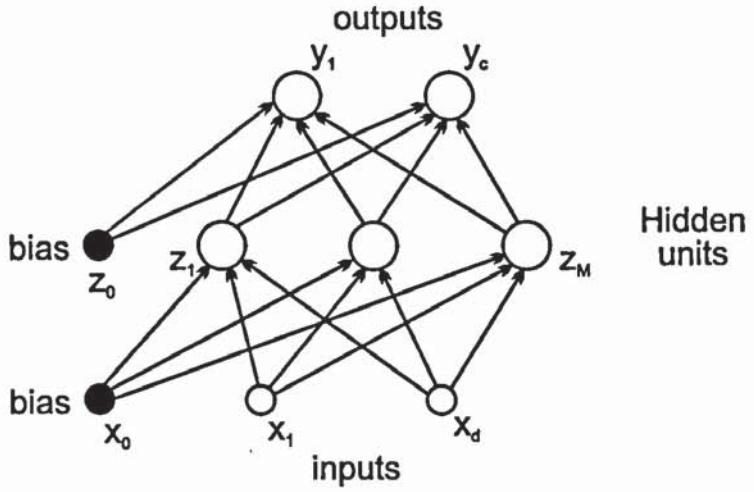

Figure 1.8: Representation of a two layer neural network.

After the definition of the network that we are going to use let us concentrate on the problem of determining suitable values for the weight parameters **w**.

This process, usually called training, involves the use of a dataset of input vectors $\mathbf{x}^n$ and the corresponding desired outputs $\mathbf{t}^n$ where n=1,..,N. t_k^n, represent the targets for the corresponding network outputs y_k. Then, the training of the network consists in the search for the weight parameters that results in the best approximation to the given set of target data. To do this there is the need of defining an error function to quantify the idea of best approximation.

A way to devise the error function is to use the concept of maximum likelihood. For a set of training data $\{\mathbf{x}^n, \mathbf{t}^n\}$, the likelihood can be written as:

$$\mathcal{L} = \prod_{n=1}^{N} p(\mathbf{t}^n | \mathbf{x}^n) \tag{1.8}$$

where $p(\mathbf{t}^n|\mathbf{x}^n)$ is the conditional density of the target data conditioned on the input data. We have also assumed that each data point $(\mathbf{x}^n, \mathbf{t}^n)$, is drawn independently from the same distribution, so that the likelihood for the dataset can be expressed as the product of the separate probabilities of each point. The idea is to maximise the probability of obtaining $\mathbf{t}^n$ given the input $\mathbf{x}^n$. Generally, it is more convenient to minimise negative logarithm of the likelihood. This can be done since the negative logarithm is a monotonic function. The negative logarithm of the likelihood usually is called error function, then we can write:

$$E = -\ln \mathcal{L} = -\sum_{n=1}^{N} \ln p(\mathbf{t}^n|\mathbf{x}^n) \tag{1.9}$$

We further assume that the individual target variables t_k^n (k=1..c) are independent so that the probability $p(\mathbf{t}|\mathbf{x})$ can be written as the product of the individual conditional densities $p(t_k|\mathbf{x})$.

To define a specific error function we must make some choice for the model of the distribution of target data. The simplest assumption is to take this distribution to be Gaussian. We assume that the target variable is given by some deterministic function of $\mathbf{x}$ with Gaussian added noise ε:

$$t_k = h_k(\mathbf{x}) + \varepsilon_k \tag{1.10}$$

we then assume that the errors ε_k have a normal distribution with zero mean, and standard deviation σ that does not depend on $\mathbf{x}$ or k:

$$p(\varepsilon_k) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{\varepsilon_k^2}{2\sigma^2}\right) \tag{1.11}$$

We now model the function $h_k(\mathbf{x})$ by a neural network with outputs $y_k(\mathbf{x};\mathbf{w})$ where $\mathbf{w}$ is the set of weight governing the neural network mapping. Using (1.10) and (1.11) we can write the probability distribution of the target variables as:

$$p(t_k|\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{\{y_k(\mathbf{x};\mathbf{w}) - t_k\}^2}{2\sigma^2}\right) \tag{1.12}$$

where the function $h_k(\mathbf{x})$ has been replaced by the neural network model $y_k(\mathbf{x};\mathbf{w})$. The equation above together with the definition of error function of equation (1.9) and with the assumption of independence of the variables t_k can be written as:

$$E = \frac{1}{2\sigma^2}\sum_{n=1}^{N}\sum_{k=1}^{c}\{y_k(\mathbf{x};\mathbf{w}) - t_k\}^2 + Nc\ln\sigma + \frac{Nc}{2}\ln(2\pi) \tag{1.13}$$

For the purpose of minimisation with respect to the weight parameters **w**, only the first term of the right hand side of the equation is important, leading in this way to the error expression known as sum-of square error.

This error function will be used in chapter 3 to design the network that will be responsible for the determination of plasma parameters (as the plasma internal inductance ℓ_i, beta poloidal, and the safety factor q at the plasma axis and at the plasma surface. The network discussed in chapter 5 is treated separately since it is used to classify disrupting plasmas. The meaning of the output is different, therefore, starting from equation (1.9) we will reach a different error function definition, more suitable for classification problems.

The process of training is then performed minimising these error functions using algorithms that relies on the gradient descent method. Thus, there is the need of determining the derivative of the error function with respect to the weights **w**. A very efficient way to differentiate the error function is called *back-propagation*. This procedure allows the calculation of the required derivative to be evaluated starting from the output units and going back to the first layer of hidden units.

Considering the output of the network we can write the equation (1.4) as:

$$y_k(\mathbf{x}) = \tilde{g}(\tilde{a}_k) \qquad \tilde{a}_k(\mathbf{x}) = \sum_{j=0}^{M} w_{kj}^{(2)} z_j(\mathbf{x}) \tag{1.14}$$

From equation (1.13) we can write the error function considering only the term that depends on **w** moreover we can consider each pattern separately writing it as $E = \sum_n E^n$ where E^n is:

$$E^n = \frac{1}{2}\sum_{k=1}^{c}\{y_k(\mathbf{x};\mathbf{w}) - t_k\}^2 \tag{1.15}$$

The derivatives of E^n can then be written as:

$$\frac{\partial E^n}{\partial w_{kj}^{(2)}} = \frac{\partial E^n}{\partial \tilde{a}_k}\frac{\partial \tilde{a}_k}{\partial w_{kj}^{(2)}} \tag{1.16}$$

The two derivatives on the right hand side of the equation can be easily solved using equation (1.14) and (1.15):

$$\frac{\partial \tilde{a}_k}{\partial w_{kj}^{(2)}} = z_j \qquad \frac{\partial E^n}{\partial \tilde{a}_k} = \tilde{g}'(\tilde{a}_k)\frac{\partial E^n}{\partial y_k} = \tilde{g}'(\tilde{a}_k)\{y_k - t_k\} \tag{1.17}$$

We then have obtained for the output layer a simple equation to differentiate E^n with respect of the weights. The equation becomes even simpler when we use as output activation function $\tilde{g}()$ the identity function. In this case the total error becomes:

$$\frac{\partial E^n}{\partial w_{kj}^{(2)}} = z_j(y_k - t_k) \tag{1.18}$$

As regard the hidden unit layer we can follow the same process to find the required derivatives:

$$\frac{\partial E^n}{\partial w_{ji}^{(1)}} = \frac{\partial E^n}{\partial a_j}\frac{\partial a_j}{\partial w_{ji}^{(1)}} \tag{1.19}$$

and remembering that

$$z_j(\mathbf{x}) = g(a_j) \qquad a_j(\mathbf{x}) = \sum_{i=0}^{d} w_{ji}^{(1)} x_i \tag{1.20}$$

we note that $\frac{\partial a_j}{\partial w_{ji}^{(1)}} = x_i$.

The derivatives with respect to the a_j can be simplified using the chain rule, and using equation (1.14) and (1.20) to find the relation between the a_j and the $\tilde{a}_j$:

$$\frac{\partial E^n}{\partial a_j} = \sum_{k=1}^{c}\frac{\partial E^n}{\partial \tilde{a}_k}\frac{\partial \tilde{a}_k}{\partial a_j} = g'(a_j)\sum_{k=1}^{c} w_{kj}^{(2)}\frac{\partial E^n}{\partial \tilde{a}_k} \tag{1.21}$$

where we can see that the derivatives can be estimated after that the derivative related to the output units are determined. It is also important to notice that the choice of using as activation function tanh eases the process of finding these second set of derivatives since the

term $g'(a_j)$ can be easily calculated from (1.7). If a linear activation function is used for the the output layer and a tanh for the hidden layer we obtain:

$$\frac{\partial E^n}{\partial w_{ji}^{(1)}} = (1-\tanh(a_j)^2)\sum_{k=1}^{c} w_{kj}^{(2)}\{y_k - t_k\} \tag{1.22}$$

Summarising, the back-propagation method efficiency relies on the fact that the error derivatives are determined backwards through the network. Moreover, using particular activation function the calculation of the error derivatives is reduced to just matrix products as it is shown in equation (1.18) and (1.22).

1.10 Disruption Prediction and Prevention System

One of the goals of this thesis is the implementation of these algorithm in real time. The Disruption Prediction and Prevention System will become the testing facility and the development system for plasma disruption avoidance algorithms. The system will have the capability to interact with JET control systems as the Plasma Position and Current Control and the Real Time Power Control systems to allow experiments where plasma parameters will be changed by DPPS to avoid disruption. The approval to perform these experiments will depend on the result of the first phase where the system will work just as a plasma disruption detection system.

The hardware chosen for this system is a dual Pentium PRO 200. The Operating System is Windows NT. Windows NT is not a real time system but preliminary test showed that its intervention is not affecting the performance when running with a sampling rate of few milliseconds as required by DPPS.

The integration of the system with JET computer network will be done by connecting it to a PC which will act as administrator using a fast Ethernet connection. The administrator will deal with the loading of the parameters and the storage of the data produced by DPPS.

Moreover a Reflective Memory Network has been set to connect DPPS to the main real time systems present at JET:

- PPCC plasma position and current control system which controls the PF coils current to obtain the desired plasma configuration.
- The Real Time Signal Server which is closely connected to the Real time Power control system and the Density feed back system. The density feedback system is controlling the gas injection in the vacuum chamber and as a consequence the plasma density. The real time power control system is in charge of delivering the additional heating power from the Lower Hybrid launcher the Radio Frequency and the Neutral Beam Injection systems.

The reflective memory network will allow the three systems to share data in real time, providing input data to DPPS. A further step would be the active role of DPPS on the JET plasma pulse where DPPS will request modification of plasma parameters to avoid disruptions.

1.11 This thesis

This thesis is written so that every chapter deals with a different algorithm used indirectly or directly to predict plasma disruptions.

The second chapter describes the XLOC algorithm developed to determine in real time the plasma boundary and its magnetic field. This information is then used as input to the neural networks designed in the following chapters to predict the occurrence of plasma disruptions.

The plasma prediction algorithm, studied in chapter three, has its basis in the experimental data observation done at JET. Using the internal plasma inductance ℓ_i and the safety factor at the plasma surface q_ψ, an operational space for the plasma configurations can be drawn. In this case, the difficulty of obtaining these parameters in real time, suggested the possibility of developing a neural network to overcome this problem. The chapter four a bayesian approach to the design of the same network has been considered to be able to add to the output of the network a confidence value that will tell how accurate the solution fund is.

In chapters five and six two methods have been studied to predict plasma disruptions using the neural network capability of classifying input patterns. In chapter five an MLP network with the softmax as activation function for the output layer has been used. In chapter six using

the philosophy of the novelty detection techniques, a network capable of mapping the JET operational space with a density probability has been designed. The input patterns that do not belong to the operational space are considered as dangerous.

Chapter seven is devoted to the prediction of the effect of a plasma disruption. A plasma pulse can be stopped if the vertical forces, that a plasma configuration can generate in case of disruption, are threatening the machine integrity. A MLP network that uses plasma parameters as input variables has been designed to predict the vertical forces. The training was done using the disruption database available at JET.

All the algorithms have been then tested on line using real time data. The first implementation of DPPS was used to test the feasibility of this system. The first provisional results are described in chapter seven.

2. XLOC algorithm

2.1 Introduction

The improvement of fusion performance at JET, relies heavily upon the knowledge and the control of the plasma position and shape. These parameters are inferred from magnetic measurements using algorithms based on the solution of the Grad-Shafranov equilibrium equation, which relates the plasma internal pressure with the external magnetic field. Unfortunately, the complexity of the solution of this differential equation does not allow a real time implementation of these full equilibrium reconstruction algorithms.

To overcome this problem, a method using Taylor series expansions of the magnetic flux surrounding the plasma has been developed in collaboration with Dr. D.O'Brien and Dr. J.Ellis. This algorithm called XLOC meets the requirements of determining the plasma boundary in less than 2 ms. A real time implementation is nowadays used by the plasma position and shape control system [12], giving the possibility to define directly the plasma shape using plasma/wall distances as feedback signals, instead of designing a plasma configuration using the PF coil currents as input to the control system [13], [14]. This not only eases the process of designing a pulse but also produces a plasma configuration much more stable to the variation of plasma *internal* parameters as ℓ_i or beta poloidal.

A further development of XLOC has been used as a data pre-processing algorithm for the inputs of the disruption prediction neural networks. XLOC will provide information not only about the plasma shape but also about the magnetic field close to the plasma surface. This pre-processing stage will protect the neural network from any faults of the magnetic sensors.

XLOC is very easy to reconfigure, allowing fast exclusion of malfunctioning sensors from its input. Thus, the neural network will not need to be retrained since the XLOC output will be not affected considerably by these changes.

In this chapter the XLOC algorithm is described showing its evolution during several years of design, to respond to the increasing demand for accuracy.

2.2 Poloidal Flux Function

In a tokamak the magnetic poloidal field configuration can be represented using the poloidal flux function [1]. The poloidal flux function assigns to each point of a cross section of the machine a magnetic flux value. Using cylindrical co-ordinates based on the major axis of the torus the flux function is defined as

$$\psi(R,Z) = \frac{1}{2\pi}\int_S \mathbf{B}_p(R,Z)\cdot \mathbf{n} ds \tag{2.1}$$

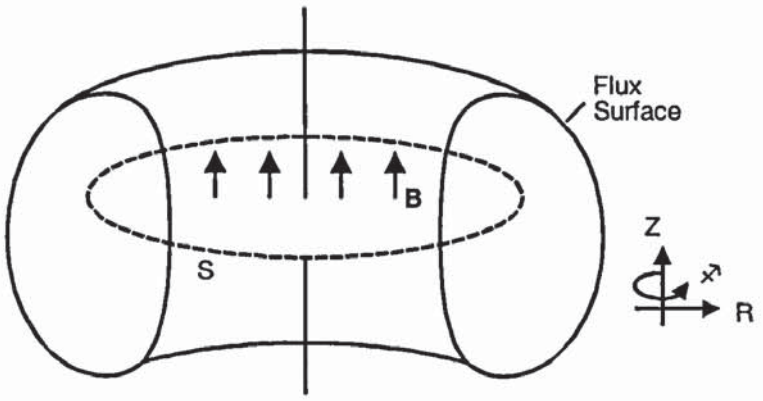

Figure 2.1: The flux is determined as the integral of the poloidal magnetic field on a circular surface S centred on the axis of the machine. An iso-flux surface is also shown.

From equation (2.1) we can see that ψ(R,Z) corresponds to the flux of the poloidal magnetic field $\mathbf{B}_p$, determined on a circular horizontal surface centred on the machine axis. R is the radius of the S circular surface and Z is the height of its centre.

We can also express the poloidal magnetic field radial and vertical components as partial derivative of the flux function:

$$B_R = -\frac{1}{R}\frac{\partial \psi}{\partial Z} \qquad B_Z = \frac{1}{R}\frac{\partial \psi}{\partial R} \tag{2.2}$$

On the plot below the various iso-flux surfaces are displayed on a cross section of JET. We can clearly see that the iso-flux surfaces are used to define the plasma magnetic configuration and particularly the plasma boundary. The plasma boundary is defined as the last closed flux surface inside the vessel. For plasma limiter configuration this line is tangent to the wall, but if the plasma is in x-point configuration, the last closed flux surface is the x-point surface. The x-point is an inflection point of the flux function situated inside the vacuum chamber. As a consequence, the representation of the flux function inside the vacuum chamber is sufficient to determine the plasma shape.

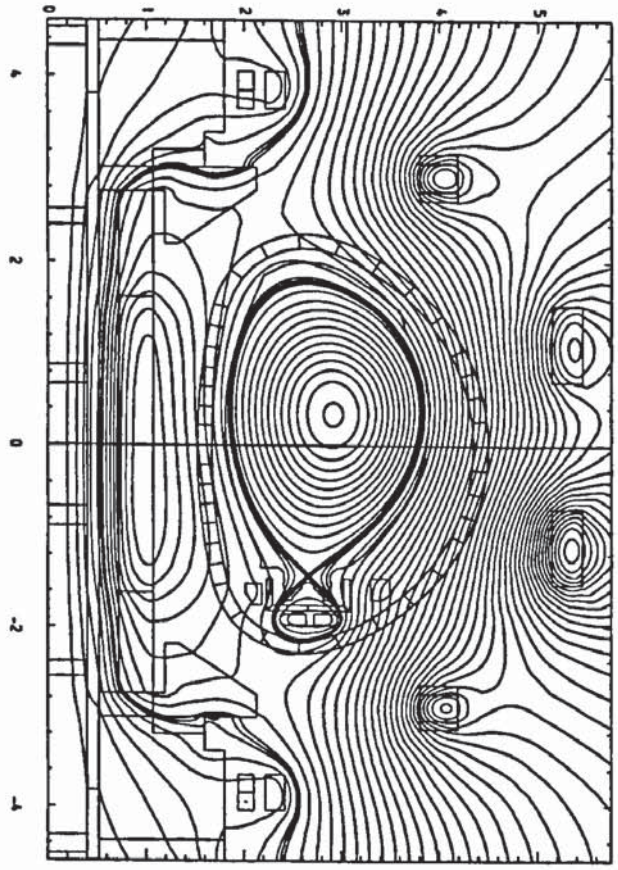

Figure 2.2: The iso-flux line for a cross section of JET. The plasma boundary is defined by the last closed flux line inside the vessel (fat line). The plasma x-point is generated by the divertor coil currents flowing in the same direction of the plasma.

The approach, commonly used to determine the plasma magnetic field configuration, is to solve the Grad-Shafranov equation. The solution requires long computational time (1 to 3 seconds per boundary reconstruction).

To allow a real-time estimation of the flux function the XLOC algorithm has been developed using a simpler approximation of ψ. To approximate ψ, XLOC uses five sixth order Taylor's expansions centred in five different points of the vacuum vessel:

$$\psi(\rho,z) = \sum_{i,j=0}^{i+j<6} a_{i,j}\rho^i z^j \tag{2.3}$$

where $\rho = \left(R^2 - R_0^2\right)$ $z = \left(Z - Z_0\right).\left(R_0, Z_0\right)$ are the coordinates of the flux expansion centre.

The vacuum chamber has been split in five regions (Figure 2.3) and for each region the set of $a_{i,j}$ coefficients (eq. (2.3)) is determined fitting the Taylor's flux expansions to the magnetic measurements available at JET.

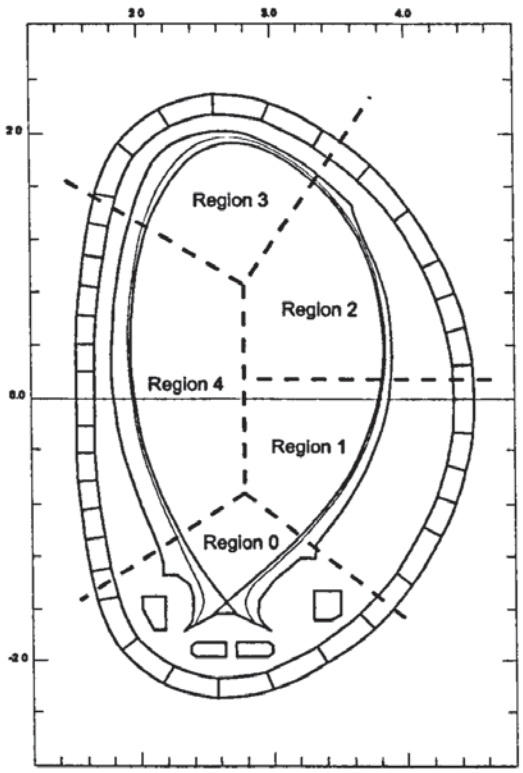

Figure 2.3: The five vessel regions have been selected so that the number of magnetic sensors available for each region is enough to determine the Taylor's expansion coefficients.

The regions where the flux extrapolation is more difficult are region 0 and region 3. This is caused by the possible presence of a magnetic field null. In the field null, the flux line closes

on itself with an angle of 90 degrees dividing the region in four parts. In each of these parts, the flux lines have a different curvature increasing the complexity of the flux line pattern to be estimated. In Figure 2.2 the x-point in region 0 can be seen.

In addition, since the x-point corresponds to a point of inflection of the flux function, its gradient is close to zero and small errors on the estimation of the flux could lead to large errors on the determination of the x-point coordinates, affecting severely the plasma boundary estimation.

2.3 JET Magnetic Measurements

To make possible the use of the magnetic sensors to determine the *a* coefficient we have to relate their measurements to the flux function. Three different types of magnetic measurements are made at JET:

Figure 2.4: Location of magnetic sensors. The saddle loops and the pick up coils are repeated for each octant. The flux loop on the contrary is a single loop wire and it is the only way to provide an absolute measurement of the flux function

2.3.1 Full flux loops

Each full flux loop consists of a wire running horizontally around the major axis of the torus. They measure the variation of the flux intercepted by the circular surface that they delimit. The signal is integrated providing the flux value in five different locations: two outside the vessel , three close to the divertor coils:

$$Full\ flux\ loop = \psi(R_F, Z_F) \tag{2.4}$$

2.3.2 Saddle loops

The saddle loops are one loop wire that covers a large poloidal section of a machine octant. They provide a signal proportional to the variation of the flux flowing through them. The signal is then integrated, multiplied by 8 and corrected by a factor that takes into account the area covered by the saddle. The final result is the flux difference between two horizontal surfaces centred on the major axis of the torus and positioned along the horizontal sides of the saddle:

$$Saddle\ flux\ loop = \psi(R_{S1}, Z_{S1}) - \psi(R_{S2}, Z_{S2}) \tag{2.5}$$

2.3.3 Poloidal pick up coils

The pick up coils are small discrete coils used to measure the poloidal field in 18 positions along the vessel, 7 positions on the outer poloidal limiter, and 22 positions in the divertor target area. Their measurement can be seen as a combination of the two flux derivatives. Using the equation (2.2):

$$Pick\ up\ Coil\ Magnetic\ Field = B(R,Z) = -\frac{1}{R}\frac{\partial \psi}{\partial Z}\sin(\theta) - \frac{1}{R}\frac{\partial \psi}{\partial R}\cos(\theta) \tag{2.6}$$

with θ pick up coil inclination and R and Z its coordinates.

Summarising, the total number of measurements, that can be used to fit the flux function Taylor expansions, is made of 5 flux loops, 27 saddle loops, and 47 pick up coils, making a total of 79 measurements. Comparing this number with the 140 coefficients to be determined,

it is clear that the system does not have a unique solution. We therefore, introduced further constraints to reduce the number of coefficients and to exploit the a priori knowledge that we have about the flux function behaviour. Two approaches can be used:

- To consider the plasma equilibrium equations available to find relations between the coefficients of a single flux expansion.
- To impose boundary constraints on two adjacent flux functions so that the values generated by the two flux functions is forced to be the same on their boundary.

In the next paragraphs the two different approaches are described. The main objective is to increase the accuracy of the approximation without increasing the complexity of the algorithm so much that it can not be implemented in real-time.

2.4 The Grad-Shafranov equation

The basic condition for equilibrium is that the force on the plasma has to be zero at all points. This requires that the magnetic force must balance the force due to the plasma pressure, that is in the simplest approximation:

$$\mathbf{j}\times\mathbf{B} = \nabla p \tag{2.7}$$

with **B** the total magnetic field, **j** the plasma current density and p the plasma pressure.

Starting from this equation and introducing the flux function and two arbitrary functions $p(\psi)$ and $f(\psi)$, the Grad-Shafranov equation can be written as [60] :

$$R\frac{\partial}{\partial R}\frac{1}{R}\frac{\partial \psi}{\partial R}+\frac{\partial^2 \psi}{\partial Z^2} = -\mu_0 R^2 p' - \mu_0^2 f f' \tag{2.8}$$

This equation represents the flux in all the vacuum vessel including the plasma itself. The p function gives the plasma pressure in each points as a function of ψ, and the f function is related to the current profile j via the following equations:

$$j_R = -\frac{1}{R}\frac{\partial f}{\partial Z} \qquad j_z = \frac{1}{R}\frac{\partial f}{\partial R} \tag{2.9}$$

The solution of the Grad Shafranov equation is usually computationally demanding, but if we reduce our model to the vacuum region outside the plasma, it can be simplified. Outside the plasma, the two functions p and f are equal to zero, then equation (2.8) depends only on ψ:

$$R\frac{\partial}{\partial R}\frac{1}{R}\frac{\partial \psi}{\partial R}+\frac{\partial^2 \psi}{\partial Z^2}=0 \tag{2.10}$$

Rewritten in ρ, z coordinates it becomes:

$$4\rho\frac{\partial^2 \psi}{\partial \rho^2}+\frac{\partial^2 \psi}{\partial z^2}=0 \tag{2.11}$$

The application of this equation to equation (2.3) leads to a set of 15 linear relations between the $a_{i,j}$ variables reducing our independent variables from 140 to 65. Each flux expansion can be rewritten as:

$$\psi(\rho,z)=\sum_{i=1}^{13} c_i \tilde{g}_i(\rho,z) \tag{2.12}$$

where the c_i are the new unknown variables. The parameters $\tilde{g}_i(\rho,z)$ are combination of the powers of the coordinates where the flux is determined.

It is useful to remember that a simple linear relation can be written between the two sets of coefficients:

$$\mathbf{a}=\mathbf{M}\mathbf{c} \tag{2.13}$$

The assumption that we reduce our analysis to the vacuum region does not affect the determination of the plasma boundary. Nevertheless, this strong limitation does not allow to use this flux function approximation to determine any *internal* plasma parameters. The plasma parameters relevant to disruption prediction will be determined using neural network algorithms as described in the next chapter.

2.5 c_i coefficient determination

The c_i are determined using the least square method to fit the magnetic measurements available. Their relation with the c_i coefficients can be written as:

Absolute flux measurement (Flux loop): In this case the equivalence between the measurement m and the ψ is immediate:

$$\psi(\rho,z) = \sum_{i=1}^{13} c_i \tilde{g}_i(\rho,z) = m \tag{2.14}$$

Relative flux measurement(Saddle loops): The saddle loops probes provide the flux flowing between two vessel points (equation (2.5)):

$$\psi(\rho_1,z_1) - \psi(\rho_2,z_2) = \sum_{i=1}^{13} c_i \tilde{g}_i(\rho_1,z_1) - \sum_{i=1}^{13} c_i \tilde{g}_i(\rho_2,z_2) = \sum_{i=1}^{13} c_i \left(\tilde{g}_i(\rho_1,z_1) - \tilde{g}_i(\rho_2,z_2)\right) \tag{2.15}$$

Magnetic field measurement(Pick up coils): Using equation (2.6) the field measured by the probes can be written as:

$$\mathbf{B} = -\frac{1}{R}\frac{\partial \psi}{\partial Z}\sin(\theta) - \frac{1}{R}\frac{\partial \psi}{\partial R}\cos(\theta) = -\frac{1}{R}\sum_{i=1}^{13} c_i \left(\frac{\partial \tilde{g}(\rho,z)}{\partial Z}\sin(\theta) + \frac{\partial \tilde{g}(\rho,z)}{\partial R}\cos(\theta) \right) \tag{2.16}$$

All the measurements described above can be expressed as a linear combination of the c coefficients with a term $g_i(\rho,z)$ which is related to the geometry of the measurement:

$$\mathbf{m} = \sum_{i=1}^{13} c_i g_i(\rho,z) = \mathbf{g}^T \mathbf{c} \tag{2.17}$$

If we write these equations for every available measurement we obtain a linear system of equations for each region:

$$\mathbf{G}^r \mathbf{c}^r = \mathbf{m}^r$$

$$\mathbf{G}^r = \begin{bmatrix} g_1^T \\ \dots \\ g_m^T \end{bmatrix} \quad \mathbf{c}^r = \begin{bmatrix} c_1 \\ \dots \\ c_{13} \end{bmatrix} \quad \mathbf{m}^r = \begin{bmatrix} m_1 \\ \dots \\ m_m \end{bmatrix} \tag{2.18}$$

with r number of the region of the flux expansion.

This system can be solved only if the number of equations is equal or bigger than the number of *c* coefficients.

If the number of measurements used is higher than the number of c variables, the least square method can be used to find the solution that minimises the difference between the calculated fluxes and the measured fluxes:

$$\min\left\{\left\|\mathbf{G}^r\mathbf{c}^r-\mathbf{m}^r\right\|^2\right\}=\min\left\{\frac{1}{2}\sum_{i=1}^{m}\left(\sum_{j=1}^{13}g_{i,j}^r c_j^r-m_i^r\right)^2\right\} \tag{2.19}$$

The minimum can be determined by finding the root of the partial derivatives of the norms with respect to the c_k^r coefficients:

$$\frac{d\left\|\mathbf{G}^r\mathbf{c}^r-\mathbf{m}^r\right\|^2}{d\,c_k^r}=\sum_{i=1}^{m}\left(\sum_{j=1}^{13}g_{i,j}^r c_j^r-m_i^r\right)g_{i,k}^r=0 \tag{2.20}$$

The resulting 13 equations are equivalent to the single matrix equation:

$$\mathbf{G}^{r^T}\left(\mathbf{G}^r\mathbf{c}^r-\mathbf{m}^r\right)=0 \tag{2.21}$$

which implies:

$$\mathbf{G}^{r^T}\mathbf{G}^r\mathbf{c}^r=\mathbf{G}^{r^T}\mathbf{m}^r \tag{2.22}$$

Thus, the solution of the system is given by:

$$\mathbf{c}^r=\left(\left(\mathbf{G}^{r^T}\mathbf{G}^r\right)^{-1}\mathbf{G}^{r^T}\right)\mathbf{m}^r=\mathbf{G}^{r^+}\mathbf{m}^r \tag{2.23}$$

where $\mathbf{G}^{r^+}$ is the so called pseudo-inverse.

The number of floating operations needed to solve the five systems is 2x13x*m*, with $m=\sum_{r=1}^{5}\dim\left(\mathbf{m}^r\right)$. This can easily be implemented in real-time, since the calculation can be performed in 80 μs using a digital signal processor TMS320C40 of Texas Instruments.

Unfortunately, the flux expansions determined solving these linear systems proved to be inaccurate mainly on the boundaries between the regions defined in Figure 2.3. Two adjacent

regions gave different flux values for the same point. To solve this problem, a new set of constraints has been introduced to link the five flux expansions.

2.6 Tie points

Up to now we have considered the flux function expansions separately. However we know that in reality they represent the same flux function and they should produce the same results for the same (R,Z) coordinates. Imposing this condition for a given set of points, we can link the flux expansions of two adjacent regions. Selecting the points on the region boundaries we will obtain the effect of forcing two adjacent regions to give the same results in those areas where the approximation is less precise.

There are two different ways of applying these constraints, called soft-tie point and hard tie point insertion.

2.6.4 Soft tie points

In this case we minimise the difference of the flux calculated in the two regions at a given set of points:

$$\min\left\|\psi^{r}(\rho_t,z_t)-\psi^{r+1}(\rho_t,z_t)\right\|^2 \quad\Rightarrow\quad \frac{\partial\left\|\mathbf{G}^{r}\mathbf{c}^{r}-\mathbf{G}^{r+1}\mathbf{c}^{r+1}\right\|^2}{\partial\mathbf{c}}=0 \tag{2.24}$$

This constraint can be applied easily adding a new set of equations to the systems above. We can solve the five equation systems together combining them in a single equation system where the G^r matrices are on the diagonal of the new **G** matrix, and the **c** vector is made by the combination of the *c* coefficients of the five expansions. Then, the tie points equations can be added to this system by simply adding an equation for each tie point:

$$\mathbf{G}=\begin{bmatrix}\mathbf{G}^1 & 0 & 0\\ 0 & \cdots & 0\\ 0 & 0 & \mathbf{G}^5\\ \mathbf{G}^r & -\mathbf{G}^{r+1} & 0\end{bmatrix}\qquad \mathbf{c}=\begin{bmatrix}\mathbf{c}^1\\ \cdots\\ \mathbf{c}^5\end{bmatrix}\qquad \mathbf{m}=\begin{bmatrix}\mathbf{m}^1\\ \cdots\\ \mathbf{m}^5\\ 0\end{bmatrix} \tag{2.25}$$

$$\mathbf{Gc} = \mathbf{m} \quad \Rightarrow \quad \mathbf{c} = \left(\left(\mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{G}^T\right)\mathbf{m} = \mathbf{G}^+\mathbf{m} \tag{2.26}$$

The columns of $\mathbf{G}^+$ that correspond to the soft-tie points can be omitted, since they are multiplied by 0. The number of operations needed to solve the system is 2x65xm, with $m = \sum_{r=1}^{5} \dim\left(\mathbf{m}^r\right)$. This, compared with the solution of the five separate equation systems, is just a factor of 5 higher and it can be solved in just 330 μs, leaving sufficient time to perform the other algorithms needed to determine the plasma boundary.

With regard to the selection of the tie points, this has been performed by aiming to increase the accuracy in the region borders. It has been discovered that by selecting the points close to the plasma boundary the overall performance increases considerably. In the current implementation of XLOC there are 31 soft tie points connecting the five regions, selected just inside the vacuum chamber, avoiding positions where they could go inside the plasma.

Using this constraint, the differences of the results produced by two adjacent flux expansions is minimised in a predefined set of points. This proved to be too weak for some plasma configurations. The difference between the two flux expansion values was still too large to be acceptable.

2.6.5 Hard tie points

The hard tie points are vacuum vessel positions where two touching flux expansions are forced to give the same results. For every hard tie point we can write the equation:

$$\psi^r\left(\rho_t, z_t\right) - \psi^{r+1}\left(\rho_t, z_t\right) = 0 \tag{2.27}$$

This constraint reduces the dimension of the equation system (2.25) imposing a relation between the c coefficients of different regions. To find the subset of linearly independent c coefficients we use an equation system where each row corresponds to a given hard tie point.

$$\begin{bmatrix} \mathbf{G}^1 & -\mathbf{G}^2 & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \mathbf{G}^4 & -\mathbf{G}^5 \end{bmatrix} \begin{bmatrix} \mathbf{c}^1 \\ \cdots \\ \mathbf{c}^5 \end{bmatrix} = \begin{bmatrix} 0 \\ \cdots \\ 0 \end{bmatrix} \tag{2.28}$$

From this system we extract a number of linearly independent c coefficient. Then, the relation between the new set of independent variables $\mathbf{c}_{red}$ and the c coefficients becomes:

$$\mathbf{c} = \mathbf{H}\mathbf{c}_{red} \tag{2.29}$$

where if h is the number of hard tie points then $\dim(\mathbf{H}) = 65\times 65 - h$, $\dim(\mathbf{c}) = 65\times 1$, and $\dim(\mathbf{c}_{red}) = 65 - h\times 1$. This relation is used to impose the new constraints to the equation system (2.25):

$$\mathbf{G}\mathbf{H}\mathbf{c}_{red} = \mathbf{m} \tag{2.30}$$

The c_{red} coefficients can be determined as before:

$$\mathbf{c}_{red} = \left(\left(\mathbf{H}^T\mathbf{G}^T\mathbf{G}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{G}^T\right)\mathbf{m} \tag{2.31}$$

The c coefficients will become:

$$\mathbf{c} = \mathbf{H}\left(\left(\mathbf{H}^T\mathbf{G}^T\mathbf{G}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{G}^T\right)\mathbf{m}$$

$$\mathbf{c} = \mathbf{X}\mathbf{m} \quad with \quad \mathbf{X} = \mathbf{H}\left(\left(\mathbf{H}^T\mathbf{G}^T\mathbf{G}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{G}^T\right) \tag{2.32}$$

At present 5 hard tie points are used to reduce the independent c coefficients to 60. These coefficients are calculated using 55 sensors, with 9 used in more than one region giving a total of 64 inputs.

Summarising:

1. A six order Taylor expansion has been used to approximate the flux function in small areas of the plasma region by dividing the cross section of the vacuum vessel in five parts and determining for each region a flux expansion.
2. The number of coefficients to be determined has been reduced from 28 to 13 using the Grad Shafranov equation. To reduce its complexity, the validity of the approximation has been reduced to the vacuum, thereby removing from the equation the terms relating to the plasma.
3. A set of tie points has been introduced to link the adjacent flux expansions and to impose the condition that their output values should be the same in the boundary region.

2.7 Plasma boundary determination

Knowing the flux function inside the vacuum chamber the plasma boundary can be easily determined. With regard to the plasma limiter configuration, the boundary is defined as the last closed flux surface inside the vacuum region. Mathematically, since the plasma centre is situated in a maximum of the flux function, the search of the plasma boundary can be done by finding the maximum value of the flux at the first wall. This is the last surface intersecting the first wall, and because it is tangent to the wall, it is also closed on itself, and by definition it is the plasma boundary flux surface.

As regard the x-point plasma configuration, the plasma boundary is defined by the flux surface of the x-point. The x-point is a flux function inflection point and can be calculated by finding the zeros of the flux function derivatives. The algorithm used to find this position is the Newton Raphson method. The value of the flux at the x-point is then compared to the maximum value of the flux at the wall. If the x-point flux is bigger, the plasma is in x-point configuration. Otherwise, it is in limiter configuration, despite the presence of the x-point inside the vacuum chamber.

On the following paragraphs the algorithms used to find the flux at the walls and the x-point flux are briefly described. It should be remembered that the aim in carrying out these calculations is to reduce the computational time needed to find the solution

2.7.6 Walls flux determination

Using the c coefficient it is relatively easy to determine the flux for a given point of the first wall:

$$\psi(\rho,z)=\sum_{i=1}^{13} c_i^r g_i^r(\rho,z) \tag{2.33}$$

The functions f can be calculated off-line reducing the computational power required. The flux is estimated for 80 points all around the vessel.

2.7.7 X-point determination

The method implemented to find the plasma x-point is the Newton Raphson method. We search for the zero of the flux expansion derivatives in the divertor region:

$$\frac{\partial \psi}{\partial R}=0 \qquad \frac{\partial \psi}{\partial Z}=0 \quad which\ means \quad B_R=0 \qquad B_Z=0 \tag{2.34}$$

The need of determining the first order and the second order partial derivatives of the flux function forces the use the 28 *a* coefficients, instead of the *c* coefficients. These are directly related to the power function of ρ and z, providing an easier way of calculating the flux functions derivatives.

Describing the flux function as a product, element by element of a matrix representing the geometric position with the *a* coefficients matrix, gives us the possibility of introducing a matrix derivative operator:

$$\psi(\rho,z)=\sum_{i,j=0}^{i+j<6} a_{i,j}\rho^i z^j \qquad \psi(\rho,z)=\begin{bmatrix} 1 & z & z^2 & \dots & z^6 \\ \rho & \rho z & \rho z^2 & \dots & 0 \\ \rho^2 & \rho^2 z & \dots & 0 & \dots \\ \dots & \dots & 0 & \dots & \dots \\ \rho^6 & 0 & \dots & \dots & 0 \end{bmatrix} \circ \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & \dots & a_{0,6} \\ a_{1,0} & a_{1,1} & \dots & \dots & 0 \\ a_{2,0} & \dots & \dots & 0 & \dots \\ \dots & \dots & 0 & \dots & \dots \\ a_{6,0} & 0 & \dots & \dots & 0 \end{bmatrix} \tag{2.35}$$

The symbol ∘ represents the matrix product element by element.

If we name the first matrix of (2.35) as **Y** and the second matrix of (2.35) as **A**, the derivatives of ψ can be written as:

$$\frac{\partial \psi(\rho,z)}{\partial \rho}=\{\mathbf{D}_\rho \mathbf{Y}\}\circ\mathbf{A} \qquad \frac{\partial \psi(\rho,z)}{\partial z}=\{\mathbf{Y}\mathbf{D}_z\}\circ\mathbf{A}$$

$$\frac{\partial^2 \psi(\rho,z)}{\partial \rho^2}=\{\mathbf{D}_\rho\mathbf{D}_\rho\mathbf{Y}\}\circ\mathbf{A} \qquad \frac{\partial^2 \psi(\rho,z)}{\partial z\partial \rho}=\{\mathbf{D}_\rho\mathbf{Y}\mathbf{D}_z\}\circ\mathbf{A} \qquad \frac{\partial^2 \psi(\rho,z)}{\partial z^2}=\{\mathbf{Y}\mathbf{D}_z\mathbf{D}_z\}\circ\mathbf{A} \tag{2.36}$$

with the matrix derivative operators defined as:

$$\mathbf{D}_\rho=\begin{bmatrix} 0 & \dots & \dots & \dots & 0 \\ 1 & 0 & \dots & \dots & \dots \\ 0 & 2 & 0 & \dots & \dots \\ \dots & 0 & \ddots & 0 & \dots \\ 0 & \dots & 0 & 6 & 0 \end{bmatrix} \qquad \mathbf{D}_z=\begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ \dots & 0 & 2 & 0 & \dots \\ \dots & \dots & 0 & \ddots & 0 \\ \dots & \dots & \dots & 0 & 6 \\ 0 & \dots & \dots & \dots & 0 \end{bmatrix} \tag{2.37}$$

The Newton-Raphson method is performed by calculating the equation:

$$\begin{bmatrix} \rho_{n+1} \\ z_{n+1} \end{bmatrix} = \begin{bmatrix} \rho_n \\ z_n \end{bmatrix} - \mathbf{J}^{-1} \begin{bmatrix} \dfrac{\partial \psi(\rho_n, z_n)}{\partial \rho} \\ \dfrac{\partial \psi(\rho_n, z_n)}{\partial z} \end{bmatrix} \tag{2.38}$$

with **J** the Jacobian matrix of the flux expansion derivatives. Its inverse can be written as

$$\mathbf{J}^{-1} = \frac{1}{\dfrac{\partial^2 \psi}{\partial \rho^2}\dfrac{\partial^2 \psi}{\partial z^2} - \left(\dfrac{\partial^2 \psi}{\partial \rho \partial z}\right)^2} \begin{bmatrix} \dfrac{\partial^2 \psi}{\partial z^2} & -\dfrac{\partial^2 \psi}{\partial \rho \partial z} \\ -\dfrac{\partial^2 \psi}{\partial \rho \partial z} & \dfrac{\partial^2 \psi}{\partial \rho^2} \end{bmatrix} \tag{2.39}$$

The number of iterations is fixed. The difference between the last two iterations will be considered as the error and checked against a given threshold to determine if the solution converged. If the error is too large the x-point is presumed not to be not present inside the vacuum chamber.

After the determination of the coordinate of the x-point the flux is easily calculated and compared with the maximum wall flux.

The first tests showed that the error is within millimetres after 4 iteration if the starting point is the x-point at the previous time step. With regard to the accuracy, XLOC has been compared with a full boundary reconstruction code EFIT giving an absolute error for the x-point determination of 5÷10 mm for a static equilibrium. If the equilibrium is evolving, especially if caused by fast changes in the divertor currents, this algorithm showed an error of several centimetres. This is mainly due to the fact that some measurements are influenced by the divertor coils flux. This "screens" the flux generated by the plasma itself and in areas close to the x-point, where the flux gradient is close to zero, small errors on the flux can yield large errors on the x-point position.

2.8 Divertor flux compensation

During the commissioning period of the plasma shape and position control system several experiments were done to prove the accuracy of XLOC. One of these tests was the control of the x-point position using the x-point position determined by XLOC as feedback signal. In the figure below it can be seen how the measurement followed accurately the reference. However, when the x-point was moved from the centre of the divertor region, XLOC started to produce wrong results. The comparison with the more accurate EFIT shows that XLOC was not able to find the x-point and as a consequence to find the magnetic field configuration correctly.

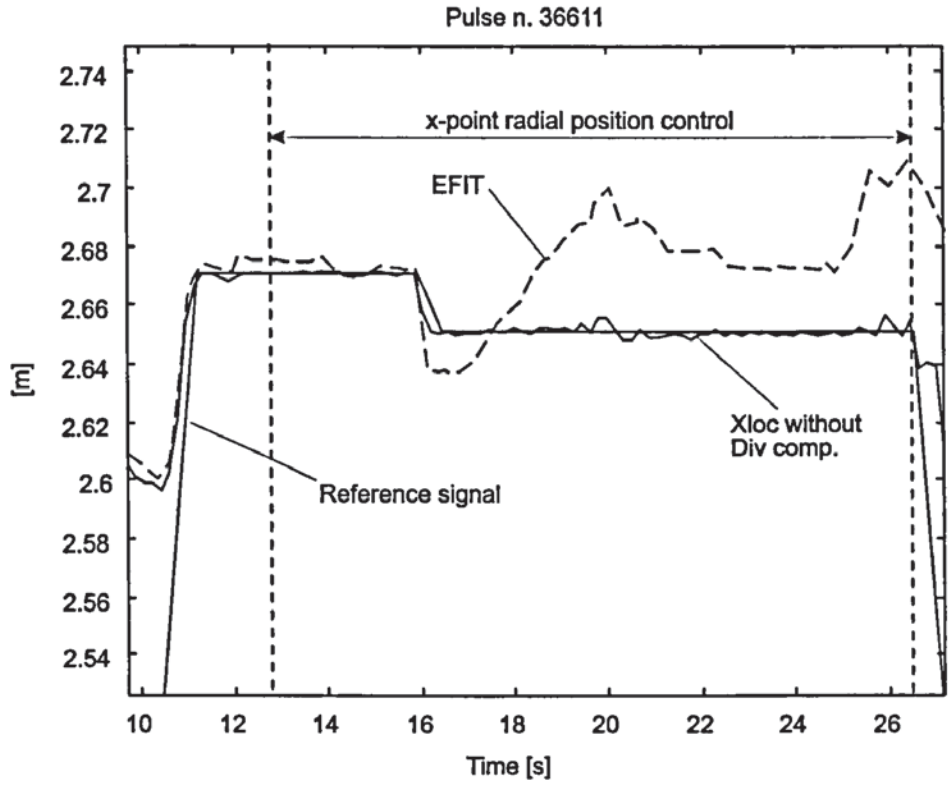

Figure 2.5: This pulse was an attempt to control the radial position of the x-point using the x point position determined by XLOC as feedback signal. The feedback is working but without divertor compensation XLOC does not find the x-point region position correctly, especially when it is moved from the centre of the divertor situated at 2.66 m.

The problem has been solved by introducing an estimate of the divertor coils flux and subtracting it from the total flux so that the flux expansion measures only the flux in vacuum.

The total flux is then made by two components: the vacuum flux and the divertor flux, where the six order expansion defined before represents just the vacuum flux:

$$\Psi_{tot} = \Psi + \Psi_{div} \tag{2.40}$$

The divertor flux is determined by approximating the four divertor coils with filamentary currents where each coil turn is represented with four filaments ($N_{j\,fil} = 4$):

$$\Psi_{div}(R_P, Z_P) = \sum_{j=1}^{4} \Psi_{Dj}(R_P, Z_P) = \sum_{j=1}^{4} \sum_{i=1}^{N_{j\,fil}} \Psi^{i}_{fil}(R_P, Z_P) \tag{2.41}$$

The flux of each filament is determined solving the integral below [52]:

$$\Psi^{i}_{fil}(R_P, Z_P) = R_P \mu I_{fil} \int_0^{\pi} \frac{R_{fil}\cos(\phi)d\phi}{\left(R_{fil}^2 + R_P^2 + Z_P^2 - 2R_{fil}R_P\cos(\phi)\right)^{1/2}} = \frac{I_{Dj}}{N_{j\,fil}} d_i(R_P, Z_P) \tag{2.42}$$

(R_P,Z_P) are the coordinates of the point where the flux is evaluated. (R_{fil},Z_{fil}) is the position of the filament and I_{fil} is equivalent to the current flowing through a divertor coil turn (I_{Dj}) divided by the number of filaments that represent it. The elliptic integral of equation (2.42) can be calculated using its fourth order expansion [1]; The divertor flux then becomes:

$$\Psi_{div}(R_P, Z_P) = \sum_{j=1}^{4} \Psi_{Dj}(R_P, Z_P) = \sum_{j=1}^{4} \frac{I_{Dj}}{N_{j\,fil}} \sum_{i=1}^{N_{j\,fil} \cdot N_{j\,turn}} d_i(R_P, Z_P) \tag{2.43}$$

where $d_i(R_P, Z_P)$ represents the fourth order expansion of the integral of equation (2.42).

2.8.8 *c* coefficients calculation with flux divertor compensation

To calculate the *c* coefficients of the flux ψ all the measurements must be compensated by the divertor flux. Hence, the divertor flux must be calculated for all magnetic sensor positions. This can be easily done since the position of the sensors is well known and therefore the integral of equation (2.42) can be determined off-line. Using the matrix notation of equation (2.31) the *c* coefficients can be expressed as:

$$\mathbf{c} = \mathbf{X}(\mathbf{m} - \mathbf{D}\mathbf{I}_D) = \mathbf{X}\mathbf{m} - \mathbf{X}\mathbf{D}\mathbf{I}_D = \begin{bmatrix} \mathbf{X} & -\mathbf{X}\mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{m} \\ \mathbf{I}_D \end{bmatrix} \tag{2.44}$$

$\mathbf{I_D}$ is the column vector of the four divertor coil currents. **D** is a matrix of four columns and each of the elements corresponds to the sum of the functions $d_i(R_P, Z_P)$ calculated for all the filaments representing a divertor. Note that all the calculation of **D** can be done off-line.

2.8.9 X-point search

The x-point search on the contrary must be performed on the total flux ψ_{tot}. As the x-point search is done iteratively, therefore the divertor flux calculation needs to be performed for every point of the search process. Unfortunately, to calculate the divertor flux, the integral of equation (2.42) must be solved for 304 times (as many as the filaments used to represent the divertor coils)for each iteration. This calculation can not be done in real-time, because of its complexity. To reduce the time spent on estimating the divertor flux the same approach used to calculate the vacuum flux has been used. The flux generated by each divertor is represented by a Taylor expansion:

$$\psi_{Dj}(\rho,z)=\left(\sum_{i,k=0}^{i+k<6} I_{Dj}a_{i,k}^{Dj}\rho^i z^k\right) \qquad j=1,..,4 \tag{2.45}$$

In this case, to determine the a^{Dj} coefficients, we fit the flux expansion to 30 points selected on the divertor region where the $d()$ functions (eq.(2.43)) are calculated for each filament.

This equation has the same structure of the equation for the vacuum flux. The x-point search is performed using the same method as before, but with the a coefficients expressed as a sum of five terms:

$$\tilde{a}_{i,j}=a_{i,j}+\sum_{k=1}^{4} I_D a_{i,j}^{Dk} \tag{2.46}$$

The calculation of the a^{Dj} is performed off line, therefore we can write the new $\tilde{a}$ coefficients as a matrix product where the divertor flux part is added to equation (2.13):

$$\tilde{\mathbf{a}}=[\mathbf{M} \quad \mathbf{M}_D]\begin{bmatrix}\mathbf{c}\\ \mathbf{I}_D\end{bmatrix} \tag{2.47}$$

The results obtained with the divertor flux compensation are quite good showing a better agreement with EFIT, as it can be seen in the figure below where XLOC was reprocessed for pulse 36611.

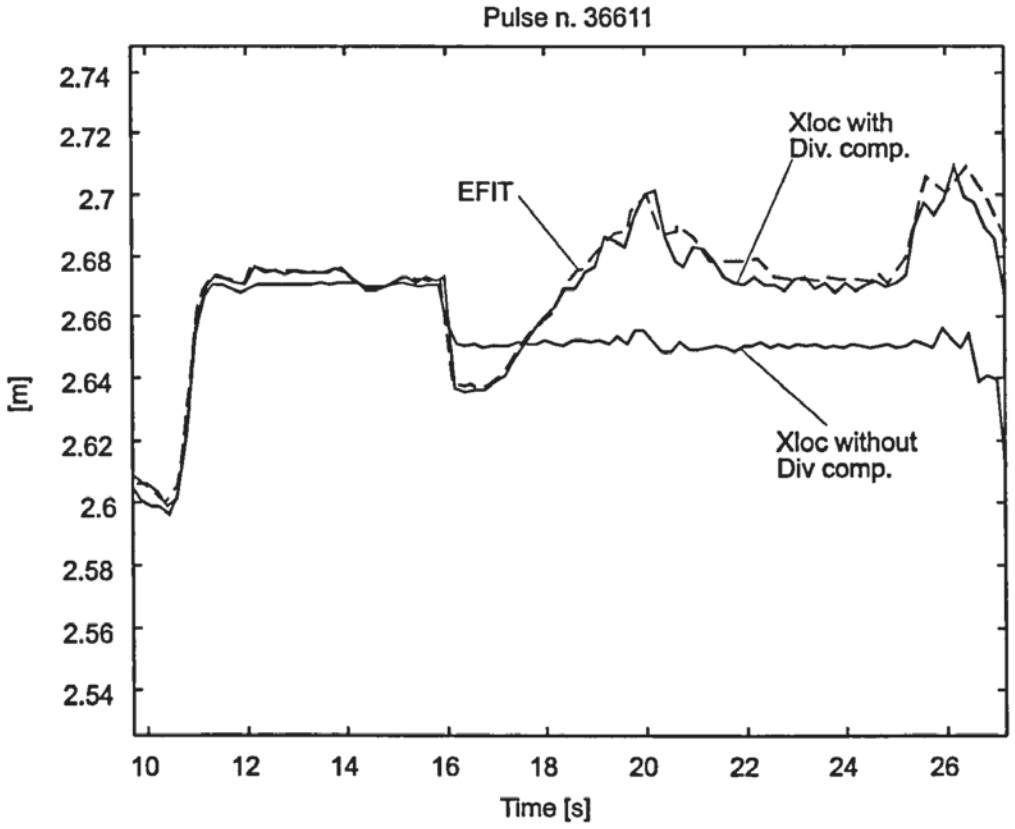

Figure 2.6: Using the divertor flux compensation XLOC reaches the desired accuracy of less than 1.0 cm.

After having reached a sufficiently good precision on the determination of the x-point and as a consequence on the boundary flux value. We can now estimate the position of the boundary in those particular areas of the vacuum vessel which are interesting for control purposes.

2.9 Plasma wall distance measurement

XLOC determines the plasma/wall distance by finding the intersection of the plasma boundary with 21 predefined segments called gaps. To speed up the search process, the segment is divided into 200 points and for each of these points the values $\tilde{g}_i(\rho,z)$ of equation (2.12) and the $\sum_{i=1}^{N_{fil}} d_i(R_P,Z_P)$ of (2.43) are calculated off-line. Then, using the bisection method, the plasma boundary is searched comparing the boundary flux with the flux estimated in these points where it can be easily determined by a product of vectors of 17 elements.. On the figure below the 21 gaps along which the plasma boundary is determined are shown. They have been chosen so that the whole vessel can be equally covered.

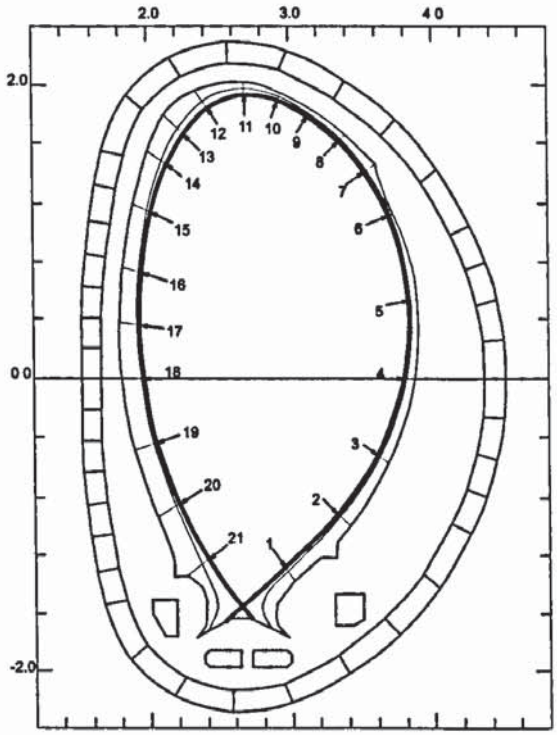

Figure 2.7: Gap positions. For each gap the distance plasma/wall is measured using the bisection method to find the intersection of the plasma boundary flux line with the segment.

Only gap 2, 4, 11, 14, 18 the x-point and the strike points positions can be used for control purposes. The control system transforms the distance error between the gap reference and the

measurement into a voltage request to the PF circuit amplifier. The current driven in the coils will move the plasma accordingly. Up to five of these measurements can be controlled simultaneously, and during the pulse the choice, of which measurements are controlled, can be changed using the appropriate set for each part of the pulse.

In addition to these measurements, the XLOC code has been further developed to provide the magnetic field at the boundary. This information proved to be very useful for the estimation of the *internal* plasma parameters needed for plasma disruption detection purposes, as discussed in chapter three.

The magnetic field radial and vertical components are calculated using a process similar to the flux calculation. A vector of seventeen elements is stored for each of the 200 points of the gaps. This vector contains the derivatives of the $\tilde{g}_i(\rho,z)$ of equation (2.12) and the $\sum_{i=1}^{N_{fil}} d_i(R_P,Z_P)$ of (2.43). When the point where the boundary is crossing the segment is found, the values of the magnetic field are simply determined by a vector product between the c coefficients and the stored values of the derivatives of $\tilde{g}_i(\rho,z)$ and $\sum_{i=1}^{N_{fil}} d_i(R_P,Z_P)$

2.10 Discussion

The final implementation of the XLOC achieved the goal to design an algorithm that could run in 2 ms. To allow a real time implementation XLOC has been split into two parts.

The off-line part performs the computationally demanding calculation of the **X** matrix of equation (2.32), the **D** matrix of equation (2.44), the $\mathbf{M_D}$ matrix of equation(2.47), and the tables to determine the gaps and their magnetic field. This reduces the time needed on line to find the plasma boundary by reducing the real time calculation to a set of matrix multiplication for the determination of the *c* coefficients, and the vessel fluxes. Also the x-point and the plasma/wall distances search has been reduced in complexity by introducing the matrix derivative operator and the discretisation of the segments used to determine the plasma/wall distances.

At present the system is implemented using two DSP processors running in parallel and is always used to control the plasma shape.

A major problem of XLOC is the lack of diagnostic algorithms to detect malfunctioning of the magnetic sensors. This could lead to plasma disruptions due to control errors, since the use of a faulty plasma signal can introduce large errors on the determination of the plasma boundary parameters used for control.

Fortunately, the insertion or exclusion of magnetic measurements in XLOC is very simple as long as there are enough magnetic sensors available to determine the *c* coefficients. This needs just a re-running of the off-line code using a new input file. The new matrices are then loaded on the on line system and the plasma operations can be immediately restarted. This is the reason why XLOC has been chosen to be the pre-processing algorithm for the neural network discussed on the next chapter. The possibility of having input signals that are not directly dependent on the magnetic sensors, gives us the guarantee that the malfunctioning of a sensor does not force the long process of retraining the network or even redesigning it.

3. ℓ_i-q_ψ diagram

3.1 Introduction

The aim of this work is to study the possibility to avoid disruptions or at least to minimise their effect on the machine structure. From experimental results it has been discovered that it is possible to delimit a safe region on the ℓ_i-q_ψ diagram. If the plasma exits from it the probability of having an increase in the MHD activities is very high, and this could lead to a plasma disruption.

ℓ_i and q_ψ are two internal plasma parameters that can be determined only using full plasma equilibrium reconstruction codes, which usually are very complex and computationally demanding. The implementation in real time of these codes is not feasible. Therefore, the strategy chosen is to use neural network methods to determine these two values from data that can be calculated in real time.

The same network has also been designed to determine other parameters that could be useful for control purposes or for predicting disruptions. The plasma parameters that have been identified to be useful in real-time, are:

- the plasma inductance ℓ_i,
- the safety factor calculated at:
  - 95% of the plasma radius, q_{95}
  - at the plasma surface for plasma limiter configuration, q_{wall},
  - at the plasma axis, q_{axis}
- the plasma poloidal beta β_p.

3.2 Safety factor q [60]

The safety factor, q, is so called because of the role it plays in determining plasma stability. In general terms, higher values of q lead to greater stability.

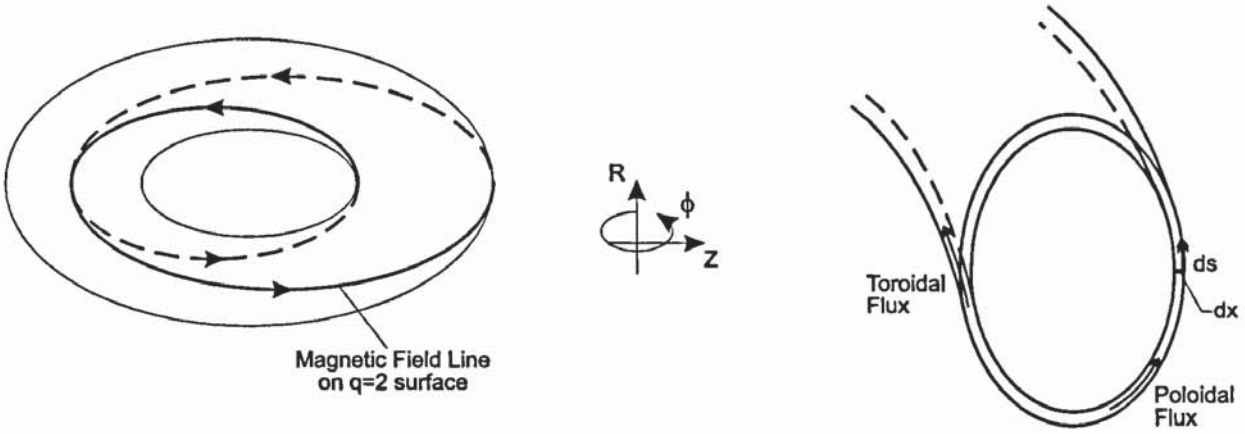

Figure 3.1: Representation of a magnetic field line with safety factor q=2 on the left. On the right the flux surface cross-section.

In an axial-symmetric equilibrium each magnetic field line has a value of q. The field line follows a helical path as it goes round the torus on its associated magnetic surface. If at some toroidal angle, ϕ, the field line has a certain position in the poloidal plane, it will return to that position in the poloidal plane after change of toroidal angle $\Delta\phi$. The q value of this field line is defined by

$$q = \frac{\Delta\phi}{2\pi} \tag{3.1}$$

Thus, if a magnetic field line returns to its starting point after exactly one rotation round the torus, q is equal to 1. If it moves more slowly in the poloidal direction it has a higher value of q. Rational values of q play an important role in stability. If $q=m/n$, where m and n are integers, the field line joins up on itself after m toroidal and n poloidal rotations round the torus.

In order to calculate the value of q it is necessary to use the equation of the field line

$$\frac{R\,d\phi}{ds} = \frac{B_\phi}{B_p} \tag{3.2}$$

where ds is the distance moved in the poloidal direction while moving through a toroidal angle $d\phi$, and B_p and B_ϕ are the poloidal and the toroidal magnetic fields.

Thus from equation (3.1)

$$q_\psi = \frac{1}{2\pi}\int \frac{1}{R}\frac{B_\phi}{B_p} ds \tag{3.3}$$

Where the integral has been done along a single poloidal circuit around a magnetic surface. The radial profile of q usually has its minimum value at, or close to, the magnetic axis and increases outwards.

When plasmas are bounded by a separatrix, as they are in x-point configuration, the q profile is fundamentally modified. The reason is that the value of q for surfaces close to the separatrix is dominated by the contribution close to the x-point. The null of the poloidal magnetic field forces q to infinity. The underlying cause is that the smallness of B_p around the x-point delays the poloidal motion of the field lines, and field lines on surfaces close to the separatrix spend almost all their trajectory close to the x-point.

In this work we will consider three different safety factor values:

- q_{wall} as the value measured at the plasma surface ($r=a$ where a is the plasma minor radius) when the plasma is in limiter configuration.
- q_{95}, defined as the safety factor calculated at the flux surface with flux value of $\psi_{axis} - 0.95(\psi_{axis} - \psi_a)$, where ψ_a is the flux at the plasma boundary. This is universally chosen as the plasma surface safety factor for x-point plasmas.
- q_{axis} defined as the safety factor determined at the plasma axis.

3.3 Plasma internal inductance ℓ_i

ℓ_i is the normalised plasma internal inductance. It is used to characterise the plasma current profile, values of ℓ_i higher than 0.5 represents current density profile peaked in the centre.

ℓ_i is defined as the plasma internal inductance normalised to the inductance of the plasma with a flat current profile [60]. Thus, it can be expressed by the ratio of the plasma magnetic energy with the energy of the same plasma if it would have had a flat current profile:

$$\ell_i = \frac{\frac{1}{2\mu}\int_v \mathbf{B}_\theta^2 dv}{\frac{1}{2\mu}\int_v \mathbf{B}_{\theta a}^2 dv} = \frac{2\int_0^a \mathbf{B}_\theta^2 r\,dr}{a^2 \mathbf{B}_{\theta a}^2} \tag{3.4}$$

where $\mathbf{B}_\theta$ is the poloidal field, $\mathbf{B}_{\theta a}$ is the poloidal field at the surface. The integral can be solved by considering a circular plasma cross section with radius a, with the poloidal magnetic field axial-symmetric with respect to the machine axis.

In normal conditions at JET ℓ_i can vary between 0.8 and 1.5. As said before there is a close relation with the shape of the density current profile. This characteristic makes of ℓ_i a very useful parameter to describe the plasma configuration together with β_p .

3.4 Poloidal beta β_p

The efficiency of confinement of plasma pressure by the magnetic field is represented in a simple way by the ratio:

$$\beta_p = \frac{\int p\,dS / \int dS}{B_a^2 / 2\mu_0} \tag{3.5}$$

where the integrals are surface integrals over the poloidal cross section, B_a is defined as the ratio of the plasma current with the length of the poloidal perimeter $B_a = \mu_0 I / l$.

Being a measurement closely related to the plasma pressure, β_p could add useful information to classify plasma configurations providing information about the plasma pressure.

3.5 ℓ_i-q diagram

During the first period of JET operation, the analysis of the behaviour of the plasma during a pulse led to the design of the ℓ_i-q diagram. A safe operational space for the two parameters was designed accordingly to the experimental results. When, during the pulse, the ℓ_i-q

breaches the limit of the safe area, the probability to have a disruption increases due to MHD instability.

Figure (3.2) shows the safe and unsafe areas and the trace of ℓ_i and q during a pulse.

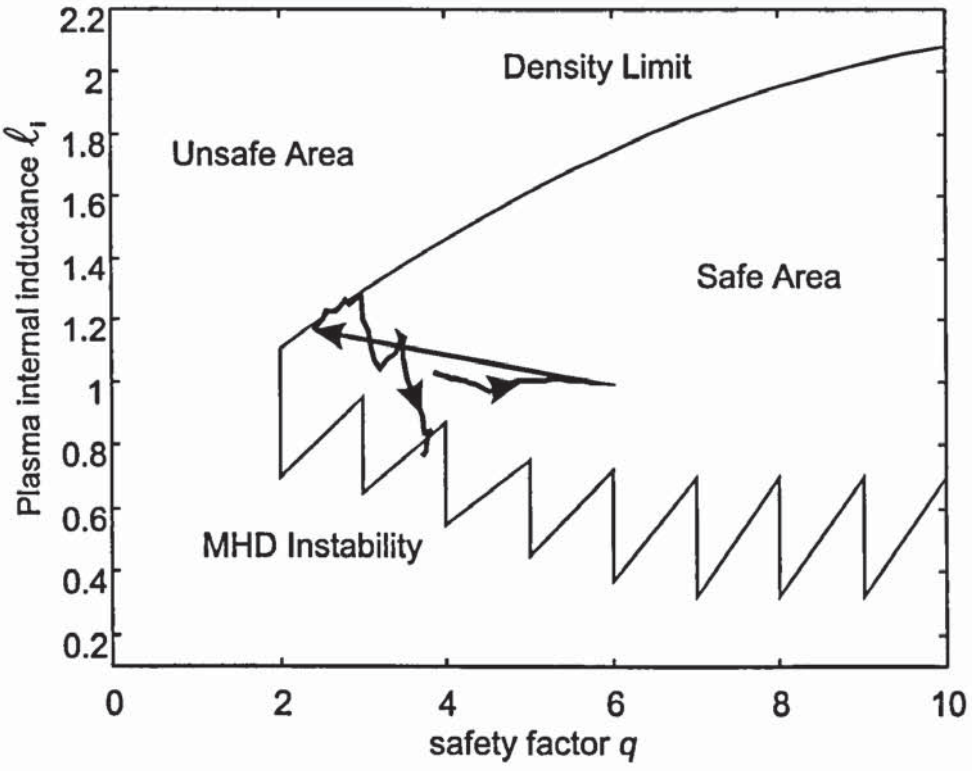

Figure 3.2: ℓ_i-q diagram.

The upper limit of the ℓ_i-q diagram is mainly related to density limit disruptions. This limit is configuration dependent and it is not considered in this thesis. The MHD instability limit on the lower part of the diagram has also a theoretical derivation from kink instability theory [60].

The kink mode is so named because it leads to the kinking of the magnetic surfaces and the plasma boundary. This perturbation of the magnetic surfaces produces an equilibrium instability of the plasma pressure with the magnetic field. The energy variation caused by the kink mode can be determined and it has been demonstrated that it mainly depends on the safety factor q and on the plasma current profile [60]. It is possible to draw a stability region on the space of q and the current profile for this energy variation. Outside this region the energy variation will lead to an unstable plasma configuration.

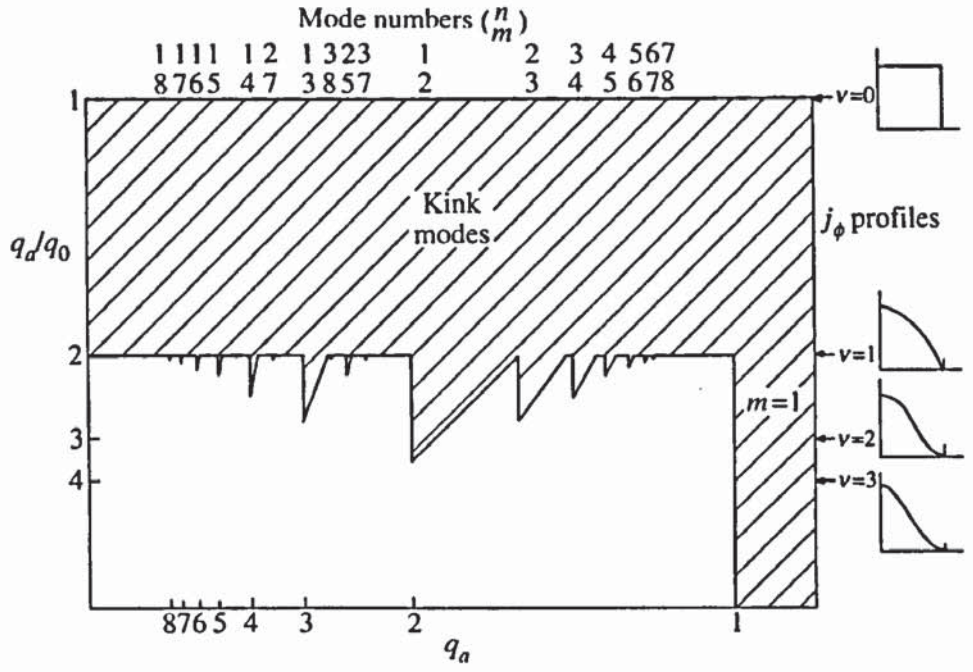

Figure 3.3: Stability diagram for kink modes for the current distributions $j=j_0(1-(r/a)^2)^\nu$. The vertical axis measures the peaking of the current and the horizontal coordinate is proportional to the safety factor at the boundary.

The assumptions made to draw this safe region are such that it is difficult to apply to the real experiment. The plasma current profile is supposed to belong to a given class of curves that does not represent the real plasma current profile. Moreover the difficulties of determining the safety factor q_0 at the plasma axis with good accuracy, suggested to use the plasma internal inductance ℓ_i to characterise the plasma current density profile. The relation between the two parameters can be given by the empirical fit:

$$\ell_i = \log(1.65 + 0.89\nu) \qquad \nu = \frac{q_a}{q_0} - 1 \tag{3.6}$$

Nevertheless the ℓ_i-q diagram method proved not to be fully reliable in the divertor campaign. An investigation of the performance of the ℓ_i-q diagram has been carried out for a set of 1290 pulses (from pulse number 30000 to 31900) with plasma current higher than 1 MA. Table 3.1 shows the results using q_{wall} for limiter plasma configurations and q_{95} when the plasma was in x-point configuration.

	Successful pulses	Disrupted pulses
X-point plasma outside lower boundary	63 (6.5%)	26 (8%)
Limiter plasma outside lower boundary	24 (2.5%)	22 (7%)
X-point plasma outside upper boundary	253 (26%)	44 (13%)
Limiter plasma outside upper boundary	18 (2%)	24 (7.5%)
Inside safe area	612 (63%)	204 (64.5%)
Total number of pulses	970	320

Table 3.1: Statistical analysis of pulses from 30000 to 31900.

The change of plasma configuration from limiter to x-point showed that the limit of the safe area had to be redesigned. Studies on the design of these new limits are currently undertaken. The ℓ_i-q diagram can still be used during the early phase of the plasma ramp-up. In this period the limits defined during the pre-divertor campaign can be used because the plasma is normally kept in limiter configuration. When the plasma configuration becomes x-point the q_{wall} can not be used. The statistical analysis proved that using q_{95} without changing the limits of the diagram increased the number of wrong predictions, making the ℓ_i-q diagram method for predicting disruptions unreliable. This does not mean that it is not possible to set new limits for the ℓ_i-q diagram, but just that more extensive studies are needed. The above statistic has been carried out automatically analysing the ℓ_i-q diagram using algorithms that are too simple to design a new safe area for the ℓ_i-q diagram. Finding the proper limit would involve several checks that are very difficult to implement automatically. The necessary activities are described below.

1. Determine if there was a disruption and find the disruption time using the plasma current and the flux loop voltage signals.
2. If a disruption occurred verify that MHD instabilities occurred before the disruption. If MHD activities occurred well before the disruption (more than 1 second), a decision has to be taken if they can be related to the disruption or if they disappeared spontaneously. It is also important to discriminate the causes that generated these MHD instabilities to be sure not to be misled by instability generated for example by plasma impurity influx.
3. In the ℓ_i-q diagram, flag as dangerous the points regarding the periods where MHD activities where present. The plasma configuration has to be taken into account, using q

at the surface or q_{95} depending on whether in limiter or in x-point configuration. The number of EFIT plasma equilibria (100) stored per pulse might be insufficient to perform this analysis, therefore there could be the need to rerun EFIT.

4. If the plasma pulse was successful, its MHD activity need to be checked anyway. A decision has to be made if these events where threatening the plasma stability sufficiently to flag in the ℓ_i-q diagram these periods of the pulse as dangerous. It may be necessary to add the time variable in the ℓ_i-q diagram to determine if the time that the plasma configuration stays in a particular area of the ℓ_i-q diagram increases the probability of having a disruption.

There is not yet an established way of automatically doing these checks and therefore the process is very time consuming, especially if EFIT needs to be re-run to add new data or to obtain a more precise plasma reconstruction. For these reasons it was preferred firstly to prove that the determination of the ℓ_i and q was possible in real time and afterwards to focus on the redefinition of the ℓ_i-q diagram using the most recent data.

3.6 The ℓ_i - q network

At JET, the EFIT code reconstructs the plasma shape, position, and current density profile by solving repeatedly the non linear Grad Shafranov equation. The magnetic measurements are used as equilibrium constraints to search the best-fit current density profile [28]. Starting from a predefined plasma density current profile, EFIT adjusts the plasma parameters to match the magnetic measurements using the Picard iteration method [3]. The process is computationally demanding taking up to 3 seconds on the IBM mainframe for each time slice. Thus, the EFIT code is executed off-line after the end of the pulse and generates plasma equilibrium results only for 100 time slices. Even if it is envisaged that the time could be reduced to 50 ms per time slice, at present we can not use it for real time purposes.

The other code available at JET capable of estimating internal plasma parameters as ℓ_i and q is called FAST. FAST does not perform a full plasma equilibrium reconstruction, but starting from the magnetic field at the plasma boundary, it calculates ℓ_i and q by determining the so called Shafranov integrals [50]. At JET these integrals are calculated along a contour on

the plasma boundary using XLOC to determine the position and the magnetic field on the plasma surface.

The results compared to the full equilibrium code EFIT are within 10% for plasmas that were used to calibrate this calculation. For new plasma configurations the error increases up to 20 % making this kind of measurement not sufficiently reliable.

The figure below shows the ℓ_i parameter estimation done by EFIT and FAST. In this case FAST presents an error up to 20% and as it can be seen it is not constant during the pulse.

Gaining computational speed caused an accuracy reduction that does not allow the use of this algorithm for control or protection purposes.

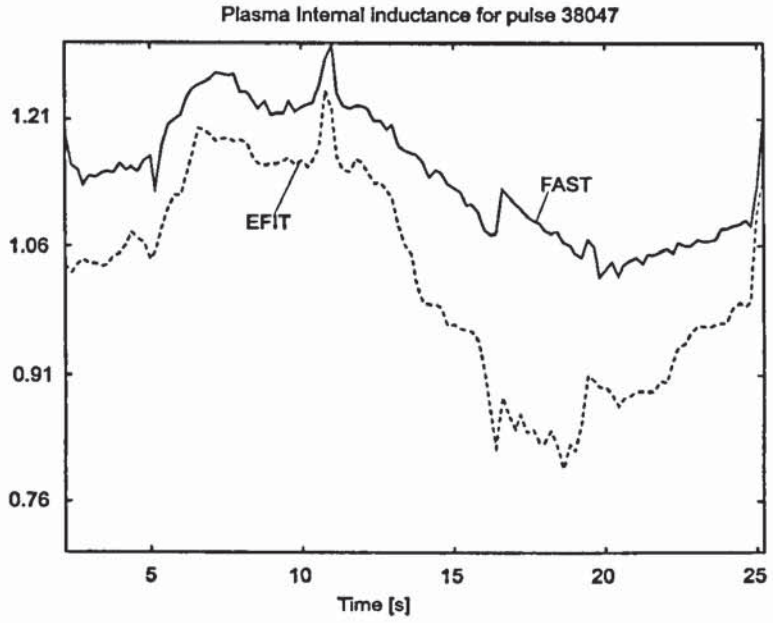

Figure 3.4: EFIT versus FAST. The plasma internal inductance is compared for pulse 38047 It can be noticed how between 15 and 20 s the difference between the two signals is higher than 20%.

Neural network approach could be the solution to the problem if it proves to be sufficiently reliable and accurate. As a neural network problem it comes down to finding a mapping from the input data, available in real time, to the plasma parameters.

3.7 Generation of datasets

The information to determine ℓ_i and q resides in the plasma magnetic configuration. Therefore the magnetic sensor measurements can be used as input to the neural network. Unfortunately these sensors are not always reliable. There were occasions when some of them worked intermittently or even stopped to produce data, due to hardware faults. XLOC has been used as a pre-processing stage to provide a more robust input to the neural network, and to avoid the retraining of the network caused by a faulty input signal.

XLOC was chosen because it provides sufficiently accurate information of the magnetic field configuration close to the plasma boundary. Moreover, it is designed so that the replacement of a faulty signal from its input does not affect its results and does not cause long delays to the machine experimental operation. The use of XLOC as a pre-processing unit also means that we can select the XLOC output signals that provide to the neural network the maximum quantity of information using the minimum number of input signals.

In practice, the XLOC code has been imported on the PC environment where it has been used to generate the training datasets for the ℓ_i-q network.

For each pulse, of the 590 selected, Matlab data files were created to store all the relevant data:

1. the magnetic signals coming from the data acquisition system called KC1D
2. The poloidal and toroidal coil currents coming from PPCC
3. Data coming from the real time signal server as the additional heating input powers and the plasma density.
4. Post-pulse processed data from EFIT and FAST.

From this database two datasets were generated to train and validate the network. A further dataset to test the final network has been generated using a set of the latest pulses carried out at JET, to check that the network was still performing reasonably.

Seventy different pulses have been selected for each dataset. Each pulse has been used for only one datasets avoiding the possibility of selecting the same pattern twice. After a thorough check to eliminate corrupted data or duplicates, thirty patterns were selected for each pulse, reaching the total number of 2100 patterns for the training and the validation datasets. The dataset size has been limited not by the scarcity of data, but by the computational capability of

the personal computer used to design the network (90-100 minutes are necessary to perform 3000 training iterations).

Several networks were investigated, varying the input variables in number and type. Starting from a small input set of 9 inputs we reached the maximum number of 30 different input variables. The final solution has been chosen to give the best results keeping the size of the network as small as possible. It consists of 16 signals:

- the plasma current
- the toroidal field coil current
- the position and magnetic field value of the plasma boundary determined along 7 different gaps.

The plasma boundary positions are calculated using the XLOC code. They have been chosen to provide information about the plasma geometry, and magnetic configuration. The magnetic field values should provide the necessary information to reconstruct the current profile necessary to determine ℓ_i.

To determine q it is also necessary to know the toroidal magnetic field. In this case, we use the TF coil current.

3.8 Data normalisation

The selected input variables (x_i) present values that differ by several orders of magnitude. These differences may not reflect the relative importance of the input variables in determining the required output. Therefore, the weights of the network that we are going to determine can present huge differences increasing the time spent on the training process.

It is preferable to normalise the data re-scaling the variable so that the network will benefit from a more homogeneous input set.

Different methods can be used to re-scale the data, here each variable is treated independently [6]. The mean $\overline{x}_i$ and the variance σ_i^2 with respect of the training set was calculated using the formulas:

$$\overline{x}_i = \frac{1}{N}\sum_{n=1}^{N} x_i^n \qquad (3.7)$$

$$\sigma_i^2 = \frac{1}{N-1}\sum_{n=1}^{N}(x_i^n - \bar{x}_i^n)^2 \tag{3.8}$$

The re-scaled variables are then given by

$$\tilde{x}_i^n = \frac{x_i^n - \bar{x}_i^n}{\sigma_i} \tag{3.9}$$

The aim is to guarantee that all the data are on the order of one. In this case we can expect that the weights will also be of the same order making their initialisation easier.

It should be noted that this linear transformation of the input values could be easily combined with the first layer network, avoiding an increase in the computational complexity of the network that is a serious constraint for the implementation in real time of these algorithms.

The same re-scaling process has also been applied to the output of the network.

3.9 Training of the network

The network used is the multilayer perceptron with the identity function as the activation function of the output units. The determination of the network topology has been done using the same training dataset for 62 different networks differing in the number of hidden layers.

As regards the training itself, several algorithms can be found in literature, C.Bishop in its book [6] lists the gradient descent, the conjugate gradient, Netwon's method, and the Levemberg-Marquardt algorithm. In this work the algorithm used for the training is the gradient descent because of its simplicity of implementation and because it is less computational demanding of the other ones.

The weight vectors are updated iteratively using the formula:

$$\Delta \mathbf{w}^{(\tau)} = -\eta \nabla E\big|_{\mathbf{w}^{(\tau)}} \tag{3.10}$$

where $\nabla E\big|_{\mathbf{w}^{(\tau)}}$ is the gradient of the error defined in paragraph 1.9 as the sum over the training patterns of E^n (see equation (1.15)) and η is the learning rate. At this stage no regularisation methods (see chapter 4 and 5) have been used, because of the increase the

number of network to be trained in order to optimise the parameters that control them. In chapter 5 and 7 where the networks size is relatively small the weight decay method of regularisation has been introduced. It is anyway recommended to use these methods to reduce the problem of over-fitting to the noise of training data.

The learning rate η plays an important role for the convergence of the algorithm to a proper solution. If it is chosen too large the algorithm may overshoot leading to an increase of the error and possibly to divergent oscillations. If the learning rate is on the contrary too small the algorithm may need a large number of iterations to come to converge to a proper solution. For these reasons, the empirical way of adapting the learning rate, introduced by T.P.Vogl [59], has been used. If the new error exceeds the old error by more than a predefined ratio of 1.05, the new weights are discarded. In addition, the learning rate is decreased by multiplying it by 0.7. Otherwise the weights are kept. If the new error is less than the old error, the learning rate is increased by 5%. This procedure tries to optimise the learning rate to increase the speed of the training process and at the same time to avoid that a high learning rate prevents the location of the minimum error. The method has been tested for a network with 10 hidden units and proved that the adaptive learning rate algorithm was converging considerably faster than the simple gradient descent and therefore it has been used for the training process of all the other networks.

The gradient $\nabla E|_{w^{(\tau)}}$ has been calculated using the back-propagation method (paragraph 1.9), and cross checked using the numerical differentiation method. This method determines the error gradient as a central difference of the error calculated after having done a small variation of the considered weight.

$$\frac{\partial E^n}{\partial w_{ij}} = \frac{E^n(w_{ij}+\varepsilon) - E^n(w_{ij}-\varepsilon)}{2\varepsilon} + o(\varepsilon^2) \tag{3.11}$$

This calculation was performed regularly after a given number (200) of training iteration and its results were compared with the back propagation method to guarantee that the back-propagation calculation was performed correctly.

The training process is usually stopped after a predefined error limit is reached. In this case, since the size of the neural network (16 input variables and 5 outputs) and the training set (2100 patters) was quite big it was decided to stop it after 3000 iteration. The number of

iterations was chosen after preliminary tests training several networks. The behaviour of the error curve showed that after 3000 iterations there was no evident improvement on the network performance as it can be seen in Figure 3.5.

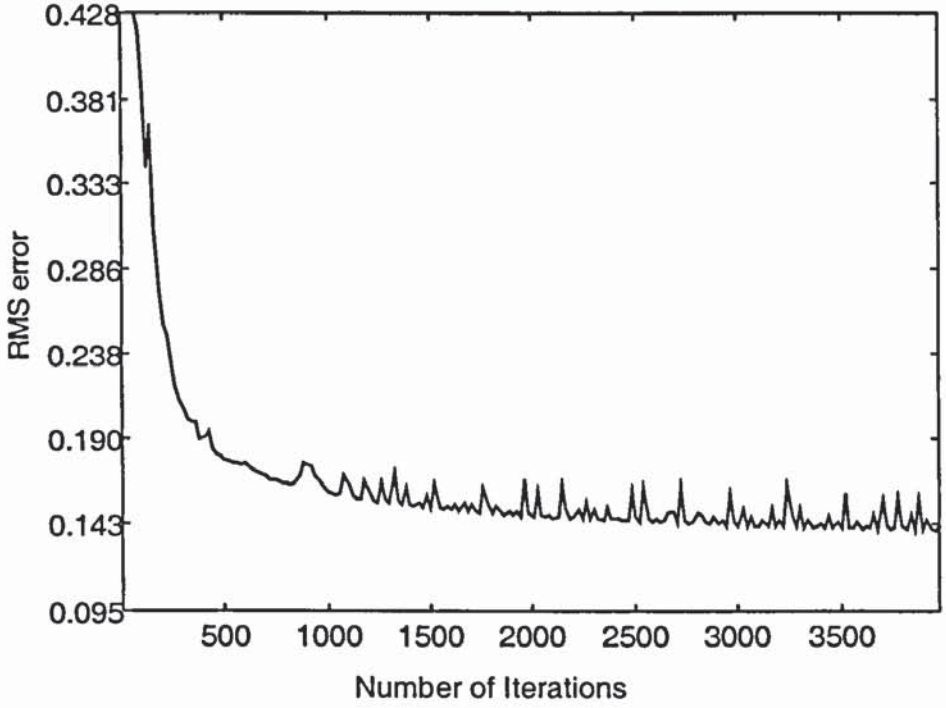

Figure 3.5: The error evolution during the training process. It can be seen how after 3000 iterations the error is not decreasing significantly.

Further studies proved that this number of iteration is close to the average number of iteration that would have been required if the early stopping method was used to control the complexity of the network. The early stopping method is presented in chapter 5 where it is used to train the disruption prediction network.

To decide the best performing network, the RMS value of the error for the trained networks was plotted as a function of the hidden units. The error is defined as:

$$E^{RMS} = \left\{ \frac{1}{cN} \sum_{n=1}^{N} \sum_{k=1}^{c} \left(y_k^n - t_k^n \right)^2 \right\}^{1/2} \tag{3.12}$$

where N is the number of patterns and c is the number of output units, in this case 2100 and 5.

To select the best performing network, the "hold out" method has been used [6]. Figure 3.6 shows the error of the training sets as black circles, and the validation set as white squares. For the same network size the training was performed twice starting from two different weight

sets. Networks with the same size have not always the same performance. This is a consequence of the fact that network training corresponds to a non linear optimisation problem that can suffer from local minima. For the same reason the ideal monotonic trend of the error that should decrease with the increase of the network size is not so evident.

As regard the validation set error, we can see that networks with more than 20 hidden units give a higher value of the RMS error. The increasing complexity of the network allows a better approximation of the training patterns but at the same time its outputs can lose generality. Thus, the best performing network is the one that presents the minimum value for the validation set using fewer hidden units. In this, one of the networks with 18 hidden units presents the lowest error value for the validation set and it is also less complex than the other networks that present a similar error.

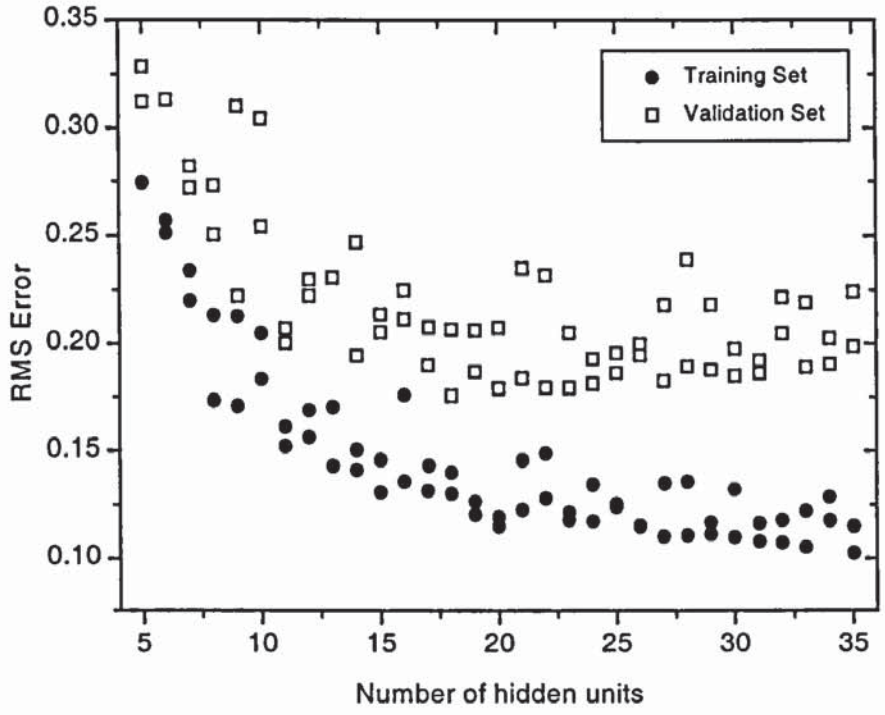

Figure 3.6: The diagram shows the trend of the RMS Error as a function of the network size. The trend of the error for the training set shows how increasing the number parameters the error diminishes. At the same time the increasing error of the validation set demonstrate that the network is loosing generality, overfitting the training set. Thus the best performing network is the one that presents the minimum value for the validation set. It can also be noticed how the the higher is the number of hidden units the higher is the difference between the error reached by different networks with the same number of units. This is most probably caused by the fact that the training reaches different local minima.

After the selection of the network, several tests were carried out to verify the network performance. One of these tests was the reprocessing of the available pulses to compare the results with EFIT and FAST. On the figure below the neural network output variables are plotted. The pulse was not part of the training or validation dataset. The overall performance of the network seemed sufficiently accurate to be implemented and used in real-time. Of a certain importance is the comparison with FAST, where we can see how the measurements seem more accurate, especially the plasma internal inductance ℓ_i.

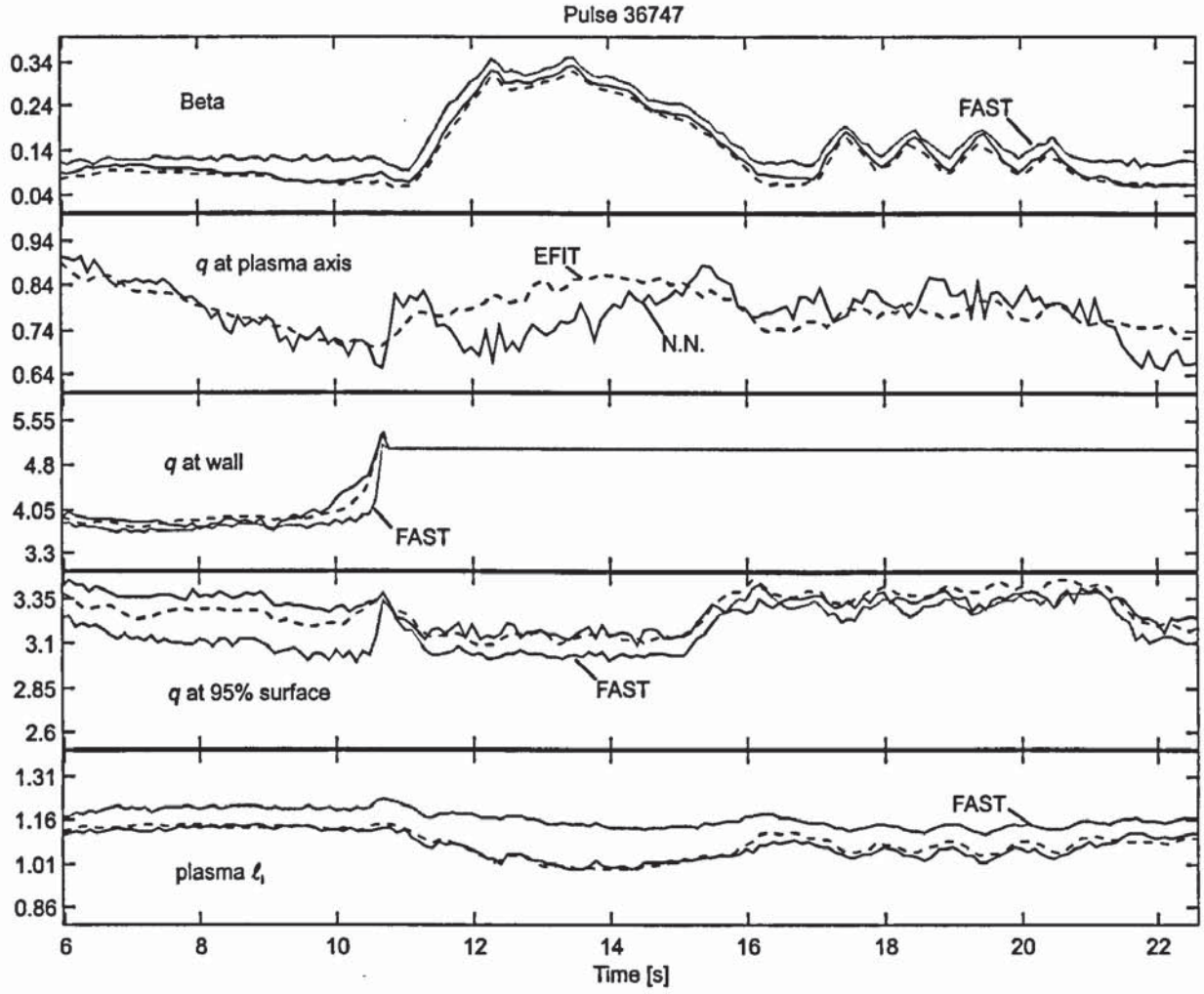

Figure 3.7: Comparison between the network output, EFIT results (dotted line), and FAST (grey line). The q_{wl} value is set to a fixed value when the plasma is in x-point configuration, because it has no meaning. The safety factor at the axis is estimated only by EFIT.

To confirm these results the test set was used to calculate the average relative error of the network compared with the EFIT results. This error has been chosen to try to give a general feeling of the performance of the network providing the error in percentage of the measurement. The formula used to calculate the error is:

$$E_{rel}^{k} = \frac{1}{N}\sum_{n=1}^{N}\left|\frac{y_k^n - t_k^n}{t_k^n}\right| \tag{3.13}$$

Plasma Parameter	Neural Network relative error
Poloidal beta β_p	0.127
Safety factor q at plasma axis	0.072
Safety factor q at wall	0.053
Safety factor q at 95% plasma minor radius	0.034
Plasma inductance ℓ_i	0.042

Table 3.2: Relative error calculated with equation (3.13).

The parameters where the network performs less accurately are β_p and q at the axis. The main problem with β_p is that its value for ohmic plasmas is very small -around 0.1- compared to the values of heated plasmas (1.0÷2.0). The accuracy of β_p for ohmic plasmas is therefore affected by the noise of the measurements. If we do not consider EFIT β_p values lower than 0.1, we obtain a relative error of 0.06. This suggests that the error increases where EFIT does not provide accurate results.

The estimation of q at the axis is very difficult. Parameters related to the centre of the plasma can not be determined with good precision using only external data. It is commonly accepted that EFIT can provide q at the plasma axis with an error of 10÷15% for the standard plasma configurations that present a monotonic q profile. When the q profile is more complex then the error can be very large and the measurement cannot be used. It is known for example that the EFIT estimation of q_{axis} is not correct for the plasma configuration called optimised shear. From our point of view, the optimised shear plasmas differ from the standard plasmas in the pressure and current density profiles. The polynomial approximation used by EFIT is

not able to correctly represent these profiles with the results that also the q profile close to the axis is wrong.

This error of EFIT has been exploited to test the capability of neural networks to estimate these plasma parameters. New datasets were created considering the pulses where the optimised shear plasma configuration was not used. A new network was designed, and some tests were performed using optimised shear scenarios pulses to check the network outputs. A comparison with the EFIT output showed that the network was able to produce results similar to EFIT.

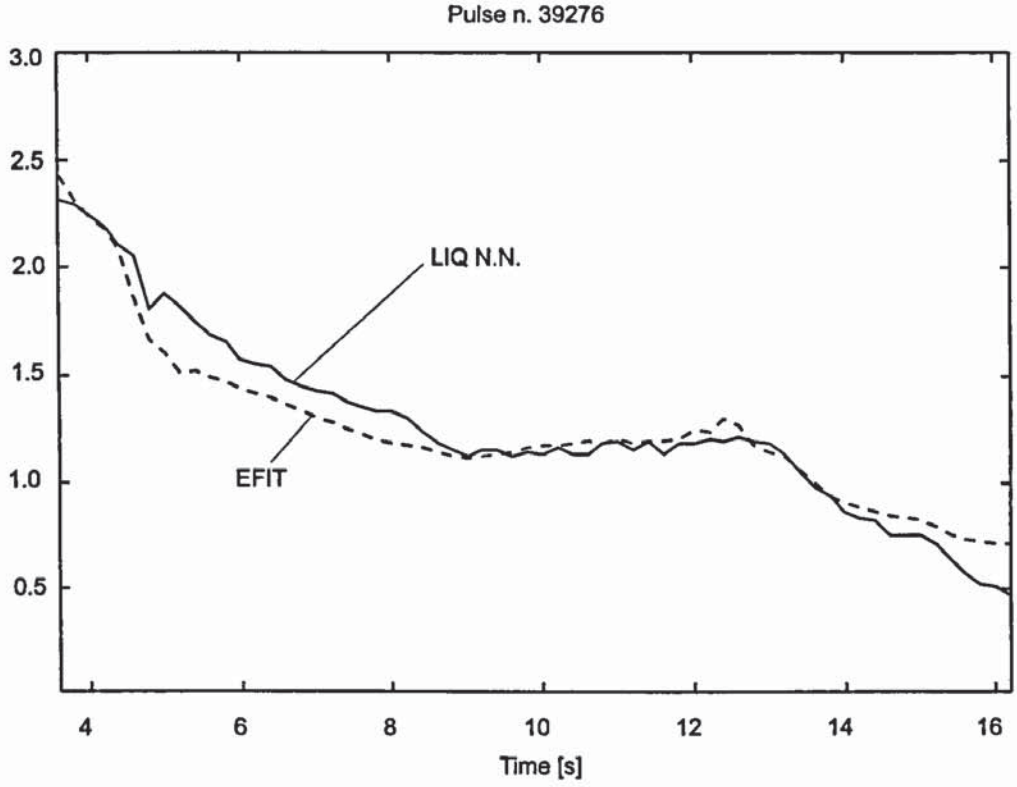

Figure 3.8: q_{axis} for an optimised shear plasma pulse. This plasma configuration type was excluded from the training dataset because it was evident that EFIT was producing wrong results for this specific configuration. The trained network output is anyway close to the values estimated by EFIT suggesting that the network was capable to learn the model of the plasma used by EFIT.

This suggests that we cannot expect more accurate results than EFIT from the network. At the same time we could say that the network learned the EFIT plasma model and therefore we could expect that even with novel data the output of the network will be as accurate as EFIT.

3.10 Conclusions

The two main goals of this part of this work were to see if there is a possibility to use in real time the ℓ_i-q_ψ diagram and if neural network algorithms could have been used to estimate plasma parameters.

A thorough study of the ℓ_i-q_ψ diagram with old disrupted pulses was done to check if it was possible to use it with the new JET configuration especially with x-point plasmas. This showed that the contour drawn for the pre-divertor campaign is not valid mainly because it was designed using q at the plasma surface that is not available for x-point plasmas. In order to use q_{95} new limits has to be set for the ℓ_i-q_ψ diagram. Up to now it has not been possible to define a separation line between good pulses and disrupted pulses just using ℓ_i and q_{95} that guarantees that good pulses will not be stopped. The approach chosen is to use the ℓ_i-q_ψ diagram only at the beginning of the pulse before the x-point formation. This excludes most of the disruptions occurring at JET and at the same time the possibility of stopping good pulses will be reduced to the minimum.

As regard to the use of neural network algorithms to determine in real time plasma parameters we have seen that they could provide sufficiently accurate results. Major difficulties on this approach are the selection of input and training data. The input data need to be reliable in order to avoid frequent retraining of the network because of malfunctioning of input signal sensors. For this reason, the XLOC algorithm has been chosen to be a pre-processing stage to the input network adding reliability to the network input and at the same time more information to the network with a lower number of input variables.

The resulting MLP network has 16 input and 18 hidden units. It provides results that are closer to EFIT and better than the FAST results available now in real time. Using novel data (optimised shear plasma) we have also seen that the error between EFIT and the network results does not increase.

4. ℓ_i-q_ψ diagram using the Bayesian approach

4.1 Introduction

In the previous chapter, the backpropagation method has been used to train an MLP network to determine internal plasma parameters. Despite the fact that the results provided by the final solution are sufficiently good, it would be beneficial to attach to the network outputs a confidence value to tell if new inputs fed into the network will perform as well as expected. The Bayesian approach for backpropagation, introduced by MacKay [32], not only offers the possibility of assigning error bars to the network outputs, but also it offers other important features including the possibility of comparing different network models using only training data, hence all available data can be used for training, which usually gives better results.

The Bayesian approach treats the issue of model design very differently from the design of a network based on maximum likelihood where a single set of parameters values is found. By contrast, the Bayesian framework considers probability distribution function over the parameter space, representing the relative degrees of belief in different values for the parameters.

A posterior probability is also considered for the whole model H_i, given the observed data $D\equiv(t^1,\ldots,t^N)$:

$$P(H_i|D)=\frac{p(D|H_i)P(H_i)}{p(D)} \tag{4.1}$$

The quantity $P(H_i)$ represent a prior probability for model H_i. If there is no particular reason to prefer one model over another, then equal priors will be assigned to all the models. Since

the denominator p(D) does not depend on the model, we can compare the different models by evaluating p($D|H_i$), which is called the *evidence* for the model H_i. The evidence measures how probable the model is, given the data, and therefore it provides a good method of comparing the various network models as it will be shown later.

4.2 Probabilistic interpretation of neural networks

As noted before, the Bayesian approach considers the probability distribution over the network parameters. Starting from the point that the architecture of the network (number of layers, number of hidden units, choice of activation functions, etc.) is given, and defining the vector of adaptive weights and bias as $\mathbf{w}=(w_1,\ldots,w_W)$, we can define the posterior probability distribution for the weights after the training data D have been observed as:

$$p(\mathbf{w}|D)=\frac{p(D|\mathbf{w})p(\mathbf{w})}{p(D)} \tag{4.2}$$

where the denominator is a normalisation factor which can be written as:

$$p(D)=\int p(D|\mathbf{w})p(\mathbf{w})d\mathbf{w} \tag{4.3}$$

p($\mathbf{w}$) is the probability distribution of the vector weights in absence of data (prior), and the p($D|\mathbf{w}$) is called the likelihood.

To determine p($D|\mathbf{w}$), the dataset $D\equiv((\mathbf{x}^1,\mathbf{t}^1),\ldots,(\mathbf{x}^N,\mathbf{t}^N))$ of the observed data is assumed to be independently drawn from a distribution P($\mathbf{x}$, $\mathbf{t}$). Then, we model $\mathbf{t}$ as a function of $\mathbf{x}$, $\mathbf{t}=\mathbf{y}(\mathbf{x},\mathbf{w})+\nu$, assuming the noise ν to be Gaussian with zero mean, and that $\mathbf{y}(\mathbf{x},\mathbf{w})$ are the outputs of our network. The probability of a single datum under this model is:

$$p(\mathbf{t}\,|\,\mathbf{x},\mathbf{w})=\sqrt{\frac{\beta}{2\pi}}\exp\left(-\frac{\beta}{2}\|\mathbf{y}(\mathbf{x},\mathbf{w})-\mathbf{t}\|^2\right) \tag{4.4}$$

The probability of the data becomes:

$$p(D\,|\,\mathbf{w})=\prod_{n=1}^{N}\prod_{k=1}^{c}p(t_k^n\,|\,\mathbf{x}^n,\mathbf{w})=\frac{1}{Z_D(\beta)}\exp\left(-\frac{\beta}{2}\sum_{n=1}^{N}\sum_{k=1}^{c}\{y_k(\mathbf{x}^n,\mathbf{w})-t_k^n\}^2\right) \tag{4.5}$$

Where $Z_D(\beta)$ is a normalisation factor given by

$$Z_D(\beta) = \int \exp(-\beta E_D) dD = \left(\frac{2\pi}{\beta}\right)^{(N*c)/2} \tag{4.6}$$

Where $\int dD = \int dt_1^1 ... dt_c^N$ represents an integration over the target variables, and E_D corresponds to the sum of square error:

$$E_D = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{c}\{y_k(\mathbf{x}^n, \mathbf{w}) - t_k^n\}^2 \tag{4.7}$$

The parameter β (also called hyperparameter) controls the variance of the noise and for the moment is treated as fixed.

After having defined the likelihood $p(D|\mathbf{w})$, we can now define the prior distribution $p(\mathbf{w})$. This distribution should reflect any prior knowledge we have about the form of network mapping we expect to find. In general we can write this distribution as an exponential of the form:

$$p(\mathbf{w}) = \frac{1}{Z_W(\alpha)}\exp(-\alpha E_W) \tag{4.8}$$

where $Z_W(\alpha)$ is a normalisation factor given by

$$Z_W(\alpha) = \int \exp(-\alpha E_W) d\mathbf{w} \tag{4.9}$$

and which ensures that the integral of $p(\mathbf{w})$ over the weight space is one.

Assuming that the functions that we wish to approximate are smooth, uniform and small values of the weights could increase the generalisation capability of the network. We can encourage this by using E_W defined as:

$$E_W = \frac{1}{2}\|\mathbf{w}\|^2 = \frac{1}{2}\sum_{i=1}^{W} w_i^2 \tag{4.10}$$

which corresponds to the use of a simple weight-decay regularizer. The prior then becomes:

$$p(\mathbf{w}) = \frac{1}{Z_W(\alpha)}\exp\left(-\frac{\alpha}{2}\sum_{i=1}^{W} w_i^2\right) \tag{4.11}$$

with:

$$Z_W(\alpha) = \left(\frac{2\pi}{\alpha}\right)^{W/2} \tag{4.12}$$

The parameter α is also called hyperparameter since it controls the distribution of other parameters. How to treat α as part of the learning process will be treated in paragraph 4.3.

Finally, we can rewrite the posterior weight distribution of (4.2) using equation (4.5) and (4.11) we can rewrite the posterior distribution:

$$p(\mathbf{w}\,|\,\alpha,\beta,D)=\frac{1}{Z_S}\exp(-\beta E_D-\alpha E_W)=\frac{1}{Z_S}\exp(-S(\mathbf{w})) \tag{4.13}$$

where

$$S(\mathbf{w})=\beta E_D+\alpha E_W \tag{4.14}$$

and

$$Z_S(\alpha,\beta)=\int\exp(-\beta E_D-\alpha E_W)d\mathbf{w} \tag{4.15}$$

If we now consider α and β fixed, the problem of finding the weight vector $\mathbf{w}_{MP}$ corresponding to the maximum of the posterior distribution can be solved minimising the negative logarithm of with respect to the weights. Since the normalising factor Z_S is independent of the weights, we see that it is equivalent to minimising S(w). For the particular prior distribution and noise model given by we can write:

$$S(\mathbf{w})=\frac{\beta}{2}\sum_{n=1}^{N}\left\|\mathrm{y}(\mathbf{x}^n,\mathbf{w})-\mathrm{t}^n\right\|^2+\frac{\alpha}{2}\sum_{i=1}^{W}w_i^2 \tag{4.16}$$

This, apart for an overall multiplicative factor is equivalent of minimising the sum-of-square function with a weight decay regularisation term.

It is important to note that the solution found does not take into account the numerous local minima of the error function, some of which may be associated with the symmetries in the network. In our case, the fully connected networks, with one hidden layer, that we are going to train will have 2^M M! equivalent minima.

Up to now, we have implied that the values of the hyperparameters α and β are known. In reality, those values are unknown, therefore using the Bayesian approach, we have to integrate them out of any predictions, for example the posterior deistribution of the network weights is given by:

$$p(\mathbf{w}\,|\,D)=\iint p(\mathbf{w}\,|\,\alpha,\beta,D)\,p(\alpha,\beta\,|\,D)\,d\alpha d\beta \tag{4.17}$$

Two approaches to the treatment of the hyperparameters have been discussed in the literature. One of these perform the integrals over α and β analytically, the other known as evidence approximation [32] [6] has been used here and discussed in the next paragraph.

4.3 α and β hyperparameters

The evidence approximation approach for the determination of α and β is based on the fact that the posterior probability distribution $p(\alpha,\beta|D)$ is sharply peaked around their most probable values α_{MP} and β_{MP}. Then (4.17) can be written as:

$$p(\mathbf{w}\,|\,D) \approx p(\mathbf{w}\,|\,\alpha_{MP},\beta_{MP},D)\iint p(\alpha,\beta\,|\,D)d\alpha d\beta = p(\mathbf{w}\,|\,\alpha_{MP},\beta_{MP},D) \tag{4.18}$$

This tells that the hyperparameters that maximise the posterior probability should be found before performing the remaining calculations.

Using the Bayes' theorem, the posterior probability can be written as:

$$p(\alpha,\beta\,|\,D) = \frac{p(D\,|\,\alpha,\beta)p(\alpha,\beta)}{p(D)} \tag{4.19}$$

Then, we can concentrate to the maximisation of the evidence $p(D|\alpha,\beta)$, if we suppose that the prior over the hyperparameters is insensitive to the values of α and β to reflect the fact that we have little idea of suitable values for these quantities.

If we make the dependencies on α and β explicit, the evidence can be written as:

$$p(D\,|\,\alpha,\beta) = \int p(D\,|\,\mathbf{w},\alpha,\beta)p(\mathbf{w}\,|\,\alpha,\beta)d\mathbf{w} = \int p(D\,|\,\mathbf{w},\beta)p(\mathbf{w}\,|\,\alpha)d\mathbf{w} \tag{4.20}$$

where we have made use that the prior is independent of β and the likelihood function is independent of α. Using the exponential forms (4.8) and (4.5) we can write $p(D|\alpha,\beta)$ in the form:

$$p(D\,|\,\alpha,\beta) = \frac{1}{Z_D(\beta)}\frac{1}{Z_W(\alpha)}\int \exp(-S(\mathbf{w}))d\mathbf{w} = \frac{Z_S(\alpha,\beta)}{Z_D(\beta)Z_W(\alpha)} \tag{4.21}$$

In order to evaluate the evidence we already know $Z_D(\beta)$ and $Z_W(\alpha)$, and we can approximate $S(\mathbf{w})$ by considering its Taylor expansion as suggested by MacKay [32]:

$$S(\mathbf{w}) = S(\mathbf{w}_{MP}) + \frac{1}{2}\Delta\mathbf{w}^T\mathbf{A}\Delta\mathbf{w} \tag{4.22}$$

where $\Delta \mathbf{w} = \mathbf{w} - \mathbf{w}_{MP}$ with $\mathbf{w}_{MP}$ defined as the most probable value, and **A** is:

$$\mathbf{A} = \beta \nabla \nabla E_D + \alpha \mathbf{I} = \mathbf{H} + \alpha \mathbf{I} \tag{4.23}$$

with H the Hessian matrix of the unregularised error function.

Then $Z_S(\alpha,\beta)$ becomes:

$$Z_S(\alpha,\beta) = e^{-S(\mathbf{w}_{MP})} (2\pi)^{W/2} |\mathbf{A}|^{-1/2} \tag{4.24}$$

The log of the evidence becomes:

$$\ln(p(D|\alpha,\beta)) = -\alpha E_W^{MP} - \beta E_D^{MP} - \frac{1}{2}\ln|\mathbf{A}| + \frac{W}{2}\ln\alpha + \frac{N*c}{2}\ln\beta - \frac{N*c}{2}\ln(2\pi) \tag{4.25}$$

To search the maximum with respect to α we simply derive the equation (4.25) and force it to zero:

$$\frac{\partial \ln(p(D|\alpha,\beta))}{\partial \alpha} = -E_W^{MP} - \frac{1}{2}\frac{\partial \ln|\mathbf{A}|}{\partial \alpha} + \frac{W}{2}\frac{1}{\alpha} = 0 \tag{4.26}$$

The derivative of the determinant of A is easily calculated if we use the fact that it is equivalent to the product of its eigenvalues which can be written as $\lambda_i+\alpha$. The λ_i are the eigenvalues of H. If we neglect the terms involving the derivative of λ_i with respect to α which are present since H is evaluated at $\mathbf{w}_{MP}$, and $\mathbf{w}_{MP}$ depend on α we obtain:

$$2\alpha E_W^{MP} = W - \sum_{i=1}^{W} \frac{\alpha}{\lambda_i + \alpha} = \gamma \tag{4.27}$$

Similarly the maximum of the log evidence $p(D|\alpha,\beta)$ can be found as:

$$2\beta E_W^{MP} = N*c - \sum_{i=1}^{W} \frac{\lambda_i}{\lambda_i + \alpha} = N*c - \gamma \tag{4.28}$$

These equations allow us to determine the hyperparameters α and β given the approximation that the posterior distributions $p(\mathbf{w}|D)$ described by a single Gaussian function of the weights and that we are using a particular set of $\mathbf{w}_{MP}$ weights. The approach adopted by MacKay is to note that we are working in a particular local minima of $S(\mathbf{w})$ and therefore the value determined for α and β are appropriate for this particular solution but different minima

may require different values of these parameters. Therefore the integral (4.20) has to be interpreted as an integral over the particular local minimum being considered.

4.4 Training of the network

The assumption done in the previous paragraphs in order to determine the most probable weights and hyperparameters suggests a set of steps to implement the Bayesian techniques to train the network:

1. choose the initial values for the hyperparameters α and β. Initialise the weights in the network using values drawn from the prior distribution.
2. Train the network using an optimisation algorithm to minimise the total error function S(w). In this case the same method described in chapter 3 has been used. Also training dataset is the same so that we can easily compare the performance of the two different networks.
3. Every few cycles of the algorithm, re-estimate values for α and β using (4.27) (4.28). This requires the evaluation of the Hessian matrix and of its eigenvalue spectrum.
4. Steps 1-3 have been repeated several times for different random initial choices for the network weights and for different network models modifying the number of hidden units.

The Hessian matrix **H** is determined using the outer product approximation where using the sum of square error function the Hessian can be written in the form:

$$\frac{\partial^2 E_D}{\partial w_{ji}\partial w_{lk}} = \sum_n \frac{\partial y^n}{\partial w_{ji}}\frac{\partial y^n}{\partial w_{lk}} + \sum_n (y^n - t^n)\frac{\partial^2 y^n}{\partial w_{ji}\partial w_{lk}} \tag{4.29}$$

Being the training dataset made of observed data independently drawn and normalised, the second term may be neglected with the assumption that for a trained network the quantity $(y^n - t^n)$ is a random variable with zero mean, and it is uncorrelated with the value of the second derivative term. To verify the correctness of our calculation we have also used numerical differentiation to determine the Hessian matrix. For this purpose the central differences method has been applied to the first derivatives of the error function.

This gives:

$$\frac{\partial^2 E_D}{\partial w_{ij}\partial w_{lk}} = \frac{1}{2\varepsilon}\left\{\frac{\partial E_D}{\partial w_{ij}}(w_{lk}+\varepsilon) - \frac{\partial E_D}{\partial w_{ij}}(w_{lk}-\varepsilon)\right\} + o(\varepsilon^2) \tag{4.30}$$

where the first derivatives of the error are calculated using the back-propagation method.

The training has been performed using the same training set used in chapter 3 and it has been performed for various networks varying the number of hidden units.

4.5 Network model comparison

The model comparison has been done using the posterior probability of the various models H_i once the data D of the training set has been observed. Using the Bayes' theorem, this probability can be written as:

$$P(H_i|D) = \frac{p(D|H_i)P(H_i)}{p(D)} \tag{4.31}$$

where $P(H_i)$ is the prior probability assigned to model H_i, and the quantity $p(D|H_i)$ is referred to as the evidence for H_i. If there is no reason to assign different priors for different models then the comparison between models can be done using only the evidence that can be written as:

$$p(D|H_i) = \iint p(D|\alpha,\beta,H_i)p(\alpha,\beta \mid H_i)d\alpha d\beta \tag{4.32}$$

To evaluate this integral we make a further assumption, we impose that the logarithm of the evidence of the hyperparameters is a quadratic form and that here is no correlation between the two. In the case of the β we obtain:

$$\ln p(D|\beta,H_i) = \ln p(D|\beta_{MP},H_i) - \frac{1}{2}\frac{(\ln\beta - \ln\beta_{MP})^2}{\sigma^2_{\ln\beta}} \tag{4.33}$$

Where

$$\frac{1}{\sigma^2_{\ln\beta}} = -\beta\frac{\partial}{\partial\beta}\left(\beta\frac{\partial}{\partial\beta}\ln p(D|\beta,H_i)\right) \tag{4.34}$$

Which can be calculated using (4.25) and (4.27):

$$\frac{1}{\sigma^2_{\ln\beta}} = \frac{1}{2}(N*c-\gamma) + \frac{1}{2}\sum_{i=1}^{W}\frac{\alpha\lambda_i}{(\alpha+\lambda_i)^2} \tag{4.35}$$

The second term can be neglected since the only significant contributions to the sum arise from the form the λ_i that are of the same order of α which will be very few. Similarly for α we can write:

$$\frac{1}{\sigma^2_{\ln\alpha}} = \frac{1}{2}\gamma + \frac{1}{2}\sum_{i=1}^{W}\frac{\alpha\lambda_i}{(\alpha+\lambda_i)^2} \tag{4.36}$$

If we then go back to the determination of the evidence (4.32) we can calculate it using the formulas just written and separating the two integrals:

$$\begin{aligned} p(D|H_i) &= \int\left(\int p(D|\alpha,\beta,H_i)p(\ln\beta|H_i)d\ln\beta\right)p(\ln\alpha|H_i)d\ln\alpha \\ &\approx \int p(D|\alpha,\beta_{MP},H_i)p(\ln\alpha|H_i)d\ln\alpha \int\exp(-\frac{(\ln\beta-\ln\beta_{MP})^2}{2\sigma^2_{\ln\beta}})\frac{1}{\ln\Omega}d\ln\beta \\ &\approx \int p(D|\alpha,\beta_{MP},H_i)p(\ln\alpha|H_i)d\ln\alpha \frac{(2\pi)^{1/2}\sigma_{\ln\beta}}{\ln\Omega} \end{aligned} \tag{4.37}$$

Where the prior distribution for $\ln\beta$ is set to be constant over some large region $\ln\Omega$ which encompasses β_{MP} as well as most of the probability mass of the Gaussian region. Similar arguments can be applied to α obtaining for the result:

$$p(D|H_i) \approx p(D|\alpha_{MP},\beta_{MP},H_i)2\pi\frac{\sigma_{\ln\beta}}{\ln\Omega}\frac{\sigma_{\ln\alpha}}{\ln\Omega} \tag{4.38}$$

An expression for $p(D|\alpha_{MP},\beta_{MP},H_i)$ can be obtained using (4.25) and setting the hyperparameters to their most probable value. The result (4.25) was obtained by integrating over the posterior distribution $p(\mathbf{w}|D,H_i)$ represented by a single Gaussian. As remarked before, the Gaussian approximation does not take account of any symmetries of the network, which implies the presence of $2^M M!$ local minima. However, the evidence that we are trying to determine depends on the ratio of the volume of the posterior distribution in weight space to the volume of the prior. The expression for the prior takes into account of the equivalent configurations related to the symmetries of the network (it has been defined as a Gaussian centred in the origin). Therefore, we have to include an extra factor to our posterior

distribution that takes into account this feature. If we assume that the posterior distribution in he weight space can be represented by a sum of Gaussian functions centred on each such minimum and that the overlap between the various Gaussians is negligible, we can just multiply $p(\mathbf{w}|D,H_i)$ by the number of local minima to take into account of the total volume of the posterior distribution in the weight space. It has to be noted that this approximation does not take into account the presence of local minima not related by symmetry transformation.

Rather than evaluate the evidence $p(D|H_i)$, it is more convenient to consider its logarithm which can be no easily determined. Omitting the terms that are the same for different networks, we obtain:

$$\ln p(D|H_i) = -\alpha_{MP}E_W^{MP} - \beta_{MP}E_D^{MP} - \ln|\mathrm{A}| + \frac{W}{2}\ln\alpha_{MP} + \frac{N*c}{2}\ln\beta_{MP} \qquad (4.39)$$
$$+ \ln M! + 2\ln M + \frac{1}{2}\ln\left(\frac{2}{\gamma}\right) + \frac{1}{2}\ln\left(\frac{2}{N*c-\gamma}\right)$$

In practice, the accurate evaluation of the evidence can be very difficult not only because of the Gaussian approximation to the posterior distribution. Another reason is its dependence on the Hessian determinant. This, contrary to the evaluation of the hyperparameters, depends on the product of the eigenvalues and therefore is very sensitive to errors in the small eigenvalues. In the case studied, to overcome to the poor confidence we have on the evidence evaluation and to crosscheck the results obtained by this method with the results in chapter 3, we have trained the networks using the same training dataset. We have then used the validation dataset to confirm that the best performing network determined using the maximum value for the evidence, is the same network that better performs with the validation set.

Figure 4.1 and Figure 4.2 show that the best performing network is the same in both cases. The size of the network is slightly different from the network found in chapter 3, 20 hidden units compared to 18. At the same time, a comparison of the validation dataset RMS error of the two networks shows that there is not an evident improvement, probably due to the fact that the network has been trained with the same dataset, and the network in chapter four has been chosen because of its performance with the validation set.

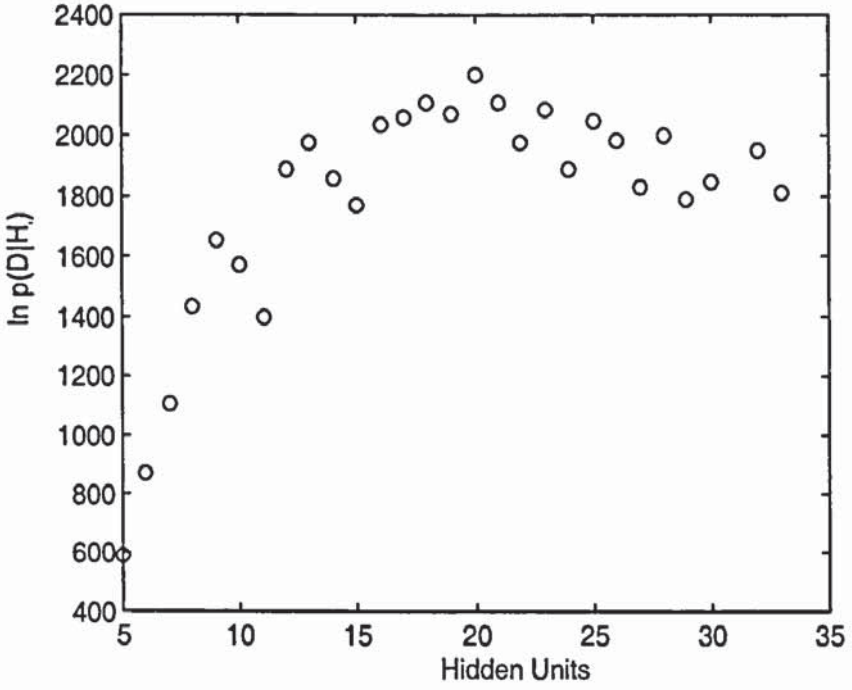

Figure 4.1: Comparison between the various trained networks using the log of the evidence $p(D|H_i)$. The diagram shows that for networks with more than 20 hidden units the augmented complexity does not increase the performance of the network. The hyperparameters for the chosen network of 20 hidden units are α_{MP}=0.46 and β_{MP}=16.51.

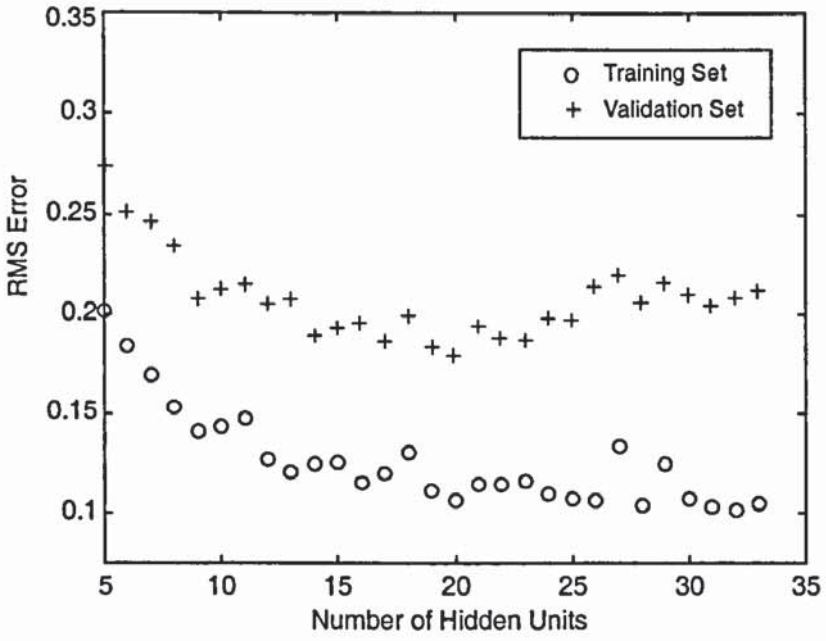

Figure 4.2: The diagram shows the trend of the RMS Error as a function of the network size for the training and the validation set. The set are the same used in chapter 3 and the trend of the error for the validation set confirm that the best performing network for data not used for training is the network with 20 hidden units.

4.6 Error bars

Using a Gaussian approximation for the posterior distribution (4.22), and the expression for the probability of observing a data value t for a given input vector (4.5), we can write the distribution of outputs as:

$$p(\mathbf{t}|\mathbf{x},D)=\int p(\mathbf{t}|\mathbf{x},\mathbf{w})p(\mathbf{w}|D)d\mathbf{w}\propto\int\exp(-\frac{\beta}{2}\|\mathbf{t}-\mathbf{y}(\mathbf{x},\mathbf{w})\|^2)\exp(-\frac{1}{2}\Delta\mathbf{w}^T\mathbf{A}\Delta\mathbf{w})d\mathbf{w} \quad (4.40)$$

In addition, if the posterior distribution is sufficiently narrow, we may approximate $\mathbf{y}(\mathbf{x},\mathbf{w})$ with its linear expansion around $\mathbf{w}_{MP}$:

$$\mathbf{y}_k(\mathbf{x},\mathbf{w})=\mathbf{y}_k(\mathbf{x},\mathbf{w}_{MP})+\mathbf{g}_k^T\Delta\mathbf{w} \quad (4.41)$$

where $\mathbf{g}_k$ is the gradient of the k-th output y_k calculated in $\mathbf{w}_{MP}$: $\mathbf{g}_k\equiv\nabla_{\mathbf{w}}y_k\big|_{\mathbf{w}_{MP}}$.

This allows equation to be written as:

$$p(\mathbf{t}|\mathbf{x},D)\propto\int\exp(-\frac{\beta}{2}\left\|\mathbf{t}-\mathbf{y}_{MP}-\mathbf{G}^T\Delta\mathbf{w}\right\|^2-\frac{1}{2}\Delta\mathbf{w}^T\mathbf{A}\Delta\mathbf{w})d\mathbf{w} \quad (4.42)$$

where $\mathbf{y}_{MP}=\mathbf{y}(\mathbf{x};\mathbf{w}_{MP})$.The integral can be evaluated to give the distribution of outputs in the form of a Gaussian:

$$p(\mathbf{t}|\mathbf{x},D)=\frac{1}{(2\pi)^{c/2}|\Sigma_t|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{t}-\mathbf{y}_{MP})^T\Sigma_t(\mathbf{t}-\mathbf{y}_{MP})\right) \quad (4.43)$$

where $\mathbf{y}_{MP}$ is the mean of the distribution and the covariance matrix is given by:

$$\Sigma_t=\frac{1}{\beta}\mathbf{I}_c+\mathbf{G}^T\mathbf{A}^{-1}\mathbf{G} \quad (4.44)$$

The covariance matrix Σ_t is then simplified to its diagonal supposing the various outputs statistically independent. This provides the possibility of considering the probability as the product of the probability relative to each single output. We can then interpret the standard deviation σ_{kt} (square if the k-th element of the diagonal of Σ_t) as an error bar on the mean value y_{MP} of the k-th output. This error bar has two contributions: one arising from the intrinsic noise on the target data, corresponding to the first term, and one from the width of the posterior distribution of the network weights, corresponding to the second term.

If we draw the output of the network for the whole length of a pulse plotting also the $\pm 2\sigma_{kt}$ error bars as dotted green lines. It is possible to see that for q_{axis}, which is not an accurate measurement, the error bars are relatively large. This could be caused by the fact that the training dataset does not cover sufficiently that particular phase of the pulse. Therefore the uncertainty on the network output increases because the input values are novel to the network.

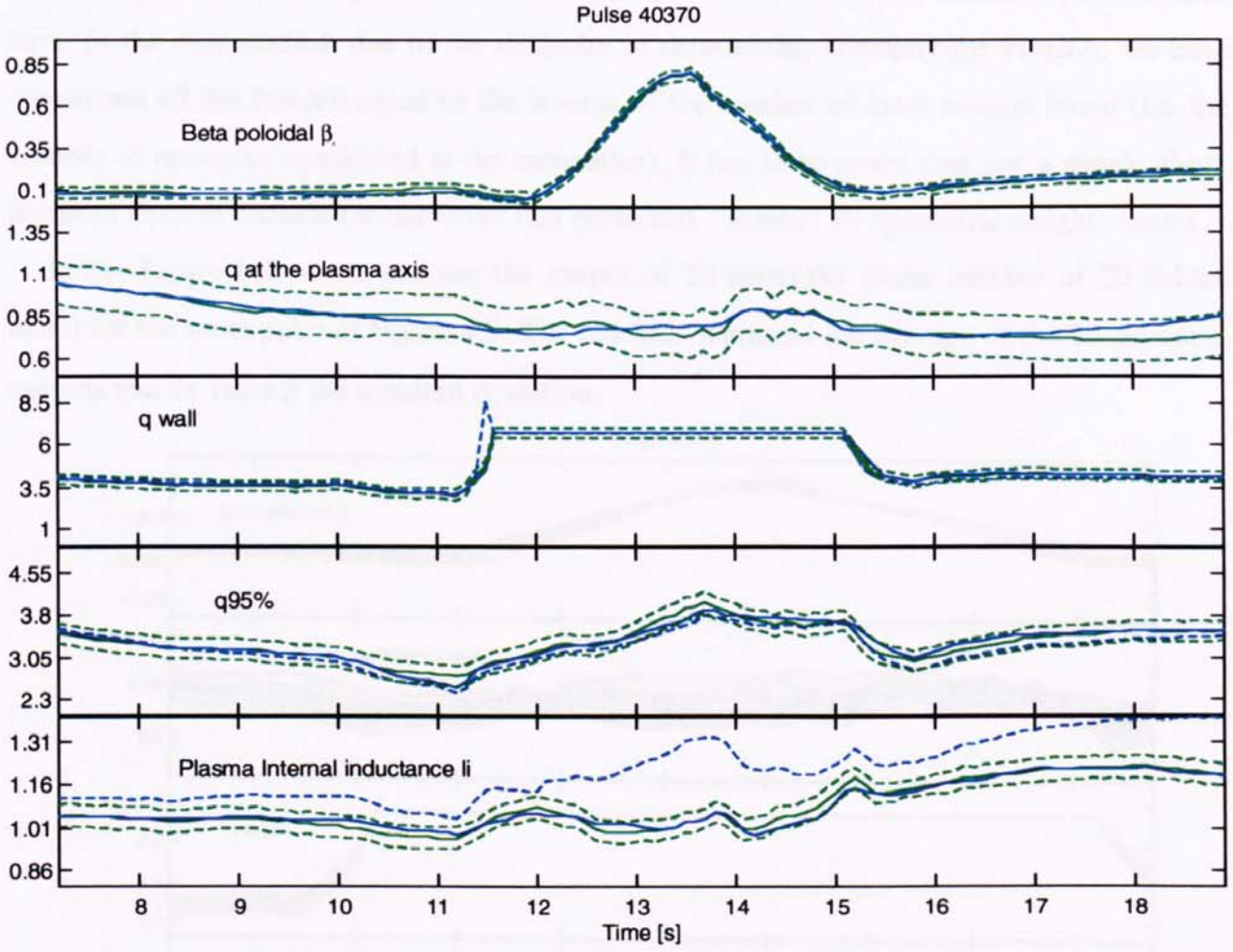

Figure 4.3: Error bars for a test plasma pulse (dotted green). The EFIT measurement (blue) stays within the error bars of the trained network. The blue dotted lines represent the FAST outputs when available. The green lines represent the network output.

It has to be noted that again in this case we are using the Gaussian approximation for the posterior distribution of the weights. This does not take into account the presence of other local minima that do not arise from symmetries of the network. A way to overcome this problem is to train the network several times starting from different random initial weight configurations. Then, we can model our posterior distribution using a set of Gaussians, one

centred on each local minimum, in which we assume that there is negligible overlap. The posterior distribution than becomes:

$$p(\mathbf{w} \mid D) = \sum_i p(\mathbf{w} \mid m_i, D) P(m_i \mid D) \tag{4.45}$$

where the dependence on the minimum m_i has been made explicit. We can then use this posterior probability to improve the accuracy of the model and of the estimation of the error bars. In the case studied, due to the difficulty of determining correctly the P(m_i|D), we have considered all the P(m_i|D) equal to the inverse of the number of local minima found (i.e. the number of networks considered in the committee). It has to be noted that just a simple check has been done to exclude the networks that presented "similar" or symmetric weights values.

In the Figure below, we can see the output of 20 networks (same number of 20 hidden units) for the same pulse of Figure 4.3. The red lines represent the average of the 20 networks outputs and its value ± the standard deviation.

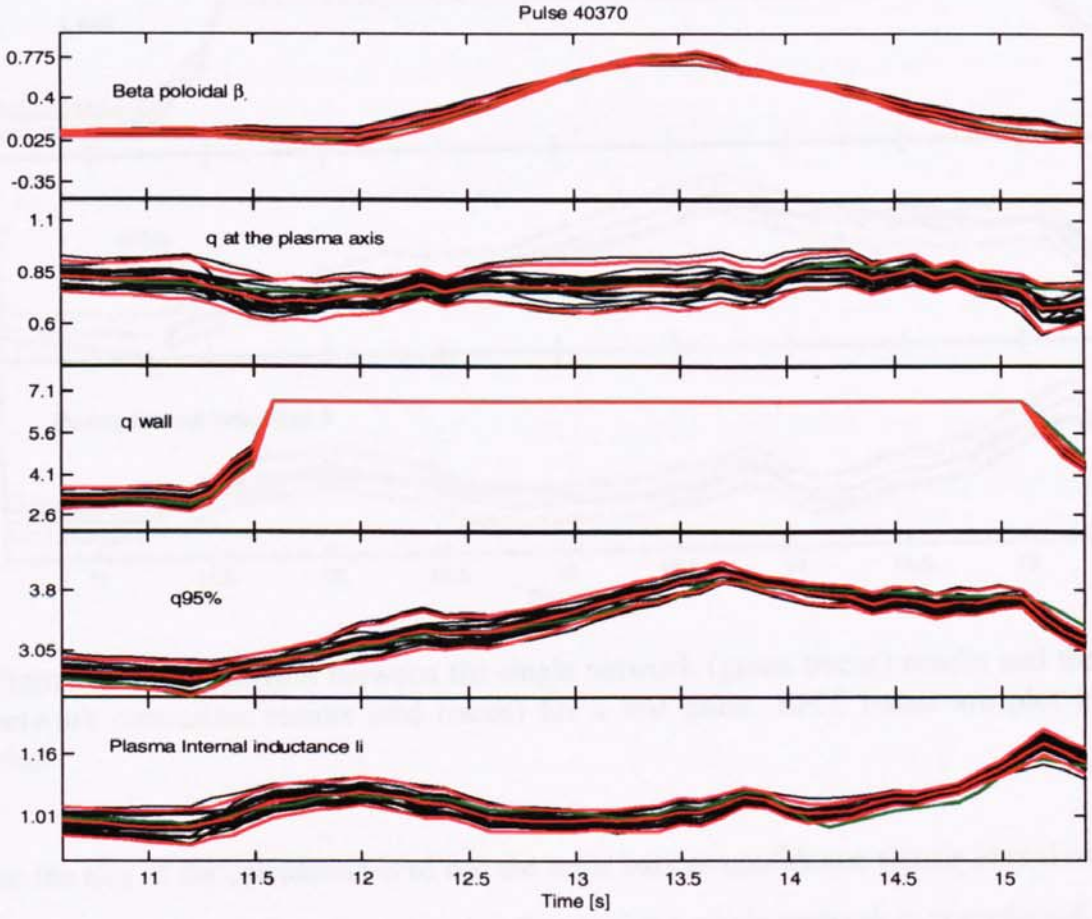

Figure 4.4: The 20 best network with 20 hidden units have been plotted. The red lines represent the average of the network outputs and twice the standard deviation. EFIT output is plot in green.

For comparison purposes, the committee output has been plot together with the single network output. The committee outputs are the average of the outputs of the networks, and its variance has been determined as the average of the networks variances determined as described above plus the variance of the committee network output [56].

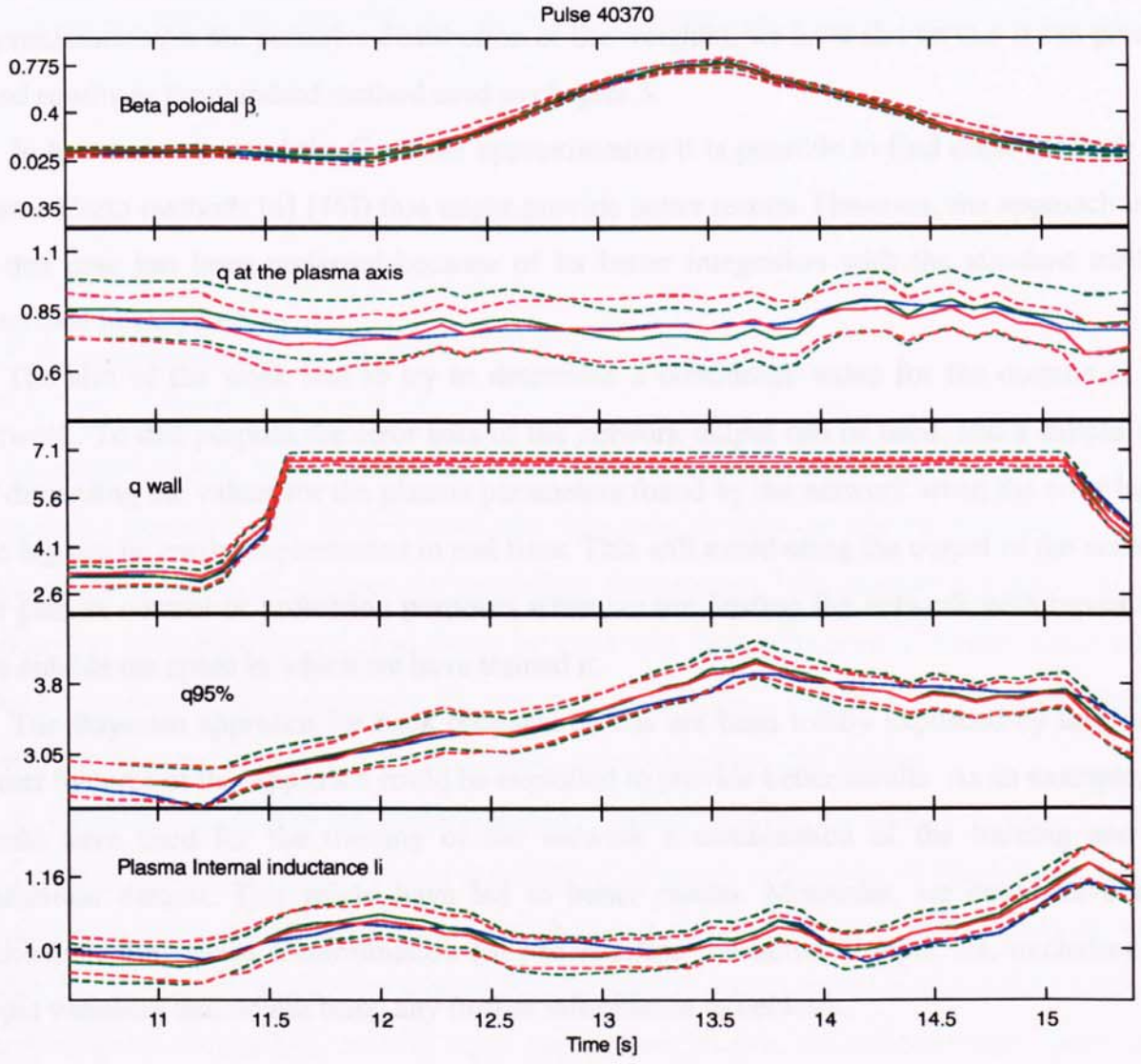

Figure 4.5: Comparison between the single network (green traces) results and the network committee results (red traces) for a test pulse. EFIT traces are plot in blue.

Since the aim of the calculation is to use the error bars as confidence signals in real time, the less computational demanding determination done using a single network in is preferred.

4.7 Conclusions

Despite the fact that the Bayesian approach, as it has been considered in this chapter, forced us to make a certain number of assumptions (particularly the fact that we are using Gaussian approximation for the posterior distribution of the weights), we have shown that it can give as good results as the standard method used in chapter 3.

In literature, to avoid the Gaussian approximation it is possible to find other methods (as Monte Carlo methods [6] [46]) that might provide better results. However, the approach used in this case has been preferred because of its better integration with the standard method described in chapter 3.

The aim of the work was to try to determine a confidence value for the outputs of the network. To this purpose the error bars of the network output can be used, and a simple way of discarding the values for the plasma parameters found by the network when the error bar is too big can be easily implemented in real time. This will avoid using the output of the network for plasma control or protection purposes when we are feeding the network with inputs that are outside the space in which we have trained it.

The Bayesian approach for back propagation has not been totally exploited by this work. Other features of this approach could be exploited to provide better results. As an example, we could have used for the training of the network a combination of the training and the validation dataset. This might have led to better results. Moreover, we could have used Automatic Relevance Determination [38] to redefine the network input set, excluding the input variables that do not bring any further information to network.

5. Disruption prediction using neural network techniques

5.1 Introduction

In chapter 3 we have considered a disruption prediction technique that relies on the experience gained at JET. Neural networks were used to estimate plasma parameters needed to implement the ℓ_i-q_ψ diagram for predicting disruptions. This method proved to be very difficult to implement in real time. It can be used only during the plasma limiter phase and its accuracy proved to be insufficient for detecting the plasma configurations that could disrupt without risking to misclassify a considerable amount of successful pulses.

A different approach to the problem is to use neural network techniques to directly predict disruptions. Using the classification capabilities of neural networks, we tried to classify the 'unstable' plasma configurations that could lead to a disruption. The challenge is to recognise those disruption precursors which are difficult to detect if each signal is taken on its own. The possibility of considering several input signals mapping a multidimensional space could provide enough information to be able to recognise plasma disrupting patterns in advance.

Moreover the use of supervised networks that are trained using a known dataset of disrupted pulses could help to find a mapping of the JET machine operational space that could be marked as dangerous for plasma experiments.

A suitable network for this purpose is the MLP network with the softmax activation function for the output layer. The output of the network will be the posterior probability that the input patterns belongs to a given class of disrupting pulses or to successful pulses.

5.2 Softmax multilayer perceptron network

We can consider the problem of predicting disruptions as a classification problem where the network output represents the probability of the presence or absence of attributes which could be related to a disruption. Using a statistical approach the output of the network is the posterior probability that the input **x** belongs to the class C_k

$$y_k = P(C_k|\mathbf{x}) \tag{5.1}$$

As in paragraph 1.9 we can use the concept of maximum likelihood to devise the error function [8]. If the distribution of the individual target variables is independent, the conditional distribution for a pattern can be written as:

$$p(\mathbf{t}^n|\mathbf{x}^n) = \prod_{k=1}^{c} (y_k^n)^{t_k^n} \tag{5.2}$$

The target vector elements t_k^n are all zeros except for the element relative to the class the input vector $\mathbf{x}^n$ belongs which has value one.

The likelihood can be easily calculated:

$$\mathcal{L} = \prod_{n=1}^{N} p(\mathbf{t}^n | \mathbf{x}^n) = \prod_{n=1}^{N} \prod_{k=1}^{c} (y_k^n)^{t_k^n} \tag{5.3}$$

If instead of maximising the likelihood function we then take its negative logarithm and minimise it. We obtain the error function:

$$E = -\sum_{n}^{N} \sum_{k'} t_{k'}^n \ln(y_{k'}^n) \tag{5.4}$$

As activation function for this network we have to consider a function whose values must lie in the range (0,1) and all the outputs must sum to unity. This can be achieved by using the softmax function:

$$\text{softmax}(a_k) = \frac{e^{a_k}}{\sum_{k'} e^{a_{k'}}} \tag{5.5}$$

This can be seen as a generalisation of the logistic sigmoid activation function $\frac{1}{1+\exp(-a)}$. A general motivation can be given starting from the posterior probability that a hidden unit activation vector $\mathbf{z}$ belongs to class C_k [8]. Supposing that the class conditional densities belong to the family of distributions of the form:

$$p(\mathbf{z}\,|\,C_k) = \exp\left\{A(\theta_k) + B(\mathbf{z},\phi) + \theta_k^T \mathbf{z}\right\} \tag{5.6}$$

then the output of the network can be written as posterior probability of class C_k and using Bayes' theorem:

$$p(C_k\,|\,\mathbf{z}) = \frac{p(\mathbf{z}\,|\,C_k)P(C_k)}{\sum_{k'} p(\mathbf{z}\,|\,C_{k'})P(C_{k'})} \tag{5.7}$$

Substituting (5.6) into (5.7) we obtain the softmax activation function:

$$p(C_k\,|\,\mathbf{z}) = \frac{e^{a_k}}{\sum_{k'} e^{a_{k'}}} \tag{5.8}$$

where:

$$\begin{aligned} a_k &= \mathbf{w}_k^T \mathbf{z} + w_{k0}^{(2)} \\ \mathbf{w}_k^T &= \theta_k \\ w_{k0}^{(2)} &= A(\theta_k) + \ln P(C_k) \end{aligned} \tag{5.9}$$

This approach gives us the possibility of classifying the data into several classes. The advantage is that the disruptions could be classified by type, giving a more precise answer about the causes of a disruptions. Then, the remedies that can be applied to avoid the disruption could be more specific increasing the chances of success.

Unfortunately due to the lack of a sufficiently big database of disruptions this differentiation was not possible. The network has been trained with just two classes. Therefore, a simpler network with a single output and with a logistic sigmoid as activation function could have been used [6]. Nevertheless, a more general approach and the possibility to upgrade in the future the network has been considered a sufficient reason for using a more complex network.

Another important issue on the design of the network is the choice of the input variables. The difficulty in this case is not the scarcity of measurements that can be used but the lack of input patterns for the disrupting plasmas dataset.

5.3 Dataset Generation

One of the major problems of designing a network for disruption predictions is that the dataset of disrupted pulses is not very large. At JET this work started with a dataset of only 360 disruptions with plasma current above 1MA (lower plasma currents were not considered because the accuracy of the parameters which could be used to train the network, is not sufficient). The size of the disruption database not only could affect the generality of the network but it also limits the complexity of the network, since in order to avoid over fitting problems during the training we would like to keep the network parameters considerably lower than the number of input patterns. For these reasons, extensive work was done to select the measurements to be used as network inputs. In order to maximise the possibility of predicting a disruption, all signals commonly used to establish its causes have been considered together with the methods used presently to predict a disruption.

To overcome the problem of a small dataset, the Bayesian approach could have been used, but this work has been performed before the start of the analysis of the Bayesian approach and there has not been time to redo it in the light of the latest work.

At JET mainly three methods are used to predict and to avoid a disruption: the ℓ_i and q diagram, the mode-lock detection, and the plasma input/radiated power ratio.

5.3.1 ℓ_i-q diagram

The ℓ_i-q_ψ diagram method has been discussed extensively in the previous chapters. Here, the importance of the two plasma parameters has to be stressed, and particularly the combination of low ℓ_i values with rational q values. Although the limits for an x-point plasma are not yet defined, ℓ_i and q give useful information on the plasma stability, therefore the q_{95} and ℓ_i will be used as input for the disruption prediction network. Moreover, the hard limit for

$q = 2$ should be taken into account, since we know that a value of $q = 2$ at the plasma surface will cause an immediate disruption. The value of ℓ_i and q will be provided by the neural network designed in the previous chapters.

5.3.2 Mode lock detection

Analysis of previous disruptions shows that 93% of them presents a locked mode [48]. Unfortunately, for some of them the appearance of the locked mode is too close to the disruption time, and the control systems do not have the time to react.

Nevertheless, at JET the mode lock disruption prediction algorithm is generally used during operation. This disruption prediction algorithm is based on the measurement of the amplitude of the magnetic islands that grow in the plasma. The islands are usually rotating along the magnetic field lines, but growing in amplitude they start to interact with the conducting vessel wall [37]. The interaction leads to the locking of the island and to an increased loss of plasma energy through it. Eventually, the plasma confinement is destroyed and a disruption occurs.

For these reasons the growth of magnetic islands during a pulse is monitored and the signal is used to stop the pulse before the island size reaches a certain value. The measurement is performed using a set of four saddle loops mounted in four octants displaced by 90 degrees. If we subtract the measurements of two opposite saddle loops we obtain a component of the field generated by the magnetic island.

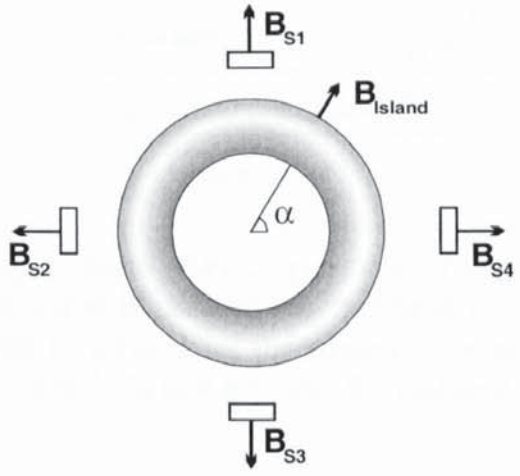

Figure 5.1: View of the plasma (grey torus) from the top. The saddle loops are represented by the squares. The magnetic island effect is represented by its perturbation of the plasma radial field

The same can be done using the other two opposite octants obtaining the orthogonal component of the magnetic island radial field:

$$B_{island}\sin(\alpha)=\frac{B_{s1}+B_{s3}}{2} \qquad B_{island}\cos(\alpha)=\frac{B_{s2}+B_{s4}}{2} \tag{5.10}$$

Then, the amplitude of the magnetic field perturbation generated by the island can be easily determined. The measurement is proportional to the island and can be used to predict disruptions.

In the figure below an example of the use of this disruption precursor can be seen. When a predefined threshold is reached the plasma shape control system starts to reduce the plasma current and its elongation in order to limit the possible effects of the imminent disruption [43].

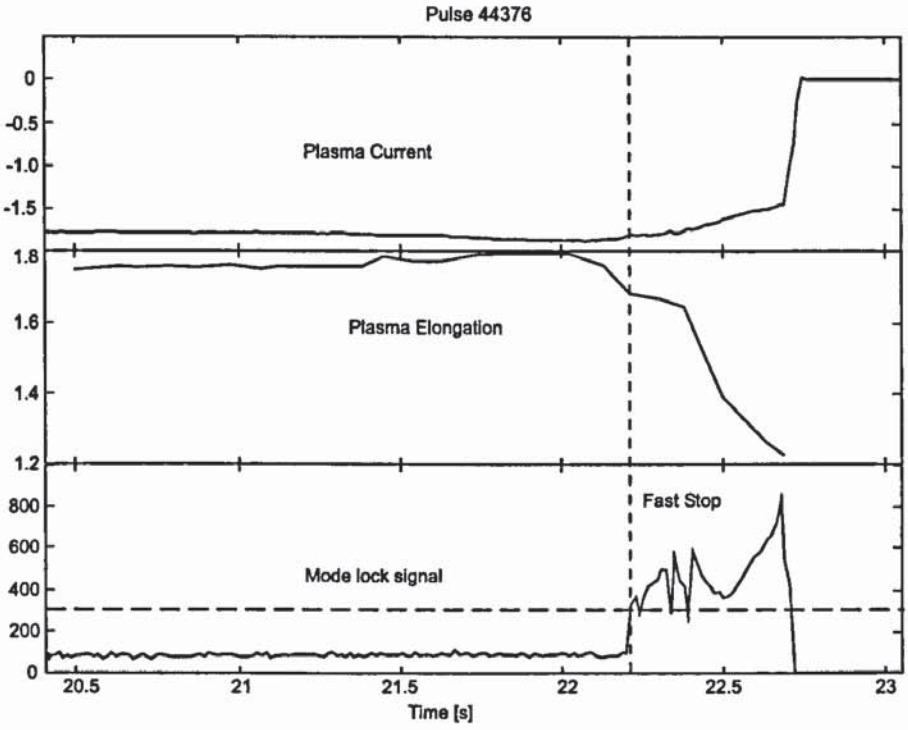

Figure 5.2: The fast stop scenario changes plasma configuration drastically. The plasma elongation is immediately reduced together with the plasma current. The mode lock signal plotted is divided by the plasma current, and the horizontal dotted line represents the threshold that activates the fast stop scenario.

This method gives good results and in some cases it is possible to avoid the disruption. For most of the pulses, the control systems do not have sufficient time to terminate the pulse, it

only considerably reduces the plasma current, and as a consequence the vessel forces, before the disruption occurs.

Two ways of using the mode lock signal, have been considered. The first method uses it to increase the number of input patterns for the training set. Patterns at the time when the mode lock signal reaches the limit value have been selected and included in the disrupted class. In this case the mode lock signal has not been selected to be an input signal of the network to verify that the network can use the disruption precursors patterns present on the other inputs to predict the occurrence of a disruption. The other method uses the mode-lock signal as an input of the network, but it considers as input also patterns taken 100 ms before the disruption. In this way the class of disrupting patterns will not have as input data the predominance of mode lock signals values above a certain limits, but a more balanced set even if the risk that 'stable' plasma patterns are included in the wrong class is higher.

5.3.3 Plasma radiated/input power ratio

Experimental studies showed that when the ratio between the radiated power of the plasma and the power delivered by the heating systems reaches a certain limit there is a high probability of disruption. Studies performed at JET shows that the limit is around 0.65 for x-point steady state plasma [44].

It is still not clear why this happen. The general idea is that when the plasma starts to radiate energy from the edge, it suddenly contracts to keep the balance between the magnetic field and the plasma forces. If the shrinking of the plasma surface is very high, the q=2 could reach the plasma edge with a subsequent plasma disruption.

An important role in this calculation is played by the plasma stored energy. Sudden variation of the input power are compensated by a drop of plasma stored energy. Thus, the ratio to be considered becomes:

$$\frac{RadiatedPower}{\left(InputPower-\frac{\partial PlasmaStoredEnergy}{\partial t}\right)} \tag{5.11}$$

The radiated power is measured with an array of bolometer detectors [34], [45]. Each detector has a different line of sight and combining the amount of plasma radiation that they detect it is possible to reconstruct the full plasma radiated power.

The input power is a sum of two components: the ohmic input power produced by PF coil currents (mainly by the central solenoid current), and the additional heating power, delivered by the neutral beam injection system, the ion cyclotron radio frequency and the lower hybrid systems.

The stored energy is calculated using the magnetic measurements and the diamagnetic loop, a full description of the calculations of the process can be found in [58].

The fact that these measurements were not available in real time until the last campaign prevented the implementation of a real-time algorithm that could be used to stop unstable plasmas. For example at the end of a neutral beam heating phase the sudden drop of the total input power is not followed by the same drop on the radiated power and temporarily the ratio on equation (5.11) can become close to, or even higher than 1.0.

Nevertheless, this method could be used for the design of the disruption prediction network, providing useful information about radiative disruptions to the network.

The input power, the radiated power and the derivative of the plasma stored energy have been included in the input of the network in two different ways. One network has been trained using as input the radiated/input power ratio, on the other the three signals have been fed directly to the network.

In addition to these signals the plasma density signal and the poloidal beta have been added. The plasma density signal is again related to the radiative disruptions. There is a limit on the density, proportional to the input power, that limits the operational space. When the plasma density breaches this limit it is very probable that the plasma will disrupt due to the cooling of the plasma and the incapability of the external heating systems to maintain its temperature.

The poloidal beta has been chosen because it can be helpful to characterise the plasma configuration more than because of its direct relation with disruption precursors.

Summarising different networks have been trained using different inputs. The inputs of the three network are shown in the table:

Input Signals	Neural Network 1	Neural Network 2	Neural Network 3
Mode lock		✓	✓
Plasma Density	✓	✓	✓
Total Input Power	✓	✓	
Total Radiated Power	✓	✓	
Stored Energy Derivative	✓	✓	
Plasma Internal Inductance ℓ_i	✓	✓	✓
Safety Factor q_{95}	✓	✓	✓
Poloidal beta β_p	✓	✓	✓
Radiated/Input Power ratio			✓

Table 5.1: Input variables of the three different network

The method to select the input patterns for the network is also different. At the beginning of this work the number of disruptions available was relatively small having only 360 disruptions. If only one pattern per disruptions had to be considered the size of the network would have been excessively limited. To increase the size of the datasets and at the same time to introduce the knowledge about disruption precursors gained during the experimental years it has been decided to select, as patterns of the disrupted class, the data taken at the time when the standard prediction methods would have generate an alarm.

After the first set of tests it was clear that the radiated/power ratio limit was difficult to define without introducing patterns of 'stable' plasmas in the wrong class.

Therefore the first network has been trained using a dataset where the disrupted class was built with patterns taken at the disruption time and patterns taken at the time when the lock mode signals breached the limit set during operation. For the other two networks the disrupted class was built with patterns taken at the disruption time plus patters taken 100 ms before the occurrence of the disruption. In this case patterns of stable plasmas could be included in the disrupting plasmas class affecting the overall network results.

5.4 Network Training

The networks have been trained using the same method used for the ℓ_i-q network. The optimisation method is the gradient descent with adaptive learning rate. To determine the gradient of the error with respect to the weights the back-propagation method has been used. It is easy to show that also in this case the determination of the error gradient is reduced to a simple formula where the two components of equation (1.16) can be written as:

$$\frac{\partial \tilde{a}_k}{\partial w_{kj}^{(2)}} = z_j \qquad \frac{\partial E^n}{\partial \tilde{a}_k} = \sum_{k'} \frac{\partial E^n}{\partial y_{k'}} \frac{\partial y_{k'}}{\partial \tilde{a}_k} \tag{5.12}$$

The derivative on the right side of the second equation can be found:

$$\frac{\partial E^n}{\partial y_{k'}} = -\frac{t_{k'}}{y_{k'}} \quad \text{and} \quad \frac{\partial y_{k'}}{\partial \tilde{a}_k} = y_{k'}\delta_{kk'} - y_{k'}y_k \tag{5.13}$$

with $\delta_{kk'} = 0$ for $k \neq k'$ and $\delta_{kk'} = 1$ for $k = k'$. Substituting equations (5.13) into (5.12) we can write:

$$\frac{\partial E^n}{\partial \tilde{a}_k} = y_k - t_k \tag{5.14}$$

This is the same result obtained for networks with linear activation function for the output layer (see equation (1.18)). This eases the implementation of the software needed for the training of the network. The same structure can be kept, the only changes needed are just the

insertion of the sigmoid as output activation function and the calculation of the cross entropy error.

The training has been performed using two different network complexity control schemes.

The first method used is the early stopping method. The method consists in stopping the training process when the error measured with respect to the validation dataset is at its minimum. During the training process, the error with respect to the training data generally decreases as a function of the iteration index. However, if the error is calculated with respect to an independent set of data, it will decrease at first, followed by an increase as the network starts to over-fit. A simple algorithm that determines this error and stops when it is at its minimum has been implemented. The training is stopped when for 30 consecutive iteration the validation dataset error does not decrease. This reduces the time spent on the training of the network giving the possibility of training the network regularly including the new data generated during the last operations period

The second method called the regularisation method [7] encourages smoother network mappings by adding a penalty to the error function avoiding over-fitting to the noise of the training data. The penalty function chosen, called weight decay adds to the error the square of the network weights sum:

$$E = -\sum_{n}^{N}\sum_{k'} t_{k'}^{n}\log(y_{k'}^{n}) + \nu\frac{1}{2}\|\mathbf{w}\|^2 \tag{5.15}$$

Heuristically, it is possible to imagine that to produce an over-fitted mapping with large curvature there is the need of relatively large values for the network weights. The addition of the weight decay term will penalise networks with large weights in favour of networks where the weights value are small. This will avoid that the training process will find a solution where the training dataset is perfectly matched, to the detriment of the generalisation capability of the network. Smaller weights mean that the sigmoidal activation functions will work in the input space where its output could be approximated as linear. Therefore, also the output of the network should be "smoother" and the network generalisation should improve.

The parameter ν called the regularisation coefficient determines the smoothness of the network. In this case ν has been chosen empirically by training several sets of networks varying the regularisation coefficient from 0.001 to 5.

Comparisons between the various networks using the validation set to select the best performing network can be seen on Table 5.2

Regularisation coefficient ν	Hidden units	Network 1 test set error	Hidden units	Network 2 test set error	Hidden units	Network 3 test set error
0.01	12	0.261	6	0.278	4	0.336
0.05	9	0.268	5	0.28	3	0.399
0.1	5	0.254	7	0.273	5	0.299
0.5	7	0.236	8	0.268	6	0.298
1.0	6	0.213	9	0.255	4	0.278
5.0	10	0.257	8	0.301	3	0.301
Early stopping	14	0.218	4	0.278	6	0.313

Table 5.2: Best performing networks obtained using different regularisation factors. The error was calculated using the cross entropy formula.

The comparison shows that the overall best performing network is network 1, with 6 hidden units. This network gives also better results than the best performing network determined using the early stopping method. After the training, it was also possible to note that the regularisation term did not influenced the final solution, even with large values of the regularisation coefficient the error (5.15) was still dominated by the first term.

5.5 Decision boundary

To minimise the probability of misclassification the graphs below have been plotted. Using the test dataset, the two curves have been plotted varying the decision boundary from 0 to 1 and plotting the percentage of misclassified patterns. The probability of misclassification is

minimised by placing the decision boundary where the curves cross. However we can see that the number of misclassified patterns is still around 10%.

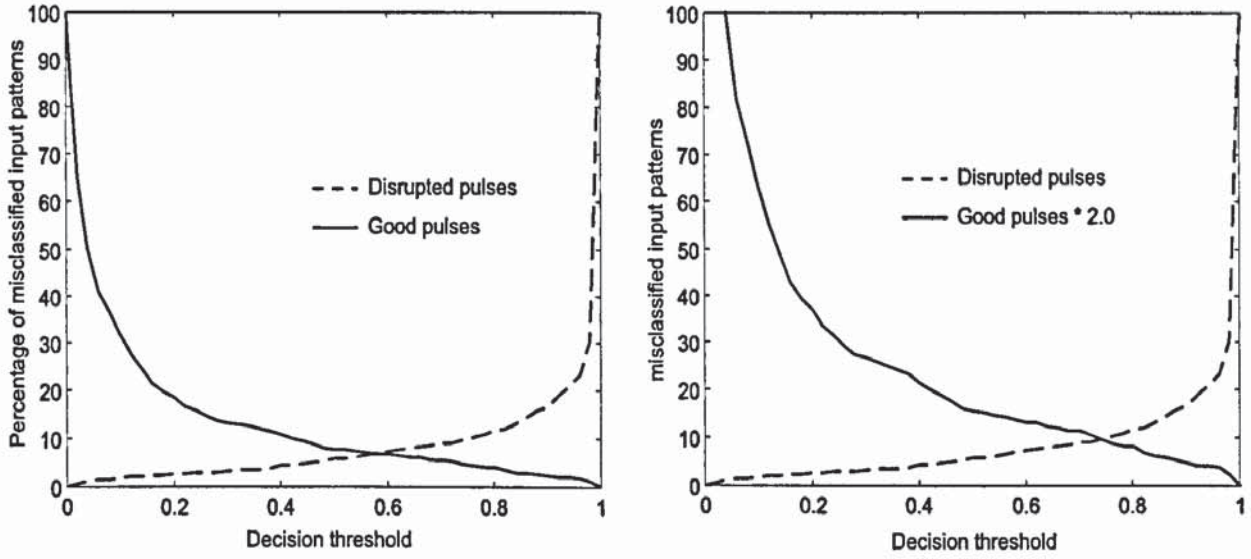

Figure 5.3: The left graph shows the percentage of misclassified patterns at the variation of the decision threshold. On the right, the curve of the misclassified good pulses has been multiplied by a weight factor of 2.0. This is a way to select a decision threshold that minimises the risk of stopping good pulses.

The percentage of good pulses that would be stopped by the network is still too high for JET, where stopping a good pulse is considered more severe than not detecting a disruption. To take into account this general feeling, the misclassification of the good pulses has been penalised multiplying the results by a weight factor of 2.0 (see Figure 5.3 right).

This is equivalent to the determination of a cost matrix where the term related to the missclassification of good pulses has been doubled.

An opposite approach would be used in a fusion reactor machine where it would be better to stop the pulse when there is a minimal risk of disruption than to risk damaging the machine structure.

5.6 Results

As already mentioned, even with the best performing network the risk of stopping good pulses is relatively high. To better understand the behaviour of the network, a set of pulses (not used to design the network) has been used to test it, to see where the misclassification occurred. The pulses that best depict the network performance have been reported here.

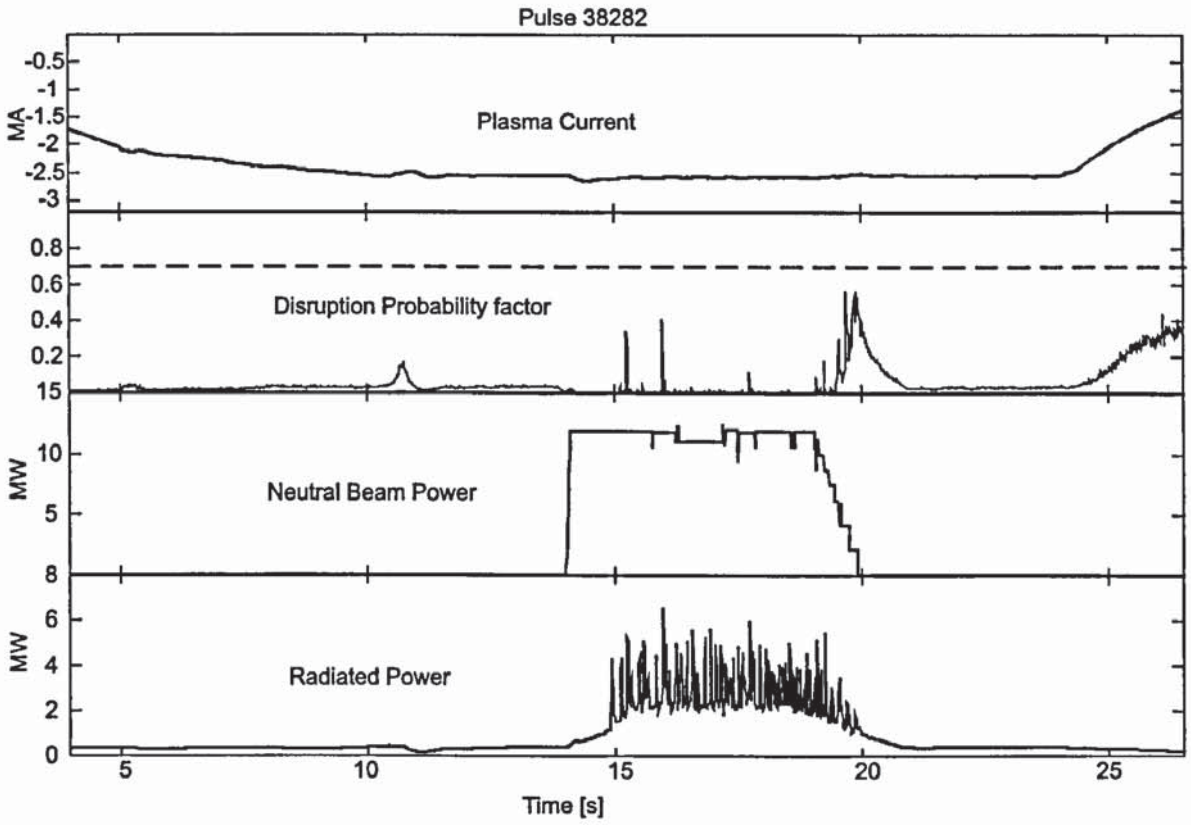

Figure 5.4: Behaviour of the network for a good pulse. The second trace from the top shows the behaviour of the probability of the input patterns to belong to the unstable plasma class. The most dangerous period of the whole pulse is during the neutral beam heating switching off (see third trace). The fourth trace shows that ELMs cause some of the spikes, present in the disruption probability. Edge Localised Modes are bursts of instabilities causing reduction in density and in temperature in the outer zone of the plasma.

On Figure 5.4 the behaviour of the network has been plotted for a successful pulse. It can be noticed that in this case the probability of disrupting stays well below the threshold for the whole length of the pulse. Only during the ramp down of the additional heating power is probability increasing. This period is commonly considered a difficult period for the plasma.

The radiated power and the density have a slow rate of decrease and it is easy to reach the density limit causing a disruption.

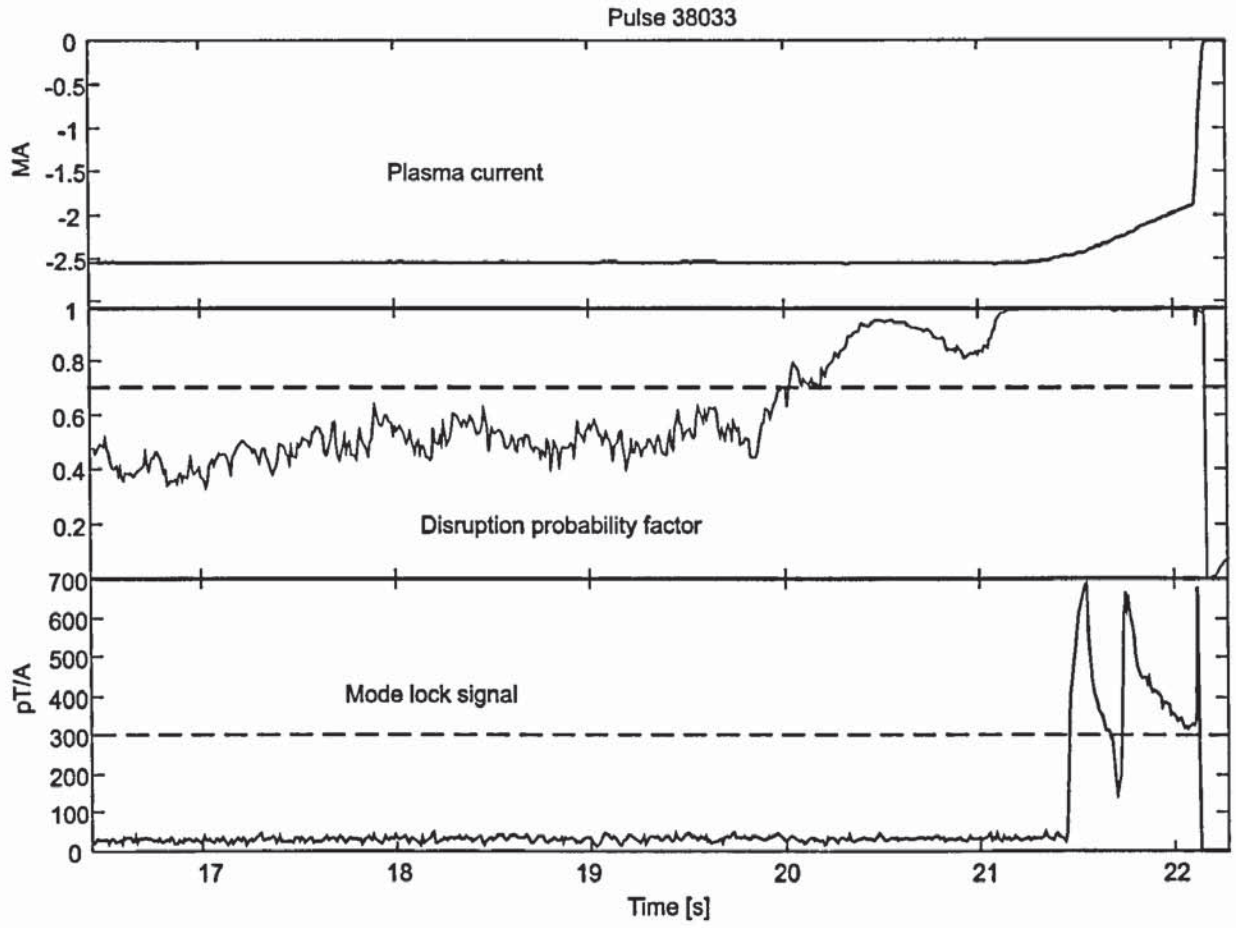

Figure 5.5: This pulse shows how it is possible for the neural network to predict plasma disruptions. The disruption probability has breached the limit well before the appearance of the locked mode.

Figure 5.5 shows how the network is able to detect plasma disruptions giving operational time to the control systems to shut down the pulse safely. The prediction of the imminent disruption occurs well before the growth of the mode lock signal which is currently used.

It can be seen how the control system started the plasma termination procedure as soon as the mode lock limit was breached. The plasma current starts to ramp down reaching a value around 1.8 MA before the disruption occurs. If the interval was longer the plasma current would have been reduced even more.

Unfortunately some time the network provides results that can be considered dubious. In Figure 5.6 a pulse considered successful is shown.

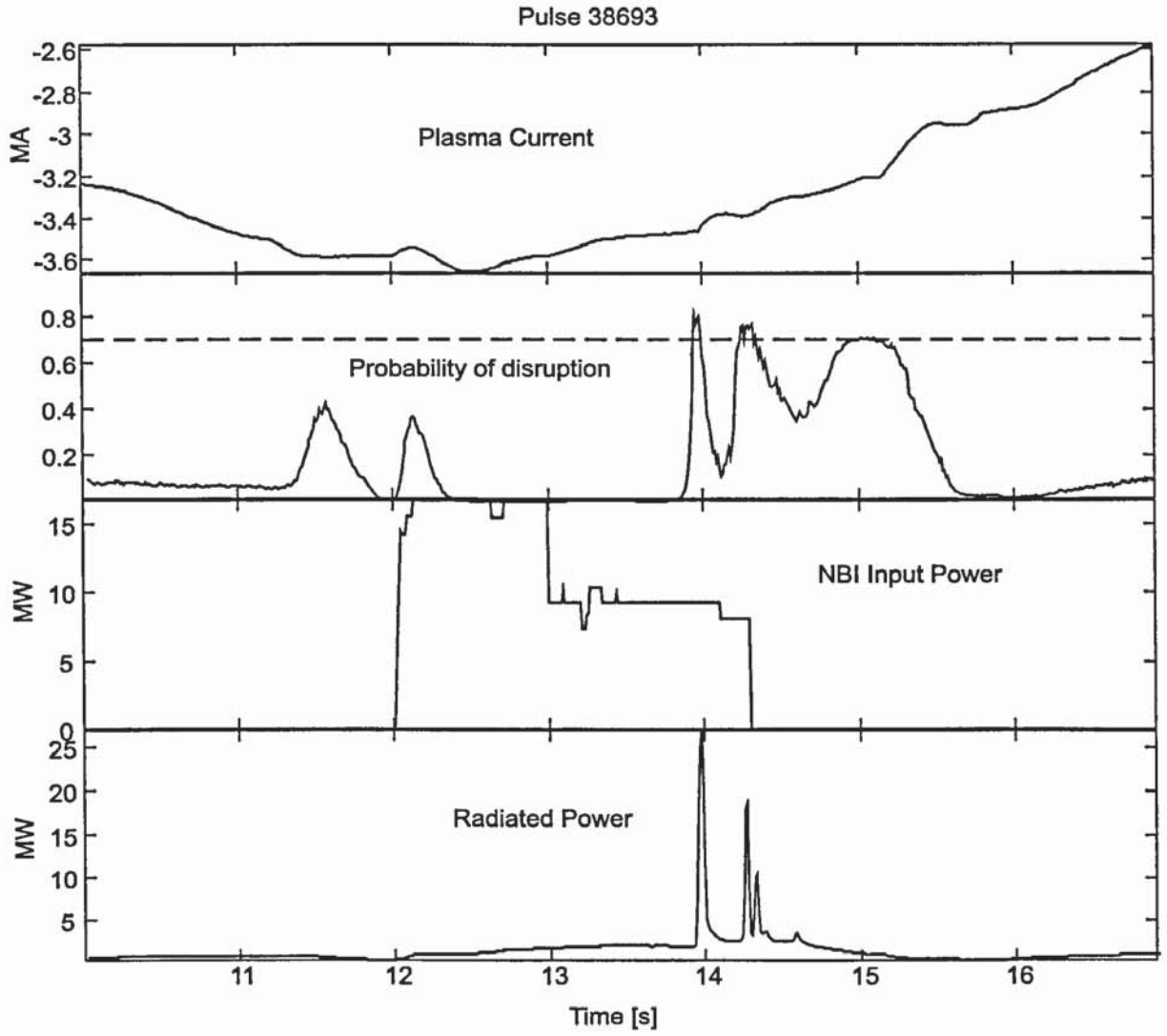

Figure 5.6: The plasma survived at the perturbation but the network detected a high probability of disruption. The reason has to be attributed to the huge burst of radiated power caused by the ELM and then to the switching off of the additional heating systems. These are the pulses that could cause doubt abut the results if stopped by the network.

In this case the network would have stopped the pulse just after a giant ELM (Edge Localised Mode). These phenomena cause a sudden drop in the plasma stored energy in the form of radiated power and loss of plasma density. If the energy loss is very high a disruption can occur. It is therefore acceptable that the network predicts a high probability of disruption at the occurrence of giant ELMs. We cannot answer the question if the plasma would have survived, if a procedure of terminating the pulse was issued by disruption prediction and prevention system.

A different issue is when the disruption is not detected or it is predicted too late to be handled by the control systems.

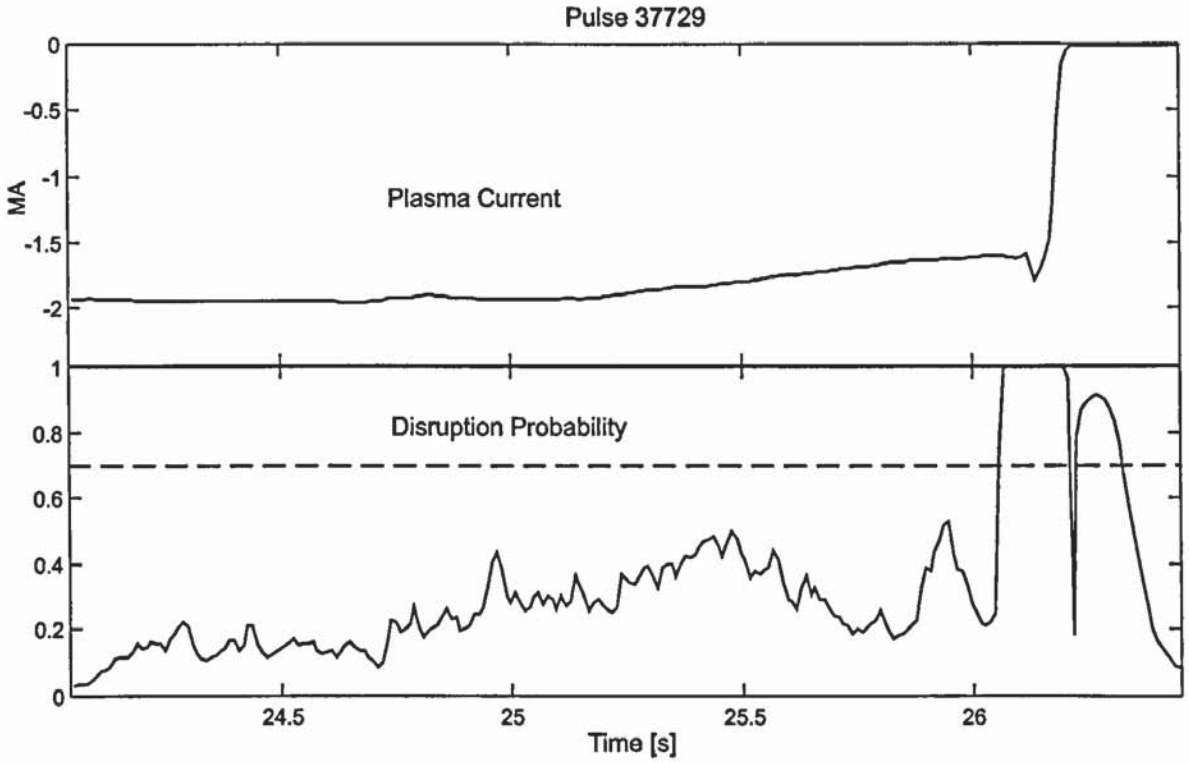

Figure 5.7: This is one of the cases where the detection of the disruptions occurred too late to allow any action to the control systems. It is not a case of misclassification. However, it suggests that some disruptions are so fast and happen without any precursors that even neural network algorithms can do nothing to predict them.

There could be many reasons for this behaviour of the network. Some disruptions have very short precursor phase that the disruption can not be detected more than a few milliseconds in advance. We also should not forget that if the plasma configuration is novel it could be difficult for the neural network to recognise it. This could lead to a late detection of the disruption.

5.7 Conclusions

In this chapter, the classification capability of neural networks has been used to predict plasma disruptions. The work involved the study of the precursors used at JET and the generation of a disruption database. Due to the lack of a sufficiently large database of disruptions, it was preferred not to subdivide the disruptions by type, but to consider them as a whole class of unstable plasma patterns. During the period of network design, the database was increased and in the future it will be possible to classify the disruptions more precisely. Then, the network will provide more information about the nature of the disruptions so that the control systems can act accordingly. This is why the network architecture chosen is a two layer network with the softmax activation function for the output layer. The network architecture together with the software developed to train it gives the possibility of an easy upgrade. New classes and new inputs can be introduced without any major modification.

The results, with the network designed for this thesis, are promising and shows that these methods could provide a form of protection from disruptions. A 100% success cannot be guaranteed, but at the same time these methods could help to at least reduce the number of disruptions. The neural network designed can be seen as way to delimit the machine operational space using the knowledge acquired during the previous experimental campaigns. It is not possible for this algorithm to detect all disruptions, but at least it could help to avoid repeating the same plasma configurations that lead to a disruption in the past.

6. Disruption prediction using novelty detection techniques

6.1 Introduction

The prediction of plasma disruption using a MLP network is done using a dataset of disrupted pulses. The likelihood that novel pattern will be wrongly classified as disruptive is therefore reduced if we use a cautious approach to stop just the pulses with very high disruption probability values. A drawback of this method is that to train the network a big disruption database is needed. This is not usually available especially if we consider the extension of this work to new bigger machines. In fusion reactor-like machines as ITER it will be impossible to generate a disruption database. The forces exerted by a disruption at full plasma current (21 MA) could cause serious damage to the machine structure involving months of shutdown to repair any possible damage. We therefore need to study methods of recognising disrupting plasma without using a disruption database.

Novelty detection methods are very appealing since they provide the possibility to work without a disruption database, using only information of good plasma configurations. This dataset is easier to obtain because during the early phase of the machine operation most of the plasma configuration will be tested and the data produced can be used to train a novelty detection network. It has to be noted that a single pulse can provide several seconds of data that can be flagged as successful.

Novelty detection methods are used to determine abnormal conditions in the input dataset. Applications can be found in the medical field or even in the electromagnetic field to detect fault conditions where direct measurements are not possible or too expensive. They can also

provide information about the novelty of the input of neural networks. They provide an indirect measurement of how much the input data differs from data of the training set. If the difference is very large the results could be inaccurate. Therefore they could be discarded or at least flagged as suspicious.

Major concern on the implementation of novelty detection algorithms for disruption detection at JET is the fact that new plasma configurations are always tested. Novelty detection algorithms could classify as disrupting not only 'unstable' plasma configurations but also new successful plasma configurations.

6.2 Novelty Detection Methods

The problem of determining the novelty of an input pattern can be seen as the problem of determining a probability density function $p(\mathbf{x})$, given a finite number of data points $\mathbf{X}=(\mathbf{x}^1,\ldots,\mathbf{x}^N)$.

The approach chosen here to probability density estimation is often called semi-parametric method. It is based on modelling the density using a class of functional forms in which the number of adaptive parameters can be chosen to build any complex model. At the same time the total number of parameters is not strictly dependent on the number of data points [6]. We therefore write our model for the density distribution as a linear combination of a class of functions. If the *mixture distribution* representation is used the density can be written as:

$$p(\mathbf{x}) = \sum_{j=1}^{M} p(\mathbf{x} \mid j) P(j) \tag{6.1}$$

The component density functions $p(\mathbf{x}|j)$ are normalised so that:

$$\int p(\mathbf{x} \mid j) d\mathbf{x} = 1 \tag{6.2}$$

and the coefficients $P(j)$ can be seen as the prior probability of the data point having been generated from component j of the mixture. These coefficients are chosen to satisfy the constraints

$$\sum_{j=1}^{M} P(j) = 1$$
$$0 \le P(j) \le 1 \tag{6.3}$$

In this work the Gaussian distribution function has been chosen as component density functions:

$$p(\mathbf{x} \mid j) = \frac{1}{(2\pi\sigma_j^2)^{d/2}} \exp\left\{-\frac{\|x-\mu_j\|^2}{2\sigma_j^2}\right\} \tag{6.4}$$

where d in the dimension of the vector $\mathbf{x}$.

The parameters σ and μ together with the $P(j)$ are determined from the set of data using the maximum likelihood technique.

Maximum likelihood technique seeks to find the optimum values for the parameters by maximising a likelihood function derived from the training data. In this case the likelihood function can be defined as the joint probability density of the whole training data set:

$$\mathcal{L}(\sigma_j, \mu_j, P(j)) = \prod_{n=1}^{N} p(\mathbf{x}^n) \tag{6.5}$$

$\mathcal{L}(\bullet)$ can be viewed as a function of the parameters $\theta = \{\sigma_j, \mu_j, P(j)\}$ for fixed training data set $\mathbf{x}^1,\ldots,\mathbf{x}^N$. The technique of maximum likelihood then sets the value of the parameters by maximising $\mathcal{L}(\cdot)$.

In practice, it is more convenient to consider the negative logarithm of the likelihood function and use algorithms to minimise it:

$$E = -\ln(\mathcal{L}) = -\sum_{n=1}^{N} \ln(p(\mathbf{x}^n)) = -\sum_{n=1}^{N} \ln\left(\sum_{j=1}^{M} p(\mathbf{x}^n \mid j) P(j)\right) \tag{6.6}$$

E can be regarded as an error function.

To minimise this error function, the *expectation maximisation* algorithm has been used.

6.3 Expectation maximisation algorithm [14],[46],[6]

The EM algorithm is a device to help find maximum likelihood estimators in a problem with unobserved data. Suppose we have data **X** which have been observed and data **Y**, which have not, and a vector of parameters θ. The goal is to find the maximum likelihood estimator of θ given the observed data **X** in a situation in which the joint density p(x,y;θ) is known, but the density of **X** p(x;θ), can only be found by numerical summation from the joint density.

This algorithm provides a simple iterative method to find the density function parameters θ. The EM algorithm starts at some value θ^{old} and alternates two steps. The expectation step determines the conditional expectation of the ln(p(**X**,**Y**;θ)) over the values of **Y** an is evaluated as if the values θ^{old} are the correct ones. This gives [46]:

$$Q(\theta,\theta^{old}) = \mathcal{E}\left[\ln p(\mathbf{X},\mathbf{Y};\theta) \,|\, \mathbf{X};\theta^{old}\right] \tag{6.7}$$

The maximisation step determines the parameters θ^{new} to maximise $Q(\theta,\theta^{old})$.

In our case the parameters of the mixture model to be found are $\theta = \{\sigma_j, \mu_j, P(j)\}$. $Q(\theta,\theta^{old})$ can be determined introducing an hypothetical complete dataset in which each data point is labelled with the component j which generated it [6]. For each data point $\mathbf{x}^n$ we introduce a variable $\mathbf{y}^n$, which is an integer in the range (1,M), specifying which component of the mixture generated the data point. ln(p(**X**,**Y**;θ)) then becomes:

$$\begin{aligned} \ln(p(\mathbf{X},\mathbf{Y})) &= \sum_{n=1}^{N} \ln\left\{P^{new}(\mathbf{y}^n)\, p^{new}(\mathbf{x}^n \,|\, \mathbf{y}^n)\right\} \\ &= \sum_{n=1}^{N}\sum_{j=1}^{M} \delta_{j\mathbf{y}^n} \ln\left\{P^{new}(\mathbf{y}^n)\, p^{new}(\mathbf{x}^n \,|\, \mathbf{y}^n)\right\} \end{aligned} \tag{6.8}$$

with $\delta_{j\mathbf{y}^n}$ the Kroneker delta symbol, equal to one when $j = \mathbf{y}^n$ and zero otherwise.

Using the conditional probability of $\mathbf{y}^n$ given $\mathbf{x}^n$ calculated with the old parameters, θ^{old}. Then the conditional expectation $Q(\theta,\theta^{old})$ as:

$$Q(\theta,\theta^{old}) = \sum_{\mathbf{y}^1=1}^{M} \cdots \sum_{\mathbf{y}^N=1}^{M} \ln(p(\mathbf{X},\mathbf{Y})) \prod_{n=1}^{N} P^{old}\left(\mathbf{y}^n \,|\, \mathbf{x}^n\right) \tag{6.9}$$

and making use of the identity:

$$\sum_{\mathbf{y}^1=1}^{M}\cdots\sum_{\mathbf{y}^N=1}^{M}\delta_{j\mathbf{y}^n}\prod_{n'=1}^{N}P^{old}\left(\mathbf{y}^{n'}\,|\,\mathbf{x}^{n'}\right)=P^{old}(j\,|\,\mathbf{x}^n) \tag{6.10}$$

we obtain the expectation in the form:

$$Q(\theta,\theta^{old})=\sum_{n=1}^{N}\sum_{j=1}^{M}P^{old}(j\,|\,\mathbf{x}^n)\ln\left\{P^{new}(j)p^{new}(\mathbf{x}^n\,|\,j)\right\} \tag{6.11}$$

It is possible to demonstrate that the maximisation of this equation corresponds to a decrease of the negative logarithm of the likelihood defined in equation (6.6). If we write the change in error when we replace the old parameters with the new parameters we obtain:

$$E^{new}-E^{old}=-\sum_{n=1}^{N}\ln\left\{\frac{p^{new}(\mathbf{x}^n)}{p^{old}(\mathbf{x}^n)}\right\} \tag{6.12}$$

Using the definition of the mixture distribution given by (6.1) we can write:

$$E^{new}-E^{old}=-\sum_{n=1}^{N}\ln\left\{\frac{\sum_{j=1}^{M}P^{new}(j)p^{new}(\mathbf{x}^n\,|\,j)}{p^{old}(\mathbf{x}^n)}\frac{P^{old}(j\,|\,\mathbf{x}^n)}{P^{old}(j\,|\,\mathbf{x}^n)}\right\} \tag{6.13}$$

Making use of the Jensen's inequality which says that given a set of numbers $\lambda_j \geq 0$ such that $\sum_j \lambda_j = 1$, then $\ln(\sum_j \lambda_j x_j) \geq \sum_j \lambda_j \ln(x_j)$.

Since the $P^{old}(j\,|\,\mathbf{x}^n)$ in the numerator of (6.13) sum to unity they can be considered as the λ_j:

$$E^{new}-E^{old}\leq-\sum_{n=1}^{N}\sum_{j=1}^{M}P^{old}(j\,|\,\mathbf{x}^n)\ln\left\{\frac{P^{new}(j)p^{new}(\mathbf{x}^n\,|\,j)}{p^{old}(\mathbf{x}^n)P^{old}(j\,|\,\mathbf{x}^n)}\right\} \tag{6.14}$$

Since we wish to minimise E^{new} with respect to the new parameters, rearranging (6.14) we obtain an upper bound for E^{new} made of two components:

$$E^{new} \leq E^{old} + \tilde{Q} \tag{6.15}$$

We can therefore minimise the upper bound with respect to the 'new' parameters to decrease the value of E^{new}. This means minimising the terms of $\tilde{Q}$ that depend on the new parameters:

$$\tilde{Q} = -Q(\theta, \theta^{old}) = -\sum_{n=1}^{N}\sum_{j=1}^{M} P^{old}(j \mid \mathbf{x}^n) \ln\left\{P^{new}(j) p^{new}(\mathbf{x}^n \mid j)\right\} \tag{6.16}$$

It is possible to notice that the equation to minimise is equal to equation (6.11) (a part from the sign) which define the expectation $Q(\theta, \theta^{old})$, proving that for each EM steps if we maximise equation (6.11) the negative log-likelihood will decrease.

Equation (6.11) can be maximised easily for the case of a Gaussian mixture model. The equation becomes:

$$Q(\theta, \theta^{old}) = \sum_{n=1}^{N}\sum_{j=1}^{M} P^{old}(j \mid \mathbf{x}^n)\left\{\ln P^{new}(j) - d\ln(\sigma_j^{new}) - \frac{\left\|\mathbf{x}^n - \mu_j^{new}\right\|^2}{2(\sigma_j^{new})^2}\right\} \tag{6.17}$$

This function can be minimised with respect to the 'new' parameters. The maximisation of μ_j and of σ_j can be done imposing the derivative to zero:

$$\frac{\partial Q(\theta, \theta^{old})}{\partial \mu_j^{new}} = -\sum_{n=1}^{N} P^{old}(j \mid \mathbf{x}^n) \frac{(\mathbf{x}^n - \mu_j^{new})}{(\sigma_j^{new})^2} = 0 \tag{6.18}$$

Thus:

$$\mu_j^{new} = \frac{\sum_n P^{old}(j|\mathbf{x}^n)\mathbf{x}^n}{\sum_n P^{old}(j|\mathbf{x}^n)} \tag{6.19}$$

Similarly for the σ_j:

$$\frac{\partial Q(\theta, \theta^{old})}{\partial \sigma_j^{new}} = -\sum_{n=1}^{N} P^{old}(j \mid \mathbf{x}^n)\left\{-d\frac{1}{\sigma_j^{new}} + \frac{\left\|\mathbf{x}^n - \mu_j^{new}\right\|^2}{(\sigma_j^{new})^3}\right\} = 0 \tag{6.20}$$

$$\left(\sigma_j^{new}\right)^2 = \frac{1}{d}\frac{\sum_n P^{old}(j|\mathbf{x}^n)\left\|\mathbf{x}^n - \mu_j^{new}\right\|^2}{\sum_n P^{old}(j|\mathbf{x}^n)} \tag{6.21}$$

For the mixing parameters, the constraint $\sum_j P^{new}(j) = 1$ has to be taken into account. This can be done introducing the Lagrange multiplier λ and minimising the function:

$$Q(\theta,\theta^{old}) + \lambda\left(\sum_{j=1}^{M} P^{new}(j) - 1\right) \tag{6.22}$$

Setting the derivatives of (6.17) with respect to $P^{new}(j)$ to zero, we obtain:

$$\frac{\partial Q(\theta,\theta^{old})}{\partial P^{new}(j)} = -\sum_{n=1}^{N}\frac{P^{old}(j\,|\,\mathbf{x}^n)}{P^{new}(j)} + \lambda = 0 \tag{6.23}$$

If we then multiply the equation by $P^{new}(j)$ and sum over j, we obtain the equation:

$$\sum_{n=1}^{N}\sum_{j=1}^{M} P^{old}(j|\mathbf{x}^n) = \lambda\sum_{j=1}^{M} P^{new}(j) \tag{6.24}$$

Using $\sum_j P^{new}(j) = 1$ and $\sum_j P^{old}(j|\mathbf{x}^n) = 1$ we obtain λ=N, that substituted into (6.23) gives:

$$P(j)^{new} = \frac{1}{N}\sum_n P^{old}(j|\mathbf{x}^n) \tag{6.25}$$

The method is very simple to implement, the only problem is that during the training the Gaussian distribution function centre could coincide with one of the input patterns. From equation (6.21) it is possible to see that this will force σ_j to zero. The Gaussian distribution (6.4) then goes to infinity. To avoid this, several techniques can be implemented, as an example the standard deviation σ_j can be set to a fixed value during all the training process. In this work a different approach has been used: new values chosen randomly are reassigned to the μ_j that collides with a data point. The optimisation process then starts from a configuration partially optimised. The training therefore is stopped when the difference between the error of step j and the error at step j-1 is less than 1% of the total error.

6.4 Dataset generation

To have a comparison with the MLP softmax network results it has been decided to use the same dataset of good pulses used previously. After this comparison, a dataset with the same input variables but made with 1400 input patterns has been generated to estimate the density distribution.

The first training set is then made by 350 patterns, and the input variables are:

1. Plasma Density
2. Total Input Power
3. Total Radiated Power
4. Stored Energy Derivative
5. Plasma Internal Inductance ℓ_i
6. Safety Factor q_{95}
7. Beta poloidal

After having proved that the method is applicable and is giving promising results, the new dataset has been generated introducing also the mode-lock signal. This can be done because this new dataset is created without making any discrimination on this signal. The patters were selected randomly from 140 successful pulses. A minimum plasma current of 1 MA has been imposed for the selection to avoid problems of reliability of XLOC at low plasma current. The final dataset is made of 1400 patterns

A disruption dataset has been created as well just to test the network. This is the same of that one used for the softmax network.

6.5 Input normalisation

Two different methods have been used to normalise the data. The first normalisation is the same as used in chapter 5 and 3. The re-scaling is performed with respect to the training set and each of the input variables is treated independently.

This normalisation has been compared with a more complex method called whitening method. This method allows also for correlation amongst the variables. The transformation forces the input data to have zero mean and the unit matrix as covariance matrix.

The mean and the covariance matrix are respectively determined as:

$$\begin{aligned} \bar{\mathbf{x}} &= \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}^n \\ \Sigma &= \frac{1}{N-1}\sum_{n=1}^{N}\left(\mathbf{x}^n-\bar{\mathbf{x}}\right)\left(\mathbf{x}^n-\bar{\mathbf{x}}\right)^T \end{aligned} \tag{6.26}$$

Defining the diagonal matrix Λ as the diagonal matrix of the covariance matrix eigenvalues, and the $\mathbf{U}$ matrix as the matrix of its eigenvectors, it is possible to calculate the linearly transformed input variables as:

$$\tilde{\mathbf{x}}^n = \Lambda^{-1/2}\mathbf{U}^T(\mathbf{x}^n-\bar{\mathbf{x}}) \tag{6.27}$$

The whitening method does increase the complexity of the calculation adding a pre-processing stage to the computation. The mean vector has to be subtracted from the input data. Then they have to be multiplied by a matrix *dxd*. However, this normalisation eases the training process and the network results using the whitening method shows a considerable improvement if compared with the linear re-scaling.

6.6 Density distribution estimation

The training has been performed several times varying the number of Gaussian distributions. The best performing network has been selected using the hold out method [6]. For each network, the error function has been calculated using a validation set. The results have been compared and the network producing the smallest error has been selected:

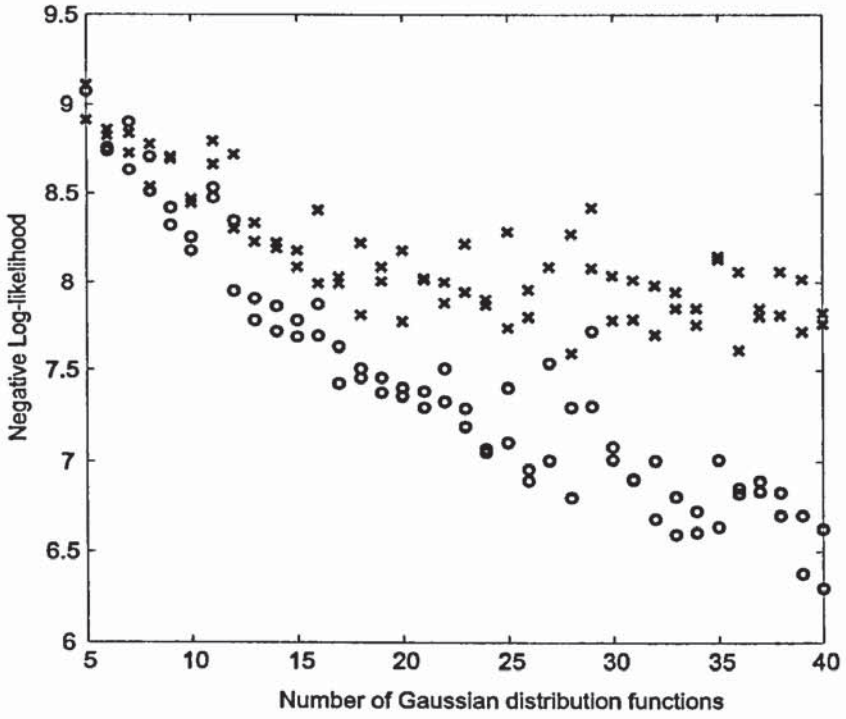

Figure 6.1: Training of the network using the same datasets used to train the softmax network type 1. The-lock signal has not been used and just 350 patterns of successful pulses were used to train the network. The circles represent the error determined using the training set and the x the training set.

The minimum error has been obtained using 28 Gaussian distributions.

To verify the results the negative log likelihood has been plot for the validation set and a test set made with the disruption dataset used to design the softmax network. Clearly there is a certain separation between the two sets giving some confidence on the possibility of predicting disruptions using this method.

On Figure 6.2 it is also possible to compare the results obtained with data normalised using the linear rescaling method against the data normalised using the whitening method. It is evident that the network trained using the second method present a better separation of the two datasets.

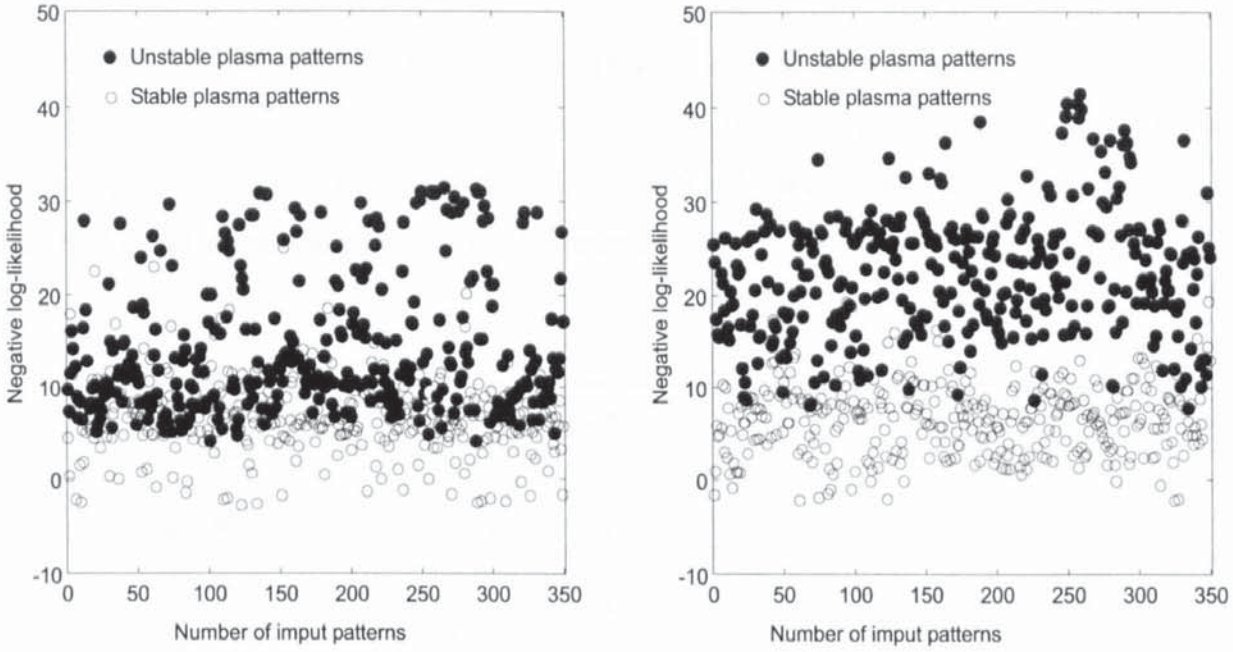

Figure 6.2: In this plot it is possible to see the separation between 'unstable' or disrupting plasma patterns and 'stable' or good plasma patterns. The drawing on the left shows the best results obtained normalising the data using the linear re-scaling method. The right drawing shows the results when the data normalisation has been done using the whitening process.

6.7 Decision boundary selection

To decide where to set the boundary that discriminates the novel data from the good data two different approaches can be used.

If a disruption database is not available, we have to rely just on the information of the successful pulses. We can therefore draw a graph (Figure 6.3) showing the percentage of patterns correctly classified varying the decision threshold value. The plot can then be used to select a decision boundary that minimises the misclassfication of the successful pulses. In this case a decision boundary that allows just 5% of misclassification has been arbitrarily chosen.

This allows to have a very low misclassification rate on the successful pulses, but at the same time it penalises the capability of the network to detect novel patterns and especially unstable plasma patterns as it is shown in Figure 6.4. However with this choice there is the hope that novel stable plasmas will not be detected.

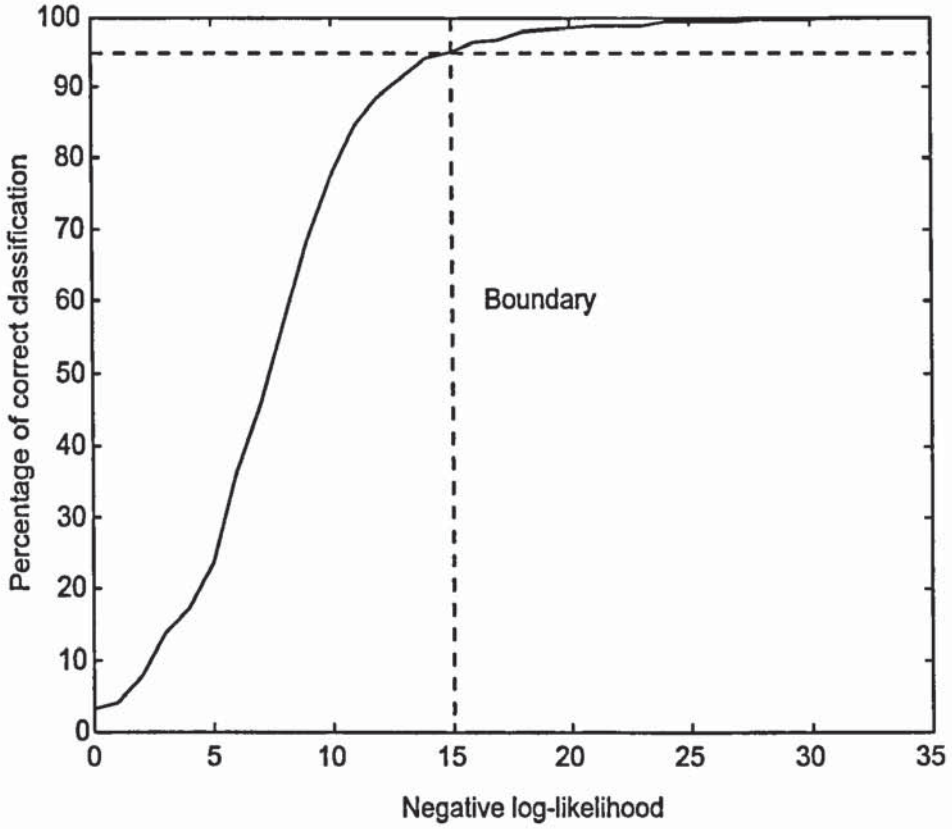

Figure 6.3: The drawing shows the percentage of correctly classified patterns at the variation of the decision threshold. To avoid to stop successful pulses because of misclassification problems, the decision boundary to decide if a pulse is to be flagged as novel has been set to 15.

The second approach is to use the disruption database and set the boundary so that it minimises the misclassification of both datasets.

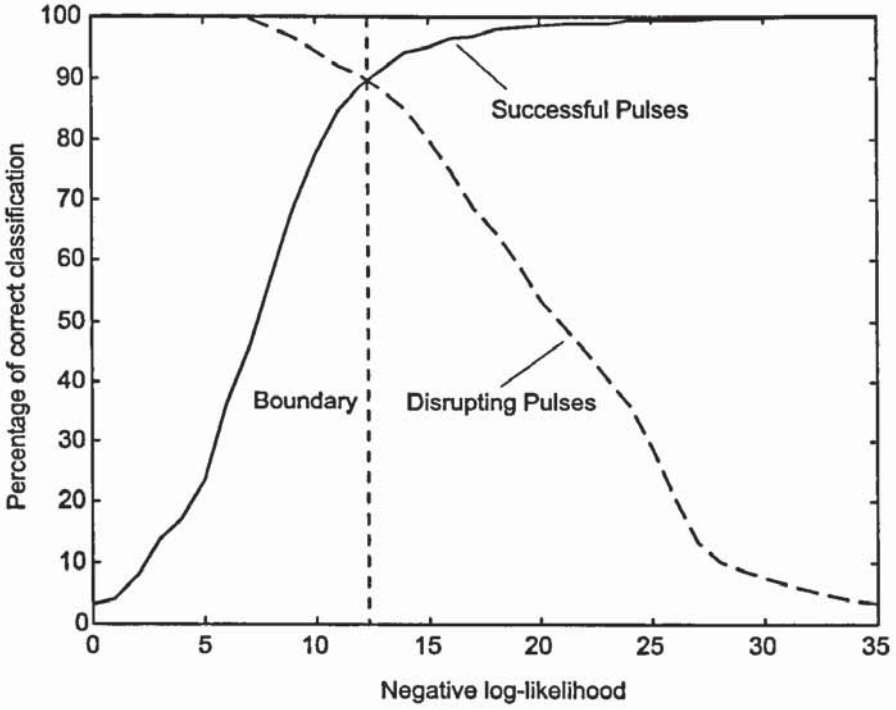

Figure 6.4: The drawing shows the percentage of correctly classified successful and disrupting patterns at the variation of the decision threshold. The same datasets of chapter 5 has been used. It can be seen that with a decision boundary of 13 we will obtain 10% error in our prediction.

The decision boundary of 13 will allow 10% of misclassification on both classes. The error is acceptable as regard the disrupting pulses but not in the case of successful pulses, where a much smaller percentage of failure can be accepted to obtain the confidence of the machine operators.

6.8 Results

From the plots drawn before good results can be expected from this algorithm. Drawing the negative log-likelihood for the whole pulse we can see if the pulse has patterns that can be flagged as novel and therefore outside the safe operational space of the machine. In the

examples below it can be seen how a good pulse stays below the threshold even during the additional heating period where the plasma parameters could vary considerably:

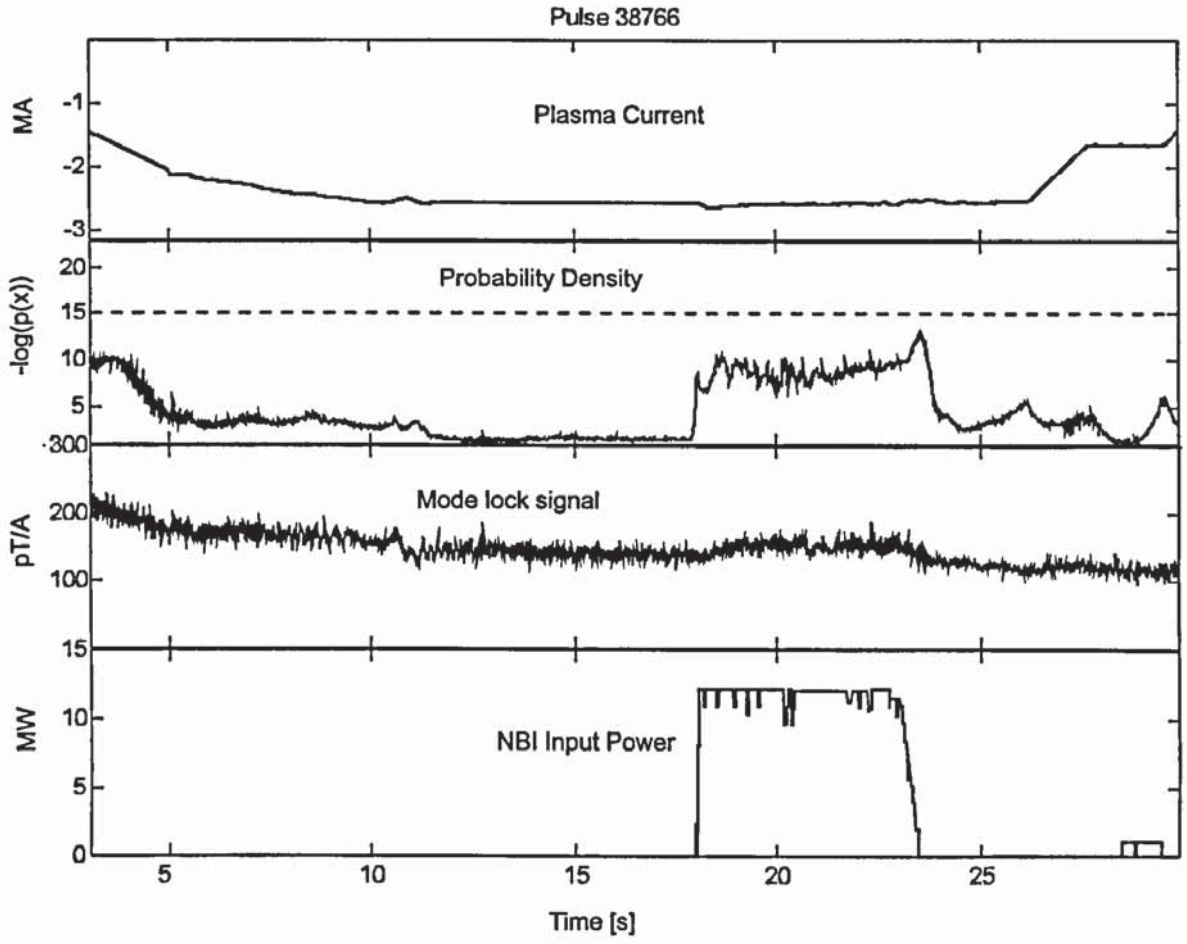

Figure 6.5: Probability density calculated for the whole length of a successful pulse.

In case of disruption, as we can see on the following pulse, the threshold is breached. It is also possible to see how the estimated value of the density probability is not strictly related to the mode lock signal or to the additional heating power.

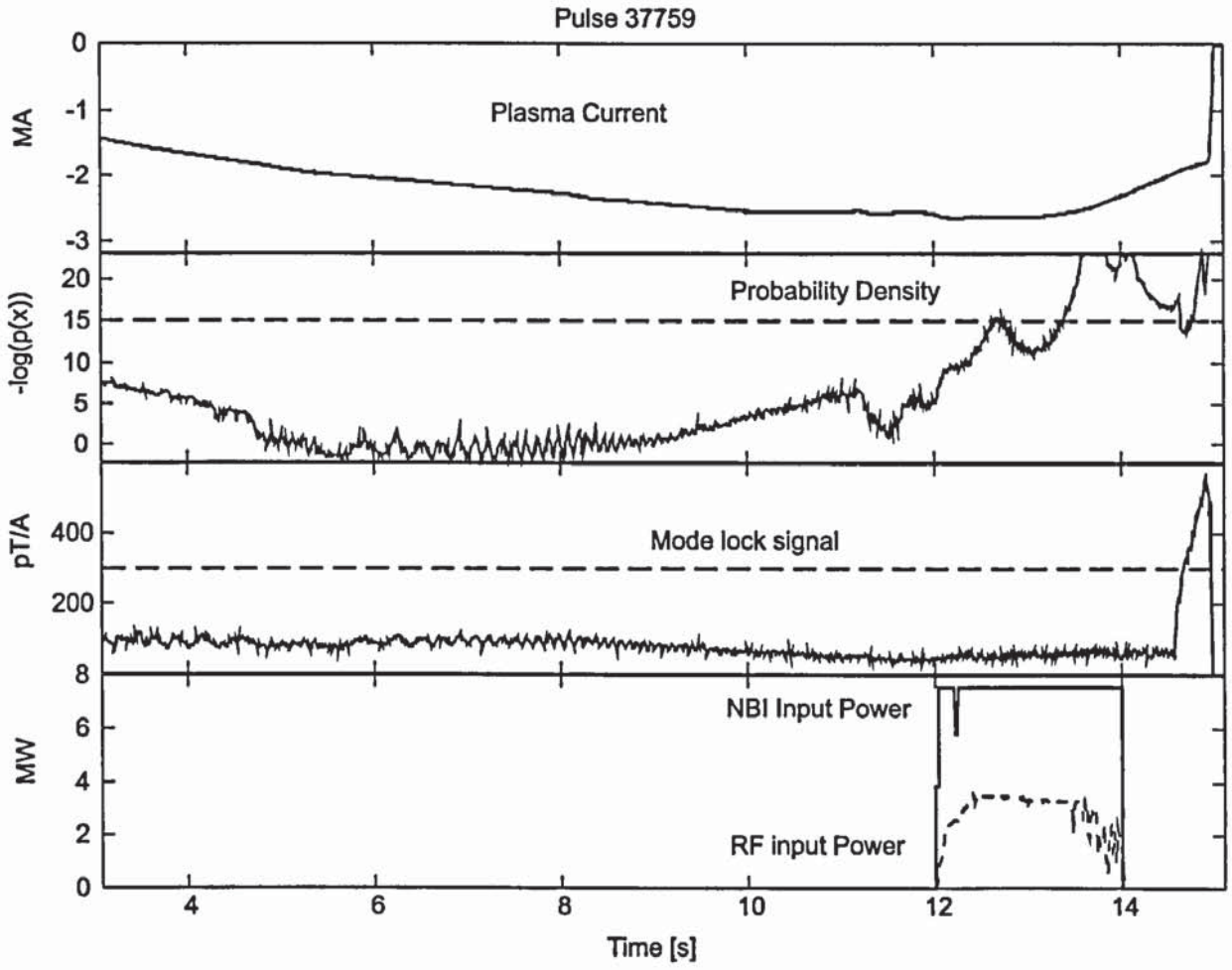

Figure 6.6: Detection of a disruption using the novelty detection algorithm.

During the last period of the thesis new plasma configurations have been attempted and further studies and optimisations on the old ones have brought new results and record improving JET performances. This constant searching of the machine limits is a feature of JET experiments and it gives little hope to novelty detection systems. It is difficult to set thresholds that can predict plasma disruptions and at the same time give the confidence that good pulses will be never stopped. In the example below a high current pulse would have been stopped if this method was used. This can be attributed to the fact that not so often pulses with around 22 MW of additional heating power has been run at JET. We have to remember that the novelty detection method does not give the possibility of distinguish between novel stable plasmas and unstable plasmas.

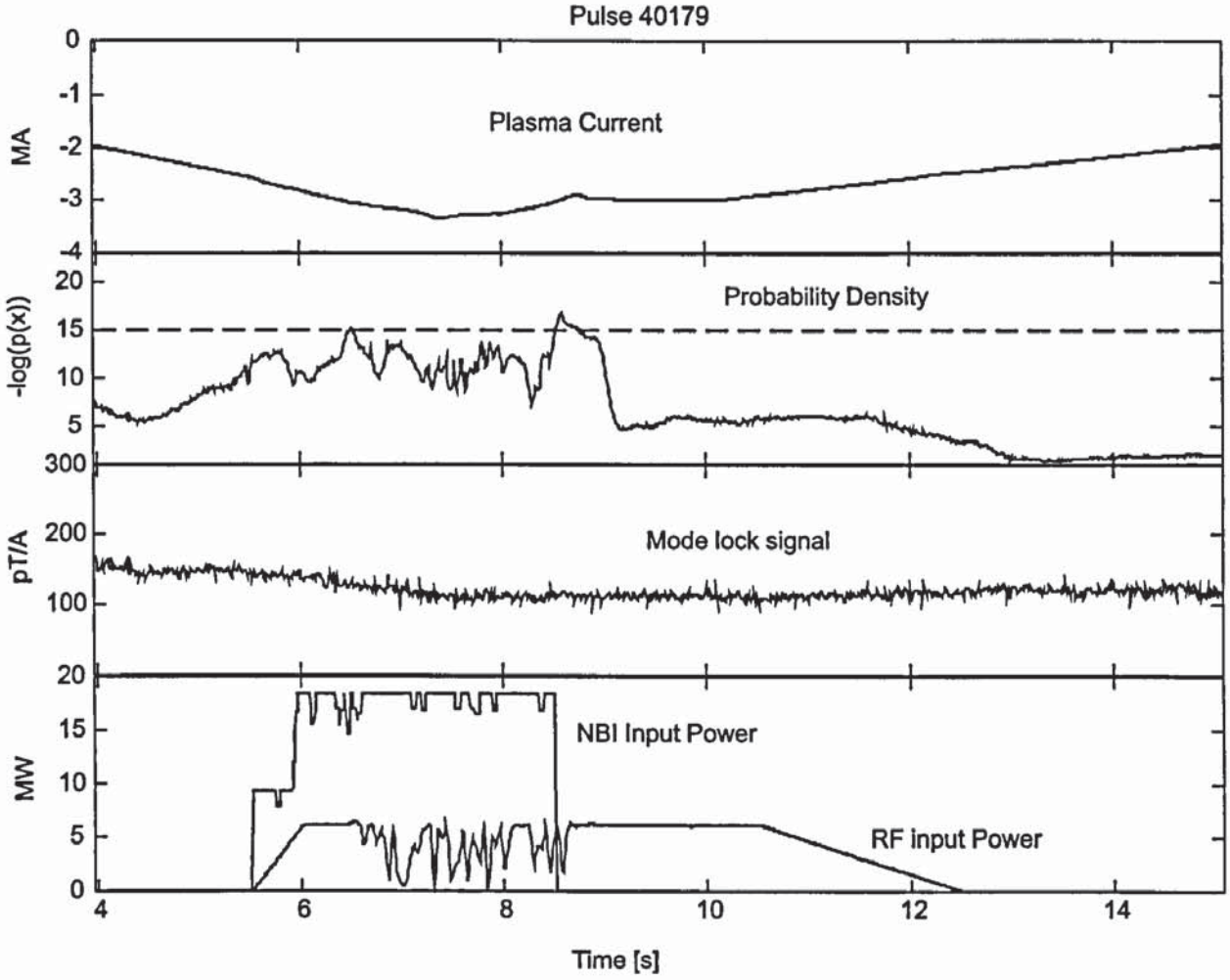

Figure 6.7: Novelty detection algorithms can not discriminate between novel stable plasma configurations, and unstable plasmas. The additional heating and the plasma current values of this pulse are close to the maximum level and therefore not so common during JET operation. The pulse was therefore novel to the network and has been flagged accordingly.

A possible solution is to retrain the network every time there are new data available. Including new stable plasma configurations and re-calculating the density distribution is unfortunately a long process. The new data needs to be checked to avoid to consider 'unstable' plasma patterns and the selection of new decision threshold has to be made. Nevertheless, in Figure 6.8 it is possible to see a comparison between the disruption likelihood determined using two different networks. The grey line shows the results for a network trained with 230 pulses (2300 patterns) compared with the network used to plot all the other results (1400 patterns used to determine the density distribution). The increase of performance is quite evident, suggesting that the inclusion of new patterns helps to better define JET operational space.

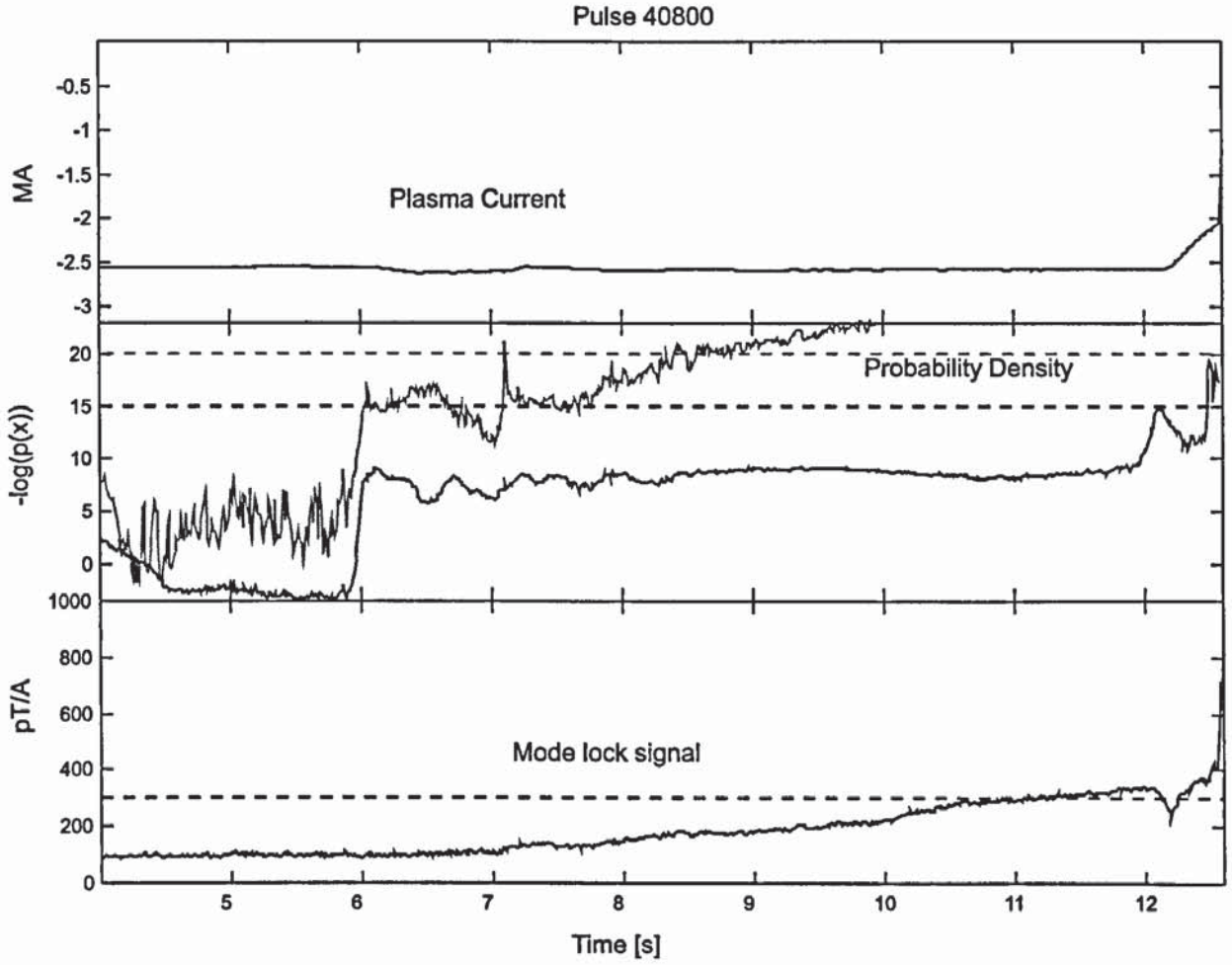

Figure 6.8: The probability density for this pulse has been calculated with two different mixture density. The first density distribution (black) has been determined using 1400 patterns and it is made of 28 Gaussian distribution functions. The second one (grey) is made of 29 Gaussian distribution functions that has been determined using 2300 patterns. It is clear to see that there is an increase of performance with the increase of pattern considered.

6.9 Conclusions

The results obtained in this chapter suggests that it is possible to use novelty detection algorithms to predict disruptions. The network developed consists of 28 Gaussian distributions and has 8 input signals. We have also seen how the whitening process used to normalise the data improves the performance of the network.

Compared to methods where a disruption database is needed, these algorithm presents the advantage that only successful patterns are used for the determination of the mixture

distribution parameters. Successful patterns are always easier to obtain and there is practically no limit on the size of the database. This gives the possibility of using a number of training patterns that is limited only by the computational power of the computer used to train the network. It has also been noted that increasing the number of input patterns could help in defining better the machine operational space and the boundaries between the plasma stable and unstable regions.

A disadvantage of these methods as seen on the previous paragraph is their incapacity of distinguishing between new and unstable plasma configurations. This can cause serious problems to their implementation on experimental machines.

However, their capability of flagging anomalies can be seen as an advantage to reactor-like-machines as ITER. In this case, if the plasma changes from its standard configuration, it is most likely that the control systems need to react to re-establish the normal conditions. Therefore, the use of novelty detection could be the answer for a prompt detection of any events that can affect the plasma equilibrium.

7. Vessel forces prediction

7.1 Introduction

One of the main reasons to develop algorithms capable of predicting disruptions is because the plasma energy released during these events, could exert forces on the machine structure threatening its integrity. A multilayer perceptron network has been developed to predict the forces generated by a given plasma configuration in case of disruption. This knowledge can be used to stop the pulse when the predicted forces will breach a given limit.

This objective is very difficult to achieve because the forces generated by a disruption not only depends on the total energy stored in the plasma but also on how the disruption develops. In the disruptions called Vertical Displacement Events (VDE) the vertical position control of the plasma is suddenly lost. The plasma moves vertically towards the vessel releasing most of its energy as mechanical force. These disruptions are the most dangerous and usually they generate very high vertical forces. In the case of radiative disruptions the plasma is kept in position by the control systems and most of its energy is lost as thermal radiation to the vessel. The forces produced by these disruptions are therefore negligible compared to the previous. In normal operation the two behaviours are combined, a disruption looses part of its energy as thermal energy and when the control systems can not control the plasma position any more a VDE occurs generating vertical forces. This increases the difficulty of the prediction because it is not possible to gather this information before the occurrence of the disruption. However, in this work, neural network techniques have been studied to see if they can provide a better prediction than that one obtained using traditional techniques.

7.2 Vessel forces measurement

At JET, the vertical forces are measured using strain gauges mounted on the main vertical port restraints. There are four strain gauges for each octant bonded on each upper and lower main vertical port restraints.

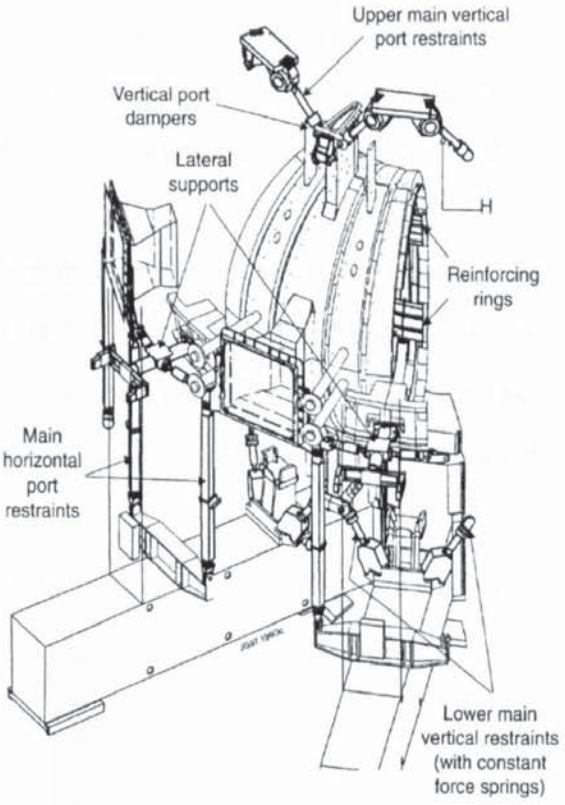

Figure 7.1: Restrains of an octant of the vacuum vessel

The measurement of the vertical forces is then performed, measuring the resistance variation of the strain gauges caused by the forces exerted on the vessel by the disruption. The total vertical force becomes:

$$Vertical\ Force = \cos(\alpha^u)\sum_{i=1}^{16} F_i^u + \cos(\alpha^l)\sum_{j=1}^{16} F_j^l \tag{7.1}$$

where α^l α^u are respectively the leaning angle of the lower and upper port restrains, and the F_i^u and F_j^l are the forces measured in Newton by each strain gauge.

The accuracy of the measurement is affected mainly by two problems:

- The resistance measurement is very difficult. It is performed using a Wheatstone bridge circuit. The signal is therefore very small and can easily be affected by noise. A complex amplifier circuit has been design to increase the ratio signal/noise. Nevertheless the measurement presents error estimated in the order of 5%.
- In several occasions some sensors were malfunctioning forcing their exclusion from the vertical force calculation. The detection of faulty measurements is done manually and from time to time, errors on the measurements are not detected, especially if the final measurement does not exceed the values that characterise a disruption as dangerous. This means that some of the data used to train the network, are not correct.

Another problem that can affect the training of the network is the lack of data for high values of vertical forces. At JET a disruption is considered to be dangerous for the vessel components when it breaches the limit of 4000 kN. The figure below shows, for the dataset considered, the number of disruptions plotted against the force that they produced on the vessel.

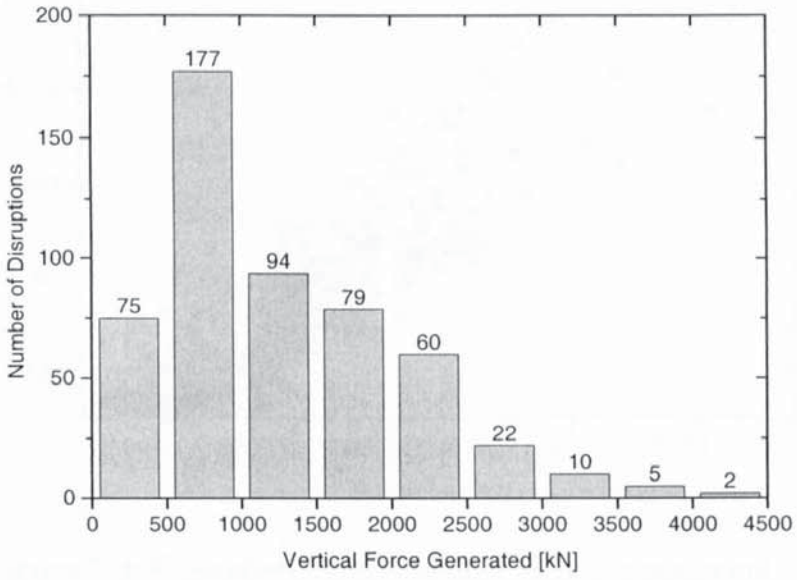

Figure 7.2: Number of disruptions plotted against the vertical forces generated.

It can be noticed that most disruptions generated forces below 2000 kN, therefore the network results for forces higher than 2000 kN are not expected to be very accurate.

7.3 F-number

Up to now few experiments have been performed to find a relation between the plasma configuration at the disruption time and the forces generated. Disruption tests, where the vertical position control system was switch off during a pulse (causing a VDE), have been used to determine a combination of poloidal field coils and plasma currents that can give a prediction of the vertical forces. The relation is based on a simplified model of the plasma and its interaction with the vessel structure and the poloidal field coils:

$$F-number = 1.0e-6 \cdot I_{Plasma}(0.29 \cdot I_{Plasma} + 0.72 \cdot I_{Shaping} - 13.6 \cdot I_{PFX}) + 2.1e-6 \cdot I^2_{Divertors} \qquad (7.2)$$

This F-number is currently used to assess before the pulse if the plasma configuration could generate forces above the allowed limit in case of disruption. The picture below compares the real forces during disruptions with the forces estimated by the F-number.

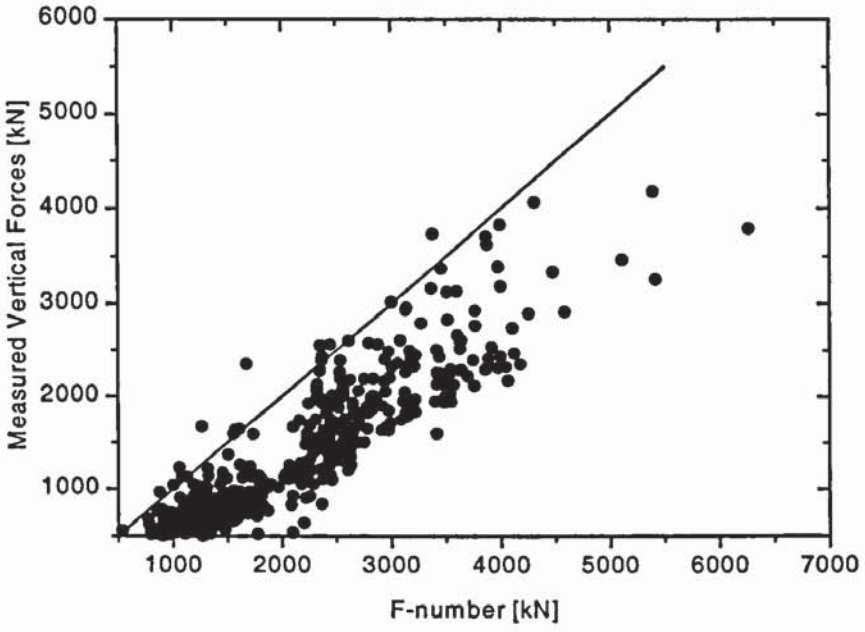

Figure 7.3: F-number against Vertical force measurement

Since the F-number shows very pessimistic values for some disruptions, the confidence on this algorithm is very low, and therefore it is not used in real-time to protect the machine structure. Neural network algorithms have been used to assess the possibility of predicting vertical forces with better precision. The requirement on the precision is quite strict because if the protection will be implemented in real time it will limit the operational space of JET

experiment. Therefore, the network prediction needs to be as accurate as possible to avoid to stop harmless pulses.

7.4 Neural Network prediction of vessel forces

Vertical force prediction using neural network presents several problems related not only to the difficulty of obtaining good data.

At JET the detection of the disruption precursor (using the mode lock signal) triggers a set of events that have the purpose to reduce the vertical forces. This involves the issue of a fast stop signal to the PPCC (Plasma Position and Current Control) Shape Controller that acts on the plasma current and configuration. The disruption usually occurs while the plasma configuration is changing, making the choice of the time when to consider the data critical. There is always the risk of selecting data that are not related to the disruption because they are taken too early, or that have no meaning because the plasma magnetic configuration is destroyed affecting considerably the accuracy of the measurements.

The force signal itself could not be used to give the precise moment when to select the data. The maximum of the forces is measured when the plasma is already disrupted. In Figure 7.4 it is possible to see a disrupted plasma and the vertical force signal. The plasma stored energy signal is plotted in the graph to show that the plasma energy is lost several milliseconds before the current drop. In this work it has been chosen to consider the plasma configuration at the energy quench. This is mainly due to the fact that at that time the plasma confinement is lost. Eddy currents starts to run through the vessel structure interfering with the magnetic signals. At this stage any plasma boundary reconstruction codes do not provide reliable data. This could create problems to the training, since the XLOC output is used as input of the network.

An example of the effects of the disruption on the magnetic signals is also the spike on the plasma stored energy after the energy drop. Even if it is known that the spike is not real the mechanisms that cause it are still not clearly understood and it can not be eliminated from the measurement.

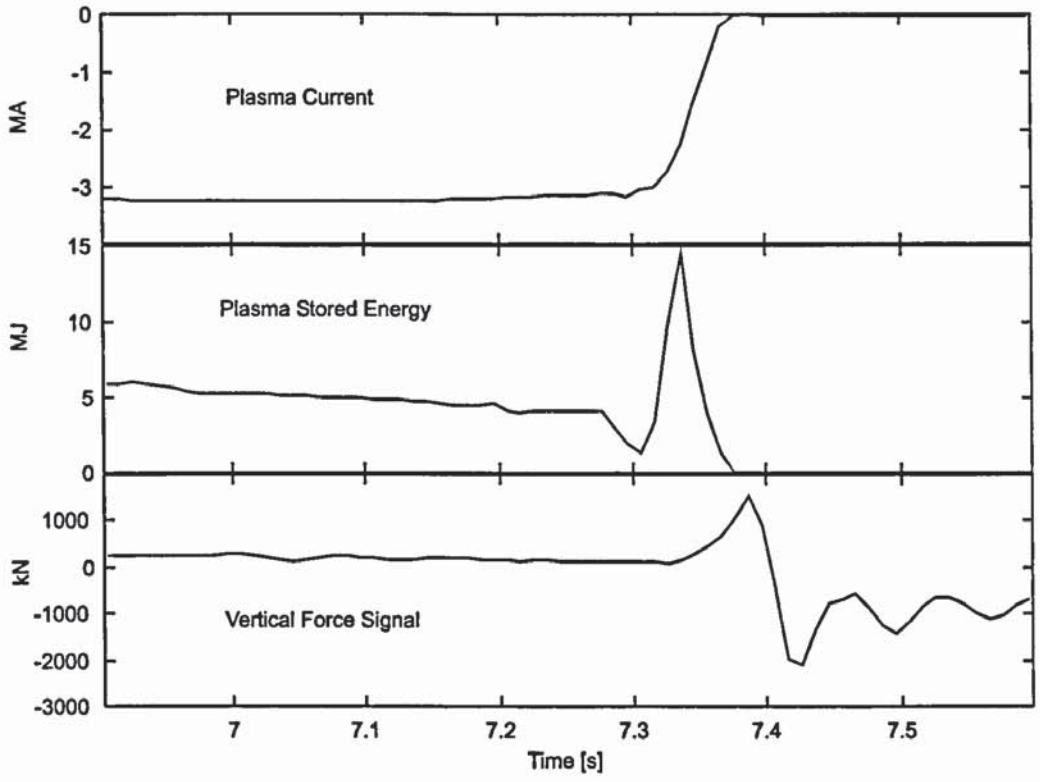

Figure 7.4: Example of plasma disruption and vertical force measurement. The disruption force is determined as the maximum amplitude of the signal variation.

7.4.1 Dataset generation

Two different networks were trained with two different input variables:

The first set is made of plasma parameters available in real time:

1. Plasma Current. In first approximation the vessel forces are proportional to the square of the plasma current.
2. Plasma cross section area.
3. Plasma Upper Elongation. It is strictly related to the vertical stability of the plasma and therefore the vertical forces should be proportional to it.
4. Radial field on the inner plasma boundary. This as the outer radial field is a first approximation of the plasma centre radial filed which should be zero if the plasma is in a stable configuration.
5. Radial field on the outer plasma boundary.

The second input set is made by variables that are available before the pulse. They are the pre-programmed plasma current and PF coils that the Session Leader pre-programs to obtain the desired plasma configuration. This gives the possibility to run the network before the start of the pulse and to decide if the plasma configuration could threaten the machine integrity in case of disruption. The inputs are:

1. Square of the plasma current.
2. Imbalance circuit current. This circuit is providing radial field.
3. PFX circuit current. This circuit is used to elongate the plasma.
4. P2+P3 coil current. They are part of the shaping circuit used as PFX to elongate the plasma.
5. Divertor coil currents. The sum of the four divertor coil currents is used because they strongly affect the plasma equilibrium, allowing the generation of the x-point formation and at the same time generating a strong radial field.

It has to be noted that these pre-programmed coil currents are not always available because the Session Leader can choose to control a plasma/wall distance instead of a coil current. In this case the network as it is implemented will not be able to provide a prediction of the vertical forces before the pulse. However, the network can be used in real time when the PF coil currents are available.

At the beginning of the training only 360 disruptions were available. A further reduction of the number of disruptions was made to exclude the disruptions that generated forces less than 500 kN because of the inaccuracy of the measurement at low forces. Also disruptions where a sensible plasma boundary reconstruction was not available 200 ms before the disruption occurrence were excluded to avoid the use of data that could be uncorrelated with the disruption itself.

The final dataset was made by 280 patterns. The modest size of this dataset limits the size of the networks due to the fact that the number of network weights that have to be found should be lower that the input patterns. Moreover also the network performance could be affected since we do not know if the 280 input patterns cover sufficiently the input space. To overcome the problem of the small number of input data the Bayesian technique describe in chapter 4, has been used for the network design. Then, all the patterns can be used for the training increasing the possibility of finding a good solution.

The testing of the network was performed using a dataset made of the disruptions that occurred during the period between the training of the networks and the final implementation. This set is made of 145 disruptions that belong to a period of operation where new configuration where tested and where a new fast stop scenario was implemented changing the plasma configuration from which the disruption started.

7.5 Network training

The network training was performed using the Bayesian technique presented in chapter 4. Also the method presented in chapter 3 has been used: gradient descent with adaptive learning rate and back-propagation method to calculate the gradient of the error function with respect to the weights.

For the Bayesian approach, all the 280 patterns have been used for training the network. The training has been repeated several times varying the number of hidden units. The best performing network has been chosen to be the network with the highest *evidence*.

As regard the method used in chapter 3, the cross-validation method was used to allow the use of a sufficiently big training set. The dataset of 280 disruptions was divided randomly in four subsets. Three of them were used to train the network and the remaining was used to validate it. This process was repeated for each of the four possible choices. The segment that was omitted from the training process was used as validation set. The error was then averaged over all the four results. The best performing network has been chosen selecting that one with the smaller error on the validation dataset after having trained several networks varying the number of hidden units. During the training phase regularisation techniques has been applied to encourage a smooth mapping. A penalty given by the sum of the squares of the adaptive parameters has been added to the error function.

$$E^n = \frac{1}{2}\sum_{k=1}^{c}\left(y_k(\mathbf{w},\mathbf{x}) - t_k\right)^2 + \nu\frac{1}{2}\sum \mathrm{w}^2 \tag{7.3}$$

where ν is called regularisation coefficient and it controls the smoothness of the model.

The optimal value of ν was found empirically varying it from 0.001 to 10 and training a set of networks for each ν varying the number of hidden units. Also the early stopping method was used to train the networks, and a comparison was done between the two methods.

The table below shows the comparison between the four best performing networks using the test dataset. Network 1 and 2 have been trained using the Bayesian approach and 3 and 4 using the second approach. Network 1 and 3 use the first set of input variables, described in the previous paragraph and Network 2 and 4 the second one.

		Hidden Units	Test Error	
Bayesian approach	Neural Network 1	5	0.36	α=0.4 β=8.9
	Neural Network 2	4	0.33	α=0.8 β=7.0
Chapter 3 approach	Neural Network 3	9	0.48	ν=0.01
	Neural Network 4	6	0.36	ν=0.01

Table 7.1: Error comparison between the four different neural networks. Network 1 and 3 have the same input variables (made by plasma parameters) and differ by the method used to determine them. Network 2 and 4 have different variables for the input (PF coil currents). The better performance of the first two networks could be explained by the fact the all the 280 patterns were used to train the network.

The results of the four networks are plotted for the test set of 145 disruptions acquired later after the training of the networks. The networks trained using the Bayesian approach have obtained a slightly better performance. This could be due to the fact that the whole 280 patterns have been used for the training, since there was no need of a validation dataset. Moreover the determination of the regularisation factor comes automatically from the training process and it has not been set empirically as with the second method.

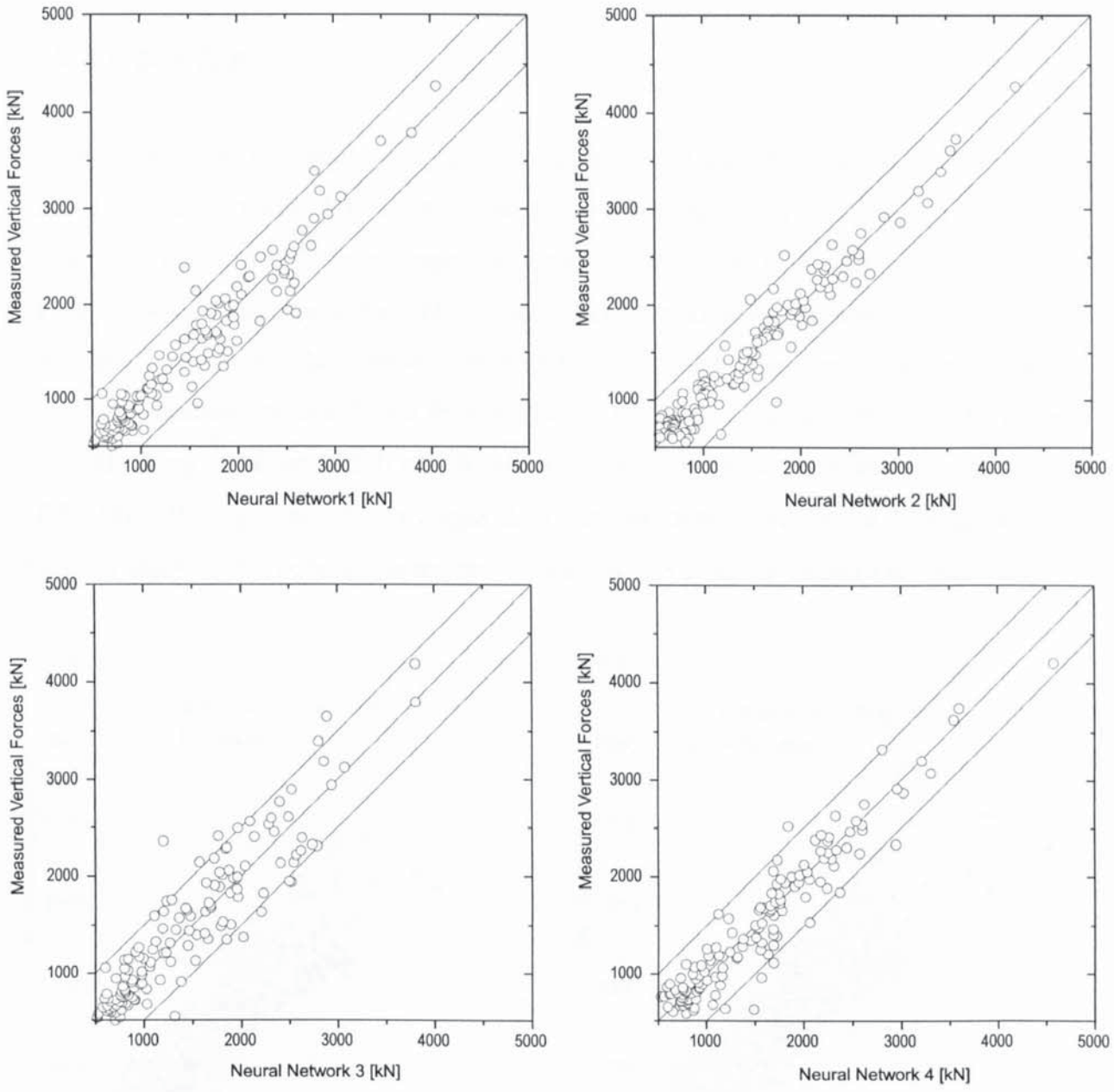

Figure 7.5: Neural networks results using the test set. Most of the network outputs has an error within 500 kN.

Most of the vessel forces predictions present an error that is less than 500 kN, and apparently this error does not scale with the absolute value of the predicted forces.

Due to the results obtained from now on we will consider only the neural network one and two, not only because their better performance, but also because the Bayesian approach allows to introduce the concept of error bars.

7.6 Error bars

As it has been said before, we are not interested in the determination of vertical force caused by the disruption but on the estimation of the maximum forces that it can exert. To obtain this value, two different approaches can be used. The first adds to the output of the neural network a constant offset. The offset is determined using the disruption database. The value is chosen so that the estimated maximum force will be the same or at least bigger than the actual measured forces. In the diagram below the results are plotted for all the disruption available using an offset of 500 kN. A different approach would have been the multiplication of the network output by a factor bigger than one. However, from Figure 7.5 it is evident that the error made by the network, is not proportional to the absolute value of the prediction.

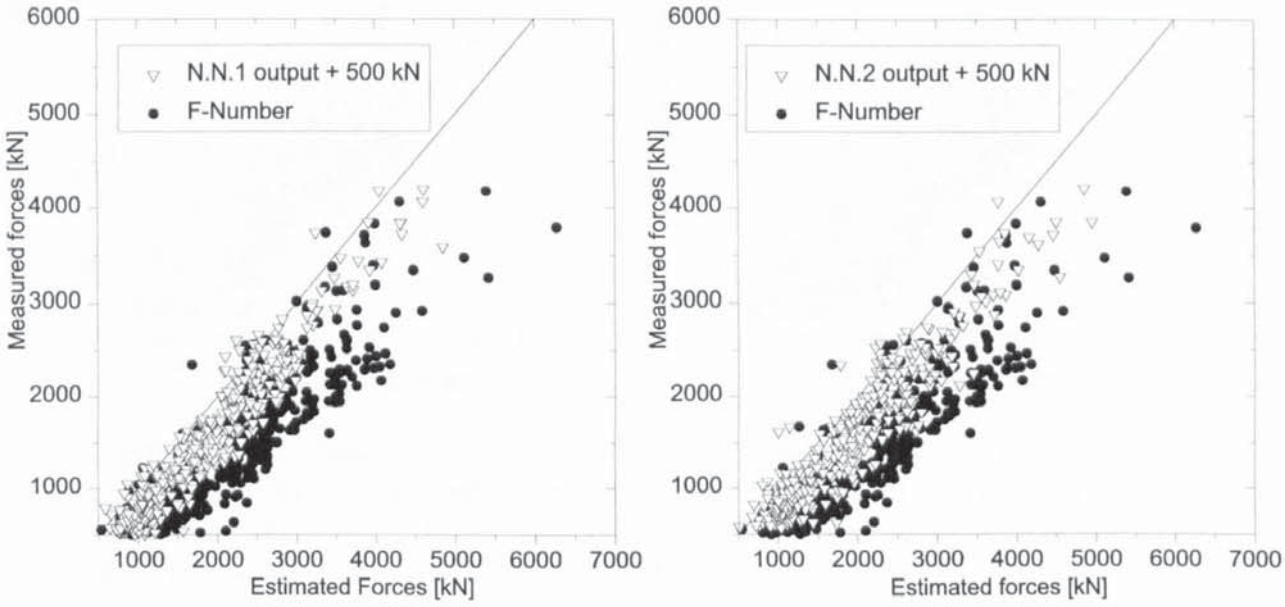

Figure 7.6: Comparison between the F-number and the output of the two networks offset by 500 kN for all the disruptions available.

The results show a reduced scattering output of the network especially for high forces, where it is more important to have a better accuracy in the prediction. Nevertheless the results are far from being optimal and this is probably due to the problems we mentioned before as lack of data, inconsistencies in the measurements, and the impossibility of predicting the kind

of disruption (radiative or VDE). The use of this method has also another drawback, which is that we are not sure that the offset chosen is correct also for future disruptions.

The second option is to use a more sophisticated approach to the problem using the error bar estimation to set the upper limit. The error bar can be calculated using the equation used to estimate the standard deviation of the network output.

This method has also the advantage to provide a confidence value on the prediction of the network. High values on the standard deviation will underline the fact that the input pattern fed into the network is novel and therefore the results have to be considered not fully reliable.

Summing $2\sigma_t$ to the networks output for the same dataset as above we obtain the plot:

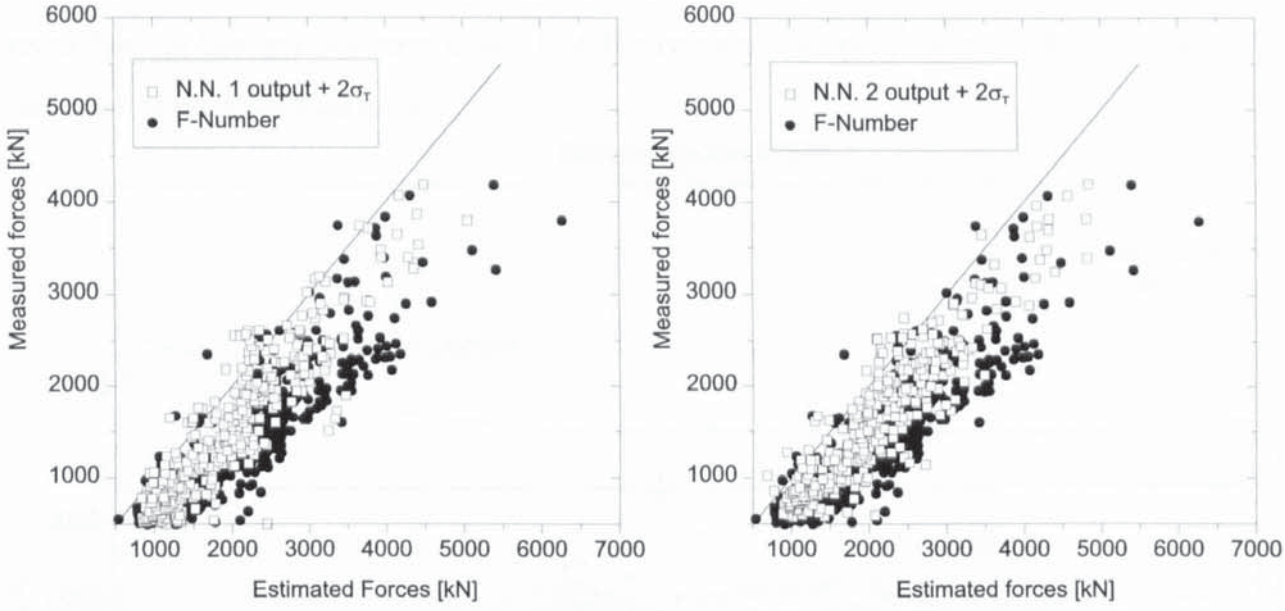

Figure 7.7: Comparison between the F-number and the network outputs for the test set. The data are plotted against the measured value. It can be seen how the results are just slightly better than the F-number. Some improvement has been obtained for the prediction of the high vertical force disruptions.

Clearly, the network prediction is more scattered than before but at the same time we will be capable to use the standard deviation to decide if the network output has to be retained as correct or not.

7.7 Results

The results of the network have been compared with the latest implementation of the F-number.

In the figures presented below there is an example of the same pulse with the prediction done by the two different networks. As in chapter 4 the results have been plotted in green for the networks considered as the best and in red the result of a committee of 30 networks, which their single output are visible in black. The vertical line represents the time when the disruption occurred. The horizontal line is the measured vertical force. The dotted curve is the F-number prediction. In this test it is easy to see how the two network predictions seem to be closer to the value of the measured forces.

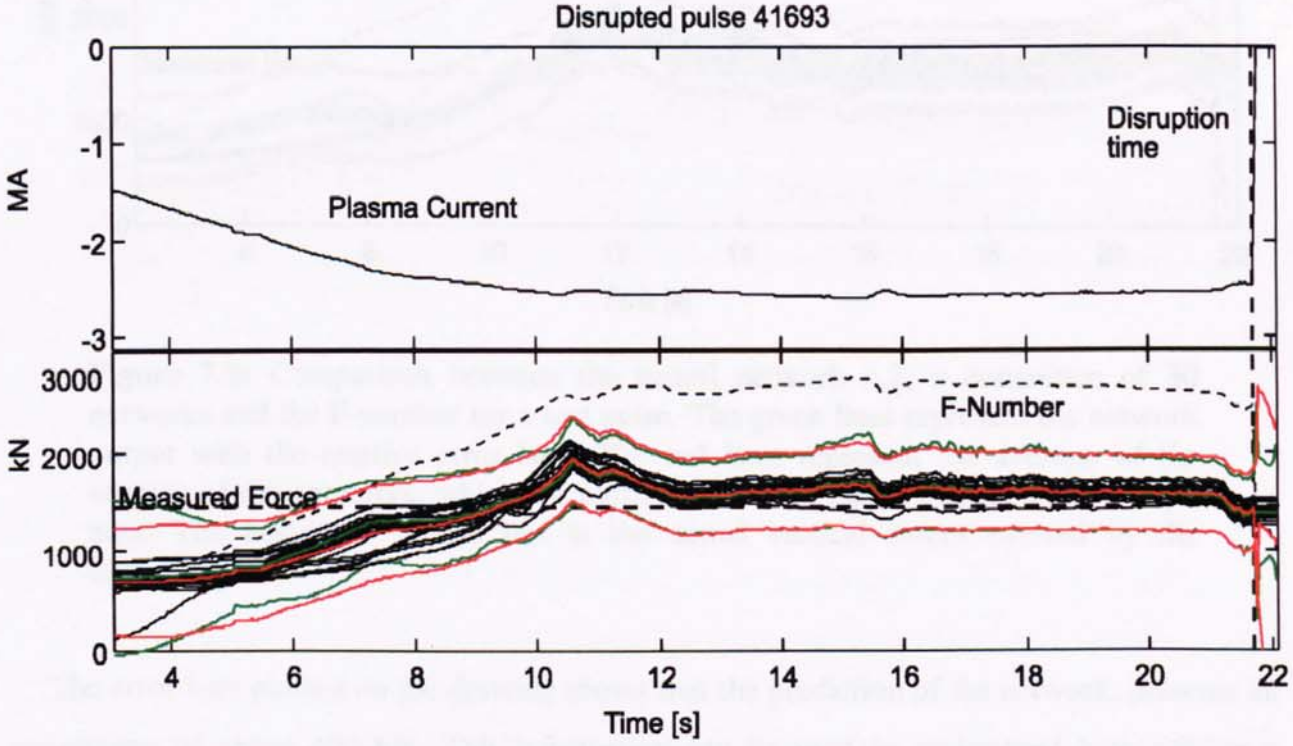

Figure 7.8: Comparison between the neural network n.1, a committee of 30 networks and the F-number for a test pulse. The green lines represent the network output with the relative error bars. The red lines represent the average of the outputs of the networks, which are plotted as well in black, and its relative error bars. The horizontal dotted line is the actual vertical forces exerted by the disruption.

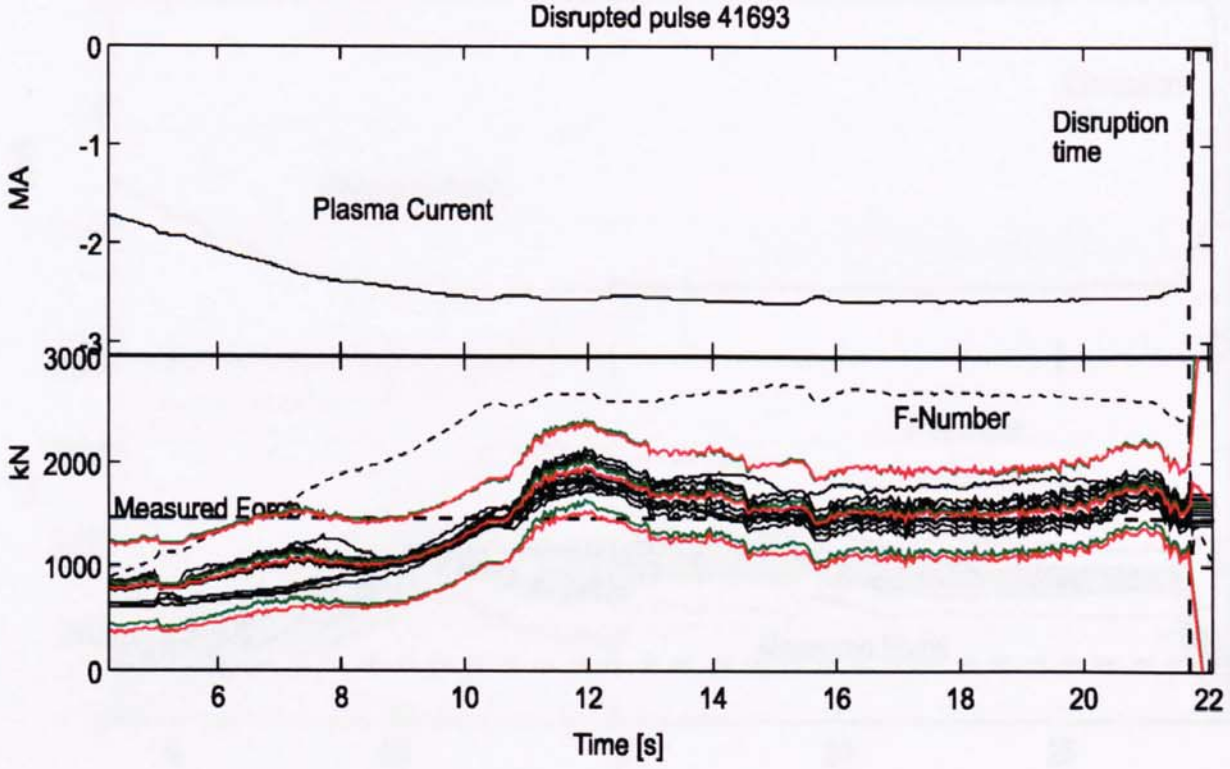

Figure 7.9: Comparison between the neural network n.2, a committee of 30 networks and the F-number for a test pulse. The green lines represent the network output with the relative error bars. The red lines represent the average of the outputs of the networks, which are plotted as well in black, and its relative error bars. The horizontal dotted line is the actual vertical forces exerted by the disruption.

The error bars plotted on the drawing shows that the prediction of the network, presents an uncertainty of about 400 kN. This information can be used to understand how reliable a prediction could be. If an extremely cautious approach has to be taken, the limits can be set using network results with the error bar added.

Unfortunately, the improvement on the prediction is not so evident for some of the tested pulses especially if the measured forces are low. In Figure 7.10, a disruption that produced very low forces on the machine structure can be seen. The higher predicted value could be connected to the fact that there is no possibility of understanding how the disruption evolves

from the data used to train the network. If the stored energy will be dissipated by thermal radiation the forces exerted on the structure will be minimal.

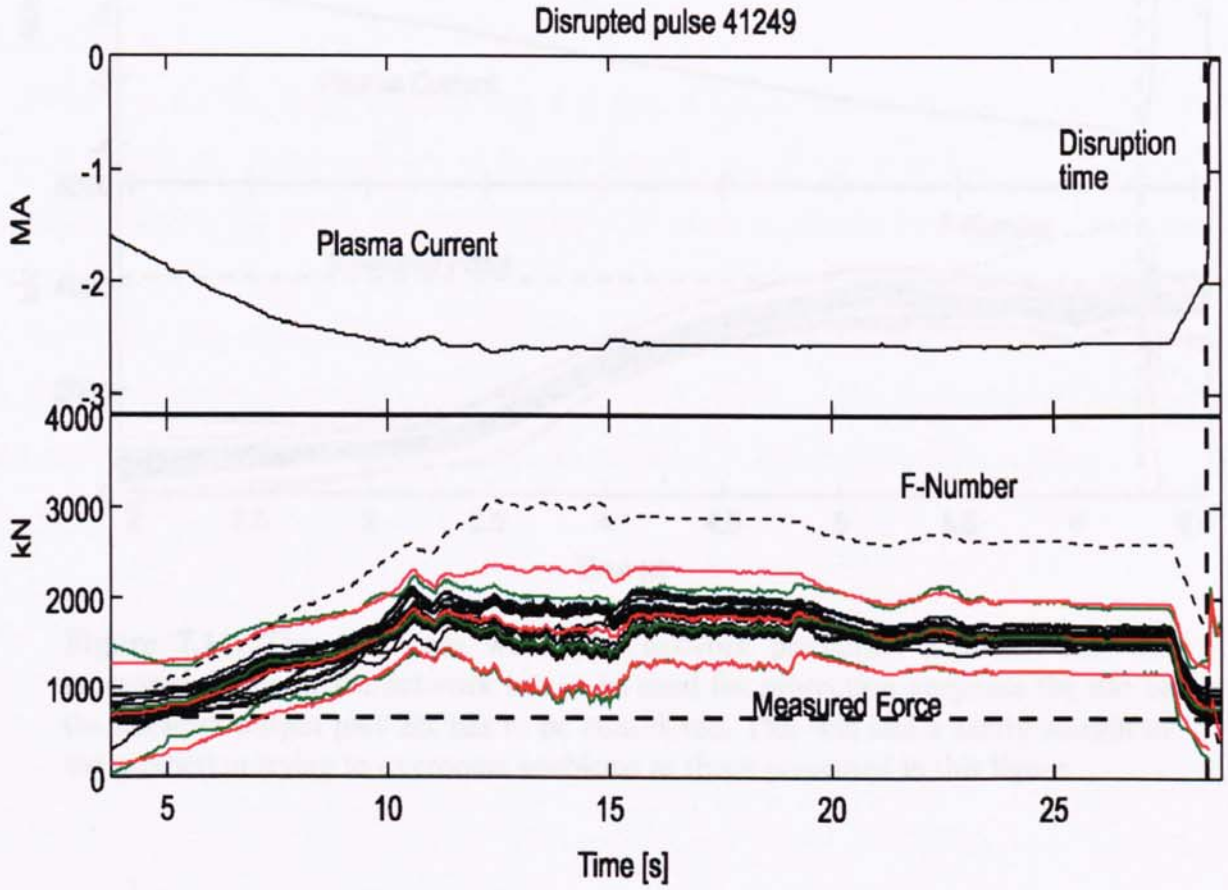

Figure 7.10: The network output for a radiative disruption. The predicted value is much higher than the actual value (horizontal line). Nevertheless the network prediction is closer than the F-number

On the contrary, if by any reasons the vertical position control is lost, all the energy i9s released on the machine structure. Because the network was trained using a mixture of the two types of disruptions, there is the possibility that for pure VDE disruptions the value estimated by the network is lower than the actual value. In Figure 7.11 it is possible to see a comparison between the actual and the predicted forces for a VDE. .

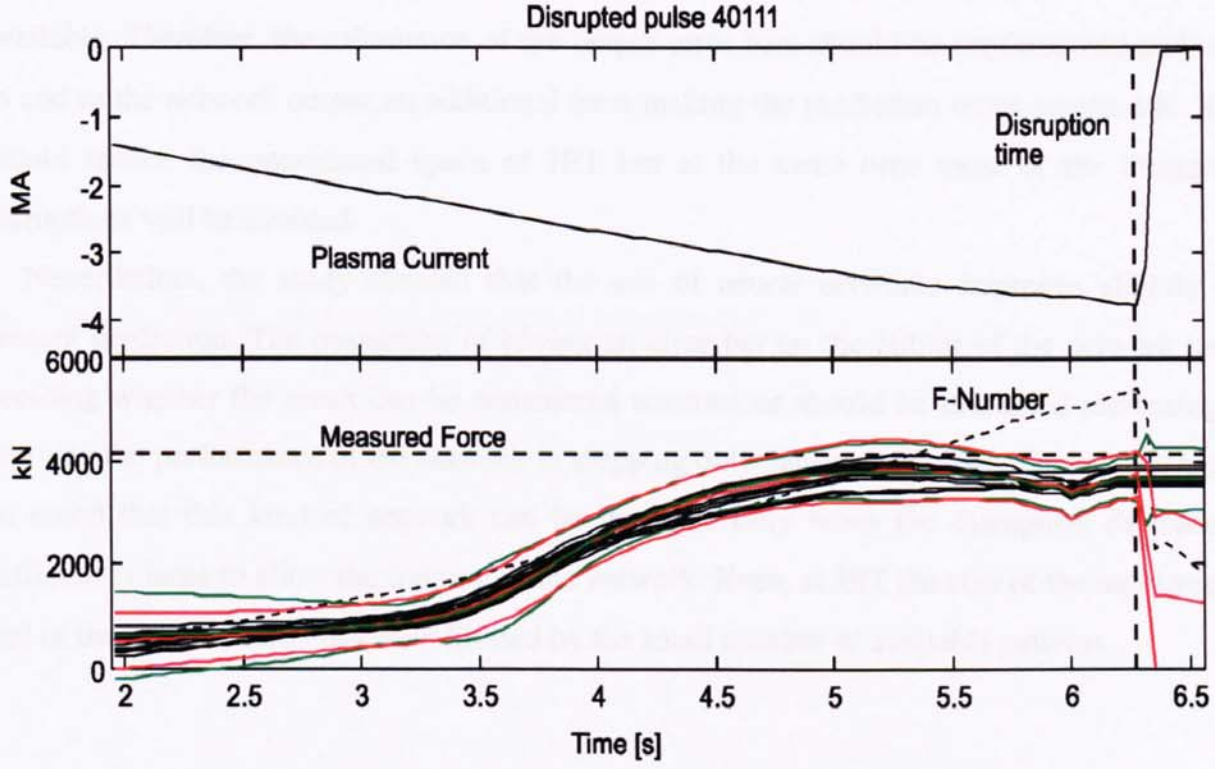

Figure 7.11: This is a case where the network prediction is lower than the measured forces. If the network has to be used for protection purposes the use of the network output plus $2\sigma_t$ has to be considered. This will add a safety margin to the prediction trying to overcome problems as those presented in this figure.

7.8 Conclusions

The design of these networks shows several problems that need to be solved before obtaining a more precise and reliable prediction of the vertical forces generated by a disruption.

The measurement of the vertical forces needs to be improved. The uncertainties on the measurements need to be removed designing an automatic method to check the correctness of the measurement removing the signals that are malfunctioning from the calculation of the vertical forces.

If the network has to be used for protection purposes, an extremely cautious approach has to be taken supposing that all the disruptions can generate the maximum possible forces, which means to suppose that all the disruptions would be VDE. Unfortunately, the network can not to be trained using just VDE disruptions because of the insufficient quantity of data available. Therefore, the calculation of the output error bars should be implemented and used to add to

available. Therefore, the calculation of the output error bars should be implemented and used to add to the network output an additional term making the prediction more pessimistic. This would reduce the operational space of JET but at the same time most of the dangerous disruptions will be avoided.

Nevertheless, the study showed that the use of neural networks improves slightly the present prediction. The possibility of having an error bar on the output of the network helps deciding whether the result can be considered accurate or should be discarded increasing in this way the performance of the network in stopping only dangerous pulses. However, it has to be noted that this kind of network can be designed only when the disruption database is sufficiently large to allow the training of the network. Even, at JET the size of the input vector and of the network itself has been limited by the small number of available patterns.

8. Disruption prediction and prevention system results

8.1 Introduction

One of the aims of this thesis is the implementation in real time of a protection system that runs all the algorithms described in the previous chapters. The disruption prediction and prevention system (DPPS) has been implemented with a combined effort of the plasma control group and particularly of Dr. B. Tubbing.

Provisional results have been obtained before the end of the Mark II divertor campaign and are briefly presented in this chapter.

As agreed with JET management at the beginning the system will be considered experimental and it has been designed not to interfere with normal operation. During this period the system will run 'on-line' synchronously with the JET pulse but it will not produce alarms or warning. Moreover, its malfunctioning will not cause any delay to the operation. Under this limitation the system has been developed and the basic algorithms are now implemented providing information on their behaviour.

8.2 DPPS Implementation

The system consists of a dual Pentium Pro 200 industrial PC to provide the computational power required to run the DPPS code in less than 10ms.

A reflective memory network has been implemented to synchronise DPPS with the JET pulse and to acquire real-time data. This high performance, daisy-chained network provides a very fast way of sharing data across distributed computer systems. Data is transferred by writing to the reflective memory board global RAM. The data is automatically sent to the memory of all the boards on the network. The network connects the Plasma Position and Current Control system (PPCC) and the Real Time Signal Server (RTSS) to DPPS providing the necessary input data and other information related to the pulse status.

The system is also connected to a PC in the control room via fast Ethernet. This PC (housekeeper) has the functions of interfacing DPPS to the operator and to the JET computer network. Through the house-keeper the operator can monitor DPPS, download new code and remotely debug it. The housekeeper is also in charge of transferring the input parameters from the JET computer network to DPPS before the beginning of the pulse. After the pulse it collects DPPS output and sends it to the Jet Pulse File archiving system.

The "Worker" software (the on-line computer where DPPS is running) consists mainly of two elements:

1. A state machine running synchronously with the JET pulse, and executing calls to 'user' functions.
2. 'User' functions, supplied in a Dynamic Link Library, which contain the DPPS algorithms to be executed.

My task was to design and to implement the user functions so that they could be integrated with the Worker software. The C code developed comprises:

1. Xloc off-line code to generate before the pulse the data necessary to run the xloc online code.
2. Xloc on-line code to determine the plasma boundary position and its magnetic field. These data are then used by the neural network algorithms.
3. The implementation of the MLP networks described in chapter 3, 4,5 and 7 to determine the plasma parameters, to predict plasma disruptions and the vessel forces.

4. The implementation of the novelty detection algorithm described in chapter 6.

All the network weights are loaded before the pulse and can be changed between pulses giving the possibility of testing different networks.

8.3 Results

In the following figures the first results of DPPS are presented. They show a good agreement with the tests that were done showing that the system is performing as expected.

8.3.1 li-q network

The comparison with EFIT and FAST suggests the aim of obtaining a good estimation of the plasma internal parameters in real time has been achieved. Our aim was to obtain these parameters every 10 ms, however the xloc code together with the neural network code runs in less the 2 ms.

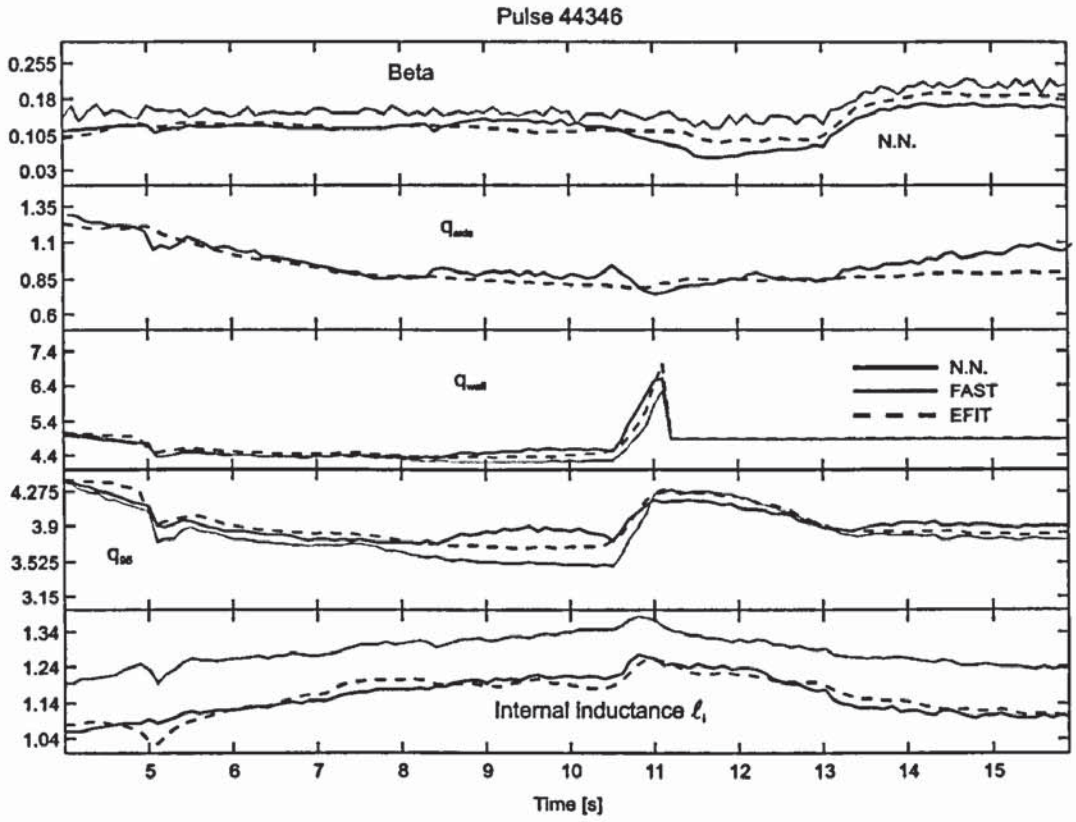

Figure 8.1: In figure we can see the results of the li-q network compared with EFIT and FAST.

8.3.2 Softmax network

For the last 116 pulses of the Mark II campaign, DPPS detected all the 22 disruptions that occurred. For 11 of them, the disruption was detected at least 100 ms before the triggering of the pulse termination by the mode lock protection.

To confirm that the network was working as expected DPPS results were compared with the simulated results. For pulse 44346 (Figure 8.2) we can see that the DPPS detection of the disruption is about 20 ms after the simulated prediction. The explanation of the difference between the two results can be found on the radiated power signal. The radiated power signal provided by the Real Time Signal Server is strongly filtered to reduce its noise. This also reduces its response to fast transients and as a consequence the network output is affected.

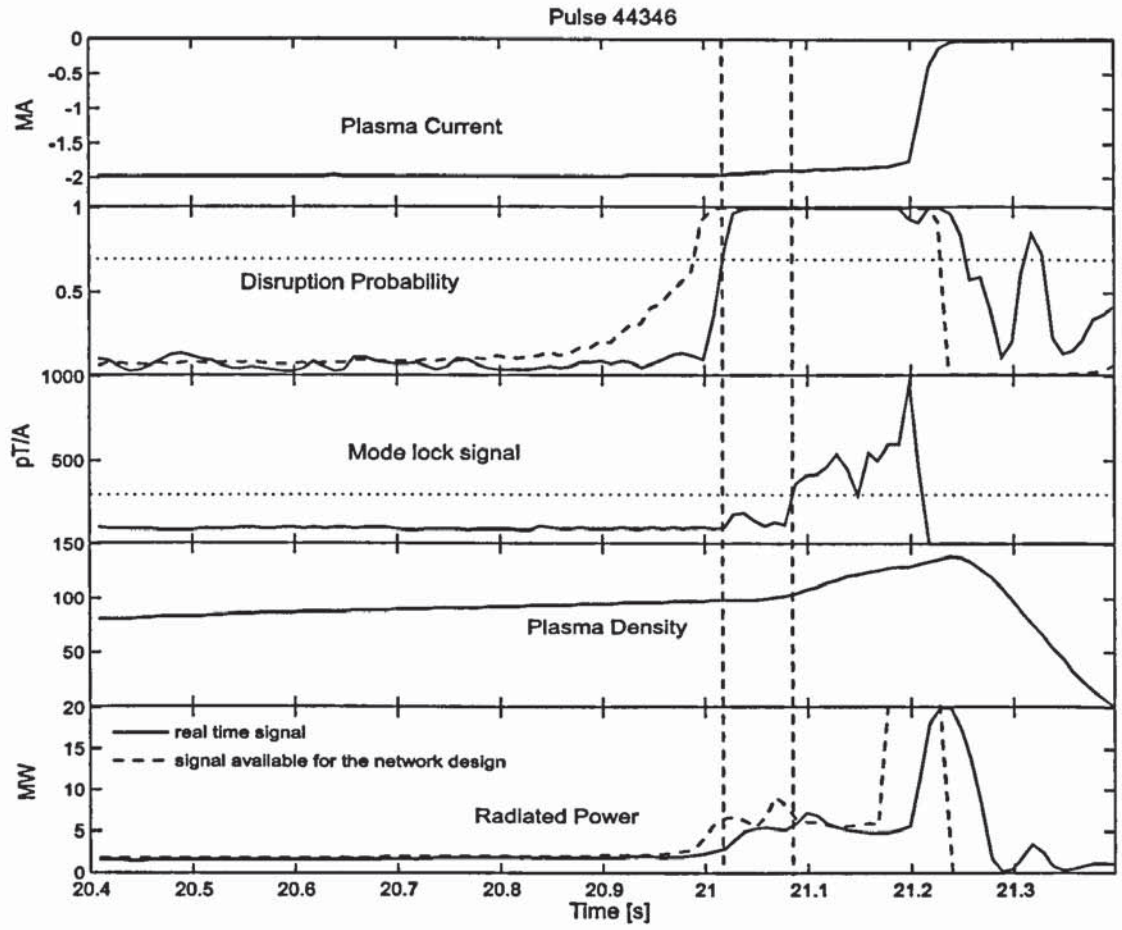

Figure 8.2: Comparison between the DPPS softmax network outputs and the results obtained in simulation. The radiated power signal available in real time is strongly filtered, causing a degradation of the performance of the network. Nevertheless, the disruption was predicted 70 ms before the mode lock signal detection.

During the same period of operations, DPPS would have stopped 5 of the 73 successful pulses. For all of them, the causes of the detection have to be related to of two events. The occurrence of a giant ELM, or the sudden switch off the additional heating systems.

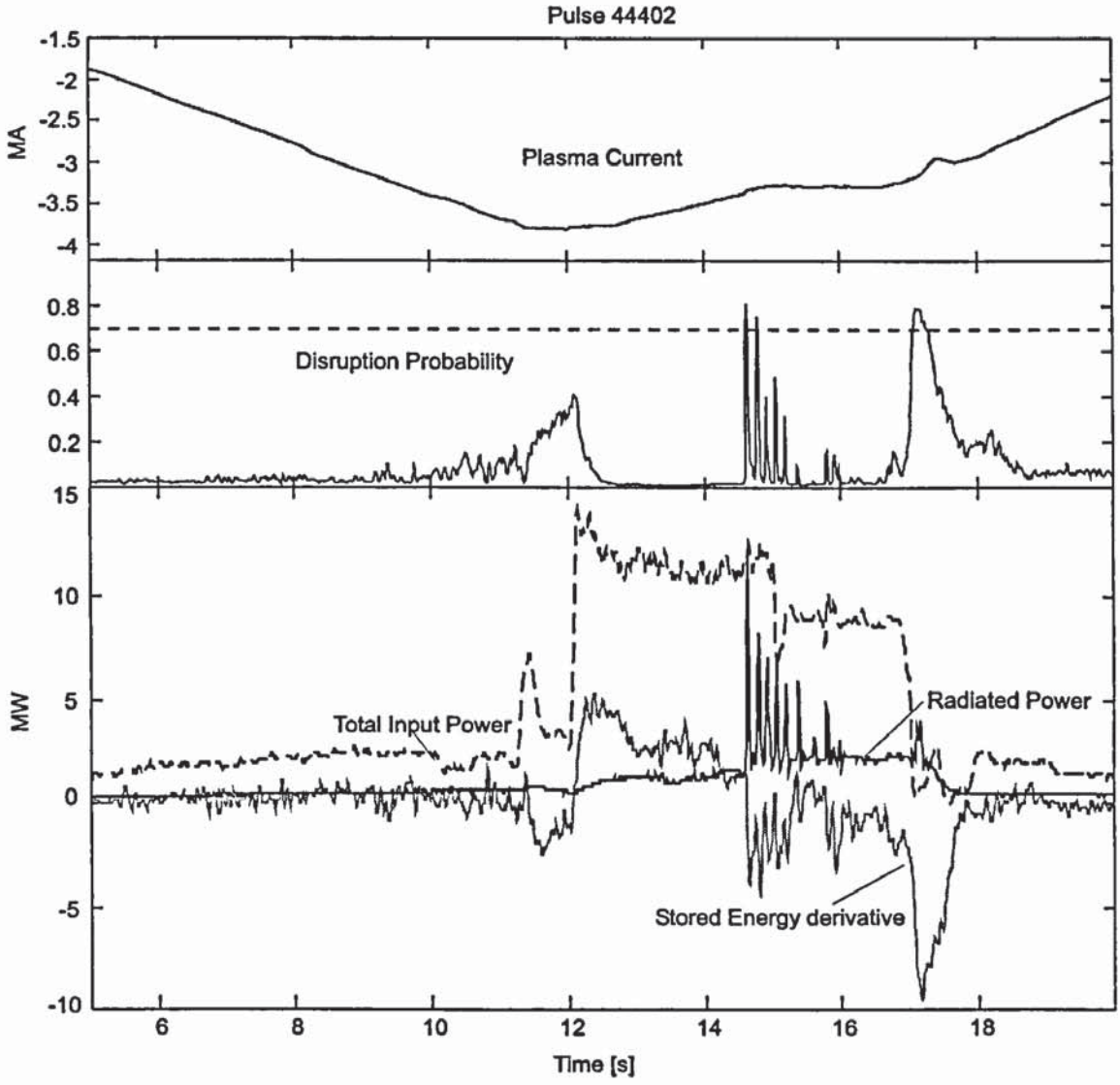

Figure 8.3: The ELM period (characterised by the energy bursts in the radiated power and by the negative spikes in the plasma stored energy), together with the additional heating systems switching off, are the most critical period of the pulse. To avoid the classification of these periods as disrupting, the decision threshold has to be set to 0.8.

Pulse 44402 is a high current pulse that presents both phenomena. The pulse would have been stopped by DPPS at the first spike caused by a giant ELM. If an assertion time of 20 ms was set before triggering the pulse termination, the pulse would have continued up to the

conclusion of the additional heating phase. This phase is very critical and in this specific case, there is a loss of 6 MJ of plasma stored energy and an increase of the radiated power to 4 MW in connection with a drop of 8 MW of input power. The plasma current itself dropped by 300 kA and the plasma moved to the outer wall changing configuration from x-point to limiter.

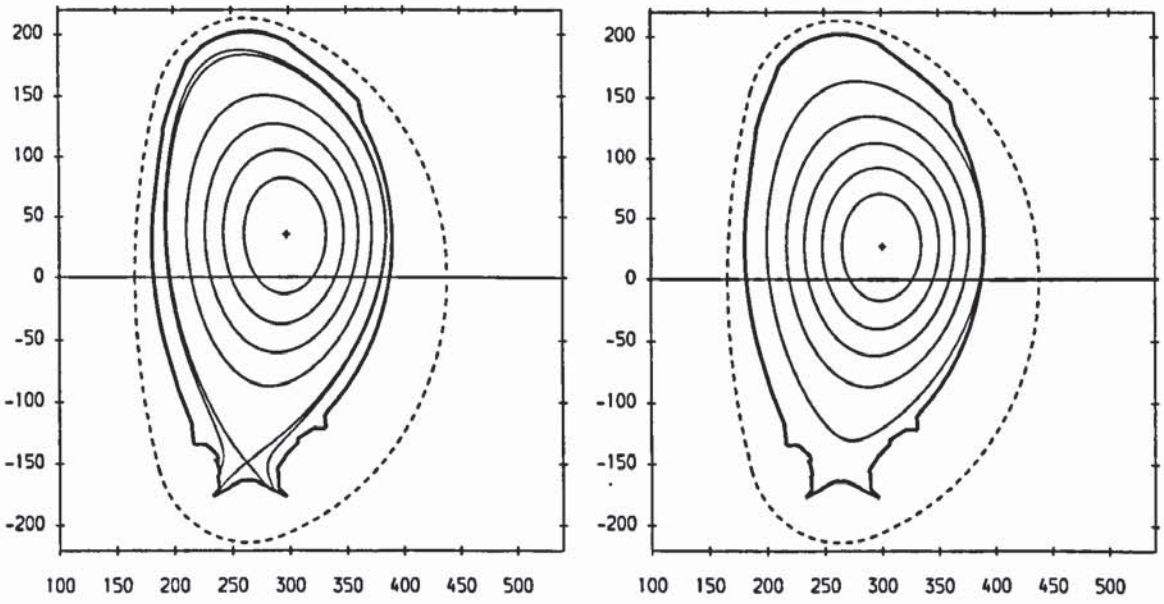

Figure 8.4: Plasma cross section for pulse 44402 at 17.3 (left) and at 17.4 (right). The plasma moved on the outer wall changing configuration. This is probably due to the reaction of the position control system to the neutral beam switching off, that caused a drop in the input power, in the plasma stored energy and in beta.

All these events combined together could be related also to a plasma disruption and are therefore very difficult to discriminate. Moreover, the pulse is novel to the network and this is reducing the accuracy of the prediction. An easy solution to the problem is to increase the boundary decision threshold. Using 0.8 as boundary decision and an assertion time of 20 ms no successful pulses would have been stopped, but the prediction of the disruptions would have been delayed by 20÷40 ms.

8.3.3 Novelty Detection

As regard the novelty detection network described in chapter 6, this problem is less serious, and no successful pulses would have been stopped. As an example pulse 44402 is shown in Figure 8.5 where the estimated probability does not breach the limit for the whole pulse.

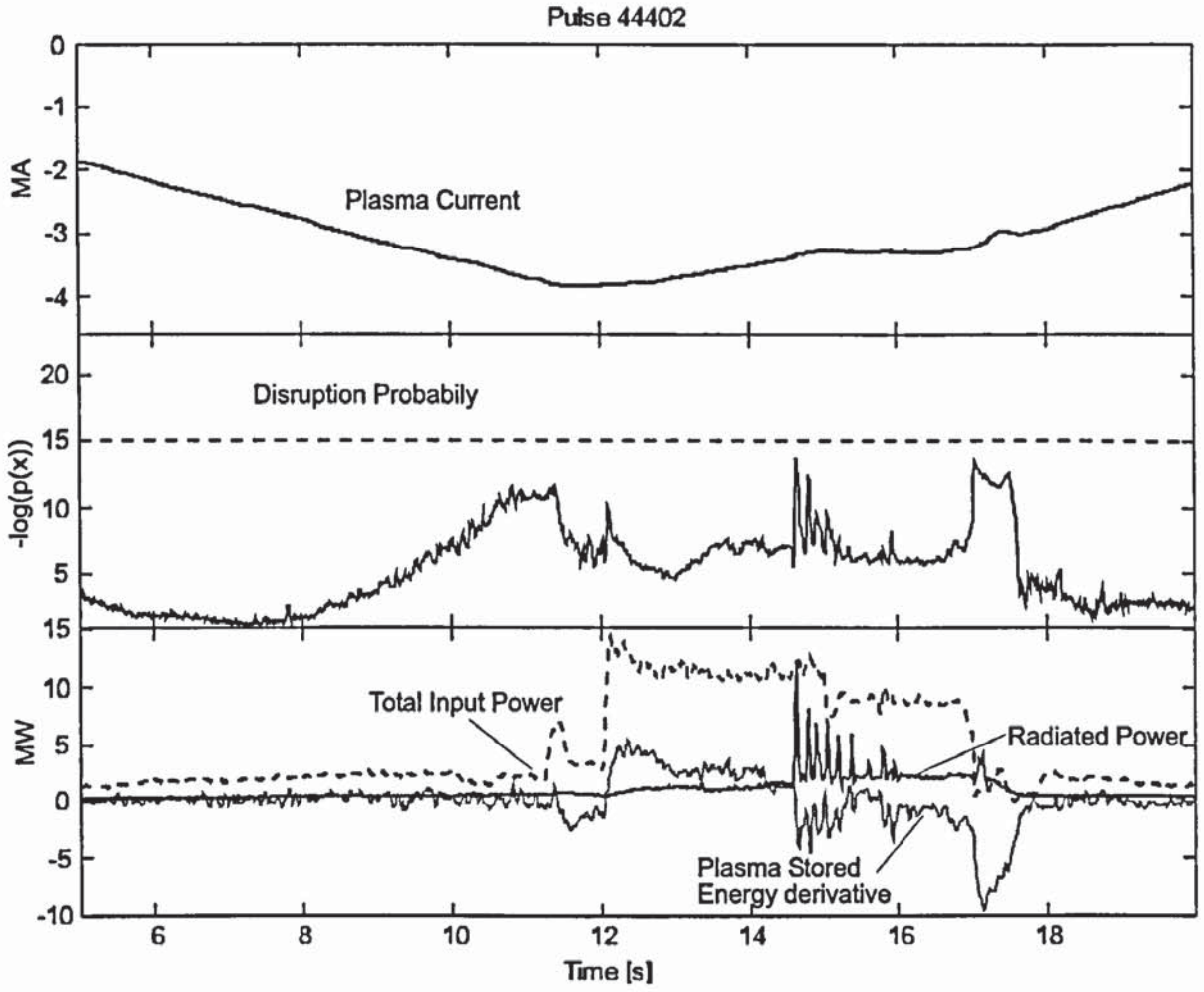

Figure 8.5: The figure shows how the novelty detection does not misclassify the ELM phase and the additional heating termination phase. However, they are still recognisable on the disruption probability density.

Unfortunately, the novelty detection network predicts the occurrence of a disruption later than the softmax network reducing the time available for action. Figure 8.6 shows the same disrupted pulse of Figure 8.2. In this case the mode lock protection is anticipated by 50 ms. Only 9 disruptions would have been detected more than 100 ms in advance. To understand

this different behaviour from the softmax network, we have to remember that this network was trained using a bigger dataset of successful pulses, and patterns that belong to the ELMy phase and to the switching off of the additional heating systems were introduced in the training set. At the same time the knowledge of the behaviour of disrupted pulses was not used to training it. On the contrary, the softmax network can make use of this information recognising promptly the unstable plasma patterns.

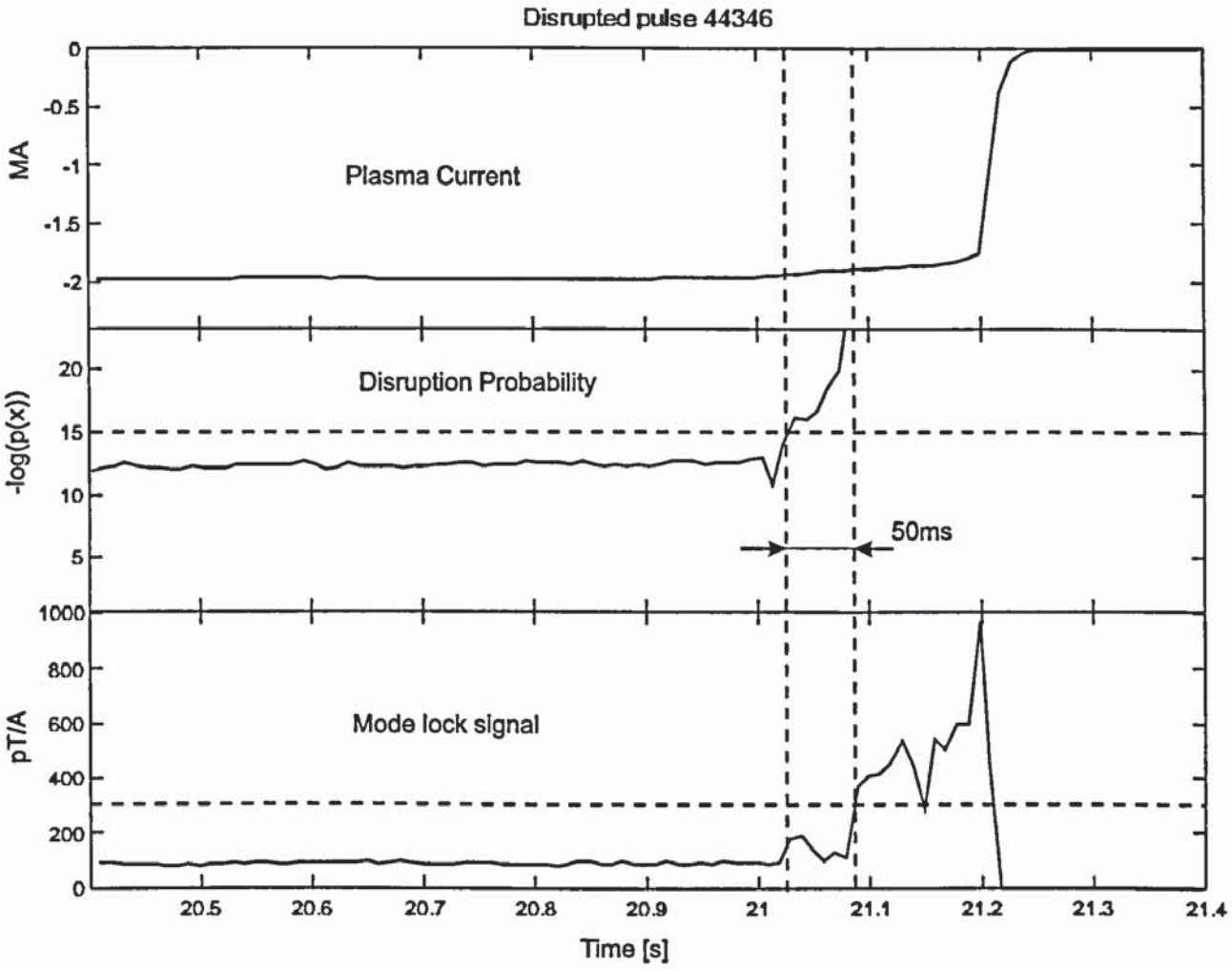

Figure 8.6: Novelty detection results for the same pulse of Figure 8.2. The detection of the disruption happens 20 ms later than the softmax network.

8.3.4 Vertical Forces Prediction

Only three disrupted pulses generated vertical forces higher than 1000 kN. With regard to pulse 44315 (Figure 8.7) we can see how the prediction is very close to the measured vertical forces(horizontal dotted line) and that the error bar is less than 300 kN.

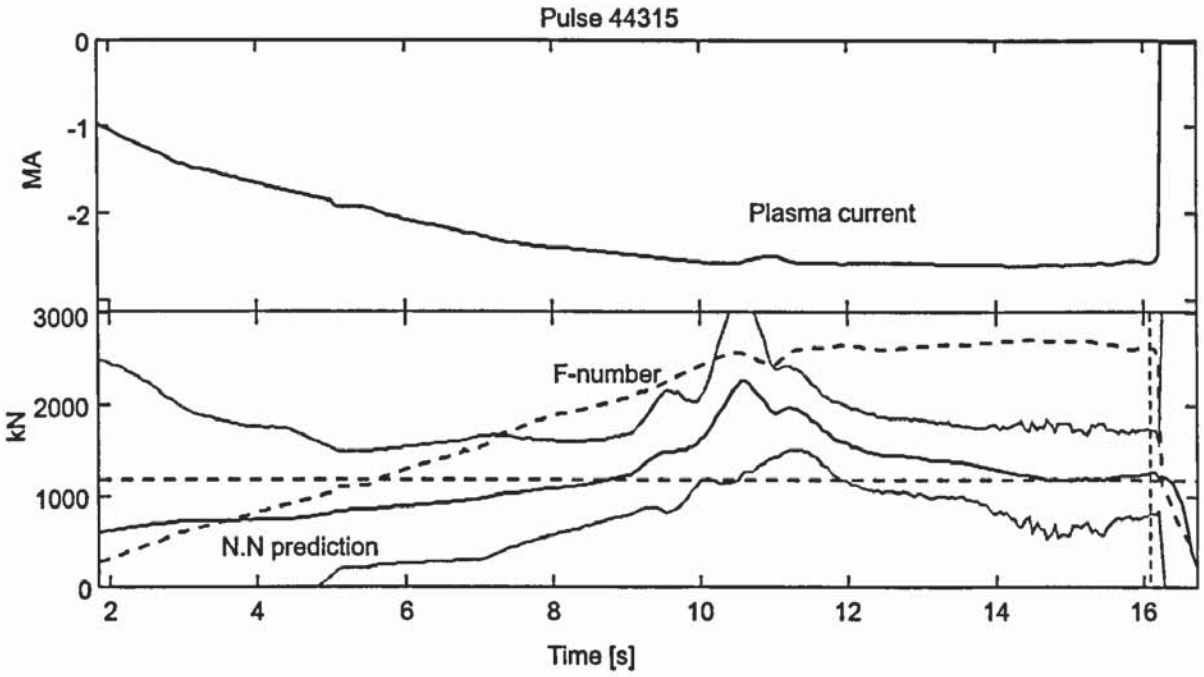

Figure 8.7: Vertical forces prediction for pulse 44315. The horizontal dotted line is the value of the actual forces measured at the disruption (vertical dotted line). The network output (black line) performs better than the f-number (dotted line). Moreover the error bars (grey lines) give a clear idea about the confidence of the prediction. It is also possible to see how in the first period of the pulse the error is very large. This is mainly due to the lack of training data for that period of the pulse.

A different result has been obtained for pulse 44371 (Figure 8.8). The value of the forces predicted are close to the achieved, but the error bar is very large suggesting that the prediction could be unreliable. This is caused by the fact that the plasma configuration was new and therefore not available during the network training process.

If we compare the results with the f-number we can immediately see that the prediction is more accurate. Moreover the error bars curves, drawn as the network output $\pm 2\sigma_t$ (see Chapter 7.6), provides further information to the user. The upper error bar curve can be

interpreted as the maximum vertical forces that the plasma configuration that is currently running can exert, or it can provide a confidence signal for the network output.

For DPPS a combination of the two meanings could be used to set a protection that stops only the dangerous pulses. If the error bar is lower than 1000 kN and predicted forces are higher than 3000 kN than an alarm could be issued and actions to terminate the pulse could be requested to the control systems.

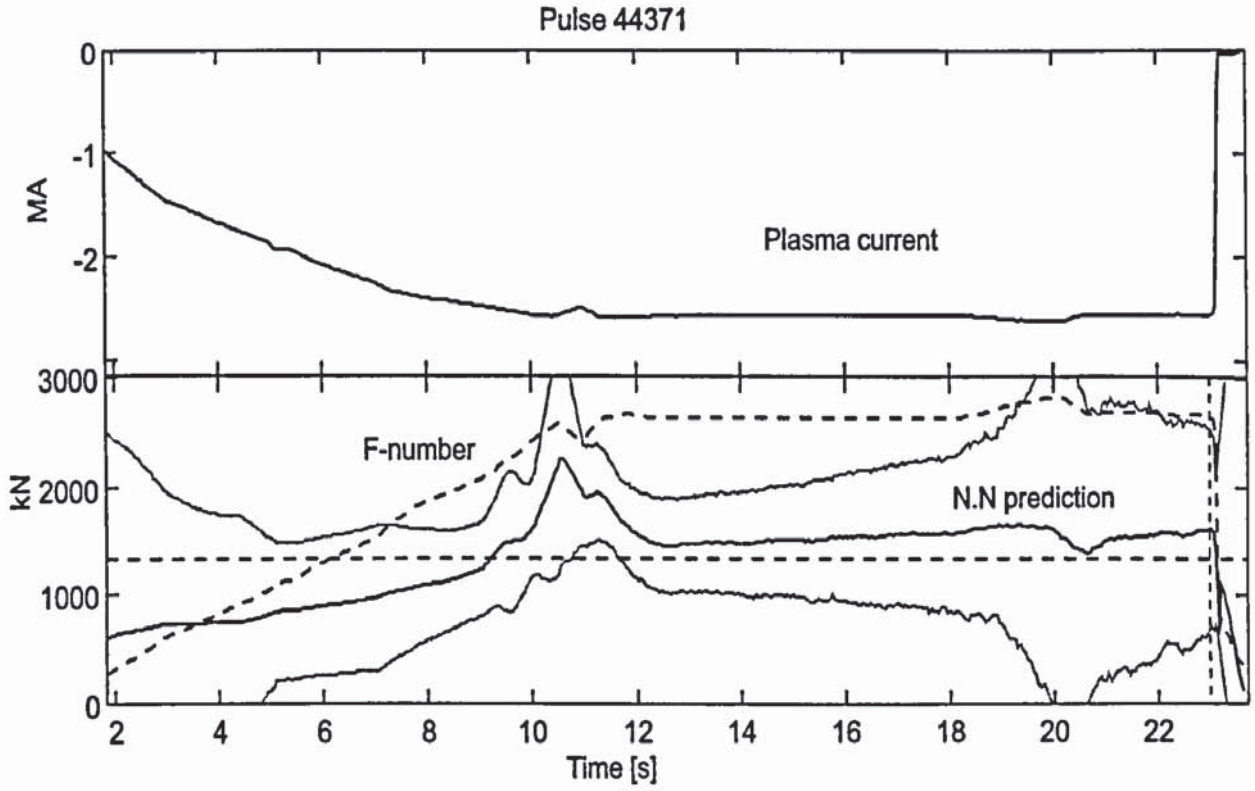

Figure 8.8: Vertical forces prediction for pulse 44371. Even if the network prediction (black line) is very close to the actual forces (horizontal dotted line), the error bars (grey lines) suggest to use the network output with caution. This further information can be used or to define the maximum forces that a plasma configuration could exert in case of disruption, or to give a confidence value to the network output.

8.4 Conclusions

The implementation of the neural networks in real time gave the possibility of gaining some experience on the performance of the networks.

Some problems immediately arose because the signals used to train the network are different from the signals available in real time. This can cause differences in the behaviour of the network as we have seen with the softmax network.

Nevertheless, it has been proved that network algorithms could be used to determine in real time plasma parameters that otherwise will not be possible to calculate because of computational power limitation.

Neural networks could also be used to improve the performance of the plasma protection systems, detecting earlier than the existent algorithms the occurrence of a disruption.

Also the prediction of vertical forces has been improved, and the addition of the error bars to the predicted value not only can be used to quantify the accuracy of the network output, but it can also be used to provide a maximum value of the vertical forces that plasma configuration can cause in case of a disruption.

9. Discussion and Conclusions

The aim of this work is to study if neural network techniques could be successfully used to estimate plasma parameters and to detect plasma disruptions in advance. This included the implementation of the Disruption Prediction and Prevention System to test in real time the performances of the algorithms studied.

The results obtained showed that the neural network algorithms can be used to determine the plasma internal parameters. The simplicity of the algorithms gives the possibility of determining plasma internal parameters in less than 4 ms with good accuracy.

Moreover, the safety of the machine could be improved, detecting in advance plasma disruptions using the classification capabilities of neural networks. The softmax MLP network and the novelty detection methods studied suggests that if used they can help to avoid disruptions or at least to minimise their effect by speeding up the response of the control systems.

Finally, the problem of qualifying the disruptions has been studied using a network that tries to predict the vertical forces generated in case of disruption. This is one of the most difficult tasks, because the information available is very poor and does not allow us to discriminate between vertical instability disruptions and radiative disruptions.

9.1 Plasma parameter estimation

In chapter 3 we have seen that multilayer perceptron networks could provide in real time the internal plasma parameters with a sufficient accuracy. The network designed uses 16 inputs generated from magnetic signals and then normalised. It consists of two layers: the hidden layer is made of 18 hyperbolic tangential sigmoids, and the output layer is made of 5 linear sigmoids providing the values of the internal plasma parameters ℓ_i, q_{95}, q_{wall}, q_{axis} and β_p. The training has been performed using 2100 patterns collected using the data generated during the Divertor Mark 2 campaign (1995-1997). The method for the training is the gradient descent with an adaptive learning rate. The training is stopped using a limit on the training epochs to reduce the training time to the minimum.

Comparisons with EFIT equilibrium code shows that we can expect errors less than 10% for the measurements of ℓ_i, q_{95}, q_{wall} and β_p. The precision of q at the axis is a bit more difficult to quantify because the EFIT estimation itself proved to be unreliable for certain plasma configurations.

This fact has been used to verify the network performance. The network showed a quite good agreement with EFIT even for those plasma configurations where it is known that EFIT results are wrong. This suggests that the network was able to learn the model used by EFIT to estimate these plasma parameters, but at the same time, it confirms that we can not expect that the network could developed a model closer to the reality than the model used to train it.

During the design of this network we have also seen the importance of having a set of input variables as robust as possible to avoid that the failure of a sensor used as input could lead to the malfunctioning of the network. To avoid this problem, the XLOC algorithm has been redesigned to provide, in real time, data that are robust to the malfunctioning of magnetic sensors. XLOC uses all the 79 magnetic sensors available, and it is now capable of determining the plasma boundary and its magnetic field in less than 2ms. XLOC has also the important function to reduce the number of network input variables using all the information available. In this way the network is fed with the position and the magnetic field of the plasma boundary determined in 7 different positions. This information more close to the plasma allows us to obtain the same accuracy in the plasma parameters estimation as a network that uses as input the direct measurement of 35 magnetic sensors.

In addition, in chapter 4, using Bayesian techniques, a new network has been designed. The aim was to test the possibility of determining a validation signal for the network outputs. The possibility of determining error bars to quantify the accuracy of the network outputs was exploited for this purpose. This, not only will give the possibility of discarding the network outputs when the error bar is to large but it will also help to overcome the scepticism of the machine operators against neural network algorithms, by showing that it is possible to discriminate if the output of the network is good or not.

9.2 Plasma configuration classification for disruption prediction

In chapter 5 and chapter 6 the possibility of using neural network classification techniques has been exploited to predict plasma disruptions. Two different techniques were used. The softmax MLP network is used to distinguish between unstable and stable plasma patterns. The advantage of this method is that the network is trained with disrupted pulses and therefore the past experience is used to detect the precursors that brought to a disruption. This gives the possibility of using the softmax network to avoid repeating the same conditions that led to a disruption. Unfortunately, the need of a large number of disrupted pulses to build the training set can pose serious problems to the design of this network. At JET we can rely on a disruption database of 360 pulses, but if the same method has to be applied to new machines the database could be even more limited or even not existing in the case of new machines.

The results obtained with this network suggest that disruptions could be predicted in advance giving precious time to the control systems to react and try to stop the pulse or at least to minimise the consequences of the disruption itself. We have also seen how the decision boundary can be used to determine the response of the DPPS system. If the decision boundary is close to one, with regard to the disruption probability, then the network will issue an alarm only for input patterns similar to those that generated a disruption in the past. In this way the capability of predicting disruptions is reduced but at the same time there will be no doubt of malfunctioning of the network. In this way, confidence in the prediction can be gained demonstrating that the system is capable of recognising plasma patterns that led to a

disruption. It is in fact very difficult to prove that the network worked correctly if no evident signs of disruption precursors can be seen after the issuing of an alarm.

The second technique estimates the density distribution of the successful pulses using a semi-parametric algorithm. The method discussed in chapter 6 is based on the representation of the density distribution by a sum of 28 Gaussian distributions. The parameters of these distributions and their relative weights are determined using the Expectation Maximisation method, using a set of 1400 stable plasma patterns. The distinction between successful and disrupting plasma patterns is done using a threshold on the probability density. If the probability value of a given pattern is lower than this threshold then the pattern is flagged as outside the space of the stable plasma patterns. This imply that the network must be trained with a set of patterns that comprises all the possible successful plasma pulses that JET can run. Any new plasma configuration that presents a pattern that the network can not recognise, will be flagged as disrupting even if the plasma is perfectly stable. Although from a mathematical point of view this could be considered the right behaviour because the pattern was novel, it is not acceptable from JET machine operation point of view. The limitation of the operational space could become too strong precluding the possibility testing new plasma configurations. Unfortunately in an experimental machine as JET the balance between the requirements of having a network capable of recognising disrupting pulses but at the same time capable of giving the possibility of exploiting the machine capabilities is very difficult to obtain.

Quantifying the dangerousness of a plasma configuration in case of disruption could help in deciding whether to stop a pulse or not. This is why an attempt to measure the forces that a plasma disruption can exert has been made. In chapter 7 we have seen the difficulty of calculating these forces using a multilayer perceptron network. The major obstacle in the design of this network is the fact that the forces generated by the disruptions depends also on how the disruption itself develops. The same configuration can generate very high forces if the disruption is a pure vertical instability, and very low if all the plasma energy is released by radiation. However, using the small database available and determining also the standard deviation of the vessel forces prediction, it is possible to provide a measurement of the vessel forces that a plasma configuration can generate, with an accuracy that the traditional methods can not achieve.

This could be a further information that combined with the algorithms used to predict the disruption can help to make acceptable the possibility of loosing some operational space. Using the two information together we can discriminate between dangerous disruptions and disruptions that do not threaten the integrity of the machine. The disruption detection can therefore be triggered immediately if the predicted forces are very high. On the contrary if the predicted forces are small, we can risk to wait for higher level of disruption probability, reducing in this way the possibility of stopping good pulses. In this way, a further reduction of the good pulses that could be stopped will give more credibility to the whole system.

As regard to the percentage of the disrupted pulses that can be avoided using these methods, we have seen that if we want to avoid restricting the operational space considerably, the success of the prediction of a disruption 100 ms before it occurs, is down to 50%.

Even with major improvements it will be very difficult for the system to become 100% successful in detecting disruptions sufficiently in advance that the control system will be able to avoid it. This would mean that a protection system based on these algorithms can not be expected to be a primary protection system. More traditional protections capable of handling the situations where the disruption protection system fails must be considered. This involves the design of protections for over-currents in the coils and in the current amplifiers as well as mechanical protection of all the machine components that could be damaged by a disruption.

9.3 Future development

Up to now only JET data have been used to design the disruption prediction networks, and the networks themselves are specifically designed for JET. After the preliminary successful results the question if the same networks can be used in different machines has been raised. The possibility of designing disruption prediction networks that uses non-dimensional inputs is now under study, to see if they can be used on different machines. This will give the possibility of using existing disruption datasets coming from different tokamaks to design disruption prediction systems for the tokamaks of the future.

An upgrade of the DPPS systems is also under development. This new upgrade will include a first attempt to automatically retrain the DPPS disruption prediction networks. The

computational power of the hardware selected should give the possibility of storing the data of the plasmas disrupted during the week. The data will be included into the datasets used to design the disruption network and finally the network will be retrained and validated. A comparison with the old network results will be executed to see if the performance has been improved before the new network will become operational.

With regard to the capability of estimating plasma parameters, it would be useful to the control systems for example to know the plasma inductance 50~100 ms in advance so that very tight control of the plasma current profile could be done. The design of an MLP network that uses as inputs the derivative of the same inputs used in the network described in chapter 3 has started. Also the poloidal field coil current reference waveforms have been used, being available through out the pulse, they can be used to know in advance the plasma configuration. The training has been performed using as target set the plasma parameters acquired 50 ms after the acquisition of the input patterns. A problem that still needs to be solved is that the coil current reference waveforms does not always exist. The operator can chose from a set of control variables and some of them are mutually exclusive. Therefore for some period of the pulse the current waveforms are not available making their use impossible. For those periods the waveforms are substituted using a linear extrapolations of the currents, knowing their values and derivative during the whole pulse. Preliminary results showed that there is a possibility of predicting the plasma current profile, if the additional heating is not used. The use of additional heating results in sudden changes in the current profile that cannot be predicted by the network. Further studies would involve the inclusion of the additional heating power in the input variables.

Another application under consideration is the generation of a network capable of classifying the various types of disruptions. This can not be done if only the data of the last campaign is used because they are too few to allow the generation of datasets sufficiently large to train a network successfully. The project involves the inclusion of the disruption database of the previous JET campaigns, including also the pre-divertor campaign. Then, enough data should be available to train a network capable at least of distinguishing between vertical instability, density limit and a low q disruptions. This will also give the possibility of studying if a machine independent network can be designed.

Other future applications of neural networks under considerations are:

- the use of density distribution determination algorithms (as novelty detection) to redefine the ℓ_i-q_ψ diagram operational space. Some test has been performed using novelty detection schemes with the full set of disruptions without a big success. I anyway believe that considering just the disruptions that are related to ℓ_i and q_ψ it will be possible to redefine the boundary increasing the reliability of the ℓ_i-q_ψ diagram.
- the use neural network techniques to detect faulty magnetic sensors, using the information provided by the poloidal field coil currents. This will improve the safety of the machine avoiding that wrong measurements are fed to the XLOC algorithm and as a consequence used by the plasma shape and position control.

It is clear that in a field as plasma fusion where the physics of the events is not yet well understood, and where there is a large amount of data that cannot yet be processed with more traditional techniques the use of neural networks can expand considerably. This will give the possibility of performing new experiments and understand better the phenomena that nowadays are not yet fully understood. Nevertheless, I personally hope that in the far future the better understanding of the plasma physics will reduce the use of neural networks for the prediction of disruptions, leaving more space to control systems that can always operate the machine in the safe region reducing the probability of disruption to the minimum.

10. References / Bibliography

[1] Abramowitz, M. et al. (1964) Handbook of Mathematical functions, Dover Publications, New York

[2] Albanese R. et al. (1988) Analysis of vertical instabilities in the JET experiment, Fusion Technology,

[3] Atkinson, K.E. (1989) An introduction to numerical analysis, ed. John Wiley & Sons

[4] Bertolini, E. et al. (1987) The JET magnet power supplies and plasma control systems, Fusion Technology vol.11, no.1, 71-119

[5] Bishop, C.M. (1994) Novelty Detection and Neural Network validation. IEEE Proc-Vis. Image Signal Process., vol 141, No. 4.

[6] Bishop, C.M. (1995) Neural Networks for Pattern Recognition. Clarendon Press, Oxford.

[7] Bishop, C.M. (1995) Regularisation and Complexity control in feed-forward Networks, EC2 & Cie. Invited paper. pp 141-148.

[8] Bishop, C.M. (1996) Neural Networks: a pattern recognition perspective, In *Handbook of Neural Computation*. Editor: E. Fiesler and R. Beale. Oxford University Press and IOP Publishing.

[9] Buzio M. (1992) Structural effects of disruptions on the vacuum vessel of the Joint European Torus Tokamak, Tesi di laurea.

[10] Cariolaro, G. et al. (1988) Teoria della probabilita' e dei processi aleatori, Patron editore, Bologna

[11] Caudill, M. (1993) Understanding neural networks, vol.1-2, The MIT press

[12] Chen F.F. (1984) Plasma Physics and controlled fusion, Plenum Press, New York

[13] Christiansen, J.P. (1987) Integrated analysis of data from JET, Journal of computational physics, vol.73, 85-106

[14] Dempster, A.P. et al. (1977) Maximum likelihood from incomplete datavia the EM algorithm. Journal of the Royal Statistical society, B 39 (1),1-38

[15] Demuth, H. (1993) Neural network toolbox, The Mathworks Inc.

[16] Ellis, J.J. et al.(1994) The rapid determination of plasma equilibrium parameters at JET from external magnetic measurements, presented at the IAEA Technical committee meeting on magnetic diagnostics for fusion plasmas, Kharkov, Ukraine, October 5-7 1994

[17] Ferron, K.E. et al (1997) Real time equilibrium reconstruction for tokamak discharge control 24th European Physical Society Conference on Controlled Fusion and Plasma Physics, Berchtesgaden, Germany

[18] Garribba, M. (1994) First operational experience with the new plasma position and current control system at JET, Proceedings 18th symposium on fusion technology, SOFT-18

[19] Garribba, M. (1995) Controllo multivariabile del plasma nell'esperimento JET, Tesi di Dottorato, Universita' di Padova.

[20] Garribba, M. et al. (1989) Analysis of Disruptions in case of large vessel displacement, JET Memorandum, 28 Jul. 1989

[21] Gross, R. A. (1984) Fusion Energy. John Wiley & Sons, New York.

[22] Hertz J. et al. (1991) Introduction to the Theory of Neural Computation, Addison-Wesley

[23] Huguet, M. et al. (1987) The JET machine: design, construction, and operation of the major systems, Fusion Technology vol.11, no.1, 43-70

[24] JET Annual Report 1996

[25] JET Progress Report 1996

[26] Khanna, T. (1991) Fondamenti di reti neurali, Addison-Wesley

[27] Kraus, J.D. (1984) Electromagnetics, McGraw-Hill International Editions

[28] Lao, L.L. et al. (1985) Reconstruction of current profile parameters and plasma shapes in tokamaks, Nuclear Fusion, vol.25, No.11, 1611-1622

[29] Lao, L.L. et al. (1985) Separation of β_p and ℓ_i in tokamaks of non circular cross section, Nuclear Fusion, vol.25, No.10, 1421-1436

[30] MacKay, D.J.C. (1992) The evidence framework applied to classification networks. Neural Computation 4 (5), 720-736

[31] MacKay, D.J.C. (1995) Bayesian Methods for Neural Networks: Theory and Applications. Papers from the Neural Networks Summer School of Cambridge University

[32] MacKay, D.J.C. (1992) A practical Bayesian framework for backpropagation networks. Neural Computation 4 (3), 590-604

[33] Mast, K.F. (1991) A low noise highly integrated bolometer array for absolute measurement of VUV and soft x radiation, Rev. Sci. Intrum., vol.62, no.3, 744-750

[34] Mast, K.F. et al. (1991) Bolometric diagnostics in JET, Rev. Sci. Intrum., vol.56, no.5, 969-971

[35] Mazzetti, A. (1991) Reti neurali artificiali, Apogeo Informatica

[36] Mukhovatov, V.S. et al. (1971) Plasma equilibrium in a tokamak, Nuclear Fusion, vol.11, 605-633

[37] Nave, M.F.F. et al. (1990) Mode locking in tokamaks, JET internal report P(90)06

[38] Neal, R.M. (1994) Bayesian learning for neural networks, Ph.D. thesis, University of Toronto,Canada

[39] Noll, P. (1988) Forces on the JET vacuum vessel during disruptions and consequent operational limits, papers presented at the 8th topical meeting on technology of fusion, Salt Lake City, Utah, JET Report JET-P(88)43

[40] O'Brien, D.P. et al. (1992) Equilibrium analysis of iron core tokamaks using a full domain method, Nuclear Fusion, vol.29, No.4, 1351-1360

[41] Picton, P. (1994) Introduction to neural networks, McMillan

[42] Press, W.H. (1988) Numerical Recipes in C, Cambridge university press

[43] Puppin, S. (1995) Sviluppo del controllo della forma del plasma nell'esperimento JET, Tesi di Dottorato, Universita' di Padova.

[44] Reichle, R. (1997) JET Memorandum to the Engineers in charge Session Leaders and Plasma Duty Officers.

[45] Reichle, R. et al (1995) Divertor radiation in JET's Mark I Divertor, Contr. 22nd EPS on Contr. Fusion and Plasma Physics, Bournemouth, 3-7 July 1995

[46] Ripley, B.D. (1996) Pattern Recognition and Neural Networks, Cambridge University Press.

[47] S.Ali Arshad (1994) Internal Memorandum on Magnetic coils for the new JET divertor configuration.

[48] Saibene, G. et al. (1973) The plasma fault protection system: disruption control in JET. Proc. of IAEA Tech. Meeting on the Avoidance and Control of Tokamak Disruptions, Culham (UK), p.1, Sept 1991

[49] Schuller, F.C. (1995) Disruptions in tokamaks, Invited papers from the 22nd European Physical Society Conference on Controlled Fusion and Plasma Physics, Bornemouth

[50] Shafranov, V.D. (1971) Determination of the parameters β_p and ℓ_i in a tokamak for arbitrary shape of plasma pinch cross-section, Plasma physics, vol. 13, 757-762

[51] Sips, A.C.C. et al. (1997) Pre-print of a paper to be submitted for publications in Plasma Physics and Controlled Fusion. Also JET internal print JET-P(97)29

[52] Smythe, W.R. (1969) Static and Dynamic Electricity 3rd. Ed., McGraw-Hill, New York

[53] Sontag, E.D. (1992) Neural Networks for Control

[54] Tanga A. et. al. (1988) Disruptions in JET, JET Report JET-P(88)44

[55] Tanga, A. et al. (1991) Study of plasma disruptions in JET and its implications on engineering requirements, Proc. of 14th IEEE Int. Symp. Fusion Eng., San Diego (California, USA)

[56] Thodberg H.H. (1996)A review of Bayesian neural networks with an application to near infrared spectroscopy, IEEE Transactions on neural networks, vol 7, no.1,56-72

[57] Thompson J.L. (1986) Contents and construction of the jet disruption database, JET Report JET-P(91)08

[58] Tonetti, G. et al.(1986) Measurement of the energy content of the JET tokamak plasma with a diamagnetic loop, Rev. Sci. Instrum., vol.57, no.8, 2087-2089

[59] Vogl, T.P. (1988) Accelerating the convergence of the Back-Propagation Method, Biological Cybernetics, vol. 59 pag. 257-263.

[60] Wesson, J. A. (1987) Tokamaks. Clarendon Press, Oxford.

[61] Wesson, J.A. (1978) Plasma equilibrium in a tokamak, Nuclear Fusion, vol.18, no.1 87-132

[62] Wesson, J.A. et al. (1989) Disruptions in JET, Nuclear Fusion, vol.29, No.4, 641-665

[63] Wroblewski, D. et al. (1997) Tokamak disruption alarm based on neural network model of the high b limit, Nuclear Fusion, vol. 37, no.6, 725-741

[64] Zacharov, L.E. et al. (1973) Equilibrium of a toroidal plasma with noncircular cross-section, Sov. Phys. Tech. Phys., vol 18, no. 2, 151-156

[65] Zwingman, W. (1997) Pre-print of a paper to be submitted for publications of the 7th European Fusion Theory Conference. Also JET internal print JET-P(97)57

[66] Zwingmann, W. et al. (1996) Current profiles in shear optimised plasmas, JET internal presentation