

# A neural network approach to the calibration of a flush air data system

W.J.Crowther, P.J.Lamont

School of Engineering, University of Manchester

Updated 27 November 2000

## Abstract

A flush air data system uses surface pressure measurements to obtain speed and aerodynamic orientation of a flight vehicle. This paper investigates the use of a neural network to calibrate the air data system on a low-observable aircraft forebody with 22 pressure tappings. Wind tunnel data were obtained for between 0 and 25° angle of attack, 0 to 10° sideslip for speeds of 32, 37 and 45m/s. Experimental data were used to train multilayer perceptron neural networks. A calibration accuracy of 0.322m/s, 0.811° and 0.552° for speed, alpha and beta respectively was achieved just using the first five tappings on the nose cone. Increasing the number of tappings used as inputs to the neural network reduces the calibration error. A neural network with 22 inputs gives a best accuracy of 0.095m/s, 0.15° and 0.085° for speed, alpha and beta respectively. Trained networks show poor robustness to single sensor failures. Robustness is improved by preprocessing input data with an autoassociative network or by introducing sensor redundancy.

## Nomenclature

$f_a$	neural network activation function
$j$	index referring to neural network input
$n$	pressure tapping number, 0 - 22.
$p$	pressure, N/m <sup>2</sup>
$u$	speed, m/s
$S$	sensitivity index
$w$	neural network weight
$x$	neural network input
$y$	neural network output
$\alpha$ , alpha	angle of attack, degrees
$\beta$ , beta	angle of sideslip, degrees
$\phi$ , phi	roll angle, degrees
$\theta$ , theta	pitch angle, degrees

## 1. Introduction

The role of an air data system is to determine the speed at which a flight vehicle is moving through the surrounding fluid and its orientation relative to the wind vector (angle of attack, angle of sideslip). Conventionally, speed is determined by measuring the difference between total and static pressure of the oncoming airstream via a dedicated Pitot static tube or Pitot tube and separate static hole, whereas aerodynamic orientation is measured using small yaw vane devices mounted on a forward part of the vehicle.

Conventional intrusive air data sensing is relatively straightforward to implement and high levels of accuracy and robustness can be achieved. However, for some

specialised applications the presence of external instrumentation is undesirable. For example, the presence of an instrumented air data nose boom on research aircraft is known to lead to increased sensitivity to asymmetric forebody vortex shedding and lateral instability at high angles of attack **(1)**. A different problem exists for hypersonic or reentry vehicles where the high-energy airflow precludes the use of intrusive air data systems **(2)**. Finally, for stealth vehicles, conventional intrusive air data systems are highly visible and thus induce an unacceptable performance penalty.

The work presented in this paper considers the use of a flush air data system for determining vehicle air speed and aerodynamic orientation. A flush air data system uses data only from surface pressure tappings to determine speed and aerodynamic orientation. These tappings may be placed at any point on the aircraft as installation, aerodynamic or stealth constraints dictate.

The simplest implementation of a flush air data system is in the form of a multi-hole probe. This type of probe has a hemispherical or conical end with a central pressure orifice surrounded by typically 5 to 7 other orifices. Multi-hole probes are used routinely in the measurement of sub and supersonic flow velocities and flow angles up to around  $70^\circ$ .

Recently, a number of flush air data systems have been demonstrated on F18 and X-33 flight vehicles, **(3)(4)**. The F-18 system uses 11 pressure orifices in the radome of the aircraft and is used for speeds up to Mach 1.6, angles of attack up to  $80^\circ$  and  $\pm 20^\circ$  sideslip. The present work is distinct from these previous studies in that a flush air data system has been implemented on a chined forebody designed for low observability. The presence of chines potentially increases the nonlinearity of variation of surface pressure distribution with angle of attack and sideslip, thus providing a more challenging calibration problem.

It should be noted that the present study considers the calibration of an air data system under *steady* conditions, i.e. the pitch rate and yaw rate is zero. Whilst this is a limitation, it is relevant to note that current flight control system design practice is to use an inertially calculated value of angle of attack and sideslip during high rate manoeuvres or during turbulence where aerodynamic attitude based on aerodynamic measurements is inaccurate or unreliable. This reduces the criticality of air data system dynamic calibration.

## 2. Air data system calibration

The flow around a flight vehicle gives rise to a pressure distribution on the surface of the vehicle. For fixed vehicle geometry, this pressure distribution is a function of wind speed and vehicle aerodynamic orientation, equation 1.

$$p = f(u, \alpha, \beta) \quad \text{equation 1}$$

Assuming that the inverse  $f'$  of the function  $f$  exists over the range of interest then the airspeed and aerodynamic orientation of the flight vehicle is given by equation 2

$$u, \alpha, \beta = f'(p) \quad \text{equation 2}$$

However, the function  $f'$  will not exist if the original function  $f$  is indeterminate or multi-valued over the range of interest. The relatively trivial indeterminate case arises at zero air speed where angle of attack and angle of sideslip do not have meaningful values. The multi-valued problem occurs if a given pressure distribution  $p$  can be obtained by one or more combinations of  $u, \alpha, \beta$ , or a given combination of  $u, \alpha, \beta$  produces different pressure distributions. The latter case is more common and occurs on slender bodies at high angles of attack where vortex instability leads to the possibility of multiple flow solutions.

The simplest implementation of the inverse function  $f'$  is through an interpolated look-up table that provides direct inverse mapping of calibration data. A more sophisticated approach is to use knowledge of the flow physics around the body to derive  $f'$  as a nonlinear analytical model with known structure but unknown coefficients. These coefficients are then obtained from experimental data using nonlinear statistical regression techniques. This technique has been used to calibrate both multi-hole probe data (5) and the F-18 flight flush air data system (3).

The use of neural networks is an evolutionary extension to the above (6). With a neural network approach neither the structure nor the 'coefficients' of  $f'$  are assumed but are obtained through the process of neural network training. A neural network approach has been used to successfully calibrate multi-hole probes in a number of recent studies (7), (8). In both cases the neural network architecture used (multilayer perceptron) was the same as in the present work. Reference 8 also compares the use of a fuzzy logic classifier to a neural network and found that similar results could be obtained.

The direct look up table is least efficient of the calibration methods in that it is slow and requires all the calibration data to be stored in memory. Model-based statistical methods are considerably faster and do not require calibration data in memory. However, these statistical methods may require iterative or multi-stage solution algorithms and can be quite slow. Neural network modelling of  $f'$  produces a highly parsimonious, noniterative model that can be implemented in a few tens of lines of code suitable for fast real-time operation. However, the neural network approach has the disadvantage that the system designer has no control over the structure of model produced and lacks access to conventional statistical measures of model performance.

### 3. Application of neural networks

#### 3.1 Overview

The fundamental processing element of an artificial neural network is the neuron, which is loosely based on its biological namesake. A typical neuron has many inputs but just one output, **figure 1**. Each input to the node is associated with a weight, with the overall input level being equal to the sum of the products of the inputs and the weights, plus a bias term. The neuron output is calculated as a function of the input level. This function is typically nonlinear and is known as the neuron activation function,  $f_a$ . With reference to figure 1, the input-output mapping of a neuron is given by equation 3.

$$y = f_a \left( \sum_j x_j w_j + w_0 \right) \quad \text{equation 3}$$

A neural network is created by assembling neurons into layers and then making connections between the layers. There are a wide variety of different network architectures in existence, but the most common is the multilayer perceptron (mlp). This is a feed forward network with the outputs of one layer acting as inputs to the next. For a wider introduction to neural networks and their engineering applications see references (9),(10),(11),(12),(13).

A three-layer mlp network is shown in **figure 2**. The three layers are denoted input, hidden and output. The number of neurons in the input layer is dictated by the number of input variables, which in the present case is equal to the number of pressure tappings used. Similarly, the number of neurons in the output layer is dictated by the required number of outputs from the network, which is three (speed, alpha and beta). The optimal number of nodes in the hidden layer is problem specific, depending on the fidelity of the mapping required and the availability of training data. The use of too many hidden layer neurons leads to over-fitting of the training data and subsequently poor generalisation properties of the network when implemented.

The optimal number of hidden neurons for the present type of problem is usually less than the number of input neurons but greater than the number of output neurons. Ultimately the number of hidden neurons is found by trial and error. A good starting point for the number of hidden neurons is the average of the number neurons in the input and output layers (14). Neural network node distribution is conveniently denoted in the form  $n_1:n_2:n_3$  where  $n_1, n_2$  and  $n_3$  are the number of nodes in the input, hidden and output layers, respectively.

### 3.2 Neural Network training and implementation

A fundamental theory in neural networks states that a network with two active layers containing nodes with nonlinear activation functions is sufficient to model any continuous function over a finite range to an arbitrary accuracy, (15). Thus a three-layer network with nonlinear nodes in layers two and three is sufficient for modelling the required air data calibration.

A neural network is trained by adjusting the weights within the network such that the required input-output mapping is obtained. The fundamental type of training is based on supervised learning. In this method, training vectors consisting of an input-output pair are presented to the network and the error between the required output and the actual output is calculated. These errors are then fed back through the network (error back propagation) to enable iterative adjustment of the network weights by means of a gradient descent algorithm. Training continues until the output error has reached an acceptably small value. This may take many thousands of epochs (one epoch equals one pass through the training data), depending on the complexity of the function being approximated.

A common problem with neural network solutions is that of over-fitting the training data. Under these circumstances, the error when the implemented network is queried with the original training inputs is small. However, when presented with novel data

that is within the bounds of the original training data but not coinciding with the specific training examples, the network performs badly. This problem is avoided by using an independent set of data to test the network as it is trained. Training continues until the point where the test set error reaches a minimum. Further training will reduce the training error but *increase* the test set error.

Once a network has been trained, the weights are frozen and the weight update algorithm disabled. The network is then queried by presenting pressure data vectors to the input layer of the network. Calibrated values of air speed, angle of attack and sideslip are obtained at the network output layer.

Networks were developed, trained and implemented using a commercial Windows package running on a Pentium PC. Linear activation functions were used in layer 1 and sigmoid in layers 2 and 3. All data were normalised between 0 and 1 with 10% headroom before training. A learning rate of 0.2 and a momentum rate of 0.8 were used in the training algorithm. Weights were updated after presentation of each pattern (as opposed to the end of each epoch), and training was terminated when the test error increased for 100 consecutive epochs.

## 4. Experimental study – stealth combat aircraft forebody

### 4.1 Experimental apparatus

A wind tunnel model of a generic chined forebody, **figure 3**, was machined from hard foam and mated with a steel backbone, **figure 4**. The model was then sting mounted in the University of Manchester 7'x9' wind tunnel. Aerodynamic orientation was varied by pitching and rolling the model. The relationship between model geometric roll/pitch angle and the aerodynamic orientation is illustrated in **figure 5**, and the analytical relationship between geometric and aerodynamic angles is given by equation 4.

$$\alpha = ATan(Tan\theta.Cos\phi)$$

$$\beta = ATan(Tan\theta.Sin\phi)$$

Pressure tappings were placed at four stations along the forebody including a five-hole probe arrangement at the nose tip, **figure 6**. Pressures were measured using a scanning pressure transducer mounted outside the tunnel and recorded using a PC-based data acquisition system.

Pressure data for all 22 tappings was recorded over a test matrix given by an angle of attack range between 0 and 25° at 2° intervals, a sideslip range of 0 to 10° at 2° intervals and for speeds of 32, 37 and 45m/s.

### 4.2 Wind tunnel data

To assess data validity before proceeding to neural network training, sample experimental pressure data were plotted against angle of attack and angle of sideslip for different speeds. Pressure data for the first five tappings are shown **figure 7**. Note that tapping one is effectively a Pitot tube and the pressure is **equation 4** invariant with angle of attack and sideslip.

For the angle of attack and sideslip range tested, the variation in pressure with aerodynamic orientation is relatively linear. Furthermore, when plotted as non-dimensional pressure coefficients, the data for different speeds collapses onto a single curve. This indicates that the flow is Reynolds number insensitive for the range of speeds tested.

### 4.3 *Neural Network training*

The experimental data matrix was divided into a training set and an independent test/query set by taking alternate alpha ‘slices’, giving training data sets at 1,5,9,13,17,32 and 25° angle of attack and the test/query sets at 3, 7,11,15,19 and 23°. This method of division was chosen for ease of presentation and interpretation of results. In practice, a statistically more robust approach would be to divide the data matrix in two randomly assigned sets.

A typical error history during the network training process is shown in **figure 8**. The training error is the mean rms error between the training target values and the actual neural network output after each epoch. The test error is the mean rms error between the test target values and the actual neural network output. The errors do not have meaningful units since they are made up from mixed quantities. The learning rate (change in training error with each epoch) is a measure of how quickly the network is training. For the present example, the learning rate decays in an exponential manner, leading to a linear error decay characteristic on a log-log error history. Note that after about 400,000 epochs the test error levels out, whilst the training error continues to decrease. Training beyond this point will lead to over fitting of the training data. Training was automatically stopped after  $1 \times 10^6$  epochs and an elapsed time of approximately 2 hours.

The issue of false minima when training using error back propagation was recognised as a potential problem. However, successful repeated network training using different initial random starting weights suggested convergence was well conditioned for the present problem.

## 5. Results and discussion

### 5.1 *Five hole probe data (station 0)*

To provide a baseline with which to compare the calibration based on the full data set; a 5:4:3 network was trained using the ‘five hole probe’ data from the five tappings at station zero on the forebody nose tip. The network reached the stopping criterion after 50,000 epochs with a best test error of 0.026. The calibration accuracy of the trained network based on test data inputs is shown in **figure 9**. In this figure, the height of each column shows the calibration error (vertical axis) for a given air data parameter, at a given alpha and beta (x and y axes), for a given speed. The calibration accuracies measured in terms of rms errors averaged over the three speeds for speed, alpha and beta are 0.322m/s, 0.811°, 0.552° respectively. To put these numbers into perspective, a simple nearest neighbour classification system for the given spacing of the training data would typically be accurate to the mean semi spacing between nodes, i.e. 5m/s, 2° and 0.5° for speed, alpha and beta respectively. It should be noted that reduced spacing of the training data, all other things being equal, would increase calibration

accuracy for any calibration method. This, however, would be at the expense of increased testing costs.

The nature of the errors shown in figure 9 is clearly not wholly random. In particular the speed errors show systematic deviation. Also, in general, errors tend to be largest at the edges of the query data envelope. This is to be expected since the fidelity with which the neural network can interpolate (generalise) at the edges of the training data is reduced compared to the middle. To investigate the repeatability of the errors shown, the original network was reset with random initial starting weights and retrained a further two times. In each case, the distribution of systematic errors was similar, and the averaged rms errors were within 3%.

## **5.2 Increasing data set scope (increased number of tappings)**

To identify the benefits of using additional (theoretically redundant) tappings in the system calibration, further networks were trained using data from stations 0 and 1 (11:6:3 network), stations 0,1 and 2 (16:8:3 network) and stations 0,1,2 and 3 (22:11:3 network). The improvement in calibration accuracy for increasing number of tapping inputs is illustrated in **figure 10**. For the all four stations case, the calibration accuracies achieved are 0.095m/s, 0.015° and 0.085° for speed, alpha and beta, respectively. It is interesting to note that for the all four stations case, the alpha calibration error is slightly worse than for the stations 0,1 and 2 case. This is most likely an artifact of the network training process.

Error surfaces for the all four stations case are shown in **figure 11**, plotted to the same vertical scale as figure 9.

## **5.3 Effect of tapping failures**

With an air data system utilising many pressure tappings, it is of great importance to evaluate the effect of sensor errors. For the present investigation, a sensor failure is modelled as a fixed zero output, i.e. a hard over fault. In terms of the neural network, this pulls the relevant input neuron down to zero activation level. The effect of single sensor failure was investigated by querying a trained neural network with 22 inputs with simulated faulty input data for each of the 22 tappings in sequence. Note that networks were not trained with fault data so an erroneous output is expected. In each case the average rms error for speed, alpha and beta was calculated. To obtain a statistically more meaningful result, the process was repeated for the three different networks trained from different initial random weights and an overall average obtained. The results from this investigation are summarised in **figure 12**.

The initial column on the left-hand side of figure 12 shows the average errors for speed, alpha and beta with no tapping failures. The other columns labeled 1 to 22 indicate the average errors for that particular tapping failure. For the physical location of the tappings please refer to figure 6.

The most important result evident from figure 12 is the large effect that a single tapping failure can have on the accuracy of the air data system. For example, the failure of tapping one on the front of the forebody leads to an increase of the speed rms error from 0.01 to 6.5m/s. The sensitivity of speed error to tapping 1 is not entirely unexpected since tapping 1 acts effectively as a Pitot tube, measuring total pressure. During training, the network has identified the strong correlation between

speed and the output from this tapping and hence has increased the weighting associated with input neuron 1. With this input subsequently disabled the predicted network speed is greatly in error.

Some of the other errors are harder to explain. For example it is not clear why failure of tapping 6 on the right upper side of the forebody at the first station behind the nose consistently leads to such a large error in alpha. On the other hand, it is not clear why failure of tappings 2 and 5 at the nose tip have such relatively small effects.

To further understand the sensitivity to sensor failure, a sensitivity index for speed, angle of attack and sideslip was developed, equation 5.

$$S_{u,n} = \frac{\partial p_n}{\partial u} \quad S_{\alpha,n} = \frac{\partial p_n}{\partial \alpha} \quad S_{\beta,n} = \frac{\partial p_n}{\partial \beta} \quad \text{equation 5}$$

Values for the sensitivity indices were calculated at  $(u, \alpha, \beta) = (37\text{m/s}, 15^\circ, 0^\circ)$  for each tapping, **figure 13**. A large sensitivity index value for a given parameter and tapping number implies that there is a strong correlation between the pressure variation at that tapping and the given parameter. Hence, it might be expected that failure of a tapping with a high sensitivity index for a given parameter would lead to large prediction errors for that parameter. However, comparison of sensitivity index to neural network prediction errors due to tapping failure (figure 12) shows that the sensitivity index is of limited usefulness in predicting the sensitivity of neural network results to tapping failure. The notable exception is that both approaches highlight the importance of tapping 1 for speed measurement.

#### 5.4 Improvements to system robustness

The high sensitivity of the neural network system to individual tapping failures prompts investigation into methods for improving the system robustness. The most straightforward solution is to use preprocessing of the input data to identify obviously faulty readings such as failed high/low. Faulty readings can then be ‘corrected’ by calculating an average between nearest neighbour tappings. In many cases this ‘corrected’ value has a far more benign effect on the network output than the original ‘uncorrected’ error. For example if tapping 6 data is recognized as faulty and simply substituted with data from tapping 7, the resulting errors in the predicted air data parameters is reduced by up to 70%, **figure 14**.

A more robust approach to dealing with faulty or missing input data is to pre-process the input data using an autoassociative network (13). This type of network is trained to reproduce its input at its output. When implemented, input patterns corrupted with noise or with missing data are correctly reconstructed at the output. The results of using a trained multi-layer perceptron autoassociative network with 22 inputs, 8 hidden neurons and 22 outputs to preprocess data with tapping 6 failed is shown in **figure 14**. In this case, the angle of attack error is reduced by over 90% compared to the error obtained using uncorrected query data .

The effect of sensor errors on a nonlinear regression-based calibration algorithm has been studied in reference (16). It is reported that small random sensor errors did not have a significant impact on calibration accuracy. However, for larger hard over type failures as investigated in the present work, the nonlinear regression algorithm failed

to converge correctly and hence led to large errors. This problem was solved in (16) by identifying failed sensors and removing them from the system model.

A further way of increasing robustness to faults is to provide redundancy by dividing the sensors up into a number of different groupings and to train a separate network for each group. To investigate this approach, the sensors were grouped into four subsets. Each subgroup was based on the complete set minus the sensors at one of the four sensor stations along the forebody. The experimental data from these sub groups were then used to train 4 separate neural networks. The accuracy obtained from each of these networks is very close to the accuracy achieved using all 22 tappings, **figure 15**. In practice, preprocessing would be used to identify sensor failures. Data would then be routed to a network trained on a data subset that did not include the faulty sensor.

An important conclusion from the data subset results in figure 15 is that system accuracy is not strongly affected by sensor location. This gives the system designer flexibility in where the sensors are placed. In particular, it is not necessary to place sensors at the nose tip to achieve accurate results. Avoidance of sensors at the nose tip is advantageous in that sensor output in this location is most sensitive to small changes in geometry (through, for example, wear and tear or accidental damage) and from an installation point of view it is desirable to keep all sensors behind the radome.

Note that a fully redundant system would employ an independent sensor set for each network. With this approach, a post-processing system would be used to identify correctly functioning sensor suites based on majority voting.

## 6. Conclusions

- A flush air data system calibration has been implemented using a multilayer perceptron neural network to model the inverse mapping function between surface pressure and airspeed/alpha/beta.
- A network with five inputs assigned to the first five tappings (five-hole probe arrangement) gives rms calibration accuracies of 0.322m/s,  $0.811^\circ$  and  $0.552^\circ$  for speed, alpha and beta respectively.
- A neural network trained with inputs associated with all 22 tappings gives a best accuracy of 0.095m/s,  $0.15^\circ$  and  $0.085^\circ$  for speed, alpha and beta respectively. This is significantly better than the best accuracy that can be obtained using 'conventional' five hole probe data from the nose tip ( $0.3\text{m/s}$ ,  $0.8^\circ$  and  $0.55^\circ$  for speed, alpha and beta respectively).
- A 22 input neural network showed relatively poor robustness to a hard over failure of an individual sensor. Failure of sensor 1 gave an rms speed error of 6.5m/s whereas failure of sensor 6 gave an rms alpha error of  $7.2^\circ$ .
- By preprocessing input data with a trained autoassociative network, the worst case angle of attack error due to single sensor failure was reduced by over 90%. This method provides a simple and effective way of improving calibration robustness with respect to sensor failure.
- Use of any three out the four available tapping stations as neural network inputs does not significantly affect calibration accuracy.

- Accurate results may be obtained *without* using tappings at the nose tip. This enables greater flexibility in the implementation of future flush air data systems.

## **Acknowledgements**

This work was funded by DERA Bedford under contract number ASF/3620. The authors would like to thank Paul Nixon for performing the wind tunnel tests and initial data analysis.

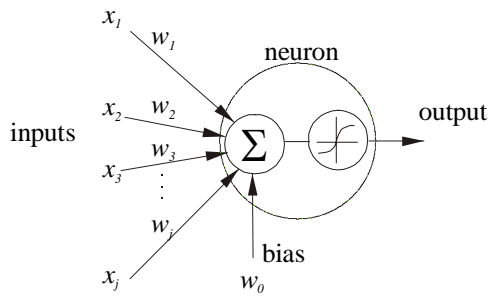
## References

1. **Cobleigh, B. and Del Frate, J.**, Water tunnel flow visualisation study of a 4.4% scale X-31 forebody, NASA TM-104276, Oct. 1994.
2. **Siemers, P., Wolf, H. and Henry, M.**, Shuttle entry air data system (SEADS) – Flight verification of an advanced air data system concept, AIAA Paper 88-2104, May 1988.
3. **Whitmore, S., Davis, R. and Fife, J.**, In-flight demonstration of a real-time flush air data sensing system, *Journal of Aircraft*, Vol. 33, No. 5, 1996, pp. 970-977.
4. **Whitmore, S., Cobleigh, B. and Haering, E.**, Design and calibration of the X-33 flush air data sensing system, NASA TM-1998-206540, Jan. 1998.
5. **Wenger, C., Devenport, W.**, Seven-hole pressure probe calibration method utilizing look-up error tables, *AIAA Journal*, Vol. 37, No. 6, June 1999, pp. 675-679.
6. **Rohloff, T.**, Development and evaluation of neural network flush air data sensing systems, Ph.D. Dissertation, Dept. of Mechanical Engineering, Univ. of California, Los Angeles, CA, May 1998.
7. **Rediniotis, O., Vijayagopal, R.**, Miniature multihole pressure probes and their neural network based calibration, *AIAA Journal*, Vol. 37, No. 6, June 1999, pp. 666-674.
8. **Rediniotis, O. and Chrysanthakopoulos, G.**, Application of neural networks and fuzzy logic to the calibration of the seven-hole probe, *Journal of Fluids Engineering – Transactions of the ASME*, 1998, Vol. 120, No. 1, pp. 95-101.
9. **Minsky, M. and Papert, S.** Perceptrons - An introduction to computational geometry, *MIT Press*, London, 1969, ISBN 0-262-630222
10. **Rumelhart, D., Hinton, G. and Williams, J.**, Learning internal representations by error propagation, In *Parallel Distributed Processing, Explorations in the Microstructure of Cognition*. Vol. 1 foundations, *MIT Press*, London, 1987, ISBN 0-262-18120-7.
11. **Rumelhart, D.E., Widrow, B., Lehr, M.A.** The basic ideas in neural networks, *Communications of the ACM*, Vol. 37, No. 3, pp. 86-92, Mar 1994.
12. **Gurney, K.**, An introduction to neural networks, *UCL Press London* 1997. ISBN 1-85728-503-4
13. **Bishop, C.**, Neural networks for pattern recognition, *Oxford University Press*, ISBN 0-19-853849-9, 1996.

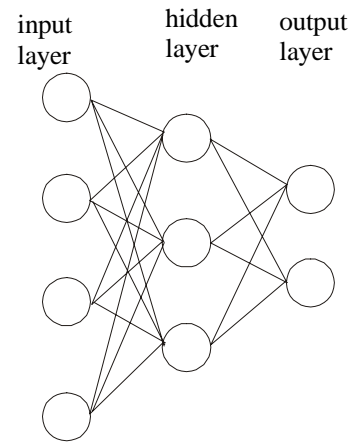
14. **Masters, T.** Practical neural network recipes in C++, *Academic Press*, ISBN 0-12-479040-2, 1993.
15. **Blum, E.** and **Li, L.** Approximation theory and feed-forward networks, *Neural Networks*, 4:4, pp. 511-515.
16. **Whitmore, S. and Moes,** Failure detection and fault management techniques for a pneumatic high angle of attack flush air data system, NASA TM-4335, January 1992.

## List of Figures

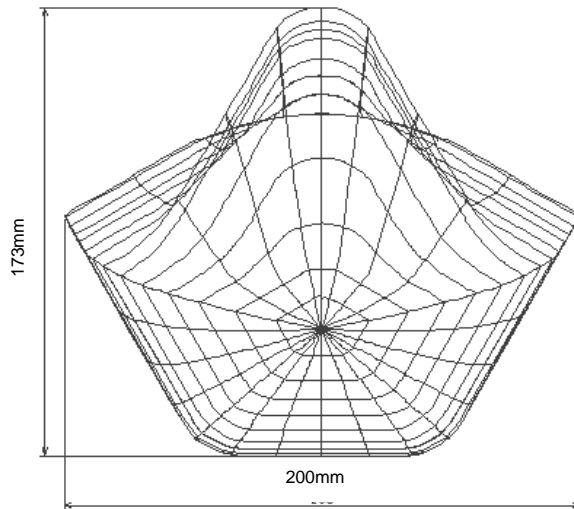
1. Schematic of an information processing node inspired by the biological neuron.
2. Three layer fully connected feed forward neural network, node structure 4:3:2.
3. Chined forebody wind tunnel model view looking aft.
4. Wind tunnel model internal structure.
5. Relationship between geometric angles and aerodynamic orientation.
6. Wind tunnel model pressure tapping layout.
7. Example pressure/sideslip data for the first five pressure tappings at three angles of attack.
8. Typical training error graph for a 22:11:3 network.
9. Error surfaces showing difference between actual and neural network predicted air data parameters. Station 0 data (nose tip) only. 5:4:3 neural network architecture. RMS errors Speed 0.322m/s, alpha 0.811 degrees, beta 0.552 degrees.
10. Increasing calibration accuracy with increasing number of sensors used as neural network inputs.
11. Error surfaces showing difference between actual and neural network predicted air data parameters. All data used for training. 22:11:3 neural network architecture. RMS errors speed 0.095m/s, alpha 0.15° and beta 0.085°
12. Calibration errors due to single sensor hard over failures.
13. Tapping sensitivity analysis.
14. Use of data pre-processing to improve fault tolerance.
15. Comparison of calibration accuracy for different sensor groupings.



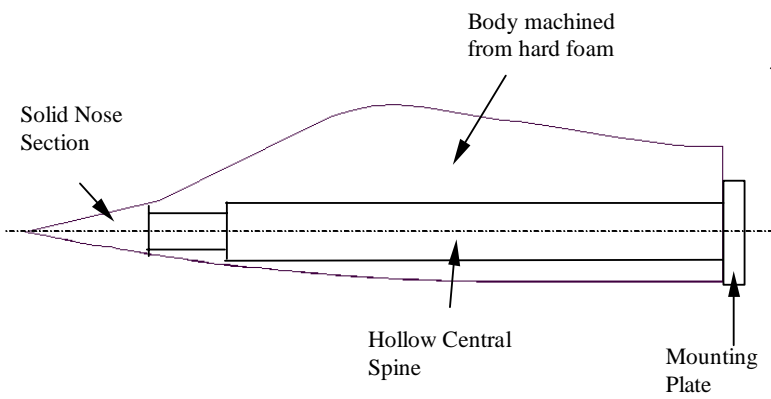
**Figure 1** Schematic of an information processing node inspired by the biological



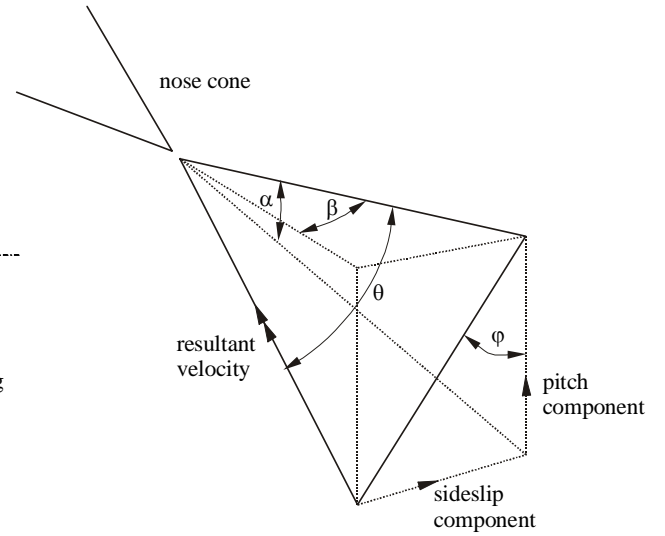
**Figure 2** Three layer fully connected feed forward neural network, node structure 4:3:2



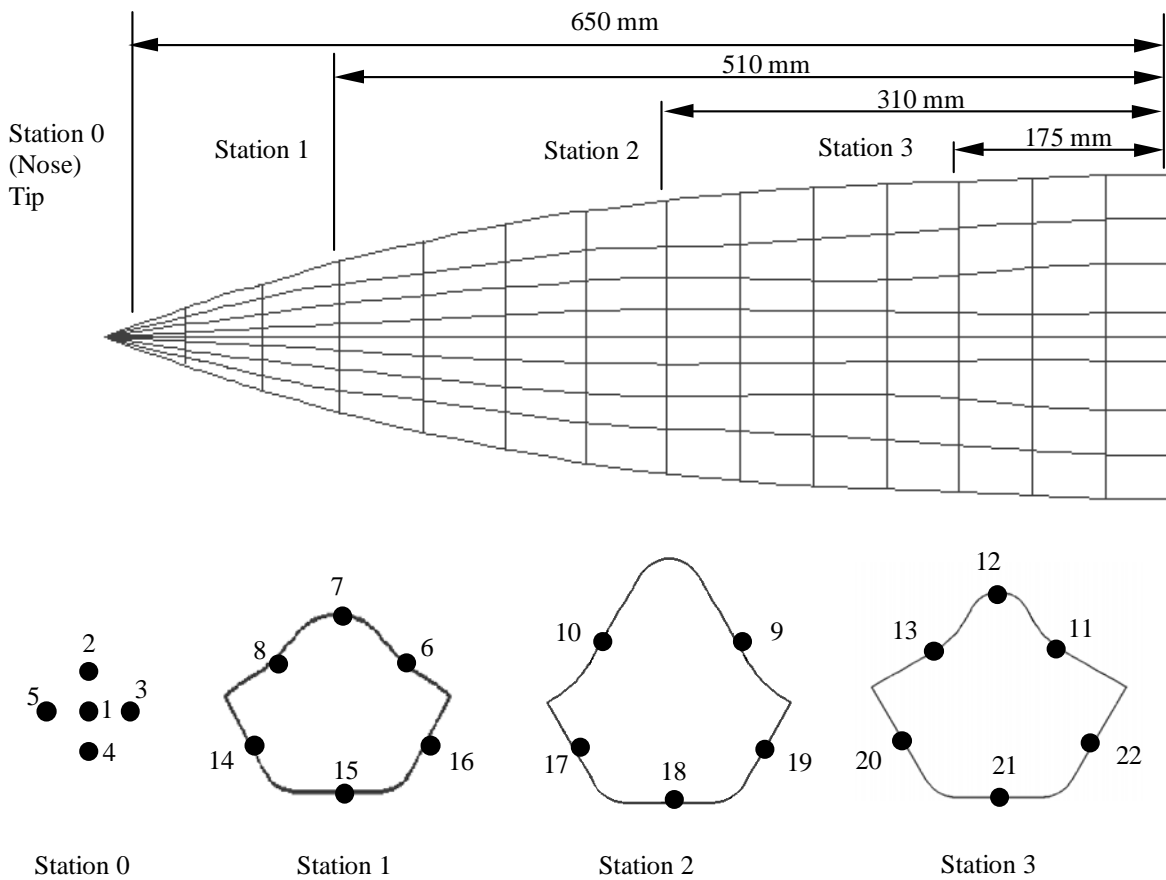
**Figure 3** Chined forebody wind model view looking aft



**Figure 4** Wind tunnel model internal structure

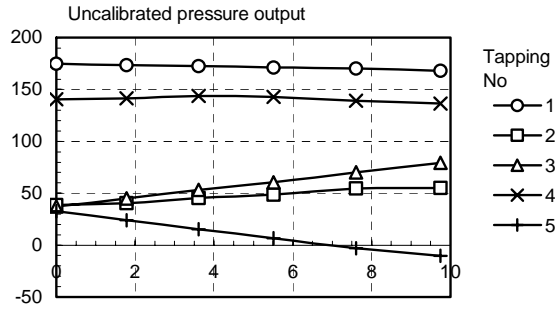


**Figure 5** Relationship between geometric angles ( $\phi, \theta$ ) and aerodynamic orientation ( $\alpha, \beta$ )

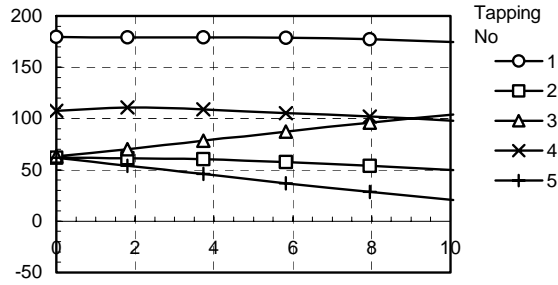


**Figure 6** Wind tunnel model pressure tapping layout. Sections viewed looking aft.

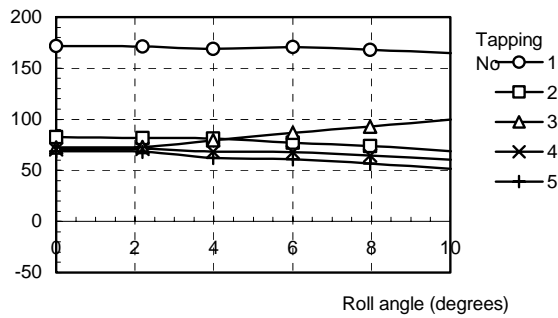
a)  $\alpha = 25^\circ$



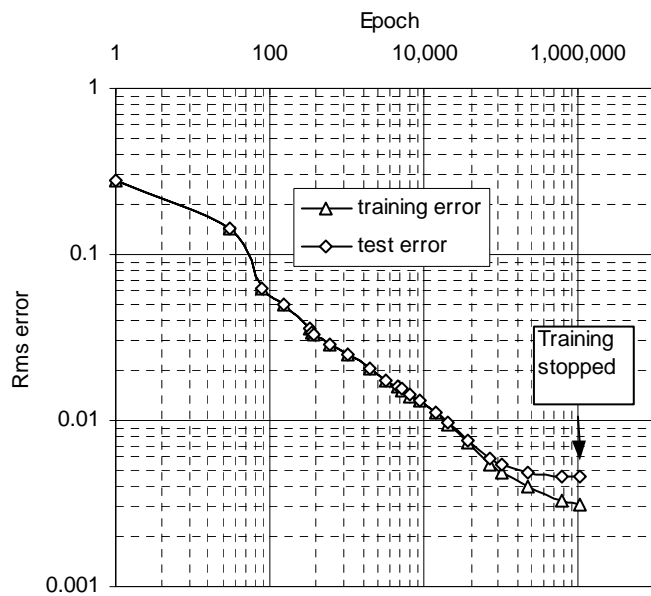
b)  $\alpha = 15^\circ$



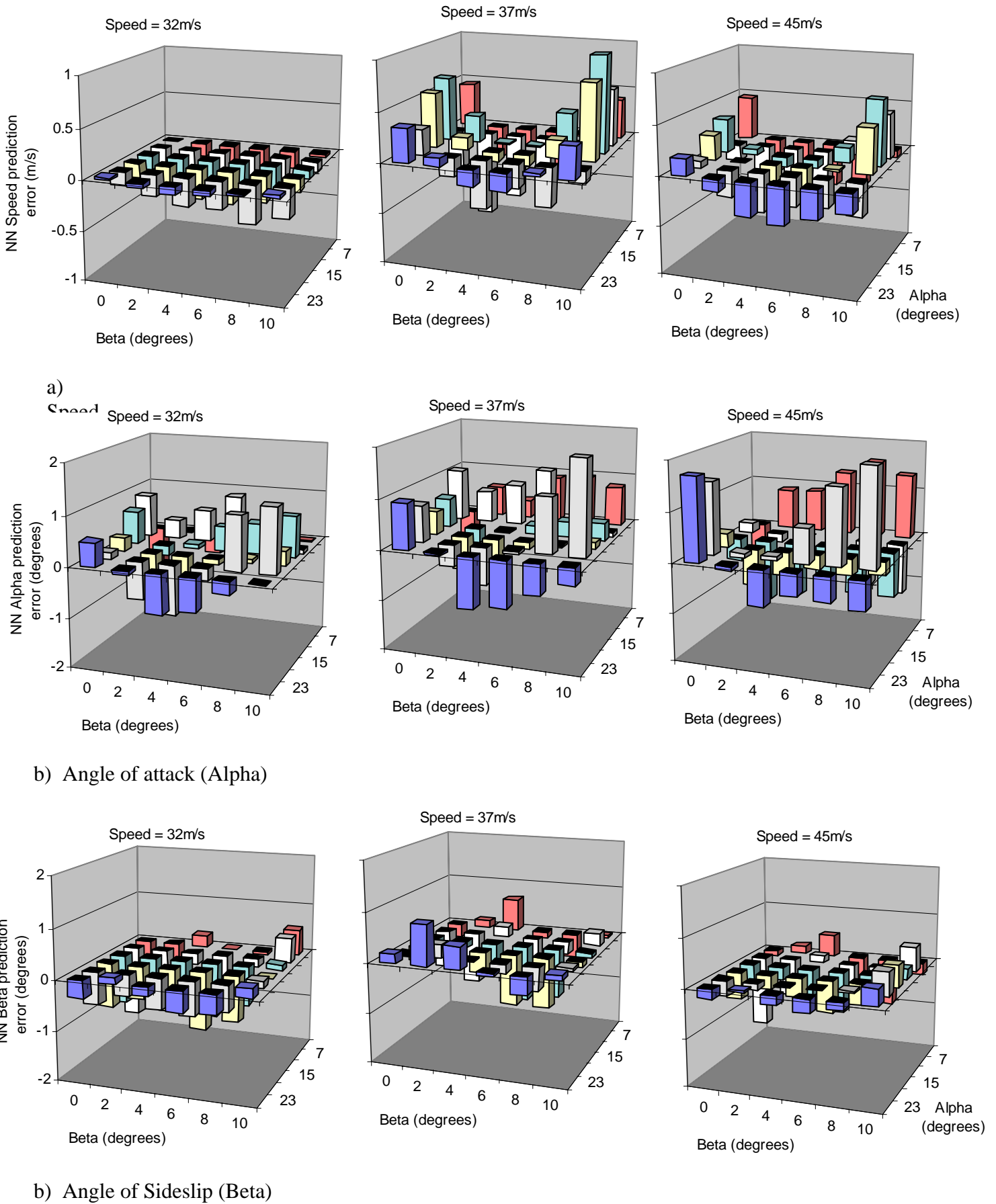
c)  $\alpha = 5^\circ$



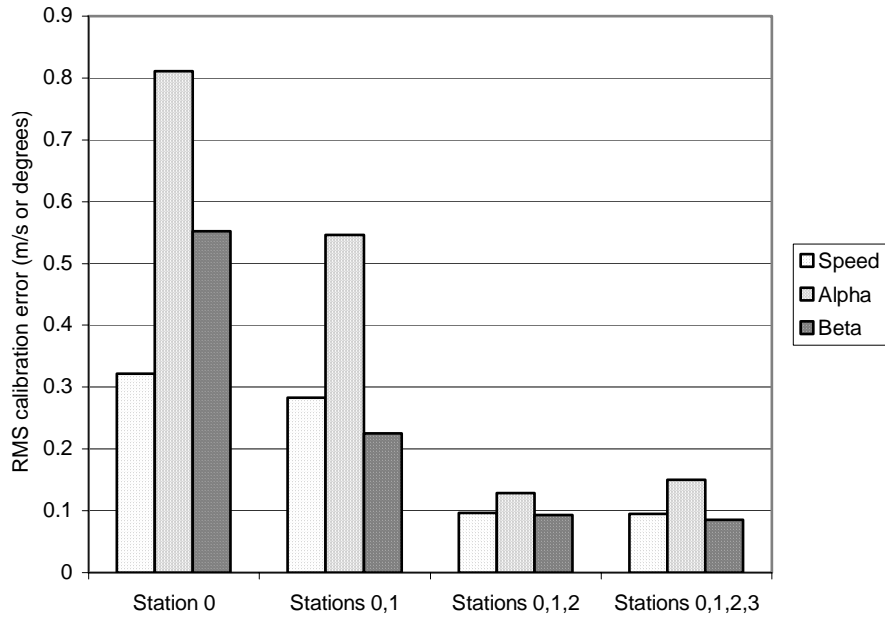
**Figure 7** Example pressure/sideslip data for the first five pressure tappings (station 0) at three angles of attack



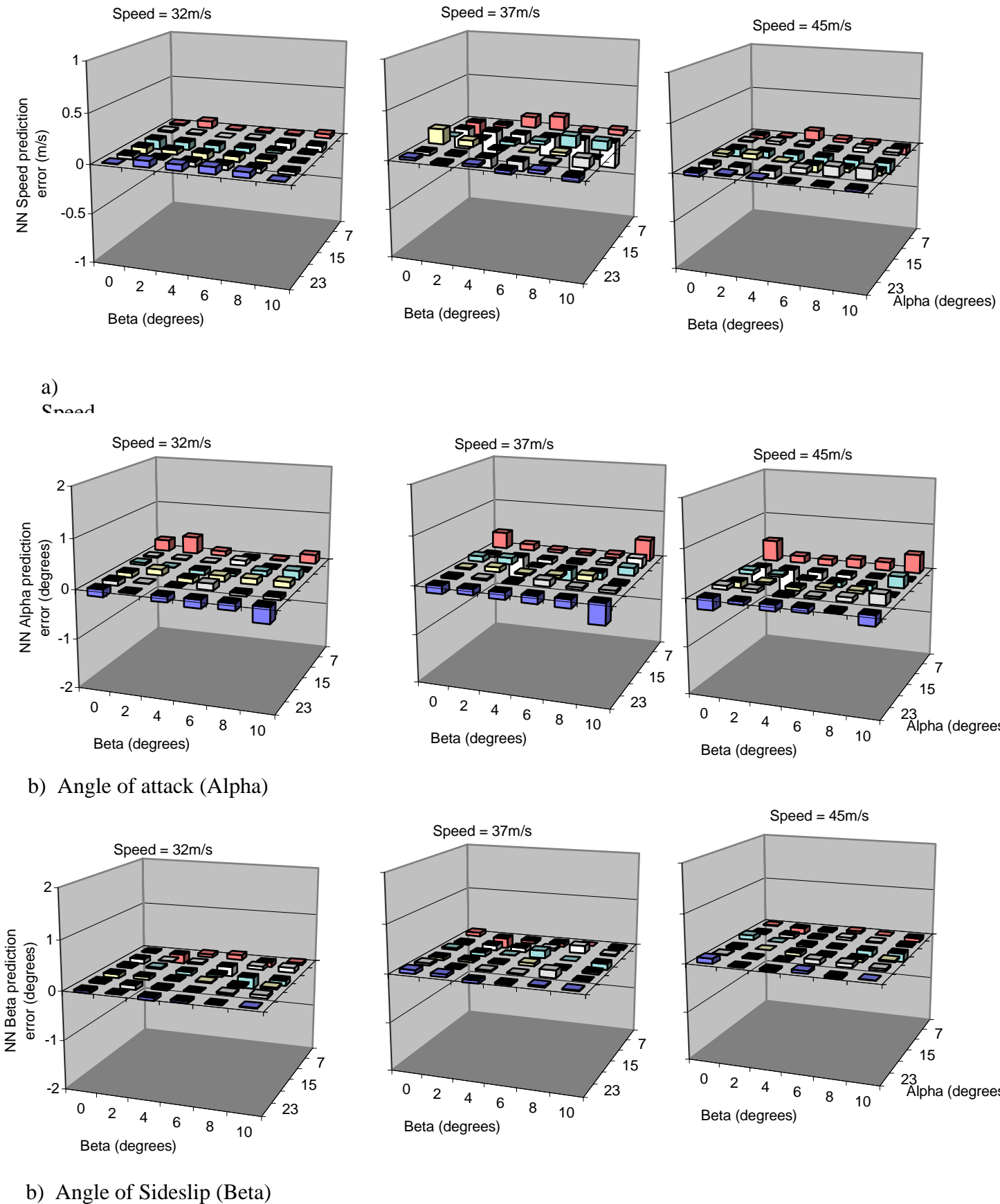
**Figure 8** Typical training error graph for a 22:11:3 network. Training is halted after 100 consecutive increases in the test set error.



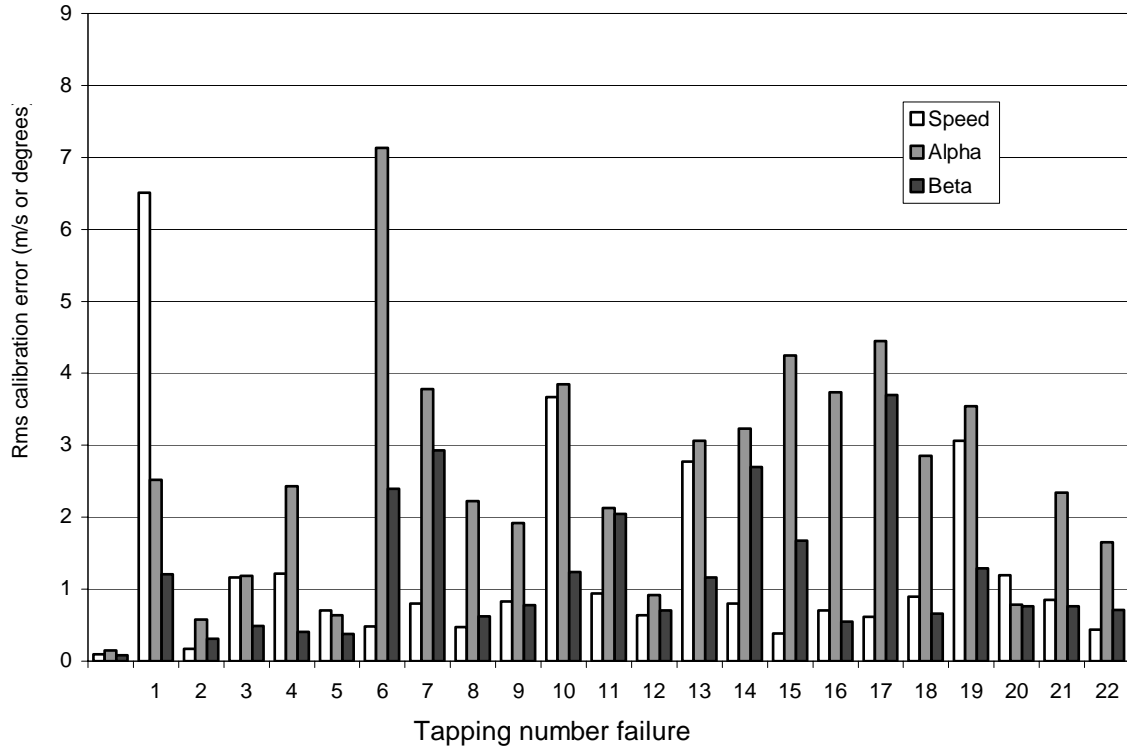
**Figure 9** Error surfaces showing difference between actual and neural network predicted air data parameters. Station 0 data only used for training. 5:4:3 neural network architecture. RMS errors



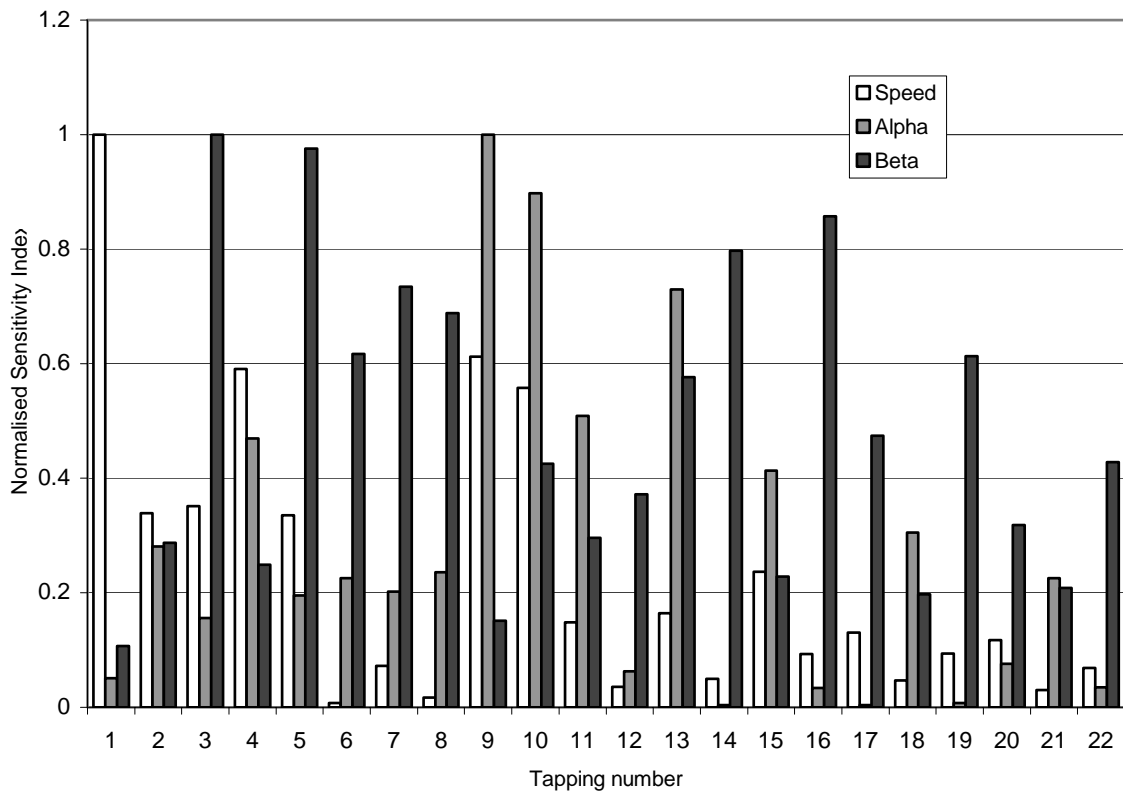
**Figure 10** Increasing calibration accuracy with increasing number of sensors used as neural network inputs



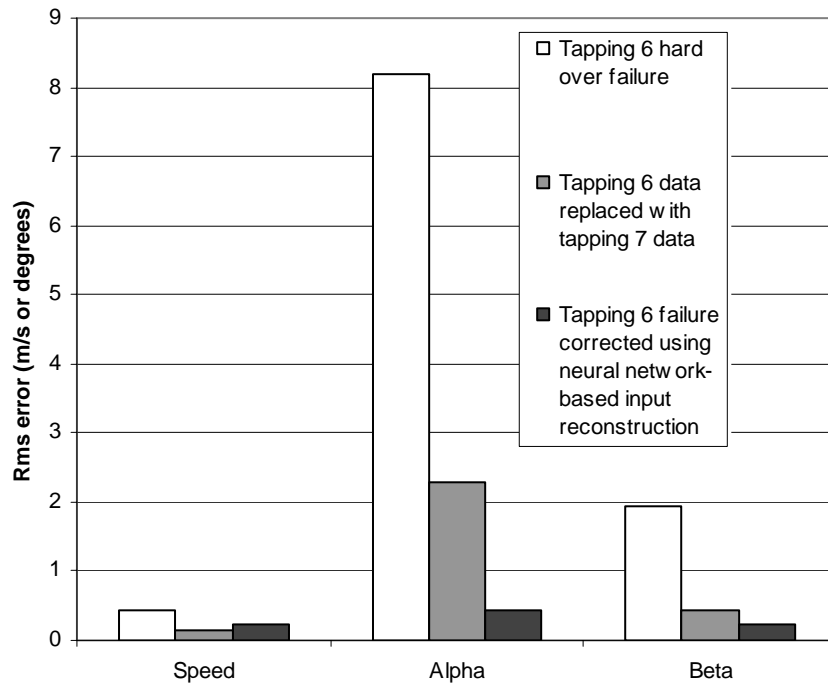
**Figure 11** Error surfaces showing difference between actual and neural network predicted air data parameters. All data used for training 22:11:3 neural network architecture. RMS errors speed



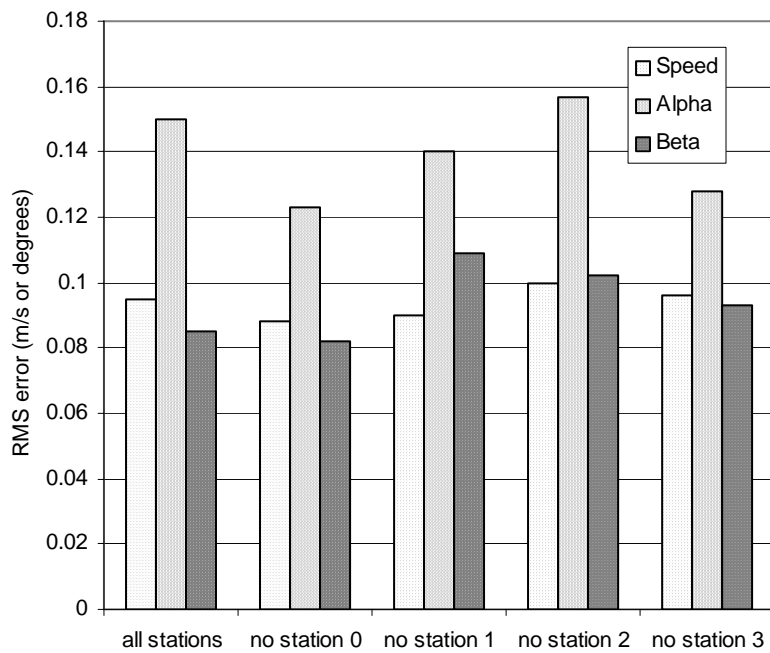
**Figure 12** Calibration errors due to single sensor hard over failures. Left-hand column is calibration error for zero failures



**Figure 13** Tapping sensitivity analysis. For a given parameter (speed, alpha or beta), a high value of sensitivity index for a particular tapping suggests that failure of that tapping would lead to large prediction errors



**Figure 14** Use of data pre-processing to improve fault tolerance.



**Figure 15** Comparison of calibration accuracy for different sensor groupings. Note that acceptable results can be achieved without using station 0 tappings