Proceedings of
The 1996 ASME Design Engineering Technical Conferences and
Computers in Engineering Conference
August 18-22, 1996, Irvine, California

96-DETC/DTM-1520

IMPROVING THE DESIGN PROCESS BY PREDICTING DOWNSTREAM VALUES OF DESIGN ATTRIBUTES

Simon Szykman
Engineering Design Laboratory
Manufacturing Systems Integration Division
National Institute of Standards and Technology
Gaithersburg, Maryland 20899

ABSTRACT

This paper presents a computational approach to developing design space models that are utilized to improve the design process by predicting values of downstream design attributes based on information available at early stages, such as preliminary design specifications. The predictive models are similar in function, though not in form, to the internal (mental) models created by experienced designers; however, the advantages of these models are that it may be possible to construct them in the absence of a designer's internal models, and that they can be passed on to and used by less experienced designers. Once created, the computational models aid designers in exploration of design alternatives and to reduce design costs and product development time.

1 INTRODUCTION

Experienced designers are often able to guide the design process toward promising designs at preliminary stages by predicting downstream design attributes using information available early on in the process. A designer may be able to provide rough estimates of the expected cost, performance, or efficiency of an artifact for a given set of specifications without having to design and test the actual artifact. These estimates are used to improve exploration by allowing the designer to focus on promising regions of the design space while avoiding less favorable ones.

This foresight, which is often referred to in the context of vague terms such as "intuition", plays a significant role in exploration of the design space. One of the mechanisms that enables designers to look ahead in this manner is the building of internal (mental) models that map early design specifications to downstream design attributes. While training in a particular domain of design is a prerequisite for this capacity, it is usually not sufficient. In general, that training must be accompanied by extensive experience designing many similar or related artifacts in the domain of interest. What the designer gains from experience is the ability to *generalize* information obtained through design of many earlier artifacts.

There are a variety of barriers that deter the use of these models in design practice. One common obstacle is a lack of expertise among designers who do not have adequate past experience from which to build these models. Because the models are represented internally, rather than explicitly, it is difficult for them to be articulated by expert designers and passed on to less experienced ones. A second difficulty is that there are problems for which mappings from specifications to design attributes may exist, but where those relationships are too complex or multidimensional for a designer to be able to capture. Another barrier is that many designed artifacts span across multiple engineering domains; in these cases, building such a model may require more knowledge than a single designer has. Lastly, although the use of concurrent engineering techniques is becoming more widespread, it is still common in industry practice for designs to be "thrown over the wall." Thus, while information about downstream attributes may exist, the lack of feedback of this knowledge to the designers makes it difficult for them to build these models.

The goal of this research is to improve the overall design process by facilitating design space exploration using a novel type of computational tool that represents and constructs predictive models. The purpose of these computational predictive models is not to supplant any part of the design process, but rather to provide designers with knowledge that may not otherwise be available to them, and which can serve to aid them in guiding search and improving designs.

The approach taken in this work is to use artificial neural networks (ANNs) to build the computational predictive models[1]. An artificial neural network is able to represent a potentially complex functional relationship by providing a numerical output for a given set of inputs. The ANN can therefore be thought of as a response surface – a representation of a function – which takes values for a variety of preliminary or upstream design attributes as input, and produces as output a predicted value for a downstream attribute of interest. Just as a designer needs experience to build an internal model, experience in the form of knowledge about classes of related designs is required to build the computational models. Thus, as will be discussed in greater detail, the approach presented in this paper lends itself to certain types of problems, such as routine or variant design tasks, where such knowledge is available through legacy information, design histories, or can be generated through simulation or experimentation.

Once created, these models can be used much as a designer would use an internal model – to provide rapid estimates early on and use this information to help in searching through the design space. An advantage of the computational models, however, is that it may be possible to construct them in the absence of a designer's internal models which, as described above, can occur for a variety of reasons. In addition, unlike the internal models, the computational models are in a form in which they can be passed on to and used by less experienced designers.

Early knowledge about downstream design attributes has a variety of uses, regardless of whether the source of that information be a designer's internal model or a computational model. Potential benefits of this knowledge include decreasing the number of design iterations, avoiding wasted resources that occur by exploring paths that lead to infeasible designs, and reducing the need for time-consuming analysis, simulation or physical prototyping that might otherwise be needed to obtain values for downstream attributes. Because a large percentage of lifecycle costs are committed at preliminary design stages, improving the designer's ability to explore the design space early in the process can translate to significant reductions in both design costs and product development time.

The following section describes various areas of related work that pertain to this research. In Section 3 the technical approach to building the predictive models is presented; the design of a motor shaft is used as an example to illustrate these models. Section 4 contains a discussion of several issues regarding the predictive models, and Section 5 presents conclusions and addresses areas for future research.

[1] While ANNs have been selected for the predictive models in this work, they are by no means the only available choice for a representation. Alternative representations and the motivation for using ANNs are discussed later in the paper.

2 RELATED WORK

There have been a number of applications of artificial neural networks to aid in the optimization portion of the design process. Liu and Gan (1991) use ANNs as components in an expert system for design of space grid structures; Rogers and Lamarsh (1992) describe an approach to reducing computational costs for optimization of structures by replacing a finite element analysis with an ANN; Berke et al. (1993) use ANNs to optimize structural components for aerospace applications. In each of these applications, artificial neural networks are used in conjunction with or as a substitute for computationally expensive structural analyses[2]. The approach in this paper is intended to predict attribute values resulting from a design process that may consist of several stages in addition to an analysis step. Mukherjee and Deshpande (1995) present an approach to initial design process modeling using ANNs as an alternative to rule-based expert systems for structural design.

It should be noted that artificial neural networks are not a unique representation for a response surface and that alternative representations exist. Other possible approaches include Bayesian surrogate models (e.g. Osio and Amon, 1994; Yesilyurt et al., 1996), nonlinear regression and discriminant models (e.g. Bates and Watts, 1988; Borowiak, 1989) and, when applicable, the more traditional polynomial functional representations (e.g. Box and Draper, 1987; Myers, 1995). An in-depth comparison between the statistical nonlinear regression and discriminant models and ANNs is given in (Sarle, 1994). Artificial neural networks and polynomial approximations are compared in (Carpenter and Barthelemy, 1993).

There is, in addition, some related work that does not use ANNs for representation of a design space. Chen et al. (1995) propose a response surface-based approach to robust design which addresses several drawbacks to Taguchi methods for robust design. Desa and Schmitz (1991) propose an iterative design procedure called virtual concurrent engineering which uses a knowledge-based expert system to predict downstream producibility characteristics for part designs.

3 CONSTRUCTION OF PREDICTIVE MODELS

This section presents the approach to building predictive models using artificial neural networks. An overview of ANNs is given in Section 3.1. Section 3.2 illustrates the use of the predictive models with an example (the design of a motor shaft) and presents experimental results illustrating the performance of a variety of predictive models that were constructed for this problem.

3.1 Artificial Neural Networks

There are many different types of artificial neural networks. This section provides a summary of one kind of ANN, called a backpropagation network, which is utilized in this research. Additional information can be found in

[2] In (Liu and Gan, 1991) separate ANN modules are also used to evaluate results of analysis and to control an optimization.

introductory texts on the subject such as (Rumelhart and McClelland, 1986) and (Hertz et al., 1991).

Given a set of data where each data point consists of one or more input values and one or more output values, an ANN can be thought of as a response surface approximation to the data. Figure 1 shows an illustration of an artificial neural network. The network consists of three layers – an *input layer* with two input units, a *hidden layer* with six hidden units and an *output layer* with one output unit. The hidden units are called "hidden" because values for those units are not specified in the data, whereas values for the input and output units are.

The lines between units in the three layers are called *weights*, and represent weighted connections from units in one layer to those in the next. Each of the units in the ANN produces a numerical output for a given input. The ANN calculates a final output for a set of inputs to the network by propagating the inputs through the network according to the weight of the connections to produce a corresponding output (these calculations are described in greater detail below). The task of building the approximation to the data, also referred to as *training* the network, consists of finding a set of weights that minimizes some measure of the error between the outputs specified in the data set and those produced by the network. Once an ANN has been trained using a data set, it can then be used to calculate outputs for inputs that were not part of the data set. As with any function approximation, accuracy is generally higher for interpolation than for extrapolation.

The numerical input to units in the ANN is referred to as the *activation*. The activation for a unit in the input layer is supplied through the data which the network is trying to represent. The activation of each unit in the hidden and output layers is equal to the sum of the outputs from units in the previous layer multiplied by the weights from the preceding layer to the current unit, plus a *bias* term. The bias term is a constant added to (non-input layer) units and may be different for each unit. The bias is implemented by having a unit that is invisible to the user, with an output of 1.0 and with varying weights for connections to all hidden and output units. This allows the bias for each unit to be calculated by changing the weights using the same algorithm used for the rest of the network. To illustrate, the activation, A_{hn}, for the *n*th hidden unit of the example network (shown in detailed view in Figure 2) is calculated as:

$$A_{hn} = O_{i1}W_{i1,hn} + O_{i2}W_{i2,hn} + (1.0)W_{b,hn} \qquad (1)$$

where O_{i1} and O_{i2} are the outputs of input units 1 and 2, $W_{i1,hn}$ and $W_{i2,hn}$ are the weights between the *n*th hidden unit and input units 1 and 2, respectively, and $W_{b,hn}$ is the weight from the bias unit to that hidden unit.

The output for the units in the input layer is equal to their activation, i.e., the values supplied by the data. The outputs of a unit in the hidden or output layer is a function of the unit's activation. There are a variety of output functions that can be used. In this research, the output function used (shown in Figure 3) is:

$$O(A) = \begin{cases} 0.0 \text{ if } A < -n, \\ \dfrac{A+n}{2n} \quad \text{if } -n \le A \le n \\ 1.0 \text{ if } A > n. \end{cases} \qquad (2)$$

where *A* is the activation of the unit (calculated according to equation (1)) and *n* is the number of weights from the previous layer to the unit (*n* = 2 for the hidden units in Figure 1).

Before training the networks, the data is normalized so that all inputs and outputs to the network fall between 0.0 and 1.0. This normalization has two advantages. First, because no unit has an output greater than 1.0, the activation for a unit in the hidden or output layers is never greater than *n*, allowing the same output function to be used for all (non-input layer) units rather than using individual output functions for different units or layers. Second, the normalization improves convergence because it tends to avoid network weights that vary greatly in order of magnitude due to inputs or outputs that differ by large amounts. Using matrix representations for the ANNs and the normalization scheme described above, calculations can be done efficiently and rapidly.

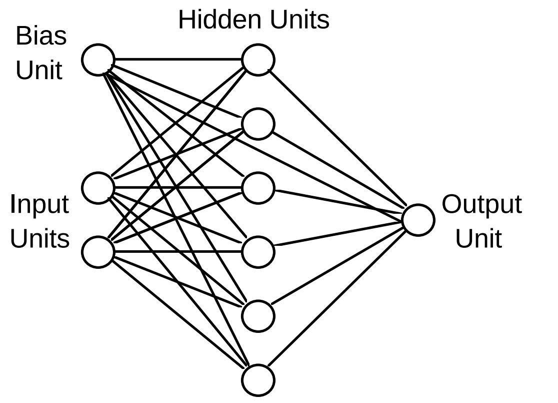

Figure 1. Example ANN Configuration

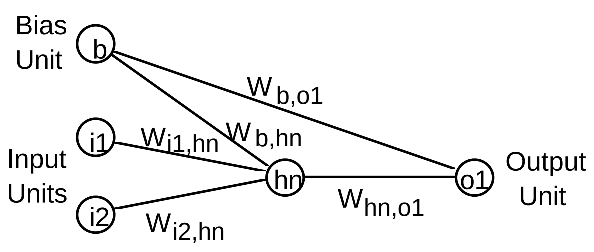

Figure 2. Detailed View of *n*th Hidden Unit

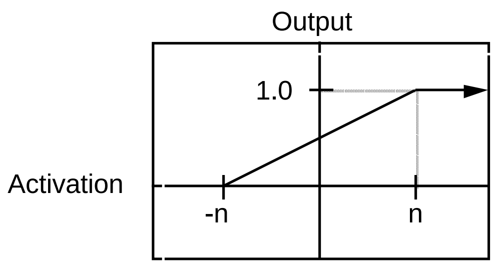

Figure 3. Output as a Function of Activation

The training of the ANN is done by initializing all weights to random values between zero and one. Then, three steps are performed: (1) every set of inputs in the data set is run through the network to obtain a set of outputs, (2) the error between the predicted outputs and the true outputs is calculated, and (3) the weights are modified. One iteration of this three-step procedure is called an *epoch*. The network is trained over many epochs by iteratively repeating the procedure, each epoch resulting in modifications to weights and ultimately converging to a set of weights that minimizes the error between the outputs in the data set and the ones produced by the network. Many schemes have been proposed to perform the optimization of weights, from gradient approaches to simulated annealing to genetic algorithms. The algorithm used in this paper is called quickprop (Fahlman, 1988), which is a gradient-based approach that makes use of information about the second derivative of the error to speed up the optimization.

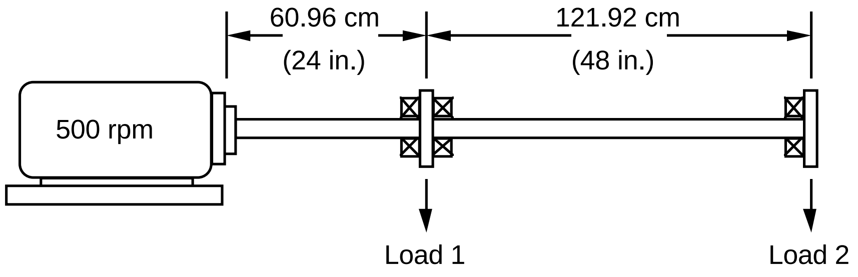

Figure 4. Shaft Design Problem

3.2 Example: Design of a Motor Shaft

3.2.1 Problem Description. A design problem was selected for which data that would be used to build the predictive models could be generated by a simulation of the design process. The task is to design a shaft which is connected to a motor and drives two belts, resulting in torsional loads on the shaft specified as power drawn from the shaft. This arrangement is shown in Figure 4.

In this scenario, the motor drive is part of several similar larger machines which are being designed for a variety of related applications. It is known that in all of the applications being considered both loads will lie somewhere between 18.64 kW (25 hp) and 93.21 kW (125 hp). However, because the designs of the machines are at a preliminary stage, the values for the loads are among the design specifications that have not yet been fixed. Despite this, the designer is interested in doing some initial exploration of alternatives. By predicting downstream design attributes (in this case shaft material, weight, stress and angle of twist) the designer can attempt to anticipate needs for raw materials, to estimate costs for this portion of the design, and possibly use that information to provide feedback that will help find good values for the undetermined load specifications at this early point in the design of the larger machine.

The design process for the shaft involves three stages: selection of the shaft material, analysis and optimization of the inner and outer radii to minimize the shaft weight. The design constraints and specifications are as follows (relevant equations are given in the appendix):

- the locations of the loads and the length of the shaft are fixed as shown in Figure 4,
- the shaft is supported by bearings, so that there are no bending loads,
- the motor runs at 600 rpm and can deliver the required power P = load 1 + load 2,
- the ratio of yield stress to shear stress must exceed a factor of safety of 1.5,
- the total angle of twist in the shaft, ϕ, may not exceed 5°,
- the outer radius of the shaft has a maximum of 2.54 cm (1.0 in.),
- the wall thickness of the shaft (outer radius - inner radius) must be at least 0.63 cm (0.25 in.).

There are three materials available to choose from: Aluminum (2014-T6), structural steel (ASTM-A36) or a high-strength steel (ASTM-A242). In this simulated design process, material selection is done using a simple rule base. Aluminum is preferred over steel due to its light weight. However, Aluminum has a much lower shear modulus which results in larger angles of twist. Accordingly, steel is chosen over Aluminum if the angle of twist is too high for a given set of loads. Because of its higher cost, the high-strength steel is only used if a loading is such that the shear stresses are unacceptably high using structural steel. After the material has been selected using the rule base, the shaft is optimized for weight, subject to the problem constraints described above, using simulated annealing, a stochastic optimization technique (Kirkpatrick et al., 1983) (the choice of simulated annealing for the optimization is discussed in Section 4).

3.2.2 Experimental Results. In order to construct the predictive models, data were generated for the three-stage design process described above. Since the range for both loads is 18.64 kW - 93.21 kW (25 hp - 125 hp), these were the upper and lower bounds for the data. Points were sampled from 18.64 kW - 93.21 kW (25 hp - 125 hp) at 7.46 kW (10 hp) intervals in both variables, for a total of 121 data points; these are referred to as the *model data* since they are used to create the model. For a predictive model to be useful, it must not only be able to predict the data that were used to construct it, but it should also be able to generalize by making predictions for points that were not used to create it. A second set of data to be used for validation of the model was created by sampling points from 22.37 kW - 89.48 kW (30 hp - 120 hp) at 7.46 kW (10 hp) intervals in both variables, for a total of 100 data points; these are referred to as the *test data*. The grid of sampled points is shown in Figure 5, with the model data set indicated by squares and the test data set indicated by diamonds.

For each of final designs resulting from the design process, the material, stress, angle of twist and optimized weight were recorded. Because of the optimization for weight, for each of the designs either the stress or the angle of twist is near or at its maximum allowable value. This "critical constraint" was also recorded for each of the designs. This gave a total of five design attributes and ten data sets (five model data sets and five test data sets).

An artificial neural network was created for each of the five design attributes. Each of the ANNs had two input units and ten hidden units. This number of hidden units was obtained empirically by creating several ANNs with varying numbers of hidden units and examining errors in predictions for each of them. The ANN which predicted material had three output units, one for each material (an output of 1,0,0 indicated Aluminum; 0,1,0 indicated structural steel; 0,0,1 indicated high-strength steel). The network which predicted the critical constraint had one output unit which had a value of 0 if the stress constraint was the critical constraint and 1 if the angle of

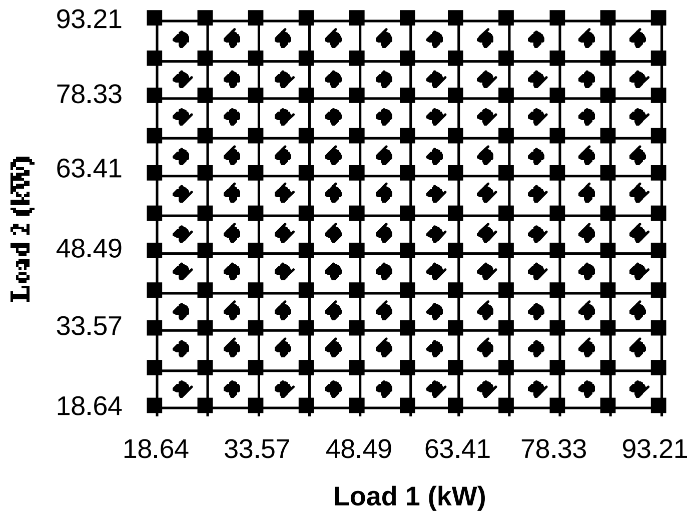

Figure 5. Sampled Points for Shaft Design Problem (Squares are Model Data, Diamonds are Test Data)

twist constraint was the critical constraint. The output units for both of these networks had output functions that forced a binary 0-1 output. The other three networks had one output unit which used the output function shown in Figure 2 and produced a (normalized) numerical output.

The ANNs were trained using the model data sets for 8000 epochs. Training took on average less than a minute per thousand epochs on a Sun SparcStation 2[3]. After the predictive models were created, they were tested on both the model data sets and the test data sets. Good performance on the model data demonstrates that the model was able to approximate the data. However, good performance on the model data does not necessarily prove that the model is a good one because it is possible that the data does not adequately represent the design space. As mentioned earlier, the test data sets are used to validate the models. Since the test data were not used to create the models, good performance on the test data implies that the models characterize the design space and are able to make predictions that generalize beyond the data used to create them.

Table 1 summarizes the results of testing the predictive models giving average percentage error on both sets of data for each of the five models. The observation that errors on the test data sets did not differ greatly from the model data sets indicates that the models were able to generalize well enough to make predictions for points that were not used to construct them. Had the models only characterized the model data sets without abstracting out more general features of the design space, predictions for the test data sets would have been much worse. The average percentage errors range from 4% to less than 6%; this accuracy is quite good considering that the range

[3] This information is given in order to provide an indication of computational requirements and does not constitute an endorsement of any hardware by the National Institute of Standards and Technology.

of loads over which the models make predictions varies by an order of magnitude, and given that the data included several shaft materials and constraints. Furthermore, perfect accuracy would not be expected since the data in both sets contains noise due to the use of simulated annealing which often converges to near-optimal rather than optimal solutions (simulated annealing was used for optimization to intentionally introduce variation in the data in order to test the ability of the ANNs to tolerate noise).

Interestingly, the models for material and critical constraint were able to make predictions with 100% accuracy. This indicates that the design space for these attributes was well characterized by the model over the given range of input loads. Thus, the designer can use the predictive model for material with high confidence and quickly get predictions for the best material to use based on shaft loads without performing an analysis. The other models, which have some error, are still useful since even rough estimates of downstream design attributes can be utilized to guide the designer at preliminary stages of the design process.

The models can also be used to determine which regions of the design space are problematic for predictions. A subsequent experiment showed that using the same predictive model for weight, average percentage errors on the test data dropped to 4.6% if predictions were made for points where neither load was greater than 74.57 kW (100 hp). This may indicate that the high-load region of the design space is more complex and either additional data or a larger ANN is required to adequately characterize it.

4 DISCUSSION

It is perhaps intuitively obvious that a completely accurate predictive model is usually, if not always, impossible to obtain. If one likens the generation of these computational models to fitting a function or response surface to a set of data, there are a number of reasons that the model may not be perfect. One potential reason is an inadequate functional representation for the target function, such as using a quadratic function when the target function is a higher order polynomial or a discontinuous function. A second problem may be an inadequate technique for constructing the model, that is, creating the model to fit the given data. With some functional representations (least squares, for example) it is relatively simple to create a model that best fits a given data set; with other representations (including ANNs) finding the best fit for a set of data is not as straightforward and multiple approaches may be taken.

A variety of additional difficulties can arise from the data themselves. These include insufficient data, statistical variation in the data, and random variation (or noise) in the data. If there are insufficient data, it may not be possible to adequately capture the characteristics of the design space even if the functional form and the construction technique are suitable for the problem. If there is statistical variation in the data, the true function value f(x) for an input vector x is not known, possibly introducing errors in predicted values. Random variation can cause similar problems but because it cannot be recognized as readily and is less predictable than statistical variation, it is more difficult to deal with. Furthermore, if a model is created to correctly fit noisy data, it is likely that it will not predict true (non-noisy) values as accurately.

Many of the potential difficulties described above are typical of data that occurs in real engineering problems. Engineering data often cannot be approximated using simple low-order polynomial functions. Discontinuities in design spaces are common, due to various types of constraints that dictate what is and is not permissible. Engineering data generally contain statistical and random variation, occurring both in the artifact and in the measurement of design or performance attributes of interest. While these problems do pose barriers to obtaining models with perfect accuracy, it is not necessary to have errorless predictions in order to benefit from the use of these models.

Table 1. Average Percentage Error in Model Predictions for Model and Test Data Sets

	Material	Stress	Angle of Twist	Weight	Critical Constraint
Model data	0.0	4.6%	4.0%	5.8%	0.0
Test data	0.0	5.1%	3.6%	5.4%	0.0

As noted previously, artificial neural networks are not a unique representation for this type of predictive model. ANNs were chosen as a representation in this research because of their ability to deal with the characteristics described above. In particular, ANNs do not presuppose any particular functional form, they can represent relationships of high dimensionality, they can approximate discontinuities, and they are tolerant of statistical or random variation in data, tending to exhibit an averaging effect that smoothes out local variations in a response surface.

The shaft design problem exhibited several of these properties: noise was introduced in the data by the simulated annealing optimization and constraints that caused changes in materials led to discontinuities in the design space. Despite these obstacles, the ANNs were able to represent models that predicted downstream attributes including weight, stress, angle of twist, and material to within a few percent accuracy without assumptions about the functional form for any equations.

The appendix contains equations used to calculate torque as a function of an input load, and maximum stress and angle of twist a function of torque. However, the response surface represented by the predictive models is far more useful than approximations to these functions would be because additional problem-specific knowledge is inherent in the predictive models that is absent from those equations. For instance, the predictive models account for the constraints on maximum stress and angle of twist so that a prediction for a small total load corresponds to the weight/stress/twist for an Aluminum shaft with one set of material properties, whereas a prediction for a large total load corresponds to a weight/stress/twist of a steel shaft with different material properties. The predictive model for angle of twist captures the fact that the design space is not symmetric with respect to loads; that is, the angle of twist for load 1 = X and load 2 = Y is not the same as the angle of twist for load 1 = Y and load 2 = X if X and Y are different (see equation (6) in the appendix). Also inherent in the model is additional information regarding the factor of safety and constraints on shaft geometry. Furthermore, the models are not simply predicting the results of an analysis, but of a three-stage design process consisting of material selection, analysis and optimization.

More importantly, the problem-specific information is incorporated in the predictive model *implicitly*, rather than explicitly. No prior knowledge about the form of the equations used was necessary, nor was information about the types of constraints or the variables they involve required in order to capture their effects on the design problem. This is of critical importance since applications of prime interest for the predictive models are to use data from previously designed artifacts to build models that can aid engineers in designing subsequent related artifacts. While data regarding downstream design and performance attributes may exist in the form of legacy information or design histories, in many of these cases complete knowledge about the constraints that impacted the design may not be available. Unlike the problem used as the example in this paper, in many engineering applications closed-form equations are not available.

As problems become more complex and the number of design specifications increases, the approach is the same as that used for the shaft design example. Say, for instance, that the designer wished to explore designs for shafts made from a variety of different materials to determine how they affect final designs. Whereas the material was a predicted downstream attribute in the shaft design scenario above, in this case the material would be considered an upstream attribute or specification. First, data would be obtained for a variety of different material properties. Next, just as the models in Section 3 were constructed to make predictions using values for the two loads as model inputs, in this case predictive models would predict downstream attributes also using material properties as additional inputs. This new model could be used to help make ensuing decisions regarding initial material selection.

In the shaft design example, the attributes that were predicted (weight, stress, angle of twist, material and critical constraint) were obtained through a simulation of the design process that consisted of material selection, analysis and optimization. However, models can be created to predict other attributes (e.g. design feasibility for a given set of specifications, downstream design costs or other performance characteristics) which result from a more complex design process consisting of a greater number of stages and design decisions.

5 CONCLUSIONS

This paper has presented an approach to developing computational predictive models, a new type of tool for improving the design process by predicting values of downstream design attributes based on information available at early stages in the process. Experienced designers make use of internal (mental) models to help guide the design process. The purpose of this research is to develop a computational method for creating predictive models since there are a number of barriers to the use of internal models in design practice. These models can be used by the designer to aid in exploration of design alternatives and to reduce design costs and product development time by reducing design iterations, avoiding wasted resources from exploring infeasible design paths, and reducing the need for simulation or physical prototyping.

In addition to the uses described above, the predictive models have a variety of other applications. A model can be used to perform parametric studies to determine the effect of changes in design specifications on downstream design attributes. This can help in achieving robust design, in which a designer attempts to find regions of the design space where performance attributes are insensitive to variations in design specifications. Models can be incorporated into design optimization codes not only to predict downstream performance or costs, but to optimize preliminary specifications as well. Returning to the shaft design problem as an example, recall that the design specifications (the two loads) were not fixed at early stages of the design; the designer can use the predictive models to find optimal input loads – where "optimal" is some measure of goodness, such as torsional stiffness to cost ratio – and propose these values as specifications during initial design reviews.

In order to build models that map specifications to downstream attributes, data corresponding to groups of related designs is required. Thus, the proposed methodology lends itself to problems where classes of similar artifacts have been previously designed (such as routine or variant design tasks), or problems where such data can be generated through simulation or experimentation. Very few artifacts that are designed begin from a clean sheet of paper; most are variations on or improvements to existing designs where information about earlier artifacts is often available in one form or another. Consequently, while the need for data does limit the applicability of the predictive models, it does not detract from their potential utility in many industry applications.

One important question that must be addressed is how much data are needed to build useful models? It is expected that the amount of data required will increase as the number of inputs to a model increases (analogous to introducing more variables in a design problem) and as the complexity of the design task increases (analogous to having a more complex function to approximate). Obtaining a more quantitative measure of data needs is an important area of future investigation.

A related issue is the level of accuracy that can be achieved with the predictive models. Because they are only approximations to a complex design space, as well as for reasons presented in the discussion section, perfect accuracy is unlikely. Although initial results presented in this paper demonstrate good accuracy, expected accuracy for larger problems remains to be obtained empirically through further applications of this methodology. Since the predictive models are intended to be used at early stages of the design process, perfect accuracy is not necessary; even approximate estimates of downstream effects of upstream design decisions can be extremely helpful in guiding the search through a design space in the same manner in which experienced designers are able to make use of rough estimates generated by their internal models.

When only limited data are available, it is desirable to determine whether or not a useful model can be constructed. In instances where it is possible to generate data through simulation or experimentation, obtaining this data may incur significant costs. Knowing how much data are required can give an indication of the cost associated with building predictive models. In these cases, costs can be reduced by generating a greater amount of data for the more critical parameters and less data for parameters that have a smaller effect on downstream attributes. Design-of-experiments techniques may be a useful tool in determining the relative importance of the various parameters as well as for selection of points to sample for data.

Current efforts are directed toward developing a prototype tool that can be applied to an industry design problem. One advantage of this approach is that problem-specific information is modeled implicitly; no *a priori* knowledge about types of design constraints, the nature or functional form of the design space, or the domain of design, is required. However, it may be possible to use these predictive models more effectively by using them in conjunction with approaches that include such knowledge. The ability to make use of problem-specific knowledge when it is available is an area long term future work. Other long term research issues include determination of the scope of applicability of the predictive models and characterization of the classes of design problems for which they can and cannot be constructed.

ACKNOWLEDGMENTS

This research was conducted at the National Institute of Standards and Technology under the National Research Council Postdoctoral Research Associateship program.

REFERENCES

Bates, D. M. and D. G. Watts (1988), *Nonlinear Regression Analysis and Its Applications*, John Wiley & Sons, New York.

Berke, L., S. N. Patnaik and P. L. N. Murthy (1993), "Optimum Design of Aerospace Structural Components Using Neural Networks," *Computers and Structures*, **48**(6):1001-1010.

Borowiak, D. S. (1989), *Model Discrimination for Nonlinear Regression Models*, Marcel-Dekker, New York.

Box, G. E. P. and N. R. Draper (1987), *Empirical Model-Building and Response Surfaces*, John Wiley & Sons, New York.

Carpenter, W. C. and J.-F. M Barthelemy (1993), "A Comparison of Polynomial Approximations and Artificial Neural Networks," *Structural Optimization*, **5**:166-174.

Chen, W., K.-L. Tsui, J. K. Allen and F. Mistree (1995), "Integration of the Response Surface Methodology with the Compromise Decision Support Problem in Developing a General Robust Design Procedure," *Advances in Design Automation 1995 – Proceedings of the 1995 Design Engineering Technical Conferences Volume 1*, DE-Vol. 82, September 17-20, Boston, MA, pp. 485-492.

Desa, S. and J. M. Schmitz (1991), "The Development and Implementation of a Comprehensive Concurrent Engineering Method: Theory and Application," *SAE Technical Paper Series*, Paper 912210.

Fahlman, S. E. (1988), "Faster-Learning Variations on Back-Propagation," *Proceedings of the 1988 Connectionist Models Summer School*, Morgan Kaufmann.

Hertz, J., A. Krogh and R. G. Palmer (1991), *Introduction to the Theory of Neural Computation*, Addison-Wesley, Redwood City, CA.

Kirkpatrick, S., C. D. Gelatt, Jr., and M. P. Vecchi (1983), "Optimization by Simulated Annealing," *Science*, **220**(4598):671-679.

Liu, X. and M. Gan (1991), "A Preliminary Structural Design Expert System (SPRED-1) Based on Neural Networks," *Artificial Intelligence in Design '91*, J. S. Gero, ed., Butterworth-Heinemann, pp. 785-799.

Mukherjee, A. and J. M. Deshpande (1995), "Modeling Initial Design Process Using Artificial Neural Networks), *Journal of Computing in Civil Engineering*, **9**(3):194-200.

Myers, R. H. (1995), *Response Surface Methodology: Process and Product in Optimization Using Designed Experiments*, John Wiley & Sons, New York.

Osio, I. G. and C. H. Amon (1994), "An Engineering Design Methodology with Bayesian Surrogates and Optimal Sampling," Technical Report EDRC 24-112-94, Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, PA.

Rogers, J. L. and W. J. Lamarsh II (1992), "Application of a Neural Network to Simulate Analysis in an Optimization Process," *Artificial Intelligence in Design '92*, J. S. Gero, ed., Kluwer Academic Publishers, Boston, pp. 739-754.

Rumelhart, D. E. and J. L. McClelland (1986), *Parallel Distributed Processing Volume 1: Foundations*, MIT Press, Cambridge, MA.

Sarle, W. S. (1994), "Neural Networks and Statistical Models," *Proceedings of the Nineteenth Annual SAS Users Group International Conference*, Cary, NC (SAS Institute), April.

Yesilyurt, S., C. Ghaddar, M. Cruz and A. T. Patera (1996), "Bayesian-Validated Surrogates for Noisy Computer Simulations: Application to Random Media," to appear in *SIAM Journal on Scientific Computing*.

APPENDIX

This appendix summarizes the equations used for analysis in the shaft design problem presented in this paper (see Figure 4). The torque due to each load is:

$$T = \frac{58{,}400 \ P}{2 \pi n} \text{N}-\text{m} \tag{3}$$

where P is the power in kW and n is the angular velocity of the shaft in rpm.

The maximum shear stress in the shaft is:

$$\tau_{max} = \frac{T_{max} \ r_o}{J} \tag{4}$$

where T_{max} is the maximum torque (for this problem the sum of the torques $T_1 + T_2$ due to the two loads), r_o is the outer shaft radius, and J is the polar moment of inertia given by:

$$J = \frac{\pi (r_o^4 - r_i^4)}{2} \tag{5}$$

where r_i is the inner shaft radius.

The angle of twist due to the torque in the shaft is given by:

$$\phi = \frac{T l}{G J} \tag{6}$$

where l is the length of the shaft and G is the shear modulus. Note that for this problem, the angle of twist is the sum of the angle of twist in the first shaft segment (between the motor and load 1), having $T = T_1 + T_2$ and $l = 60.96$ cm (24 in.), and the second shaft segment having $T = T_2$ and $l = 121.92$ cm (48 in.) (see Figure 4).