 CMES, vol.14, no.3, pp.171-180, 2006

The Application of a Hybrid Inverse Boundary Element Problem Engine for the Solution of Potential Problems

S. Noroozi[1], P. Sewell[1] and J. Vinney[1]

Abstract: A method that combines a modified back propagation Artificial Neural Network (ANN) and Boundary Element Analysis (BEA) was introduced and discussed in the author's previous papers. This paper discusses the development of an automated inverse boundary element problem engine. This inverse problem engine can be applied to both potential and elastostatic problems.

In this study, BEA solutions of a two-dimensional potential problem is utilised to test the system and to train a back propagation Artificial Neural Network (ANN). Once training is completed and the transfer function is created, the solution to any subsequent or new problems can be obtained quickly and in real-time without any further modelling or processing time with a high degree of accuracy. This provides substantial savings on computational time and provides instant solutions to a given problem with infinite combinations of alternative boundary conditions. This approach is particularly useful when parametric optimisation of an existing component is required, which may typically involve several iterations in order to obtain valid results. In this paper the inverse problem engine will be explained in detail.

The logic behind its Graphical User Interface (GUI) will be explained and results will be discussed. Using this technique we can for example identify the temperature at the cutting tool tip or on the external surface due to cutting force, from accurate internal temperatures.

keyword: Boundary Element Modelling, Artificial Neural Network, Back Propagation, Linear Extrapolation, Weight Updating.

1 Introduction

Boundary Element Analysis (BEA) is now an established techinique widely used in many industries and fields of research ranging from biomechanics [Muller-Karger, Gonzalez, Aliabadi and Cerrolaza (2001)] to aircraft structural health monitoring [Forth and Staroselsky (2005)]. In terms of structural analysis, the technique is almost exclusively used for discrete event simulation. In practice BEA of a large structure involves the discrete solution for a single time-step. If the effect of changing loads or other boundary conditions is required, this will usually involve expensive and often time consuming re-modelling and/or re-runs of the solution software.

The inverse problem method has been identified as one technique avaialable for reducing the number of model iterations required and/or determining key locations to apply thermocouples. Previous research has utilised the fundemental solution method [Hon and Wei (2005)] and matrix algebraic tools [Ling and Atluri (2006)] for solving inverse heat conduction problems.

Researchers at the University of the West of England (UWE) have also been investigating and developing hybrid solution schemes for solving inverse problems. This has led to a vast amount of research being conducted in the application of inverse methodology in engineering, computing and biomechanics. This Research began in 1995 and since then has been gaining momentum.

This methodology has been successfully applied by the authors to solve inverse experimental (strain gauge and photoelasticity) mechanics such as the determination of the interfacial load at the interface between prosthetic socket and the residul limb for below-knee amputees [Amali, Noroozi, Vinney, Sewell and Andrews (2006)]. The inverse problem engine has also been sucessfully applied to FEA, where both the direct and inverse FEA has been solved in real-time. This was demonstrated in previous papers where the internal stresses, displacements or strain data from FEA analysis was used to determine the load on a simplified aircraft wing [Noroozi, Amali, Vinney and Sewell (2005)]. Research is currently underway to develop a generalised Hybrid Inverse Problem Engine (HIPE), which uses a combination of experimental, numerical and artificial intelligence methods to solve most

[1] Faculty of Computing, Engineering and Mathematical Sciences, University of the West of England (UWE), Bristol, BS16 1QY, UK.

classes of inverse problem.

This paper discusses the application of one version of this inverse problem engine to solve an inverse Boundary Element problem utilising a modified Trefftz method which was first introduced in 1926 [Trefftz (1926)]. The inverse methodology is based on the combined application of Artificial Neural Networks (ANN) and BEA. The proposed solution strategem has been compiled into an automated inverse Boundary Element problem solver with a powerful Graphical User Interface (GUI). It is assumed that the reader is familiar with BEA and its application to both potental and elastostatic problems.

2 Artificial Intelligence - Inverse Problem Analysis

An Artificial Intelligence approach, specifically an ANN, as an Inverse Problem Solver can be utilised for predicting the potential (external temperatures on a body) from the known internal temperatures as has been shown from preliminary evaluation of the inverse technique detailed elsewhere [Noroozi, Amali and Vinney (2001); Noroozi, Amali and Vinney (2002); Noroozi, Amali and Vinney (2003)]. In general, ANNs were established from the study of the biological nervous system. In other words Neural Networks are an attempt at creating machines to behave like the human brain by using components that behave like biological 'neurons'. These elements are all interconnected and work in conjunction with each other, hence the development of the name 'Neural Network'. According to Kohonen (1998), the first theorists to conceive the fundamentals of neural computing were W. S. McCulloch and W. A. Pitts in 1943. An ANN can often be thought of as a black box device for information processing that accepts inputs and produces outputs (Fig. 1).

Inverse problems are classed as problems where the responses (i.e. internal temperatures on a body) are known however the external temperatures, which generated them, are not. The ANN works by forming a mathematical relationship (function) between example data (i.e. internal/external temperatures) which are given to it (Fig. 1). Looping of the network continues until the network response (output) matches the desired response or at least the error function reaches an acceptably small value (i.e. minimisation of the error function). Many input/desired response pairs (training data) are used to train a network. A Backpropogation ANN uses a Mean Square Error (MSE) error function, which is defined as a sum of

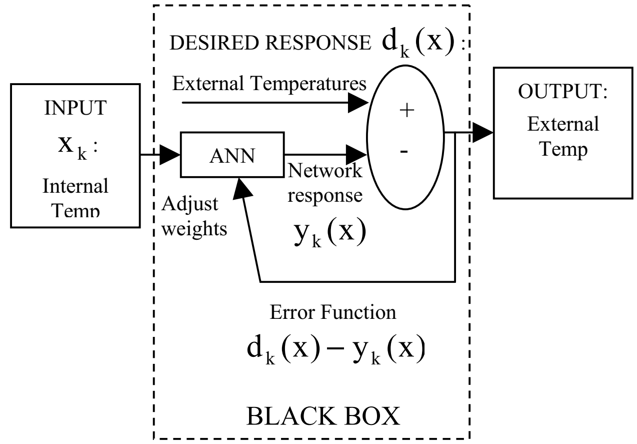


Figure 1 : An Artificial Neural Network (ANN) as a black box for the prediction of external temperatures from internal temperatures

the squared errors between the desired and network response over all the network responses for all the training patterns.

Once the relationship is formed new input data (problem data) can be given to the system, which will output the expected temperatures causing this response. A trained network can generate real-time solutions to any inverse problem, accurately, quickly and indepent of problem size. Results obtained so far using this automated solver now called Hybrid Inverse Problem Engine (HIPE) have been very encouraging [Noroozi, Amali and Vinney (2001); Noroozi, Amali and Vinney (2002)]. A detailed theoretical treatment of the ANN can be found elsewhere [Amali, Noroozi, Vinney, Sewell and Andrews (2001)].

3 Hybrid Inverse Boundary Element Problem Engine Design

Developing an Inverse Problem Engine, utilising an ANN, that is capable of determining the loads on a complex component requires four main areas to be considered based on the discussion in the previous section:

1. *Acquisition of training data* – this should allow the reliable capture and generation of training data (i.e. external temperatures and the internal temperature responses).

2. *ANN architecture* – this should determine an ac-

curate mathematical relationship (function) for the training data in as short a time period as possible (i.e. minimisation of the error function to an acceptable accuracy in the minimum number of loops).

3. *ANN software* – this should enable all the tasks required for network training, utilisation and data analysis to be accessed and performed within a simple and logical GUI.

4. *Acquisition of problem data* – this should allow the reliable capture of problem data (i.e. internal temperatures due to real external temperatures).

The developments in each of these areas will be discussed in the following sections and the resultant system is presented through a case study.

3.1 Acquistion of training Data

To find the relationship between the internal and external temperatures on a body the ANN requires a method of reliably determining the internal temperature responses on the body to be analysed when known external temperatures are applied to that body. The external temperatures and its related internal temperature data must then be stored as ANN inputs and outputs pairs. A large number of these pairs are required for the ANN to accurately train (i.e. find an accurate relationship between the inputs and outputs). A simulation of the body under consideration can be produced with the known external temperatures applied. The Boundary Element program will then determine the internal temepratures at key locations that were caused by the external temepratures.

$$\begin{bmatrix} I11 & I21 & I31 & I41 \\ I12 & I22 & I32 & I42 \\ I13 & I23 & I33 & I43 \\ I14 & I24 & I34 & I44 \\ I15 & I25 & I35 & I55 \\ E1 & E2 & E3 & E4 \end{bmatrix}$$

Figure 2 : Example training file

The number of required external/internal temperature patterns required will be large which means that a large number of simulations with varying external temperatures would be required to gather enough training data. However it was found that the amount of data collected could be dramatically reduced using the theory of superposition to generate training and testing patterns from the independent parent patterns. The example training file in Fig. 2 is for four external temperature surfaces (E1 to E4) and five internal temperatures (I1 to I5) which are captured when the temperature E is applied at that position on the body. This is collected by the software applying a temperature to surface one and collecting the five internal temperatures that are produced on the body surface due to this external temperature. This would be repeated for external temperature surfaces two, three and four.

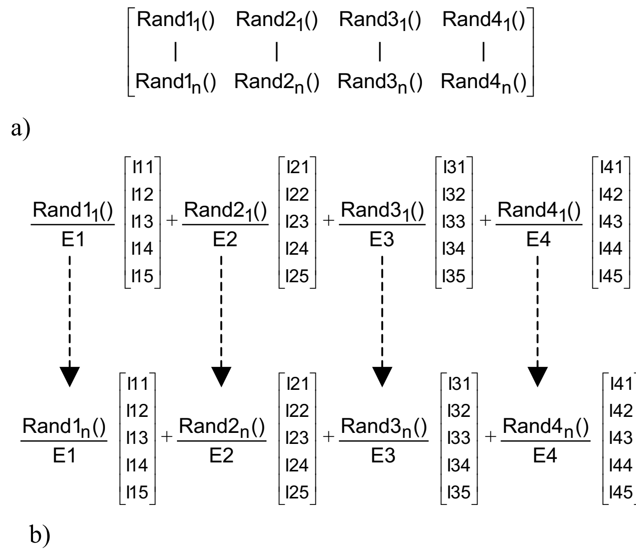


Figure 3 : Generation of a) training output (external temperatures) patterns and b) training input (internal temperatures

Using superposition any number of training patterns (n) are generated (Fig. 3) using a random number generator (rand()) which produces values between the minimum and a maximum temperature value specified by the user. The maximum and minimum values should be the limits of the temperature that can occur on the body.

3.2 ANN Architecture

The architecture of the ANN (i.e. the number of layers, the amount of training and testing data, the values assigned to the network learning variables) will determine the accuracy of the predicted temperature data. The setting of the architecture should require as little intervention from the user as possible. Ideally the architecture of the ANN should be preset and be suitable for all Boundary Element problems. The full ANN architecture used for the inverse problem engine can be found in Tab. 1.

Another important factor in accurately and efficiently training an ANN is to determine the number and positions of the internal temperature readings required to predict the external temperatures. It is recognised that to successfully train a Backpropagation ANN the layers in the network should be convergent (i.e. the number of inputs should be greater than the number of outputs). If the converse is true the network would be divergent and the network could not be trained. The positions of the internal temperature readings are also important as some regions of a body are much more sensitive to external temperature changes than others. To produce efficient training data the internal temperature data should be captured at the sensitive regions.

The number of required patterns was found to be 1420 which meant that 1420 sets of random external temperatures on each surface and the resultant internal temperatures caused by these external temperatures were required to provide enough input/output data to find the relationship between them.

Rarely is the data collected from Boundary Element Analysis completely accurate or data captured from thermocouples free from noise as there are many possible sources in a normal working environment. Therefore the software adds further noisy patterns to the training files to account for any noise that may come from the inaccuracy of the model or from thermocouple data. The number of noisy patterns, maximum absolute value of noise to add to the surface data and the percentage of each pattern to modify are set in the ANN architecture.

3.3 ANN Software

The ANN software must make the process of collecting, generating, training and analysing ANN data as simple as possible for the user who may not have any previous experience of ANN analysis. This has been achieved by logically ordering and standardising the ANN software tasks (Fig. 4).

All data generated from each task is automatically transferred to other tasks that require this data. Collected data is hidden from an inexperienced user but is available if required for troubleshooting purposes. The data analysis software is also simple to interpret, is referenced to the original model.

Table 1 : Architecture of the Artificial Neural Network

Architecture	Feed Forward Back propagation
Data process	Normalisation
Noise generator	+/-5 degrees on 10% of training patterns
Range of Temperatures	user-defined minimum and maximum
Number of nodes in input layer (Internal Temperature Nodes)	Defined in the Boundary Element Model
Number nodes in output layer (External Temperature Surfaces)	Defined in the Boundary Element Model
Number of nodes in hidden layer	Equal to the number of layers in the output layer
Number of training patterns	1420
Number of testing patterns	50
Number of problem patterns	1
Number of loops	Defined by user
Learning rate	0.0005
Momentum constant	0.0003

3.4 Acquisition of Problem Data

Problem data is the captured internal temperature data on the body due to the real external temperatures. It is introduced to the trained network so that it can predict the external temperatures on the body's surface. It is essential that the internal temperature data is captured at identical locations for both the training and problem data, if not the results will be invalid. Therefore this data should also be collected using the same Boundary Element model or from thermocouples attached to the actual body being analysed in the same position as for the collection of training data.

4 Hybrid Inverse Boundary Element Problem Engine Case Study

A study was performed to assess the developed system which is detailed in the following sections. The problem chosen for the case study is shown in Fig. 5 where

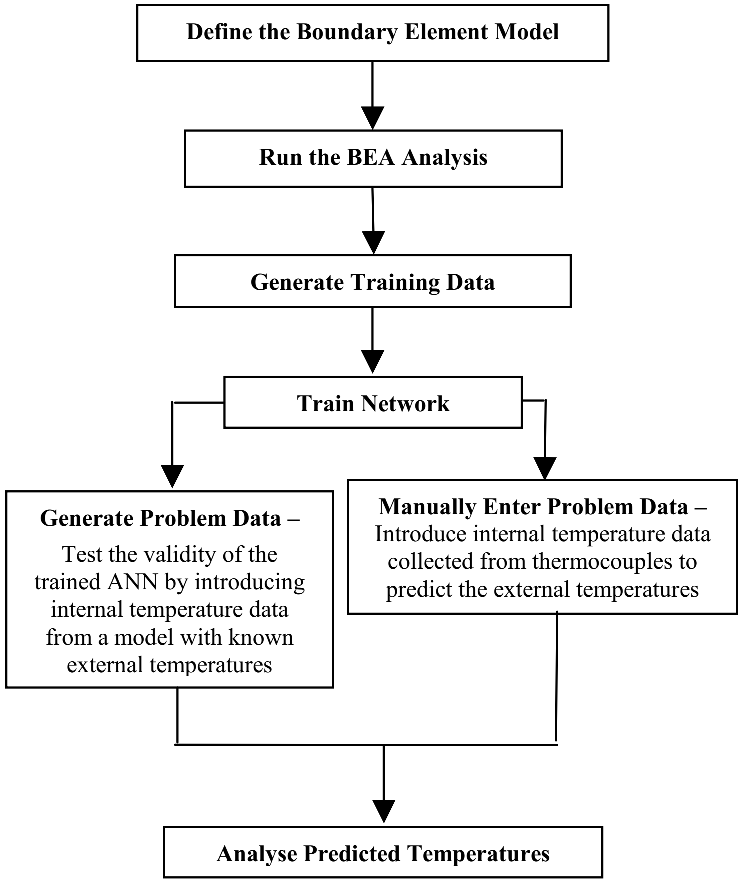


Figure 4 : Software flowchart

a rectangular plate is heated to 1000^{o}C on the left edge, -500^{o}C on the right edge and insulated on the other two sides. The aim of the case study is to determine the usability and accuracy of the Inverse Problem Engine by comparing the actual temperatures on the left and right edge found using direct analysis with the predicted temperatures found using the Inverse Problem Solver.

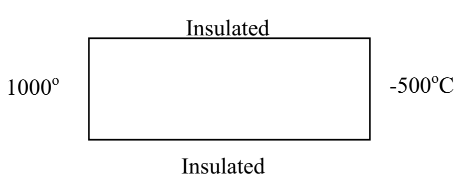


Figure 5 : Schematic diagram of the plate with typical boundary conditions

4.1 Define the Boundary Element Model

The model geometry, boundary definitions and internal nodes (Fig. 6b) are defined by creating a text file in the format shown in Fig. 6a.

The locations of the internal nodes to be analysed must be chosen for their sensitivity to changes in the external temperatures as discussed previously. It must also be ensured that there are more internal nodes defined than temperature surfaces to produce a convergent network. Eighteen internal nodes (network inputs) have been defined for this problem which is more than the four surface temperatures (network outputs).

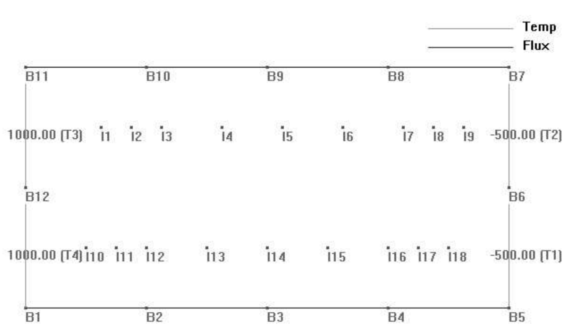
The defined surface temperatures will determine the maximum and minimum temperature range the network will train between. In this problem the ANN will generate training data between -500 and 1000^{o}C at locations T1, T2, T3 and T4.

4.2 Running the Inverse Problem Solver

The solver has been developed into a windows-based graphical interface which can be accessed via an executable file. When first entered the user is asked to select

HEAT FLOW USING TREFTZ	Title	
12 18	Number of: external nodes internal nodes	
1.25 3.0		
1.75 3.0		
2.25 3.0		
3.25 3.0		
4.25 3.0		
5.25 3.0		
6.25 3.0		
6.75 3.0		
7.25 3.0		
1.0 1.0	Coordinates of internal nodes	
1.5 1.0		
2.0 1.0		
3.0 1.0		
4.0 1.0		
5.0 1.0		
6.0 1.0		
6.5 1.0		
7.0 1.0		
0.0 0.0		
2.0 0.0		
4.0 0.0		
6.0 0.0		
8.0 0.0		
8.0 2.0		
8.0 4.0	Coordinates of external nodes	
6.0 4.0		
4.0 4.0		
2.0 4.0		
0.0 4.0		
0.0 2.0		
1 0.0		
1 0.0		
1 0.0	1 = Flux	0.0
1 0.0		
0 -500	0 = Temp	Temp value
0 -500		
1 0.0		
1 0.0		
1 0.0		
1 0.0		
0 1000		
0 1000		
****	Marks end of file	

a)



b)

Figure 6 : a) Model file and b) final model

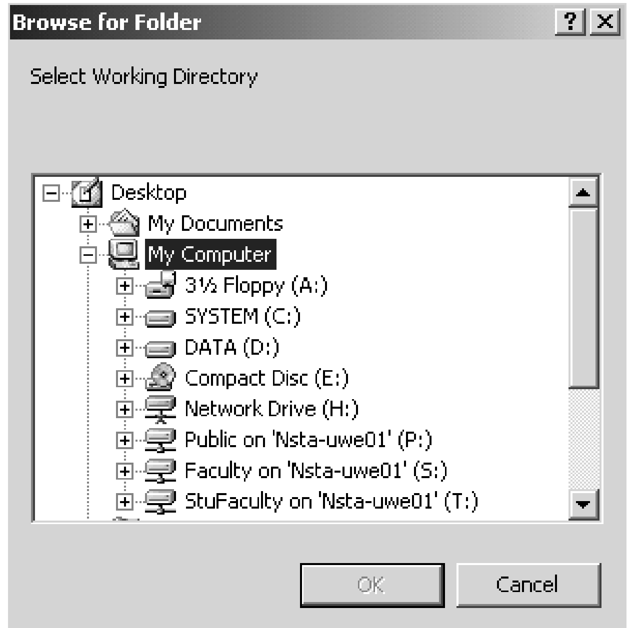


Figure 7 : Selecting a working directory

a working directory where all files related to the analysis will be stored (Fig. 7).

The main dialog window will then appear (Fig. 8) where all functions relating to the solving of the inverse problem can be accessed. The model file can be loaded by clicking the '*Load Input Data*' button. The contents of the file will appear in the Model Details window for review.

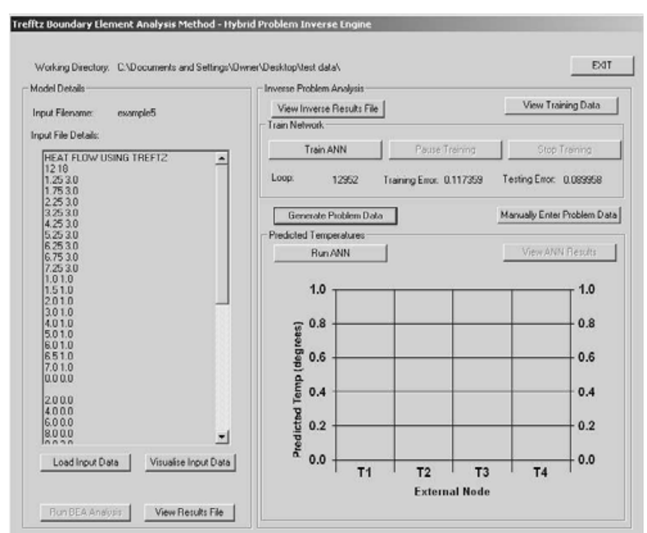


Figure 8 : Main dialog window

4.3 Run the BEA Analysis

Figure 9 : Dialog appears when analysis is successful

To find the direct solution to the model click the '*Run BEA Analysis*' button. If the analysis is successful the dialog box shown in Fig. 9 will appear. It is then possible to view the results file detailing the internal temperatures by clicking on the '*View Result File*' button.

4.4 Generate Training Data and Train Network

For inverse analysis it is required that the model is run several times to gather the superposition data. In this case four runs are required as the temperature is to be predicted in four locations (T1 to T4), where:

run 1: T1= maximum external temperature value and T2 to T4 are set to zero.

run 2: T2= maximum external temperature value and T1, T3 and T4 are set to zero.

run 3: T3= maximum external temperature value and T1, T2 and T4 are set to zero.

run 4: T4= maximum external temperature value and T1 to T3 are set to zero.

The results of each run can be viewed by clicking the '*View Inverse Results File*' button. From these runs a training file of the form shown in Fig. 2 will be created and which can subsequently be utilised to produce the required training patterns to train the ANN. The created training data file can be viewed by clicking the '*View Training Data*' button.

Training of the network is performed by clicking the '*Train ANN*' button (Fig. 10a) which when pressed will firstly generate the specified number of training patterns from the generated training data file and then will perform training. Training can be manually paused or stopped using the '*Pause Training*' and '*Stop Training*' buttons. The training will stop automatically if the testing error goes below 10% (i.e. the network should predict accurately to within 90% of the actual temperature value) as the network will be deemed to be trained at this error level. Fig.10b shows the reduction in error as the network trains for the model in Fig. 6. The graph shows the error reaches 0.1 in approximately 13000 loops (i.e. iterations of the network). This took approximately five minutes to achieve.

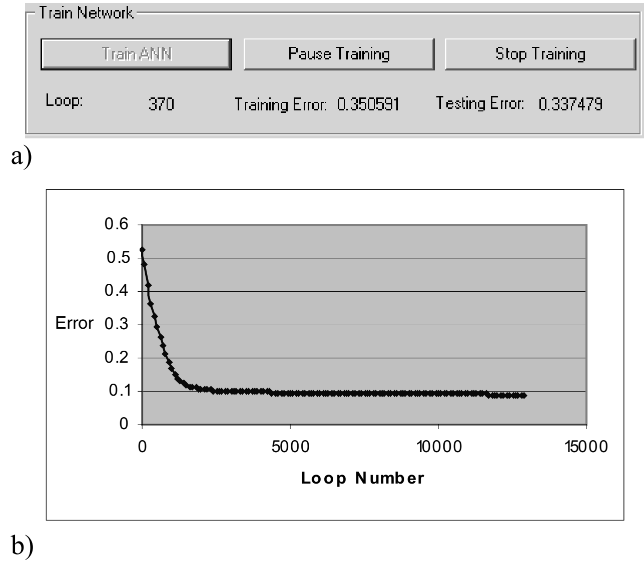


Figure 10 : ANN training a) dialog window and b) graph

4.5 Generate Problem Data

The validity of the trained network can be assessed by entering internal temperatures as problem data for known eternal surface temperatures and ensuring that the ANN predicts these correctly. Click the '*Generate Problem Data*' button to enter four surface edge temperatures between the maximum and minimum values set in the model (Fig. 11). The program will then perform a **direct** BEA analysis of the model using the entered values of edge temperatures. The calculated internal temperatures from this model will then be used as the input problem data to the network.

4.6 Manually Enter Problem Data

Click the '*Manually Enter Problem Data*' button to enter the temperature for each internal node of the model in a dialog window (Fig. 12). As there are eighteen internal nodes in the model eighteen internal temperatures will need to be specified to predict the temperatures on the four surfaces that caused them.

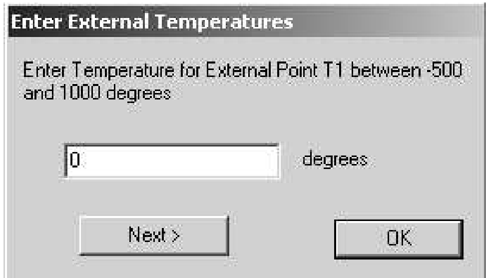


Figure 11 : Generating problem data dialog window

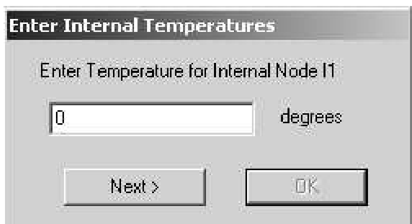


Figure 12 : Entering problem data dialog window

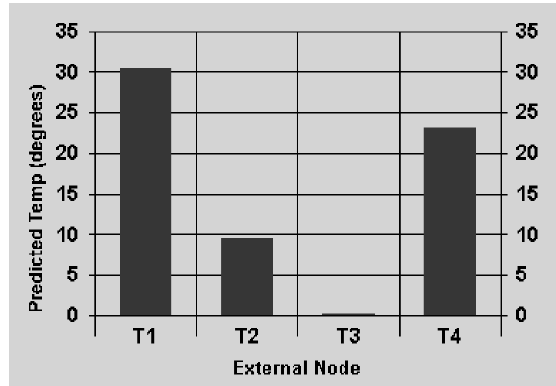


Figure 13 : Predicted external temperature graph

4.7 Analyse Predicted Temperatures

Once the problem data has been entered it is possible to use the trained network to predict the external temperatures caused by the given set of internal temperatures (problem data). Clicking the '*Run ANN*' button will produce a graph of the predicted external temperatures as shown in Fig. 13. Column T1 to T4 represents the predicted temperature on surfaces T1 to T4 as defined in the model file (Fig. 6).

To validate the trained ANN the '*Generate Problem Data*' function was used to apply six sets of four external temperatures (Actual Temperatures) generating six problem patterns. The problem patterns were then introduced to the network and the predicted external temperatures that caused them found (ANN Temperatures). The comparison of the Actual and ANN predicted temperatures can be found in Fig. 14.

The results in Fig. 14 clearly shown that the ANN can predict four external temperatures caused by eighteen internal temperatures with a high level of accuracy within the specified temperature range.

5 Discussion

The problem engine can solve both direct and the inverse problems with ease, however due to current fast computing powers available, the direct solution can easily be performed using multiple runs of the same problems with varied Boundary conditions. However inverse problem, in a classical way requires a vast number of iterations, which even with high-speed computing power can still be time consuming and there is no guarantee of a unique solution. This is when the Hybrid Inverse Problem Engine (HIPE) comes to its own and identifies a unique solution.

In industrial applications, it will be quite time consuming to generate surface temperature for over 1400 sets of training data using thermocouples. However, an optimised BEA model of a thermal problem can easily be validated by the data obtained from thermocouples located at strategic location on the surface. Once it is sure that BEA is a true representation of the industrial case then, combined with superposition, it can be used to generate all the required training data quickly and efficiently. This way the real-time solver can easily be generated in an industrial enviroment. This ability is invaluable for real-time thermal analysis and monitoring of processes where the knowledge of temeperature distributions are key factors and can change randomly (i.e. the temperature at a cutting tool tip or on an external surface due to cutting force).

So far this system has only been used for steady-state thermal conduction problems however, the system can easily be used or the network can easily be trained to have time dependent parameters as input data. Here the validation and convergence may take longer but the benefits it can offer in the long term can by far outweigh the intial setup.

Once trained the real-time inverse problem solver is in-

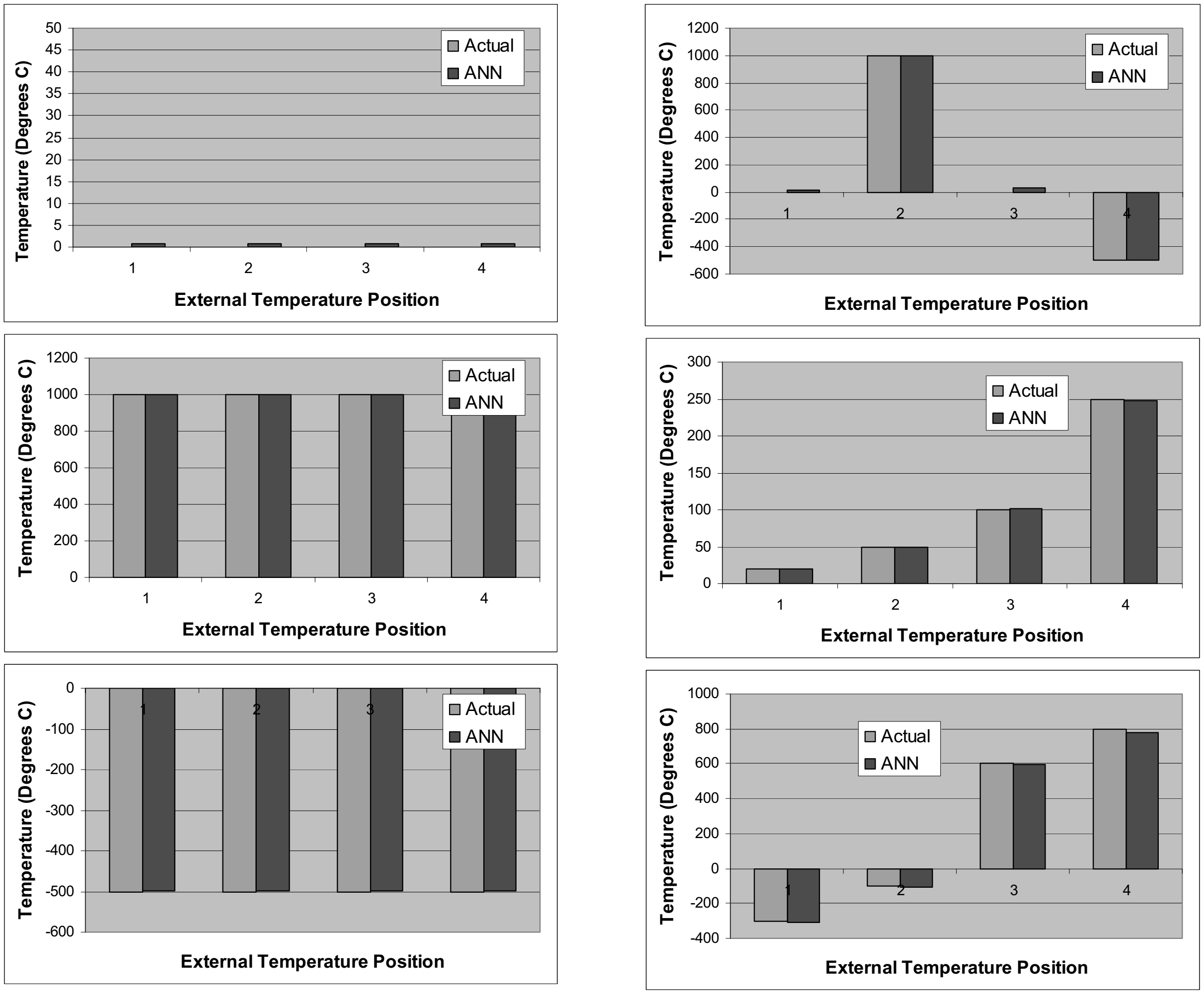


Figure 14 : Comparison of actual and predicted external temperatures for six problem patterns

dependent of the size of the problem therefore the solution to any inverse thermal monitoring problem will be quick and the outputs can be almost instantaneous. This makes the application of such a numerical system in control system, processes or in an embeded real-time system a possibility.

The main focus of this paper was to demonstrate how fast and efficient this hypothesis is. Further work may be undertaked to improve the solution integrity and enhance its reliability and accuracy. Other objectives will be to assess its potential benefits in terms of time, cost and relative accuracy.

6 Conclusions

A tool has been developed and tested through a case study that enables the user to solve both direct and inverse Boundary Element problems with ease. The software developed provides a simple to use interface, which allows simple analysis of the obtained results. A summary of the main development and achievements to date are detailed below, the developments are:

- an easy to use Graphical User Interface (GUI).
- a training methodology.
- a Boundary Element specific Inverse Problem Engine.

A critical appraisal of the system has highlighted several areas that require future investigation, which include:

- the development of a thermocouple data acquisition collection interface.

- investigation of the technique utilising time dependent parameters.
- the assessment of the potential of the tool for investigation of complex problems.

If these recommendations for further investigation can be achieved a complete and fully robust system could be developed.

References

Amali, R.; Noroozi, S.; Vinney, J.; Sewell, P.; Andrews, S. (2001): A Novel Approach for Assessing Interfacial Pressure between the Prosthetic Socket and the Residual Limb for Below Knee Amputees using Artificial Neural Networks. *International Joint Conference on Neural Networks*, Washington DC, 15-19 July 2001.

Amali, R.; Noroozi, S.; Vinney, J.; Sewell, P.; Andrews, S. (2006): Predicting Interfacial Loads Between the Prosthetic Socket and the Residual limb for Below-Knee Amputees - A Case Study. *Strain*, vol. 42(1), pp. 3-10.

Forth, S.C.; Staroselsky, A. (2005): A Hybrid FEM/BEM Approach for Designing an Aircraft Engine Structural Health Monitoring. *CMES: Computer Modeling in Engineering & Sciences*, vol. 9(3), pp. 287-298.

Hon, Y.C.; Wei, T. (2005): The method of fundamental solution for solving multidimensional inverse heat conduction problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 7(2), pp. 119-132.

Kohonen, T. (1998): An introduction to neural Computing. *Neural Networks* 1998, vol. 1(1), pp. 3-16.

Ling, X.; Atluri, S.N. (2006): Stability analysis for inverse heat conduction problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 13(3), pp. 219-228.

Muller-Karger, C.M.; Gonzalez, C.; Aliabadi, M.H.; Cerrolaza, M. (2001): Three dimensional BEM and FEM stress analysis of the human tibia under pathological conditions. *CMES: Computer Modeling in Engineering & Sciences*, vol. 2(1), pp. 1-14.

Noroozi, S.; Amali.; R, Vinney, J. (2001): The application of combined Artificial Neural Network and Boundary Element for potential problems. *Joint Meeting of the International Conference on Boundary Element Techniques and International Association for Boundary Element Methods*, Rutgers University, NJ, USA, 16-18 July 2001, pp. 47-56.

Noroozi, S.; Amali, R.; Vinney, J. (2002): Development of a generalized inverse problem engine for boundary element analysis of potential problems. *3rd International Conference on Boundary Element Techniques and International Association for Boundary Element Methods (BETEQ2002)*, Beijing, China, 10-12 September 2002, pp. 169-175.

Noroozi, S.; Amali, R.; Vinney, J. (2003): Combined BEA and ANN for the analysis of 3D inverse boundary domain problems. *4th International Conference on Boundary Element Techniques and International Association for Boundary Element Methods (BETEQ2003)*, University of Granada, Spain, 15-17 July 2003, pp. 439-444.

Noroozi, S.; Amali, R.; Vinney, J.; Sewell, P. (2005): Use of Artificial Neural Networks for real time determination of loads on in-service components. *International Journal of Engineering Simulation*, vol. 40, pp. 73-77.

Trefftz, E. (1926): Ein Gegenstuck zum Ritz'schen Verfahren. In: *Proceedings of the Second International Congress on Theoretical and Applied Mechanics*, Zurich, Switzerland, 12-17 September 1926.