TABLE OF CONTENTS

AUTOMATED INSTRUMENT LOOP SKETCH SYSTEM

Geo-Ry Tang
Department of Mechanical Engineering, National Taiwan Institute of Technology, Taipei, Taiwan

Shield B. Lin
Department of Mechanical Engineering, Prairie View A&M University, Texas, U.S.A.

Yi-Hua Wei
Instrumentation and Control System Department, CTCI Corporation, Taipei, Taiwan

I. Abstract

A Computer-Aided Design System for instrument loop sketch is presented in this paper. The structure of the Automated Instrument Loop Sketch System (AILSS) consists of a Preprocessing Unit, a Layout Unit, and a Postprocessing Unit. Professional knowledge, artificial intelligence algorithms, and computer graphics techniques have been applied to develop the system. As an explicit example of using the AILSS, we consider an application to the control loop of a reaction furnace. The output of the various units of the AILSS is shown. The correctness of the results supports the algorithms used in the program.

II. Introduction

Instrument loop diagrams are widely used in industrial plants for design, construction, operation, and maintenance. An instrument loop diagram usually contains information of components and connections among the components. The typical information includes instrument identification, type, function, location, purchase specification, installation details, connection type, signal level, identification number of junction box, computer I/O assignment, and emergency shutdown system [1, 2]. Since most of the information in loop diagrams depends on vendors' drawings on individual instruments, loop diagrams can not be constructed until the end of a designing process. Therefore, instrument engineers tend to prepare loop sketches as the first step towards designing a new plant [3].

A loop sketch, the skeleton of a loop diagram, consists of instrument components and their connections. Traditionally, a well-experienced engineer is appointed to be a loop sketch designer of an instrument design group. He refers to the documentation supplied by other design groups such as process group and piping group, analyzes control strategies based on process diagrams, determines the necessary instrument components, designs appropriate connections among the components, and produces loop sketches.

An effort to use computer-aided design to modernize the design process of a loop sketch is undertaken. This effort is intended to combine the techniques of computer graphics, artificial intelligence and database management with instrument design knowledge. This paper describes the authors' work on developing an **Automated Instrument Loop Sketch System.** The system can automatically generate loop sketch drawings from the given process and piping information.

III. System Configuration

The AILSS consists of a preprocessing unit, a layout unit, and a postprocessing unit as shown in Figure 1. The functions of each unit are as follows:

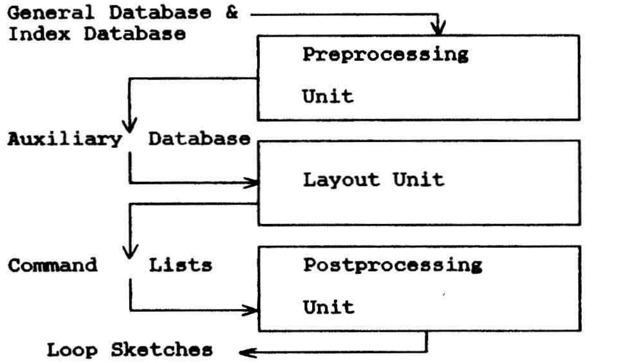

Figure 1. Configuration of the Automated Instrument Loop Sketch System

(1) Preprocessing Unit

To draw a loop sketch, several information sources must be referred to. They are Piping and Instrument Diagrams (P&ID's), Process Flow Diagrams (PFD's), instrument indexes, and instrument specifications. For computerization, the related information has to be stored in terms of databases on a computer. The authors have defined two databases for the information sources: (1) General Database - the data collected from given P&ID's and PFD's. (2) Index Database - instrument indexes and specifications; this is a fixed database which provides detailed information of all standard instruments.

Since a General Database includes information not only for the loop sketches but also for the other items of the process instrumentation, it is quite inefficient to pass all the data to main body of the Loop Sketch System. By executing the preprocessing program, an Auxiliary Database can be created. The preprocessing unit excludes all individual, stand alone instrument components such as level gauges, thermometers, and safety relief valves. It also ignores all unrelated data fields like service fluid, operation conditions and mounting specifications. Besides screening data from the source database, the preprocesser is also used to check correctness and completeness of the created Auxiliary Database. The reasons are twofold: first, the final version of P&ID's and PFD's is not determiend at the beginning of the project; secondly, not every instrument component can be explicitly shown on those diagrams. The former fact could result in inaccurate data to be stored in the Auxiliary Database. The latter might cause the incompleteness of the Database. The effect of these drawbacks may be reduced by assigning a senior engineer who takes the responsibility to correct the Database. An alternative method, which is presented here, is to establish rule-based expert algorithms to fulfill the assignment originally dedicated to the senior engineer. A typical rule in preprocessing unit is in the following:

```
(∃x)(∃y) {        [TYPE (x CONTROLLER)       & LOCATION (x DCS) ]
             &    [TYPE (y CONTROL-VALVE)    & POWER (y AIR)    ]
             &    [CONNECTED (x y)                              ]   }
& ¬(∃z)  {        [TYPE (z I-P)                                 ]   }

=>  MODIFY-DB     (     (add I-P FILED        )
                        (disconnect x y       )
                        (connect x I-P        )
                        (connect I-P y        )    )
```

This rule states that within a loop if there is a Distributed Control System (DCS) controller in the control room, and a pneumatic control valve in the field but without a Current to Pneumatic (I-P) transducer, then an I-P transducer will be added to the component list of the Auxiliary Database. The connection list of the Auxiliary Database will also be revised in terms of the signal from the controller through the transducer to the control valve, instead of the signal from the controller directly to the control valve.

In a practical application, several rules may satisfy the same set of preconditions. Weighting factors or probability functions are introduced to discriminate the priorities of rules to be fired. A more reliable Auxiliary Database can therefore be obtained.

The output from the preprocessing unit is a text file with ordered dataset. Each instrument component is separated from the other by the end-of-line character, and each group of instruments is separated from the other by a blank line. An example of the dataset of the Auxiliary Database generated by the preprocessing unit is shown on Table 1.

Loop ID.	Instr.ID.	Loc.	Type	Conn.No.1	Conn.No.2	Conn.No.3	Conn.No.4
P016	FT-041	A	T1	ISB3 S			
	ISB3	C	ID	FC-041 W			
	FC-041	G	SP	FIC-041 N	ISB1 W		
	FIC-041	H	DB				
	ISB1	C	ID	FY-041 S			
	FY-041	A	IP	FV-041 A			
	FV-041	A	CC				
	PY-016B2	G	MU	PY-016B1 N	PY-016A W	AC-033 F	FC-042
	PY-016B1	H	DB				
	PY-016A	G	SU	PY-016 W			
	PY-016	G	LE	FC-041 W	LC-016 F		
	PT-016	A	T1	ISB2 S			
	ISB2	C	ID	PC-016 W			
	PC-016	G	DA	PIC-016 N	PY-016B2 W	PY-016A W	
	PIC-016	H	DB				

Table 1. Output of the Preprocessing Unit - Auxiliary Database

(2) Layout Unit

The layout unit consists of two steps: instrument placement and routing. First, a specific, unique position can be assigned to each instrument, which is represented by a predefined symbol [1, 3]. Then, symbols are connected by lines according to signal transmitting direction. More discussion of the two steps is described in the following:

(i) Instrument Placement

Several techniques for automated placement problems have been applied to various areas such as integrated circuit design, cutting plate arrangement, and architectural plan layout. Typical algorithms include pattern recognition [4], force directed and annealing method [5, 6]. By convention, a loop sketch is partitioned into several divisions, each of which corresponds to a particular physical location as shown in Figure 2. From the bottom of the loop sketch to the top, the instrument components should be placed in order of the divisions of Local Devices, Junction Box, Marshaling Cabinet, Input/Output of Distributed Control System, Distributed Control System, and Operator Console.

Among the divisions, Distributed Control System, and Local Devices usually have more components and much complicated connections. Hence, the placement will begin with either of these two divisions. Once the arrangement of the two divisions have been completed, the rest of the divisions can be easily defined. Another constraint of Instrument Placement is the alignment of a group of instrument components among divisions.

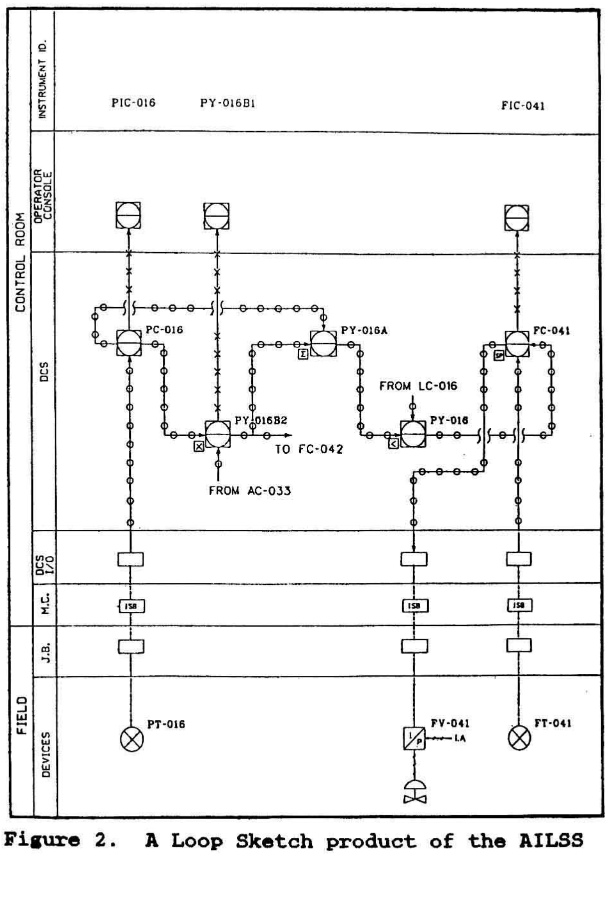

Figure 2. A Loop Sketch product of the AILSS

(ii) Routing

The routing among a number of predefined symbols in a loop sketch is similar to that of finding a free path for a mobile robot among obstacles. Two algorithms have been used in the routing process.

(A) Direct Connection Approach:
A straight line segment connecting a pair of symbols is generated first. If the line segment intersects with other symbols, the original straight line is either completely replaced by two line segments or partially substituted by a few line segments around the boundary of the obstacle as shown in Figure 3. Selected control strategies will determine the location of deflection point 'D' in Figure 3-b or to compute the minimal safety distance 'e' in Figure 3-C. The same algorithm can then be applied to the modified line segments recursively until no more intersection exists.

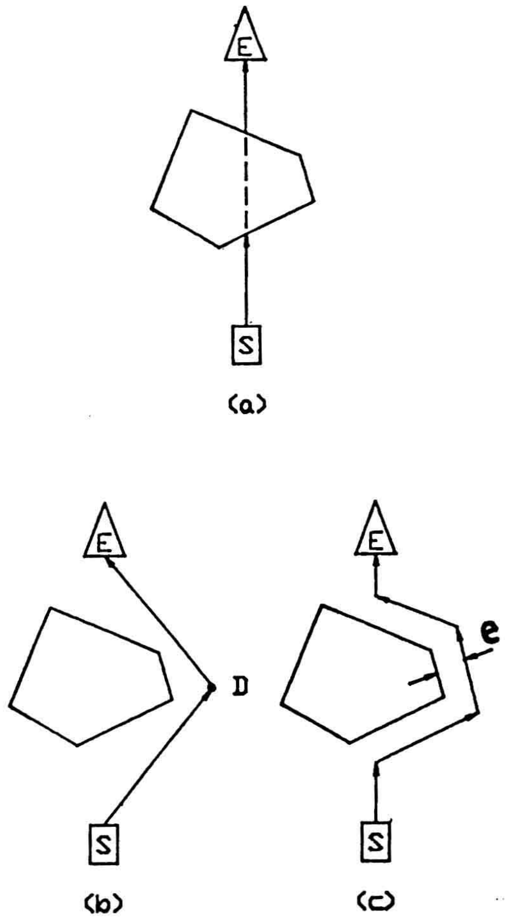

Figure 3. Direct Connection Approach of Routing

(B) Free Space Approach
The empty space in a loop sketch is converted into many convex polygons. The overlaps of any two or more such polygons can be specified as the transition zones [7,8]. Since any two points in a convex polygon can be connected by a line segment which does not intersect instrument symbols, finding a connection between two instrument symbols is achieved by connecting a series of transition points which represent the end points of line segments.

Since there may exist a large number of connections between two symbols, some constraints have been introduced to reduce the size of the searching tree. For instance, the geometric center of a transition zone is taken to be the only transition point in that area. Then the routing program will handle a problem involved with a finite number of convex polygons and a finite number of transition points. An alternative constraint can also be imposed to simplify the routing problem. A loop sketch can be drawn on a quadrille paper. The line segments connecting two symbols are restricted to follow the grids on the paper. The resolution of grids can be decided according to the capability of the computing facility. Higher resolution requires more computer memory and greater computing power.

The output of the layout unit will be stored as text files in standard ASCII code. An example of the output file is listed in Table II that was created from the input information shown in Table I. The first column of Table II shows types of drawing elements, such as block, line segment, or character string. The rest of the parameters specify the dimension, location, and other options.

NAME	PIC-016	PY-016B1		FIC-041
BLOCK	PIC-016	DB	4	15
BLOCK	PY-016B1	DA	9	15
BLOCK	FT-041	T1	26	-10
BLOCK	FV-041	CC	20	-14
BLOCK	PY-016	SU	20	3
...	...	.	.	.
...	...	.	.	.
...	...	.	.	.
LINE W	25 8 21 24 8 3 7 24 1 2 4 20 1 3 1			
LINE W	21 30 7 28 3 1 5 28 8 2 1			
LINE S	4 -9 4 1			
LINE A	20 -11 3 2			
LINE W	3 8 2 1 2 8 1 2 2 10 0 13 15 10 3 1			
... .	....			
... .	....			
... .	....			

Table II. Partial Output of the Layout Unit - Command Lists

(3) Postprocessing Unit

The postprocessing unit functions as an interpreter to transfer the output of the Layout Unit to instrument loop drawings. For the purpose of generating drawings on various machines, the text files from the layout unit are designed to be machine-independent.

Interactive graphics software packages are selected to produce loop sketches. This is because loop sketches are often subject to revision during the progress of a project progressing and the modification of drawings can easily be done on the interactive graphics systems. Two systems have been chosen to generate graphical outputs for the prototype AILSS. A program written in AutoLISP can be executed to produce loop sketches through AutoCAD on a personal computer. The other program, composed by the routines written in C and User Command, can coupled with the Interactive Graphics Design Software, can be executed on a VAX8600 computer. An example of a loop sketch product has been shown earlier in Figure 2.

IV. CONCLUSIONS AND RECOMMENDATIONS

A prototype Automated Instrument Loop Sketch System has been developed. It is a computer-aided design tool for instrument loop design. The software system mainly written in LISP uses artifical intelligence rules to determine the locations and connections of instrument components. A personal computer or a miniframe system can be used to execute the program. The AILSS can generate correct loop sketches from given plant design specifications. However, the products of the AILSS look like the design of a junior instrument engineer. The main problem is that it is quite difficult to transfer the concept of a 'good' design to computers through mathematical equations or rule declarations. Another limitation is that the emergency shutdown signals were not considered in developing the AILSS. The performance of the AILSS can be improved upon by overcoming either of the drawbacks.

V. REFERENCES

[1] CTCI, "The Design Guide of Instrument Loop Diagrams," CTCI Corporation, Taipei, 1989.

[2] ISA, "Instrument Loop Diagrams", Instrument Society of America Standard and Recommended Practices,ANSI/ISA-S5.4-1976(R 1981).

[3] ISA "Instrumentation Symbols and Identification," Instrument Society of America Standard and Recommended Practices, ANSI/ISA-S5.1-1984.

[4] M.W. Firebaugh, "Artificial Intelligence A Knowledge-Based Approach," Body & Fraser, Boston, USA, 1988.

[5] A.D. Brown, "Automated Placement and Routing," Computer-Aided Design, Vol. 20, No. 1, 1988, pp. 39-44.

[6] N.R. Quinn and Breuer, M.A., "A Forced Directed Component Placement Procedure for Printed Circuit Boards," IEEE Transactions on Circuit and System," Vol. CAS-26, No.6, 1979.

[7] R.A. Brook, "Solving the Find-Path Problem by Good Representation of Free Space," IEEE Transaction on System, Man, and Cybernetics, SMC-13, No. 3, 1983.

[8] S. Singh and M.D. Wagh, "Robot Path Planning Using Intersecting Convex Shapes," Proceedings of IEEE International Conference on Robotics and Automation, Apr. 1986.

Keywords: CAD, Instrumentation, Loop Sketch, Expert System.

COMPUTER-AIDED DESIGN OF POWER ELECTRONIC DEVICES USING A FINITE ELEMENT METHOD OF THERMAL PHENOMENA

A.Skorek *, M.B.Zaremba **, I.Popescu **, V.Rajagopalan *

* Université du Québec à Trois-Rivières, Département d'ingénierie, Groupe de Recherche en Électronique Industrielle C.P.500, Trois-Rivières, Québec, Canada / G9A 5H7

** Université du Québec à Hull, Département d'informatique, C.P.1250, succursale "B",Hull, Québec, Canada, J8X 3X7

Abstract

This paper examines the modelling and analysis of some classes of power electronic devices, using the finite element method. A simple analysis of the thermal coupling in thyristors working in a 2 kW, 16 kHz power supply unit is given. The calculation have been done on an HP Vectra ES/12 microcomputer using the NISA II PC program. The utility and the capabilities of this type of analysis are discussed with emphasis on the computer-aided design process for power electronic elements and devices.

I. INTRODUCTION

The analysis of thermal phenomena is a very important step in the design of power electronic devices. An increase in the amount of heat dissipated inside a power electronic device during its operation results in an increase of junction temperatures.One of the characteristics of power electronic devices is the existence of thermal couplings that may produce adverse changes in electrical parameters. The problem of thermal couplings has not been fully resolved. Complex element shapes and differences in their location make a generic solution of the problem impossible.

Numerical solution of the thermal conductivity equation in 3 dimensions requires very large memory and long computation times. The problem is even more complex if the analysis includes not only the temperature distribution in the chip,but also the temperature in the housing. Consideration of the non-linear relationship between the conductivity of Si and the temperature, as well as other heat transfer mechanisms such as convection and radiation, add to the computational complexity.

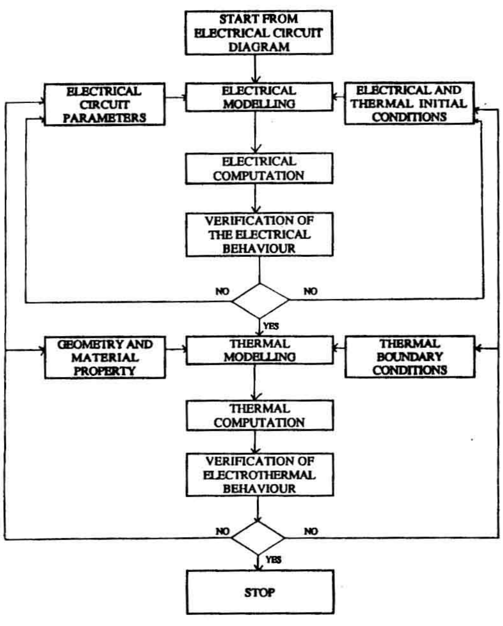

Figure 1: Illustration of the Design Method for Power Electronic Circuits

Recent developments in computer technologies lead to the replacement of many traditional design methods by newer, faster and more accurate ones. This is also the case in power electronics, where computer-aided design is more and more frequently used.

Computer-aided design and analysis in power electronics is mainly based on the application of mainframe computers (e.g.CYBER,VAX). The use of PCs for these types of computation have not been fully realized so far; because limitations in the thermal analysis are an inherent part of the design process of every electrical circuit, and the power electronic circuit in particular (Fig.1)

Package designers can perform thermal analysis using a number of methods. Finite element analysis is particulary popular but, so far, it has required considerable computing, which practically excludes the possibility of the use of PCs. PC programs elaborated recently show the remarkable usefulness of finite element method (FEM) for thermal analysis of power electronics devices. The programs equipped with preprocessing and postprocessing modules allow not only fast numerical modelling of the analysed problem but also provide clear and convenient interpretation of computations as well.

The capabilities of one such program NISA IIPC/DISPLAY II are presented in an example of the analysis of thermal coupling of a power electronic system.

II. SYSTEM DESCRIPTION

Let us consider a system consisting of two thyristors located on a common heat sink, each of them dissipating power p_i which is a function of the electrical state of the components (Fig.2)

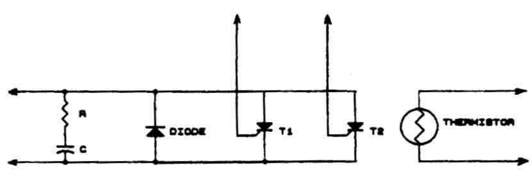

Figure 2. Electrical Diagram of the Thyristors Block in the 2 kW, 16 kHz Power Supply

The power dissipated in a thyristor is described by a function

$$\Sigma q = \frac{1}{VT_p}\int_0^{t_p} U_p(t)\, i_p(t)\, dt + \frac{1}{VT_p}\int_{t_p}^{t_z} U_z(t)\, i_z(t)\, dt + \frac{P_g}{V} + \frac{P_p}{V} \qquad (1)$$

where

$u_p(t)$, $i_p(t)$ - instanteneous values in on-state of voltage and current,respectively; [V], [A].
$u_z(t)$, $i_z(t)$ - instantaneous values in off-state of voltage and current,respectively; [V], [A].
P_g, P_p - gate power dissipation and switching power dissipation;[W].
t_p, t_z - on-state time and off-state time; [sec].

V - volume of source region; [mm^3].

T_p - period; [sec]

In practice, it is difficult to obtain the individual terms of equation 1. The problem becomes even more complex when the deformation of current and voltage waveforms and the increase of frequency are considered (Fig.3).
For these reasons, the application of computer-aided analysis to electrical behaviour using one of the standard PC programs is of much interest (e.g. ATOSEC,PSPICE).
Electrical analysis of our system allowed us to determine the steady state power p_1=0.025 W/mm^2 and p_2=0.02 W/mm^2 for thyristors T_1 and T_2.

voltage current

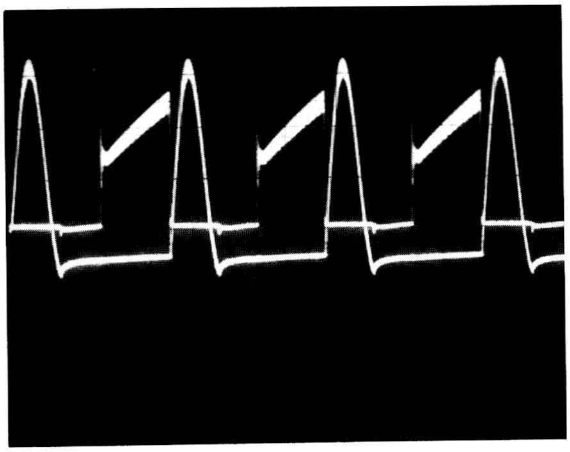

Figure 3. Current and voltage oscillograms for the thyristor T_1
a) voltage (20 µs/div and 100 V/div)
b) current (20 µs/div and 5 A/div)

The system is located in a room having a mean temperature of 30 °C. The dissipation of heat to the environment from the thyristors and the heat sink take place by convection (h = 0.000016 W/mm^2 °C).
The system design requires a determination of the temperature field, with particular consideration of the junctions, and the identification of the thermal coupling.
In order to solve the problem, modelling and simulation are applied using a PC program for finite element analysis.
It is of interest to determine the total time requirement and ease of the modelling, the computation time, and the precision and interpretation of the results.

III. MODELLING AND SIMULATION

The modelling steps are shown in Fig.4.
The following stages of modelling and simulation are involved in the example under discussion:
- Generation using a CAD technique of the drawing and the mesh of finite elements
- Input of the properties of the materials
- Input of the boundary conditions
- Calculations using the heat transfer analysis module
- Display of the results

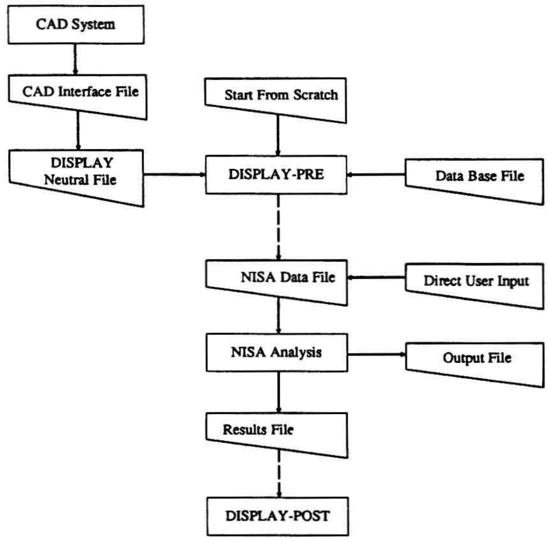

Figure 4: Preprocessing,Analysis and Postprocessing using NISA and DISPLAY [3]

Due to PC memory constraints, only a limited number of the nodes can be taken into account, and some of the geometry simplifications are usually adopted. They do not have a significant impact on the precision of the description of the system's thermal phenomena however and consequently, on the total error. A simplified geometric model together with a generated mesh of finite elements is presented in Fig.5.
All the geometrical data for the model are automatically translated into a NISA digital file,thus providing easy access to an editor.

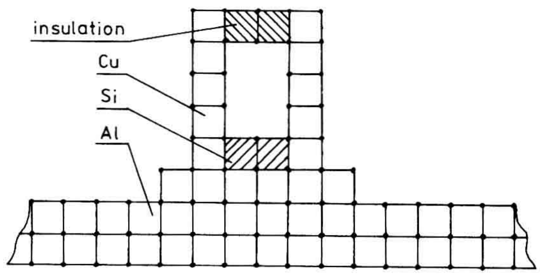

Figure 5: Geometrical model of the thyristor

```
*MATHEAT
 KXX ,      1 , 0, 0,  0.18
 KXX ,      2 , 0, 0,  0.384
 KXX ,      3 , 0, 0,  0.152
 KXX ,      4 , 0, 0,  0.01
*ELHEAT
164,0.025,165, ,0,0
143,0.02,144, ,0,0
*CONVBC
      82,136,2,3,-1,0,0,0
     0.0000166,     30
      81,135,2,1,-1,0,0,0
     0.0000166,     30
      81, 82,1,4,-1,0,0,0
     0.0000166,     30
     135,136,1,2,-1,0,0,0
     0.0000166,     30
```

Figure 6: Properties of the Materials and some Boundary Conditions stored in the NISA Date File

The properties of the materials and the boundary conditions may be given as input either during the preprocessing stage (Fig.4) or directly to the NISA Data File (Fig.4) using a text editor. A sequence of simple instructions formulating the properties of the materials and some of boundary conditions for our example are presented in Fig.6.
The geometrical and physical data presented above for the system are encoded entirely in the NISA Data File which constitute the input to the heat transfer analysis program (NISA Analysis in the figure 4). In our case the numerical analysis required 118 seconds for the system without the coupling,and 119 seconds for the system with coupling for HP Vectra ES/12 Personal Computer operating at 12 MHz.
The resulting temperature values along with a precise description of the individual stages of the modelling process are stored in the Output File (Fig.4), that can be displayed on a monitor or printed in hard-copy form. A very convenient and fast form for presenting the results of the analysis is a DISPLAY-POST (Fig.4) with the capability of zooming and scaling of graphical maps of the temperature field.
The graphical maps of the temperature field in our example are shown in Fig.7 (page 4)
It can be clearly concluded from the maps that, in our case, more advantageous solution is a system without thermal coupling.
This allows the system to operate at lower temperatures. Elimination of the coupling can be achieved in a straightforward way in this case by locating the thyristors on separate heat sinks or by performing appropriate cut-out in the common heat sink.
It should be noted here, that there are power electronic systems with intentionally introduced thermal couplings in order to increase temperature to achieve required electrical parameter values. This is the case for some systems with power transistors in particular. An obvious benefit of such a design is the possibility to determine previously the temperature in the anticipated state of system operation.
The obtained simulation results can be verified by measurements performed on the physical system (at the prototype stage). However,this requires costly measurement devices and the need to assemble the system itself. In our example,measurements using an Inframetrics infrared camera, and also measurements using the HP Data Acquisition System confirm the modelling results with an error less than 8 % which is entirely sufficient for the purpose of thermal analysis in power electronics.

IV. CONCLUSIONS

The method presented, which uses small computing systems seems to be suitable for performing inexpensive thermal analysis of power electronic devices using the CAD techniques.
In this computation commercial programs NISA II PC and DISPLAY II,developed by Engineering Mechanics Research Corporation and an HP Vectra ES/12 with a 80286 microprocessor were used, operating at 12 MHz clock speed.
The total time for data input, calculations and the analysis of the results, presented here as an example, should not exceed 1 hour when performed by one person.
The PC version of the program for finite element analysis can be successfully used not only in power electronics, but also in thermal analysis of microelectronic systems, where the stile increasing concentration of electrical elements requires particular consideration of thermal behaviour.

REFERENCES

[1] A.Skorek,K.Al-Haddad,"Un nouveau standard dans l'analyse thermique de circuits en électronique de puissance", Canadian Conference on Electrical and Computer Engineering, Montréal,September 17-20,1989,Conference Proceedings, pp.256-298.
[2] V.Rajagopalan,"Computer-Aided Analysis of Power Electronic Systems",New York,Marcel Dekker,Inc.,1987.
[3] EMRC,"NISA II PC/DISPLAY II User's Manual",Troy,MI,1989.
[4] P.Leturcq,J.M.Dorkel,A.Napieralski,E.Lachiver, "A New Approach to Thermal Analysis of Power Electronic Devices", IEEE Trans.Electron Devices, Vol.ED-34,No 5,pp. 1147-1156, May,1987.
[5] G.N.Ellison,"Thermal Computation for Electronic Equipment",New York, Nostrand Reinhold Company Inc., 1984.
[6] Powerex Inc.,"Power Semiconductor Data Book" Pensylvania,1988.
[7] O.C.Zienkiewicz,"The Finite Element Method",London, McGraw-Hill,1979.

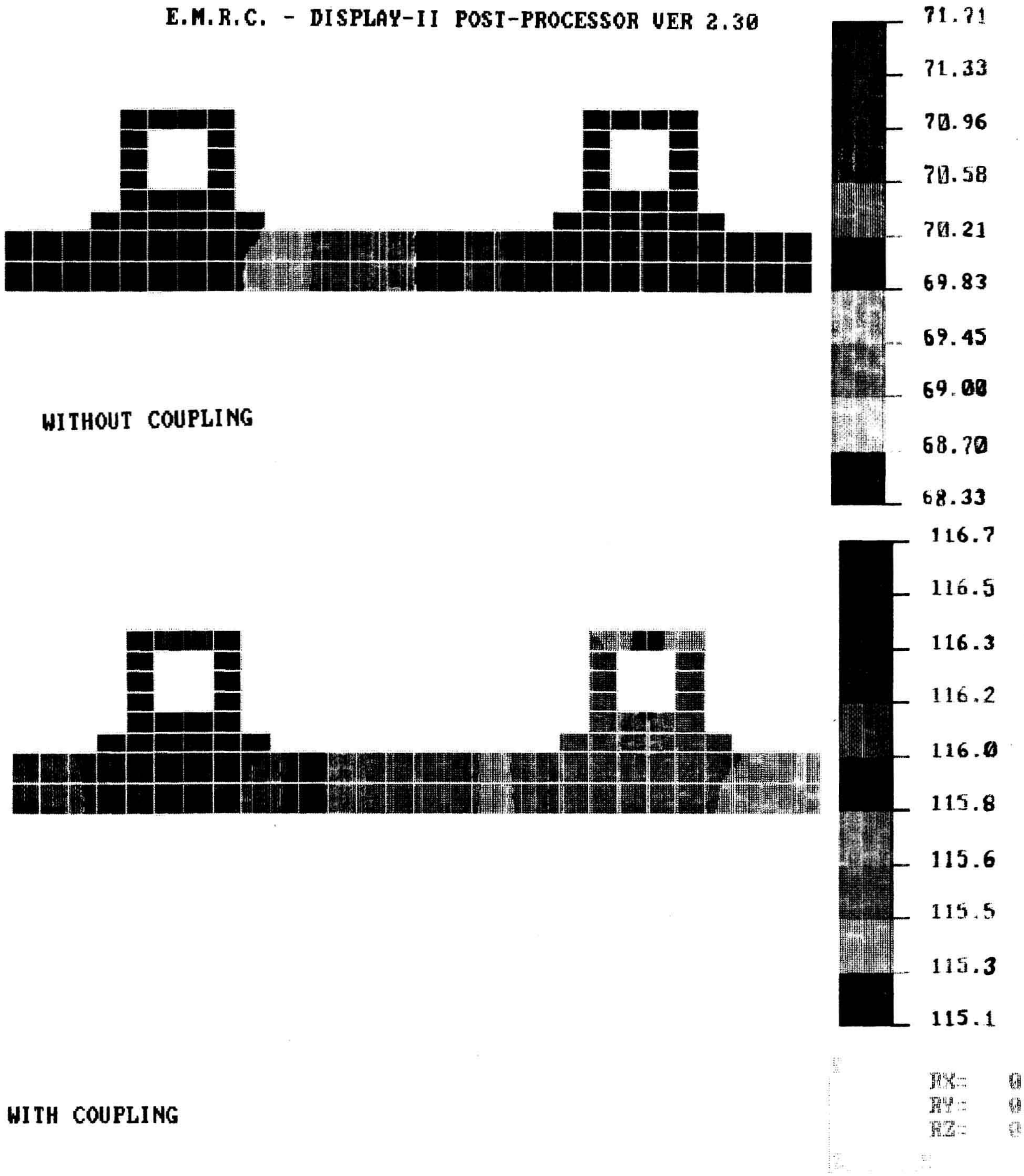

Figure 7: Temperature distribution

COMPUTER AIDED SPRING DESIGN

G H Lim
School of Mechanical & Production Engineering
Nanyang Technological Institute
Nanyang Avenue
Singapore 2263

ABSTRACT

The design of springs can be a time consuming activity involving many repeated calculations. With the advent of low cost micro-computers, such design function can be more effectively and efficiently undertaken by the computer.

A suite of computer programs, incorporating user-friendly features, has been developed with this in mind. In addition to the standard design considerations such as loads, stresses and tolerances, it incorporates facilities for selecting spring materials from a large material property database as well as evaluating various working environments in which it will be used in service.

Developed for use on the IBM PC/XT/AT or compatible computer, the program uses QUICKBASIC as the programming language. Some graphics have also been incorporated at the input data portion, which makes it easier for the user to identify readily as to what kind of spring data is required to be entered.

When all the necessary computations have been completed, a fully documented specification sheet is produced and may be used as part of an order form. This paper describes the overall design concept and the major factors considered in the software.

Keywords: Computer Aided Design, Spring Design, Programming, IBM PC/XT/AT, QUICKBASIC

INTRODUCTION

Springs are important engineering components used in many machines and mechanisms. However, they are frequently not given sufficient attention at the design stage. The main reason for this is that the design of a spring often depends on many interrelated variables and can be a very time consuming task.

With the advent of low cost microcomputers, such design function can be more effectively and efficiently undertaken by the computer. Software programs using microcomputers have been developed [1], but they did not appear to be comprehensive enough. For instance, reference [1] made use of the minimum weight spring design and assumed a normal working environment. It did not provide for the selection of materials which is an important aspect of spring design. It assumed that a material selection has already been made. However, it is not always a simple matter to make such a decision. There can be several conflicting requirements for the material and in such situations a best compromise must be adopted with detailed consideration of the relative importance of each requirement.

There is also a need to consider the working environment, as abnormal working environment will affect the performance and life of the spring. As a result factors such as spring relaxation, corrosion and pre-stressing should be taken into consideration.

The objective of this software development is to simplify the design process, improve the standard of spring design, incorporate the selection of spring materials and evaluate the different working environments in service so as to enable an optimum design to be produced.

DESIGN CONSIDERATIONS

To make the software self-contained and easier to debug, the program was split into smaller discrete modules, each fulfilling a particular design function. The main design classifications are as follows:

a) Functional Consideration

The two main functional considerations are service and load classifications. The former is concerned with the service life of the spring and its service criticality. Long service life is taken to mean a minimum service life of not less than one year whereas for service criticality, a spring is regarded to be critical if its failure is likely to cause considerable damage to equipment or human injuries. The latter is required for calculating the working stresses and is classified into three categories viz. static loading, intermediate loading and fatigue loading [2].

b) Material Selection

An important part of the software is concerned with the design selection of spring material. In fact, this part of the program is first performed before the design for load consideration. It is generally recognized that an ideal spring material should have high tensile strength, high elastic limit and low modulus. Among the factors considered are the stress range through which the spring operates, the desired load, mass and space limitations, the environment in which the spring will operate in service with respect to temperature or corrosive conditions and finally the cost. Careful selection will need to be made in order to arrive at the best compromise.

The four main groups of materials considered in the program are 1) Carbon Steels and Alloy Steels 2) Stainless Steels 3) Copper Alloys 4) Nickel Alloys. The tensile strength data of these materials are keyed into the computer data base.

Carbon Steels

Springs made from these materials have high working stresses and also they are of low cost.

Stainless Steels

Stainless steels wires should be used for springs when resistance to either corrosion, creep or relaxation at elevated temperatures are required.

Copper Alloys

These materials do not allow for high working stresses and are therefore recommended for light duty springs. They are non-magnetic and have high resistance to corrosion.

Nickel Alloys

These materials have many desired properties e.g. high resistance to creep or relaxation at high temperature etc. but they are expensive.

c) Design Consideration

The design consideration incorporates the types of spring ends, end fixation (which will affect, for example buckling of compression spring), mean coil diameter (used for calculating spring stress and deflection), wire diameter (which affects the spring tensile strength), spring index (ratio of mean coil diameter to wire diameter), free length, solid length, number of coils, and stress [2].

The tolerances for this spring design is taken from reference [3] and is applied to round wire springs. Two grades, grade one and two are recommended for economic production.

SOFTWARE DESIGN

The flow diagram for the program is shown in Figure 1. As can clearly be seen, the program comprises five major modules. These are (1) Functional Classification (2) Environmental Classification (3) Material Selection (4) Design Consideration (5) Load Consideration. User friendly menu displays are provided to lead the user through the program. Furthermore, at the input data portion, all entered data are checked, wherever possible, to detect any gross input errors.

Figure 2 shows part of the flow chart for design and load considerations. The design part will consider the type of end, type of end fixation for the spring and the pre-stressing requirement. Under load consideration, the maximum and minimum loads are required to be entered. User-friendly features include a help screen, where the user can call for assistance by typing "H" and hitting the RETURN key.

There are up to seven design modes which may be selected. Mode 1 is for no coil diameter is specified, Mode 2 to 5 are for specifying either outer or inner diameters, Mode 6 is for specifying both outer and inner diameters and Mode 7 is for specifying the free length.

The data files for wire sizes are in metric units and are those for the British Standard material specification. They link the tensile strength with standard wire sizes. This is to ensure that a spring is not designed using a non-standard wire size which cannot be obtained. Furthermore, since the tensile strength is read for each wire size, due allowance is made for the variation in strength with wire diameter which exists for spring wires.

A feature of the software is concerned with the relationship between the wire diameter with the mean coil diameter and the maximum load permissible. As mean coil diameter is calculated earlier, the relationship would be helpful in estimating the initial wire diameter. However, most of these data are in the form of tables and so it would be very time consuming to key in all these into the data files.

This problem was overcome by transforming the data and the affecting factors into equations form by the use of the polynomial least square method. To ensure a high

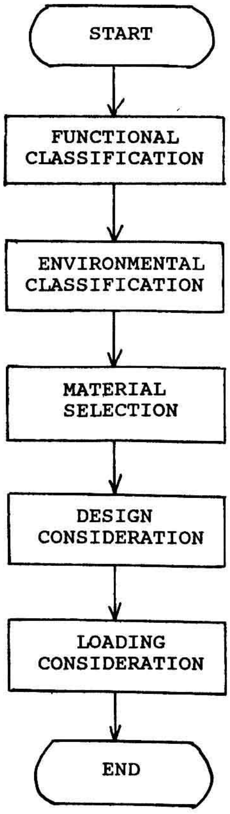

Figure 1 The Spring Design Flow Chart

degree of accuracy, a percentage of 99 goodness of fit was given for equations used to calculate wire diameter. But for coil diameter, a percentage of at least 90 goodness of fit was used. This is because in the case of the coil diameter, it was just an estimate unlike the wire diameter case where the stress calculated is being used as a reference.

SOFTWARE

The computer language used is BASIC. Although there are many versions of BASIC that have been developed in recent years, QuickBASIC version 4 [4, 5] has been chosen for this design.

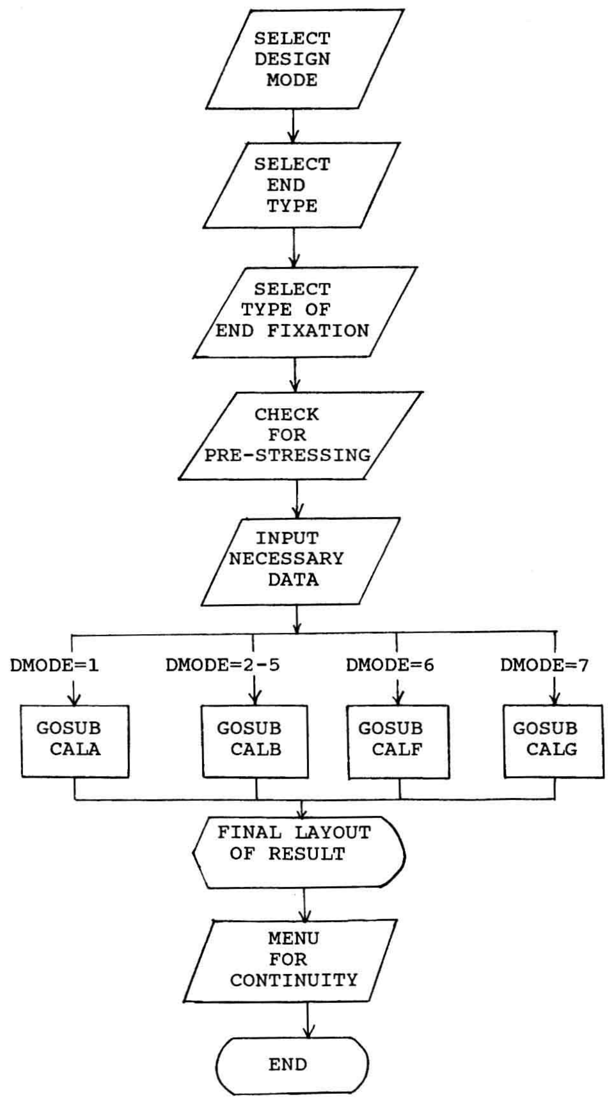

Figure 2 The Flow Chart for Design Consideration

The QuickBASIC compiler has a simple user interface and menu structure. Its design helps the software developer in developing programs with the least number of steps. It has an intelligent editor that checks syntax as the program lines are entered. An useful feature is that the program is compiled as it is typed, thereby allowing the program to be run in a single step. Also, errors are identified and reported in a way that allows the user to correct them immediately.

This program made use of the graphic facility at the input data entry portion, as provided by QuickBASIC. This makes it easier for the user to identify readily as to what kind of spring data is required to be entered. When the program testing and development is completed, it can be compiled into an .EXE file allowing the user to run them subsequently without using QuickBASIC.

DESIGN FOR OPTIMIZATION

The use of the computer is found to be most effective when design for optimization is required. Design optimization normally begins with the independent design variables, i.e. the quantities that the designer can directly specify such as material properties etc. These then generate the next level of variables, i.e. quantities that can only be predicted from given values of the independent design variables. They are called dependent design variables such as weight or cost, etc.

In arriving at an optimum spring design, the designer has to ensure that the required spring would meet the loads requirement, fit within the available space envelope and be safely stressed. It may be apparent from Figure 2 that these design programs are much more involved since they are solving an open-ended problem to produce an optimum design and they contain lengthy iterative routines in order to solve the complex polynomial expressions.

To the inexperienced designer, it is quite possible to over constrain a design such that no feasible solution is obtained. When the computer indicates this to be the case, then one or more of the input specifications may be relaxed e.g. the maximum load for the spring. It is clear that without the computer, such design optimization exercise would be a major task to undertake manually.

```
Material Selected     : Hard Drawn Carbon Steel
Specification         : SAE J113
Working Temperature : 0 °C to 100 °C
Max. Allowable Temp : 150 °C
Modulus of Rigidity : 79300 MPa
Plating Code          : Can be Pre-stressed
Material Code         : m1
Type of Loading       : Static Loading

End of material selection program.
Press 'C' or 'c' to continue for stress design.
Press 'E' or 'e' to exit from program.
```

Figure 3 Typical Printout for Material Selection Program

RESULTS

As mentioned earlier, the design program consists of two main parts, i.e. the material selection and the stress design programs. Figure 3 shows the typical output of the material selection and Figure 4 shows the complete spring design related to a compression spring. In the course of evaluating the results, manual calculations were carried out. It has been found that the computer aided design package yields accurate results. Furthermore, both outputs consume very little CPU time on the computer and it has been possible to obtain useful results without resorting to complex calculation procedures. A limitation of the program currently is that it does not incorporate impact loading.

CONCLUSIONS

The program has proved to be a useful aid to the designer by virtue of the speed and accuracy with which tedious design calculations are effected. The designer is then left free to concentrate on the fundamental design concepts and other more productive activities.

REFERENCES

[1]J. N. Siddall, Optimal Engineering Design, Marcel Dekker Inc., New York, 1982.

[2]H. Carlson, Spring Designer's Handbook, Marcel Dekker Inc., New York, 1978.

[3]British Standard B S 1716 : Part 1 : British Standards Institution, 1987.

[4]D. Inman and B. Albrecht, Using QuickBASIC, McGraw Hill, Berkeley, 1988.

[5]V. V. Hari, M. Tanik and U. W. Pooch, Illustrated Quick BASIC, Wordware Publishing Inc., Plano, 1989.

COMPRESSION SPRING SPECIFICATION CHECKLIST	
Material : Hard Drawn Carbon Steel Specification : SAE J113	Design No. : 3301 Date : 02/20/1990
WORKING ENVIRONMENT Maximum Temperature = 100 °C Minimum Temperature = 0 °C Magnetic Permeability : Not Important Work Environment : Inland Atmosphere Spring Rate = 10 N/mm	DESIGN CONSIDERATION Type Of Ends : Squared And Ground Load Frequency : < 100,000 cycles Pre-stressing : No Maximum Outer Diameter = 50 mm
LOADING ENVIRONMENT Minimum Load = 100 N Maximum Load = 300 N Working Delection = 20 mm	DESIGN DATA Wire Diameter = 3.55 mm Mean Coil Diameter = 40 mm Free Length = 48.83 mm Solid Height = 15.83 mm Total Number Of Coils = 4.5
FUNCTIONAL CLASSIFICATION : No load measurement	Tolerance Grade : 2 (Refer To BS 1726 : Part 1)

Figure 4 Typical Printout for Compression Spring Specification

Power Transmission Mechanism Recognition System

by
John R. Goulding
H. Zarefar

Portland State University
Department of Mechanical Engineering
P.O. Box 715, Portland, Oregon 97207

Abstract

The application of mini and microcomputer knowledge-based Computer Aided Design and Computer Aided Manufacturing (CAD/CAM) software to mechanical engineering design (drafting, simulating, and planing) is both technically feasible and economically profitable. When knowledge-based expert rules, equations, and proprietary languages extend CAD/CAM software, previously designed mechanisms can be *scaled* to satisfy new design requirements in the shortest time.

Two major drawbacks exist in current technology. First, embedded design alternatives needed by design engineers during product conception and rework stages are lacking. Second, an operator is required who has a thorough understanding of the intended design and the how-to expertise needed to create and optimize the design alternatives.

Our solution to eliminate the two drawbacks in a power transmission CAD software system was to develop a neural network shell. The shell automates the *intellectual* operations (questioning, identifying, selecting, etc.) of the design process. A robust system emerged which selects the *best* mechanisms necessary to implement a design, and aids the inexperienced operator in developing complex design solutions using decision-based criteria.

6 Key Words

Design automation, gearboxes, CAD, neural networks

Introduction

Engineers are often assigned projects which, to them, incorporate new or vaguely familiar design skills and practices. To gain the necessary design expertise, handbooks and the like must be consulted for design equations and rules of thumb. Generally, senior design engineers can design complex mechanical parts better than novice designers. Through their broader knowledge and experience, they can recognize similarities in known components of the overall design. This is a process of pattern recognition.

Mechanical engineering industries purchased first generation CAD/CAM software to reduce labor costs involved in product design. Second generation CAD/CAM software, recently introduced, incorporates knowledge-based expert rules, equations, and proprietary languages to *scale* previously designed products to satisfy new design requirements [1-3].

Two major drawbacks exist in current software. First, embedded design alternatives needed by engineers during the mechanism conception and rework stages are lacking. Second, the software operator needs a thorough understanding of the intended design and the how-to expertise needed to create and optimize the design alternatives.

Some experienced managers and supervisors argue that product background and design expertise can be learned only through years of practice [4]. If this is so, more can be done to ensure that the *intellectual* operations of the design process are automated. That is, the operations of design questioning, identifying (sub)components, selecting the *best* design, etc., are all decision based. The intent of this paper is to discuss an implementation of a Power Transmission Mechanism Recognition System (PTMRS) capable of identifying the *best* of 22 designs. Table 1 lists the 22 gearbox mechanism design alternatives.

Designing Power Transmission Mechanisms

For this research, the 22 different gearbox design solutions comprise the total set of design alternatives. One of the authors of this paper has developed a knowledge-based spur gearbox design system [5] which is capable of scaling gearbox system dimensions and generating AutoCAD draft files. Eleven fundamental requirements are necessary to scale a gearbox, as listed in Table 2. The fundamental requirements are not enough to select the *best* of 22 designs. Therefore, a literature search of gear handbooks [6-14] was performed to extract decision-based design criteria.

Table 1.	Table 2.	Table 3.
1. Bevel, Skewed 2. Bevel, Spiral 3. Bevel, Straight 4. Bevel, Zerol 5. Bevloid 6. Crown-Pinion 7. Face 8. Formate 9. Harmonic 10. Helical, Crossed 11. Helical, Double 12. Helical, Single 13. Helicon 14. Herringbone 15. Hypoid 16. Spiroid 17. Spur 18. Worm, Cavex 19. Worm, Cylindrical 20. Worm, Cone-Drive 21. Worm, 2-Enveloping 22. Worm, 1-Enveloping	1. Axes orientation 2. Relative direction of shaft rotation 3. Speed reduction 4. Input (pinion) speed 5. Ambient temperature 6. Application environment 7. Horsepower transmitted 8. Gear finishing process 9. Required life (cycles) 10. Input/Output shaft overhung length 11. Overhung load	171 Input Questions Overview DESIGN Environmental Functional Geometrical Mechanical CORPORATE Capital Labor Managerial Marketing MANUFACTURING Assembly Forming Generating Noncontrolled

Over 16,000 combinations of case-example designs, rules-of-thumb, and decision-based criteria were developed from the literature. Factors considered in designing the PTMRS included customer and corporate relationships in procedure, engineering analyses, sales and marketing strategies, facilities resource uses, and design criteria. One hundred seventy-one questions were developed to select the *best* of 22 designs. Table 3 illustrates how the 171 questions were grouped into categories of mechanism design, corporate practice, and manufacturing methods. The mechanism design input is controlled by the user who responds to 21 questions which have 84 possible results.

The overall objective of the PTMRS environment was to assist the user in designing the *best* of 22 types of gearboxes. Economically, the application to mini and microcomputers was desirable. Most important, the system had to provide expertise when human expertise wasn't available. Because product cost is locked in during the initial design phase, and because biased part specifications propagate through the design evaluation, modification, and analysis stages, the proposed system had to be robust and all-inclusive. Finally, the system had to be capable of arriving at solutions when design information was incomplete.

Recognition System Alternatives

Thus it was determined that a neural network (N-Net) shell would be developed to mimic the resources of design handbooks and the like. The N-Net implementation of the PTMRS is, foremost, a pattern recognizer, and will classify problems that can't be expressed as algorithms, searches, configuration matrices, calculations, or somatic networks. N-Net programs use mathematical relationships to mimic the subjective judgment of the expert designer, not the part, and perform iterative design-rule testing in a fraction of the time needed by similar data bases and expert systems [15].

Like fuzzy-logic programs, N-Nets are capable of statistical decision making with incomplete information. Further, fuzzy logic programs use a comprehensive language system that may have built-in biases, embedded goals, and hidden information structures which may result in errors [15]. For all of the above-mentioned reasons[1], it was determined that the PTMRS should be modeled using neural networks programmed[2] at the "software house."

---

[1]Fuzzy-logic inference engines were considered, but ruled out due to unavailability.

[2]NeuralWare's NeuralWorks PC software was chosen to train the N-Net, and was subsequently ported to the PTMRS system in the OPS83 PC environment.

The PTMRS was divided into three feed-forward back-propagation N-Nets which passed information using a 9-digit analog code defining the features of the 22 gearboxes. A back-propagation neural network simulates a parallel distributed processing system in which a matrix of artificial neurons, or nodes, is connected by variable weights. An input and output nodal layer, connected through several hidden middle layers, directly corresponds to predefined problems and solutions appearing as confidence levels. The weights are adjusted, or mapped, through an external training session using the back-propagation of errors method based on correct input and output pair examples [16]. A well-trained N-Net can generalize solutions to untrained problems because the embedded if-then relationships have been synthesized during the training process.

Conclusion

Through testing, the Power Transmission Mechanism Recognition System (PTMRS) demonstrated the viability of automating the *intellectual* operations of the design process In each test case, the PTMRS successfully indicated the *best* of 22 designs. When the overall system was used by novice engineers (senior design students in the mechanical engineering curriculum), several aspects became apparent. Designers who had little knowledge about mechanical power transmissions were able to identify the gearbox systems. Those who supplied ambiguous data were able to play what-if games and foster creative abilities. Finally, designers who used the system reduced the time necessary to develop mechanical power transmissions.

When the results of the PTMRS were passed to the knowledge-based Spur Gearbox Design System, the N-Net was shown to yield the correct results. An example of one of the four possible spur gearbox CAD generated designs[3] appears in Figure 1. The primary contribution of the PTMRS is to expand knowledge-based CAD software systems to make decision-based reasoning information accessible to the user throughout the engineering process.

Future work will interactively link the PTMRS with the knowledge-based CAD system. An iterative process, similar to the role of the traditional design supervisor, should emerge. Thus, back-to-the-drawing-board paradigms will be solved using decision-based criteria.

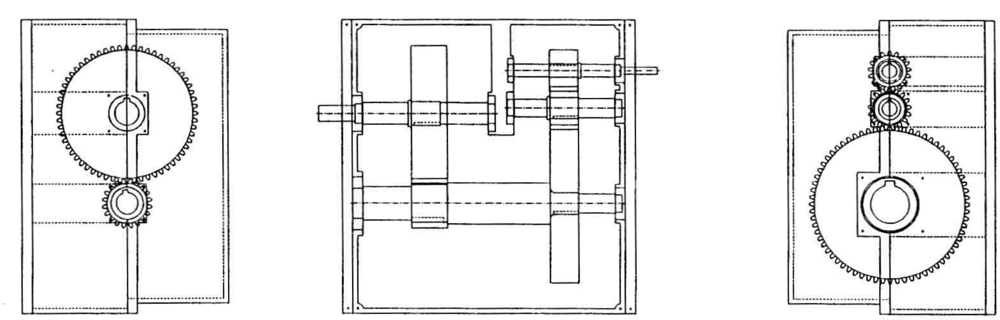

Figure 1. Sample of Spur Gearbox CAD Output.

[3]Dimensions and text were not included for the sake of clarity.