Initial Design of Gears Using an Artificial Neural Net

Taesik Jeong
SECOM Intelligent Systems Lab, Toyko, Japan

Thomas P. Kicher
Case Western Reserve University, Cleveland, OH

Ronald J. Zab
Joy Technologies Inc., Bedford Gear Division, Solon, OH

Introduction

Many CAD (Computer Aided Design) systems have been developed and implemented to produce a superior quality design and to increase the design productivity in the gear industry. In general, it is true that a major portion of design tasks can be performed by CAD systems currently available. However, they can only address the computational aspects of gear design that typically require decision-making as well. In most industrial gear design practices, the initial design is the critical task that significantly effects the final results. However, the decisions about estimating or changing gear size parameters must be made by a gear design expert.

To move one step forward, two new system developing techniques have been investigated. One is the artificial neural net, and the other is the expert system known as artificial intelligence. The former is well-suited to estimating initial gear size, while the latter is the choice for changing parameters. This article demonstrates the adaptability of an artificial neural net for the initial gear design which is a part of the Intelligence GearCAD system under development, that emulates the entire gear design procedure, including the decision-making tasks.

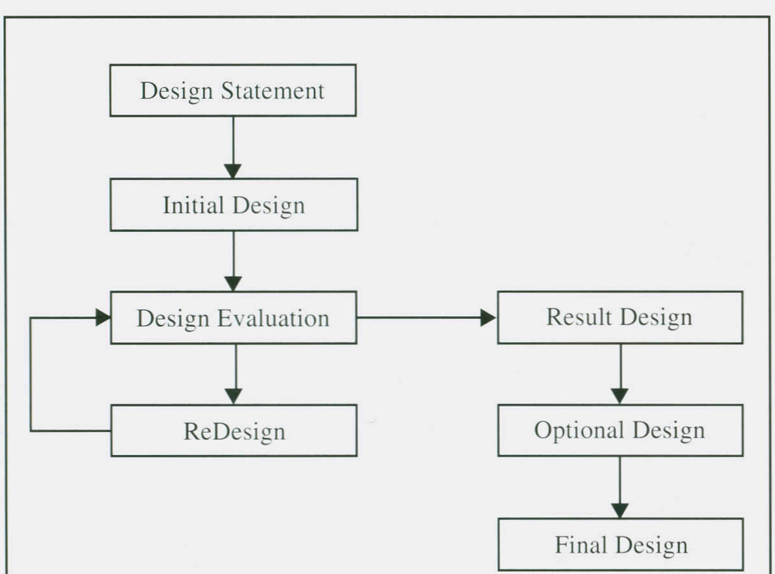

Fig. 1 - Simplified mechanical design stages.

Initial Gear Design

In Fig. 1, a model of the mechanical design procedure is illustrated. Similar models have been used to develop mechanical engineering CAD and expert systems.[10-11] This simplified design model is adaptable to most mechanical element designs including gear design. A specific model representative of gear design which corresponds to Fig. 1 is shown in Fig. 2.

The first stage of designing a gear set is estimating the necessary gear size parameters based on user-specified requirements. Once these parameters are selected, gear and tool geometries will be calculated and evaluated by the AGMA (American Gear Manufacturers Association) power rating standard.[8] If the power rating result is unsatisfactory, the result will be analyzed and the necessary parameters will be changed. The second and the third stages will be repeated in an iterative manner until the AGMA power rating is satisfied. The final stage is designing a gear blank, which is customarily done after a successful power rating is achieved.

In practice, engineers go through the initial design stage only once during the entire design procedure. The number of iterations carried out to complete the gear design depends upon how well the gear size parameters are estimated in the initial design stage. Consequently, an efficient gear design can only be achieved by properly estimating the initial gear size parameters.

The estimated parameters required for the initial design stage consist of the center distance, diametral pitch, pinion teeth number, and gear teeth number, or alternately, the total number of teeth. These four are the essential parameters necessary to carry out the AGMA power rating procedures. Equation 1 illustrates how these four parameters are related to each other while assuming the helix angle is zero.

$$DP = \frac{N_T}{2\ CD} \qquad (1a)$$

$$N_T = N_P + N_G \qquad (1b)$$

where, *DP* Diametral Pitch
CD Center Distance
N_G Gear Teeth Number
N_P Pinion Teeth Number
N_T Total Teeth Number

The determination of one parameter in Expression 1a is dependent on the two other parameters. Therefore, at least two parameters must be estimated by the engineer. There may be many combinations of solutions which satisfy Equation 1 for a single example. Finding a superior solution among a myriad of possibilities depends upon the ability of an engineer. Proper initial parameter estimations usually require years of experience, as well as an organized knowledge of the field. In most cases, the accumulated design data through the history of a company is also an essential factor. This type of design task is known as decision making. Fig. 3 shows the factors involved in a gear engineer's decision making.

Two Steps of Initial Gear Design

The initial gear design stage consists of two steps. First, an engineer refers to a standard product catalog to identify the proper model. The selection is based on the user's specifications, which include horsepower, speed ratio, and input RPM. At this step, the center distance is obtained with the proper selection of model size. Next, the number of pinion and gear teeth will be estimated by a trial and error method. The ratio of estimated number of pinion and gear teeth must not exceed the predetermined percentage of error over the required speed ratio. The diametral pitch can then be calculated using these estimated values. This procedure is only one example of a number of initial gear design methods used in the industry. The method shown here was obtained from an engineer with many years of experience in both designing and manufacturing, actively working in the gear industry.

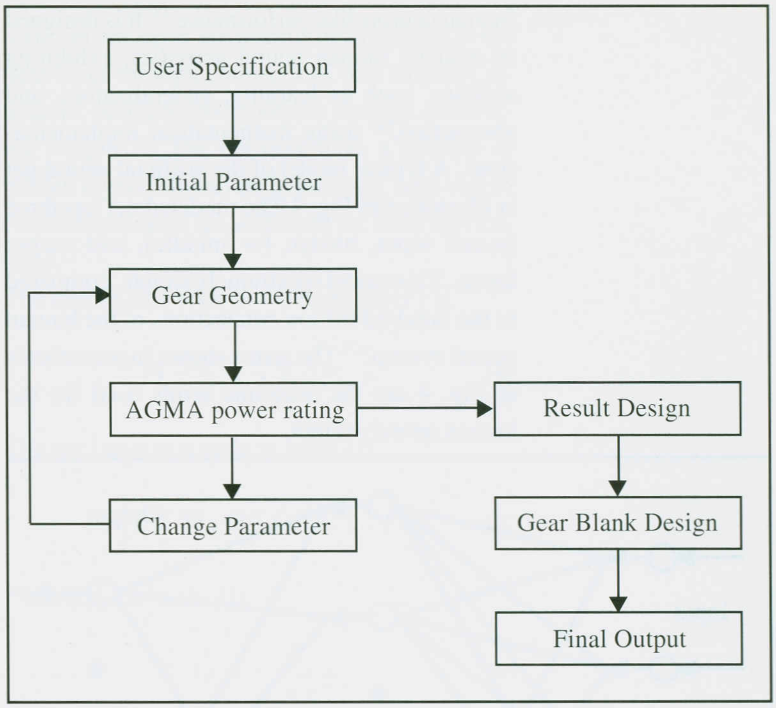

Fig. 2 - Modeled gear design stages.

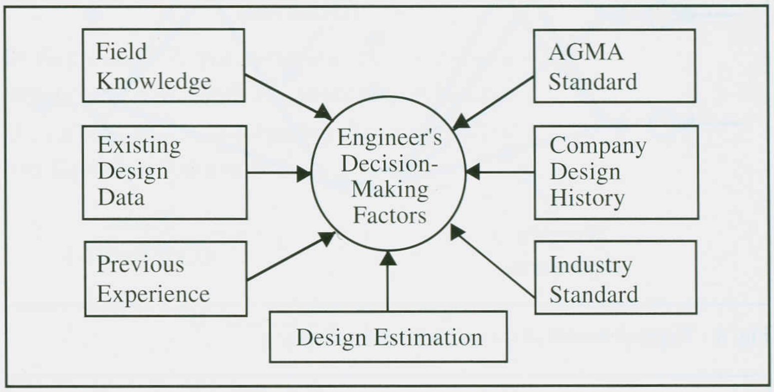

Fig. 3 - Engineer's decision-making factors.

Artificial Neural Net

The artificial neural net is composed of highly interconnected layers which attempt to achieve

Dr. Taesik Joeng
is a reseacher at SECOM Intelligent Systems Lab, Tokyo. His research interests include expert systems and object-oriented programming for engineering design automation.

Dr. T. P. Kicher
is the Armington Professor of Engineering at Case Western Reserve University, Cleveland, OH.

Dr. Ronald J. Zab
is Engineering Manager at Joy Technologies Inc., Bedford Gear Division. He is a member of ASME and a licensed professional engineer with over 15 years' experience in gear design and manufacture.

human neuron-like performance.[3] It is designed to emulate human neural activities, exhibiting abilities, such as learning, generalization, and abstraction,[4] using mathematical implementations. A typical model of the artificial neural net is illustrated in Fig. 4 The modeled net has three layers: input, hidden (or middle), and output layers. This model is extremely simple, compared to the hundred trillion connections of the human neural system.[2] The terms shown in parenthesis in Fig. 4 are the anatomic terms used for the human neural system.

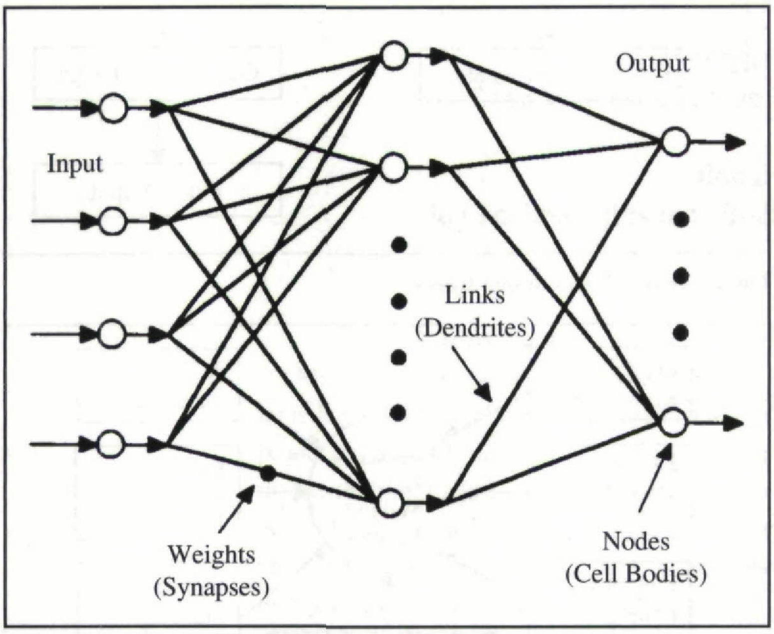

Fig. 4 - Typical model of an artificial neural net.

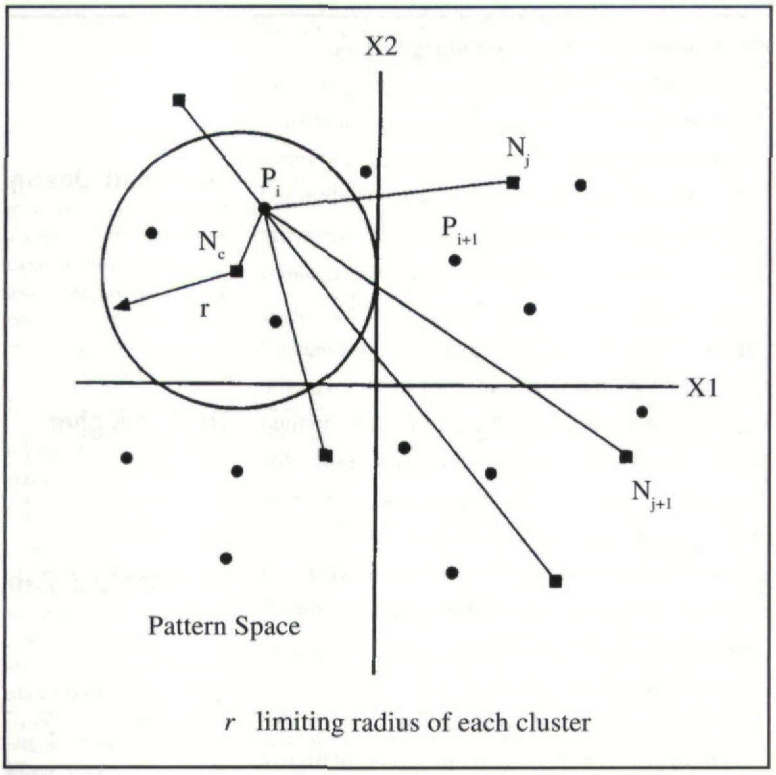

Fig. 5 - Single step of LVQ algorithm.

In Fig. 4, each node in one layer receives multiple signals from the nodes in the previous layer. The strength of each signal is determined by the value of the connecting weight between paired nodes. The signals conveyed to the node are summed and averaged (or mathematically evaluated) to decide whether this node will activate or not. If the node activates, the signal generated will be transmitted to the nodes in the next layer.

The artificial neural net is not functional without existing knowledge, just as a human engineer cannot perform a task without pre-existing knowledge of the field. The net must be trained with known knowledge patterns that consist of input and the corresponding target output. The knowledge patterns are fed through the net so that the connecting weights can be learned and memorized. Once all the connecting weights are established, the net will produce the proper output when the same or similar input pattern is seen. Accordingly, the quality of the knowledge patterns used for training influences the quality of the estimated outputs. The net is said to be successfully trained if the estimated outputs match the target outputs within a certain level of error. Because the training knowledge patterns may not be perfect, there is always the chance that an errant estimation may appear, just as the performance of the human engineer will be inaccurate if incorrect knowledge was used in training.

Artificial Neural Net Algorithms

Many artificial neural net algorithms have been developed and implemented. Although there are some structural variations, the basic idea is equivalent in terms of implementing a human neural system. Each algorithm has its own characteristics and applicable regime. After the nature of initial gear design was investigated, two algorithms, namely LVQ (Learning Vector Quantization) and GDR (Generalized Delta Rule), were selected to emulate two steps of initial gear design.

LVQ is also known as the pattern recognition or classification method, which classifies available knowledge patterns in a pattern space.[5] Each pattern must have its own class label (or class I.D.). LVQ forms clusters, which include identically labeled patterns, while remembering their weight centers. When a new input pattern without a class label, not encountered previously, is seen, LVQ locates the cluster weight center which is closest to the new input pattern and sends the class label of the selected cluster as the output. In other words, LVQ simply tells where the new input pattern belongs.

In Fig. 5, a single step of the LVQ is illustrated. At any k^{th} step, the distances between one of training patterns $P_i \in R^n$, $i = 1,2,..., l$, and the neurons (or reference vectors[3]) $N_j \in R^n, j = 1, 2,..., m$, are measured using Euclidean distance (ED) metric to find the nearest neuron N_c.

$$ED_j = \sum_{q=1}^{n} (p_q - n_q)^2 \quad (2)$$

where, P_q Elements of P_i

n_q Elements of N_j

The neurons, N_j's, are initially located randomly in the pattern space, and the closest neuron, N_c, becomes a candidate for one of the many cluster centers that will appear after all steps are performed. If the closest neuron has the identical class label as the pattern, this neuron is moved toward the pattern as the reward for a correct classification. Otherwise the neuron is moved away from the pattern as the punishment for an incorrect classification.[3] Equation 3a is used to represent the move toward the pattern, and Equation 3b is used for the move away. For all other neurons, Equation 3c is applied.

$$N_c^{k+1} = N_c^k + \alpha(P_j - N_c^k) \quad (3.a)$$

$$N_c^{k+1} = N_c^k - \alpha(P_j - N_c^k) \quad (3.b)$$

$$N_j^{k+1} = N_j^k, \text{ for } j \neq c \quad (3.c)$$

where, α is a monotonically decreasing momentum rate and preferably less than 1.0.[3] In practice, the determination of α in non-trivial. When the neuron N_c is moving toward the pattern, it is known that the pattern belongs to this neuron at the k^{th} iteration. The same method will be applied to all available patterns, and the step will be repeated iteratively until all the clusters are formed.

GDR also requires knowledge patterns which have inputs and corresponding target outputs for training. The knowledge patterns are supplied to the net in a feed-forward manner to find a connecting weight matrix, and then those weights are adjusted by the back-propagation of error to reduce the total net error. The GDR net shown in Fig. 6 uses the typical artificial neural net construction introduced in Fig. 4. The outputs of the nodes in one layer are transmitted to nodes in the next layer through connections that amplify, attenuate, or inhibit such outputs through connecting weights.[1] The net may have a number of hidden layers. However, in practice, only one or two hidden layers are sufficient for most applications.[5]

The output of a node in the input layer i is

$$O_i = I_i, \; i = 1, 2,...,n \quad (4)$$

The net input to a node in layer j is

$$net_j = \sum_i W_{ji} O_i, \; j = 1, 2,..., m \quad (5)$$

The output of node j is

$$O_j = \frac{1}{1 + e^{-f}} \quad (6)$$

$$f = net_j + \theta_j \quad (7)$$

In Expression 7, the parameter θ_j serves as a threshold or bias. Similarly, input net_k and output O_k can be found by substituting the subscript j to k in Equations 5 through 7.

$$net_k = \sum_j W_{kj} O_j, \; k = 1, 2,..., l \quad (8)$$

$$O_k = \frac{1}{1 + e^{-f}} \quad (9)$$

$$f = net_k + \theta_k \quad (10)$$

All knowledge patterns will be fed through the net by the feed-forward procedures, Equations 4

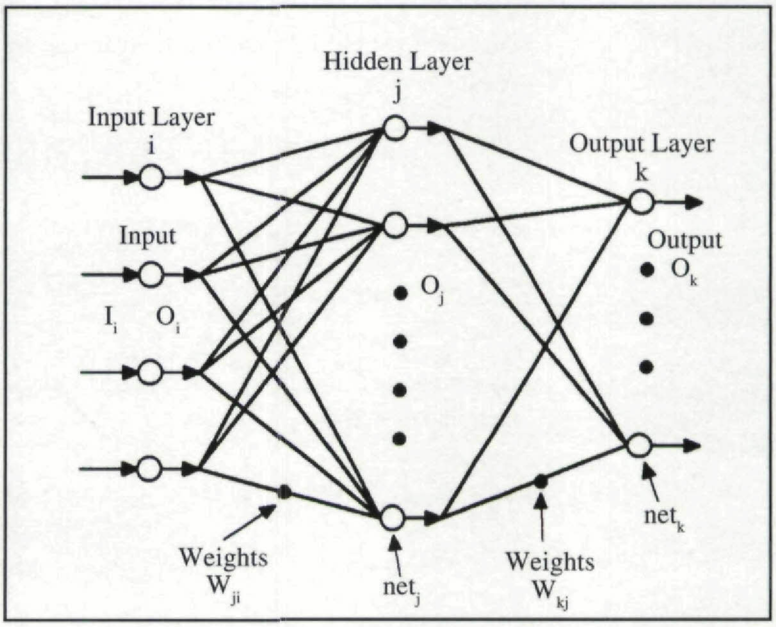

Fig. 6 - Net construction of GDR algorithm.

through 10. Usually, outputs $\{O_{pk}\}$ generated by the net will not be the same as the target or desired outputs $\{T_{pk}\}$. The square of the difference (or pattern error) between these two values is

$$E_P = \frac{1}{2}\Sigma\,(T_{pk} - O_{pk})^2 \qquad (11)$$

and the average net error is

$$E_{net} = \frac{1}{2p}\sum_p \sum_k (T_{pk} - O_{pk})^2 \qquad (12)$$

$$p = 1, 2, \ldots, P$$

where, P Number of Patterns

If E_{net} falls into the acceptable error range, the net is successfully trained. Otherwise, the following procedures are necessary to minimize the error. The convergence toward improved values for the connecting weights and thresholds can be achieved by taking incremental changes ΔW_{kj} proportional to $\partial E/\partial W_{kj}$.[1]

$$\Delta W_{kj} = -\eta\,\frac{\partial E}{\partial W_{kj}} \qquad (13)$$

$$= -\eta\,\frac{\partial E}{\partial\, net_k}\,\frac{\partial\, net_k}{\partial W_{kj}}$$

where, η Learning Rate

Therefore,

$$\Delta W_{kj} = -\eta\,\delta_k\,O_j \qquad (14)$$

$$\text{where, } \delta_k = -\frac{\partial E}{\partial\, net_k},\quad O_j = \frac{\partial\, net_k}{\partial W_{kj}}$$

The term δ_k, which is the error to be propagated backward for the k^{th} node in the layer, can be rewritten as

$$\delta_k = -\frac{\partial E}{\partial O_k}\,\frac{\partial O_k}{\partial\, net_k} \qquad (15)$$

$$= (T_k - O_k)\,f'_k\,(net_k)$$

$$= (T_k - O_k)\,O_k\,(1 - O_k)$$

By similar mathematical procedures (details can be found in Ref. 1),

$$\Delta W_{ji} = -\eta\,\delta_j\,O_i \qquad (16)$$

$$\delta_j = O_j\,(1 - O_j)\sum_k \delta_k\,W_{kj} \qquad (17)$$

The δ's at an internal node can be evaluated in terms of the δ's at an upper layer. Thus, starting at the highest layer (or output layer), δ_k can be evaluated using Expression 15, and the errors can be propagated backward to the lower layers. The connecting weights now will be updated as follows,

$$W_{ji}^{n+1} = W_{ji}^{n} + \Delta W_{ji}^{n} \qquad (18)$$

$$\text{where,}\quad \Delta W_{ji}^{n} = \eta\,(\delta_j O_i) + \alpha\Delta\,W_{ji}^{n}$$

The momentum rate α has been added to Expressions 14 and 16 to reduce the risk of oscillations while training the net in the iterative approach.[1] The α also allows a larger value of η, thereby speeding convergency.[4] Both η and α influence the training results and should be carefully selected by trial and error. The improved connecting weight matrix will be used at the next iteration, and the procedure is repeated until the system error reaches the desired level.

Applications

As mentioned earlier, two steps of the initial gear design are emulated using the artificial neural nets. Although it is possible to apply a single neural net to perform the desired task, two different algorithms, LVQ and GDR, are used intentionally in order to emulate human performance more accurately. It will also prevent from training a single neural net with the entire patterns which may be thousands.

The product catalog[12] obtained from the local gear manufacturing company served as the training knowledge patterns. The catalog contains three input values, horsepower, input RMP, and speed ratio. In addition, the catalog also includes the model number which implies proper center distance. The patterns are neatly tabulated to the ones which may use the same center distance. The model numbers in the catalog were used as the class labels, as well as the desired outputs of each pattern.

The patterns of four selected models are plotted in Fig. 7. From this figure, it can be seen that the patterns belonging to one model are scattered along the axes of speed ratio and input RPM. The patterns in each model tend to form a distribution surface which may be the portion of a sphere. However, it is almost impossible to form any clusters with this kind of pattern. Thus, the original three dimensional patterns are transformed and mapped onto a two dimensional pattern space using Equations 19 through 22. Fig. 8 shows the transformed patterns mapped onto the new space.

$$A = 100\ \frac{I_1\ I_2}{I_3} \tag{19}$$

$$B = 10\ \frac{e^{I_1} I_2^3}{I_3^2\ \overline{I_3}} \tag{20}$$

$$X_1 = 5\ \overline{A}\ \ \overline{B} \tag{21}$$

$$X_2 = 3A \tag{22}$$

where, I_1 Speed Ratio

I_2 Horsepower

I_3 Input RPM

X_1, X_2 Transformed Pattern

Fig. 9 illustrates the multiple GDR net construction connected to a single LVQ net for the initial gear design application. The number of GDR nets required is determined by the number of models available in the product catalog. Accordingly, each GDR net is to be trained with the patterns that belong to the same model. A single hidden layer with three nodes is used for each GDR net. For a triple-reduction case, three such multiple nets should be combined.

The number of clusters formed using the LVQ net depends upon the size of the limiting radius r in Fig. 5, which controls the size of the clusters. If the limiting radius is overly large, some clusters having different model numbers (or different class labels) will overlap. If the limiting radius is too small, too many clusters will be formed. Therefore, an optimized value is required.

After the each net is successfully trained, the LVQ net can produce the model number and center distance when a new input pattern (horsepower, speed ratio, and input RMP) is provided. The output, a model number, will serve to determine the matching GDR net which will estimate the diametral pitch. The GDR net also uses the same input as the LVQ net. In real-world design, the number of pinion and gear teeth are estimated, and the diametral pitch is calculated using this estimation. However, the number of pinion and gear teeth relative to the speed ratio are not functionally distributed. Therefore, the diametral pitch is selected as the target output in this application. Afterwards, the other parameters can be calculated sing the estimated param-

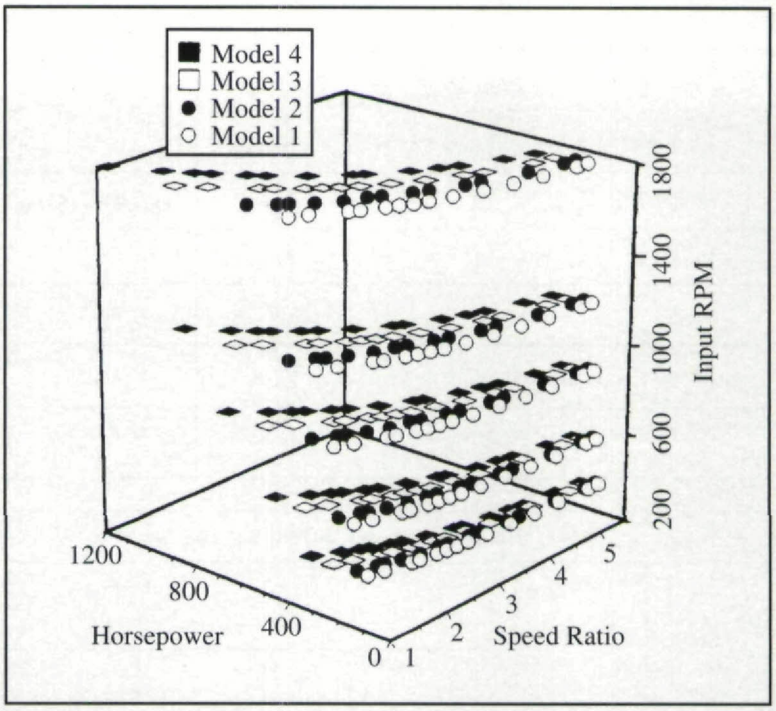

Fig. 7 - Original catalog training patterns.

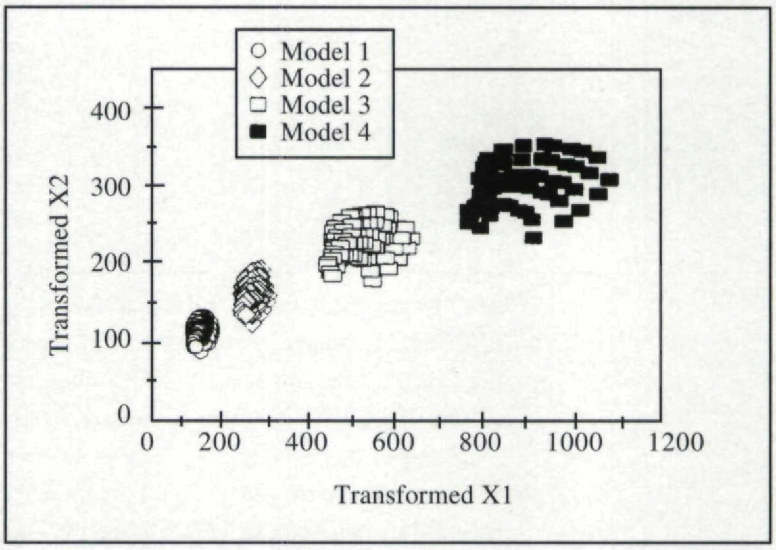

Fig. 8 - Transformed training patterns.

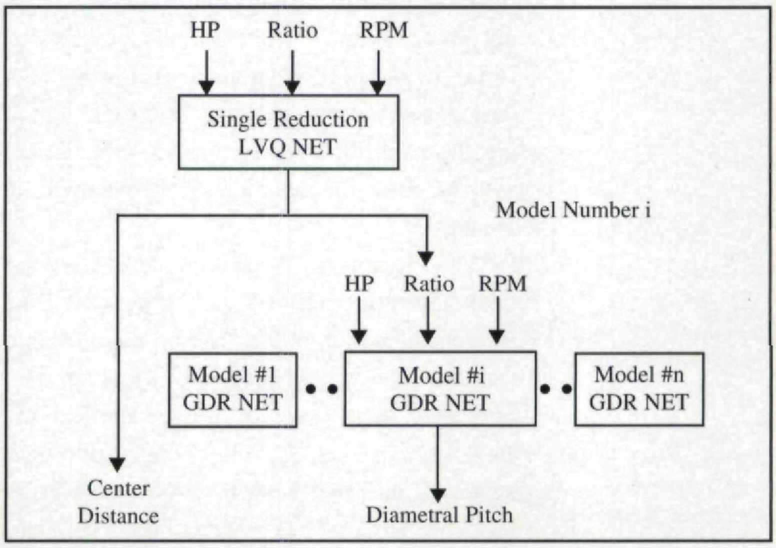

Fig. 9 - Multiple network construction for initial gear design application.

Table I - Error of Estimated Outputs			
70 patterns in each model	Error % before adjusted	Error % after adjuster	Total Net error
Model 1	20	7	0.000041
Model 2	14	7	0.000043
Model 3	5	5	0.000018
Model 4	9	9	0.000023
Average	14	7	0.000031

Table II Error Factors for different η and α					
		Error Factors			
η	α	Model 1	Model 2	Model 3	Model 4
0.70	0.50	1.03	1.05	1.01	1.01
0.90	0.70	1.00	1.00	1.00	1.00
0.95	0.90	0.92	0.94	0.93	0.97
0.99	0.95	0.93	0.97	0.90	0.90

eters, the center distance, and diametral pitch.

In Table I, the percentages of the net errors after 20,000 iterations are tabulated. The percentages indicate the number of incorrect estimations made by the GDR nets over the number of the training patterns. While investigating those incorrectly estimated diametral pitches, it was found that some of the values were not commonly used in the gear industry. Thus, those uncommon diametral pitches must be adjusted to the recommended values.(6) The error percentages were decreased after adjustment, which are shown in the third column in Table I. The average error for the four selected models is practically acceptable.

There are several considerations in using the GDR algorithm for initial gear design task. The first consideration is how to find the adequate learning rate, η, and momentum rate, α. The typical values of η and α for most applications are 0.9 and 0.7, respectively.(3) Suggestions can be found in Table II, which shows the error factors relative to the typical values. The η can be selected between 0.95 and 0.99, while the α can be selected between 0.9 and 0.95. When the α was increased higher than 0.95, the training seemed to become trapped in a local error minimum, and the error was not improved. It was also found that the number of iterations higher than 20,000 did not improve the results.

How the available training patterns were organized was also important. The test was performed with three different sorting methods of training patterns; sorted by input RMP, by horsepower, and by speed ratio. As a result, it was learned that the training patterns sorted by input RMP order produced the best results. When the number of nodes in the hidden layer was increased to six, no improvement was observed at the same number of iterations. When the number of decimal places was increased from two to four, the number of iterations was decreased by 25% at the same error level.

Numerous test designs were completed with the entirely trained artificial neural net. Each test design was evaluated by a commercially available AGMA power rating software.(13) About 60% of the test designs passed the power rating without changing any initial gear size parameters, while the balance required several changes to pass within a few iterations.

Conclusions

Once the net is trained with the available design knowledge, it can provide the estimated output in a single iteration, usually in seconds. If the outputs generated by the net have been approved as good estimations, these input and output patterns can be added to the existing design knowledge in order to achieve better performance in the future. The company's design knowledge will grow automatically by adding new patterns to the knowledge data base. It will ensure that all available design knowledge of engineers is collected and organized without special effort. By using the artificial neural net, the design time for inexperienced engineers can be reduced, and a design consistent with past designs achived.

Another advantage is that the artificial neural net can be trained to deal with incomplete and uncertain evidence. It understands the relationship between inputs and outputs, and does not burden the engineer with specific analyses. If conventional techniques are used, the engineer must find their mathematical relationship before developing any system, which may require many years of field experience and an extensive mathematical background.

Although the artificial neural net successfully emulates the performance of the human engineer for the initial gear design task, there are still some disadvantages to overcome. The most critical disadvantage is the slow training time. It took hours to train a neural net with 70 knowledge patterns in one model, which consisted of only three inputs and one output, on a fairly capable personal computer, such as a 80386-based PC. In the case of single reductions, 22 such models are to be found in the catalog used. Furthermore, when new knowledge patterns are to be added to the existing patterns, the entire neural net must be retrained.

Human neurons transmit signals at a very slow speed, considering the immense velocity of signal transmission in a modern digital computer. However, the brain's huge computational rate is achieved by a tremendous number of parallel computational units.(2) The most advanced modern computer systems are packed with only a few parallel processing units, implying that the ability of the artificial neural net is limited by current computer hardware technology.

As previously mentioned, another important fact is that inaccurate training knowledge patterns will lead to inaccurate estimated outputs. Thus, knowledge patterns must be prepared carefully before any artificial neural net is applied in real practice.

Nevertheless, the results of this work provide the applicability of the artificial neural net to the initial gear design to emulate the decision-making tasks of the human engineer using the identical design steps. In general, similar methods can be adapted to many mechanical engineering design problems. More detailed implementation must be carried out to enhance the quality of estimations of the artificial neural net. ■

Acknowledgements: *Deepest appreciation goes out to Mr. J. R. Dammon of Fairfield Manufacturing Company, Inc. Without his generosity in providing the AGMA power rating software, none of the results of this work could have been properly evaluated.*

Originally presented at the AGMA Fall Technical Meeting, 1991. Reprinted with permission. The opinions, statements, and conclusions presented are those of the author and in no way represent the position or opinion of AGMA.

References:

1. Pao, Yoh-Han. *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley Publishing Company, Inc., 1989, pp 113-140, 197-222.

2. Wasserman, Philip D. *Neural Computing: Theory and Practice*, Van Nostrand Reinhold, New York, 1989, pp 43-59, 189-199.

3. Kohonen, Teuvo. *Self-Organization and Associative Memory*, 3rd ed., Springer-Verlag, New York, 1989, pp 185-209.

4. Wasserman, Philip D. and Schwartz, Tom. "Neural Networks, Part 2: What are they and why is everybody so interested in them now?", *IEEE Expert*, Spring, 1988, pp 10-15.

5. EEAP484: Adaptive Pattern Recognition and Neural Networks. Class Lecture Note, Case Western Reserve University, 1990.

6. Dudley, Darle W. *Handbook of Practical Gear Design*, McGraw-Hill Book Company, New York, 1984.

7. Drago, Raymond J. *Fundamentals of Gear Design*, Butterworths, Stoneham, MA, 1988, pp 265-377.

8. AGMA Standard 2001-B88. "Fundamental Rating Factors and Calculation Methods for Involute Spur and Helical Gear Teeth," American Gear Manufacturers Association, 1988.

9. AGMA Standard 6010-E88. "Standard for Spur, Helical, Herringbone, and Bevel Enclosed Drives," American Gear Manufacturers Association, 1988.

10. Dixon, John R. and Simmons, Melvin K. "Expert Systems for Mechanical Design: A Program of Research," ASME Technical Paper 85-DET-78, ASME, New York, 1985.

11. Kumar, A., Kinzal, G.L., and Singh, R. "A Preliminary Expert System For Mechanical Design," *Proceedings*, 1986 ASME International Computers in Engineering Conference, pp 29-35.

12. "MARK II Parallel Shaft Speed Reducers." Product Catalog H-87, The Horsburgh & Scott Co., Cleveland, OH.

13. "Gear Design Software." Fairfield Manufacturing Company, Inc., Lafayette, IN, 1988.