Nuclear Instruments and Methods in Physics Research A327 (1993) 580–593
North-Holland

NUCLEAR
INSTRUMENTS
& METHODS
IN PHYSICS
RESEARCH
Section A

Analysis of multiphase flows using dual-energy gamma densitometry and neural networks

C.M. Bishop [a] and G.D. James [b]

[a] *Neural Networks Group, AEA Technology, Harwell Laboratory Oxfordshire, OX11 0RA, UK*
[b] *Harwell Instruments, AEA Technology, Harwell Laboratory, Oxfordshire, OX11 0RA, UK*

Received 5 August 1992 and in revised form 6 November 1992

Dual-energy gamma densitometry offers a powerful technique for the non-intrusive analysis of multiphase flows. By employing multiple beam lines, information on the phase configuration can be obtained. Once the configuration is known, it then becomes possible in principle to determine the phase fractions. In practice, however, the extraction of the phase fractions from the densitometer data is complicated by the wide variety of phase configurations which can arise, and by the considerable difficulties of modelling multiphase flows. In this paper we show that neural network techniques provide a powerful approach to the analysis of data from dual-energy gamma densitometers, allowing both the phase configuration and the phase fractions to be determined with high accuracy, whilst avoiding the uncertainties associated with modelling. The technique is well suited to the determination of oil, water and gas fractions in multiphase oil pipelines. Results from linear and non-linear network models are compared, and a new technique for validating the network output is described.

1. Introduction

The accurate mass-flow metering of oil well products is of great importance to the oil industry. The high cost of sub-sea production has led to the use of multiphase pipelines to transfer mixtures of oil, water and gas, sometimes derived from more than one field. This provides a requirement for multiphase metering in which the flow rates of oil, water and gas can be determined with sufficient accuracy for reservoir management, for monitoring the withdrawal of fluids from a reservoir, and for custody transfer purposes. Ideally the measurements should be carried out non-intrusively. To this end, nuclear techniques, notably neutron interrogation and activation, and gamma densitometry and cross correlation, have an important part to play. It is unlikely, however, that any one technique will provide all the information required for the mass-flow metering of all three components. Restricting our attention to nuclear based techniques, we note that neutron interrogation can provide phase fraction measurements, and oxygen activation by pulsed fast neutrons can provide separate phase velocities for phases which contain oxygen [1]. Dual-energy gamma densitometry (the subject of this paper) gives the oil, water and gas fractions along the gamma beam path, and this information can be translated into phase fractions for known three phase configurations [2,3]. The cross correlation of single energy densitometry data has been applied to provide velocity measurements for two phase liquid-gas systems [4]. Computer modelling studies of the cross correlation of two dual energy densitometry measurements indicated that the velocity of the liquids in a three phase system can be obtained independently of the gas phase velocity [5].

In this paper we present the results of an investigation in which neural network analysis is used to derive the phase configuration, and thus oil, water and gas phase volume fractions, from simulated data for a six beam dual energy densitometer. It is shown that neural network techniques provide a powerful and practical solution to the problem of interpreting gamma densitometer data which can readily be applied in the operational environment. The network can learn the required non-linear transformations needed to extract phase fractions from the densitometer data by using a set of examples, thereby avoiding the need for complex and uncertain modelling of the 3-phase flow. In fact, as we shall show, there is no need to determine the phase configuration at all.

In section 2 we give an overview of the dual-energy gamma densitometry system and its application to oil pipeline monitoring. An introductory overview of neural networks is given in section 3, followed in section 4 by a discussion of the dataset used to train and test the networks. A detailed account of the mathematical derivations used to define the data set are given in the appendix.

The first problem addressed is that of using the gamma densitometry information to determine the ge-

Elsevier Science Publishers B.V.

ometrical configuration of the multiphase flow, and this is discussed in section 5. We show that the neural network is able to distinguish several different configurations with high reliability. In section 6 we then exploit neural networks to determine the oil fraction directly from the gamma densitometry measurements. Note that this technique does not itself require the phase configurations to have been determined.

An important practical consideration is the length of time for which the gamma signals have to be integrated, since this determines the overall accuracy of the system as well as the length of time needed to make a measurement. The effects of integration time on network performance are investigated in section 7. It is shown that good performance can be obtained with integration times of a few tens of seconds.

Another important issue concerns the robustness of the neural network solution. Since the neural network learns by example its performance may deteriorate if it is presented with input data which differs significantly from that used during training. While this problem can largely be avoided by careful selection of the training data set, it is important to have a means for verifying the phase fractions predicted by the network. In section 8 we demonstrate that a verification system can be designed which is able to detect novel input data with high reliability.

Finally, the significance of the results is discussed in section 9.

2. Dual-energy gamma densitometry

Here we give a brief overview of the technique of dual-energy gamma densitometry. We do not aim to give a complete treatment, but sufficient detail is given to make the paper self contained.

Gamma densitometry [2] makes use of the attenuation of a beam of gamma rays passing through matter. The degree of attenuation is dependent on the path length within the material, the nature of the material and the wavelength of the gamma rays. Since, for a given substance, the fraction of the beam attenuated per unit length is constant, over a finite distance the beam intensity will fall exponentially. We therefore write the intensity of a gamma beam (of given wavelength) after passing through a length x of material in the form

$$I = I_0\, e^{-\mu\rho x}, \tag{1}$$

where ρ is the mass density of the material, μ is the mass absorption coefficient of the material at the given wavelength, and I_0 is the intensity of the gamma beam in the absence of material. Since, for a given material, μ and ρ can be measured separately, determination of I allows the length x of material between the gamma source and the detector to be determined.

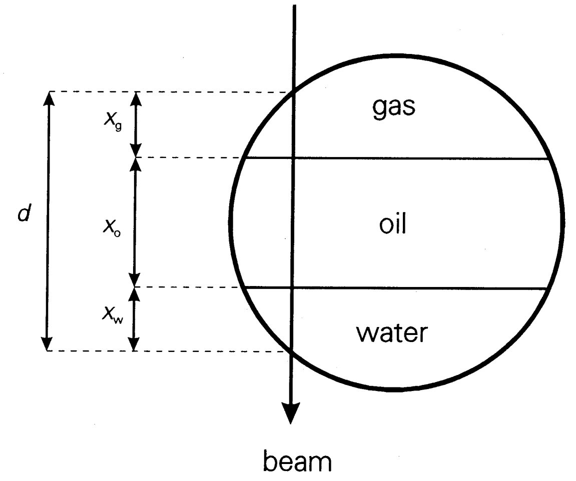

Fig. 1. Schematic cross section of a pipe containing oil, water and gas in a stratified configuration, showing the path of a gamma beam together with the definitions of the path lengths x_o, x_w, x_g and d.

A gamma beam passing through a multiphase pipeline will be attenuated by water, oil and gas each of which will have its own density and absorption coefficient. Fig. 1 shows a schematic illustration of a collimated gamma beam passing through a circular cross section pipe containing oil, water and gas in a stratified configuration. The intensity of the beam after passing through the pipe will be given by

$$I = I_0\, e^{-\mu_o\rho_o x_o}\, e^{-\mu_w\rho_w x_w}\, e^{-\mu_g\rho_g x_g}, \tag{2}$$

where I_0 is now the beam intensity if there were no material in the pipe (i.e. a vacuum), and x_o, x_w and x_g are defined in fig. 1. Here we have introduced separate densities and mass absorption coefficients for each of the three phases. Note that even if the three phases are in a different geometrical configuration, for instance if they are homogeneously mixed, we can still apply eq. (2) provided we interpret the x's as the total effective path length through each phase.

The basic problem is to determine the values of the three unknowns x_o, x_w and x_g. In dual-energy gamma densitometry this is achieved by sending a second gamma beam through the pipe along the same chord as the first beam. This second beam has a different wavelength, and so the absorption coefficients will have different values. For the second beam we have

$$I' = I'_0\, e^{-\mu'_o\rho_o x_o}\, e^{-\mu'_w\rho_w x_w}\, e^{-\mu'_g\rho_g x_g}. \tag{3}$$

It is convenient at this point to define the quantities

$$L = -\ln(I/I_0), \quad L' = -\ln(I'/I'_0). \tag{4}$$

We can then write eqs. (2) and (3) in the form

$$L = \mu_o\rho_o x_o + \mu_w\rho_w x_w + \mu_g\rho_g x_g, \tag{5}$$

$$L' = \mu'_o\rho_o x_o + \mu'_w\rho_w x_w + \mu'_g\rho_g x_g. \tag{6}$$

Since L and L' are measured from the gamma beam attenuation, and since the ρ's and μ's are known, eqs. (5) and (6) represent two equations in three unknowns. The third equation needed to determine the x's comes from the geometrical constraint that the sum of the x's must equal the total path length

$$x_o + x_w + x_g = d, \tag{7}$$

as shown in fig. 1. We can therefore solve the three simultaneous algebraic eqs. (5), (6) and (7) to give the path length in oil in the form

$$x_o = \frac{\left\{ \dfrac{L - \mu_g \rho_g d}{\mu_w \rho_w - \mu_g \rho_g} - \dfrac{L' - \mu'_g \rho_g d}{\mu'_w \rho_w - \mu'_g \rho_g} \right\}}{\left\{ \dfrac{\mu_o \rho_o - \mu_g \rho_g}{\mu_w \rho_w - \mu_g \rho_g} - \dfrac{\mu'_o \rho_o - \mu'_g \rho_g}{\mu'_w \rho_w - \mu'_g \rho_g} \right\}}. \tag{8}$$

A similar expression is obtained for x_w. The value of x_g can then be found from eq. (7). Thus, dual-energy gamma densitometry allows us to determine the path length through each phase along the line of sight of the gamma beam. This, however, does not allow us to determine the oil fraction directly since the path lengths depend both on the oil fraction and on the geometrical configuration of the phases. Since many different phase configurations are possible with multiphase flows, we need further information.

One approach is to introduce a baffle system into the pipe which is designed to induce turbulence into the flow and thereby homogenise the three phases. If the gamma measurements are made a short distance downstream of the baffle, then we can assume that the three phases are well mixed, and the phase fractions can be determined from the path lengths using

$$f_o = x_o/d, \quad f_w = x_w/d, \quad f_g = 1 - f_o - f_w, \tag{9}$$

where f_o, f_w and f_g represent the volume fractions of oil, water and gas respectively. One significant drawback with this approach is that there must be a suitable baffle plate already installed in the pipeline. This technique cannot therefore be used with a portable monitor system intended to be used on a new section of pipe at short notice. A second disadvantage with the baffle system is that the baffle itself can introduce significant drag into the flows and so reduce pumping efficiency.

We describe here an alternative technique which does not require any modification to the pipeline and which can therefore be used as a truly portable flow monitor. It is based on the use of multiple beam lines, each comprising a dual-energy gamma densitometer, to give additional information concerning the phase configuration. Throughout this paper we shall consider a particular system, employing six beam lines arranged as shown in fig. 2, which together provide 12 independent measurements giving information on the phase configuration. The principles are quite general, however, and can be applied equally well to any similar system. Details of the geometry are given in the appendix. If the phase configuration is known then the phase fractions can usually be obtained by straightforward calculation, as was done for the case of homogeneous flows in eq. (9). The task, therefore, is to use the 12 measurements to determine the phase configuration. Note that the phase configuration may not be static, but may have significant time variations. Since the dual-energy gamma densitometry technique must take a measurement over a period of several seconds it will average over any time variation on a shorter time scale. The system therefore sees a time averaged phase configuration.

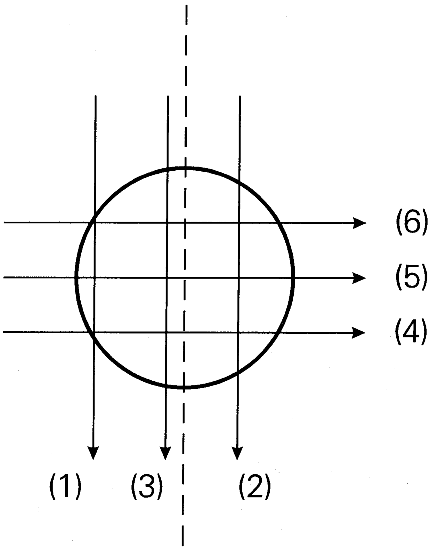

Fig. 2. Cross section of the pipe showing the arrangement of the six beam lines. Each beam line comprises a single dual-energy gamma densitometer.

A key difficulty in determining the phase configuration is that the various possible configurations are not usually known in advance. In principle, one approach to the determination of the phase configuration from the dual-energy measurements is to use tomographic reconstruction techniques to obtain an image of the phase configuration directly. In practice such an approach cannot be adopted due to the very limited number of beam lines available. (Practical tomography systems generally make use of hundreds or thousands of independent lines of sight.) In this paper we shall show how neural network techniques provide an effective and practical solution to the problem of determining the configuration.

Furthermore, we shall show that a neural network can be used to map the 12 measurements directly onto the phase fractions themselves, without the need to determine the phase configuration as an intermediate step.

3. The neural network algorithm

Before describing the neural network approach to the determination of phase configurations and fractions, we give here a brief overview of the neural network algorithm which we shall use. More general accounts of neural networks can be found in refs. [6,7]. For a recent review of neural networks and their applications see ref. [8].

Neural network techniques have their origins in ideas of brain function dating back to the 1940s. One of the key aims of researchers in this field is to provide new algorithms and techniques for solving pattern recognition and data analysis problems. After a period of considerable activity in the 1960s, research in this area diminished during the 1970s as alternative approaches were pursued. The tremendous resurgence of activity in this area in the late 1980s can be attributed in part to the availability of high speed computer hardware and in part to the development of powerful new algorithms such as the technique of "error back-propagation" which will be reviewed shortly. Neural network techniques have now been proven to be of considerable practical benefit in a wide range of application areas.

The biological inspiration of neural networks has led to much misplaced optimism and exaggeration of their current capabilities. This is unfortunate, since they represent a genuinely new set of data processing tools with many useful attributes. Here we shall view neural networks as an extension of the family of statistical pattern recognition techniques. As such, many aspects of their performance can be understood without appealing to loose biological analogies.

Our approach to the interpretation of data from the dual-energy gamma densitometer is based on a particular class of neural network known as the multilayer perceptron (MLP). We therefore begin with a general description of MLP and highlight its key features. The MLP can be regarded as a class of nonlinear functions which map a number of input variables onto a number of output variables. The particular transformation is governed by a set of parameters (known weights and biases) whose values can be chosen with the aid of a set of examples of the desired mapping. The procedure of determining the weights and biases is called network training, and the set of examples is called a training set. First, we shall describe the MLP in terms of a class of parameterised functions. Then we will show how it can be represented as a network diagram. Finally, we shall discuss the algorithms which we use to train the network.

Consider a set of N input (independent) variables $x_1, \cdots, x_N$, which can be grouped together to form an N-dimensional vector $\boldsymbol{x}$. The MLP is a functional mapping which transforms these input variables into a set of M output (dependent) variables $y_1, \cdots, y_M$, which can similarly be grouped together to form an M-dimensional vector $\boldsymbol{y}$. The MLP transformation is specified by the following three steps.

1) Multiply $\boldsymbol{x}$ by a matrix of weight parameters $\mathbf{W}^{(1)}$ of size $N \times L$ and add an L-dimensional bias vector $\boldsymbol{b}^{(1)}$ to give an L-dimensional intermediate vector $\boldsymbol{a}$

$$\boldsymbol{a} = \mathbf{W}^{(1)} \cdot \boldsymbol{x} + \boldsymbol{b}^{(1)}. \tag{10}$$

2) Transform each of the components of the vector $\boldsymbol{a}$ using a nonlinear function $g(a)$ to create a new vector $\boldsymbol{z}$.

$$z_i = g(a_i) \quad i = 1, \cdots, L. \tag{11}$$

3) Finally, multiply the vector $\boldsymbol{z}$ by a second matrix of weight parameters $\mathbf{W}^{(2)}$ of size $L \times M$ and add an M-dimensional bias vector $\boldsymbol{b}^{(2)}$ to give the output vector $\boldsymbol{y}$

$$\boldsymbol{y} = \mathbf{W}^{(2)} \cdot \boldsymbol{z} + \boldsymbol{b}^{(2)}. \tag{12}$$

Thus the whole transformation generated by the MLP can be written in the form

$$\boldsymbol{y} = \mathbf{W}^{(2)} \cdot g(\mathbf{W}^{(1)} \cdot \boldsymbol{x} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)}, \tag{13}$$

where it is understood that the function $g(\ )$ acts on each component of its vector argument separately. There are several possible choices for $g(\ )$, although it must be differentiable. We shall use the sigmoid

$$g(a) \equiv \frac{1}{1 + \exp(-a)}, \tag{14}$$

which is plotted in fig. 3.

The form of MLP represented by eq. (13) will be used for the determination of phase fractions in sec-

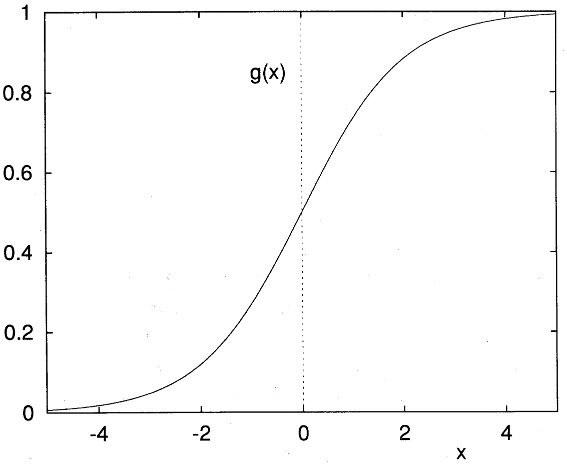

Fig. 3. Plot of the sigmoid function defined in eq. (14).

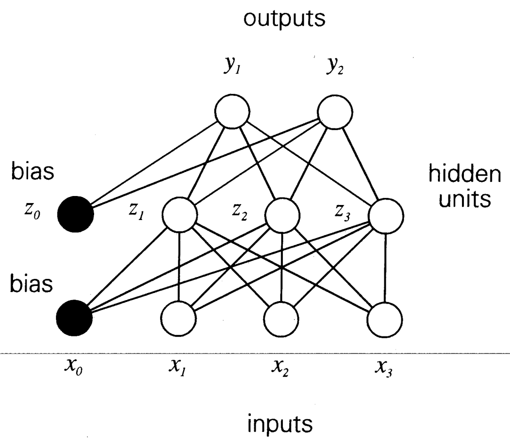

Fig. 4. An example of a multilayer perceptron having three input units, three units in the hidden layer and two output units. The black circles correspond to bias units whose activation is fixed at +1.

tion 6. For problems in which we wish to classify the input vectors into a number of discrete categories it is convenient to transform the output variables also using the sigmoid function, so that

$$y = g(\mathbf{W}^{(2)} \cdot z + b^{(2)}) \qquad (15)$$

which, for the sigmoidal function in eq. (14), ensures that the outputs of the network remain in the interval (0, 1). This form of the MLP will be used in the determination of phase configurations in section 5.

Before we discuss the properties of the MLP in detail, we first show how it can be represented in terms of a network diagram. This will also serve to introduce some of the terminology. In fig. 4 we show an example of an MLP, for the particular case of $N = 3$, $L = 3$ and $M = 2$. The three inputs $x_1 - x_3$ are represented by circles at the bottom of the diagram, the three intermediate variables $z_1 - z_3$ correspond to the three circles labelled hidden units, and the two outputs y_1 and y_2 are represented by the two circles at the top of the diagram. The elements of the two matrices $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$ are represented by lines connecting the corresponding units. The biases can be regarded as special weights from an extra input or hidden unit whose activation is fixed at +1. These extra units are shown by the black circles. The term neural network arises because there is a superficial analogy between the mathematical operations performed by the hidden and output units in the artificial network, and the neurons in the human brain.

The functional form in eq. (13) has two important properties. First, it can be shown that, for a sufficiently large number L of hidden units, the MLP can represent any functional mapping (subject to some mild restrictions) from one multidimensional space to another multidimensional space with arbitrary accuracy. This universality property is discussed in ref. [9]. Second, there exist computationally efficient procedures which allow the weight matrices and bias vectors to be chosen on the basis of a set of examples of the desired mapping. These are referred to as training algorithms and the set of examples is usually called a training set.

A useful analogy to help understand the MLP is that of curve fitting. In a standard curve fitting problem we are given a set of values of the independent variable x together with the corresponding values of the dependent variable y. The problem is to find a smooth curve which provides a representation of the underlying trend of the data points. For example we might wish to represent the trends in the data using a quadratic polynomial of the form

$$h(x) = w_2 x^2 + w_1 x + w_0. \qquad (16)$$

The curve is parameterised by the values of w_2, w_1 and w_0. Their values can be chosen by minimising an error function which is often taken to be a sum of squares of the form

$$E^q = \sum_p (h(x^p) - y^p)^2 \qquad (17)$$

where p is an index which labels the individual data points (x^p, y^p).

The MLP neural network generalises the technique of curve fitting to allow for arbitrary numbers of input (independent) and output (dependent) variables, and to allow for essentially arbitrary functional forms (from the universality theorem mentioned above). The weight and bias parameters in the neural network are analogous to the coefficients of the polynomial in eq. (16). The determination of the weight and bias values can be done in a similar way to that used for polynomial curve fitting. We shall therefore consider the minimisation of a sum-of-squares error of the form

$$E = \sum_p | y(x^p) - y^p |^2 \qquad (18)$$

where $| \cdots |$ denotes the length of a vector, and again p labels an individual point (x^p, y^p) in the training data set. There is an important difference between the error functions in the two cases, however. The error function E^q for the polynomial was a quadratic function of the coefficients, and so its derivatives with respect to the coefficients form a set of linear equations which are easily solved and which in general have a unique solution. The error function E for the neural network, however, is a complex nonlinear function of the weights and biases, and the determination of its minimum represents a complex problem, with the possibility of local as well as global minima.

Fortunately the MLP exhibits an important property which drastically reduces the computational effort needed to train it. Most practical optimisation algorithms for minimising non-linear functions make use of the derivatives of the error with respect to the adjustable parameters. If the function depends on $\mathcal{N}$ parameters (the weights and biases in the case of the neural network error function) then the computational effort needed to evaluate the error will scale like $\mathcal{N}$. Furthermore, there are $\mathcal{N}$ derivatives which need to be calculated, and so the total computational effort which is needed to evaluate all of the derivatives will in general scale like $\mathcal{N}^2$. For the MLP, however, there is an efficient algorithm, known as *backpropagation* [10], which allows all $\mathcal{N}$ derivatives to be obtained in $\mathcal{O}(\mathcal{N})$ steps. The discovery of this algorithm is one of the reasons why neural network techniques have found such a large number of practical applications in recent years. It is somewhat analogous to the discovery of the fast Fourier transform (FFT) algorithm which reduced the computational effort needed from $\mathcal{O}(\mathcal{N}^2)$ to $\mathcal{O}(\mathcal{N} \ln \mathcal{N})$, and led to the widespread use of the Fourier transform. A detailed description of the backpropagation algorithm can be found in refs. [7,8,10].

The backpropagation algorithm allows the derivatives of E with respect to the weights and biases to be evaluated efficiently. These derivatives can then be used in a variety of standard nonlinear optimisation algorithms, such as conjugate gradients, to find the minimum of E starting from some randomly chosen initial starting point in weight space. The original backpropagation technique made use of gradient descent to optimise the weights and biases. We employ a technique known as the BFGS (Broyden–Fletcher–Goldfarb–Shanno) memoryless quasi-Newton algorithm which we have found to be very effective for a wide range of neural network problems. As with the method of conjugate gradients it involves successive line minimisations along carefully selected search directions. One "epoch" of training involves $\mathcal{N}$ line minimisations, where $\mathcal{N}$ is the total number of weights and biases in the network. Training of a neural network generally involves many complete epochs. A more detailed discussion of training algorithms can be found in ref. [8].

In curve fitting problems involving noisy data, if we want to get a good representation of the trends in the data, and not simply fit the curve exactly through each data point, it is important to have an overdetermined problem in which there are more data points than there are degrees of freedom (i.e. coefficients) in the fitting function. A similar situation holds for the case of neural network training. For a given data set we must determine the appropriate number of adjustable parameters in the network. The number of inputs and outputs is generally fixed by the problem being solved. We can, however, adjust the number of parameters by varying the number of hidden units. If there are too few parameters then the range of functions which the network can generate will be too restricted, while if the number of parameters is too great the network function will fit to the noise on the data and will not generalise well when tested with new data. In order to determine the optimum number of hidden units we will use the technique of cross validation. This involves using independent training and test sets, and training networks with various numbers of hidden units on the training set and testing their performance on the test set. The optimum network is the one which gives the smallest test set error.

As we have seen, neural networks with hidden units can approximate a large class of nonlinear transformations. It is also of interest, however, to consider networks having a single layer of adaptive weights. These are much more restricted in the class of functions which they can represent, but this can be advantageous in pattern recognition applications [8]. It also provides a useful indication of the degree of non-linearity needed for the particular problem. Thus, for the determination of phase fractions we will consider transformations of the form

$$y = \mathbf{W}^{(1)} \cdot \boldsymbol{x} + \boldsymbol{b}^{(1)}, \tag{19}$$

which represents a linear relation between input and output variables. For the classification of phase configuration it is again convenient to transform the outputs using a sigmoid function to give

$$y = g(\mathbf{W}^{(1)} \cdot \boldsymbol{x} + \boldsymbol{b}^{(1)}). \tag{20}$$

The weights and biases in these networks will be determined using the same algorithm as for the networks with hidden layers. It should be noted, however, that the solution of the least squares problem for eq. (19) can be formulated in closed form in terms of the pseudo-inverse matrix [8].

4. Generation of datasets

For the purposes of this study we have used synthetic data sets generated in such a way as to model closely the kind of data to be expected from a hardware realisation of the dual-energy gamma densitometer system. In particular, we have based the generation of data on the beam configuration shown in fig. 2. The dominant source of noise in the system arises from photon statistics (except at long integration times), and this too has been accurately modelled.

Multiphase flows along pipes can lead to many different geometrical configurations of the phase fractions. Due to the complexity of modelling multiphase flows this is the most difficult part of the problem to stimulate. For the purposes of this study, however, it is sufficient to consider a small number of example con-

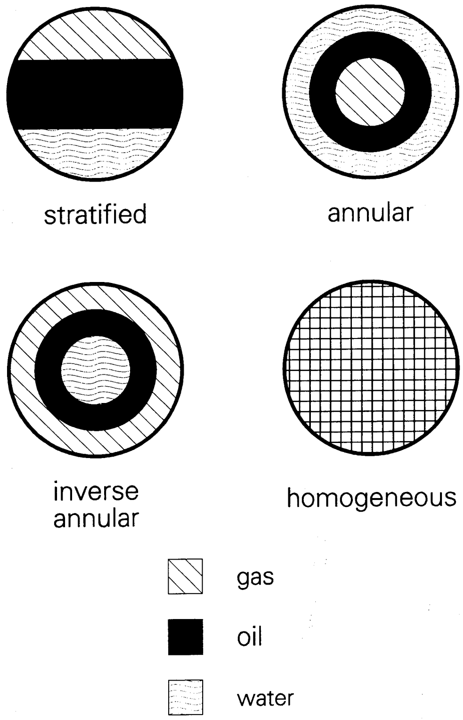

Fig. 5. The four model phase configurations used in this study.

figurations which are representative of the kinds of flows which might be expected in practice. We have considered four different configurations as shown in fig. 5. These are not intended to be accurate representations of true flows, but have been chosen to allow for straightforward analysis. Stratified, annular and homogeneous configurations occur in practice, and "inverse annular" provides a fourth configuration which can be calculated using the same algebra as used for the annular configuration.

For each of these configurations we have allowed all possible values of the oil and water fractions. Data sets have been generated using the following procedure.

1) Choose one of the four phase configurations at random with equal probability.

2) Choose three random numbers f_1, f_2 and f_3 selected uniformly in the interval (0, 1), and set

$$f_o = \frac{f_1}{f_1+f_2+f_3}, \quad f_w = \frac{f_2}{f_1+f_2+f_3}. \tag{21}$$

This procedure treats each of the three phases on an equal footing and ensures that $f_o + f_w + f_g = 1$.

3) For each of the six beam lines, calculate the effective path lengths through oil and water for the given configuration and phase fractions. The equations needed to de this are derived in the appendix.

4) Perturb the path lengths to allow for the effect of photon statistics.

Each data set entry contains the following information: 12 values of path lengths (one in oil and one in water for each of the 6 beam lines), a label describing the phase configuration, f_o and f_w.

The effect of photon statistics is included as follows. If the expected number of counts on a particular beam during a given time interval is $\langle n \rangle$, then the actual number n observed on any particular measurement will be governed by a Poisson distribution $\mathscr{P}$, so that

$$n = \mathscr{P}(\langle n \rangle). \tag{22}$$

For each data point we begin by calculating the expected beam intensities using eqs. (2) and (3). The intensities I_0 and I_0' are calculated by requiring that, for each beam line, the count rate should be 60000 counts s^{-1} when the pipe contains only gas. This sets the maximum count rate to 60000 for each of the detectors and thereby ensures that they are operating in a range where the count rates are as high as possible without saturation (since 60000 cps is a typical upper limit for scintillation counters). Thus we have

$$I_0\, \mathrm{e}^{-\mu_g \rho_g d} = 60\,000, \tag{23}$$

$$I_0'\, \mathrm{e}^{-\mu_g' \rho_g d} = 60\,000. \tag{24}$$

The expected (i.e. average) number of photons is obtained by integrating the intensity over a time interval Δt

$$\langle n \rangle = I\, \Delta t, \quad \langle n' \rangle = I'\, \Delta t. \tag{25}$$

We then replace $\langle n \rangle$ by n using the Poisson distribution (22) to give the observed number of photons for the particular measurement (and similarly for n'). We then retrace our steps and calculate the path length in oil, using eqs. (4) and (8), together with the relations

$$I = n/\Delta t, \quad I' = n'/\Delta t. \tag{26}$$

A similar procedure is used to calculate the path length in water.

The integration time Δt determines the noise level on the measurements. If Δt is small the number of photons detected will be small and will therefore be subject to large fluctuations. Conversely for long integration times the error due to photon statistics will be small. In practice, Δt would be chosen as a comprehensive between minimising noise levels and minimising the time needed to take a measurement.

Note that, for each configuration, all possible values of f_0 and f_w have been allowed. In practice we would anticipate significant correlations between the phase fractions and the phase configuration, which would be expected to simplify the pattern recognition problem. In this sense the data set used in this study may be more challenging than data obtained from an actual experiment. However, this may be offset to some extent by the fact that in practice there may be more than four possible phase configurations.

5. Classification of phase configuration

So far we have described the generation of the data set and we have given an overview of the multilayer perceptron neural network architecture. We now put these together and investigate the capabilities of the neutral network to perform automatic analysis of the data. In this section we consider the problem of using the network to determine the phase configuration, and in the next section we use the network to determine the phase fractions. The use of neural network to predict phase configuration in two-phase air-water flows from air and water flow rates was discussed in ref. [11].

A training set and test set were generated, each containing 1000 examples, but differing in the seed used for the random number generator (which determines the phase configuration and phase fractions). An integration time Δt of ten seconds was chosen as being typical of the value which might be used in an experimental setup. The effects of varying the integration time will be studied in section 7.

Networks were used which had 12 inputs corresponding to the effective path lengths in oil and water for each of the 6 beam lengths. The networks had 4 outputs with one output corresponding to each of the 4 possible configurations, and sigmoidal functions were used for the network output units in accordance with eq. (15). Target values for training the network were chosen using the "one-of-M" coding scheme whereby the target for the output unit corresponding to the configuration for the particular example is set to one, with the targets for all other output units set to zero. Thus, if the 4 output units are assigned to (homogeneous, stratified, annular, inverse annular) respectively, and we have a particular example which belongs to the annular configuration, then the output targets for the example would be (0, 0, 1, 0).

One important feature of the one-of-M coding scheme is that when the error measure is a sum of squares error (as used here) then the outputs of the trained network, (when the network is presented with a new input vector) approximate the Bayesian a-posteriori probabilities of membership of the corresponding class (i.e. phase configuration in this case). The outputs of the network should then sum to one. It is well known that in pattern classification problems the minimum misclassification rate is obtained when new data vectors are assigned to the class having the largest a-posteriori probability. In the case of the neural network trained using the one-of-M coding this means that when a new pattern is presented to the network it should be assigned to the class for which the corresponding output unit has the highest activation. For a more detailed discussion of this result see refs. [8,12].

Networks having various numbers of hidden units were trained, each for 300 complete epochs of the BFGS optimisation algorithm (see section 3). To assess the results from the trained networks we evaluate a root mean square error defined by

$$E^{\mathrm{RMS}} = \left\{ \frac{\sum_{p=1}^{P} |\mathbf{y}(\mathbf{x}^p) - \mathbf{y}^p|^2}{MP} \right\}^{1/2} \tag{27}$$

where P is the total number of patterns and M is the total number of output units. In addition, we also calculate the predicted configuration for each example and compare it with the known configuration from the data set. This allows us to evaluate the fraction of the examples in the data set which the network has classified correctly. Results are shown in table 1.

Table 1
Results for neural network prediction of phase configurations. Values of zero indicate errors of less that 1.0×10^{-4}

N_{hidden}	E^{RMS} (train) $\times 10^2$	E^{RMS} (test) $\times 10^2$	% correct-train	% correct-test
1	26.9	27.0	98.4	98.2
2	3.86	9.51	99.7	98.1
3	3.16	5.47	99.6	99.2
4	0.0	7.98	100	99.0
5	0.0	6.16	100	99.3
6	0.0	6.33	100	99.2
Linear	13.3	14.8	98.9	98.6

Here "linear" denotes the corresponding results obtained from a network having a single layer of weights and biases (i.e. no hidden units) and with sigmoid output units. This corresponds to a form of linear discriminant function.

It is clear from table 1 that even small networks can give excellent discrimination of the phase configuration. Once the phase configuration is known it is then, in principle, a straightforward matter of geometry to calculate the phase fractions given the 12 path length measurements.

More detail on the performance of the network in determining the phase configuration can be obtained from "confusion matrices" which show, for each actual configuration, how the examples were distributed according to the predicted configurations. For perfect classification all entries would be zero except on the leading diagonal. Here the configurations have been ordered as (homogeneous, stratified, annular, inverse annular). The confusion matrices for both training and test sets generated by the trained network having 5 hidden units are

$$\underset{\textbf{Training}}{\begin{array}{cc} & \textbf{Predicted} \\ \text{Actual} & \begin{pmatrix} 259 & 0 & 0 & 0 \\ 0 & 239 & 0 & 0 \\ 0 & 0 & 242 & 0 \\ 0 & 0 & 0 & 255 \end{pmatrix} \end{array}} \qquad \underset{\textbf{Test}}{\begin{array}{cc} & \textbf{Predicted} \\ \text{Actual} & \begin{pmatrix} 255 & 0 & 0 & 0 \\ 1 & 247 & 0 & 1 \\ 2 & 0 & 241 & 0 \\ 3 & 0 & 0 & 250 \end{pmatrix} \end{array}}$$

Similarly, the confusion matrices for both training and test sets for the linear discriminant are

$$\begin{array}{cc} & \text{Predicted} \\ \text{Actual} & \begin{pmatrix} 259 & 0 & 0 & 0 \\ 4 & 232 & 3 & 0 \\ 0 & 0 & 247 & 0 \\ 4 & 0 & 0 & 251 \end{pmatrix} \\ & \textbf{Training} \end{array} \qquad \begin{array}{cc} & \text{Predicted} \\ \text{Actual} & \begin{pmatrix} 255 & 0 & 0 & 0 \\ 4 & 244 & 1 & 0 \\ 2 & 0 & 241 & 0 \\ 7 & 0 & 0 & 246 \end{pmatrix} \\ & \textbf{Test} \end{array}$$

The excellent discrimination performance is related to the fact that the positioning of the beam lines is well matched to the four configurations used in this study. For more complex configurations, such as might occur in practice, the discrimination problem may be more difficult. In this case the advantages of nonlinear networks versus linear networks may become more pronounced.

6. Prediction of phase fractions

We have shown in the previous section that neural network techniques can provide an effective approach to the problem of determining which of a number of (known) phase configurations is present, given the information from the dual-energy gamma densitometer. For a given phase configuration it is then a straightforward geometrical calculation to determine the phase fractions. We have therefore shown that the densitometer data contains enough information to determine the phase fractions. This suggests that we try to extract the phase fractions directly from the densitometer data, without going through the intermediate step of determining the configuration. This is the approach which we explore in this section.

The same data sets were used as in the previous section, and again the networks had 12 inputs corresponding to the 12 densitometer measurements. Two outputs were used, corresponding to the oil and water fractions, and the target values for network training were given by f_o and f_w. When presented with a new input vector, the trained network should then produce an estimate of f_o and f_w directly at the outputs. In this case the sigmoidal activation functions were omitted from the output units, (since they are not needed here) so that the MLP transformation had the form given in eq. (13).

For comparison the same data were used to train a simple linear network of the form given in eq. (19). The results on both training and test sets for the trained neural networks are shown in fig. 6. Here the RMS error defined by eq. (27) is plotted against the number of hidden units. This graph shows a typical trend in which the training error decreases steadily as the number of hidden units, and hence the number of parameters in the network, is increased. The test error, however, decreases at first as the greater flexibility in the network allows more complex functions to be generated, and then increases again when there are too many degrees of freedom in the network. For a discussion of this effect see ref. [8]. The minimum test set error occurs for 8 hidden units as shown by the arrow in fig. 6. The errors both for this network and for the linear mapping are summarised in table 2.

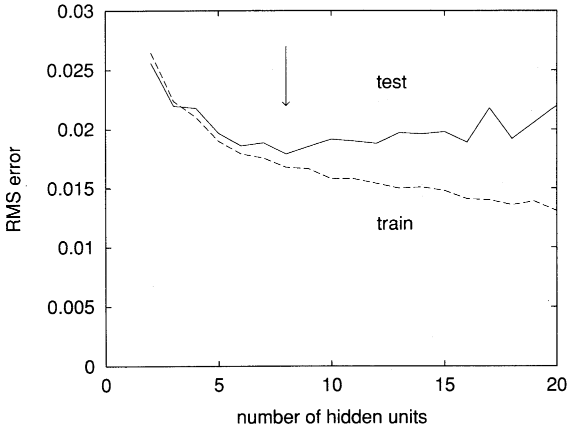

Fig. 6. Plot of the RMS error produced by the neural network mapping for both the training and test sets as a function of the number of hidden units. The arrow indicates the minimum in the test error which occurs for the network having eight hidden units, which is therefore selected as the best network.

Since the phase fraction is a number in the range (0, 1) we see that an error of 1.79×10^{-2} on new data means that the network is able to determine the phase fractions typically to a few percent.

Fig. 7 shows a scatter plot of the oil fraction predicted by the network with 8 hidden units versus the actual oil fraction for all of the points in the test set. The corresponding plot for the linear mapping is shown in fig. 8.

Similar results are obtained for the water fraction f_w.

7. Effects of integration time

As explained in section 4, the integration time Δt determines the noise level on the data arising from

Table 2
Results for neural network prediction of phase fractions for the network with the optimal number of hidden units, and for the linear map

N_{hidden}	E^{RMS} (train) $\times 10^2$	E^{RMS} (test) $\times 10^2$
8	1.68	1.79
Linear	2.72	2.60

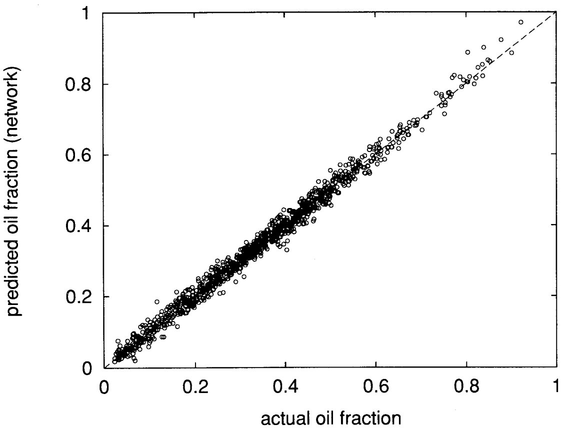

Fig. 7. Scatter plot of the predicted oil fraction from the neural network vs the actual oil fraction, for all points in the test set.

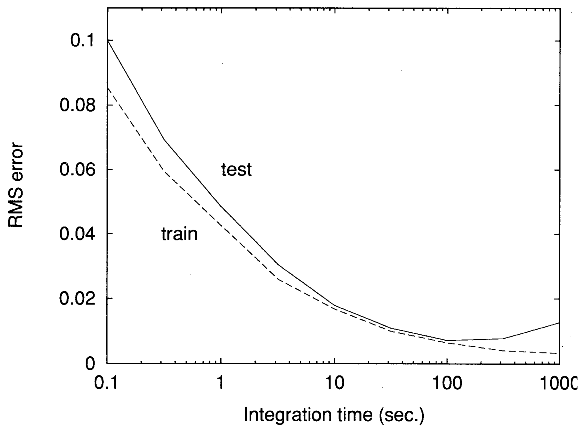

Fig. 9. Plot of the RMS error from both training and test sets vs the integration time for networks having eight hidden units.

photon statistics. For short integration times, therefore, we expect the performance of the network to deteriorate. In order to provide some indication of the integration times which would be needed in a practical system we have investigated the dependence of E^{RMS} on Δt.

We have generated several data sets corresponding to various values of Δt, each with 1000 entries in both training and test versions (differing only in the seed for the random number generator). All training runs were performed with networks having 8 hidden units since this was found to be the optimum number when Δt was set to ten seconds. The results are plotted in fig. 9.

As expected the error increases on both training and test sets as the integration time is reduced below 100 s. Fig. 9 provides an indication of the integration time which might be needed to achieve a given level of accuracy. It is interesting to note that the test RMS error actually increases with increasing Δt for values of Δt greater than about 100 s. This is interpreted as follows. The network architecture was optimised for a value of Δt of 10 s. For much larger values of Δt the data is much smoother and the optimum number of hidden units may be smaller than 8. In this case there will be some tendency towards overfitting, leading to a decrease in the training set error but an increase in the test set error. We have not, however, investigated this effect in detail. It is clear from fig. 9 that for integration times of a few tens of seconds the effects of photon statistics should not give rise to significant errors in the predicted phase fractions.

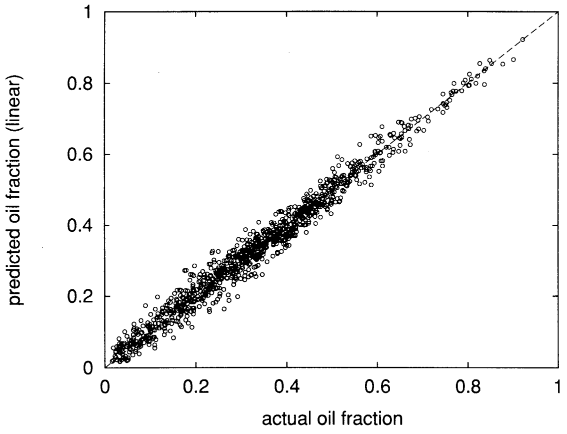

Fig. 8. Scatter plot of the predicted oil fraction from the optimal linear mapping vs the actual oil fraction, for all points in the test set.

It might be supposed that an appropriate strategy would be to train the network using data collected with large integration times (since the network training only needs to be performed once) and then to use shorter integration times when the system is put to use. In fact such a strategy is likely to yield poor results since the new data would then have a substantially different distribution to the training data. Training the network with noisy data avoids the problem of the network being excessively sensitive to noise on new data.

8. Detection of novel configurations

As we have seen, the neural network approach to the analysis of data from dual-energy gamma densitometers relies on the provision of a suitable set of training data. When the trained neural network is applied in the field to new data it can be expected to perform satisfactorily provided the input data is similar to that used during training. More precisely, a correctly trained network should have predictable performance provided the data presented to it is drawn from the same statistical distribution as was used to generate the training data. If substantially novel data is presented to the network then it will be prone to significant errors. In practice, the availability of a controlled test rig should allow the collection of extensive data under a sufficiently wide range of conditions to ensure that the network performs satisfactorily in the field. Nevertheless, it is important to provide an operational system with some means of detecting novel input data in order that some degree of validation can be performed on the results generated by the network. Here we describe a relatively simple approach to the detection of novelty in the input data and show that it is able to detect the occurence of erroneous output signals.

When new data is presented to the network, the novelty detector classifies it according to whether or not it is similar to the data used to train the network. The technique involves evaluating a likelihood function (obtained from the training data) and determining whether its value exceeds a given threshold. While this criterion is intuitively plausible, it can also be derived from the optimal Bayesian formalism for classification. This allows us to clarify the assumptions which are being made in this approach.

We shall suppose that the data which is provided to the network is drawn from one of two classes each described by a fixed probability density function. The class $\mathscr{C}_1$ corresponds to the data used to train (and test) the network and is governed by the class conditional density function $p(\boldsymbol{x} \mid \mathscr{C}_1)$ where $\boldsymbol{x}$ denotes the vector of 12 inputs to the network. This should be read as the probability density of $\boldsymbol{x}$ *given* that the class is $\mathscr{C}_1$. The second class $\mathscr{C}_2$ represents the novel data corresponding to new configurations of multiphase flow which were absent from the training set (or to similar configurations to those in the training set but with novel values of the phase fractions). Again, we shall suppose that this data is drawn from a fixed class conditional density function $p(\boldsymbol{x} \mid \mathscr{C}_2)$. We shall suppose that, once the network is installed in the field, the data vectors are drawn from $\mathscr{C}_1$ with a-priori probability $P(\mathscr{C}_1)$, and from $\mathscr{C}_2$ with a-priori probability $P(\mathscr{C}_2)$, where $P(\mathscr{C}_1) + P(\mathscr{C}_2) = 1$.

Given a particular input vector $\boldsymbol{x}$ we wish to assign it to one of the two classes $\mathscr{C}_1$ or $\mathscr{C}_2$. The probability of misclassification is minimised if we assign it to the class for which the a-posteriori probability is largest [13]. From Bayes theorem the a-posteriori probability of the vector belonging to class $\mathscr{C}_k$ can be written as

$$P(\mathscr{C}_k \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid \mathscr{C}_k) P(\mathscr{C}_k)}{p(\boldsymbol{x})}, \tag{28}$$

where the unconditional density $p(\boldsymbol{x})$ is given by

$$p(\boldsymbol{x}) = p(\boldsymbol{x} \mid \mathscr{C}_1) P(\mathscr{C}_1) + p(\boldsymbol{x} \mid \mathscr{C}_2) P(\mathscr{C}_2). \tag{29}$$

This acts as a normalisation factor to ensure that the a-posteriori probabilities sum to 1

$$P(\mathscr{C}_1 \mid \boldsymbol{x}) + P(\mathscr{C}_2 \mid \boldsymbol{x}) = 1. \tag{30}$$

Since the denominator in eq. (28) is the same for both classes it can be omitted from the comparison of probabilities needed to assign $\boldsymbol{x}$ to a particular class. We therefore assign $\boldsymbol{x}$ to class $\mathscr{C}_1$ if

$$p(\boldsymbol{x} \mid \mathscr{C}_1) > \frac{p(\boldsymbol{x} \mid \mathscr{C}_2) P(\mathscr{C}_2)}{P(\mathscr{C}_1)} \tag{31}$$

and to class $\mathscr{C}_2$ otherwise.

We can obtain an expression for $p(\boldsymbol{x} \mid \mathscr{C}_1)$ using standard techniques of non-parametric density estimation [13]. We have chosen a kernel based approach with Gaussian kernel functions in which the kernel function centres are given by the data points in the training set. We therefore have

$$p(\boldsymbol{x} \mid \mathscr{C}_1) = \text{const.} \sum_{p=1}^{P} \exp\left\{ -\frac{|\boldsymbol{x} - \boldsymbol{x}_p|^2}{2h^2} \right\}, \tag{32}$$

where $\boldsymbol{x}_p$ represents the pth data point from the training set. The parameter h controls the degree of smoothness of the estimated density function and its value must be neither too large nor too small. We have adopted the simple heuristic of setting h to the average distance of the ten nearest neighbours, averaged over all points in the training data set. This gave a value of $h = 0.252$.

The determination of the density function for the novel data is more problematic since by definition we do not know what sort of configurations to expect. We do know that the measured path lengths are subject to some simple constraints, so that if d is the total path length in the pipe for a given beam we must have

$$0 \le x_\text{o} \le d, \quad 0 \le x_\text{w} \le d - x_\text{o}. \tag{33}$$

These constraints are illustrated in fig. 10. In practice these constraints are "softened" somewhat by the effects of photon noise and other sources of error. Beyond this we have little idea of what to expect and

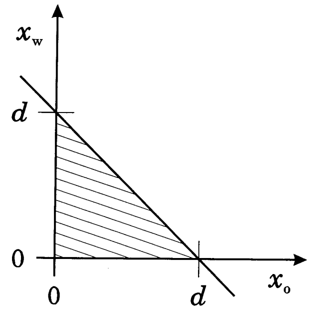

Fig. 10. Accessible values of x_o and x_w for a single beam line, before allowing for the effects of photon statistics.

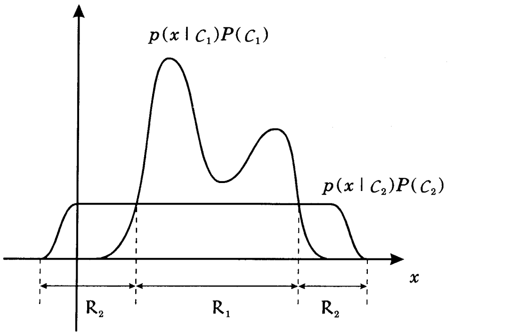

Fig. 11. Schematic illustration of the Bayesian formulation of novelty detection. Points which fall in the region $\mathscr{R}_2$ are classified as novel.

so we shall take the density function for class $\mathscr{C}_2$ to be uniform within the boundaries of the regions defined by the constraints. Since the a-priori probabilities are also constants the decision criterion (31) amounts to setting a constant threshold for $p(\boldsymbol{x}\mid\mathscr{C}_1)$. Any given value for the threshold defines a decision surface in the 12-dimensional input space which divides it into two regions $\mathscr{R}_1$ and $\mathscr{R}_2$ according to whether the criterion classifies an input vector $\boldsymbol{x}$ as belonging to class $\mathscr{C}_1$ or $\mathscr{C}_2$. This is illustrated schematically for a one-dimensional space in fig. 11. As we shall see, one way to choose the threshold is to set it to the largest value that still allows the bulk of the test set examples to be classified as class $\mathscr{C}_1$. Such a procedure avoids having to choose values for the a-priori probabilities. In order to test the performance of this novelty detector we have generated a further data set, consisting of 1000 examples with randomly chosen oil and water fractions, corresponding to a 5th configuration referred to as "inverted-stratified" which is obtained by inverting the stratified configuration of fig. 5. Again, this is not intended to be a realistic configuration but is chosen for computational simplicity. It suffices, however, to illustrate the detection of novel configurations. As before, photon statistics were included, with an integration time of 10 s.

When viewed as a function of $\boldsymbol{x}$, the density function $p(\boldsymbol{x}\mid\mathscr{C}_1)$ is generally called a likelihood. In order to represent the results graphically, it is convenient to consider the logarithm of this quantity. Since "log" is a monotonic function it does not affect the location of the decision boundaries, provided the threshold is adjusted accordingly. Fig. 12 shows a plot of the log likelihood versus the magnitude of the error between the oil fraction predicted by the neural network and the true value obtained from the data set. Here the network with 8 hidden units described in section 6 was used. The crosses show the 1000 points from the test set (with an integration time of 10 s) as used in section 6. It is clear that for these data the network gives a small error (as we have already seen) and that the log likelihood is always larger than (say) -5. The circles show the 1000 samples corresponding to the inverted-stratified configuration. The majority of these points have log likelihood values which are substantially smaller than those of the test set points, and a correspondingly larger range of oil fraction errors. We see that the network can indeed generate poor results when presented with data from this new configuration. Such data points can, however, easily be rejected on the basis of their log likelihood values. Setting a threshold anywhere between -5 and -10 would reject all data points having significant phase fraction errors.

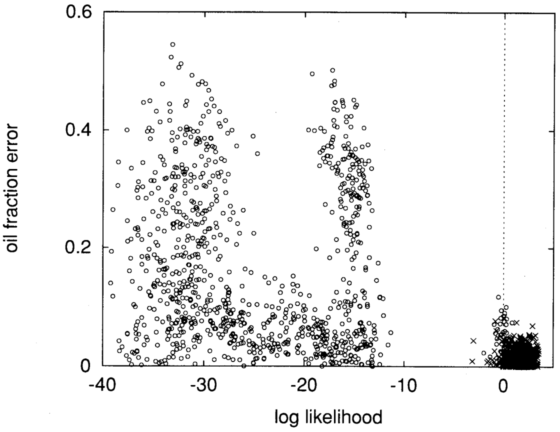

Fig. 12. Plot of log-likelihood from the novelty detector vs absolute oil fraction error from the neural network. Crosses show data from the original test set, while circles correspond to data from a new 5th phase configuration.

It can also be seen from fig. 12 that there are some inverted-stratified points lying within the cluster of crosses. Examination of the phase fractions for these points shows that they represent configurations which

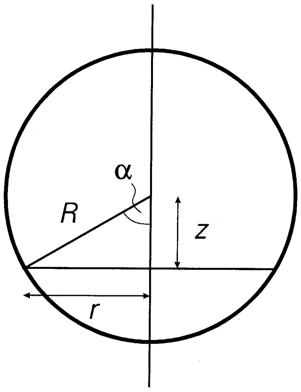

Fig. 13. Definition of the variables to find the height of an interface between two phases, given the phase fractions and the pipe radius.

could also be classified as stratified configurations. For instance, if the oil phase fraction is sufficiently large then the three horizontal beam lines pass through oil only and there exist stratified and inverted-stratified configurations having the same phase fractions which give rise to the same 12 path length measurements. The novelty detector "correctly" interprets these as being similar to the training data, and indeed the network predicts the phase fraction to high accuracy.

In practice, we would expect the neural network to generate accurate phase fraction predictions under most circumstances. The role of the novelty detector is to prevent occasional novel configurations from generating spurious outputs. This ensures that the neural network can form the basis for a robust system which can be deployed in the field.

A more severe problem would arise if there existed two distinct configurations for which different values of the phase fractions gave rise to similar measurement values from the densitometer beam lines. In such cases any data analysis method which worked solely with the densitometer data would necessarily be prone to serious error. This problem can be avoided by careful selection of the number and placement of the beam lines so as to ensure that the densitometer always provides sufficient information to allow the different configurations to be discriminated. This in turn requires some prior understanding of the qualitative nature of possible configurations.

9. Discussion

In this paper we have shown that neural network techniques, combined with dual-energy gamma densitometry, provide a powerful and accurate approach to the non-invasive analysis of multiphase flows. Although there is insufficient information to perform a tomographic reconstruction of the phase configuration, a neural network can learn to distinguish between a finite number of previously characterised configurations with a high degree of discrimination. In most situations the quantity of primary interest is the phase fraction, and we have shown that this can be obtained directly by a nonlinear mapping from the path length information generated by the densitometer.

The neural network approach described here relies on a set of characterised data for training the network. In a practical realisation this would be obtained using a calibration ring in which the phase fractions and fluid velocity could be controlled and measured. The need for any modelling of the multiphase flow, which would probably be complex and of limited accuracy, is thereby avoided.

One difficulty which can potentially arise in practice is the occurrence of phase configurations which were not present in the training set. Under such conditions the network could not be expected to generate reliable results. This problem can largely be circumvented by the provision of a suitably general training set spanning the full range of conditions likely to be encountered in practice. However, in order to ensure the robustness of a practical system, we have developed a technique for the detection of novelty in the input data and we have shown that it is able to signal the presence of new phase configurations which are not present in the training set. Such a novelty detector can be used to validate the phase fractions generated by the neural network.

Although we have worked with the path length information in this study, the network could equally well take the photon count information directly as input data, thereby avoiding the need to perform the calculations normally required to extract the path lengths.

Appendix A

Calculation of path lengths

In this appendix we derive the expression needed to evaluate the path lengths in oil and water (as illustrated for stratified flow in fig. 1) for each of the four configurations shown in fig. 5. The beam positions are specified in terms of the radii of the circles (concentric with the pipe) to which the beam lines are tangent. For a pipe of radius R, these radii were 0, $R/6$, $R/2$ and $5R/6$, as indicated in fig. 2. All calculations were for $R = 7.5$ cm. We treat each of the configurations in turn.

A.1. Homogeneous

Here the path lengths x_o and x_w are related to the phase fractions f_o and f_w by

$$x_o = f_o d, \quad x_w = f_w d, \tag{34}$$

where as before d is the total path length of the beam within the pipe.

A.2. Stratified

It is first necessary to determine the heights of the oil/water and water/gas interfaces. The basic calculations which has to be performed is indicated in fig. 13. Let f be the fraction of material below the interface, and R be the radius of the pipe, with the other variables as shown in the figure. Then we have the following three equations

$$z^2 + r^2 = R^2, \tag{35}$$

$$z = R \cos \alpha, \tag{36}$$

$$\left(\frac{2\alpha}{2\pi}\right)\pi R^2 = f\pi R^2 + 2\left(\frac{zr}{2}\right). \tag{37}$$

We can now eliminate both α and r from these equations to give an equation which determines z for a given pipe radius R and phase fraction f. This can be expressed in terms of the root of a transcendental function $G_f(z)$ given by

$$G_f(z) \equiv f\pi + \frac{z}{R}\left(1 - \frac{z^2}{R^2}\right)^{1/2} - \cos^{-1}\frac{z}{R}. \qquad (38)$$

A Brent algorithm routine [14] was used to find the root of $G_f(z)$. Care is needed to ensure that the correct branch of $\cos^{-1}$ is taken.

Having found the heights of the two interfaces it is then necessary to determine which of the three phases each beam passes through. This will depend on the location of the beam line and on the heights of the interfaces, and six possibilities have to be considered according to whether the beam passes through one, two or all three phases. The path lengths are then obtained by straightforward algebra, with one further application of Pythagoras' theorem.

A.3. Annular

First the phase fractions are used to evaluate the radii of the interfaces. Then for each beam line it is necessary to determine which of the phases the beam passes through, given the position of the beam line and the phase fractions. There are three possibilities here since the beam may pass through water only, water plus oil, or water plus oil plus gas. Finally the various path lengths are evaluated using Pythagoras' theorem.

A.4. Inverse annular

The procedure is the same as for the annular configuration but with the rôles of oil and gas reversed.

Table 3
Values of the coefficients used in generating the data set. Here the μ's are in units of cm^2/g and the ρ's are in units of g/cm^3

	μ (@ 0.06 MeV)	μ (@ 1.33 MeV)	ρ
Oil	0.197	0.062	0.9
Water	0.220	0.058	1.05
Gas	0.213	0.068	0.2

In generating the data set we have assumed gamma wavelengths of 0.06 MeV and 1.33 MeV, and the values used for the densities and mass absorption coefficients are shown in table 3. Here the water has been taken to be "formation water" of a composition associated with oil production.

References

[1] P. Kehler, IEEE Trans. Nucl. Sci. NS-26 (1979) 1627.

[2] M.S. Abouelwafa and E.J.M. Kendall, J. Phys. E: Sci. Instr. 13 (1980) 341.

[3] K. Rafa, T. Tomoda and R. Ridley, Proc. Energy Sources Technology Conference and Exhibition, eds. B. Chrisman and J.V. Serve, ASME (1989) 89-Pet-7.

[4] M.S. Beck and A. Plaskowski, Cross Correlation Flowmeters (Adam Hilger, Bristol, 1987).

[5] J.S. Watt, H.W. Zastawny, M.D. Rebgetz, P.E. Hartley and W.K. Ellis, Nuclear Techniques in the Exploration and Exploitation of Energy and Mineral Resources (IAEA, Vienna, 1991) p. 481.

[6] R.P. Lippmann, IEEE ASSP Magazine 4 (1987) 4.

[7] J.A. Freeman and D.M. Skapura, Neural Networks: Algorithms, Applications and Programming Techniques (Addison-Wesley, New York, 1991).

[8] C.M. Bishop, Review article: Neural Networks and their Applications, to appear in Rev. Sci. Instr. (1993).

[9] K. Hornik, M. Stinchcombe and H. White, Neural Networks 2 (1989) 359.

[10] D.E. Rumelhart and J.L. McClelland, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol 1: Foundations (MIT Press, Cambridge MA, 1986).

[11] N. Baba, Y. Yamashita and Y. Shiraishi, Proc. 1991 Int. Conf. on Artificial Neural Networks, Finland, eds. T. Kohonen et al. Vol. II (North-Holland, Amsterdam, 1991) p. 1617.

[12] M.D. Richard and R.P. Lippmann, Neural Computation 3 (1991) 341.

[13] R.O. Duda and P. Hart, Pattern Classification and Scene Analysis (Wiley, New York, 1973).

[14] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, Numerical Recipes in C, the Art of Scientific Computing (Cambridge Univ. Press, Cambridge, 1988).