Neural network tool for rapid recovery of plasma topology

V. Tribaldos and B. Ph. van Milligen
Asociación EURATOM-CIEMAT para Fusión, 28040 Madrid, Spain

(Presented on 15 May 1996)

A general method for the rapid recovery of plasma topology based on a neural network fit of the normalized magnetic flux is presented. We propose a general method for coordinate inversions that is based on a neural network fit of the normalized magnetic flux. The neural network provides a flexible and compact base for representing the plasma topology and allows the evaluation of spatial derivatives by analytic methods (as opposed to finite-difference methods), making it faster than other techniques. We present examples of this technique for both two-dimensional plasmas (tokamak D shaped, X point) and stellarators.
[S0034-6748(97)64701-2]

I. INTRODUCTION

The evolution of plasma topology in tokamak physics has been towards complexity; shapes have developed from simple circular axisymmetric tori to D-shaped plasmas with X points and large Shafranov shifts due to high β. In stellarators, shapes have always been complex. The high degree of complexity implies that the interpretation of diagnostic signals and, more generally, the link between theory (that often calculates in flux coordinates) and experiment involves costly transformations from magnetic or flux coordinates to real space coordinates.

This transformation is usually performed by interpolating the plasma normalized magnetic flux on a predefined three-dimensional grid. This approximation can become quite cumbersome for complex three-dimensional topologies, since large data sets are needed to achieve good accuracy, and the interpolation scheme is time consuming. Thus, for example, ray-tracing codes may spend a major part of their time in coordinate inversions. Further, spatial derivatives derived from such an interpolation based on a finite spatial mesh often show oscillating behavior unless very fine grids are used. We propose a general method for coordinate inversions that is based on a neural network fit of the normalized magnetic flux.

The article is organized as follows. In Sec. II the use of neural networks as fitting tools is described; examples of the method for a D-shaped tokamak with an X point and two stellarators are presented in Sec. III; finally, in Sec. IV the results are discussed and general conclusions are drawn.

II. USING NEURAL NETWORKS AS FITTING TOOLS

The major problem of fitting when dealing with complicated data is the necessity of using a fixed functional dependence on a certain number of free parameters. The result of the fitting procedure depends strongly on how good the estimation of this functional dependence is. Thus when the base of functions used departs from true dependence, the fitting methods need a lot of free parameters and usually gives poor results. It is therefore desirable to find an adaptive base of functions for fitting proposes. The procedure we have used to fit the normalized magnetic flux is based on the use of a neural network[1] as a flexible representation instead of a base of (fixed) functions. This is because neural networks[2,3] are capable of approximating, with arbitrary accuracy, any nonlinear multivariate mapping for a sufficiently large number of hidden units.

Here we have considered the most widely used neural network, known as multilayer perceptron (MLP), with only one hidden layer, MLP1, shown schematically in Fig. 1. The inputs x_m $(m=1,\ldots,M)$ and the outputs y_n $(n=1,\ldots,N)$ of this network are related through a set of coefficients $\{\theta_n^0, W_{nh}^0, \theta_h^1, W_{hm}^1\}$ $(h=1,\ldots,H)$, usually known as weights, by

$$y_n(x_1,\ldots,x_M)=\theta_n^0+\sum_{h=1}^{H} w_{nh}^0\sigma\left(\sum_{m=1}^{M} w_{hm}^1 x_m+\theta_h^1\right), \quad (1)$$

where σ is the sigmoid function $\sigma(x)\equiv 1/(1+e^{-x})$.

A numerical code was developed for the MLP1 neural network with the coordinates $\mathbf{r}$ as inputs and the normalized magnetic flux ψ as output. The weights of the neural network are calculated minimizing the sum of squares of the difference between the network output and the training values through the penalty function E:

$$E=\sum_{k=1}^{K} [\psi_{\text{net}}-\psi_{\text{train}}]^2. \quad (2)$$

Minimization of Eq. (2) provides a good description of ψ when (a) the number of hidden nodes H is sufficient, (b) the training data set $\{\mathbf{r},\psi\}$ is well distributed so that ψ is well described, and (c) the number of training points K is much larger than the number of weights $(M+1)H+(H+1)N$, to avoid overfitting. For the examples discussed in this work the training data sets consist of 16 384 random samples, covering a toroidal cross section (tokamak) or one half of a period (stellarators), and the maximum number of hidden nodes used was 255. The minimization procedure used was the standard quasi-Newton gradient descent algorithm and, although the computation is very quick, an effort has been made to parallelize the code, and a degree of parallelism of more than 90% was achieved.

An advantage of a representation based on neural networks is that spatial derivatives can be calculated analytically from the trained network. This is especially useful when the neural network is used for fitting the normalized magnetic flux since the density and temperature gradients can be computed analytically, in any direction.

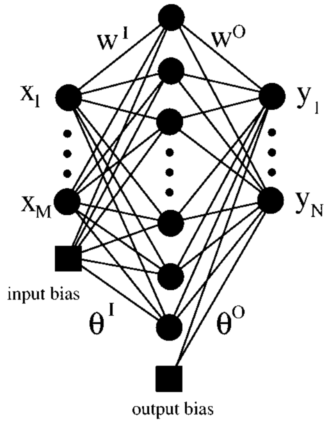


FIG. 1. Schematic view of a MLP1 neural network.

III. EXAMPLES OF THE METHOD

The normalized magnetic flux ψ determines the magnetic surfaces in fusion devices, and is usually computed using equilibrium codes. These codes often make their calculations in magnetic or flux coordinates and except for simple plasma geometries the transformation from these coordinates to real space coordinates is very complicated. Usually this transformation is performed by interpolating the magnetic flux on a predefined grid. The main shortcomings of this method are that large data sets need to be stored for each plasma configuration and that the method is CPU time consuming. Moreover, the computation of derivatives can show oscillating behavior. The advantages of using neural networks for fitting this dependence versus interpolation methods are (a) only small data sets containing the weights of the trained network need to be stored (using three coordinates as inputs and ψ as output the number of coefficients is $5\,H+1$); (b) computing the network output is inexpensive; (c) the derivatives can be calculated analytically and show nonoscillating behavior; (d) the weights for one plasma configuration can be used as input values to recalculate new weights for other configurations; and (e) since no particular topology of the plasma is supposed (nested flux surfaces), the method can incorporate X points or islands. Here we will show the capabilities of the neural network as a fitting tool in determining the normalized magnetic flux for two different types of fusion devices, namely, a tokamak with an X point and two stellarators.

A. Tokamak

The plasma we have considered as an example of a tokamak with an X point is an ITER-like[4] plasma with a major radius $R_0=8$ m and a minor radius $a=3$ m. A training set consisting of 16 384 random samples $\{R,z,\psi\}$, generated from an equilibrium code, uniformly distributed over a toroidal section of the vacuum chamber was used to train a neural network with two inputs, one output, and 31 hidden nodes (thus with 125 coefficients). The results of the pretrained neural network are checked versus a new data set consisting of $N_{\text{test}}=4096$ random samples not contained in the training set. The root mean squared (rms) error defined as

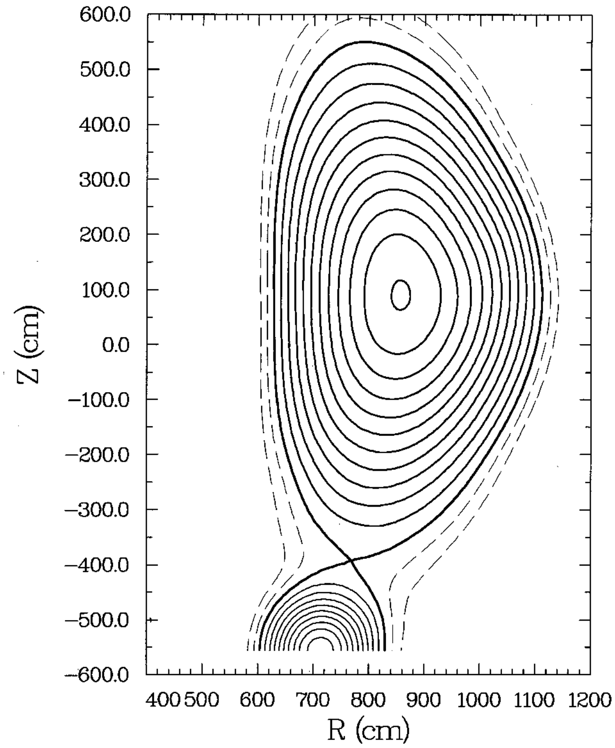


FIG. 2. Curves of the constant magnetic flux computed with the neural network for an ITER-like plasma with an X point.

$$\epsilon_{\text{rms}}=\sqrt{\frac{1}{N_{\text{test}}}\sum_{n=1}^{N_{\text{test}}}(\psi_n^{\text{net}}-\psi_n^{\text{test}})^2}$$

is found to be 0.3% after 2000 iterations of the code. From the pretrained neural network it is possible to compute, at any point, the normalized magnetic flux. The magnetic surfaces obtained as curves of constant ψ, shown in Fig. 2, are indistinguishable from those obtained from the original equilibrium.

B. Stellarator

As examples of stellarators we have used TJ-I U[5] and TJ-II.[6] TJ-I U is an $l=1$ $m=6$ torsatron constructed at CIEMAT, Madrid, with a major radius $R_0=0.6$ m and an average plasma radius $a=0.1$ m. Here we have used a MLP1 with three input units (x,y,z), one output unit (ψ), and 63 hidden nodes, i.e., 316 coefficients. The neural network was trained using a data set consisting in 16 384 random examples $\{x,y,z,\psi\}$, generated using the output of the VMEC code for vacuum conditions covering one-twelfth of TJ-I U. This is sufficient because of the six periods of the device and the stellarator symmetry of TJ-I U plasma, i.e., $\psi(R,\varphi,z)=\psi(R,\varphi+\pi/2,z)$ and $\psi(R,\varphi,z)=\psi(R,-\varphi,-z)$. Although TJ-I U can access a certain range of plasma configurations, we will only show results for the standard configuration. Using random weights after 2000 iterations of the code the rms error found was 0.8%. Curves of constant ψ are shown in Fig. 3 for different toroidal angles. Similar results are obtained for other configurations.

TJ-II is a four-period midsize helical axis stellarator under final assembly at CIEMAT that has a major radius

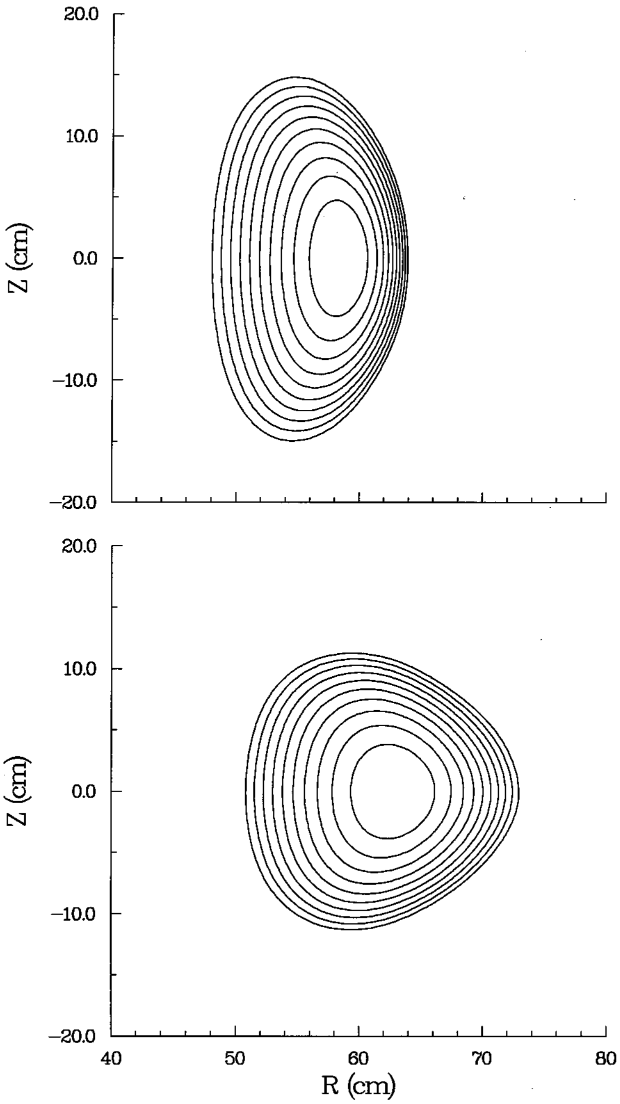


FIG. 3. Curves of the constant magnetic flux computed with the neural network for the standard configuration of TJ-I U for $\varphi=0°$ and 30°.

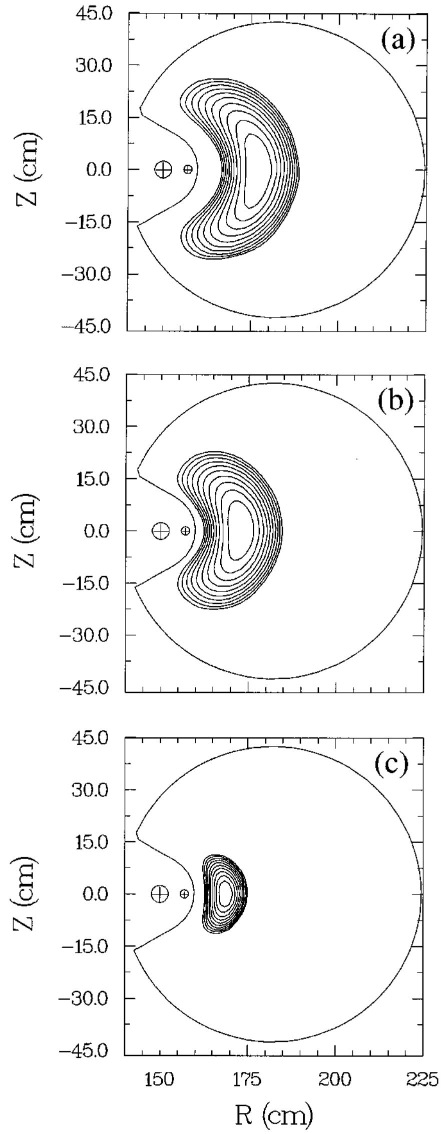


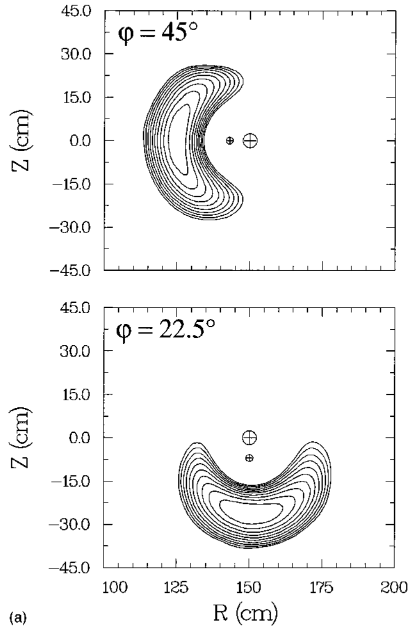
FIG. 4. Toroidal section of TJ-II plasma at $\varphi=0°$ for several configurations.

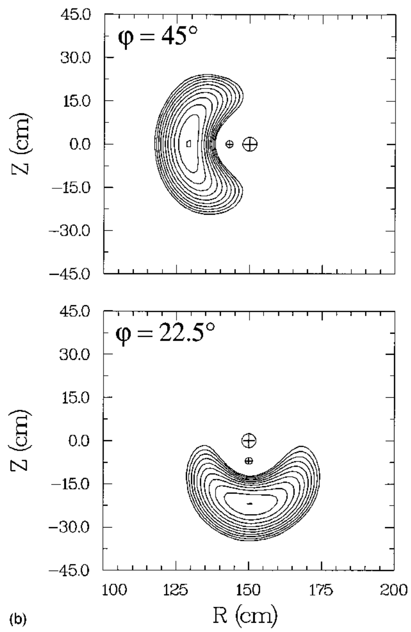
$R_0=1.5$ m, a bean-shaped plasma cross section with an average minor radius between 0.1 and 0.25 m, and a nominal toroidal field of $B_0=1$ T. An attractive property of TJ-II is its flexibility that allows a wide range of configurations having different plasma positions, shapes, and sizes; see Fig. 4. The neural network used for this case is the same as TJ-I U but with 255 hidden nodes, i.e., with 1276 free parameters. It was also trained using a data set consisting of 16 384 random examples $\{x,y,z,\psi\}$, generated using the output of the VMEC code for vacuum conditions covering an octant of TJ-II once again because of the four periods of TJ-II and its stellarator symmetry. In Figs. 5(a)–5(c) level curves of the constant magnetic flux are shown for three plasma configurations of TJ-II. To compute the network coefficients for the plasma presented in Fig. 5(a) we have started with random weights, and found a rms error of 0.5% after 20 000 iterations of the code. To fit the other two configurations we have started the calculations using these coefficients as input values and found, in only 5000 iterations, rms errors of 0.3% and 0.2% for the cases presented in Figs. 5(b) and 5(c), respectively.

Although the structure of TJ-II plasma is completely three dimensional, the network can successfully reproduce its topology with only 1276 coefficients. A comparison between these 1276 coefficients and the 9108 coefficients needed in the VMEC representation (276 Fourier modes for each of 33 magnetic surfaces) shows that the network representation is much more compact.

Moreover, since the network was trained using the magnetic flux, defined only inside the plasma, the output of the network for a point **r** outside the plasma is always larger than 1. In this sense the network not only describes the plasma accurately but also can be used to test if a point is outside the last closed magnetic surface.

Finally we would like to point out that the results of this work can be readily extended to finite beta conditions. A network of this type can, in addition, also be used to solve the ideal magnetohydrodynamic plasma equilibrium.[8]





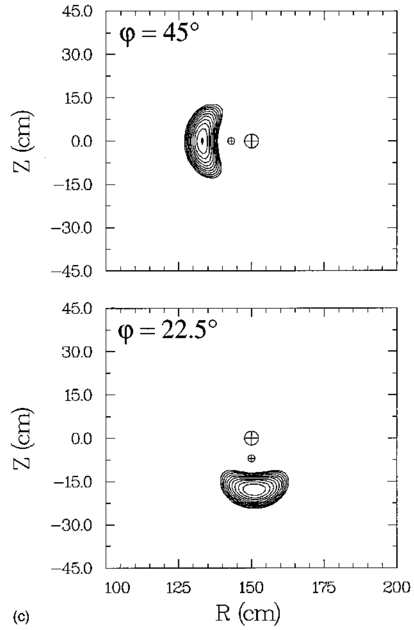


FIG. 5. (a) Curves of the constant magnetic flux computed with the neural network for configuration of Fig. 4 of TJ-II for $\varphi=22.5°$ and 45°. (b) The same as in (a) but for case (b) of Fig. 4. (c) The same as in (a) but for case (c) of Fig. 4.

IV. CONCLUSIONS

We have presented a new method for representing the magnetic flux in common fusion devices based on a neural network fit. The procedure presented here overcomes the usual difficulties related with the inversion from magnetic or flux coordinates to real space coordinates, and especially those related with interpolation schemes. Its main advantage is the radical reduction of CPU time needed for performing the coordinate transformations once the net is trained. The network weights of the trained network are stored for each plasma configuration needed, and the storage space needed is small due to the fact that the network provides a very compact representation.

This method has been shown to give an accurate description of plasma topology for D-shaped tokamak plasma with an *X* point and also for the complex three-dimensional structure of stellarator plasmas.

Although in this work the neural network was used to fit the normalized magnetic flux the method, it can, without modification, be extended to fit any other quantity.

ACKNOWLEDGMENTS

The authors would like to thank A. Lopez-Fraguas and J. A. Jimenez for their equilibrium calculations.

[1] P. D. Wasserman, *Advanced Methods in Neural Computing* (Van Nostrand-Reinhold, Princeton, NJ, 1993).
[2] K. Hornick *et al.*, Neur. Networks **2**, 359, (1989).
[3] K. Hornick, Neur. Networks **4**, 251 (1991).
[4] D. V. Bartlett, JET Joint Undertaking (personal communication, 1995).
[5] E. Ascasibar *et al.*, Trans. Fusion Technol. **27**, 198, (1995).
[6] C. Alejaldre *et al.*, Fusion Technol. **17**, 131, (1990).
[7] V. Tribaldos and B. Ph. van Milligen, Nucl. Fusion **36**, 283 (1996).
[8] B. Ph van Milligen, V. Tribaldos, and J. A. Jimenez, Phys. Rev. Lett. **75**, 3594 (1995).