Investigating manufacturing data for use within design

M.D. Giess and S.J. Culley

Abstract

The largely informal use of manufacturing information within design often results in such information not being utilised to its full potential. This paper proposes an approach where data generated is collated and analysed using Data Mining methods for reuse in design. This approach, in effect, seeks to mimic the type of knowledge experts accumulate, and embody this within a computational model.

In order to demonstrate the approach the results of an example are presented. A simple mechanical model has been computationally modelled, and aspects of the simulated dynamic behaviour of the mechanism have been recorded. A production run has been replicated by generating 100 mechanism models, each with slightly different geometries which represent variations within manufacture. The dynamic behaviour was then modelled for each in turn, replicating aspects of testing which could be carried out during manufacture.

Data Mining methods were then used to investigate the resultant data, indicating which of the geometric entities most influenced dynamic behaviour and providing a predictive tool which, if given the geometry of a specific model, could estimate the likely dynamic behaviour. Such models allow useful investigation of the manufacturing domain and provide a more formal means of passing manufacturing knowledge into design.

Keywords : Knowledge Acquisition, Design Information Management, Design for Manufacture

1. Introduction

The iterative and collaborative nature of design requires information gathered in one area to be available for use in another, something that methods such as concurrent engineering take into account [1]. Within many practices a clear delineation between design and manufacture still exists, where boundaries may be physical as well as procedural [2] which influence the exchange of information. Figure 1 shows an aggregated and condensed version of the design processes as put forward by experts such as Ullman [3] and Pahl and Beitz [4], and indicates where information from manufacturing data may be used within the design process. This data may take the form of measurements recorded during manufacture or assembly and data recorded during testing. At present the experiences and impact of these data and trends are fed back into design through mostly informal means, such as through some form of network, or occasionally more formally through devices such as review meetings.

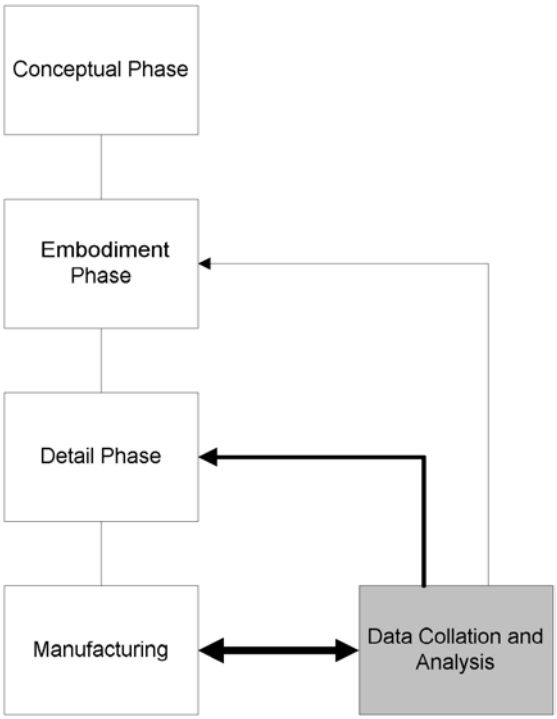

Figure 1 Data Analysis within Design and Manufacture

The thrust of this research work is that the emerging Data Mining (DM) methodologies and techniques present potential opportunities to formalise the collation and analysis of manufacturing data, and transform it into useful and useable design information. DM is a methodology which allows data to be algorithmically interrogated, free of prior assumptions, to reveal novel or unexpected patterns. The authors are currently undertaking research into using DM techniques to establish patterns and relationships within data describing the manufacture of industrial power generation gas turbines [5].

To illustrate some of the underlying principles, this paper presents a more general case, where the results of DM analysis of a computationally-generated linkage mechanism are used to illustrate and validate the approach. The parameters defining the geometry of the mechanism were artificially varied, representing the type of measurements which may be taken during a typical manufacturing operation, and the simulated dynamic behaviour of the mechanism was recorded giving data representative of testing during manufacture and assembly. DM is then used to unearth relationships between these data, indicating the effects of geometry on dynamic behaviour. In this way it is possible to 'design in' greater control over critical aspects of the geometry and infer suitable geometric values to enhance the performance of the design.

Such an approach is most useful in situations where the common approaches to modelling, physical experimentation and computational representations of physical systems, are either impractical or insufficient. It is anticipated that the DM approach would most beneficially be used on data generated during the manufacturing process, thus being in effect an analogy of physical experimentation, but for the sake of convenience and simplicity the example presented in this paper will take data generated from a computational representation. The actual generation of data will be described to indicate the type of data that is suitable for analysis, but it is the *interrogation* of this data that is of greatest significance.

2. Data mining

The underlying motivation of the work presented here is to enable both designers and manufacturers to use Data Mining (DM) techniques on data generated within the manufacturing domain, thus allowing manufacturing data to be investigated with a degree of autonomy, reducing the degree of assumption and manual delineation of search necessary prior to analysis. It is entirely possible that, free from the bias which directs search towards an area suspected or favoured by an individual analyst, more accurate and novel results may be found, This point is highlighted by Smyth [6] who states that the DM approach generates "..previously unsuspected structure and patterns in data."

The field of DM encompasses the collection and interrogation (modelling) of data and evaluation and deployment of the results [7]. DM has been deployed with great success in the financial and marketing arenas, such as by Landrover who improved the response from customers targeted during their marketing campaign from 2% to 85% [8].

Of the raft of algorithms available for modelling, the actual analysis of the data, two lend themselves in particular to this application, namely Decision Tree Induction (DTI) and Artificial Neural Networks (ANNs). DTI [9] automatically generates decision trees which can classify single output variables into different ranges depending upon a series of logical tests of the input parameters. This approach is transparent, where the influences of various parameters are indicated, and is easily implemented. ANNs [10] deal well with noisy, incomplete data and allow for multiple output parameters within an individual model, addressing some of the shortcomings of the DTI approach. ANNs suffer from a lack of transparency and the optimum architecture (layout) of the network is difficult to pre-determine, a major concern as the data requirements for training ANNs increase dramatically with increases in network size.

3. Mechanism problem

The following example, whilst simple, indicates the implementation and scope of the approach and its deterministic nature allows the effects of contrived variations to be evaluated and their effects seen on the simulated performance. In this example a simple mechanism was computationally modelled using the SWORDS constraint modeller [11]. This package allows for mechanisms to be constructed and manipulated to simulate a working cycle, whilst constraint rules ensure specific criteria are met during this cycle. Initially developed for mechanism design, it has proved versatile and has been successfully used for a number of tasks including the modelling of wrist replacements [12].

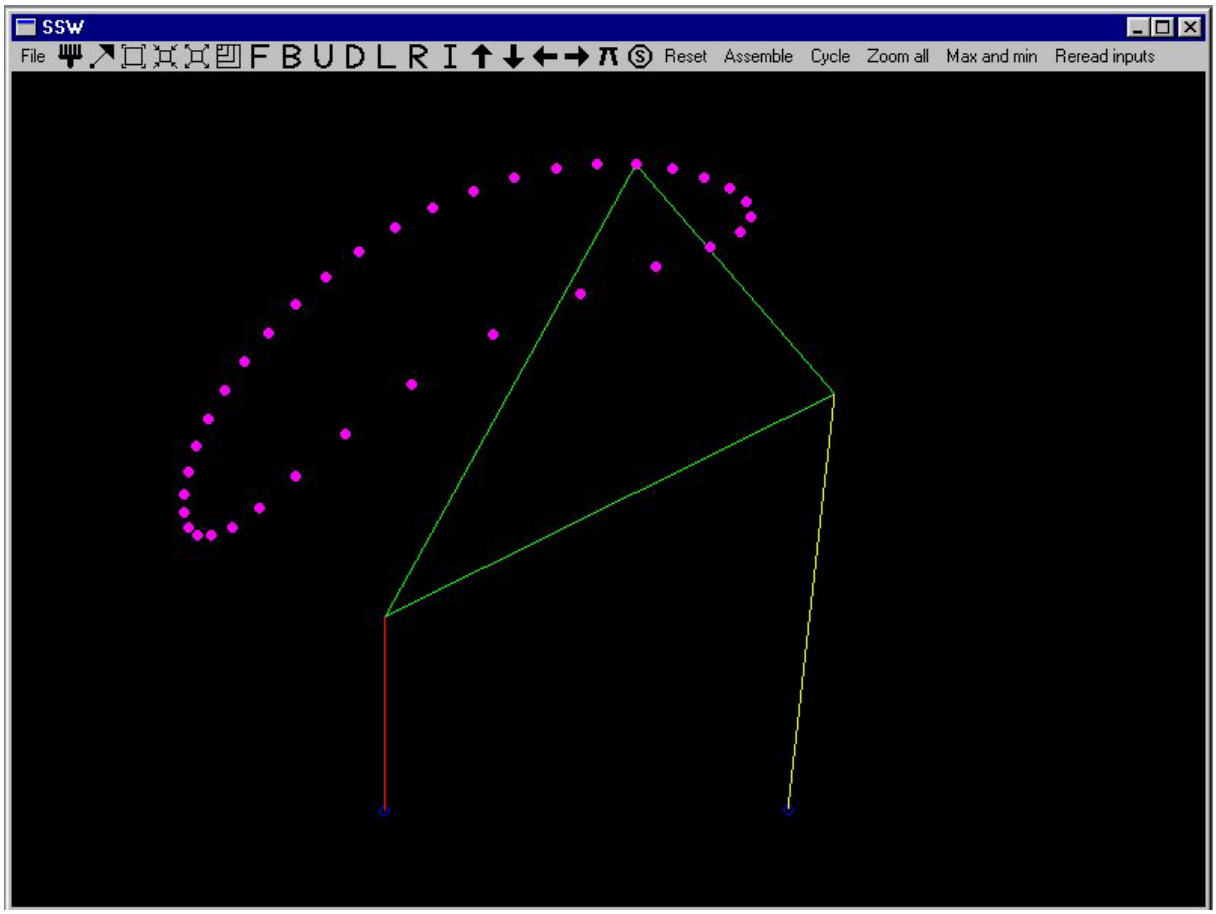

Figure 2 Screenshot of SWORDS Constraint Modeller

A screenshot of the mechanism may be seen in Figure 2. The left leg forms the crank of the mechanism, and when rotated through one revolution the dotted crescent indicates the path of the upper-most point of the assembly, ostensibly representing the working head of the packaging machine. It is the characteristics of this path that primarily determine the operational success of the mechanism, and as such these were selected as the output metrics. These output metrics, together with the input data required to define the mechanism, are listed in Table 1.

Table 1 Input and output parameters for simple linkage

Input	**Label**	**Outputs**	**Label**
Crank length	d1	Max/min head x-displacements	px_max, px_min
2nd leg length	d2	Max/min head y-displacements	py_max, py_min
Crossbeam length	d3	Maximum head velocity	vel_max
Cross-link lengths	d2a, d2b	Maximum head acceleration	acc_max
Crank pivot co-ordinates	p1x, p1y		
2nd leg pivot co-ordinates	p2x,p2y		

The DM approach requires data from a series of exemplars (such as from each item in a production run) to be recorded. In this example the input variables were randomly varied by +/- 5%, representing a rather generous geometric tolerance (thus mitigating computational

rounding errors). One hundred sets of input data were generated in this way, and the SWORDS package was used to simulate models created from each of these 100 input datasets.

3.1. Data mining of mechanism results

In order to be able to act on information gleaned from DM models, it is vital that faith can be placed in their accuracy. Validation of such models is one of DMs cornerstones, and of the various methods of validation that are commonly used, such as bootstrapping (investigated using ANNs in [13]) and the more widely-used cross-validation (CV) [14], the majority require that a certain portion of the available data is kept to one side to test the fully trained model. It is not always apparent which method is the most suitable (certain limitations of CV were identified in [15]) and various approaches were used during modelling.

The two methods of modelling previously described, DTI and ANNs, could each be used to address a different aspect of this problem. The transparency of DTI is useful in determining the areas of the mechanism which are critical to obtaining a specified output, whilst ANNs can provide numerical output prediction.

3.1.1. DTI approach

In classification problems where the output is continuous, the delineation of suitable classification ranges can be informed by identifying and using as boundaries certain values of significance, for example maximum permissible velocities or displacements. In this example the values used, outputs included, are arbitrary, where informed delineation is not possible. Therefore 5 equally divided ranges were constructed, ranges A to E with A representing the lowest velocity and E the highest. This approach gives unequal numbers of instances in each range,. This can be problematic for modelling, as certain areas of the output space will be sparsely populated, but is more representative of data which might be recorded in practise.

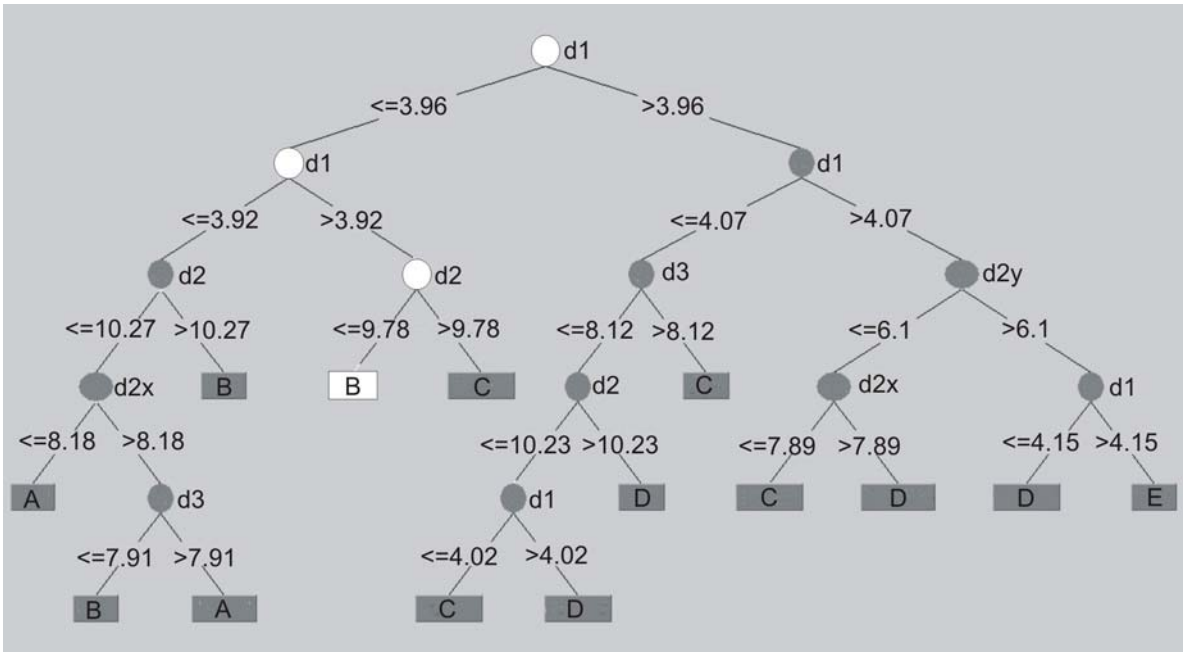

Figure 3 Decision Tree Classifying Maximum Head Velocity

Figure 3 shows a decision tree as generated by the DTI algorithm. This example classifies the maximum head velocity, a parameter whose value has large ramifications upon the

performance (and safety) of the system. This tree was created using the C4.5 algorithm [9], arguably the most common of DTI algorithms. This was implemented within the WEKA Machine Learning Environment [14]. The nodes of the tree contain logical statements which, depending on whether the value of the input variable in question is lesser or greater than the specified value, determine which branch to proceed along. This process continues until a leaf is reached, defining inside which of the 5 specified ranges the maximum velocity is likely to reside.

In Figure 3 the first, root node states that if the value of d1, the length of the left, crank 'leg' of the mechanism is less than a specified boundary value search should proceed down the left branch. It rapidly becomes apparent that this leg contains only leaves specifying ranges A, B and (in certain cases where d1 is only marginally lower than the previously noted boundary value) C. The right branch emanating from the root node leads to leaves classifying into ranges C, D and E. A further node on this right branch repeats the process, once again using d1 as the delineator, where large values result in classification into ranges D and E and smaller values into ranges C and D. Such overwhelming evidence reveals that the maximum velocity of the head is influenced most greatly by changes in the crank length, where longer crank lengths result in larger maximum velocities. It is this form of information that is of great use within design, and illustrates one of the strengths of DTI. Also of significance is the absence of any variable defining the pivot points of the system within the tree, suggesting that these do not influence the maximum velocity of the system, which is in itself a useful piece of information [16]. It is possible to extract individual 'rules' from trees, the following example is taken from Figure 3, and for clarity is highlighted in white within that figure:

```
IF 3.92 < d1 <= 3.96 AND d2 <= 9.78 THEN vel_max = B
```

This rule states that if the parameter d1 lies between 3.92 and 3.96 and the parameter d2 is less than 9.78 then the value for vel_max is likely to reside in range A, corresponding to a low value. These rules form a valuable set of heuristics for use within certain design situations such as initial configurations and scheming.

Table 2 Percentage Accuracy of Various Validation Schemes

Validation scheme	Use of training set	66% Split	20-fold CV	10-fold CV	10-fold CV with Bagging	10-fold CV with Boosting
Accuracy %	97	61.7	50	46	57	56

The accuracy of the model was evaluated using 6 methods in turn, the results of which are given in Table 2. The first method simply passes the training data back through the tree to evaluate the percentage of correctly classified instances, giving a significantly optimistic view of how the tree might perform on new data. The use of a 66% split, where one third of the data is kept to one side for validation, is more representative but is largely dependant on the makeup of the separated sets, where repeating this process with different instances can lead to considerably different results. This is addressed in the CV methods, which divide the training data into separate notionally same-sized sets with the number of folds dictating how many sets to construct. A model is created using the data from all but one of the sets, which is used for validation. This is repeated until all of the sets have been used exactly once for validation, and the results are averaged to give an estimate of overall accuracy. 10-fold validation has been seen in several studies to be the most reliable [14] but there is little theoretical foundation suggesting the superiority of one method over another [14,15].

The use of bagging [17] is similar to CV, where as series of essentially random subsets are created from the main dataset using bootstrapping [18], each with different makeups, and models are created and validated from each subdataset. The overall error and indeed prediction can then be aggregated. This can be effective as 'if perturbing the learning set can cause significant changes in the predictor constructed, then bagging can improve accuracy' [17], where bagging will give an indication of accuracy less subject to the instabilities of the modelling technique. Boosting [19] is an iterative approach, where erroneously classified instances will be weighted in the subsequent model to ensure they receive priority in terms of accurate classification. This process continues for a preset number of cycles, whereupon an aggregate prediction is taken, with the more accurate models receiving higher seeding. This technique is prone to overfitting [20], where the instances are learnt by rote, rather than the underlying pattern, and it is difficult to ascertain the required number of iterations [19].

3.1.2. ANN approach

A feed-forward network with a single hidden layer was selected, extending the possibility of the later use of one of a raft of proposed rule extraction algorithms such as the works surveyed in [21]. Succinctness prevents much detail of such networks being discussed here, for more information the reader is directed to [22]. Whilst robust and capable of dealing well with noisy (error-strewn) data, ANNs suffer from a lack of transparency and do not scale well – the computational expense of training a network (adjusting the internal parameters to fit the required output) and the volume of training data required increases massively with small increases in network size, the so-called 'curse of dimensionality'. The authors anticipate that the requirement for larger datasets will be most problematic within the engineering domain. The parallel use of DTI techniques (and to a lesser extent the promise of rule extraction) address the issue of transparency, and there are measures that can be taken to deal with network complexity. Principle Component Analysis (as discussed in [23]) can be used to reduce the dimensionality of data, thereby reducing network complexity. This approach is widely used in conjunction with ANNs (for example in [24]), and indeed a PCA function forms part of the MATLAB ANN toolbox, the environment used in this study [23]. Whilst sufficient network complexity is required to ensure the underlying pattern of the data is modelled, efforts can be made to ensure the network does not become excessively large. In many cases Simulated Evolution [25], a form of directed random search akin to Darwinian Evolution, has been successfully used to iteratively deduce the optimum network architecture [26,27] and network parameters [28]. This approach is will be introduced in later work where more complex domains are modelled and network architecture becomes more critical, however in this example manual optimisation was used.

The specific instances that are selected for training and validation influence the success of modelling. In [13] the uses of bootstrapping in dataset selection for use in feed-forward ANNs are discussed. Whilst their work focuses on extracting representative datasets from large databases, the use of bootstrap resampling accounts for and mitigates the detrimental influence of erroneous instances, which are not encountered in this example but will become important in practical applications. The work in [29] also considers the composition of datasets and highlights the necessity of boundary samples within the training data, where examples covering the extremes of each solution space are included. This situation is extremely difficult to engineer, but by using bootstrapping it is possible to trial different dataset compositions and select the most accurate – a series of networks are trained using the bootstrap samples, and the network with the highest validation accuracy can be selected for use.

The developed network utilised 9 nodes in the input layer (the Cartesian components of both pivot points and the five link lengths) and 6 in the output (the Cartesian components of minimum and maximum displacement and the maximum velocity and acceleration). The number of nodes in the hidden layer was varied, 3 runs were completed with 4, 5 and 6 nodes in this layer. The computationally quick Levenburg-Marquardt algorithm [30] was used for training as numerous bootstrap iterations would require processing.

A linear regression between predicted and achieved network output was used to provide a metric of the network accuracy, a standard Matlab post-analysis function [23], although other packages allow for the use of alternate measures of correlation. This function returns a correlation value indicating the level of agreement between the predicted and desired outputs, ranging from 0 (no correlation) to 1 (complete agreement). Table 3 shows the highest correlation coefficients for a series of 3 runs with different numbers of hidden nodes in the output layer. The 6 individual correlation coefficients correspond to the 6 outputs, and whilst a higher coefficient might have been noted on another network within the sample these were recorded from the network with the greatest summed coefficient.

Table 3 Correlation Coefficients for ANN Output

No of Nodes	Summed r-value	Average r-value	R1	R2	R3	R4	R5	R6
6	5.3732	0.8955	0.83481	0.8671	0.77121	0.91288	0.86301	0.84536
5	5.312	0.8853	0.89949	0.89444	0.82829	0.86127	0.91398	0.91453
4	5.2614	0.8769	0.90264	0.86003	0.88115	0.84049	0.89119	0.8859

It can be seen that the fit is good, as the individual correlation values are approaching 1, although there appears to be some drop-off in the summed correlation coefficient with a reduction in the number of hidden layer nodes. This would suggest that a larger number of hidden layer nodes would result in a more accurate network, although must be balanced against a tendency for over-fitting, where the network records the individual instances, errors included. The use of a small a network as possible, whilst ensuring integrity, generally avoids this problem and also reduces computational overheads.

3.2. Discussion of results an developments of DM

The predictive models created as part of the DM methodology provide, once trained, useful tools to estimate the likely performance of a system given information regarding its early characteristics. With some thought, these models can be used in a greater capacity, to indicate which of these early characteristics most influence the performance and to give some idea whether these characteristics should be muted or amplified.

In the mechanism example described previously, it was noted that the DTI model for predicting maximum head velocity revealed within its structure that longer crank lengths act to increase this maximum velocity. Whilst perhaps easily deduced by other methods, this indicates the type of information it is possible to extract from a DTI model. The parallel use of ANNs allow for the much more granular prediction of maximum velocity given the geometric properties of the system, where a numeric output replaces the ranges of the DTI approach. It is also possible to extract rules, similar to those generated by the DTI approach, from trained ANNs. This will be attempted in later work, along with a sensitivity analysis of the network, where the effects of perturbations in the input parameters are traced through to the outputs, indicating the degree of influence each input has on the output [31]. To ensure the approach is applicable to actual engineering problems, this approach is being concurrently developed on data obtained during the manufacture of gas turbine engines [5].

4. Conclusions

Applying DM methods to manufacturing data has the potential to reveal useful information for designers. The complementary parallel use of DTI and ANN techniques have provided both a method for inferring parameters critical to, or highly influential of, system behaviour and also a predictive tool for determining likely system behaviour given specified conditions. In this paper these techniques have been trialled and thus validated on a deterministic model, and seen to provide information useful in understanding and estimating the likely performance of that model with good levels of accuracy. The rules that are generated, although complex, show how the approach can provide the 'unsuspected patterns' referred to with respect to the DM process, and can thus be used to potentially support the design and manufacturing processes.

References

[1] Kusiak, A. (ed), "Concurrent Engineering: Automation, Tools and Techniques", John Wiley and Sons, USA, 1992

[2] Colton, J.S., "An Intelligent Design for Manufacture System" , in "Concurrent Engineering: Automation, Tools and Techniques" ed Kusiak, A. John Wiley & Sons, USA, 1992

[3] Ullman, D.G., "The Mechanical Design Process", 2nd ed. McGraw-Hill, Singapore, 1997

[4] Pahl, G. & Beitz, W., "Engineering Design: A Systemic Approach". Springer-Verlag, Great Britain, 1996.

[5] Giess, M.D., Culley, S.J. and Shepherd, A., "Informing Design with Data Mining Methods". Proceedings of ASME DETC2002/DAC, Montreal, 2002, p98

[6] Smyth, P., "Data mining: data analysis on a grand scale?", Statistical Methods in Medical Research, vol. 9. 2000, pp 309-327.

[7] CRISP-DM Consortium, "CRISP-DM 1.0: Step-by-step users guide", [WWW] http://www.crisp-dm.org. (accessed Feb 7, 2003)

[8] Forcht, K.A. & Cochran, K., "Using data mining and data warehousing techniques", Industrial Management and Data Systems, vol. 99(5), 1999, pp 189-196.

[9] Quinlan, J.R., "Induction of Decision Trees", Machine Learning, vol.1. 1986, pp 81-106.

[10] Rumelhart, D.E. & McClelland, J.L., "Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations" MIT Press, USA, 1986

[11] Kenney, L.P.J., Rentoul, A.H., Twyman, B.R., Kerr, D.R., Mullineux, G., "A Software Environment for Conceptual Mechanism Design", Proceedings of the Institution of Mechanical Engineers, Vol. 211(C), 1997, pp 617-625

[12] Leonard, L., Sirkett, D.M., Langdon, I.J., Mullineux, G., Tilley, D.G., Keogh, P.S., Cunningham, J.L., Cole, M.O.T., Prest, P.H., Giddins, G.E.B. and Miles, A.W., "Engineering a new wrist joint replacement prosthesis – a multidisciplinary approach", Proceedings of the I MECH E Part B Journal of Engineering Manufacture, vol. 216(9), 2002, pp. 1297-1302

[13] Dupret, G. & Koda, M., (2001) "Bootstrap re-sampling for unbalanced data in supervised learning", European Journal of Operational Research, vol. 134, 2001, pp 141-156

[14] Witten, I.H. & Frank, E., "Data Mining", Morgan Kaufmann, USA, 2000

[15] Goutte, C., "Note on free lunches and cross-validation". Preprint: Neural Computation, Vol. 9(6), 1997, pp. 1246-1249

[16] Westphal, C. & Blaxton, T., "Data Mining Solutions – Methods and Tools for Solving Real-World Problems". Wiley, USA, 1998.

[17] Breiman, L., "Bagging Predictors", Machine Learning, Vol. 24., 1996, pp 123-140.

[18] Efron, B. & Tibshirani, R.J., "An introduction to the bootstrap", Chapman and Hall, London, 1997

[19] Freund, Y. and Schapire, R.E., "A Short Introduction to Boosting" (In Japanese, Translated by N. Abe), Journal of Japanese Society for Artificial Intelligence, vol.14(5), 1999, pp 771-780.

[20] Schapire, R.E. and Singer, Y., "Improved Boosting Algorithms using Confidence-rated Predictions", Machine Learning. Vol. 37, 1999, pp 297-336.

[21] Andrews, R., Diederich, J. and Tickle, A.B., "Survey and Critique of techniques for extracting rules from trained artificial neural networks", Knowledge-Based Systems, Vol. 8(6), 1995, 373-389.

[22] Fu, L.M., "Neural Networks in Computer Intelligence", McGraw-Hill, Singapore, 1994

[23] Matlab, "Neural Network Toolbox Users Guide Version 3". Mathworks inc., USA, 1998

[24] Li, W., Li, D. and Ni, J., "Diagnosis of tapping process using spindle motor current", International Journal of Machine Tools and Manufacture. Vol. 43, 2003. pp 73-79.

[25] Fogel, D.B., "An Introduction to Simulated Evolutionary Optimisation", IEEE Transactions on Neural Networks, Vol. 5(1), 1994, pp 3-14

[26] Angeline, P.J., Saunders, G.M. and Pollack, J.B., "An Evolutionary Algorithm that Constructs Recurrent Neural Networks", IEEE Trans on Neural Networks, Vol. 5(1), 1994, pp 54-65

[27] Fang, J. and Xi, Y., "Neural network design based on evolutionary programming", Artificial Intelligence in Engineering, Vol. 11, 1997, pp 155-161.

[28] Pham, D.T. and Karaboga, D., "Self-Tuning fuzzy controller design using genetic optimisation and neural network modelling", Artificial Intelligence in Engineering, Vol. 13, 1999, pp 119-130.

[29] Mehrotra, K.G., Mohan, C.K. and Ranka, S., "Bounds on the Number of Samples Needed for Neural Learning", IEEE Trans on Neural Networks, Vol. 2(6), 1991, pp 548-558.

[30] Hagan, M.T. and Menhaj, M.B., "Training Feedforward Networks with the Marquardt Algorithm", IEEE Transactions on Neural Networks, Vol. 5(6), 1994, pp 989-993.

[31] Buczak, A.L and Ziarko, W. "Neural and Rough Set Based Data Mining in Engineering Applications" in "Handbook of Data Mining and Knowledge Discovery", eds Klosgen, W and Zykow, J.M. Oxford University Press, 2002

Corresponding Author's contact details:

M.D. Giess, Faculty of Engineering and Design, The University of Bath, Claverton Down, Bath, BA2 7AY, UK
Tel: Int +44 1225 385366 Fax: Int +44 1225 386928 email: m.d.giess@bath.ac.uk