Pre-publication draft of a paper which appeared in the Proceedings of DETC2002, The 2002 ASME Computers and Information in Engineering Conference, pp 1-8, ASME Press.

Improving an Inverse Model of Sheet Metal Forming by Neural Network Based Regression

Y. Frayman, B.F. Rolfe and G.I. Webb

School of Computing and Mathematics, Deakin University, Geelong, Vic. 3217, Australia

Abstract

An inverse model for a sheet metal forming process aims to determine the initial parameter levels required to form the final formed shape. This is a difficult problem that is usually approached by traditional methods such as finite element analysis. Formulating the problem as a classification problem makes it possible to use well established classification algorithms, such as decision trees. Classification is, however, generally based on a winner-takes-all approach when associating the output value with the corresponding class. On the other hand, when formulating the problem as a regression task, all the output values are combined to produce the corresponding class value. For a multi-class problem, this may result in very different associations compared with classification between the output of the model and the corresponding class. Such formulation makes it possible to use well known regression algorithms, such as neural networks. In this paper, we develop a neural network based inverse model of a sheet forming process, and compare its performance with that of a linear model. Both models are used in two modes, classification mode and a function estimation mode, to investigate the advantage of re-formulating the problem as a function estimation. This results in large improvements in the recognition rate of set-up parameters of a sheet metal forming process for both models, with a neural network model achieving much more accurate parameter recognition than a linear model.

Keywords: *sheet metal forming, inverse models, regression, classification, neural networks*

1 INTRODUCTION

The tuning of a process is a difficult problem. Nevertheless, tuning of a process is important to the manufacturing industry in at least two ways. First, the tuning of a die's profile to produce a correct part has been largely a trial and error procedure, which is both expensive in time and cost [1]. Second, tuning is necessary to control the output of the process and this is often performed manually by experienced operators.

Often an inverse model can assist tuning problems by providing a relationship between the output results and the input process parameters. Developing an inverse model is, however, a challenging proposition if the process is non-linear nature. In the case of sheet metal forming, one must model a large deformation process with non-linear material properties and many process parameters (blank holder force, lubrication, geometric parameters of the die).

A majority of inverse models in sheet metal forming have concentrated on obtaining a desired final part geometry [2,3,4]. The inverse model approach has been used to improve die design by using an iterative approach of optimising the shape of a finite element model [2,3]. Inverse modelling is also used to tune the process parameter levels in sheet metal forming to give a desired output. Gelin and Ghouati [4] optimise the geometry of a finite element model to determine the appropriate initial material properties.

This paper is also concerned with determining the initial parameter levels of a process, but instead of using iterative approach to optimize the finite element model we are using advanced artificial intelligence techniques to gain a greater accuracy of the inverse model. The eventual aim of this research is to develop an automatic process tuning algorithm. This work has improved upon the initial inverse model results obtained by Rolfe *et al.* [5] with regard to channel forming.

There are several possible ways to approach inverse modeling using artificial intelligence techniques. One possible way is to formulate the problem as a classification task and use some well established classification algorithms, such as decision trees. In this formulation a winner-takes-all approach is generally used to associate the output value with the corresponding class. That is, only the largest likelihood value of the outputs is taken into account and the other values are ignored.

On the other hand, when formulating the problem as a regression task, all the estimated output values are combined to produce the corresponding class value. In case of a multi-class problem, this may result in very different associations between the output of the model and the corresponding class. Such formulation makes it possible to use some well known regression algorithms such as neural networks.

The main contribution of this paper is, therefore, to investigate the advantage of re-formulating the problem as a function estimation with a neural network based inverse model. This identifies process set-up parameter levels by analyzing the variations in geometric shape of a part from a sheet metal forming process. A comparison with a standard linear regression model is also made. The shape variation of a stamped component is described by a shape variation that elicits the mayor modes of shape variation from the sheet metal components based on the Point Distribution Model (PDM) [6].

2 POINT DISTRIBUTION MODEL (PDM)

The Point Distribution Model (PDM) [6] is a statistical deformable model which underpins the feature analysis of this paper. The PDM compares the variation of points on the boundary of shapes within a set of training shapes. This is performed by comparing each coordinate of each point versus the coordinates of every other point on each shape and across the training set in the form of a co-variance matrix. The training set should include all of the variations that need to be recognised. The shapes in the training set are aligned to remove any registration errors, and each shape in the set is labelled with a series of points that are recognisable and consistent. A shape can then be represented by a vector of the coordinates of the labelled points, for example, X= $[x_l, y_l, z_l, ..., x_k, y_k, z_k]^T$ .The PDM then uses principal component analysis to reduce the dimensionality of the co-variance matrix. The resulting vectors are the major modes of variation of the coordinates of the shapes in the training set. Therefore, any shape in the training set can be represented by,

(1) $X_i = \bar{X} + \mathrm{P}b$

where b is the weighting vector that shows how much of each mode is needed to vary the mean shape $\bar{X}$ to the shape X_i, **P** is the matrix which contains the principal or major modes of variation.

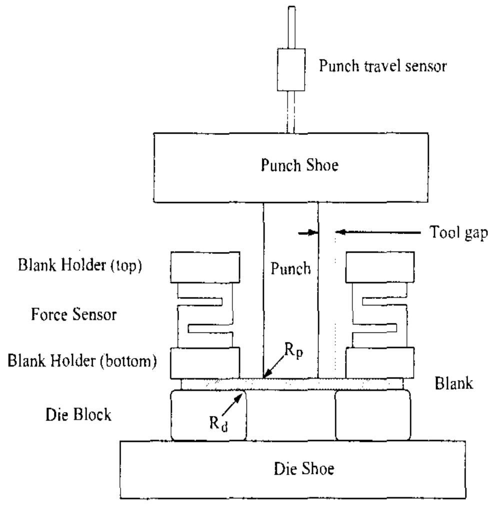

Figure 1. The "U" channel forming test.

2.1 EXPERIMENTAL SHEET METAL FORMING SET-UP

The shape variation/process parameter model is developed via a set of standard 2D drawn "U" bend channels. A three factor (blank holder force, die radius and tool gap) three level factorial experiment with four repetitions was performed to create this set of channels. The set of channels was then used as a training set for the PDM method to determine the shape variation modes of the process.

The "U" channel forming consisted of forming a blank into a "U" shape while the edges of the blank (flanges) were held down by a blank holder. All forming was conducted on a 30 ton Heine & Sons press. The experimental setup used two load cells to measure the blank holder forces and a linear potentiometer to measure the travel of the punch as can be seen in Figure 1. The sample blanks were made out of Zinc anneal G3N hot-dipped zinc/iron alloy coated drawing steel with a sheet thickness of 0.76 mm (Yield strength = 130 - 1 70MPa, Tensile = 280 - 320MPa, r_{45} = 1.2 - 1.6). The G3N material is used for several different car panels within a large saloon vehicle stamping plant. Each blank was cut into (150mm x 20mm) strips with a guillotine and then the edges were de-burred. The blanks were formed to a depth of approximately 32mm.

This paper investigates varying three parameters, Blank Holder Force (B), Die Radii (D) and Tool Gap (T), to obtain geometric variations in the deformed channel. These parameters were varied according to a three level three factor factorial experiment. The ranges of the parameters are given in Table 1. After forming, each sample was scanned to obtain its 2D cross section. The scanned cross section was then imported into an analysis program where a chain coding (edge detection) algorithm was used to segment out the lower boundary of the scanned channel.

Set-up Parameter	Levels		
Blank holder force	B1 = 3KN	B2 = 7.5KN	B3 = 12KN
Die radii	Dl = 3.175mm	D2 = 4.7625mm	D3 = 6.35mm
Tool gap	TI = 1.05mm	T2 = 1.30mm	T3 = 1.62mm

Table 1. Set-Up Process Parameters Levels.

2.2 SHAPE ALIGNMENT AND POINT PLACEMENT

All the channels were aligned to remove registration errors. Initially the mid-point of the floor of each channel was found and this was used as the first alignment point (see Figure 2(a)). The channel floor's mid-point was found by bisecting the imaginary line between the two corners of the channel (see Figure 2(b)). The corners of the channel were determined by using orthogonal regression to fit five lines to the two flanges, two side walls and the floor of the channel where the corners are defined as the intersection between the side walls and the floor. The registration error was gradually removed by an iterative process that rotated and translated the channels to minimise the error between the average shape of the channels and each particular channel.

The points were then evenly distributed around the shape from a consistent datum point. The datum point selected was the lower floor mid-point of the channel which was fixed due to the width of the punch being the same for all the samples. The points were distributed equally (with an equal distance between each point) on each half (left and right) of the channel.

2.3 SHEET METAL PDM

After the set of channels is created and aligned, the PDM is applied to the set. The PDM equation (1) for the sheet metal process is updated as follows:

(2) $$X_{channel} = X_{mean\_channel} + \mathbf{P}b,$$

where $X_{channel}$ is the list of boundary point coordinates for any channel that can be described by a combination of the major shape variation modes and $X_{mean\_channel}$ is the list of boundary point coordinates for the mean channel of the data set. Thus, an independently formed channel not of the set of original channels can be described by a b vector as seen by rearranging equation (2):

(3) $$b = \mathbf{P}^{+}(X_{measured\_channel} - X_{mean\_channel})$$

where $X_{mean\_channel}$ is the mean channel defined above and the variable $X_{measured\_channel}$ shape contains the points on the surface of the newly produced channel. The **P** matrix holds the eigenvectors of shape variation and **P**+ is the pseudo inverse of **P**, that is, $\mathbf{P}^{+} = (\mathbf{P}^{T}\mathbf{P})^{-1}\mathbf{P}^{T}$.

3 RESULTS

Inverse model set-up

As stated previously, the major aim of this paper is to develop an inverse model to identify process set-up parameter levels by analyzing the variations in geometric shape of a part from a sheet metal forming process. That is, shape variations from the nominal sheet metal part are to be linked to various levels of the process set-up parameters.

There are several possible approaches to the development of such an inverse model. The most common approach is to formulate the problem as a classification (pattern recognition) problem [7]. Traditionally, classification methods were used in cases where both the inputs (attributes) and the outputs (classes) are binary or discrete levels. Although it is possible to use traditional classification methods, such as decision trees, with the continuous data, this generally requires discretization of continuous variables in order to make the learning possible [7]. This may, however, result in some loss of the information.

At the same time, little attention is paid to the possibility of re-formulating the classification problem as a function estimation (regression) problem. This is in contrast with attempts to re-formulate a regression problem as a classification one, which are quite popular [8,9]. There is essentially only one difference between the classification and regression: in case of a classification, the outputs (classes) have discrete values, while in case of a function estimation (regression) the output is a continuous variable which is converted into discrete values afterwards. As the inputs to the inverse model (shape variations from the nominal sheet metal part) in our case are continuous variables, our hypothesis is that a function estimation (regression) will give a lower misclassification error than a pure classification. Our reasoning is based on the fact that in a classification approach, when associating the output value with the corresponding class, only the largest value of the outputs is taken into account (winner-takes-all approach) [10] and the other values are ignored. Whereas, in a function estimation approach, all the output values are combined (using for example a weighted sum) to produce the corresponding class value. In a multi-class problem, this may result in very different associations between the output of the model and the corresponding class.

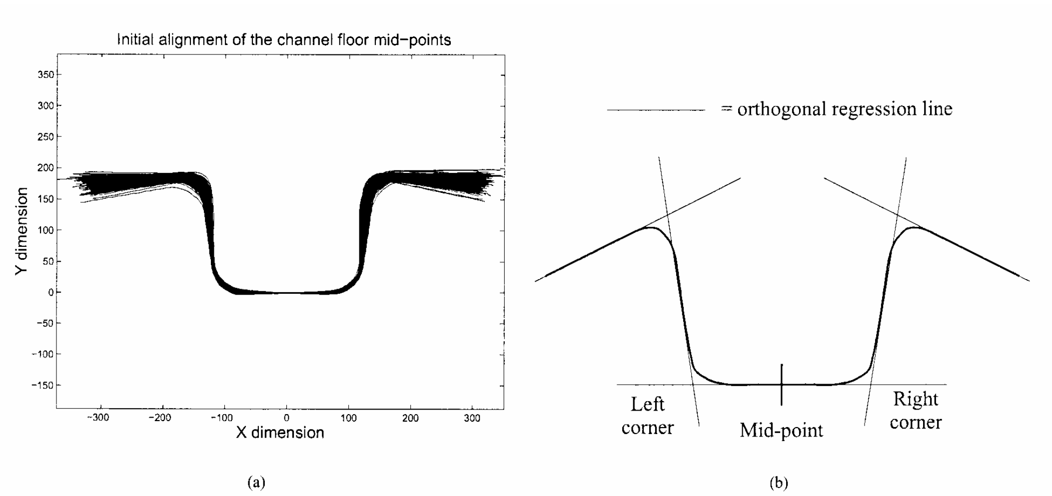

Figure 2. (a) Diagram Showing the Initial Alignment of Boundary Nodes.

(b) Diagram Showing the Determination of The Mid-Point of the Channel.

Function estimation formulation makes it possible to use a well known regression algorithm such as neural networks [11].We have selected two possible approaches to our inverse modeling: 1) a traditional linear regression; and 2) a feed-forward multilayer perceptron (MLP)(neural network).

The reasons for such a selection are to test both the linear (linear regression) and nonlinear (MLP) methods [11]. Both methods were used in two modes:

1. a classification mode; and
2. a function estimation (regression mode).

In a function estimation mode the final output (class values) of the predictor is in a continuous form, which is then converted to an integer (class value) by rounding up the continuous number. For example, the output value of 1.49 will be converted to 1, and the value of 1.51 will be converted to 2.

In a classification mode three output nodes are required for the MLP to account for all of the output classes, and in a function estimation mode one output node is sufficient.

The linear regression used is a standard multivariate linear regression algorithm that generates a linear weighted sum of the inputs plus a constant bias for each output. The coefficients (weights) of a linear regression minimize the least-mean-square error between the desired outputs and a linear regression outputs.

The feed-forward multilayer perceptron (MLP) used is a neural network algorithm which generates input-output mappings based on computations of interconnected nodes [11]. Nodes are arranged in layers. Each node's output is a nonlinear function of the weighted sum of inputs from the nodes in the preceding layer. The learning algorithm used was back-propagation. The activation function was a hyperbolic tangent.

The optimal parameters of the MLP were selected based on a combination of a genetic algorithm search [12] through different sets of network structures and parameters to limit the search space; and then an exhaustive search to fine tune the network structure and the parameters found by the genetic search. The parameters of the genetic algorithm are:

number of generations —8;

number of samples of each generation— 25; and

mutation rate 10.

The parameters which resulted in smallest root mean squared error (RMSE) between the predicted output values and the actual output values (classes) were used.

The following parameters were tested:

the number of hidden layers being either 1 or 2;

the number of nodes in the hidden layer 1 ranging from 0 to 40;

the number of nodes in the hidden layer 2 ranging from 0 to 10;

the gradient descent step size (learning rate) ranging from 0.005 to 1.0;

the gradient descent momentum ranging from 0.0 to 1.0;

the method of weights update being either per epoch (batch) or pattern;

the output node using either a linear function or a sigmoid function;

the output normalization using either a z score or none;

the normalization of inputs being either none, or a z score, or a min-max cutoff, or a sigmoidal.

As a stopping condition a minimal RMSE between the predicted output values from the model and the actual output values were tested ranging from 0.01 to 0.2.

The optimal models were selected using ten-fold cross- validation [7] where the sets of channels were randomized and split into ten equal parts. One part was held out in turn while the model was trained on the remaining nine parts. The held out part was then used as a test set to measure the model's accuracy. The process was then repeated ten times.

The optimal structure and parameters of the MLP's models found are shown in Table 2.

Inverse models created by both the linear regression and the MLP's methods were then used to predict the original levels of the process set-up parameters from the shape metric data.

Description	**Blank holder force**		**Die radii**		**Tool gap**	
	Classification	Regression	Classification	Regression	Classification	Regression
Number of inputs	20	20	20	20	20	20
Hidden layers	1	1	1	1	1	1
Neurons in hidden layer	2	1	2	2	3	2
Neurons in output layer	3	1	3	1	3	1
Activation function	tanh	tanh	tanh	tanh	tanh	tanh
Learning Rate	0.2	0.05	0.1	0.2	0.1	0.15
Momentum	0.6	0.5	0.6	0.8	0.7	0.7
Weight Update	pattern	pattern	pattern	pattern	pattern	pattern
Output Function	linear	linear	linear	linear	linear	linear
Input Normalization	zscore	zscore	zscore	zscore	zscore	zscore
Output Normalization	zscore	zscore	zscore	zscore	zscore	zscore
Minimum RMSE	0.1	0.1	0.1	0.1	0.1	0.1

Table 2. The Optimal Structure and Parameters of the MLP's Models.

Modeling Results

The data was initially analyzed using a linear cluster separation measure to indicate the general success of a linear recognition. This simple measure was calculated by taking the trace of the between class variance matrix multiplied by the inverted within class variance matrix [13]. A higher value of the linear measure implies that the set of data is more separable. The data is split into the three levels for each parameter and the linear measure estimates the potential that each parameter can be recognized correctly for the appropriate parameter level.

The resulting values from the measure are:

Blank holder force	22.0238
Die radii	20.0000
Tool gap	6.2938

The tool gap parameter is therefore the most difficult to recognize correctly.

The linear and the MLP inverse models were then used to determine the original levels of the process set-up parameters from the shape metric data. The results from the models for each parameter can be seen in Table 3.

The recognition rate of the models for each parameter's levels boundaries can be seen in Table 4.

		Linear regression model						**MLP model**					
		Classifier			Function estimator			Classifier			Function estimator		
Blank holder force	Recognition rate	86.7%			98.5%			96.3%			100.0%		
	Confusion matrices	42	10	0	45	1	0	56	2	0	45	0	0
		3	35	5	0	44	1	0	41	1	0	45	0
		0	0	40	0	0	44	0	2	44	0	0	45
Die radii	Recognition rate	88.2%			95.6%			93.3%			98.5%		
	Confusion matrices	44	5	0	42	0	0	43	1	1	44	1	0
		1	31	1	3	42	0	2	40	1	1	44	0
		0	9	44	0	3	45	0	4	43	0	0	45
Tool gap	Recognition rate	69.6%			83.0%			76.7%			92.6%		
	Confusion matrices	42	7	1	37	0	0	43	3	0	44	0	0
		2	15	7	8	39	9	0	29	14	1	40	4
		1	23	37	0	6	36	2	13	31	0	5	41

Table 3. Average Model Prediction Results for Each of the Process Parameters

		Linear regression model		MLP model	
		Classifier	Function Estimator	Classifier	Function Estimator
Blank holder force	low-high	100.0%	100.0%	100.0%	100.0%
	low-med	85.6%	98.9%	97.8%	100.0%
	med-high	94.4%	98.9%	96.7%	100.0%
Die radii	low-high	100.0%	100.0%	98.9%	100.0%
	low-med	93.3%	96.7%	96.7%	97.8%
	med-high	88.9%	96.7%	94.4%	100.0%
Tool gap	low-high	97.8%	100.0%	97.8%	100.0%
	low-med	90.0%	91.1%	96.7%	98.9%
	med-high	66.7%	83.3%	70.0%	90.0%

Table 4. Average Model Recognition Rate for Each of the Process Parameters' Levels Boundaries

4 DISCUSSION

Linear classifier model

The linear regression model used in a classifier mode appears to be able to identify the channels into the high and low parameter levels for all parameters with almost 100% accuracy, see Table 4. The medium level, however, is obviously too close to either the low or high parameter levels. The blank holder force parameter appears to have more data clusters that are close together for low and medium levels than for medium and high levels. This is due to the fact that for blank holder force the higher forces have much greater impact on the end shape particularly as the blank is very near to necking and tearing.

The difficulty with the tool gap data for the linear regression model used in a classifier mode is mostly between the medium and high levels. This is primarily due to the errors in the tool gap set-up. The medium tool gap uses three shims to create the gap whereas high tool gap uses only two shims. There is a greater possibility for errors when aligning the three shims, moreover, this error will always cause an increase of the actual tool gap. This resulted in a consistent problem when discriminating between high and medium levels for the linear regression inverse model used in a classifier mode.

MLP classifier model

The MLP model in a classifier mode is able to distinguish between the medium and high parameter levels much better than the linear regression model for all parameters, see Table 4.

Once again, the tool gap parameter is the hardest to recognise correctly (the recognition rate is 76.7%). Note that the MLP model in a classifier mode does identify the tool gap parameter with greater accuracy than the linear model. Similarly to the linear model, it is still difficult to discriminate accurately between the medium and the high tool gap channels.

Linear and MLP predictor models

It can be seen from the results in Table 3 that there is benefit to re-formulating the problem as a function estimation task. The misclassification error, in case of the linear regression model, has been reduced from:

13.3% to 1.5% for the blank holder force;

11.8% to 4.4% for the die radii; and

30.4% to 17.0% for the tool gap.

There is a clear distinction between the low and high parameters levels for all parameters. There is, however, still some difficulty in distinguishin g accurately between the medium and high tool gap parameter levels.

The corresponding results for the MLP model, when used as a function estimator, are even better:

perfect recognition for the blank holder force;

1.5% error for the die radii; and

7.4% error for the tool gap.

There is practically no difficulty in distinguishing between the low and medium parameters levels for all parameters. The only difficulty remains in distinguishing between the medium and high levels for the tool gap parameters, however there is much improvement when compared to the MLP model in a classifier mode.

To compare the recognition rates of the two modes, twotailed sign test [14] was performed. The difference was considered significant if the significance level of the sign test was smaller than 0.05. Table 5, shows the numbers of wins, ties, and losses between the recognition rates of the corresponding two algorithms for all levels, and the significance level of twotailed sign test [14] on these win/tie/loss records. From Table 5 it can be seen that, for both the MLP and the linear regression, there were statistically significant improvements in the function estimation mode results over that of the classifier mode.
Improvements in results for the MLP method over that of the linear regression method in a function estimation mode are also statistically significant. The improvements in the MLP results over that of the linear regression in a classifier mode are not statistically significant to the level of 5%. The MLP results, however, are still much better than the corresponding linear regression results.

But more importantly, both models used in a function estimation mode consistently improved the recognition rate by 10 - 20% compared to the corresponding classification mode. This demonstrates the advantage of formulating the problem as a function estimator over a classifier formulation. The function estimation mode suits the continuous data better than the classification mode, especially for the case of a multi-class problem. As indicated above, in the classification mode, when associating the output value with the corresponding class, only the most likely of the outputs is taken into account (winner takes all approach) and the other probability values are ignored. Whereas, in the case of function estimation, all the output values are combined to pro duce the corresponding class value. In a multi-class problem this results in very different associations between the output of the model and the corresponding class.

Method	Comparisons	win-tie-losses	p of win-tie-losses
MLP	regression vs. classifier	8-1-0	p = 0.0078
Linear	regression vs. classifier	7-2-0	p = 0.0156
Regression	MLP vs. Linear	6-3-0	P = 0.03l2
Classifier	MLP vs. Linear	6-2-1	P = 0.2187

Table 5. Statistical Significance Test of the Results

5 CONCLUSION

In summary, the inverse models were created using linear regression and MLP methods that related the *b* vectors to the process parameter levels. The MLP model in a function estimator mode created the most accurate predictor for all the parameters with the average recognition rates for:

Blank holder force	100.0%;
Die radii	98.5%;
Tool gap	92.6%

The linear regression model in a function estimator mode still consistently confused the medium level parameter with the high level parameter for all parameters. In addition, the linear model was much less accurate than the corresponding MLP model with the aver age recognition rates for:

Blank holder force	98.5%;
Die radii	95.6%;
Tool gap	83.0%.

The nonlinear nature of the data suited the MLP model better than the corresponding linear model.

Given the similar prediction accuracies from analyzing forging data [15] it may be generalised that the PDM shape error metric will provide reasonable classification data in most cases for other shape manufacturing processes. Moreover, this shape-process parameter model allows the further development of auto mated intelligent process control within sheet metal forming by providing a mechanism that can recognize how distant the cur rent set-up parame ters are from the optimal parameter set-up in order to produce the desired geometric shape.

Acknowledgeme nts

The authors are grateful for support from the Centre for Stamping Technology for Automotive Manufacturing Processes (STAMP) http://www.et.deakin.edu.au/stamp/.

References

[1] S. Yang and K. Nezu. **Application of an inverse FE approach in the concurrent design of sheet stamping**. *Journal of Materials Processing Technology*, 79:86 - 93, 1998.

[2] Y.Q. Guo, J.L. Batoz, H. Naceur, S. Bouabdallah, F. Mercier, and O. Barlet. **Recent developments on the analysis and optimum design of sheet metal forming parts using a simplified inverse approach.** *Computers and Structures*, 78:133 - 148, 2000.

[3] S.H. Park, J.W. Yoon, DY. Yang, and Y.H. Kim. **Optimum blank design in sheet metal forming by the deformation path iteration method**. *International Journal of Mechanical Sciences*, 41:1217 - 1232, 1999.

[4] O. Ghouati and J. C. Gelin. **A finite element-based identification method for complex metallic material behaviours.** *Computational Materials Science*, 21:57 - 8, 2001.

[5] B.F. Rolfe, M.J. Cardew-Hall, SM. Abdallah, and G.A.W. West. **Mapping of geometric shape variation into process parameter settings in a sheet metal context**. In *Proceedings of the 3rd International Conference on Intelligent Processing and Manufacturing of Materials*, Vancouver Canada, 2001.

[6] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. **Active shape models - their training and application.** *CVGIP: Computer vision and Image Understanding*, 61(1):38 - 59, January 1995.

[7] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Pacific Grove, CA, 1984.

[8] SM. Weiss and N.R. Indurkhya. **Rule-based machine learning methods for functional prediction**. *Journal of Artificial Intelligence Research*, 3:383 - 403, 1995.

[9] S. Weiss and N. Indurkhya. *Predictive Data Mining: A Practical Guide*. Morgan Kaufmann, San Francisco, 1998.

[10] T. Mitchell. *Machine Learning*. McGraw Hill, New York, 1997.

[11] B.D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, 1996.

[12] R. Belew, J. McInemey, and N.N. Schraudolph. **Evolving networks: Using the genetic algorithm with connectionist learning**. *Cse technical report* cs90 - 174, University of California, San Diego, 1990.

[13] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press Inc., London, 1990.

[14] W. Mendenhall, R.L. Scheaffer, and D.D. Wackerly. *Mathematics Statistics with Applications*. Duxbury Press, Boston.1986.

[15] B.F. Rolfe, M.J. Cardew-Hall, SM. Abdallah, and G.A.W. West. **Geometric shape errors in forging: developing a metric and an inverse model**. *Proceedings of the Institution of Mechanical Engineers, Part B. Journal of Engineering Manufacture*, 215(9):1229 - 1240, 2001.