Publishing House
AKAPIT

COMPUTER METHODS IN MATERIALS SCIENCE
Informatyka w Technologii Materiałów

Vol. 6, 2006, No. 3-4

THE USE OF AN ARTIFICIAL NEURAL NETWORK TO ESTIMATE TOOL COSTS IN COLD ROLL-FORMING PROCESSES

ANTHONY DOWNES, PETER HARTLEY*

Department of Mechanical Engineering, University of Birmingham, Edgbaston, Birmingham, B15 2TT, UK
*Author for correspondence: email: p.hartley@bham.ac.uk

Abstract

The cold roll-forming industry is extremely competitive and the majority of work tenders that are submitted are unsuccessful. There are several issues that influence the tool costs, but the central problem lies in predicting the number of rolls that are required to roll-form the section and therefore determine the forming machine size. This paper discusses a method that assists tool cost estimations in cold roll-forming processes. The objective is to reduce the cost of generating work tenders whilst ensuring that the accuracy of the cost estimation is maintained, or improved. To facilitate this approach a LISP program was developed to process AutoCAD drawings of section geometry and to evaluate selected section features such as the total number of bend regions. The section features were then processed by an artificial neural network that was trained to predict the size of the forming machine that would be required to roll-form the section.

Key words: cold roll forming; artificial neural networks; lisp

1. INTRODUCTION

In a highly competitive industrial marketplace it is likely that most work tenders will be unsuccessful, and this is the case in the cold roll-forming industry. When large numbers of work tenders are submitted the cost of producing them becomes significant. This paper discusses the development of a computational system involving the automatic identification of geometric features together with an artificial neural network approach to reduce the overall cost of generating work tenders. The most important aspect of the work tender is the estimation of the overall tooling costs. This figure will normally become part of the contract with the customer and is not renegotiated at a later date. If it is inaccurate it may result in significantly smaller profit margins than planned. When there is a work shortage the profit margins are often reduced to increase the likelihood of winning the order. Therefore it is essential that cost estimations are accurate to avoid financial losses. Every work tender must be regarded as important because it may be the one that wins highly profitable work.

Cold roll-forming is an efficient method for the manufacture of long thin-wall sheet metal products. The resourceful design of the section geometry will result in an excellent strength to weight ratio and resistance to local buckling, [Castellucci et al., 1997].

A vast range of section geometry can be produced, from 1.5 metre wide profiles to narrow profiles of less than 50 mm. There is no limitation on the complexity of the section geometry, although a very large forming machine consisting of many forming stations will be required for a section with a considerable number of bend regions. The sheet metal is guided through sets of profiled rolls at room temperature which progressively form the sheet to the desired section geometry. The objective is to manufacture bend regions in the sheet, but unwanted

ISSN 1641-8581

strains are generated which are inherent in the roll-forming process. If the unwanted strains become excessive, geometric defects in the section will result, [Fewtrell, 1990]. Longitudinal curvature and twist is a common quality problem with roll-formed products. Additionally, defects at the edge of the sheet such as edge wave and wrinkles may occur and will often be difficult to remove. To reduce the unwanted strains it is important that every bend is formed gradually. Unless the bend angle is relatively small, 15 degrees for example, each bend is formed by several sets of profiled rolls. For example, the forming of a 90 degree bend will typically require 6 to 8 sets of rolls and therefore there will be 6 to 8 forming stations on the machine which will be involved in forming the bend.

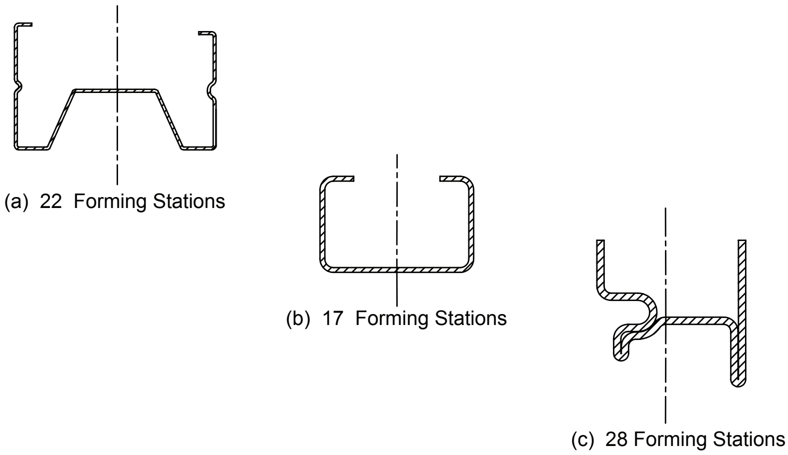

Figure 1. Examples of Roll-Formed Products

In figure 1(a), a section geometry with 8 bend regions and two ribs is shown, which was roll-formed using 22 forming stations, [Hadley Group, 2006]. There are only 4 bend regions in the section shown in figure 1(b) but this required 17 forming stations and the complicated section geometry shown in figure 1(c) was roll-formed using 28 forming stations. The average cost of a forming station using current prices is used to calculate the overall cost of the tooling. Therefore the first step in the tool costing process is to determine the number of forming stations that will be required. There are several features that are not related to the section geometry that cannot be overlooked. For example, the sheet feed method, bend angle tolerances and the sheet material mechanical properties, which will all impact on the number of forming stations. However, it is common industrial practice to construct a set of rules to adequately cover these conditions, whereas the prediction of the number of forming stations from studying the section geometry is the more complex task. Unless the section geometry is simple there usually are several bending sequences which may be chosen. One option is to select the two bends closest to the sheet edge in the section shown in figure 1(b) as the first bends to form. An opposing viewpoint is that it is beneficial if the sheet edge is flexible as it travels through the rolls and therefore any bend close to the sheet edge is formed in the last few forming stations on the machine. For complex section geometry such as that shown in figure 1(c) there are many bending sequences that may be selected. Rarely will there be a unique bending sequence. The one that is finally used is often determined from trial and error testing in the workshop.

Modifications to the bending sequence, regarding the amount of bending applied or the addition of more forming stations, are often carried out to overcome section defects. The amount of bending that is applied at each forming station can be vital. This is particularly true at the first forming station where the bend originated in a previously flat region of the sheet. Defects such as edge buckling may result if the amount of bending is excessive, [Salmani Tehrani et al., 2006]. Quality issues are not the only consideration, however, because tool costs are also paramount. An important objective is to minimise the number of forming stations in every tool design solution. A customer is unlikely to purchase tooling that is more expensive compared to that of a rival company, regardless of the quality of the section it may produce. Therefore as much bending as possible is applied at every forming station on the machine, whilst avoiding the creation of section quality problems.

A work tender is relatively costly to generate because an experienced tool designer is employed to predict the number of forming stations that is required. Roll-forming tool design involves expertise that is inherently subjective. There is not a sufficient number of well-established rules to construct an expert system to produce the best bending sequence. A mathematical model of the forming process, using the finite element method, will involve massive computation times unless the section geometry is simple. When this approach is chosen the simulations have usually involved circular tube or single channel sections, [Brunet et al., 1998]. Artificial neural networks (ANN) have been successfully used in problem domains where it is difficult to formulate a solution using a set of rules. A frequent problem with the finite element method is that computation times are too large to provide feasible solutions to practical applications involving metal forming. This is not the case with ANN systems which have been applied to complex problems such as the optimising of process parameters [Gunaskera and Zhengjie, 1998, Kim and Kim, 2001]. There are several examples of ANN systems being used in metal forming in areas such as predicting tool wear [Raj and Sharma, 2000, Kong and Nahavandi, 2002] and dealing with quality issues [Xiaodan et al., 1999, Lorenzo and Ingarao, 2006]. Therefore ANN technology was applied in this project to contend with the subjective nature of roll-forming tool design. An ANN system was trained to predict the size of the forming machine, regarding the number of forming stations that will be required, to produce a section using input data that is based solely on the section geometry [Downes, 2007].

The industrial objective of the project was to develop a method that reduces the cost of generating work tenders. This will be achieved if a computer program is developed which predicts the number of forming stations and the attention of an experienced tool designer is not required. It will also be beneficial, however, if the accuracy of the tool costs estimation is improved.

2. ARTIFICIAL NEURAL NETWORKS

Artificial intelligence software has become more widely used as a result of the expansion in computer-based information systems but has largely involved rule-based expert systems. A comprehensive set of rules had to be constructed to cover the whole of the subject area. The rapid growth of artificial neural networks has provided additional options. Originally ANN systems were based on studies of the biological neural structures [Anderson and Rosenfeld, 1998] but have evolved into the development of useful software systems, separate from contemporary research into biological neural networks. They obtain their problem solving capabilities by training techniques that involve the selection of descriptive data from the problem domain. An ANN system is normally used when the subject area involves uncertainties, therefore they have the potential to formulate a solution that involves highly subjective reasoning. There are many different ANN systems, having a variety of attributes, which makes them suitable for a wide range of applications. MatLab^{TM} has a number of additional programs called "toolboxes" which provides solutions to several engineering topics including artificial neural networks. All of the widely used ANN systems are provided along with the more specialised networks such as the Elman and Hopfield. The ANN system used in this project was chosen from the range of options in the MatLab ANN toolbox [Matlab, 2000].

A very simple processing unit was initially used to model the biological neuron or "brain cell". Every input signal is multiplied by a positive or negative value called a "weight". A summation of the weighted input signals is carried out and the result is presented to a "Step" transfer function. If the summation is equal or greater than the threshold value the processing unit emits an output signal, otherwise it does not respond to the input signals. The artificial neurons used in contemporary ANN systems are based on a similar principle. It is the various methods of connecting the artificial neurons into a network and the numerous training techniques which result in the wide range of different ANN systems. The most frequently used is the feed-forward back-propagation network, (FFBP) which has artificial neurons with either linear of sigmoid transfer functions connected in layers [Bishop, 1995]. All the network weights are set to small random values, typically between +5 and -5, prior to the commencement of training. The input vectors from the training data are then presented to the network and the output is compared to the desired output that is defined by the "target" vectors. Therefore the training data for this type of network involves input and target vector pairs. The training techniques adjust the network weights until the network output is satisfactory. This involves the minimisation of an error function, usually the "sum-of-squares" equation.

When successfully trained the ANN system performs a mapping operation which transposes data from the input space to the output space with satisfactory accuracy. The design of a FFBP network involves the determination of the network architecture, regarding the number of layers and the number of artificial neurons in each layer. Additionally, the transfer functions of the artificial neurons must be chosen. A satisfactory mapping operation will not be obtained if the network architecture does not fit the problem being processed. Several numerical minimisation algorithms are provided by MatLab. For most practical applications the error surface will be highly convoluted and a rapid convergence to an acceptable error value is unlikely. Generally, the design of the network architecture and the training of this type of ANN system is a time-consuming trial and error procedure. If the addition of new knowledge is not involved, therefore the occasional re-training of the network is not required, and suitable input and target vectors can be obtained the FFBP network may be the best option. A storage and retrieval system [Downes et al., 2004] is a good example of a suitable application. Viswanathan et al., [2003] also used this type of ANN system to predict the springback during the forming of a channel section using a stamping operation.

If the application requires the continuous addition of new knowledge it will be necessary to repeatedly re-train the ANN system. In this case the radial basis functions network (RBF) would be a feasible option because it has rapid training characteristics. The RBF network was developed from a method for performing exact interpolation from the input to output space [Powell, 1987]. It consists of two layers of artificial neurons. The first layer transforms the input data to a higher dimensional space called the "hidden" space, using artificial neurons with Gaussian distribution transfer functions. Complex classification tasks become easier in a high dimensional space. The second layer of artificial neurons has linear transfer functions and performs the mapping from the hidden space to the output space. There are no network design tasks or minimisation of error functions which results in extremely rapid training. The RBF network does require input and target vectors, therefore suitable training data must be available. A good example of an application where the RBF network is suitable is a pattern matching task where the strategy is to continually add new patterns to the system [Downes and Hartley, 2006].

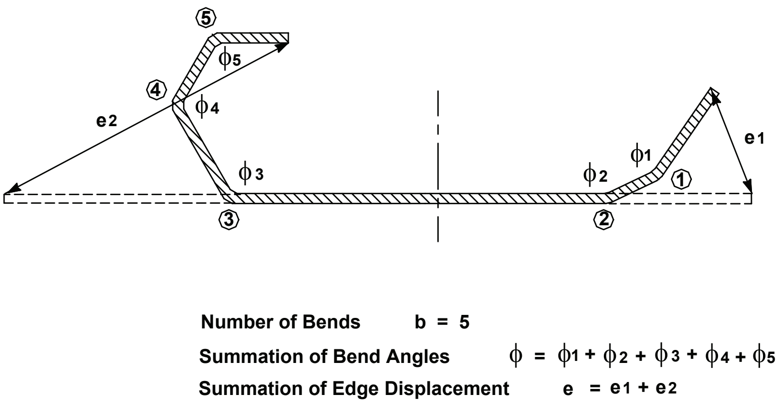

Figure 2. Section Features: Standard Case

Some practical applications involve problem domains that involve uncertainties and the classification of data is not straightforward. In this case it may not be possible to generate a sufficient number of input and target vectors to train an ANN system. The self-organising network [Kohonen, 2001] processes the input vectors during the training phase and will classify them if a statistical trend is detected. It is useful for establishing if a meaningful classification of the training data can be obtained. Target vectors are not required in the training data and it has straightforward network architecture. There is only one layer and the total number of artificial neurons must be equal or greater than the number of classes in the categorisation process. The training technique results in an artificial neuron, or a group of artificial neurons, responding each time a specific type of data is processed by the network. It is not associated with quick training times, however, and will typically take longer to train than FFBP networks.

3. EVALUATING SECTION FEATURES

The quality of performance from all types of ANN systems will depend largely on the training data that was used. Selecting the training data is one of the most important phases of ANN system design. The section features that are chosen will have a prime influence, for the majority of sections, three section features were considered sufficient, although wide section multiple channel profiles would require special consideration. The number of bends and the size of the bend angles will be a good indication of the amount of forming that must be applied and therefore will influence the number of forming stations on the machine. One of the most important unwanted strains that are generated during roll-forming is, longitudinal strain. This is the result of arbitrary points on the section moving different distances between the forming stations. The peak value for the longitudinal strain is at the edge of the sheet. Consequently, the displacement of the sheet edge as it moves through the forming machine is an important tool design parameter. The three section features that were chosen are shown in figure 2.

The roll-forming tools for wide profiles such as roof panels and cladding are based on a different design principle. They are a special case because the objective is to form channels in the flat sheet using profiled rolls. A very large number of bend regions are often involved but neither the number of bends or the size of the bend angles is an indication of the number of forming stations that must be used. The number of channels and their overall depth is a better guide to the number of forming stations. When designing the roll-forming tools for wide profiles the reduction in overall width of the section at each forming station on the machine must be monitored. The difference in section width between adjacent forming stations is called the "pull-in" and is an important tool design parameter. One of the section features that is recommended was the overall "pull-in" value. The three section features used for wide profiles are shown in figure 3.

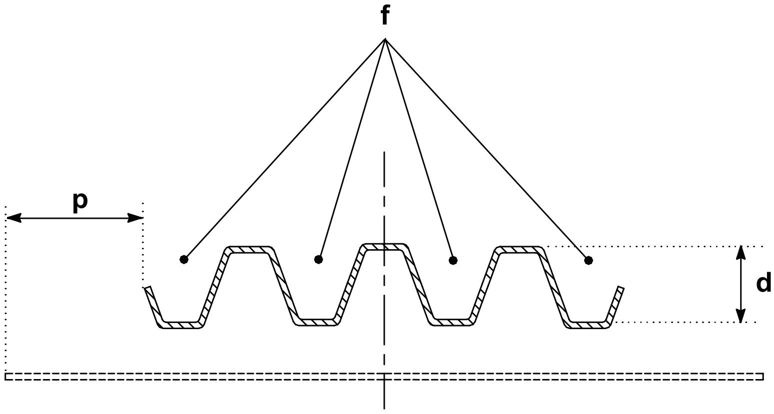

Figure 3. Section Features: Special Case

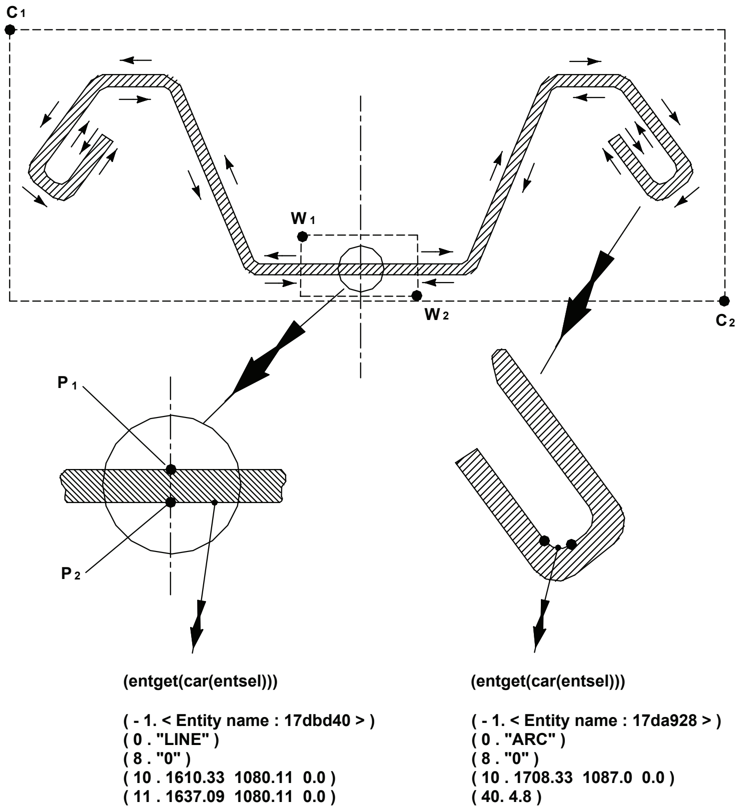

Figure 4. Using LISP to Evaluate the Properties of AutoCAD Drawing Entities

A LISP program was developed in this project to process the AutoCAD drawing of the section geometry and evaluate the three section features. This was developed using AutoLISP^{TM}, which can be invoked within the AutoCAD environment and has a similar syntax to Common LISP [AutoCAD, 1996]. The corner points, C_1 and C_2, are selected by the user to "window" the profile to be processed, as shown in figure 4. All the AutoCAD drawing entities inside the window are listed in an AutoLISP function called a "selection set". The points P_1 and P_2 in figure 4 are defined by the LISP program after the user picks the window corner points, W_1 and W_2. All roll-formed profiles have a geometry that consists of straight lines and circular arcs. Using the AutoLISP command "entsel" and "entget" the definition data of AutoCAD drawing entities is obtained from the AutoCAD drawing database. Examples of the information that can be retrieved are shown in figure 4. The line entity end-point co-ordinates follow the prefix "10" and "11" and the arc radius of an arc entity follow the prefix "40". Every drawing entity in the section profile is processed sequentially by the LISP program, beginning at P_1 and moving along the upper surface to the sheet edge and then along the lower surface to P_2. First the left side of the centre line is processed followed by the right side, as indicated by the arrows in figure 4. To move around the profile the LISP program searches all the drawing entities in the selection set to find a line or circular arc drawing entity which has end-point co-ordinates that coincide with the end-point of the drawing entity that is currently being processed. Having processed all the drawing entities in the profile, the section features are rapidly and accurately evaluated. A LISP program was developed for the general case which involves the evaluation of the section features shown in figure 2, and an additional LISP program was developed for the special case involving the section features shown in figure 3.

4. LEARNING VECTOR QUANTISATION NETWORKS

A Learning Vector Quantisation, (LVQ) network was used in this project and its two-layer architecture is shown in figure 5 [MatLab, 2000]. The input layer is similar to a self-organising network, although the training data involves both input and target vectors. Prior to the commencement of training, the elements in the input layer weights matrix are set to small random values similar to the FFBP network. Every artificial neuron has a unique weight vector, which is one row of elements in the input layer weights matrix. When training begins and an input vector is presented to the network, the artificial neurons compare their weight vector with the input vector. The vector distance between the two vectors is calculated and MatLab uses the function “ndist”. It evaluates the negative of the Euclidean vector distance. The competitive activation function, C in figure 5, processes the results and finds the largest vector distance value and the corresponding artificial neuron is termed the “winning” neuron.

It then generates the output vector, X in figure 5, which has elements equal to zero apart from one unity element that identifies the position of the “winning” neuron in the input layer. The X vector becomes the input vector to the output layer of the network.

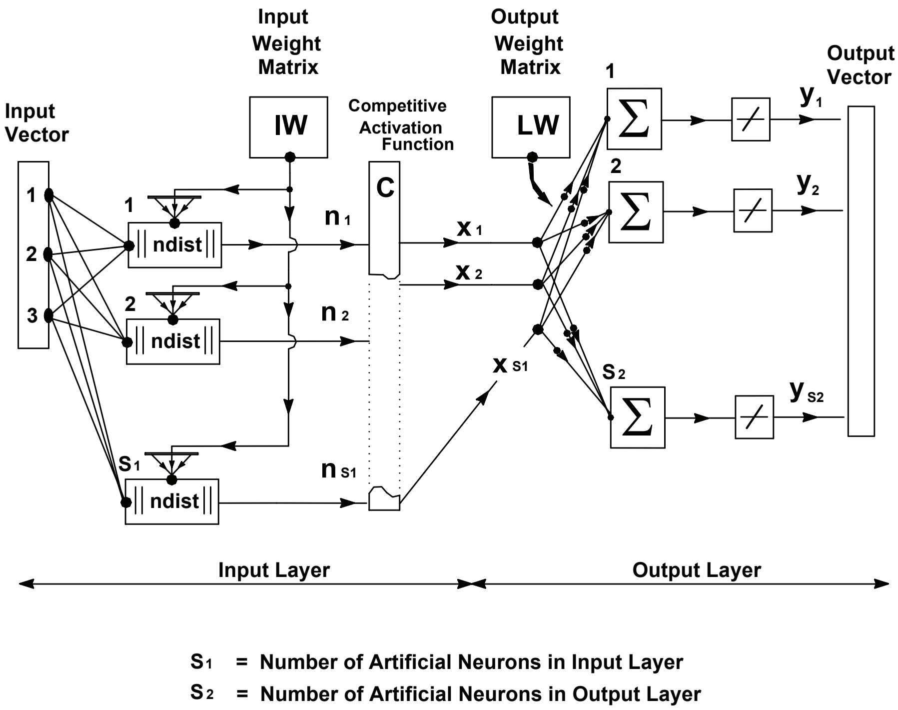

Figure 5. Architecture of LVQ Network

The output layer has artificial neurons which carry out a summation of the weighted input signals and the result is passed to linear transfer functions. Contrary to the input layer weights matrix, every element in the output layer weights matrix is either zero or unity and they are not randomly distributed. They are arranged so that on completion of the training phase the output vectors from the network, Y in figure 5, will equal the corresponding target vectors if the input layer has processed the input vector in the desired manner.

The LVQ network is trained by presenting an input vector from the training data and comparing the network output vector, Y to the corresponding target vector. If they are not the same, the weight vector of the “winning” neuron is adjusted so that its vector distance calculated by “ndist” is decreased. This modification will tend to reduce the likelihood of the artificial neuron becoming the “winner” the next time the input vector is presented. If the Y vector is the same as the target vector the “winning” neuron weight vector is adjusted so its vector distance is increased. Training continues with small increments in the weight vectors of the winning neurons until the network performance is satisfactory. The network weights in the output weights matrix are not changed during the training phase.

The number of artificial neurons in the output layer, S_2 in figure 5, must equal the number of categories in the classification process. A roll-forming machine will rarely have more than 30 forming stations and even a very simple geometry is unlikely to require less than 5 forming stations. Therefore the LVQ network used in the project had 25 artificial neurons in the output layer to correspond to the range of forming stations from 5 to 30. If the first element in the Y vector is approximately unity the network is predicting that 5 forming stations will be required. The number of artificial neurons in the input layer, S_1 in figure 5, must be equal or greater than S_2 and the best value is often determined by experimentation. Consequently, the LVQ network is not associated with vital decisions regarding the network architecture design that is often the bugbear of FFBP networks. Training times will normally be faster than FFBP networks but much slower than RBF networks.

5. TRAINING AND TESTING THE LVQ NETWORK

The industrial collaborator in this research provided a selection of tool design information for narrow profiles. From this data 40 sections were selected to train the LVQ network. These cover the full range of forming machine sizes, varying from 4 sections that required 6 forming stations to one section that required 28 forming stations. Unfortunately the information supplied did not fully satisfy the requirement for training an ANN system. There were no sections that required 19, 21, 23, 24 and 26 forming stations. Additionally, there was only one example of a section that required 15, 16, 27 and 28 forming stations. The training data was therefore incomplete and sparse in some regions of the problem domain, but was adequate for the preliminary investigations reported here.

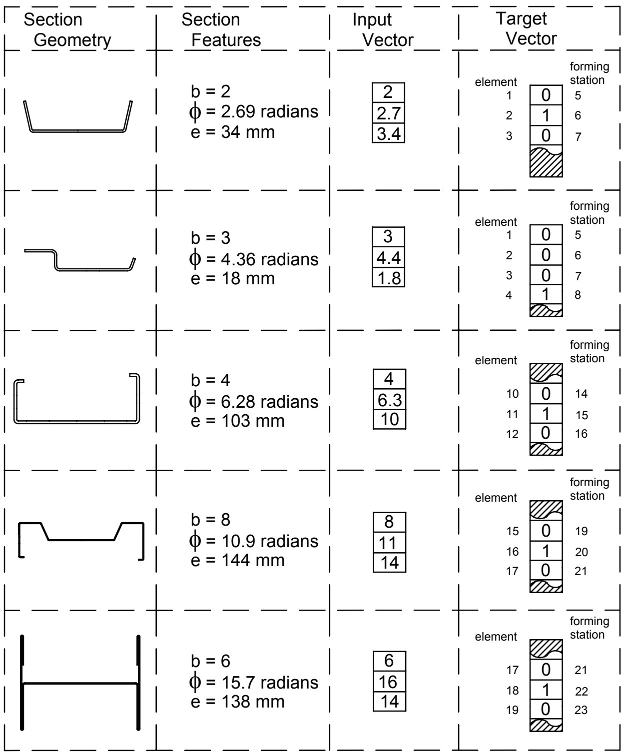

Figure 6. Examples of the Training Data

Each of the 40 sections was processed by the LISP program and the number of bends, a summation of the bend angles in radians and the sheet edge displacements (in millimetres) were evaluated for each section. A simple re-scaling of the edge displacement value was necessary to ensure that the numerical values of the section features had a similar order of magnitude. Therefore the value of the edge displacement *'e'* was divided by 10 so that the section features in the training data are all in the range of 1 to 25. The target vectors had elements equal to zero except one unity element to identify the categorisation that had been made. Five of the sections used to train the LVQ network and their corresponding input and target vectors are shown in figure 6.

After the completion of the training phase, if an input vector used in the training data is processed by the trained LVQ network it will produce an output vector which will approximately equal the corresponding target vector. Therefore an accurate prediction for the number of forming stations will be made. The performance of the trained network is assessed using test data that was not used during the training phase. When the testing was carried out encouraging results were obtained. For example, three sections were all classified as requiring 6 forming stations when the tool designs actually used 6, 7 and 8 forming stations. An error of two forming stations was deemed acceptable by the industrial collaborator. Unfortunately there were examples of unacceptable error. Three sections that were categorised as requiring 8 forming stations actually required 8, 9, and 12 forming stations. The limitation on the range of data that was available from the industrial collaborator restricted the quantity of data used for both training and testing. Although the testing phase demonstrated that the approach can produce satisfactory results, the overall performance of the LVQ network was not sufficiently accurate to be used by the industrial collaborator without further development work to re-train the network. The testing phase had established, however, that the three section features that were used will probably be sufficient. It is unlikely that the development work will involve the addition of new section features.

6. DISCUSSION

After the AutoCAD drawing of the section geometry is completed, a prediction for the number of forming stations is extremely rapid. The LISP program evaluates the section features and this information is exported to the MatLab environment in ASCII format. A trained ANN system will process information at a rapid rate by taking advantage of pseudo parallel processing.

Raw data from a practical application will often require pre-processing before being presented to the ANN system. This may involve only a simple linear re-scaling to ensure the numerical values of the elements in the input vectors are in the same order of magnitude. Normalisation of the training data may also be required and MatLab provides functions that process data samples and sets the mean to zero and the standard deviation to unity. Pre-processing of the data was carried out in this project. Training times were smaller after the training data was normalised, otherwise no notable difference was found.

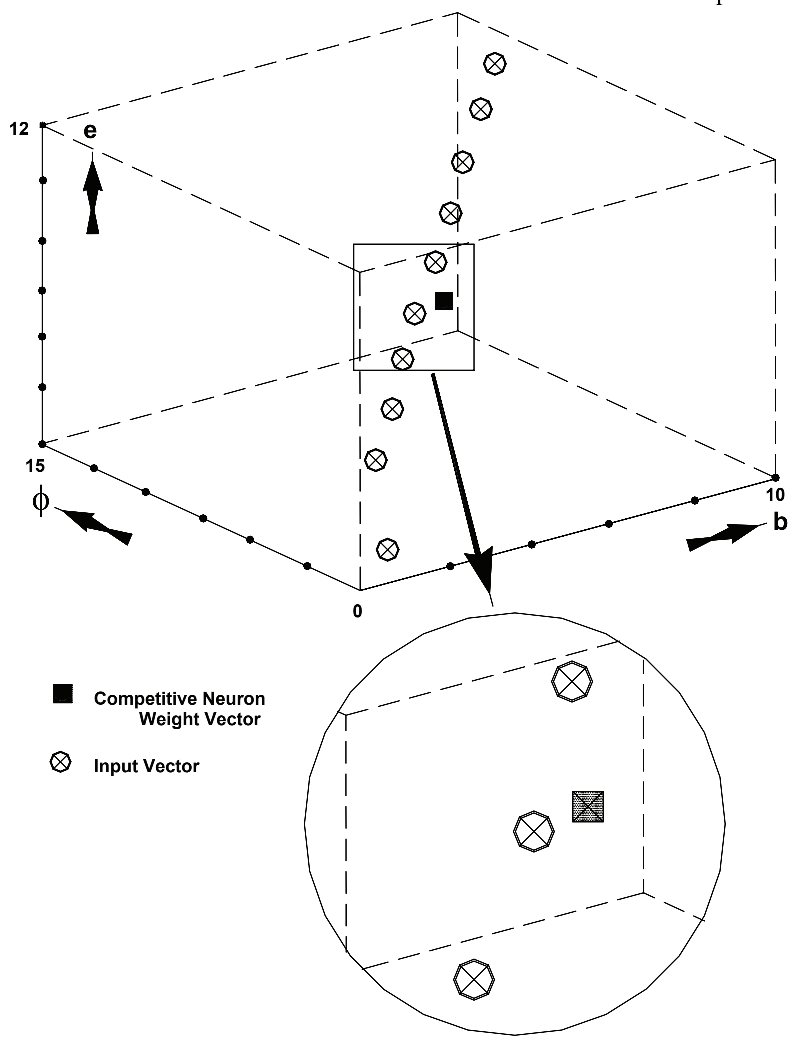

Figure 7. Weight Vectors in the Input Space Before Training

The LVQ network uses a combination of training techniques. A self-organising principle is used in the input layer. The ANN system designer has no influence on which artificial neuron in the input layer becomes the "winner" for a specific input vector that is presented. Therefore the classification that is carried out by the input layer is self-organising. The training technique results in the weight vectors of the artificial neurons in the input layer moving in the input space towards the training input vectors. This movement of the network weights is analogous to the storing of knowledge to allow the network to formulate a solution to the problem being processed. In figure 7 the position of the artificial neuron weight vectors in the input layer is shown prior to the commencement of training. There are several options offered by the MatLab program for initialising the network weights but the "midpoint" function is used in this example. Prior to the commencement of training all the weight vectors are identical and are positioned at approximately the mid-point in the space occupied by the input vectors in the training data. Therefore all the weight vectors are represented by a single point in figure 7. The position of the weight vectors after a period of training is shown in figure 8. Several of the weight vectors have shifted towards the 10 training input vectors. This indicates that the training is progressing in a satisfactory manner. The training procedure compares the network output to the corresponding target vector and then makes adjustments to the appropriate network weights, which is similar to the principle used when training FFBP networks. If the performance of an ANN system is unsatisfactory, a frequently applied strategy is to add new information to the training data. It is difficult to improve a self-organising network using this approach, however, because the selection of suitable new data that will improve performance is not straightforward. If target vectors are used this problem is overcome. This is the foremost reason why an LVQ network was used in this project. The input layer establishes if the section geometry features that have been chosen will result in the desired classification. Using target vectors will allow the training data to be effectively expanded if the performance requires improvement. It is then easy to identify the region of the input space where mapping is consistently poor.

The procurement of training data for the LVQ network requires careful consideration. Examples of previous tool designs from industrial examples are not normally suitable. The number of forming stations used to roll-form a particular section will often be influenced by other features. There are several conditions, such as the bend angle tolerances, which will require additional forming stations but this will not be evident from studying the section geometry. The most influential issue, however, is that a previous tool design will, in many cases, not be exemplary in minimising the number of forming stations for a particular section. It is vital that the number of forming stations predicted by the LVQ network is based on a minimum estimation. It is evident from the testing phase that the development work will require new training data. Obtaining the training data is a formidable problem and a strategy similar to an expert system approach was chosen. If the training data is obtained by consulting an experienced tool designer, who studies the section geometry and from this information predicts the minimum number of forming stations, the training data will not be distorted. Although the task will be time-consuming, an adequate range of training data can be constructed using this approach. By continually creating fictitious section geometries for the tool designer to study, a sufficient number of examples can be created for each of the 25 forming machine sizes.

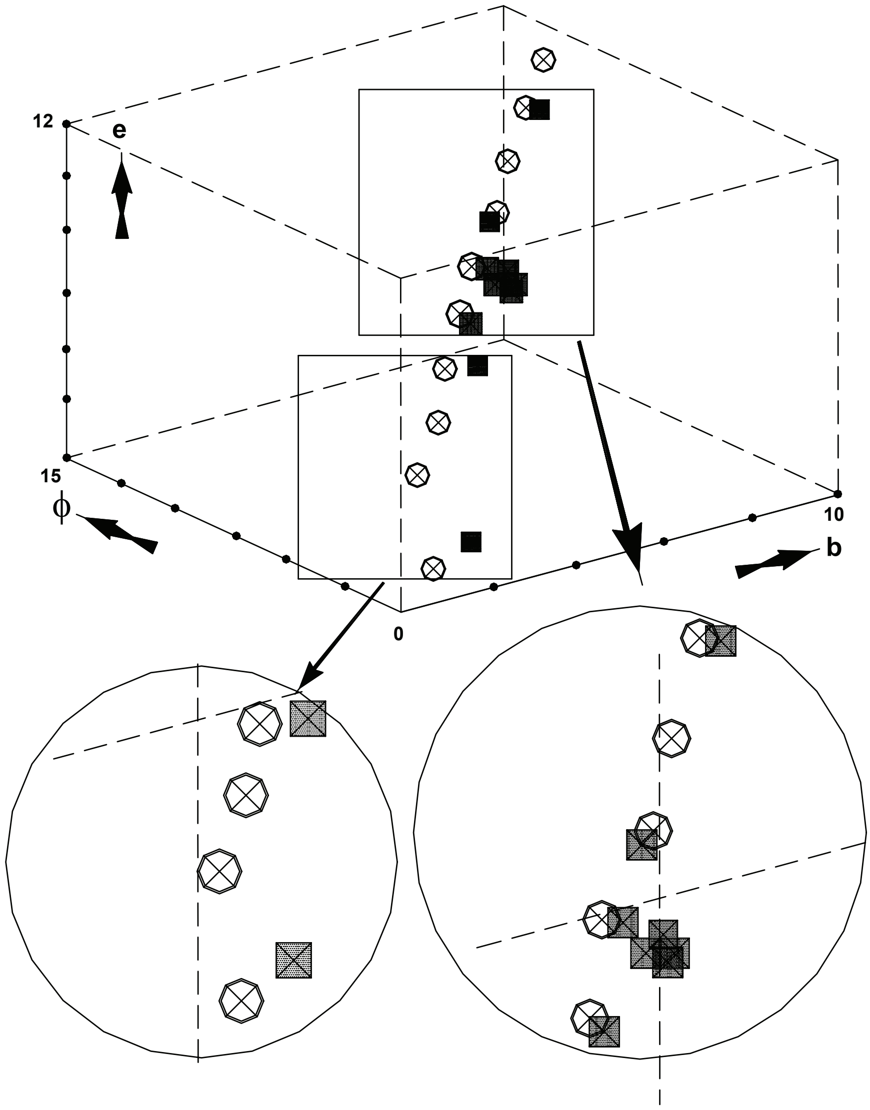

Figure 8. Weight Vectors in the Input Space After Training

After the LVQ network makes a prediction for the number of forming stations, the final number used for costing purposes will usually be larger. Simple rules can be applied to allow for features such as the sheet metal properties or sheet feed method. An advantage of using MatLab, in addition to providing the LVQ network program, is that a computer program using the MatLab language can be developed to carry out these adjustments and produce the final tool cost estimation

7. CONCLUSION

The work undertaken in this project has developed a method for generating work tenders which does not require the attention of an experienced tool designer. This will result in a significant reduction in the cost of producing a work tender. There is also a possibility that the tool cost estimation that is produced by the computer-aided method will result in an improvement in the accuracy compared to the previous practice. If the most experienced tool designer is commissioned to generate the training data for the LVQ network, the performance of the network should reflect the superior judgements that have been made.

There are many different types of ANN systems offering a wide range of capabilities. It is essential that the most suitable network is selected for the problem being processed.

When choosing the ANN system it is beneficial to consider the modifications to be carried out if the network performance is unsatisfactory. If the strategy is to enlarge the training data it may be helpful to choose an ANN system that involves target vectors to ensure the new data will improve the network performance.

The training data for the ANN system will largely influence the overall network performance. It is essential that the input data will allow the ANN system to produce the desired classification. One of the useful attributes of ANN systems is the ability to accurately process input data which is distorted, but important information cannot be omitted. A self-organising network is useful for testing if the features selected in the input data will result in the desired classification, and this is also an attribute of LVQ networks.

The project described in this paper demonstrates how an ANN system can provide a successful solution in a problem domain that involves largely subjective knowledge. It is important to acknowledge, however, that a human expert in the subject area was consulted to assist in the construction of the training data, regarding the selection of section features for the input vectors and the pairing of input and target vectors. Obtaining the training data was a time-consuming task. The application of ANN technology is capable of producing impressive solutions, but the network design and training data procurement can be challenging in some cases. If a solution can be formulated using a rule-based algorithm an ANN system is probably not the best option

Acknowledgements. The authors are grateful for the support given by Michael Castellucci and the contributions of the roll-forming tool design team at the Hadley Group.

REFERENCES

Anderson, J., Rosenfeld, E., 1998, *Neurocomputing: foundations of research*, MIT Press, Cambridge, MA, 18-27.

AutoCAD Reference Manuals, 1996, *AutoLISP programmers manual*, AutoDesk Ltd.

Bishop, C. M., 1995, *Neural networks for pattern recognition*, Oxford University Press, Oxford.

Brunet, M., Mguil, S., Pol, P., 1998, Modelling of a roll-forming process with a combined 2D and 3D FEM code, *J. Mat. Proc. Techn.*, 80-81, 213-219.

Castellucci, M., Pillinger, I., Hartley, P., 1997, The optimisation of cold rolled formed products, *J. Thin-Walled Structures*, 29, 159-174.

Downes, A., 2007, *The Use of Artificial Neural Network Techniques in Cold Roll-Forming Processes*, PhD Thesis, University of Birmingham, Birmingham.

Downes, A., Hartley, P., 2006, Using an artificial neural network to assist roll design in cold roll-forming processes, *J. Mat. Proc. Techn.*, 177, 319-322.

Downes, A., Hartley, P., Pillinger, I., 2004, A storage and retrieval system for roll-forming design data using a neural network, *Steel-Grips: Journal of Steel and Related Materials*, 2, 235-239.

Fewtrell, J., 1990, *An experimental analysis of operating conditions in cold roll forming*, PhD Thesis, University of Aston, Birmingham.

Gunasekera, J., Zhengjie, J., 1998, Development of a neural network model for a cold rolling process, *Engineering Applications of Artificial Intelligence*, 11, 597-603.

Hadley Group (Smethwick), UK, 2006, www.hadleygroup.co.uk.

Kim, D., Kim, Y., 2001, Optimization of the irregular shape rolling process with an artificial neural network, *J. Mat. Proc. Techn.*, 113, 131-135.

Kohonen, T., 2001, *Self-organising maps*, Springer-Verlag, London.

Kong, L., Nahavandi, S., 2002, On-line condition monitoring and control system in forging processes, *J. Mat. Proc. Techn.*, 125, 464-470.

Lorenzo, K., Ingarao, G., 2006, On the use artificial intelligence tools for fracture forecast in cold forming processes, *J. Mat. Proc. Techn.*, 177, 315-318.

MatLab Reference Manuals, 2000, *Neural network toolbox user guide*, The Mathworks Inc.

Salmani Tehrani, M., Hartley, P., Moslemi Naeini, H., Khademizeh, H. 2006, Localised edge buckling in cold roll-forming of symmetric channel section, *J. Thin-Walled Structures*, 44, 184-196.

Powell, M.J.D., 1987, *Radial basis functions for multivariable interpolation", Algorithms for Approximations*, Clarendon Press, Oxford, 143-167.

Raj, K., Sharma, R., 2000, Modelling of manufacturing processes with ANNs for intelligent manufacturing, *Int. J. Machine Tools and Manufacture*, 40, 851-868.

Viswanathan, V., Kinsey, B., Cao, J., 2003, Experimental implementation of neural network springback control of sheet metal forming, *J. Engineering Materials and Technology*, 125, 141-147.

Xiaodan, W., Jianwen, W., Flitman, A., 1999, Neural and machine learning to the surface defect investigation in sheet metal forming, *Proc. the 6th Int. Conf. on Neural Information Processing*, 3, 1088-1093.

ZASTOSOWANIE SZTUCZNEJ SIECI NEURONOWEJ DO OCENY KOSZTÓW NARZĘDZI DO GIĘCIA TAŚM

Streszczenie

Przemysłowe procesy kształtowania blach są poddane dużej konkurencyjności i większość projektów sekwencji gięcia jest nieskuteczna. Jest szereg parametrów, które mają wpływ na koszt narzędzi, ale głównym problemem jest przewidzenie liczby rolek potrzebnych do uformowania danego kształtu i, w konsekwencji, określenie rozmiaru maszyny. W artykule omówiono metodę oceny kosztów narzędzi w procesie kształtowania blach na zimno. Celem pracy było obniżenie kosztów projektowania układów rolek przy zapewnieniu utrzymania dokładności oceny kosztów narzędzi, lub nawet jej poprawie. Program LISP został opracowany dla realizacji tego celu. Program przetwarza rysunki AutoCad kształtów przekroju i ocenia wybrane cechy kształtowników takie jak całkowita liczba stref zginania. Te cechy kształtowników są następnie przetwarzane przez sztuczną sieć neuronową, nauczoną przewidywać rozmiar maszyny formującej wymagany do wytworzenia danego kształtownika.

Submitted: December 11, 2006
Submitted in a revised form: January 10, 2007
Accepted: January 12, 2007