Int J Adv Manuf Technol (2008) 37:314–322
DOI 10.1007/s00170-007-0964-5

ORIGINAL ARTICLE

Yield prediction for seamless tubing processes: a computational intelligence approach

Saurabh Dwivedi · Samuel H. Huang · Jun Shi · William H. VerDuin

Received: 13 October 2006 / Accepted: 23 January 2007 / Published online: 16 February 2007
© Springer-Verlag London Limited 2007

Abstract Seamless tubing is the most commonly used process for high quality pipe products due to its ability to provide material consistency. However, the yield is hindered by complex workmanship and manufacturing operations, with the resulting risk of flaws, low productivity, and high manufacturing costs. This paper describes a systematic procedure to implement a data-driven process modeling technology in a seamless tubing factory, using neural networks to model relationships between controlled and uncontrolled process parameters and the yield. The issues of data collection/analysis and dimensionality reduction are discussed. It was found that model predictive ability improves when only important variables are included in the model. A software tool is developed, which offers such functions as response surface visualization and yield prediction.

Keywords Data collection · Dimensionality reduction · Neural networks · Seamless tubing process

1 Introduction

Pipe products made from carbon and alloy steel are commonly used in oil and natural gas exploration and production, fluid transmission, and the construction industry. A market survey of the American steel industry indicated that the American oil rig count experienced a decline in 2001, a smooth year in 2002, and an increase during the beginning months of 2003. This suggests an increasing demand for seamless tubing for gas and oil production [1].

Steel tubes are made by several manufacturing processes, including forming, joining, and casting. Welding (joining process) and extrusion (forming process) are the most commonly used processes to produce tubes of different sizes. Extrusion processes generally provide a higher strength to weight ratio and better chemistry than welding, due to difficulties in welding some carbon and alloy steels. The welding process is typically used for small diameter tubes. This process begins with a strip of metal, usually in coil form. The strip is fed into machines that form the flat strip into a continuous tube with an open seam. The seam is then continuously welded to be permanently joined. Further drawing and annealing operations are carried out until the final size required is reached.

The alternative process is seamless tubing, which is mostly used for tubes with large outer diameters. Seamless tubing is produced without welding down the length of a seam. In this process, the steel, in the form of a red-hot solid billet, is pierced and extruded to form a hollow pipe. The diameter of the tube is then increased and the wall thickness decreased by a succession of "rolling" and "rocking" processes to swage the inside diameter larger while forming the wall thickness successively smaller. Successive processes then straighten the tube while improving internal and external surface finish.

Welded tubing generally provides better control of wall thickness than the seamless tubing process. However, the welded seam may introduce manufacturing difficulties and resulting flaws. Although a homogenizing process can be applied to improve the tube quality at the seam, it also

S. Dwivedi · S. H. Huang (✉) · J. Shi
Intelligent Systems Laboratory, University of Cincinnati,
Cincinnati, OH 45221, USA
e-mail: sam.huang@uc.edu

W. H. VerDuin
VerTech LLC,
Chagrin Falls, OH 44022, USA

increases the manufacturing cost. In contrast, seamless processing is much more expensive to implement, generally costing two to four times more than seamed tubing in sizes up to two inches and four times more in sizes of three inches or more. It also gives a less uniform wall width. Its consistency in material properties, however, gives it a good reputation in industry, for numerous applications such as rollers, shafts, fluid transfer lines, sleeves cylinders, etc. [2].

Figure 1 shows a procedure of a tubular factory to manufacturing tubes of diameters ranging from 10.75 to 26 inches. The billets, coming out of casters or ovens, are gripped by rolls and go through piercers that form a hole through its length. The tubes are then reeled (expanded) through a succession of rolls to achieve diameter requirements. After passing through a series of sizing rolls, the tubes become truly round and sized to the exact required diameter. The process then ends with finishing and inspection.

This is a long, complex and costly manufacturing line. To ensure the high quality and continuous production of tubing, the factory undertakes periodic checking and replacement of equipment and tools of the manufacturing system. However, at the site studied, yield and quality objectives are not fully met by current maintenance strategy. The factory is interested in improving yield and quality, and believes that it is worthwhile to carry out analysis of the manufacturing system to find factors affecting production yield. A model to predict yield will be useful to, for example, understand how types and condition of plugs and reelers affect the yield. This would enable equipment maintenance strategies to be driven by yield, as well as observation of equipment wear. Towards this end, we have undertaken an implementation of computation intelligence to simulate the methods by which an expert manufacturing engineer solves such problems.

Since artificial intelligence (AI) emerged in the mid 1950s, a number of tools have been used in engineering applications to solve the problems where human intelligence is normally required. Major types of AI tools and techniques were reviewed in [3], including traditional expert systems using IF-THEN rules, inference theory using fuzzy logic, computational intelligence using neural networks or genetic algorithms. The expert system is the most popular approach adopted by the industry to solve numerous problems. However, this approach suffers from the so-called knowledge acquisition bottleneck, since it is challenging and time-consuming to acquire and represent knowledge in terms of IF-THEN rules. In other words, the bottleneck hinders the effectiveness and efficiency of building expert systems [4]. In addition, the rule base that is the core element of an expert system tends to have flaws such as redundancy, inconsistency, inflexibility, etc., which finally impairs the performance of the system [5].

Alternative approaches to computational intelligence such as neural networks and genetic algorithms are gaining attention. Genetic algorithms are a new advanced technology mainly used for optimization. Neural networks is a modeling method based on a learn-by-example paradigm where the problem solving knowledge is automatically acquired according to the historical data records presented to them [6]. The intelligence inside neural networks is implicit, which is criticized by many researchers. However, this issue can be alleviated when techniques are developed to extract rules from trained neural networks, with or without some priori knowledge [7–9].

A data driven approach is developed to model the manufacturing process, by building up the relationship

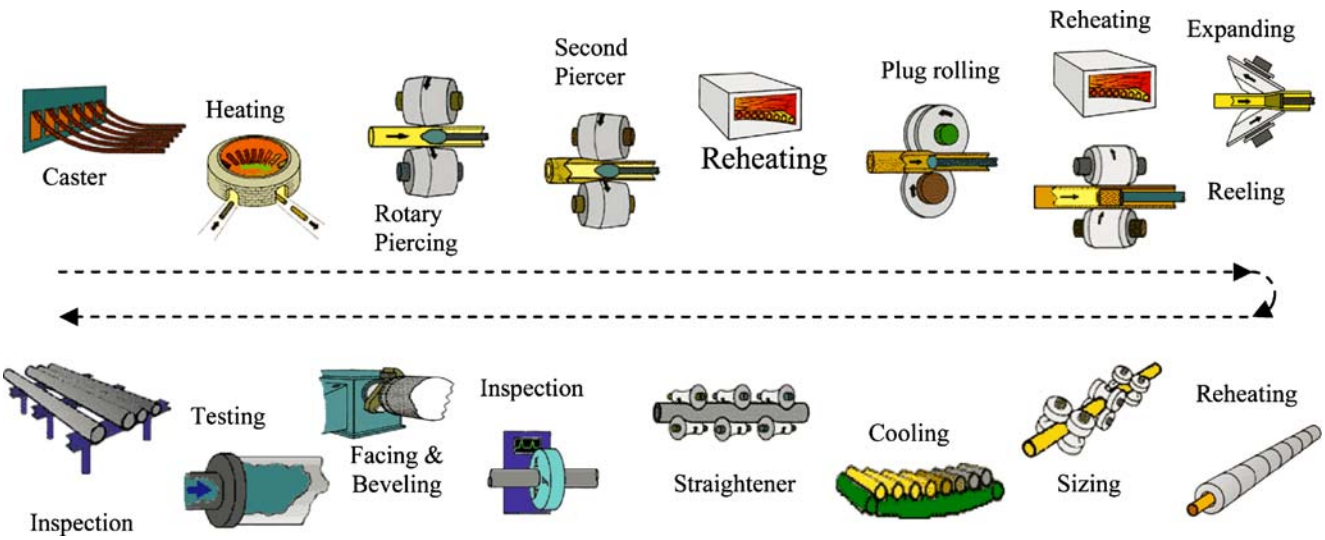

Fig. 1 Seamless tubing processes

between manufacturing conditions and production yield. Figure 2 gives an overview of this methodology. In this methodology, firstly the manufacturing engineer and data analyzer together carry out data collection from the manufacturing locale. The data is pre-processed and stored in certain computer file formats. The data analyzer then explores the data set using computational techniques such as statistics. The results might be fed back to the data collection system for verification and perhaps prompt additional data acquisition. Normally, many records with a large number of parameters are gained from the raw data. Not all parameters will be necessary or helpful for modeling the manufacturing process. The dimensionality reduction technique is introduced to deal with the high volume of data and potentially unnecessary data types. The retained parameters are highlighted and others are discarded. The final data gained from above steps is divided into two sets. One set is used as the training data set to build the neural network models. The accuracy of these models is then verified by the testing data set. The trained neural networks become the core element of the knowledge system, which can help workers, engineers and managers to find trends in manufacturing process and make decisions regarding process improvement.

The following sections provide detail regarding the implementation of this methodology in a seamless tubing mill. The discussion of data collection and analysis is provided in Sect. 3. Section 4 describes the development of the yield prediction model. The issue of dimensionality reduction is highlighted. Lessons learned are then presented.

2 Data collection and analysis

Data collection is performed at the seamless tubing mill, focusing on the plug rolling, reeling and expanding processes illustrated in Fig. 1. The data collection is carried out in five days of two different weeks. A sequence number, called AMALOG, is assigned to track product as it is transformed from the beginning as billets (at the so-called hot end of the process), to the finishing end (the cold end). Parameters monitored at the hot end are called input parameters on the assumption that they drive process outcomes, while those at the cold end are called output parameters because they represent the process outcomes such as yield. The data was collected in two phases. Parameters monitored in phase I are listed in Table 1. Table 2 summarizes the number of records acquired in phase I.

One of the problems observed in the data set is that of incomplete records. Elements of records are seen to be missing at both at the input end (hot) and the output end (finishing/cold). Although literature shows a number of methods proposed to fill the missing values [10, 11], these data samples can be directly discarded before further data processing as there is enough data left and the missing data mechanism is completely random. Although 596 tubes were monitored; 380 data samples remain after discarding those with missing elements. An additional four records are discarded because the tubes they represent were scrapped before they reached the rotary billet heating furnace, which is the beginning of the tubing process under consideration. A total of 376 tube records were used for analysis and model building.

Another problem is the inconsistencies in the data. Several can be identified. Firstly, the data is collected at two separate locations: the input end and the output end. Although an AMALOG sequence number is assigned to each tube, the number is difficult to recognize at the finishing end, due to the heating and rolling processes to which the tube is subjected. The possibility of error-in-matching the data from the input end and the output end for a single tube cannot be denied, with the occurrence of intermittent problems suspected in the transfer of data from

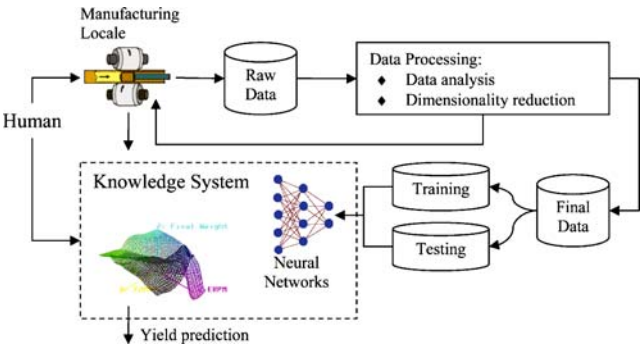

Fig. 2 The data driven approach

Table 1 Parameters monitored (phase I)

Input parameters	Output parameters
(1) Which plug is used (A or B);	(1) Tube outside diameter;
(2) Shell temperature;	(2) Actual cooling table length;
(3) Motor load west (start);	(3) Weight;
(4) Motor load west (end);	(4) Length after cropping; and
(5) Motor load east (start);	(5) Yield.
(6) Motor load east (end);	
(7) Disc speed (east end);	
(8) Disc speed (west end);	
(9) Tube outside diameter;	
(10) To which reeler the tube goes (east or west);	
(11) Heat; and	
(12) Wall thickness	

Table 2 Numbers of data collected (phase I)

Date	Oct 1st		Oct 2nd		Oct 3rd		Oct 4th		Oct 9th		
Tube numbers	Lot#1	68	Lot#1	96	Lot#1	96	Lot#1	8	Lot#1	20	
	Lot#2	40	Lot#2	90			Lot#2	28	Lot#2	11	
	Lot#3	20					Lot#3	6	Lot#3	57	
							Lot#4	6			
							Lot#5	30			
							Lot#6	20			
Total	128		186		96		98		88		596

system to system along the production line. Finally, the heat and limitations in access to the tubes inherent to this manufacturing environment hinders the engineers from measuring values such as finished length and diameter accurately.

This "messiness" affects the performance of the data-driven prediction model. A simple neural network is built up with 12 input parameters, 1 output parameter (yield), and 12 hidden neurons. The final MAP (Mean Absolute Percentage error) values for the training data and testing data are 6.44% and 11.7%, respectively. Moreover, using phase I data, we obtained a regression model where the R^2 value is only 0.66. It means that the predictive model explains only 66% of the observed deviations in process output. The level and discrepancy between the two values of MAP and the low R^2 value are believed to be a result of data inconsistencies.

Since our modeling approaches are solely based on the data collected, data validity is crucial to the resulting model accuracy. Due to the difficulty encountered in phase I data collection, it was realized that there was a possibility that the collected data was not reliable. Therefore, a second phase of data collection was performed. Keeping in mind the problems faced during phase I data collection and analysis, more sophisticated data collection methods were used in phase II. Specifically, information on most of the input parameters was collected electronically. To help ensure accurate tracking of tubes through the process, a numbering system was implemented that was independent of that used by the plant. This additional code called the 'KBE number' was marked on the tubes passing from the rotary mill to the finishing end.

Table 3 includes all parameters monitored in phase II. Data on the first six input parameters are recorded electronically in time series. The average of the time series is used for analysis. Although phase II data collection was much better coordinated, there are still problems. Duplicate electronic readings are observed for certain parameters. In addition, the missing value problem still exists, although at a much smaller scale. Specifically, data is collected on 328 tubes, while 272 records contain complete information. These complete data samples are used for analysis and model building.

3 Yield prediction models

3.1 Model development

A neural network model was developed for yield prediction. The yield of a tube is defined as the ratio of its final weight to its initial weight, as this is a key productivity measure. Such definition of yield requires matching the data describing incoming steel billets to the data describing finished tubes. Matching the data is done by time stamping. Unfortunately, phase II data also had duplicate records and incorrect matching of input to output data. Furthermore, it was found to be not possible to obtain original weight information for individual tubes. However, the lots on which the data were collected do not have a significant variance in the initial weights of individual tubes, and so yield was calculated on the basis of an assumed average initial weight.

The multi-layer back-propagation neural networks are used for model building. This type of network is characterized by one input layer of neurons, one output layer and several hidden layers (normally one or two hidden layers are used). Each layer contains several neurons with certain

Table 3 Parameters monitored (phase II)

Input parameters	Output parameters
(1) East disc speed (ERPM)	(1) Crop length N/S
(2) West disc speed (WRPM)	(2) Recut N/S
(3) Motor load west (WAMPS)	(3) Final length
(4) Motor load east (EAMPS)	(4) Final weight
(5) Temperature of the shell (Temp)	(5) Crop defect code
(6) East disc position (EDisc)	(6) Recut defect code
(7) Plug A/B	
(8) Reeler E/W	
(9) Initial weight	
(10) Initial length	

processing functions. Between two layers there are links represented by numerical weights. In this case, eight input neurons and one output neuron are used, related to input and output parameters respectively.

The training is carried out for networks with one hidden layer and with a single neuron in it. Then we tried to increase the numbers of neurons and layers and checked the training and testing errors. The final model provided is the one with the least training and testing errors. The network has two hidden layers in it, each layer has eight neurons.

A total of 272 data samples are available. Approximately 64% of the data is used as training data and the remaining as testing data. The MAP is found to be 4.62% (95.38% accuracy) and 10.47% (89.53% accuracy) for the training and testing data sets, respectively. Note that the model accuracy is better than that achieved in phase I, as shown in Table 4. The testing error is still much higher than the training error (almost the same as in phase I). This still indicates that there are problems in the process parameters collected. However, since the data collection is well coordinated, the "messiness" in the data should have been significantly reduced. The high testing error with a low training error is mostly caused by model overfitting. It happens commonly in the neural network approach, especially when irrelevant parameters are included into the model and/or too many training epochs are used. This leads to the dimensionality reduction of the model, which is discussed in the subsection that follows.

3.2 Dimensionality reduction

When developing data-driven models, the domain experts may have enough knowledge of the process to identify the appropriate variables for which data should be collected. Unfortunately, this is not always true, in which case some important process variables may be left out while unrelated variables are included in data collection. Including unrelated variables in a process model has many undesirable consequences, known as the *curse of dimensionality*. Because the neural network model is trying to fit unrelated information, model development becomes more complex and time consuming. The resulting model usually suffers from the "overfitting" problem, i.e., the model will overfit the training data set and have a much lower accuracy when applied to the testing data set. Therefore, we applied a clustering-based method to reduce the model dimensionality.

Table 4 Model accuracy

Yield prediction model (MAP)	Accuracy (Training)	Accuracy (Testing)
Complete parameters	95.38%	89.53%
Dimensionally reduced	94.20%	92.93%

Dimensionality reduction research originated from feature extraction techniques, which mainly contain four steps of operations: (1) generate feature subsets; (2) develop evaluation criteria; (3) stopping; and (4) validation [12]. Much work has been done in steps 1, 3 and 4; and a number of methods have been proposed [13–15]. However, most of these methods suggested generally deal with discrete values. In industrial applications, a common scenario is the need to perform dimensionality reduction on both continuous and discrete parameters. The method should be suitable for a classification problem and also for the approximation problem where the response variable is continuous.

A clustering-based approach is developed to reduce the dimensionality of the model. The major idea of this method is to remove as many parameters as possible from the model while preserving its clustering accuracy. The reduction could be an iteration of two steps of operations. The first step is to find a suitable parameter by ranking them and then selecting one of them based on some criterion. In the proposed method, PCA (principal components analysis) is used. Details of PCA can be found in [16].

The second step is an algorithm to discard or retain a parameter. It begins with subtractive clustering [17]. Consider a collection of n data samples $\{x_1, x_2, ..., x_n\}$ in R^p where x_{kj} is the jth coordinate of the kth data sample (k= 1, 2, ..., n; j=1, 2, ..., p). Without loss of generality, assume that the 1st to the qth coordinates represent the input parameters (i.e., process setting) and the (q+1)th to the nth coordinates represent the output parameters (i.e., process response). Note that here we treat a data sample as a point (henceforth referred to as a *data point*) in the p-dimensional space, and do not distinguish the input and output parameters. Let $D(x_k, x_i)$ denote the distance between x_k and x_i, usually the Euclidean distance (any other suitable distance measures are acceptable). A mountain function at data sample x_i can be calculated as $M(x_i) = \sum_{k=1}^{n} e^{-\alpha D(x_k, x_i)}$ here α is a positive constant.

It is evident from the above equation that a data point will have a higher mountain value when it has more nearby data points. Such a data point is more suited to be a potential cluster center of the given dataset. Subtractive clustering starts by calculating the initial mountain values for all the data points, denoted as $M_1(x_i)$, i=1, 2, ..., n. Then, it finds $M_1(x_m)$=max$[M_1(x_i)]$, and thus, identifies x_m as the first cluster center. Let x_1^* and M_1^* denote the first cluster center and its mountain value, respectively, i.e., $x_1^* = x_m$, $M_1^* = M_1(x_m)$. Mountain values for all the data points are then updated using the equation $M_k(x_i) = M_{k-1}(x_i) - M_{k-1}^* \times e^{-\beta D\left(x_{k-1}^*, x_i\right)}$, where β is a positive constant.

The update brings down the mountain value of the first cluster center and that of its near-by points. In this way, the second cluster center (which has the maximum updated mountain value) will be found somewhere else. This procedure repeats until the current maximum mountain value falls below a certain level compared with the original maximum mountain value. In other words, subtractive clustering stops when $M_k^* < \delta M_1^*$, where δ is a positive constant between 0 and 1.

Now we can calculate the evaluation criterion based on the clustering results. For a classification problem (with a discrete output variable), PCE (Percentage Classification Error) is used, defined as following:

$$PCE = \sum_{k=1}^{K} CE_k \Big/ M = \sum_{k=1}^{K} \left[\sum_{l=1}^{L} c_{kl} - \max_{l}(c_{kl}) \right] \Big/ M$$

Where,

M	Number of data samples (one data sample is also called one pattern)
CE_k	Classification error of cluster k. It is a percentage of patterns that don't belong to the class with maximum patterns.
K	Number of clusters
L	Number of classes (output values)
c_{kl}	Number of patterns that belong to cluster k and class l

For approximation problem (with a continuous output variable), we use *EI* (Evaluation Index), given by

$$EI = Inter - Cluster\ Variance\,(Separation) / Intra - Cluster\ Variance\,(Homogeneity)$$

$$Inter - Cluster\ variance = \sum_{k=1}^{K} \left[\sum_{i=1}^{M_k} D(g_i, C(g_i)) \Big/ M_k \right]$$

$$Intra - Cluster\ variance = \sum_{k1=1}^{K} \left\{ M_{k1} \sum_{k2=1}^{K} [M_{k2} D(C_{k1}, C_{k2})] \right\} \Big/ \sum_{k1=1}^{K} \left[\sum_{k2=1}^{K} (M_{k1} M_{k2}) \right]$$

Where,

g_i	the i^{th} pattern that belongs to a certain cluster
$C(g_i)$	the center of the cluster that pattern g_i belongs to
M_k, M_{k1}, M_{k2}	the number of patterns in cluster k, $k1$ and $k2$
D	the distance function between two variables
C_{k1}, C_{k2}	the centers of the cluster $k1$ and $k2$
K	Number of clusters gained after subtractive clustering

The pseudo-code algorithm of dimensionality is shown in Fig. 3, based on following expressions:

N	Number of input parameters (dimensionality)
x_{ij}	Input data (i = 1, 2, ..., M; j=1, 2, ..., N)
y_i	Output data (i = 1, 2, ..., M)
θ	Threshold value for *PCE* or *EI*
r	Clustering radius
α	PCA percentage threshold
C_k	Cluster centers (k=1, 2, ..., K)
PC_{ij}	Principal component index (i=1 to N; j=1 to N). It is actually the correlation coefficient between one parameter and the principal component.
Con_i	Contribution of input parameters (i=1 to N)

This algorithm requires the user to define the following parameters: cluster radius r, PCA percentage threshold value α, and evaluation threshold value θ. The selection of cluster radius r influences the final number of clusters K. From our experiment, a good value for r is between 0.1 and 0.25. The smaller α is, the more parameters will be evaluated. If the total number of parameters is small (like in our case which is 8), á can be set to 0. Otherwise, we can plot the values of PC_{ij} and observe the plot to help us determine an appropriate α, i.e., find the cut-off point where the change of *PCij* accelerates. The selection of θ influences the number of parameters retained and thus the final model accuracy. We recommend a trial-and-error procedure to select θ. Specifically, the value of θ can be set higher initially. After the retained parameters are used to build a model, evaluate model accuracy. If higher accuracy is desired, observe the relationship between θ and the model accuracy and reduce the value of θ accordingly.

```
Read the data (x_ij, y_i). i = 1 to M. j = 1 to N.
Read θ, r, and α.
Normalize (x_ij, y_i) and get (x'_ij, y'_i).
Perform subtractive clustering on (x'_ij). Get C_k = subclust(data, r). k = 1 to K.
Repeat until all the parameters are checked {
    Perform PCA on (x'_ij). Get PC_j. j = 1 to N.
    Select PCs and determine L. l = 1 to L. Such that Σ_{l=1}^{L} PC_il ≥ α but Σ_{l=1}^{L-1} PC_il < α.
    Calculate Con_i = sqrt(Σ_{l=1}^{L} (PC_il)^2) . i = 1 to N.
    Repeat until all variables are checked but no parameter is discarded {
        Remove the variable with the least contribution. Say j'
        Perform subtractive clustering on the new data (Parameter j' removed)
            Get C_k = subclust(data, r). k = 1 to K.
        Calculate PCE or EI.
        If (PCE or EI > θ)
            Retain the Parameter
        Else
            Discard the Parameter
    }
}
```

Fig. 3 Dimensionality reduction algorithm

For the yield prediction model, only two parameters are retained after dimensionality reduction: EAMPS and Plug. (EAMPS being the current for roll drive E and Plug being the nomenclature of the forming plug used). A neural network model is developed with 64% of the data for training and 36% for testing. The results, compared with that obtained before dimensionality reduction, are shown in the following table. From the table, it can be seen that the dimensionally reduced model performs well compared with the full model. It actually has a higher accuracy when applied to the testing data. This also shows that EAMPS and Plug are indeed important in determining the yield.

4 System implementation

The commercial software tool MATLAB is used for developing the neural network model. MATLAB has a variety of neural network training and testing facilities that can be used to build up an expected model. For example, various neurons with different transfer functions such as "hardlim", "log-sigmoid" and "tan-sigmoid" can be tried; finally we will select the most suitable functions. All these are only demanded during model development, while the users of the model only need a deliverable system.

Our design objectives included the ability to run a delivered predictive modeling system on a personal computer, without any support from MATLAB, since some companies do not have the MATLAB software. Moreover, a customized software tool is easier to use. The interface and functions can be designed directly according to the requirements of final users. So, the structure and setting of the trained neural networks are exported. A stand-alone computer program is developed based on the information exported, including neuron numbers, connecting weights between them and the bias values for all neurons in the network. Two major functions are offered. One is to show the response surfaces of the processing model. The user can observe the impact of varying any two input parameters on the desired response, the yield value. The user can find what regions of input parameters are sensitive to the yield output. The tool also offers a yield prediction function. The user inputs detailed values of input parameters, and the tool runs the neural network model and provides the predicted yield.

Figure 4 shows the main interface of the software. The top-side of the interface is the menu. The toolbar below the menu gives short-cuts for menu functions. The main part of the screen displays the response surface. In Fig. 4, the relationship between PLUG, EAMPS and Final Weight is shown. The objective is to maximize final weight, and the user may employ this and other response surfaces to determine approaches and tradeoffs to achieve maximum final weight. By comparing this response surface to other response surfaces, it also indicates that the impact of the two parameters is larger than other parameters. The surface also can help find certain parameter values which could be possible to gain highest final weight, by rotating, zooming and shifting the surface.

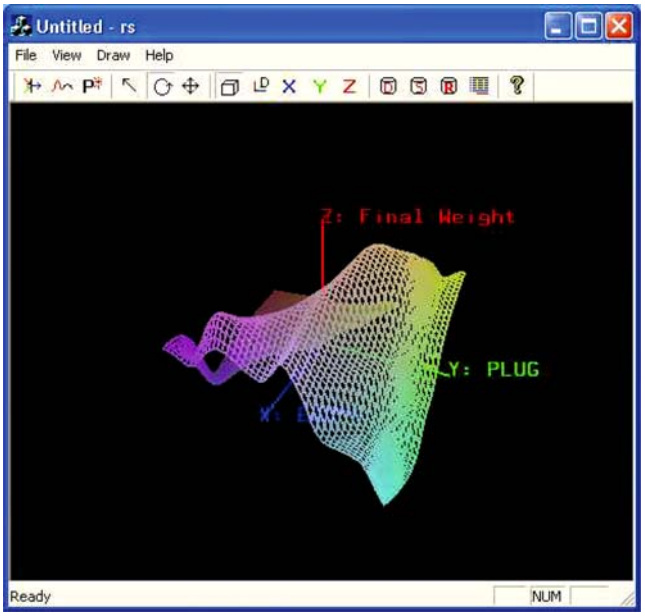

Fig. 4 Software tool - RS

The parameter selection is easy. The user clicks the "select parameters" button in the toolbar. A dialog box pops up, as shown in Fig. 5. In it, the left-hand side contains three list view controls, which are used to select two input parameters and the only output parameter, final weight, respectively. The right-hand side contains values for all

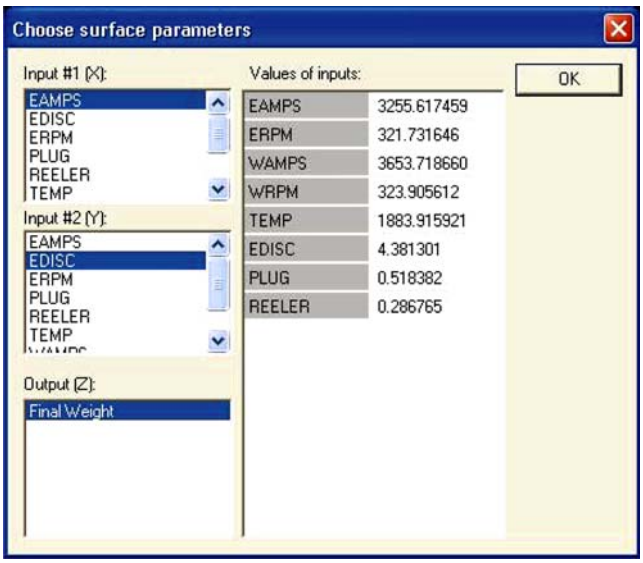

Fig. 5 Select two parameters

input parameters. The two of them that has been selected in list view controls will be ignored. The user can edit the text box for each input item. After this, the tool will generate response surface data for the two input parameters selected and with the values of other input parameters entered by the user.

Yield prediction operation is also simple. The user clicks the "Predict" button. A dialog will pop up as shown in Fig. 6. The left-hand side list view offers the user the facilities to enter input parameter values, while the right-hand side list view enables the user to control display of the predicted output after the middle "Predict" button is pressed.

5 Concluding discussion

This paper presents a systematic approach of data driven process modeling, which has been successfully applied to prediction of yield for a seamless tubing fabrication process. In this approach, the processing data are collected at the manufacturing site. Statistical analysis and pre-processing is carried out, and then the transformed data set is fed into neural networks to build a model for predicting the yield of the seamless tubing process.

During the data collection procedure, two major issues that are common in computational intelligence applications are identified, i.e., missing attributes and data errors in the collected data. It is important to evaluate the data quality before applying them to model building. Two phases of data collection seem costly but at times necessary to get useful and practical data.

Dimensionality of the model is also an important issue in the procedure of data driven process modeling. The technique for dimensionality reduction not only decreases the model complexity and computation cost, thus improving the model operating performance, but also improves the accuracy, generality and robustness of the model by removing irrelevant information.

The seamless tubing process case also indicates some potential future work of the research related in this paper. First, although the issues of missing and poor quality data are pointed out; no cost-effective solution is provided. When the situation becomes more serious than the current case study, the operations of missing attributes treatment and data cleansing are definitely required. Secondly, two parameters are highlighted among the original eight after dimensionality reduction. The dimensionality-reduced model could have higher accuracy. It can be concluded that more attention should be put one the two parameters: EAMPS and Plug. However, it will be more important to understand reasons behind the selection of the two parameters, and to determine if other important parameters must be identified and data for them collected and incorporated in model development. Process improvement suggestions should be provided based on further study. Finally, although the current prediction model is stable and fixed, one cannot assume that it will run perfectly forever when manufacturing situations change. New parameters might be found and the importance ranking of parameters might vary. The current neural network is not suitable for the requirements of adaptability and real-time response to new environments. Other computational intelligence methods such as adaptive neuro-fuzzy technique would be appropriate to enable adaptability and real-time response and thus should be introduced.

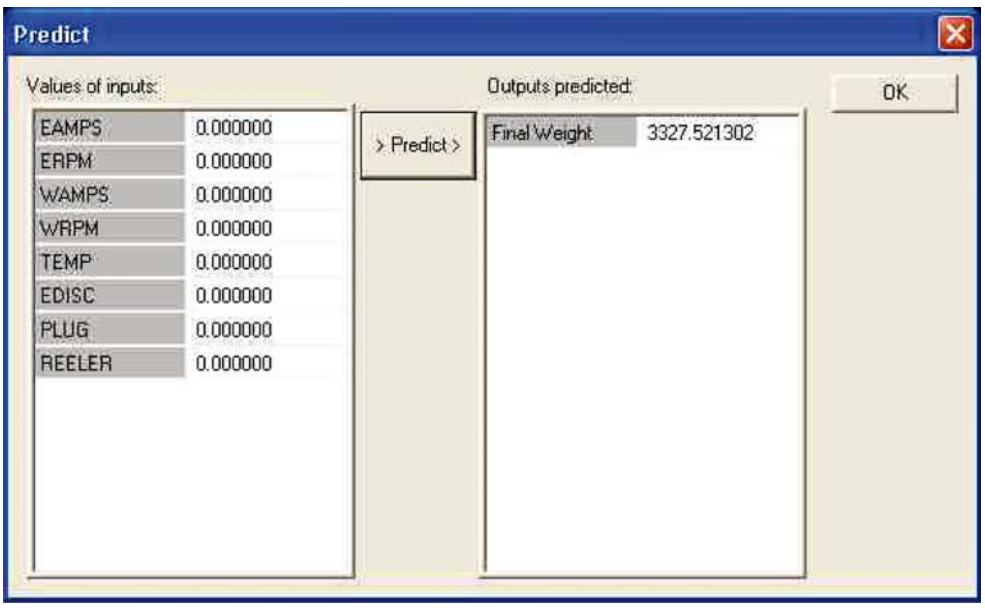

Fig. 6 Predict yield

References

1. USS Tubular products, http://www.usstubular.com.
2. Mohinder LN (1992) Piping handbook. McGraw-Hill, New York
3. Pham DT, Pham PTN (1999) Artificial intelligence in engineering. Int J Mach Tools Manuf 39(6):937–949
4. Huang SH, Xing H (2002) Extract intelligible and concise fuzzy rules from neural networks. Fuzzy Sets Syst 132(2):233–243
5. Xing H, Huang SH, Shi J Rapid development of knowledge-based system via integrated knowledge acquisition, accepted by *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*
6. Huang SH, Zhang HC (1994) Artificial neural networks in manufacturing: concepts, applications, and perspectives. IEEE Trans Compon Packag Manuf Technol Manuf Part A 17(2):212–228
7. Fu LM (1994) Rule generation from neural networks. IEEE Trans Syst Man Cybern 28(8):1114–1124
8. Setiono R (1996) Extracting rules from pruned neural networks for breast cancer diagnosis. Artif Intell Med 8:37–51
9. Setiono R (1997) Extracting rules from neural networks by pruning and hidden-unit splitting. Neural Comput 9(1):205–225
10. Liu C (1997) ML estimation of the multivariate *t* distribution and the EM algorithm. J Multivar Anal 63(2):296–312
11. Patrician PA (2002) Multiple imputation for missing data. Res Nurs Health 25(1):76–84
12. Dash M, Liu H (1997) Feature selection for classification. Intelligent Data Analysis 131–156
13. Liu H, Setiono R (1998) Some issues on scalable feature selection. Expert Syst Appl 15:333–339
14. Liu H, Setiono R (1996) Dimensionality reduction via discretization. Knowl-Based Syst 9:67–72
15. Mirkin B (1999) Concept learning and feature selection based on square error clustering. Mach Learn 35:25–39
16. Jolliffe IT (1986) Principal component analysis. Springer, Berlin Heidelberg New York
17. Chiu SL (1995) Extracting fuzzy rules for pattern classification by cluster estimation. Proceedings of the Sixth International Fuzzy Systems Association World Congress, pp. 1–4