INTELLIGENT HYBRID APPROACH FOR INTEGRATED DESIGN

MARK WAKELAM

A report submitted in partial fulfilment of the
requirements of The Nottingham Trent University
for the degree of Doctor of Philosophy.

This research programme was carried out in the
Department of Mechanical and Manufacturing Engineering,
Faculty of Engineering and Computing,
The Nottingham Trent University,
Burton Street, Nottingham, UK.

October 1998

ABSTRACT

The process of Total Design consists of numerous stages, such as the formulation of product design specifications, development of conceptual designs, detail design and manufacture. Conducting a design throughout the entire process is tedious and time-consuming, due to the complexity of each stage, often requiring redesign. In order to reduce production costs and time-to-market, it is highly desirable to automate the design process, using modern artificial intelligence (AI) techniques.

An intelligent hybrid approach to integrate the stages of the Total Design process within a single environment has been developed. Integration has been achieved through a combination of rule based systems, artificial neural networks (ANNs) and genetic algorithms (GAs) with multi-media and CAD/CAE/CAM, providing a powerful tool for design automation. Both design integration and application of AI in engineering are currently attractive research topics, with a number of successful applications. However, the integration of multi AI techniques with CAD/CAE/CAM for Total Design has never been reported, hence, this project is novel research.

The Total Design process has been evaluated with regard to identifying stages and decision making processes required to generate successful designs. The results of the evaluation are formulated, considering the methods of knowledge representation with an emphasis on a modular structure, forming the intelligent hybrid approach. Several methodologies have been developed within the intelligent hybrid approach including: design evaluation and knowledge acquisition, knowledge encapsulation, AI integration, system structure, adaptive design selection/retrieval, GA optimisation and ANN training.

These methodologies combine to form an intelligent integrated system (IIS). An IIS for the design of mechanical power transmissions has been used as an application to help develop and validate the approach. The conceptual design stage combines a rule base with a series of ANNs to generate the conceptual arrangement and method of transmission between shafts. The detail design stage takes particular advantage of the modular structure that is encouraged within the IIS to breakdown the design process. Design modules relating to individual component designs interact with each other, successfully applying AI for decision making, information manipulation and design optimisation, using the single environment to combine and exploit the strengths of different techniques.

The use of backpropagation ANNs provoked an investigation into the training process. The conclusion from which indicated that no general rule exists to determine the training parameters that create high performance networks. The necessity for a method of simplifying training led to the integration of a GA to the training process which, adaptively alters training parameters, improving network performance irrespective of the application.

The application of GAs to design optimisation has proved very effective at emulating expertise. The technique enables high quality designs to be tailored to applications without extensive knowledge of the particular field. Additionally, an investigation into the evolutionary process has overcome the traditional GA problems of computational expense and repeatability of results, enabling their inclusion in the approach.

ACKNOWLEDGEMENTS

I have received a tremendous amount of support from colleagues at The Nottingham Trent University, during the course of my research work, for which I am sincerely grateful. There are, however, certain persons who deserve a special mention; I would like to thank my Director of Studies, Dr Daizhong Su, for his continuous support, encouragement, guidance and for having faith in me throughout this project. Sincere thanks are due to my supervisors; Dr Keith Henthorn and Dr Nath Jambunathan for their advice, comments, backing and interest in my work.

Special thanks to my fellow researchers and colleagues; Dr Stephan Ball, Mr Uwe Ruedel, Mr Rob Goodson, Dr Paul Wilson, Mr Mark Aldred and Mr Neil Templeton for their friendship, advice, encouragement, sarcasm and gibberish.

Finally, my greatest thanks to my parents, Jim and Doris, for their unequivocal love and support and to my brother, Steven 'Our Kid', for his advice, help and friendship.

DEDICATION

To my friend and father, Jim.

PUBLICATIONS ARISING FROM THIS WORK

1. Su, D., Henthorn, K., Jambunathan, K., and Wakelam, M., 1996, A Rule Based and Artificial Neural Network System for Conceptual Design. *International Journal of INGENIUM 96,* No. 1, pp 1-9.
2. Wakelam, M. and Su. D., 1996, Knowledge Representation Incorporating Artificial Neural Networks for Design of a Mechanical Transmission System. *Proceedings, IEE Colloquium on Knowledge Discovery and Data Mining*, 17th–18th October, Vol. 2, London, pp 3/1-3/3
3. Jambunathan, K., Wakelam, M., Henthorn, K. and Su, D., 1996, Integration of Multi-Media, Artificial Neural Networks and Rule Base Systems for Gear Design, *Proceedings, International Conference on Theory and Practice of Gearing*, 4th–6th December, Izhevsk, Russia, pp 463-468.
4. Su, D. and Wakelam, M., 1997, Intelligent Integrated System for the Design of power Transmission Systems, *Proceedings, International Conference on Mechanical Transmissions and Mechanisms*, 1st-4th July, Tianjin, China, pp. 1010-1014.
5. Henthorn, K., Wakelam, M., Jambunathan, K. and Su, D., 1997, Application of Artificial Intelligence into the Design and Manufacture of Mechanical Power Transmission Systems, *Proceedings, 13th National Conference on Manufacturing Research*, 9th-11th September, Glasgow, pp 596-601.
6. Wakelam, M., Su, D., Henthorn, K. and Jambunathan, K., 1998, Development Strategies for Intelligent Design Integration, *Proceedings, International Conference on Mechanics in Design*, 6th-9th July, Nottingham, pp 857-846.
7. Wakelam, M., Su, D., Jambunathan, K. and Henthorn, K., 1998, Genetic Optimisation of Spur and Helical Gears, *Proceedings, International Conference on Mechanics in Design*, 6th-9th July, Nottingham, pp 129-138. (Awarded best presentation)
8. Wakelam, M., Su, D., Jambunathan, K. and Henthorn, K., 1998, Evolutionary Aid for Training Backpropagation Neural Networks, *Journal of Computing and Information Technology – Special Issue on Evolutionary Computing*, (Submitted 2nd July 1998).
9. Su, D., Wakelam, M., Henthorn, K. and Jambunathan, K., A Software package for Evolutionary Optimisation of Gear Design, *Evolutionary Design Using Computers*, Academic Press.

GLOSSARY OF TERMS

ANN	Artificial Neural Network
BP	Backpropagation
EP	Evolutionary Programming
ES	Expert System
GA	Genetic Algorithm
GUI	Graphical User Interface
KBS	Knowledge Based System
HIS	Intelligent Hybrid System
IIS	Intelligent Integrated System
PDS	Product Design Specification
GEN-NEU	GENetic Algorithm Training Program for Backpropagation NEUral Networks
OPTGEAR	Spur and Helical gear Optimisation Program

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

This chapter gives a thorough introduction for the research undertaken for this project. The background to the research is explained, followed by a detailed statement of the project's aim and objectives. A brief summary of the chapters contained within this document concludes this chapter. The chapter summaries highlight significant discoveries and achievements attained within the course of the research investigation. These achievements are explained relative to their respective applications.

1.1 Background

Total Design, as defined by Pugh (1990), 'is the systematic activity necessary, from the identification of the market / user need, to the selling of the product to satisfy that need'. Within Total Design exists a central core, illustrated in Figure 1.1, which envelops the products design stages and encourages interaction to optimise design performance.

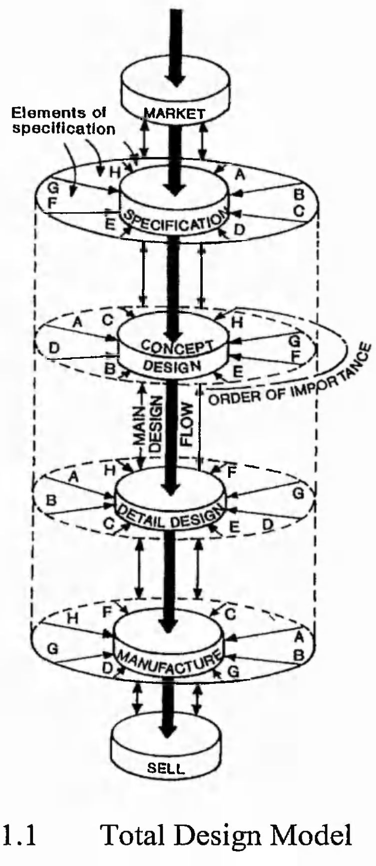


Figure 1.1 Total Design Model

The major elements within Total Design are developed within this project, including:

- Specification: formulation of product design specifications.
- Conceptual design: concept generation and evaluation.
- Detail design: analysis, sub-assembly/ component design, assembly, engineering drawing and design retrieval.
- Manufacture: the manufacturing processes require to produce a product have been considered throughout the design approach developed within this project. However, the implementation of this Total Design element has not been implemented as it is beyond the scope of the project, but is considered as further work.

Due to the number of stages a design must pass through and the information and expertise that is required for each, the entire process can be laborious and time consuming. Therefore, redesign if required toward the final stages of the design process can be costly if performed manually. Since the introduction of computers in the 1940s, engineers have been applying their rapid computational qualities to all aspects of engineering, including design. The large amounts of information required by the individual stages of the design can be stored and rapidly retrieved and manipulated, giving a speedy response. However, the application of computers to design has generally concentrated on the reduction of time taken to perform individual design tasks. Interactions between the tasks are generally not considered together with their effects upon the form of the design stages.

Integration of the design tasks and stages to form a single complete design system appears to be the solution as design problems and redesign can be achieved relatively quickly and cheaply. A few integrated design systems have been developed (these will be discussed in the literature review), however, due to the shear diversity of design they have been limited to particular design areas.

Increasing research into the development of design systems has lead to the incorporation of Artificial Intelligence (AI) to emulate the human decision making process. As the design process is not just a series of calculations and drawings, but also the application of experience to the selection and generation of solutions, the application of AI within the system will increase performance and versatility.

AI has been implemented into the design process in the form of Expert Systems (ES), also referred to as knowledge-based systems (KBS), successfully, although displaying limitations with regard to flexibility. Additionally the use of other AI techniques such as Artificial Neural Networks (ANNs) and genetic algorithms (GAs) has not been extensively investigated. Advantages that these techniques may offer and methods of integrating them into the design process need to be explored further to develop their full potential. Therefore, a system that applies a blend of state of the art AI techniques for design would result in an intelligent design system to increase the efficiency of design.

1.2 Aim and Objectives

The aim of this project is to study the Total Design process with regard to developing an integrated approach to encompass the complete process, from the formulation of product design specification to the point of manufacture. To increase the performance and capabilities of the system, an intelligent aspect is to be introduced to emulate the expertise and knowledge that expert designer's utilise throughout a designs development. Intelligence will be installed in the system by a blend of proven and developing AI techniques.

An intelligent integrated system (IIS) will be developed that combines various forms of AI with numerical calculation functions, databases and multi-media techniques together with existing design packages to develop a single hybrid system. The IIS will be applied to the design of mechanical power transmissions. The application will form a development test case to determine the characteristics of the various techniques when applied to different aspects within the Total Design process.

To achieve these goals a number of specific objectives are require to be met, including:

- review current work and associated studies relating to Total Design and is incorporation into computer aided systems.
- develop computerised methodology for Total Design into the intelligent hybrid approach.
- develop software integration strategy for combining AI techniques, with existing CAD / CAE commercial packages.
- incorporate ANNs into conceptual and detailed design stages.
- develop design optimisation strategy using GAs.

- develop design retrieval method incorporating ANNs for component design.
- implement a parametric design approach to generate component and assembly drawings using CAD/ CAM package (ProEngineer).
- apply the intelligent hybrid approach to develop an IIS for the design of a mechanical power transmission.

1.3 Outline of Project

Chapter 2 Literature Review

Research and literature relating to the major topics of the project are investigated and reviewed. The topics to be reviewed are design methodologies and artificial intelligence. Manual design methods are reviewed with respect to adapting, modifying and incorporating their principles into a computerised process for product designs of a similar calibre. The AI techniques ESs, ANNs and EP are investigated to identify their capabilities with respect to knowledge representation and their current uses in the field of engineering design.

Chapter 3 Artificial Intelligence Techniques and Multimedia Developed for the Intelligent Integrated System

Within this chapter the AI techniques used within the project are discussed together with methods of application and investigations conducted for knowledge representation.
Evolutionary Programming. The use of EP within the IIS is described, illustrating the method in information encoding within the GAs Genomes and a unique cascade optimisation application. Additionally the formation of the fitness functions is described together with the types of limitations that are imposed on the feasible search area.
Artificial Neural Networks The training of Backpropagation networks is investigated to determine if a series of training guides or rules of thumb could be determined. A series of tests have been conducted, the results from which are analysed and a series of rules generated. However, the guides are not conclusive. This lead to the development of the GEN-NEU training aid. GEN-NEU is a program that incorporates a GA and applies it to the modification of the major parameters that influence the performance of the networks.

The characteristics of the GEN-NEU program are described together with test results for an application to the design of gears.

Rule Based Systems. The type of rule base implemented within the IIS is discussed. The IIS utilises rules in the form of Production Rules. This section describes the forms of the Production Rules and the types of rule interpretation, (goal or data driven). The application of the rules to the IIS is described illustrating that the rule base is dispersed, localising on the area of application instead of in a central rule base.

Multimedia. The application of multimedia and the development of their application are described. Providing a general method of developing multimedia interfaces for inclusion within the IIS.

Chapter 4 Intelligent Hybrid Approach to Design and Intelligent Integrated System Development

The concept of the intelligent hybrid approach is defined, providing an overview of the approach and the structure for the IIS. The various concepts that build up the approach are explained, describing the development process for the IIS with regard to the process of structuring the system and knowledge contained within. The intelligent hybrid approach to design relates to the intended application, the combination of knowledge representation techniques and the structured development of the IIS. The processes and factors to be considered when developing the IIS are described within this chapter.

The method of optimising component designs using the EP approach is briefly explained, defining the function that this process will perform and how it is implemented within the IIS. The use of the ANNs for design retrieval is explained. Illustrating the method of implementing it within the IIS and the training process. The training process is described in detail within this section, defining how the training information is accumulated during the initial use of the system direct from the user/designer via a multimedia interface and trained using the GEN-NEU training program. This approach has not been reported for design retrieval or the use of backpropagation networks.

Chapter 5 Intelligent Integrated System for the Design of Mechanical Power Transmission Systems

The development of the IIS for the design of mechanical power transmission systems is described within this chapter. A detailed description of a mechanical power transmission system is given together with knowledge and information relating to design. The iterative control process is described, defining the sequence of stages that the design will progress through as well as the redesign routes. This process is controlled by the system controller.

Detailed information relating the integration and use of AI techniques within the design process is discussed, defining their applications with regard to the IIS. It is within this chapter that the hybrid system is illustrated. Combining all the stagse of design within the single environment, enabling direct communication between linked tasks.

Within this chapter the unique application of the EP for component optimisation is demonstrated. The optimisation of gears provides the initial test case. Detailed information and explanations are provided, defining the process of developing the gear application and test results for several case studies.

Additionally within the chapter backpropagation networks are applied to the retrieval and selection of two types of components for the transmission system, gears and bearings. The development of the graphical user interface is explained, basing the selection criteria presented to the user upon opinions from experts within these areas.

Chapter 6 Validation of the Mechanical Power Transmission System Application

The IIS for the design of mechanical power transmission systems is evaluated using an industrial application. The outcome from the IIS is compared with an existing design produced using an expert system, thus providing an assessment of the IIS performance and feasibility of combining AI techniques.

Chapter 7 Discussion and Conclusions

A summary of the conclusions drawn throughout the project is contained within this chapter. Commenting upon the intelligent hybrid approach, the application of the IIS to the mechanical power transmission system and the use of the various AI and multimedia techniques employed within the system.

Chapter 8 Further Work

This project forms the outlines the IIS approach to engineering product design and has implemented it to the design of mechanical power transmission systems. This chapter discusses the further development of the approach and the mechanical power transmission application. The chapter mentions the further improvements to the system that would increase its potential as a design system, but concentrates on the progression of the approach and the development of techniques for the manufacture of the final design. The various areas of manufacture are discussed together AI techniques that have been applied in these areas.

CHAPTER 2

LITERATURE REVIEW

A review of current literature relating to the project has indicated that there is increasing interest in the development of intelligent systems in the field of design to aid the reduction of development and production time, although the topic is still in its infancy. The papers and works that form the review create a general overview of work relating to design methodologies and the implementation of AI into engineering, concentrating upon its applications toward design.

The structure of the review will consist of a review of relevant works relating to each topic followed by comments relating to the works with respect to the project. Finally, conclusions are drawn in the last section which will highlight the necessity and innovative features which will be incorporated into this project.

2.1 Design Methodologies

Engineering design is an important part within the route of product development and production. Many researchers have made efforts to investigate the activities involved and to develop methodologies, which are studies of principles, practices and procedures of design in a broad and general sense, guiding the designers to conduct systematic and effective designs. Amongst the existing methodologies those developed by Hubka and Eder (1996), Pugh (1990), Archer (1965) and Pahl and Beitz (1988) are well recognised and used in practice by both researchers and designers.

Pugh developed the total design approach which included; investigation of market needs, formulation of product design specifications (PDS), concept generation and evaluation, detail design, manufacture and sale. The procedures relating to PDS formulation and concept evaluation are particularly useful and are frequently cited by other researchers.

The methods developed by Pahl and Beitz concentrate more upon mechanical design unlike the general approaches of Pugh, Archer and those described by Cross (1984, 1990). Pahl and Beitz's methodologies add an 'embodiment design' stage, which is located between the conceptual and detail design stages and is more suitable for the complicated design methods relating to information flow and data exchange between stages.

Cross (1984, 1990) has produced works defining the popular design methodologies, principles and practices. These works cover a range of methodologies ranging from methods for systematic design by Jones, which covers classification of factors to clarify disorderly information, synthesis of concepts and evaluation, to complex models such as that by Pahl and Beitz and the VDI 2221, which define interaction and re-evaluation paths between stages.

It must be pointed out however, that the methods and models discussed so far have been developed for manual design purposes rather than for a computer aided process, although some of the concepts are applicable after modification. The computer aided process incorporates the essence of the manual methods and models due to the generic nature of design. Starting with the development of a concept to meet a need, which progresses and develops into the final refined product. However, the means by which the concepts will be generated, evaluated and refined are of a different form.

Engineering design involves complicated activities, amongst which decision making plays an important role. The quality of the design is dependant upon the designer's knowledge and experience. In order to speed up the whole design process and achieve a high quality of design it is desirable to adapt and implement the design process into a computerised system. The computerised system must be capable of emulating and manipulating the expert designer's domain knowledge to achieve a high level of design quality. The system must therefore, possess the capabilities to perform informed decision making for a variety of situations, judgement and evaluation, historical reference to similar designs, etc. Traditional computer-aided design techniques such as numerical analysis, databases, and computer aided drafting provide facilities to aid the design process but cannot emulate the designer's decision and reasoning process.

2.2 Artificial Intelligence

The concept of incorporating AI into the design process, to emulate the human thought process for decision making, has been a facility that engineers have desired for many years. The modern concept of AI was developed at approximately the same time as the von Neumann computer and is regarded, in several texts, as a relatively new branch of computing science. Although it is a relatively young discipline it has developed rapidly with the majority of the advances made within the last 30 years. The subject has attracted the interest and attention of many researchers from a wide variety of fields and core disciplines. However, the wide variety of applications AI has been applied to, coupled with the young age of the subject has lead to several definitions of AI being developed; below are just a few extracted from the Internet, the most up to date source of information:

1. *AI is a field of study concerned with the development and use of computer systems that have some resemblance to human intelligence, including such operations as natural-language recognition and use, problem solving, selection from alternatives, pattern recognition, generalization based on experience, and analysis of novel situations.* (http://www.europe.apnet.com/inscight/11151996/artificial-intelligence.htm)

2. *AI is the branch of computer science concerned with making computers behave like humans.* (http://www.pcwebopedia.com/artificial_intelligence.htm)

3. *AI is a multi-disciplinary field encompassing computer science, neuroscience, philosophy, psychology, robotics, and linguistics; and devoted to the reproduction of the methods or results of human reasoning and brain activity.* (http://www.webcom.com/~bsmart/aidef.html)

4. *AI is the design of a computer system that exhibits characteristics we associate with human behavior: understanding language, reasoning ,solving problems, learning and planning (scheduling).* (http://phoenix.liunet.edu/~edelson/kbs03.html)

5. *AI is the study of mental faculties through the use of computational model, eg, the brain.* (http://phoenix.liunet.edu/~edelson/kbs03.html)

6. *Artificial intelligence (AI) concerns making machines behave intelligently, i.e. to achieve goals requiring intelligence to solve them. It is better to accept the*

partial circularity of the above definition than to attempt to define intelligence. (http://www-formal.stanford.edu/jmc/aiintro/aiintro.html)

7. *AI can be described as the attempt to build machines that think and act like humans, that are able to learn and to use their knowledge to solve problems on their own.* (http://central.netaxs.com/~charvey/artificial.html)

The definition of AI varies depending upon the perspective of the application. From the point of view of this project, the use of AI is to emulate the decision making and creativity of a expert designer. Therefore, the term AI, within this project, will refer to

The emulation of intelligent behaviour for autonomously solving problems through the use of computational processes.

The main AI techniques developed which enable a computer to emulate the way a human thinks are Expert Systems (ES) and Artificial Neural Networks (ANNs). They will be reviewed in the following sections. In addition to ESs and ANNs, Genetic Algorithms (GAs) are another branch of AI. Based on the natural evolutionary process the GA is an adaptive search technique for determining optimums and design refinement.

2.3 Knowledge Based Systems / Expert Systems

Expert System (ES) were first development at Stanford University to help diagnose infectious blood diseases (1984) and consisted of three main parts: explanation generator and user interface, inference engine and a knowledge base (KB). This remains the basic structure even today. The KB consisted of medical rules in the form of IF *symptom* THEN *condition* statements with an associated confidence factor. Selection of the rules were performed by the inference engine. Since the first application of ESs they have been applied to a multitude of problems where a computer is required to emulate human decision making from banking to process control. The definition of an ES varies from source to source and is often interchangeable with Knowledge Based Systems (KBS). Some practitioners of ES and KBS argue that there is a difference, but for the purpose of this review and project they will be considered interchangeable and the same.

ES have been applied successfully to the majority of engineering topics, including design, for many years. The cases identified in the review give an insight into their application in engineering and for design in particular.

2.3.1 Expert Systems for Design Knowledge Processing

A KBS has been employed by Swift and Doyle (1987) for 'design for assembly'. This system analyses designs based upon the features such as fitting, manufacturing and functionality to determine possible problems upon assembly before the design goes into production. The KB contains rules that help the designer constrain the design within the realms of possibility; for example checking that gripping faces will be available for manufacture. If a possible problem is identified, design advice from the KB is relayed to the designer for reconsideration and redesign.

Woodward and Corbett (1990) similarly devised a system using an expert system to help in the design of components to be manufactured by aluminum die casting. A feature orientated approach for the design for manufacture of complex components with regard to the type of shapes that can be achieved by die casting. The ES draws upon expertise contained within the KB to determine if the features of the design may be produced, if draft angles are correct and identify if costly moving cores are required for unnecessary undercuts. These applications demonstrate how ES have been used for the analysis of an existing design.

Pandiarajan and Dwivedi (1993) have applied ES to another area of engineering, the process planning of components. They have integrated ES with a CAD system, process library, tool library and sequence optimiser to bridge the gap between design and manufacture. A modular approach is taken to separate the different stages of the planning process which is controlled by the ES. Control of the process is not the main application the ES is used for, its main purpose is to identify possible operation procedures based upon the CAD models which are passed to the sequence optimiser. The process of identifying procedures is the area of the planning process that traditionally requires expertise, which has been translated into heuristics and degrees of confidence, forming the Knowledge Base.

2.3.2 Knowledge Based Systems for Design Integration

Su and Forgie (1989) and later Su (1993) have addressed the problem of integrating the design procedures and implementing AI to form an integrated intelligent system. Su and Forgie developed an approach to computerise the design process devised by Pugh (1990) from specification to detailed design. The structure of the computerised system consisted of five basic elements. Firstly an inference engine is used to control the process and a blackboard is used to transfer of information between modules. A designer's experience and expertise which is captured within a knowledge base. A numerical calculation facilities which determine design parameters such as dimensions and stress levels, A data base containing general information relating to the design and materials and finally, a facility to generate engineering drawings form the output of the final design.

The approach consisted of breaking the process down into nine modules which are performed by one or more of the structural elements; product design specification (PDS), generation and identification, criteria generation, concept generation, rating mark generation, concept evaluation, design parameters and initial layout, component design, design co-ordination and final layout. These modules are then grouped according to the stage in the design process that they act which in turn are constructed into a hierarchy to represent the design sequence. Interaction, both back and forth, between the groups is allowed to optimise the design but only between adjacent levels of the hierarchy.

This approach was developed by Su (1994) into an intelligent system which used the design of a transmission system as a test case. The system performs the design process with the minimum interaction with the user generating concepts and evaluating them based upon the user's PDS and the knowledge base. If upon evaluation of the concepts a problem occurs the PDS or concepts are modified by the user, guided by knowledge retrieved from the system. Once a design is successfully evaluated a detailed analysis is performed before final drawings are produced..

Kaftanoglu, Ulugul and Carkoglu (1995) have recently developed a similar system to Su again applying it to transmission systems, but on a smaller scale. A design approach or strategy as they term it has been developed which again divides the whole design system into sub systems, but additionally applies a hierarchical approach to the analysis of the problem. The sub-systems are divided into modules, each module tackling a section of the design process. These modules are given priority levels to determine the importance of the section to the overall design. The shafts of a transmission system are given as an example. The shaft is given a high priority level and forms the backbone of the design. If the shaft changes all of a lower priority leveled also will change to compensate. A final design is established when there is no change to the shaft. Once the final design is achieved the system links with a CAD package to create the drawings and also produces the data sheets about the components. The main drawback with the system appears to be the large amount of information required from the designer to produce the final design.

Additionally Sharpe (1995) of Lancaster University Engineering Design Centre is a member of a team developing an integrated approach to the conceptual design process. Sharpe appears to be integrating the stages of design into a single system comprising of smaller modules whose combination cover the design process. Sharp uses a computer-aided design tool, called Schemebuilder, which provides an environment where schemes or concepts may be generated in a short time together with rough costs, weights and basic dimensions. Additional projects are discussed in this paper covering the implementation of knowledge systems for conceptual design, a quantitative development schemes from first principles, generic function and component database for conceptual design, configuration layout of schemes function mapping and multiple-criteria decision making and a module for mechatronic product design. Each project is intended to interact with one another through the Schemebuilder platform in an interactive approach. Industrial collaboration has been established with Sharps work with a number of major companies indicating that integrated design systems are required and relevant and that trust in AI is increasing and now desirable.

Two of the modules currently added to the Schemebuilder project are for mechatronic systems and hydraulic systems. The hydraulic module, as discussed by Silva and Dawson (1997) and Silva and Cheung (1997). The form of the module to the Schemebuilder platform is described giving a clearer insight into the methodology that is used to develop modules. The structure of the design process is highly object-orientated, allowing the design to be built up from sub-functions, which contain certain attributes that describe the design. The second project for the design of mechatronic systems, discussed by Counsell (1997) describes how similar principles to those used by Silva will be applied to conceptual design of servomechanisms.

The review of KBSs indicates that they are a powerful technique for handling and interpreting knowledge for many applications including the design process and some of the existing methods they employ may be usefully implemented within this project. However, weaknesses of the technique have also been revealed such as :

- Alteration of the knowledge base is almost impossible by the user.
- In the majority of applications, commercial expert system shells are used which provides a useful medium in which to develop the system. However, within the shell numerous facilities are offered which may not be necessary for the application, but contribute to the cost of the shell and the memory it occupies which may lead to problems for large integrated systems.

From the literature, there has been no application of KBSs for the whole process from design to manufacture which incorporates concurrent engineering. The applications found that address the integration stages within the design process have been limited by their software techniques and packages, which are now out of date. For example in Su, (1993 & 1994), AutoCAD version 10 was used. Integration of current advanced and more powerful software such as ProEngineer will offer increased possibilities.

With regard to the project a simple production rule system may present a more suitable solution than the larger more comprehensive expert shells.

2.4 Artificial Neural Networks

The systems discussed so far have covered the design process, but only from the ES or KBS point of view for decision making and knowledge representation. The disadvantage of knowledge based systems is that the knowledge they contain is not flexible to alter after the system is completed and is almost impossible to alter by anyone other than the system designer. In order to overcome this problem researchers have been seeking an alternative AI technique for decision making. Artificial neural networks offer an alternative method of capturing knowledge to KBS and provides a means of overcoming their inflexibility. The knowledge stored within an ANN system can be changed by training the system using different training data. This provides a solution to change the knowledge without altering the structure of the system itself. Research into ANNs commenced at about the same time as the computer with work by McCulloch and Pitts (1943), Hebb (1949) Rosenblatt (1959) and others. However, after a condemning paper by Minsky and

Papert (1969) and the lack of resources, interest began to fall, but due to continued work by researchers in developing new algorithms. Hopfield (1984) and Rumelhart (1986) were amongst others developing these algorithms and with advances in computer simulation of the networks using software was made easier resulting in ANNs re-emerging in the 1970s. Interest in ANNs has increased dramatically since this point and is continuing to increase with new applications in a wide variety of areas being found all the time.

ANNs have been applied to engineering for a number of years since the 1970s but the majority of applications have been since the mid 80s. Lippmanns paper (1987) and Hush and Hornes expansion (1993) have proven an excellent source of information on the different ANNs, their properties and their relevant applications. They have been applied to a number of cases in engineering areas from condition monitoring Lapdes and Farber (1987) to finite element analysis Takeuchi and Kosugi (1994).

2.4.1 Artificial Neural Networks for Engineering Design

ANNs have been applied to the majority of engineering disciplines in one form or another, however, this review will concentrate on the areas of engineering that relate to the design process, conceptual design, detail design, process planning and costing.

To understand the capabilities of ANNs their basic principles and methods of processing information need to be understood. An explanation of each of the popular ANN techniques can be obtained from many sources. (general papers and books such as Lippmann (1987), and Beale and Jackson (1990) and the Internet). A general appreciation of some of the areas ANNs have been applied to may provide an approach to similar structured problems. Wu (1992) has realised this and examines the structures and functions of ANNs and provides some examples of their manufacturing applications. The paper concentrates on a multilayer perceptron incorporating the back propagation training technique before giving examples for tool condition monitoring and cellular formation. Despite a few minor mistakes the paper provides an excellent basis to build experience with ANNs.

A general review by Udo and Gupta (1994) presents applications of ANNs to manufacturing management. Giving examples of applications for resource allocation and constraints satisfaction, scheduling, maintenance and repairs, process control and planning database management, simulation, robotics control, quality control and machine vision, together with the types of network used. It can be seen from the study that the structuring of the application is most important to the performance and the type of ANN used as different ANNs possesses different strengths and weaknesses.

2.4.2 Applications of Different ANNs to Engineering Design

ANNs can be divided into two broad categories, supervised and unsupervised, each with their strengths and weaknesses. Of all the ANNs two types of ANNs have been selected, one from each category, that posses the majority of the qualities that are expected to be required in the project.

Supervised (Multilayer perceptron)

Shtud and Zimerman (1993) used an ANN to estimate the cost of an assembly system. Selection of the most appropriate assembly system is an important activity in the life cycle of many products and is usually based on a cost-benefit analysis. This analysis requires engineering expertise for the selection of the best assembly system. The assembly systems are classified with respect to six proposed factors and the cost estimate equations that together provide the training data for the networks. A feed-forward network is selected to perform the analysis and a common problem of network structure (topology) is highlighted.

The general impression from the papers relating to feed-forward networks such as MLPs, it appears that identifying the correct topology can prove difficult as there are no set rules to indicate number of layers and elements within each layer. Shtud and Zimerman determined the network topology by trial and comparison, selecting the best topology from several attempts. Analysis of the trained networks performance indicated that it outperformed the traditional regression model showing that they can effectively represent both linear and non-linear models and represent data without constructing equations.

Unsupervised (Adaptive resonance theory)

One of the areas unsupervised networks have been applied to is Group Technology (GT). GT is an approach to manufacturing that attempts to enhance production efficiency by grouping tasks together. Escobedo, Smith and Caudell (1993) have applied an unsupervised adaptive resonance theory (ART) network to the group technology approach. Due to the classification capabilities of ART, they have been applied to the grouping of similar parts. ART has been used due to its ability to create new classifications if an unfamiliar part is represented to the network. This approach is effective for classification provided the number of classes in not restricted or defined.

Family forming can be applied to conceptual design with respect to designers using associate memory to retrieve similar designs. Kumara and Kamarthi (1992) have used ART networks to mimic this human action. Families of design problems are created by the network, characterised by their functional requirements or specifications. This application appears to be effective provided the number of classes or families is not restricted to a fixed structure.

Different ANN techniques posses different qualities and require or perform better under certain circumstances. The unsupervised ANNs has a dynamic ability, capable of adapting to suit the current situation, which limits its fixed structure applications, but emulates the designer's knowledge learning process. Supervised networks on the other hand require a fixed structure, which must be defined, allowing them to be easily integrated with other systems, but lack the dynamic qualities.

The ANNs have been applied to many aspects of a products design life and have shown to be a potentially powerful technique for decision making, classification and knowledge manipulation. However, it appears that the main area of research interest and research, with respect to engineering, is not in the performance of the ANN, but adapting the application to enable implementation of the ANNs. This is not to say that the performance of the ANN is not important, as without it the results cannot be achieved, but from the engineering perspective it is the ability that is important. Therefore, successful application of ANNs appears to depend mainly upon the pre and post processing of the data / information, describing the application.

2.5 Genetic Algorithms in Engineering

The Genetic Algorithm (GA), developed by Holland in 1975, is an adaptive search technique based upon Darwinian survival of the fittest and has been applied to many optimisation problems from the training of ANNs by Maniezzo (1994) to dam design as mentioned by Bullock et al (1995). Within the field of engineering (as with many AI techniques) the use of GAs have been investigated to determine their beneficial qualities.

Due to the wide variety and frequency that they currently are being used, it is suggested that this search technique appears to be very useful.

The main application of the GA is for the optimisation of parameters where the search area is vast or discontinuous. Within engineering the applications of this range from job scheduling as used by Kumar and Srinivasan (1996) to performance optimisation by Rowlands (1996). The GA has therefore, displayed flexibility for a variety of optimisation applications that could be integrated within this project.

During the review it was observed that although a number of the publications on the subject gave a brief description of the mechanisms of a GA, few described the two factors that structure and control the process. Primarily the composition of the genomes and the fitness functions. One of the works reviewed however, did explain their structure, Gen and Chen (1997). The different methods of manipulating the parameters within the genome are discussed together with the types of fitness function that can be applied. However, these are explained in a general form. It is therefore, deduced that the composition of the genomes and the structure of the fitness functions will be unique to each application.

One of the main applications that the GA will be used for within this project will be for the optimisation of component designs. In Bullock *et al* (1996) review, GAs have been used for the optimisation of turbine components, using their performance as the benchmark to evaluate the success or fitness of the design. This approach opens up a number of possibilities for GAs in design, enabling the target characteristics and goals of the design to actively influence the resultant design.

The application of the GAs to enhance component design allows the quality of the component design, which is improved conventionally through development from model to model to be accelerated within a single model of the design. Thus, enhancing the evolution of a series or range of designs.

2.6 Concluding Remarks.

From the general review of AI techniques used to simulate human activities in engineering and design in particular, the area of AI is developing in its capabilities and applications, achieving effective results that are being implemented in industry. Both ES and ANNs have been applied to the engineering design process individually, (as illustrated by the works of Su (1994), Sharpe (1995) and the review by Udo and Gupta (1994)). However, the combination of individual stages and various techniques have not been integrated into a single system following the total design process. Although the Schemebuilder project, (Sharpe 1995), who's use of a modular structure could have the potential to combine techniques within separate modules.

Each technique has strengths and weaknesses to different aspects of the design process. For example the ES excels at representing well defined knowledge while the ANNs posses the capability to generate a solution based upon random, incomplete information. Combining the best features of each AI technique offers a computerised system that is capable of dealing with knowledge in both well defined and ambiguous forms, which is often the case with design information based upon similar circumstances, exact rules, physical laws and experimental data. Incorporation of ANNs to the system also improves the process of modifying the system, as alteration to the knowledge consists of changing the network for one trained with the new information. Thus, ensuring that a design system, once developed, will not become obsolete.

The use of GAs for component optimisation offers a variety of advantages for the design system the main one being increased performance of design as the designs are optimised towards a specific goal. However, this is often computationally expensive.

It also appears that by considering the possibilities that AI holds from the point of view of developing an intelligent integrated system, opportunities become apparent that allow the level of the user's expertise to be reduced by increasing the decision making capabilities of the system, additionally retaining expertise indefinitely.

CHAPTER 3

ARTIFICIAL INTELLIGENCE TECHNIQUES AND MULTIMEDIA DEVELOPED FOR THE IIS

The field of Artificial Intelligence (AI) comprises several sub-divisions relating to different techniques of emulating intelligence. These include: Artificial Neural Networks (ANNs), rule based systems (RBS), evolutionary programming (EP), fuzzy logic, robotics, case based reasoning among others. This project combines techniques from the ANN, RBS and EP areas of AI in the development of the intelligent hybrid system. Each of these techniques is described in greater detail within this chapter together with their adaptation to suit the requirements of the application.

3.1 Evolutionary Programming

Evolutionary programming (EP) is an area of AI that is primarily devoted to adaptive searches. Using EP it is possible to search for optimums when the situation prevents other techniques such as gradient decent and direct analytical discovery from being used, due to its ability to simultaneously modify multiple non-related parameters. These situations arise when the plotted contour of the search for a solution comprising several parameters is discontinuous.

The EP search space refers to the envelope within which the search is confined. This space can become difficult to envisage, as the application of the EP becomes increasingly more complex. The number of dimensions of the search space is dependant upon the number of parameters (one for each), therefore, visualization becomes increasingly difficult after three parameters. The value of each parameter corresponds to a position within its dimension, the intersect of which will represent the solution. As the search converges, the intersection points will become increasingly closer together, until theoretically they will be coincident.

EP combines the fundamentals of the natural evolutionary process in the form of a genetic algorithm (GA) with structured knowledge relating to a combination of the problem's parameters and method of determining the success. The GA search technique is explained in Appendix A. The GA allows multiple parameters to be modified simultaneously, while the structured knowledge enables these modifications to be restrained within known physical or practical limits.

The EP additionally allows enhancements to be introduced to the optimisation process. These enhancements are in the form of a tiered process, section 3.1.1, or restrictions upon parameters as optimisation proceeds. These features of EP enable the refinement of the search area and prevent the parameters drifting into areas that are known to cause failure of design or fitness function.

3.1.1 Methods of Controlling the GA Process

The GA performs the actual optimisation process but evaluation procedures and control rules determine its success, especially when applied to complicated problems with considerably large numbers of parameters. The evaluation procedures and control rules form a shell within which the optimisation technique (usually GA) is contained and controlled. The process control program comprises of a conventional linear program which controls the optimisation process, performing pre and post process operations upon the optimisation parameters. The program utilises a series of production rules, which structure and contain knowledge relative to the optimisation process. The program determines when an optimum has been achieved, limits the modification of values and produces the initial starting values, if they are not random.

Determining the Initial Values for the GA

The initial values applied to the GA can be either random or pre-defined. The random values require no pre-processing, yet still require guidance from the program, as the value must fall within limits that the gene can contain. Similarly for the pre-defined values, variations about the original must be limited within a range. The information contained within the gene does not have to refer directly to the value. It can refer to a variable within a mathematical function that in turn produces the parameter value. The use of functions allow for a wide range of values to be contained within the gene. Part of the process control program therefore, refers directly to the parameters to be optimised that are unique to the application.

Analysis of the GA Results

The structured knowledge can also be applied to the solutions obtained from the GA. The program can be used as a governor with regard to the modification of the parameters allowing physical limitations on the values to be applied. These limits control the values from the decoded genes, restricting them to ensure that they fall within maximum and minimum levels regardless of the value contained within the gene. This process is described in greater detail in section 3.1.3.

Termination of the Optimisation Process

The final, main feature that the structured knowledge is applied to is the number of generations that are performed before the optimisation process is terminated. Two methods of terminating the GA have been identified and investigated for their application to this project, fixed length and convergence criteria.

The fixed length method requires the number of generations to be defined. Once all the generations have been completed the process terminates. This is a crude method of control as the process may either continue unnecessarily after the optimum has been achieved or more importantly terminate prematurely. If used this method requires extensive trial and error testing and analysis of the results to determine a suitable length increasing development time considerably. However, this method can be used to limit the optimisation process when convergence upon a single solution is not necessary.

Convergence criteria forms an adaptive approach to the termination of the optimisation process. This method compares the parameters of each genome with others from within the population. The comparison will identify if the population's genomes are identical. If this is the case, convergence has been achieved and the process terminates. The principle behind the method is based upon the survival of the fittest process that the GA applies. As the generations increase and the optimum solution develops the stronger genes will pass through crossover more frequently due to the weak genes being removed from the population. Therefore, the number of identical genomes within the population will increase. Once the entire population is identical the optimum achievable solution and convergence will have been obtained. It is not practical to continue the generations until the population total converges. For this reason and to decrease the time taken, a convergence limit is set. The convergence limit corresponds to the percentage of the population that must be identical before termination of the optimisation process. Setting the limit to allow for the maximum percentage of mutations prevents the possibility of an infinite loop developing. However, the higher the convergence level setting, the longer the process will take. Therefore, it is necessary to trade off the possibility of achieving a global maximum against process time. This compromise is achieved through experience and trial and error and has been determined from their use in this research as between 50% and 80%.

The process of comparing the genomes can be performed in two ways, one relating to the structure of the genome (with respect to the bit formation), the other relating to the decoded information that is contained within each gene. The first method checks each bit within the genome, basing the convergence check on the complete genome composition being identical. This method comprehensively covers the search space irrespective of if limits have been imposed on the parameters during decoding. The second method checks the decoded parameters extracted from the genome enabling limits that are imposed on the parameters to affect the rate of convergence. This method therefore, theoretically enables quicker convergence when limits are regularly imposed on parameters by treating the contents of the respective genes as the same. However, to ensure that the convergence process does not exit with false parameters the convergence level should be set toward the top end of the range recommended above. The actual level again is dependant upon the application and determined by trial and error.

The EP approach to optimising parameters enables the modification of the values without extensive knowledge about the problem or the related fields of science. This knowledge would describe the relations between parameters and preferred values analytically or experimentally derived. This may be seen by some as a removal of the emphasis on discovery and understanding of knowledge of the application, but from analysis of the trail of parameter modifications, direct analytical relations could be established. However, this is not within the scope of this project.

3.1.2 Cascade Application of the Genetic Algorithm

The GA optimisation process is applied in a cascade fashion, along a similar principle to that discussed by Patnaik et al. (1997). They discussed a project undertaken at the NASA Lewis Research Centre where a number of optimisation techniques were combined in series. Each technique in the series used the results from the previous as the starting positions. This procedure was concluded to be generally more successful and robust than any single technique tested when presented with a variety of problems.

The cascade procedure has been adapted for application to this research. The application is shown in Figure 3.1. The procedure comprises of two tiers but can be increased if higher resolution is required.

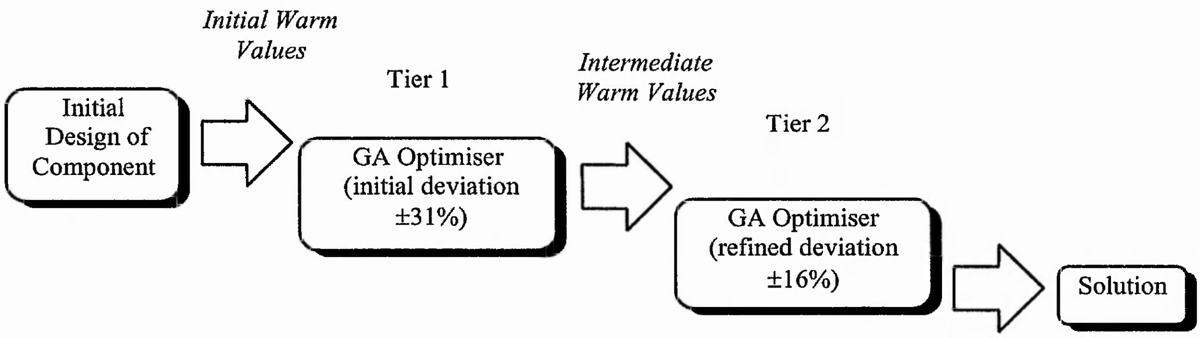


Figure 3.1 Cascade Procedure

The cascade procedure requires an initial, rough starting design to base the optimisation process upon. The initial values of the rough design form the starting positions and limiting conditions for the parameters that are to be optimised. The initial values provide 'warm' starting points within the region of a solution, as opposed to totally random, 'cold' values that increase the search area dramatically. Additionally the 'warm' values help prevent the search producing solutions that do not comply with practical or physical laws. The first tier of the optimisation is invoked using the 'warm' values to adjust the parameters in search of a global optimum. This tier searches a broad range of values ranging about the initial, 'warm' starting points. The bandwidth for this tier has been set to 31% as it provides a feasible maximum boundary for the search area, while giving a broad sweep about the initial value. However, the bandwidth can be modified to suit different applications and goals. In addition a bandwidth of ± 31% is easy to encode into the genes using $\log_2$ notation.

The optimisation process continues until a convergence level has been met, corresponding to the percentage of genomes within the population that are identical. This limit has been determined by trial and error and taken as 60% for its application described in section 5.5. At this point the information encoded in the genome with the highest fitness is decoded and forms the solution to this tier and the intermediate 'warm' starting values for the next tier.

The GA optimiser is initiated again with the solution from the initial optimiser, applying a narrower, more accurate band to the search. The bandwidth is decreased by halving the value within the gene. The effect of this is to limit the boundaries of the search and double the resolution. The increase in resolution localises the search in the region of the global maximum, enabling the accuracy of the optimisation to be increased. Again the search is repeated until the limiting percentage of the population are identical, at which point the fittest genome is decoded to form the final solution.

3.1.3 Genome Encoding

The method of encoding the information within the gene has a dramatic effect upon the performance of the optimisation process. Three methods of encoding are used, each applicable to different circumstances: direct, percentage deviation of the value and position within a pre-defined list.

Direct

The direct method is the most straightforward way of encoding the value within a gene. The value is encoded directly into the gene either in its entirety or as a proportional value that is manipulated after decoding. For example, from a range of 0 to 31, dividing by 40 alters the range, giving 0 to 0.775. If 0.8 is then added to the value the range is moved to between 0.8 and 1.575.

Percentage Deviation

This method of representing the encoded value within each gene requires two stages of decoding to extract the information. This enables the value to be stored in a condensed fashion and is ideally suited for use with initial starting points or values. This method is an alternative to storing the entire value directly within the gene in binary form, as only a percentage deviation from an initial starting point is stored. Thus, the size of the gene to can be reduced considerably for large values. The principle of the percentage deviation method is as follows. The initial encoded values within the population covers a band of possible solutions, about either side of an initial start-point. This is illustrated in Figures 3.2a and 3.2b below.

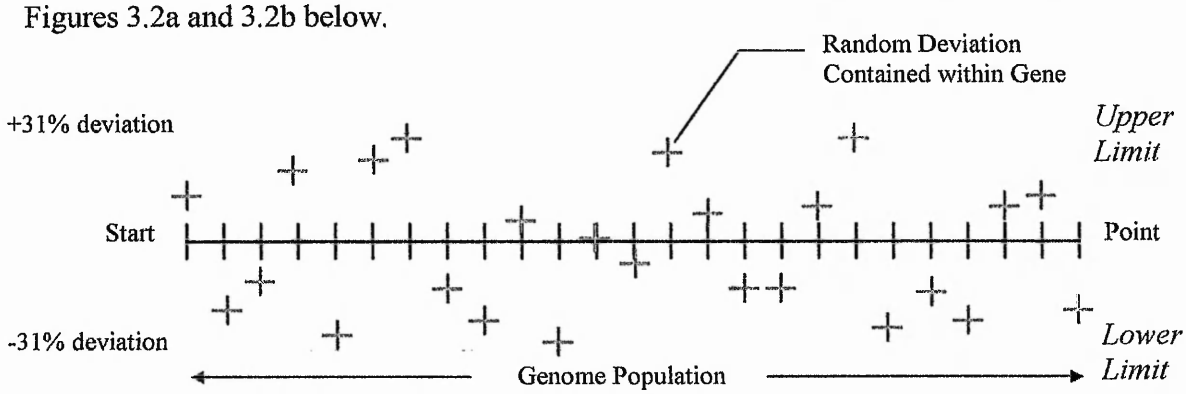


Figure 3.2a. Deviation About Initial Start Point

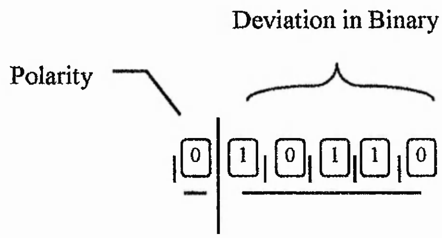


Figure 3.2b. Structure of Gene

The encoded deviations are contained within binary strings. For example a five or seven bit length gives a range of 0 to 31, or 0 to 127 respectively. The range is then effectively doubled by the addition of an extra bit to indicate the polarity of the deviation, + or -. Decoding the gene and applying the percentage deviation to the start-points value produces the new value corresponding to the search. The upper and lower limits represent the bounds of the search area in which the parameters are retained.

Pre-defined List

The third method of encoding information is by referring to a list of pre-defined values, such as standard gear modules. A pointer is moved up or down the list from a reference point to identify the position where the information is contained. The position of the pointer relative to the start-point is determined by the information contained within the gene. The decoded value from the gene relates to the number of positions within the list that the pointer is moved. Information is encoded within the gene using the same technique as for percentage deviation. For this method it is advised that the size of the gene be reduced. For example, instead of a five bit string giving a range of 0 to 31, a three bit string is used which gives a 0 to 7 range. The smaller range enables the movement within the list to be restricted. Restriction of movement is often required due to the nature of lists of information, which can change dramatically within the movement of only a few places. Also the lists size may be relatively small in comparison to the range. Reducing movement within the list will thus reduce the necessity to truncate movement, ensuring the position is within its bounds. Additionally, this reduction helps restrict the size of the genes and therefore, aids convergence by a reduction of the search area. The encoded value directly represents the number of spaces shifted up (+) or down (-) the list. The process is illustrated in Figure 3.3 below showing how a –5 deviation from the gene relates to the movement of the pointer relative to the start-point within the predefined list. The full extent of the deviation range and the list are also given in the figure, illustrating that the movement of the pointer is restricted within the bounds of the list.

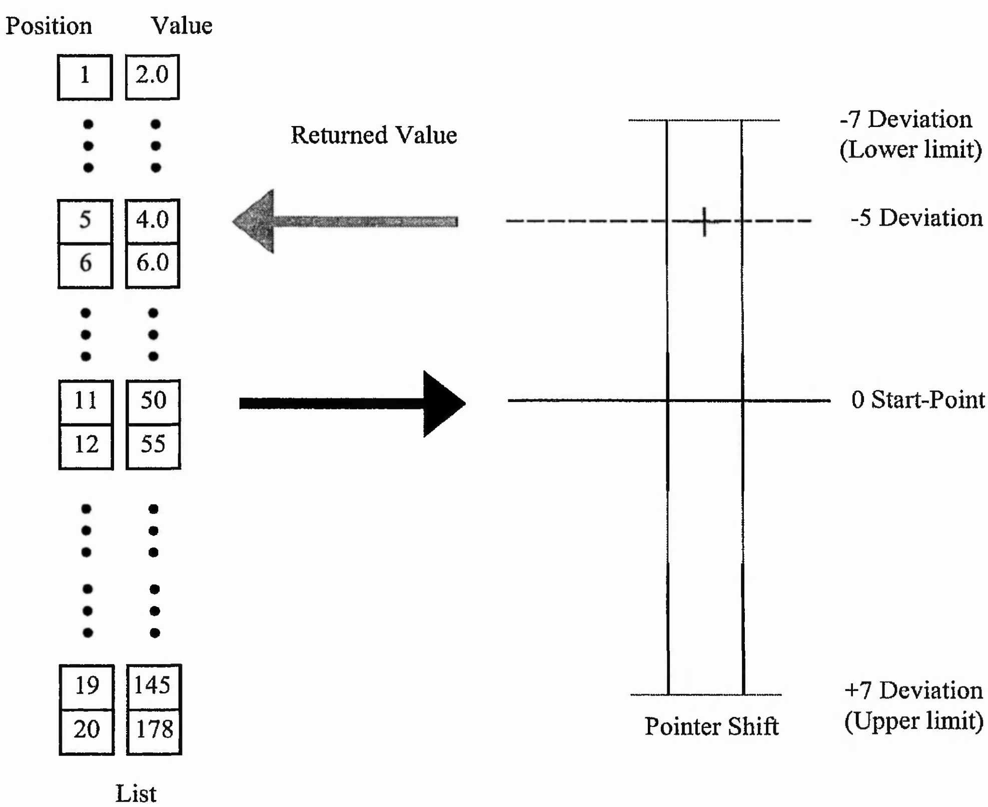


Figure 3.3 Pre-Defined List

Limits are imposed on the generated parameters to ensure that the optimisation process does not cause the values to drift into regions where the result would be invalid. These limits are maintained and implemented by production rules at the pre-process stage of the EP. The method of limiting the parameter values used is as follows. The binary value contained within a gene is decoded and pre-processed to produce the value of the parameter required by the analysis stage of the GA. If this value exceeds the limits imposed, the value is set to its relative limit. The action of constraining the value should not affect the GA process with regard to regeneration fitness levels, due to the method of encoding. As the encoded information is the deviation from a start position or a list position and the optimised value is the decoded resultant, the limit will produce the desired output provided the fitness for the limit is high. The gene characteristics that caused the high fitness will thus be passed on to the next generation.

Although the limiting process will not degrade the GAs capability for producing an optimum solution, it will retard the convergence of the population. The convergence of the population toward a single solution can be retarded if different gene configurations produce the same value. This can be the case when modifying a decoded value without reference to the gene and is due to a potentially high fitness rating being awarded to a gene that should be modified or removed from the population. Modification of the gene would be a complex process requiring detailed knowledge about the problem and the decoded parameter value. Therefore, the use of a penalty system is used to penalise genes and in turn the genomes whose content has required limitation. The penalty process is described in section 3.1.5.

3.1.4 Crossover and Mutation Operators

The levels at which crossover and mutation are performed upon the population are determined by the ability of the search to converge. A general description of the process of crossover and mutation is given in Appendix A. For their application to GAs used within this project, parent genomes for crossover are selected based upon the roulette wheel approach and use multiple random crossover points. Mutation is implemented both as a secondary operator with a low level of probability, mutating genes within a selected genome and more dynamically by the introduction of completely new genomes to the population, diversifying the search area.

3.1.4.1 Crossover

Due to the possible complexity of the search area the roulette wheel method of selecting the parent genomes for the next generation has been implemented for the GA. This method also allows fit genes contained within unfit genomes to remain in the population past the initial generations of the evolution, however, only at a low level of probability.

The effect of this may result in an increased level of optimisation, which direct Darwinian survival of the fittest will stifle, as unfit genes will instantly be removed from the population. Additionally due to the implementation of the convergence level, for determining when the search should terminate, direct entry of the best portion of genomes to the next generation cannot be used as this could lead to premature convergence.

The crossover ratio is set to a level that allows a small proportion of the previous generation through to the next generation while the remainder comprised of the offspring from crossover (reproduction). This allows parents to exist for more than one generation during the initial stages of the search and also add an historical element. The level of crossover is set based upon experience with GAs and trial and error. Many researchers have suggested that that the level should be very high, Goldburg (1989), Gen and Cheng (1997) amongst others. However, the exact value cannot currently be calculated, but should be in the region from 70% to 95% depending on the application

. Multiple, random crossover points have been used during reproduction of the genomes. This is due to the complexity of the genome and the potential size of the search space. The complexity of the genome is due to the number of genes and their non-uniform size, both of which the multiple random approach is unaffected by. Also this approach aids the search process. As the genes are spliced at random points, instead of at the intersections, the encoded information contained within the spliced gene will change, thus generating a new parameter value. This feature of multiple random crossover allows the population size to be reduced whilst covering the same search area. The convergence process is not affected by this method as the splicing of the genomes that will not alter the contents of the population.

3.1.4.2 Mutation

Three methods of mutation have been tested: total genome, bit transfer and gene mutation.

Total Genome Mutation.

Total genome mutation consists of selecting a genome from the population based upon a probabilistic rate of mutation. This genome is then totally mutated and re-introduced back into the population. The rate of mutation is varied, ranging from 2% to 10% in an attempt to increase the repeatability of the results. Several tests have been performed to evaluate the repeatability attainable from this type of mutation. The levels of repeatability form these tests were poor. However, the operator does allow similar results to be obtained from small populations when the higher mutation rates are used as for larger populations using a low rate of mutation. In addition the time to convergence is dramatically smaller in comparison to the bit transfer method described by Goldberg (1987).

Bit Transfer Mutation.

Applying the mutation operator on a bit transfer basis is the method described by Goldberg (1987). Goldberg notes that the mutation operator plays a secondary role within the GA, recommending that the frequency of mutation to obtain good results, based upon empirical GA studies, should be in the order of 1 per 100 bit transfers. However, applying this mutation operator to the search has resulted in similar results as for the total mutation operator. In addition, this method required more generations to achieve convergence of the population.

Gene Mutation.

Gene mutation is a combination of both the Total Genome and Bit Transfer mutation operators. This process, not found in reviewed literature, applies the bit transfer operator in its secondary role to make minor subtle alterations to the information held within the population, enabling the search to gradually converge, while the total mutation operator periodically introduces a new genome to the population. The introduction of the new member to the population allows the search to adaptively increase the global search area covered, thus allowing the initial population to be reduced enabling computational expense to be reduced. The new genomes are not added to the population for the duration of the search. Instead they are added until a predefined level of convergence has been attained. At this point the general area of the solution should have been found and the population begin to home in upon it. Continued addition of the new genomes will now have an adverse affect upon the search. This trend was found during tests performed on the affects of mutation operators and can be seen in Appendix C. The level at which the new genomes are no longer introduced to the population is taken as half the convergence level. Bit transfer mutation operator is applied throughout the search, as the effects are subtler.

With all the mutation operators, determining the level of mutation is achieved by trial and error, as this is the only method available. This is due to the random nature of the search process and the uniqueness of every application.

3.1.5 Fitness Functions

The fitness function, as mentioned in Appendix A, has a dramatic effect upon the convergence of the search and the parameters contained within genomes that are transferred through the generations. Therefore, the selection of criteria that comprise the fitness function must mirror the desired characteristics of the target design. Due to the nature of design two categories of function have been applied. These categories have been termed *fitness rating criteria* and *fitness conditional criteria* and apply either a gradient or step function to determine the level of fitness.

3.1.5.1 Fitness Rating Criteria

The gradient functions are used by the fitness rating criteria that determine the fitness of the designs non-critical characteristics. These characteristics do not directly cause failure of the design, but do influence its performance and guide toward optimisation targets to be achieved. The range that is given for each rating criteria is from 0 to a maximum, the greater the fitness, the higher the value. Figure 3.4 and Table 3.1 below illustrates the principle.

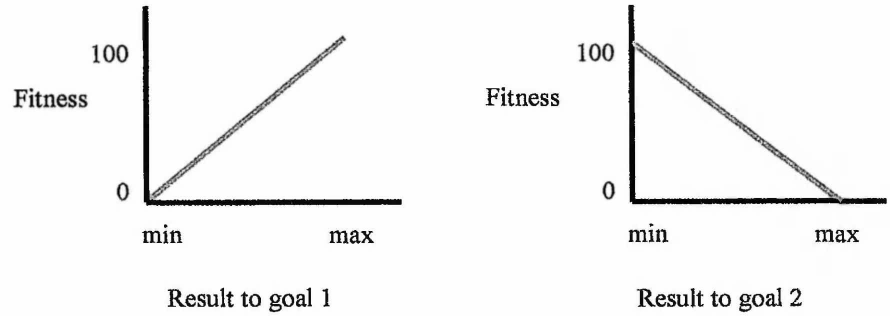


Figure 3.4 Gradient Fitness Functions

Fitness Rating Criteria	Max. Fitness (max. Value)	Min. Fitness (0)
Result to goal 1	Largest Value	Smallest Value
Result to goal 2	Smallest Value	Largest Value

Table 3.1 Fitness Rating Criteria

Set-up of the fitness rating criteria functions also has a dramatic effect upon the performance of the optimisation process and the characteristics of the search. The set-up comprises; determining the range of the fitness function limits, the affect of conflicting parameters contained within more than one fitness function and the relative rating of the maximum fitness value for individual fitness functions.

Selection of the appropriate limit range is based upon part reason, part experience and part trial and error. Setting the limits is one of the stages of the GAs development where knowledge about the design process of the product is advantageous, but not vital. The limits can be in one of two forms, global or local.

The global limits are taken from the extreme fitness values experienced throughout the entire search, thus reducing the possibility of convergence on a local optimum. However, once the search begins to localise the difference between the fitness values in the current population will reduce. This is due to the extreme unfit values continuing to influence the search. During the development of the gear performance application (section 5.5) it was observed that the use of these limits caused problems for the repeatability of results and was attributed to the reduction in resolution toward the end of the search.

Local limits are set relative to the performance of the current generation, again taking the extreme values, but only from within the current population. These limits adaptively alter their range depending upon the performance of the current generations population. This constant alteration of the fitness functions continuously applies pressure on the search to converge upon the fittest genome within the population. The rate of convergence of the search upon a solution that fulfills the optimisation goals is thus increased. Although this can have the undesirable side effect of localised convergence on a false optimum, but use of a sufficiently large population will prevent this. Local fitness limits can therefore, maintain pressure throughout the search, even once convergence occurs and variations in genomes becomes small.

3.1.5.2 Fitness Conditional Criteria

The second category of fitness functions relates to critical characteristics of the design that directly influence or cause failure of the design. These fitness functions form *conditional criteria* and apply a step function to the level of fitness. As failure to meet the requirements of the function will result in failure of the design, these fitness functions have the effect of causing the overall fitness value of the genome to be drastically reduced thus encouraging removal from the population. Figure 3.5 and Table 3.2 illustrate the principle.

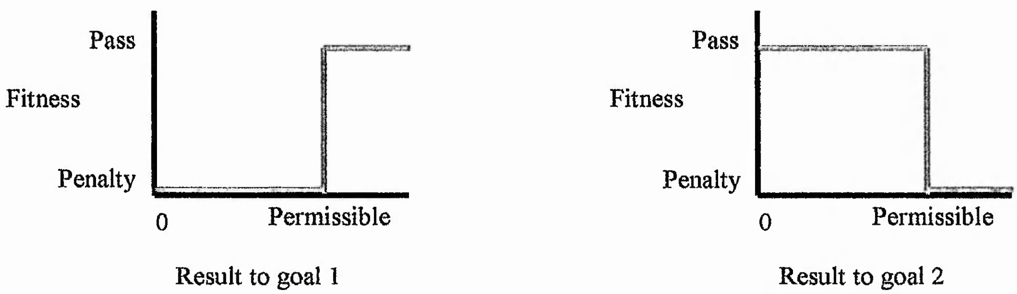


Figure 3.5 Step Fitness Functions

Fitness Conditional Criteria	Pass	Penalty
Result to goal 1	>= Permissible Value	< Permissible Value
Result to goal 2	<= Permissible Stress	> Permissible Stress

Table 3.2 Fitness Conditions

Penalties

Awarding penalties is the means by which the probability of a genome reproducing is controlled. Penalties are awarded for two reasons: if a parameter value when decoded is outside its limits or if the design has failed to meet one or more of the critical *fitness conditional criteria*.

Penalties are applied to the overall fitness of the genome instead of immediately ejecting the genome from the population. Immediate ejection removes the complete genome from the population. Therefore, possible fit characteristics of the genome can be permanently lost from the search. The use of penalties allows the genome to remain in the population for crossover and transfer its genes to the next generation, but with less probability of success. If unfit genes are transferred to the next generation, penalties will be awarded again and the probability of the genome to reproduction again reduced. The probability of the gene transferring through to a third generation is low and decreases as the generations increase, due to encouragement of fitter genomes to reproduce.

The method of applying the penalty to the genome is to scale the overall fitness. Summing the fitness values for each fitness criterion forms the overall fitness of the genome. The penalty is applied through a function that scales the fitness proportional to the number of penalties that have been awarded to the genes that the genome comprise. The value of the penalty increases as limits are imposed or conditional criteria are not met. Therefore, the greater the penalty the greater the reduction in fitness.

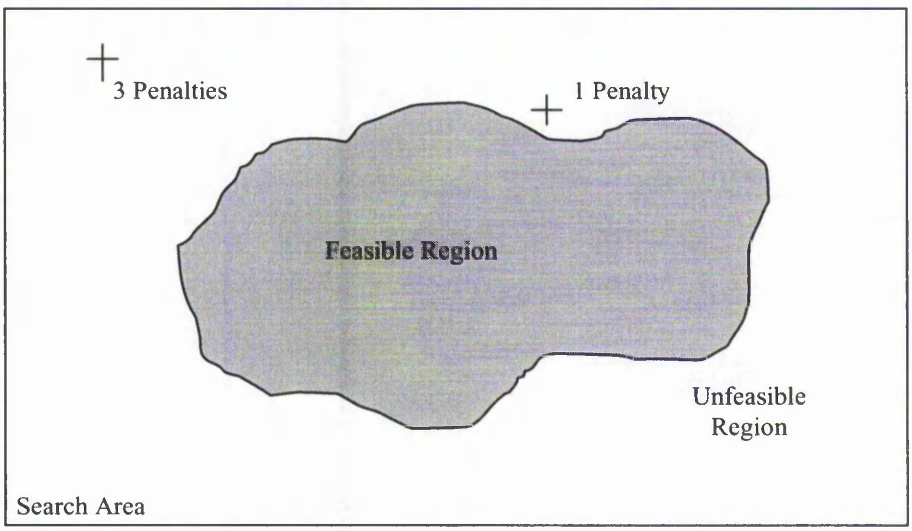


Figure 3.6. Affect of Fitness Penalties

Awarding penalties instead of ejecting the genomes that cause failure of design allows genomes that are just outside the feasible boundary to pass on fit genes to the next generation and very unfit genes to be ejected rapidly from the population. Figure 3.6 illustrates the feasible region of the search and the effect of the penalty function. The position of the single penalty genome is just outside the feasible region and therefore probably contains only genes that are fit and could aid the search. However, the 3 penalty genome is well beyond the feasible region and therefore, should be ejected from the population.

Fitness Scaling

The fitness values applied to the roulette wheel often require modification, due to the possible close proximity of some fitness values. This close proximity of genome fitness increases as the population begins to converge causing only small deviations in the fitness. Therefore, the fitness values for two different genes will cause little difference to the size of the respective roulette wheel segments. To overcome this problem scaling methods are applied to the fitness values to emphasise the relative fitness of the genome within the population

Several methods of scaling exist, using the sorted fitness order of the genomes and the actual fitness value. Several of the methods are described by Gen and Cheng (1997). Trials have been performed to evaluate the effects and success of several of these techniques.

Direct Fitness

The direct fitness rating takes the raw fitness rating from the fitness function and converts it to a percentage of the combined total of the entire population. The resultant value then represents the Genomes portion of the selection roulette wheel for reproduction.

$$\text{Fitness Rating} = \frac{\text{Raw Fitness}}{\text{Total Fitness of Population}}$$

The effects of this function can be seen in Figures 5.33a and Appendix C.1 to C.5. The test relating to the data in Figure 5.33a was discontinued. This was due to the excessive time taken to achieve convergence of a population of 600 genomes, $5.78\text{x}10^6$ generations of the gear application described in section 5.5.

Linear Ranked Fitness

Linear Ranked fitness applies the linear ranked function to the fitness rating direct from the fitness functions. This scaling function scales the fitness of the sorted genomes according to their position in the ranked order from most to least fit and is give by the function:-

$$p_k = q_0 + (k-1)\left(\frac{q_0 - q_{pop}}{popsize - 1}\right)$$

where : p_k = scaled fitness; q_0 = fitness of highest ranked genome

k = ranked position; q_{pop} = fitness of lowest ranked genome

$popsize$ = population size

Combined Fitness

The combined function combines the linear ranked fitness with the direct fitness value. The function exploits the ability of the ranking function to establish a continuos gradient to the fitness values within the population even when high levels of convergence have been reached. Super imposing the direct fitness value upon the ranked value allows significant increases in the performance of a genome to be transferred to roulette wheel and crossover. The effects of combining these functions on the fitness ratings and the convergence process can be seen in Figures 5.33 & 34 in section 5.5 and in the tables in Appendix C

The combined function combines the actual fitness value together with an element relating to its ranked position from the most to the least fit. The function is described by:

$$p_k = q_k + (k - popsise)\left(\frac{q_0}{1 - popsize}\right)$$

where : q_k = fitness direct from fitness function

This function is applied in conjunction with the deviation in fitness, thus providing greater resolution to the selection roulette wheel. The effects of the scaling functions can be seen in Appendix C, where graphical representations of the roulette wheels for scaled and non scaled searches can be found. As can be seen the functions define a noticeable ranking order without encouraging super convergence, caused by excessively biased scaling of fit genomes.

3.1.6 Summary of GA Set-up

The observations and techniques discussed relate to the applications that the GAs have been used for within this project, but are applicable for any application. Particular attention has been paid to the construction of the GA and the factors that effect its performance, which comprised:

- population size
- mutation operators
- construction of genome
- encoding and decoding information within genes
- construction and type of fitness function
- scaling of fitness values within population

These are features of GAs that have been clarified and developed specifically for this project. The intent of which is to improve the performance of the GA, reduce the computational expense and aid the design process, all of which have been achieved.

3.2 Artificial Neural Networks (ANNs)

Artificial Neural Networks, as mentioned in the literature review, are a branch of AI developed from analogies of the natural nervous system for storing and manipulating knowledge and information. Many types of networks have been developed, each developed or adapted to perform a particular type of task. A brief summary of the main forms of ANNs have been constructed into a table, giving a brief description and typical application. This table can be found in Appendix A, Table A.1. This table is not comprehensive but does include the ANNs considered for inclusion in this project. The networks considered were; the Perceptron, Multilayer Perceptron, Hopfield, Boltzmann Machine, Kohonen, and the Adaptive Resonance Theory (ART) networks. This list of networks includes both supervised and unsupervised networks, each of which has obvious advantages relating to the training process.

Neural networks form an integral part of the intelligent approach being developed for design, capturing information and knowledge in a medium that produces responses to situations which emulate an expert. The ANNs are used to aid the modular structure of the system and enable easy alteration of the information held within them. Alteration of information within the system is achieved by replacing the relevant neural network with one trained to incorporate the new information. Within the project ANNs have been used for three applications, concept selection, information storage and manipulation, and component selection. This section will consider the applications that the ANNs are to be set to and determine the appropriate type.

Each network has its own special qualities, however, the Multi-Layer Perceptron trained with the Backpropagation training paradigm (commonly known simply as backpropagation (BP) networks) displays the qualities required of a network required for the project; classification, structure and prediction.

3.2.1 Backpropagation Qualities Required within the Project

The features of the network that make it suitable of the applications within the project are:

- set structure
- ability to interpolate between trained values
- prediction
- robustness, (its ability to always produce a solution)

The set structure of the network, due to it being a supervised network, allows the network to integrate effectively with a structured system, where a pre-defined number of inputs and outputs are required. Without the constant structure of the network, modification of information and knowledge contained within them could cause major knock-on effects throughout the IIS.

Interpolation of data enables the system to evaluate situations not experienced before. The network output is based upon a comparison with similar situations, resulting in new solution if necessary.

Prediction is a feature of the network that the system uses extensively with regard to classification and suitability of designs. Its application, often in conjunction with interpolation, is required when a novel solution must be derived.

The network displays robustness to the presentation of incomplete data, always producing an output. This output may not be exact but should be within its region, thus demonstrating an advantage over rule based systems.

3.2.2 Backpropagation Training Investigation

Training of a backpropagation network has proved to be a difficult process, both for their applications within this project and for others. Difficulties arise due to the lack of effective rules and guides that help in adjustment of several independent factors which influence convergence and increase the network performance. The main factors affecting the performance of the network are: the transfer function, network topology, training period, learning coefficient and the momentum term. Several papers have been published to help determine the values of the factors, Amirikian and Nishimura (1994) and Wang *et al* (1994) amongst others, however, these guides are only relevant to the application they have been derived from. Therefore, the present method of optimising a network performance is to repeat training for different configurations of the network factors in a trial and error manner. Manual optimisation of a network performance is a tedious, time consuming and sometimes fruitless process, where an optimum solution is unlikely and therefore, needs to be replaced by a more efficient, labour saving process.

As BP networks are repeatedly used within the project an investigation was performed to determine if a series of rules of relationships could be constructed to aid in their training.

The investigation comprised of training a series of networks with varying configurations and evaluating the results relative to one another. This was performed for two separate applications, the first, based upon training data for concept generation, the other a representation of graphical information. Both in-house and a commercial ANN training package were used within the investigation. The two methods of training were used to firstly, remove the affect of any individual characteristics of the packages toward the training results and secondly, as a check that the in-house package perform correctly. The check of the in-house program is required as it was to be used throughout the IIS where Backpropagation networks are required.

3.2.2.1 Training Methods

As mentioned two methods of training have been employed. The first used a program developed for use in conjunction with the conceptual design stage of the project. The second uses the commercial package, Professional II (1993).

Several tests were conducted to determine the effects that modifications to training conditions would have on the network performance. Networks were trained with modified topology, training period, and algorithm coefficients, with the intention of increasing the performance of the network by reducing the error between the target and actual outputs. The only restricting conditions that must be adhered to were the input and output configurations, which are defined by the number of inputs and outputs required for the application. Various networks were trained, increasing network connectivity by increasing the number of elements within a hidden layer and the number of layers. The period over which the training process was conducted for was varied to determine if the length of the training period could be determined. The affects of the training coefficients within the Backpropagation paradigm are investigated with regard to their modification and their relation to the training period. The order in which the training data is presented to the network is considered together with the transfer function that is used within the process elements. All of the factors affect the performance of the network and one another, therefore, it is easy to see why the training of a Backpropagation network can be difficult and time consuming to achieve the desired results.

3.2.2.2 Training a Conceptual Design Network for the IIS

The concept design network, used for the first investigation, determines the type of transmission between stages of the transmission system. The network forms one of four networks used to generate the conceptual design. Detail of the network's purpose can be found in section 5.3.

Basic Network Structure and Training Data

The network's input and output architecture and size of training set are fixed and cannot be altered. The network requires 10 input and 5 output elements. These correspond to the structure of the training and test sets, which contain 1024 and 11 examples, respectively. The training set contains the information that the network is to retain while the test set contains examples that cover the bounds of the training data.

Performance Test.

Testing, performed by the training program, ensures the ANN is representing the training set and producing the correct output to a series of input patterns. If during testing unacceptable discrepancies between the desired and actual output from the ANN are obtained, retraining is necessary. Correct training of the ANNs is essential as they contain the knowledge and expertise of the design system. However, with ANNs it is common and expected that the output from the network will produce an error when compared with the target output. Therefore, a method of determining successful training has been defined.

The method of testing is based upon the way that the network is used within its application. The ranked output from the network, (from highest to lowest) is used to assess performance. Provided the ranked order of the network output matched that of the test case, the network is deemed to have passed. During application of the trained network, failure to match this order could result in an unsuitable component or arrangement being specified for the design.

All the results shown are typical results and have been repeated to verify their validity, showing similar trends and values for each test.

Order of training set presentation.

The effect of the order in which the training data is presented to the network was tested. An arbitrary network configuration was trained for varying periods from 1000 to 3000 iterations, applying the training sets to the network in both serial and random order. This process was repeated several times with varying network topologies and training algorithm coefficients. For the analysis of the network test results only the error between the target and network outputs have been considered. This provides a good representation of both presentation methods.

Figures 3.7a, 3.7b and 3.7c represent typical results from the network. The tests were conducted several times for varying and identical networks and the results were found to be similar in shape and value. The large difference between the first values in Figure 3.7c have been assigned to the effect of the increased training period and possibly the initial random values from which the network started its training.

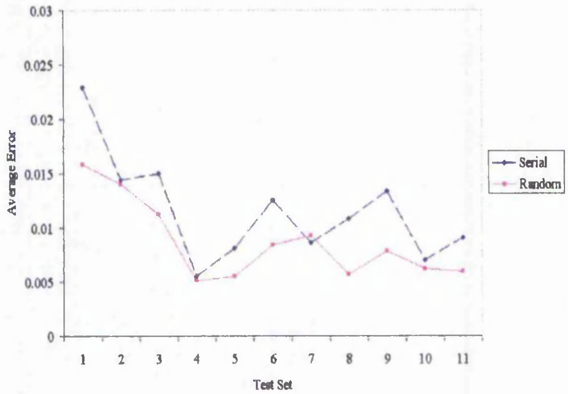


Figure 3.7a
Network Output Errors after 1000 Iterations

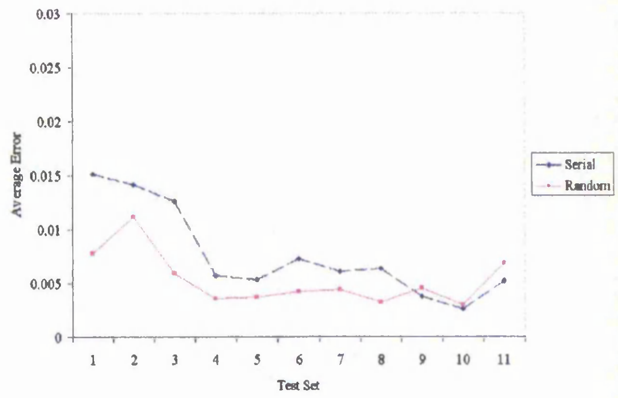


Figure 3.7b
Network Output Errors after 2000 Iterations

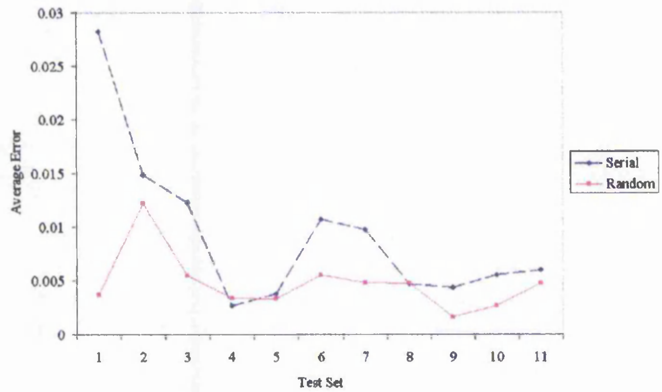


Figure 3.7c Network Output Error after 3000 Iterations

It can be seen that the order in which the test cases are presented has a small effect upon the performance of the network. Random presentation appears to give slightly better performance, which could make the difference between network performing correctly or not. Additionally it can be seen that increasing the training period past 1000 iterations does not produce notably increased performance from this network.

Modification of topology and backpropagation coefficients.

Order of testing. The parameters to be modified in search of trends upon the training of networks are the topology, training period and the learning and momentum coefficients. A series of tests have been performed with modified training parameters.

Topology only. Networks have been trained with varying element architectures. The training period for analysis of the effect of topology has been taken as 350 iterations of the entire training set. Network structures will range from 3 to 19 elements across 3 hidden layers. The number of successful test cases for each network is illustrated in Figures 3.8a, 3.8b and 3.8c below.

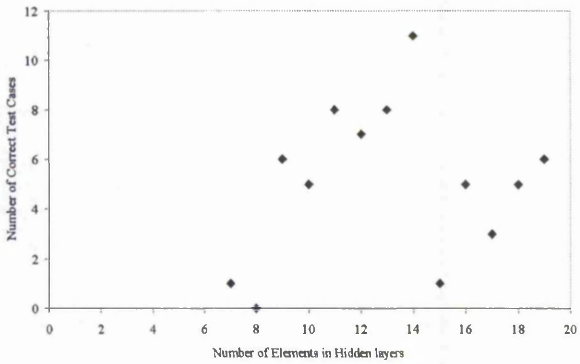
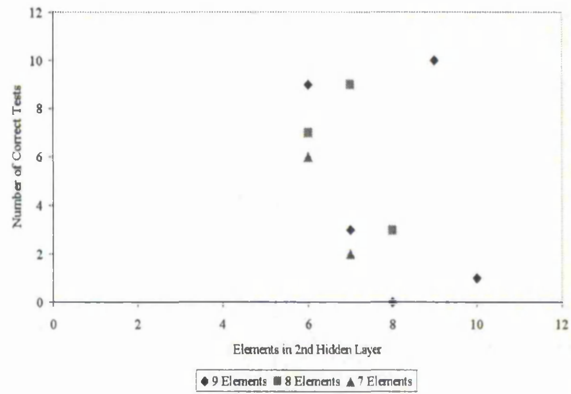


Figure 3.8a
Performance of Single Hidden Layer Network



Elements in 1^{st} Layer

Figure 3.8b
Performance of 2^{nd} Hidden Layer Network for Different Sizes of 1^{st} Hidden Layer
(Varying number of elements within layers)

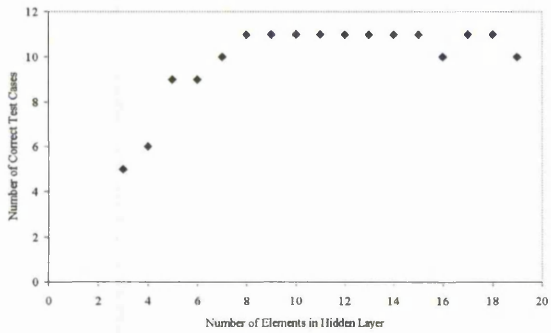


Figure 3.8c
Performance of 3 Hidden Layer Networks (Same number of elements in each layer)

It has been deduced from these results that the performance of the network is greatly affected by its topology. However, it is not clear as to how the structure should be modified to increase the performance. Increasing the number of elements within the hidden layers does aid the performance as would be expected, due to the extra connections between elements. However, the increase in performance appears to level off after a point and even decreases if too many elements are added to the layer. This may be due to saturation of the network.

The effect of increasing the number of hidden layers is inconclusive. From comparison of the results for the 2 and 3 hidden layer networks shown in Figures 3.8b and 3.8c it appears that the performance increases with the number of layers. But comparing the results from networks with 1 and 2 hidden layers, (Figures 3.8a and 3.8b) this does not appear to be true. Comparison of the results for 1 hidden layer networks and 3 hidden layer networks again indicates that the performance increases with additional layers, but not significantly.

Training Period.

The networks tested have one hidden layer of elements. This structure has been chosen, as they have displayed reasonable performance for the low training period tests and are expected to display more evident effects from the additional training. Figure 3.9 below illustrate the performance of the networks

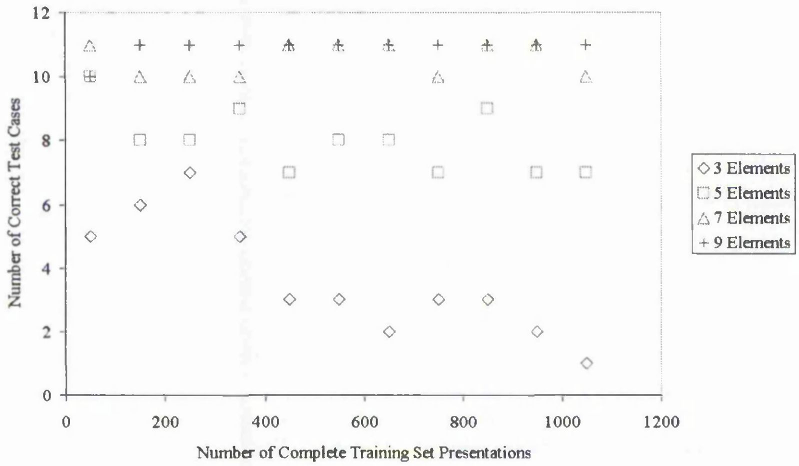


Figure 3.9 Effects of Increasing Training Period on Network Performance

Increasing the training period appears to aid the training of the network up to a point. After this point the performance of the network either does not change or has a detrimental effect.

Training Coefficients

To determine the effects of the learning and momentum coefficients upon the network, several tests have been performed, modifying their effect as training proceeds. Firstly, the effect of the learning coefficient is determined. The value is decreased from its maximum of 1.0 by increments of 0.1 down to a minimum of 0.0 every 100 presentations of the complete training set. Secondly, the effect of the momentum term is decreased in the same way, by decreasing the value of the momentum coefficient. The initial value is taken as 0.6 and reduced in steps of 0.03 down to a minimum of 0.0. The initial starting point is limited to 0.6 as the effects of this term can have a detrimental effect on the network's ability to converge. This is due to the magnitude of the terms effect being proportion of the previous value, thus a resonant effect can be caused. Figures 3.10a and 3.10b present the results.

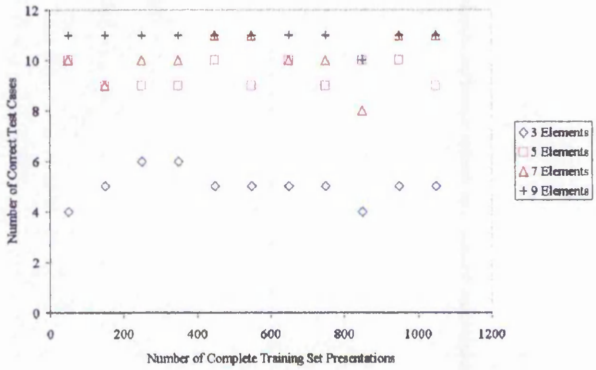


Figure 3.10a
Effect of Reducing Learning Coefficient as the Training Period Increases

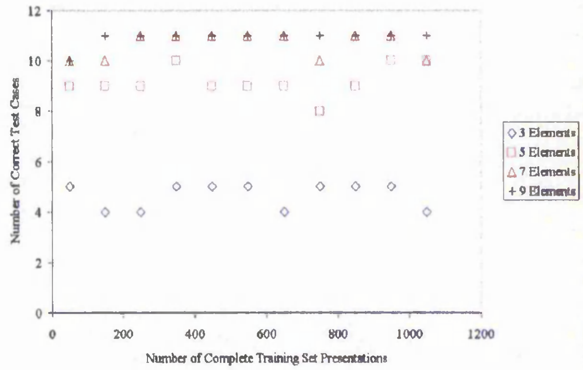


Figure 3.10b
Effect of Reducing Momentum Coefficient as the Training Period Increases

In comparison with each other, modifying the coefficients has had little effect upon increasing the performance of the networks. However, if their effect is compared with the standard network (Figure 3.9) it can be seen that a general trend appears indicating that the performance of the network continues to increase as the presentations increase and the coefficients are gradually reduced. This effect can be seen more clearly when comparing networks comprising 3 elements in the hidden layer, (Figures 3.9 and 3.10a).

Summary of observations

The following observations have been made about network training:

1. Randomly applying the training set during training has a beneficial effect upon performance.
2. Increasing the number of hidden layers will not necessarily increase the performance of the network.
3. Increasing the number of elements within a layer can increase the network's level of performance up to a point. After which a plateau effect is observed with regard to the performance. After a period of increasing the number of elements with no increase in performance end of the plateau is reached and the performance will decrease.

4. Increasing the training period increases performance up to a point, after which the network is over trained and has difficulty producing outputs other than in the region the most frequent training examples.
5. Decreasing the learning and momentum coefficients as the training process proceeds has a beneficial effect on final performance.

3.2.2.3 Training a Graphical Information Network for the IIS

The graphical application networks are employed to simulate information for gear design presented within a diagram. The data presented is derived experimentally and is therefore, of an ideal form to be represented by an ANN. This is due to its ability to interpolate between points. The purpose of each network is to interpret the design diagrams and produce an output similar to results obtained by manually reading from the graph, interpolating if necessary.

Basic Network Structure and Training Data

Basic structure of the network requires 3 input elements and 1 output, which correspond to the training and test data sets. The training set contains 50 examples of uses of the diagram, distributed across its entire scope. The test set comprises of 2 sets of test cases, derived from a combination of direct reading and interpolated values from the diagram. Each test set comprises of 10 test patterns of increasing difficulty for the network. The difficulty is increased by increasing the complexity of interpolation required to obtain the output.

Network Test

Training and generation of the source code for the networks was performed using Professional II, a neural network package. The trial and error approach was used, modifying the topologies and coefficients until the final networks were produced.

Transfer Function

Training was attempted using one of two transfer functions, a hyperbolic tangent (tanh) and a sigmoid. A selection of network configurations were trained and tested using both of the transfer functions. Figure 3.11 below presents the results from one of the tests and illustrates the general trend observed.

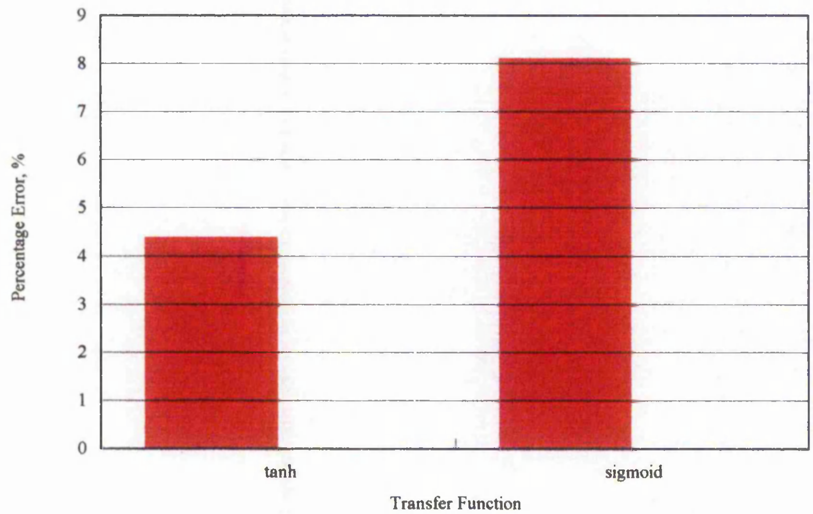


Figure 3.11 Comparison of Transfer Functions

From the results it can be seen that the transfer function can have a dramatic effect upon the performance of the network. As demonstrated by the error from the tanh network being almost half that of the sigmoid.

Topology A series of structures were tested in an attempt to increase the performance of the networks. These included increasing the number of elements within a single layer and by a combination of increasing both the number of layers and elements contained within.

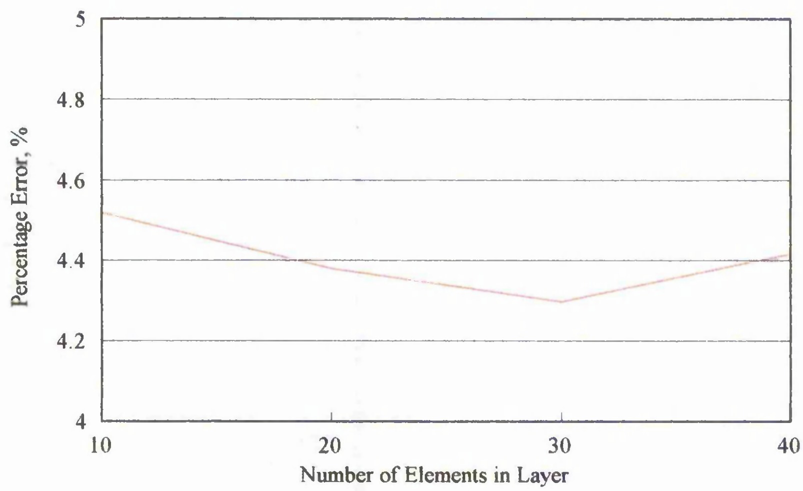


Figure 3.12a Increasing Elements within Hidden Layer

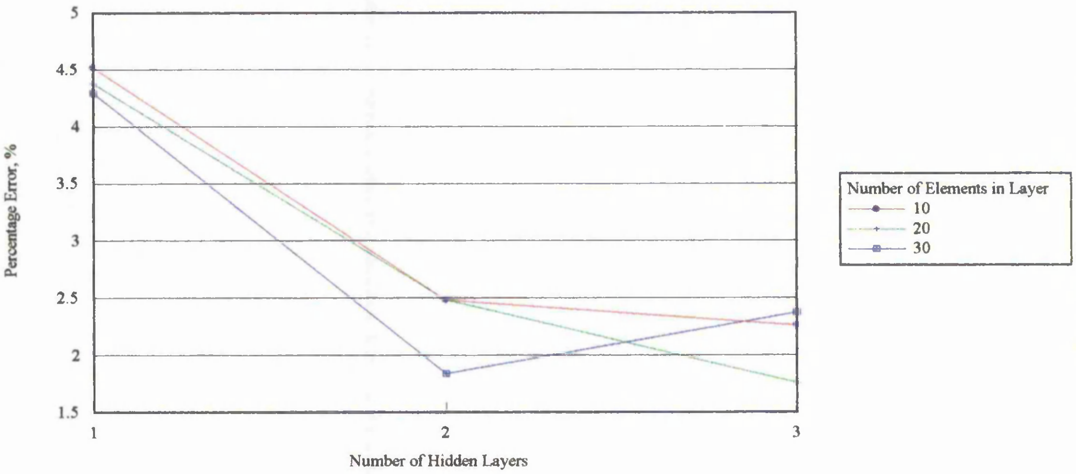


Figure 3.12b Increasing Hidden Layers

From Figure 3.12b it can be seen that increasing the number of elements in a hidden layer increases performance up to a point, after which performance did not significantly increase, but actually decreased.

The effect of increasing the number of hidden layers and therefore the connectivity of the network is to improve performance. The increase in performance is most evident when increasing the number of layers from 1 to 2. Increase in performance has generally been achieved by the introduction of a third layer but not to the same degree as the introduction of the second. These observations are evident in Figure 3.12b.

Summary of Observations

- Transfer function affects the performance of the network. The tanh function appears to achieve better results than the sigmoid.
- Increasing the number of elements within a layer improves performance up to a point. After this point the performance can diminish.
- Increasing the number of hidden layers increases performance. However, the greatest increase in performance is by the introduction of the second hidden layer.
- Modifying the coefficients throughout the training period, starting with high values (i.e. 0.9 for learning coefficient) and finishing with very small (i.e. 0.001) improved network performance.

3.2.2.4 Network Training Conclusions and Guide Lines

From the two case studies, it can be seen that the topology of the network has a profound effect upon the performance. Similarities have been observed in both cases that relate increases in performance to an increase in the number of elements within the network. This is as expected due to the increased connectivity of the net. The arrangement of the elements also has a direct relation on the connectivity and the performance. The number of elements in the surrounding layers governs the connectivity. For example, if the number of elements in the next layer was small and the current layer large, greater connectivity would be achieved if the element was added to the next layer. If the performance of the network was determined by the connectivity of the net it would be reasonable to expect an increase in performance as the number of connecting weights increases. Thus, increasing the number of layers should increase performance. This has been found not to be the case in general. Increasing the number of layers can increase performance, but networks with extra elements and lower connectivity (more elements within one or two layers, instead of distributing the elements across more layers) have performed better than networks with high connectivity of elements. As can be seen from Figures 3.13a and 3.13b increasing the connectivity can always lead to improved performance. Therefore, there does not appear to be a definite relation between the connectivity, the number of layers and number of elements of the network and its performance that can be applied in general to all network applications.

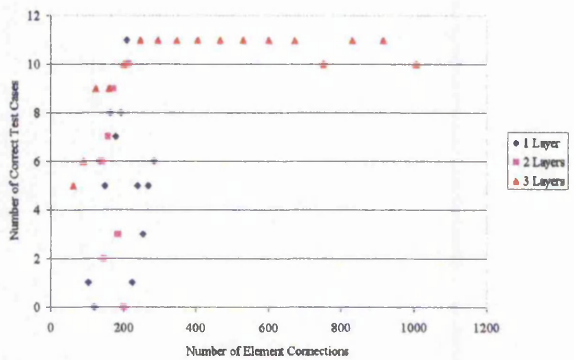


Figure 3.13a
Effect of Element Connectivity
(Case 1 - Concept Network)

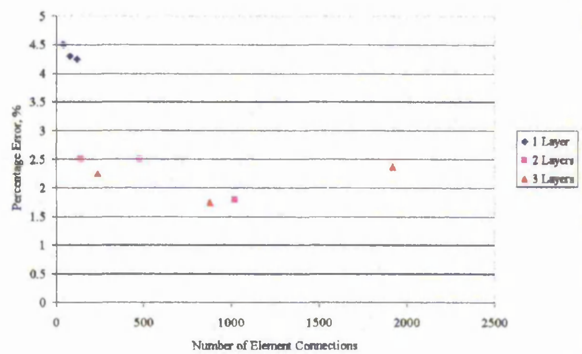


Figure 3.13b
Effect of Element Connectivity
(Case 2 – Graph Network)

Modifying the Learning coefficient and the size of the momentum term has had a beneficial effect on both test cases. Starting with large values for these factors and reducing their effect on the error fed back from the training algorithm appears to initially shake up the network then allow the weights to settle at values that produce the minimum error. The rate at which the coefficients are reduced will be dependant upon the size of the training period, which in turn will be dependant upon the size of the training set and the topology of the network. Unfortunately attempts to define a relation between these parameters have proved fruitless.

From the results of the tests for the two applications it is concluded that the training parameters for each application obey different characteristics unique to that application. Therefore, it is concluded that the best training procedure for training a backpropagation network is by trial and error. The trial and error process can be guided by the following general rules illustrated in Figure 3.14.

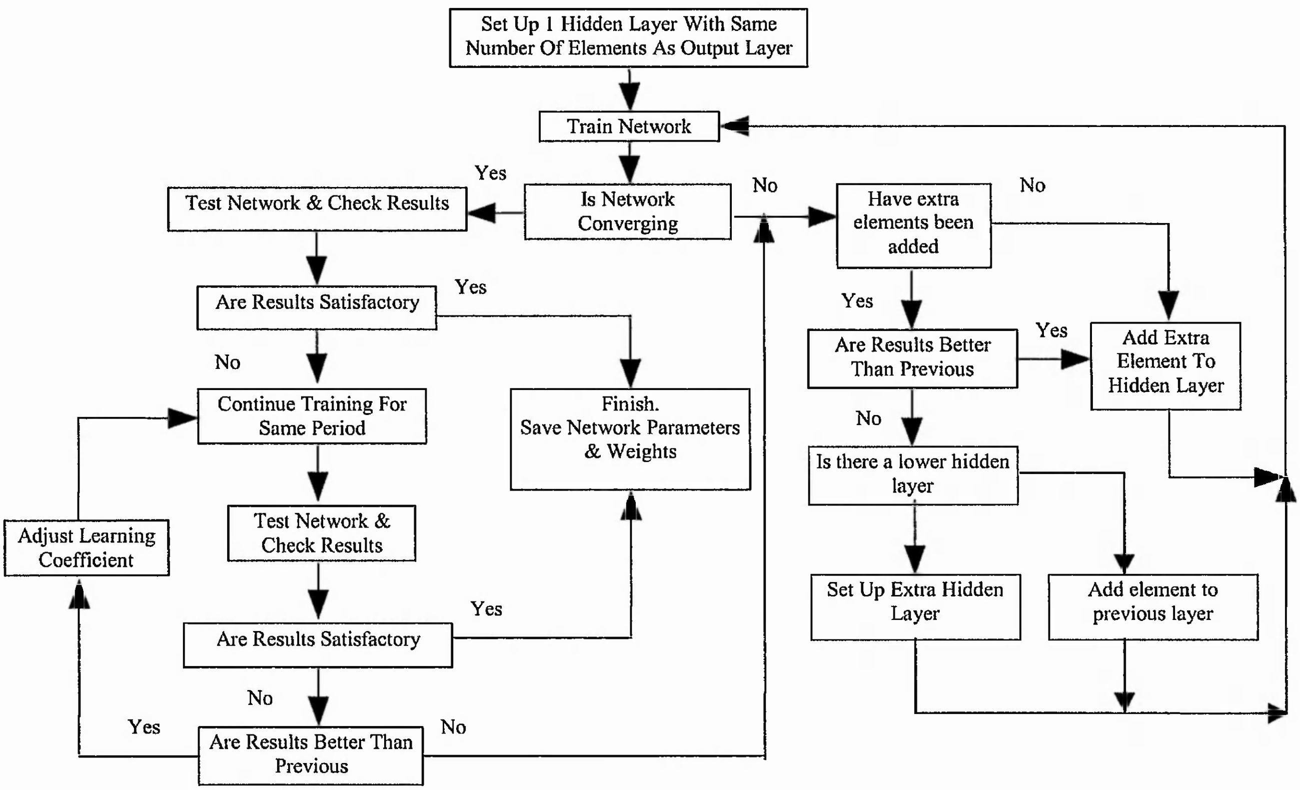


Figure 3.14 Manual Training Guide

The compound effects of the network parameters upon its performance prevents certain definition of the topology and training coefficients of a network to achieve successful representation of information presented to it. Even with the aid of the flowchart in Figure 3.14, the training process can be slow and possibly fruitless. Therefore, for the ANNs to form an integral part of the IIS, a means of training the network quickly, requiring little expertise must be developed. A solution to this problem has been achieved by applying the adaptive search quality of a genetic algorithm to aid network training. This process is described in the next section, 3.2.3.

3.2.3 GEN-NEU Approach as an Aid for Backpropagation Network Training

Training an ANN requires expertise in the process, due to the factors mentioned in the previous section, 3.2.2. Therefore, a means of removing the expert element from the training process is required to allow modification of the knowledge within the design system by the user. A training aid has been developed to train backpropagation networks requiring only the information to be represented.

An optimisation search technique has been implemented to control the training process. A number of search techniques were considered such as hill climbing and the Newton Raphson method but due to the magnitude of the search area a continuous search would be too time consuming. An adaptive search technique appears to provide a suitable solution, as it is capable of covering the search space without analysing every point. A genetic algorithm based approach has been developed within the project, which implements the optimisation capabilities of a GENetic algorithm to define the factors that determine a NEUral networks performance (GEN-NEU).

Researchers have been making efforts to apply GA into the optimisation of backpropagation networks. For example, Caudell and Dolan (1989) used them to adjust the connective weights during the training process to improve the networks convergence and Miller *et al* (1991) and Maniezzo (1994), have applied them to the optimisation of networks topologies. However, the adjustment of the connective weights by implementing GAs does not provide significant improvements in network performance over the backpropagation technique, and therefore have not been used for this purpose. Optimisation of network topology on the other hand has provided more encouraging results, although only this factor which affects the network's performance has been addressed.

The GEN-NEU approach takes into account three major factors that affect network performance: topology, transfer function and training period. Optimisation of these factors is performed simultaneously, considering their combined effects upon performance and convergence, thus making the optimisation more effective.

3.2.3.1 Adaptive Optimisation Process of the GEN-NEU Approach

The basic principle of the GEN-NEU approach, developed for this project, is to adjust the factors required by the backpropagation training technique based upon the performance (fitness) of the network being trained. Gene sequences which produce high fitness levels combine with other fit sequences to form an optimum. Figure 3.15 demonstrates the process.

The approach encodes the values for the factors that influence the performance of the network, including the transfer function, topology and training period into separate genes within the genomes. Upon initiation the values contained within the genes are randomly set from values within the search space. The networks corresponding to the information contained within all the genomes of the population are trained and tested then sorted into order of descending. The fitness value of the genome is determined from the network output response to a series of test cases applied after training. The lower the root means squared (RMS) error between the target and the networks output the fitter the genome, which in turn determines the genomes probability of reproduction. Each genome's fitness rating, relative to the rest of the population, is proportional to its probability of reproduction. Thus, fitter genomes are given a greater chance of transferring their genes to the next generation, while unfit genomes are gradually removed. The fitness value is therefore, taken as the reciprocal of the network error ensuring the fitter genomes get a greater probability of reproduction. The combined scaling function, described in section 3.1.5, is then applied too the enhance the performance of the GA.

The genomes reproduce to form the majority of the next generation. Reproduction accounts for approximately 98% of the next generation. The remainder of the next generation comprises of random selections from the current population that pass unaltered through to the next generation.

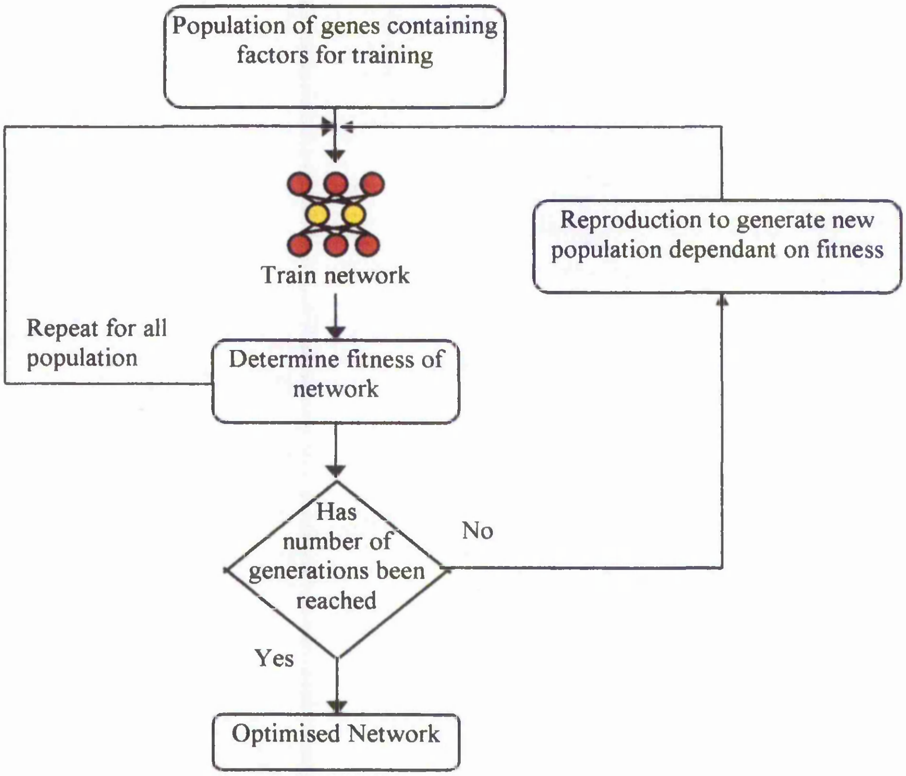


Figure 3.15 Basic GEN-NEU Process

Once the next generation is established a mutation operator is applied to the population with a probability of 5%. The mutation operator applied alters the encoded information contained within the complete genome as opposed to the typical bit mutation operator, Gen and Cheng (1997). Total mutation produces a new member to the population allowing the search to rapidly ingress into new areas of the search space without extending the population size. Thus, the total area covered by the search adaptively increases as the process progresses. Mutation also helps prevent localised minimums limiting the scope of the search. As the affects of the mutation can be dramatic upon the search the probability of mutation is set low to limit disruption once convergence commences upon a solution.

Once crossover and mutation have been performed the new generation is complete and ready for the new fitness ratings to be determined. After the final generation the network corresponding to the fittest genome becomes the resultant network ready for application and all the connective weights, topology and transfer functions recorded.

The number of generations that the GA performs is set to a finite amount instead of using the convergence of the population. This is due to the effect of the random initialisation of network connections at the beginning of training, which can lead to variations in performance. These variations can disrupt the GA convergence process, therefore, the fixed length prevents excessive computational expense for a small increase in network performance.

The information relating to the factors that affect the performance of a network during training are encoded into the genes within the genome in binary form. An example is shown in Figure 3.16. The decoded values of which correspond to a sigmoid transfer function, 19 elements in the first hidden layer and 9 in the second with a training factor of 5.

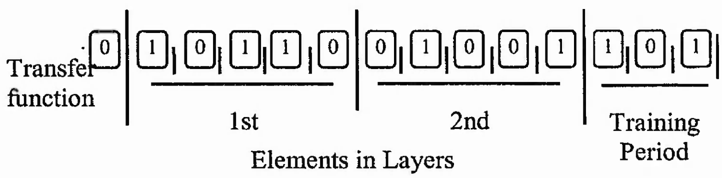


Figure 3.16 Combination of Genes to Form Genome

Binary coding has been selected as opposed to others, such as Gray coding for two reasons. Firstly, as the type of coding will have little effect on the performance of the GA, as the genomes comprise multiple genes whose information does not necessarily relate directly to one another. Secondly, crossover is performed at random points during reproduction allowing genes to be spliced, resulting in significant changes in the encoded values.

The length of the training period for different applications has a direct influence on the performance of the network. Therefore, integration of the training period into the GA optimisation approach will aid successful training and reduce user interaction with the training of the network. However, the encoding of the length of the training period into the genome required important consideration.

From experience in training networks the length of the training period for different applications varies. Therefore, integration of the training period into the GA optimisation approach will aid successful training and reduce user interaction with the training of the network. However, the encoding of the training period information into the genome required important consideration.

The training period has been encoded in binary form as for the topology, from 0 to 31. The value from the gene is pre-processed before training to obtain a realistic value. Figure. 3.17 shows the effect that the training period has upon the performance of a network and two per-process functions, *150x* and $10x^2$, using the range of values available from the genome.

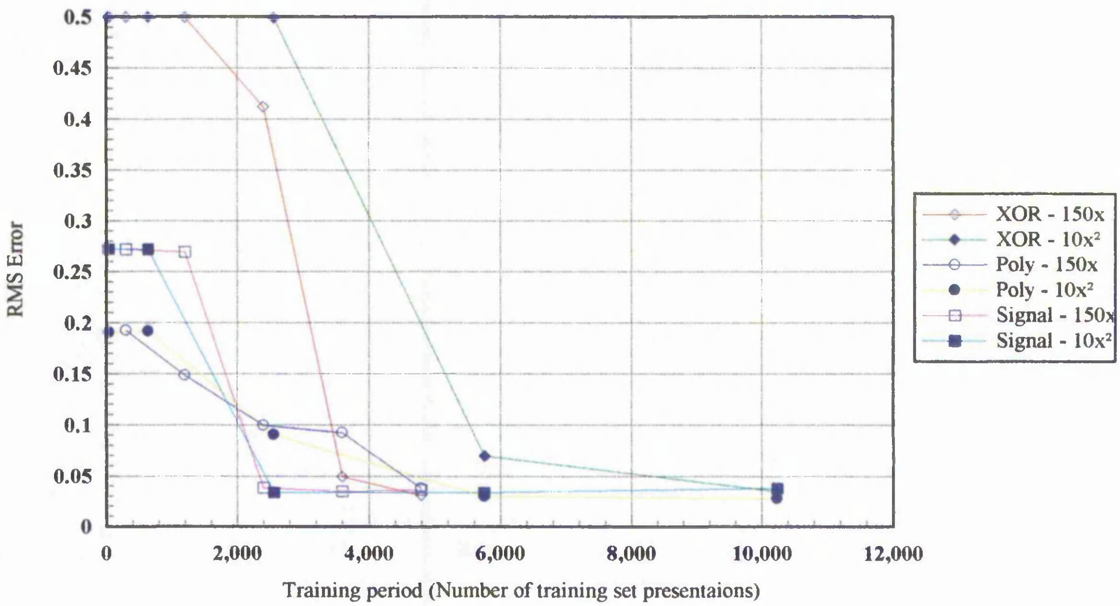


Figure 3.17 Effects of Training Period

The effect of the training period and the pre-process encoding functions were tested for the three test cases used later in this paper, XOR, a polynomial (poly) and a feedback control loop (signal). The pre-process encoding functions obtain similar results, where x is the value encoded in the gene (0-31). However, these results are from the training of relatively simple problems, however, these functions provided training periods in excess of the levels require. This is illustrated by the leveling of the network error. More complex problems may require extensive training presentations, which the $10x^2$ function can offer

3.2.3.2 Validation of GEN-NEU Program

To ensure that the GEN-NEU approach performs the optimisation process correctly and aids training, a series of validation tests have been performed, checking the BP technique, the GA process and the network performance achievable. These tests cover a variety of different applications including: classification, mathematical modelling and signal processing.

Validation of Backpropagation Network Performance

Validation of the network consisted of the exclusive OR (XOR) benchmark test. Classification relates to many applications where information appears to be in a random fashion, displaying no apparent order. The purpose of the network is thus to sort the information into classes or groups. The XOR problem has been presented to the network to prove that the network can classify data requiring more than one hyperplane.

The GEN-NEU program successfully trained a network to model the XOR problem, returning the network weights and structure information necessary to duplicate the network. The resultant network comprised of two inputs, a single hidden layer containing two processing elements (PEs) and a single output PE, (2-2-1), trained for 5000 presentations of the training data using the sigmoid transfer function. The trained network achieved an RMS error of 0.021 from the target values which after rounding to the nearest integer was removed.

To ensure that the training process was performed correctly a network was trained under the same conditions as those determined by the GEN-NEU program using the commercial ANN development tool, Professional II. The resultant RMS error was 0.024, thus indicating that the training process had no errors. The small variation in results is accredited to the random initialisation of the connection weights.

Validation of GA Approach

Validation of the GA approach consists of running the GEN-NEU program to optimise the network topology and transfer function when presented with the XOR problem. The topologies and transfer functions contained within the genomes of each generation are represented in Figures 3.18a and 3.18b. These figures give different perspectives of the optimisation process. Figure 3.18a illustrates the topology of the ANNs, while Figure 3.18b illustrate the transfer function of the PEs (where 1 = sigmoid and 0 = *tanh*).

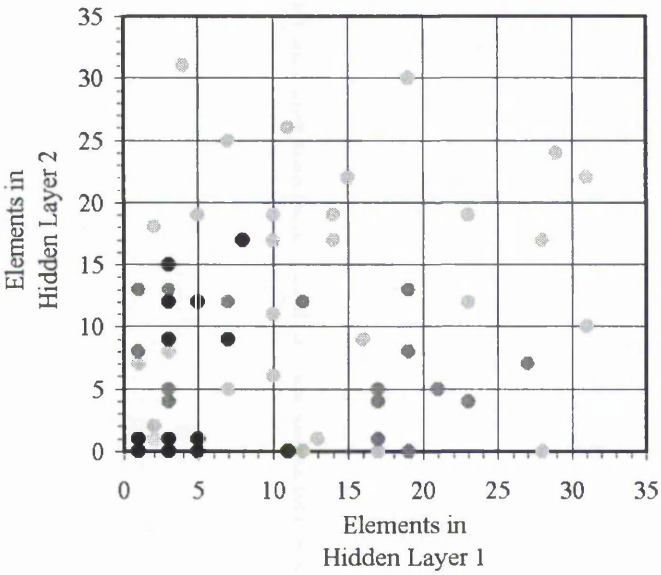


Figure 3.18a Convergence of Elements in Layers

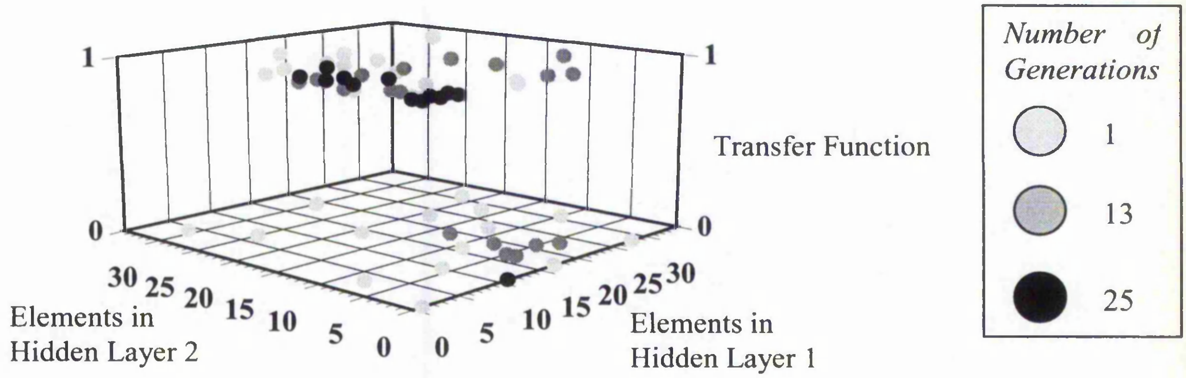


Figure 3.18b Convergence of Transfer Function

It can be seen from the localising of shaded points in Figures 3.18a and 3.18b that as the generations increase the search begins to converge towards the optimum network. The initial random scatter, shown by the light gray points, migrates from a low density concentration to a higher density at the end of the process. As the populations converge, multiple genomes generate the same network configuration and thus form coincident points on the graphs. This accounts for the apparent reduction in points as the generations increase. These figures demonstrate the GAs ability to converge upon a solution.

Performance of GEN-NEU Approach

The performance of the GEN-NEU approach has been tested using two ANN applications, modelling a mathematical equation and signal processing. These applications require different qualities of the ANN, therefore, providing a means of establishing application independence.

Application 1: Equation Modelling

The ability of the network to follow the contour of a line demonstrates a network's ability to model mathematical functions. A practical application of this feature would be for the control of a robot arms movement. For the purpose of validation, the polynomial $y = x^3 + 4x^2 - \frac{1}{x}$ was represented by the network. The resultant network from the GEN-NEU program was capable of modelling the function with an RMS error of 0.0025. This was achieved with a 1-7-3-1 topology, a *tanh* transfer function and a training period of 7680 iterations of the training set. The progress of the GEN-NEU program for the training of this network is illustrated in Figure 3.19. From the path of the RMS error for the population it can be seen that the trend of the training aid is to improve performance. Erratic fluctuations in the performance can be attributed to the introduction of new unfit genomes to the search population.

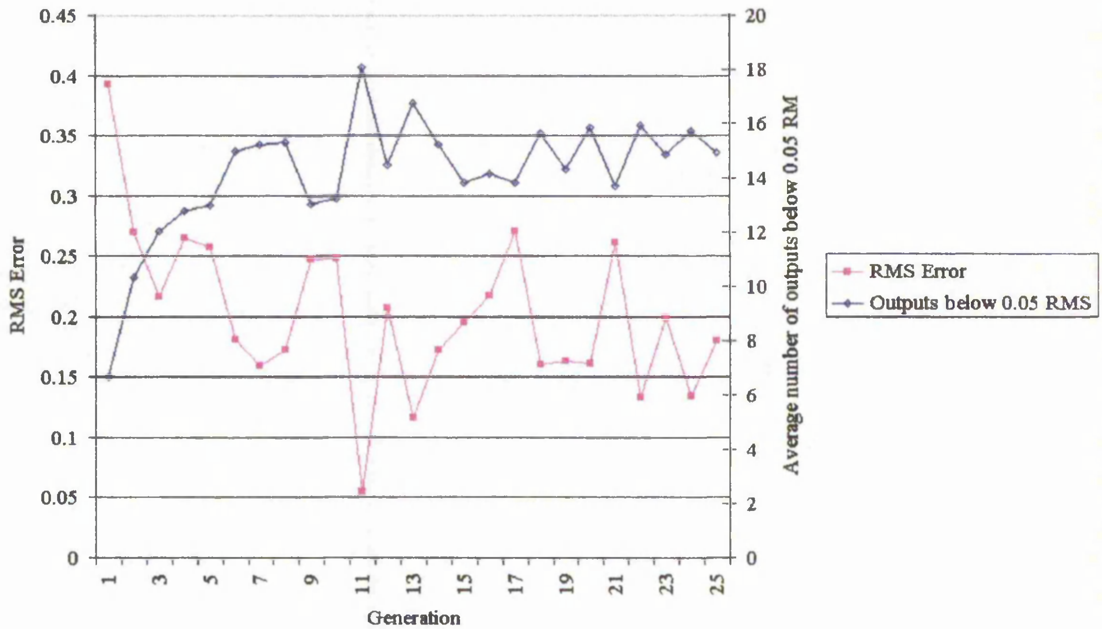


Figure 3.19 Average Performance of Networks During Training (Equation Modelling)

Application 2: Signal Processing- Prediction

The capability of the network to predict values is another frequently used feature of ANNs, where an output is required from a region the network has not previously encountered, between training areas. Prediction was tested by applying Lapedes and Farber's (1987) signal processing problem, $x(t+1) = 4x(t) \cdot (1.0 - x(t))$. This application requires an output based upon the effects of the previous output, as in the case of a feedback loop.

The resultant network from the GEN-NEU program comprised of a 1-4-20-1 topology, using *tanh* transfer functions and trained for 7680 iterations of the training set. This training combination produced a RMS error of 0.0386 to the test data.

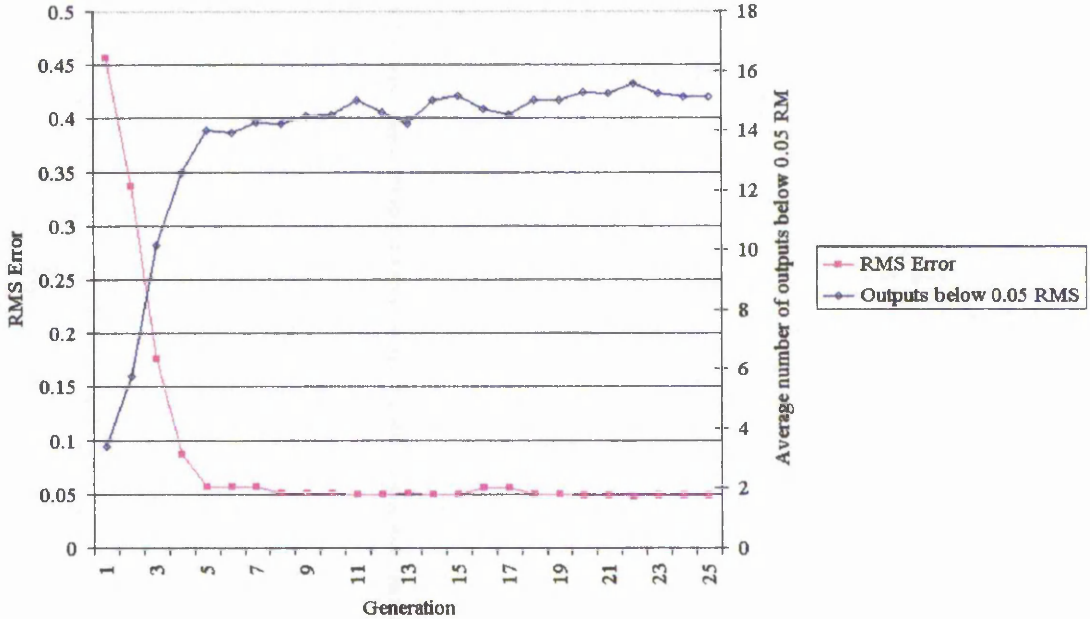


Figure 3.20 Average Performance of Networks During Training (Signal Processing)

As the training factors begin to converge into the region of the solution the average performance of the networks within the populations also begins to improve. This trend is evident in Figures 3.19 and 3.20 from the number of test RMS errors below 0.05. Thus, the average network performance increases as the configurations of training factors that produce low performance networks are removed from the search, while the remainder combine to improve performance.

3.2.3.3 Application to Project : Optimisation of ANNs for Detail Design of Gears

The GEN-NEU program has been used to supervise and control the training process to optimise the performance of four networks applied to the detail design of gears, within the detail design module of the integrated system. These networks encapsulate and interpret information held within graphical design aids.

The analysis of the training and results of the network for the determination of the face load factor for contact stress, $K_{H\beta}$, will demonstrate how the GEN-NEU approach to training backpropagation networks has been applied to the integrated design system. As the approach unfolds and the network is trained, the various stages of the GEN-NEU approach are explained. The requirements of the user describing the application and the preparation of the training data are defined together with an evaluation of the resultant trained network. The performance of the GEN-NEU process and results are analysed and for the purpose of this example a comparison with a conventionally trained network using Professional II is made.

Data Preparation

The information to be encapsulated within the network is contained in the graphical design aid shown in Figure 3.21. The graph contains a large amount of information that would be difficult to interpret into a formula for computing without the original data that the graph is constructed from. As the original information is not available, (as is the typical case if the graph is taken from manufacturers literature), the training data is extracted directly from the graph.

Data from the graph is in two forms, direct and interpolated. The direct data is formed from the existing curves, while the interpolated data represents the intermediate values between. The second form requires the designer's judgement and expertise to be determined. The interpolated curves are represented by the dashed b/d_1 curves in Figure 3.21.

The training data preparation consists of determining the values of $K_{H\beta}$ (network output) for combinations of pinion diameter, gear accuracy and facewidth ratio (network inputs). The example in Figure 3.21 illustrates how to use of the diagram. For a pinion diameter of 115 mm a line is projected to intersect with the curve corresponding to a gear grade of 5. From the intersection, a second line is projected up until it crosses the 0.8 facewidth ratio curve. From this intersection a final line is projected to cross perpendicular to the axis. The point at which the line crosses the axis gives the face load factor for contact stress, $K_{H\beta}$. The arrows on Figure 3.21 illustrate the path taken, resulting in $K_{H\beta}$ = 1.42. As can be seen interpolation plays a vital part in attaining and accurate result.

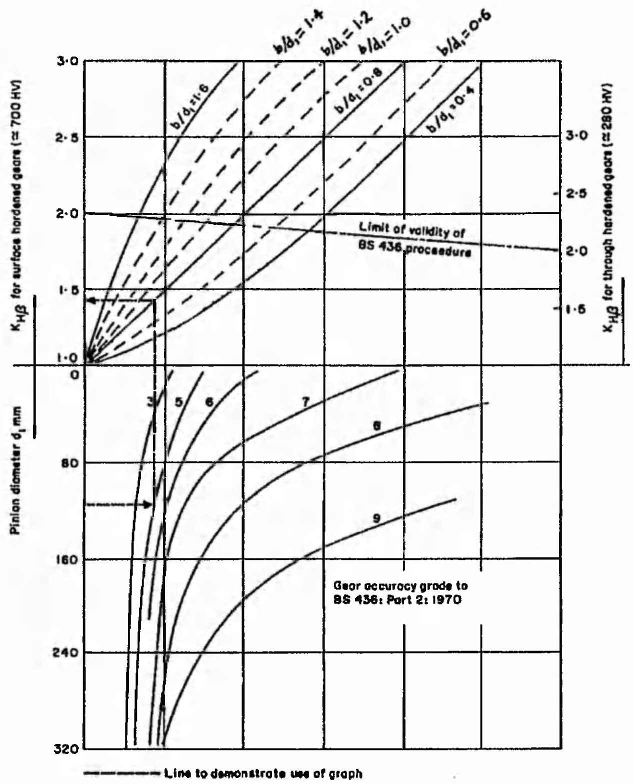


Figure 3.21 Graphical Design Aid for $K_{H\beta}$ to be Represented by Network (extracted from ESDU 88033)

A broad distribution of training cases within the limits of the graph is used during the preparation of the data. This helps ensure that the network will represent the entire region of the graph. The structure of the training data consisted of 360 cases, covering all the accuracy grades, the direct curves for facewidth ratio and the interpolated curve for b/d1=1.2. The test data consists of a combination of cases within the regions of the training cases and cases from the remaining interpolated curves. The performance of the network with regard to the interpolated test cases will demonstrate the networks ability to perform the application successfully.

Results of KHß Application

The results for the GEN-NEU training demonstrate that as the number of generations proceed the performance of the networks increase. Figure 3.22 illustrates the results, showing that the average error between the target and actual outputs from the network, decreases to a final value of 0.01753. At the same time the average number of successful outputs producing an output below 2% of the target increases to 11.333. These values indicate that GEN-NEU is converging upon an optimum and its general performance is encouraging.

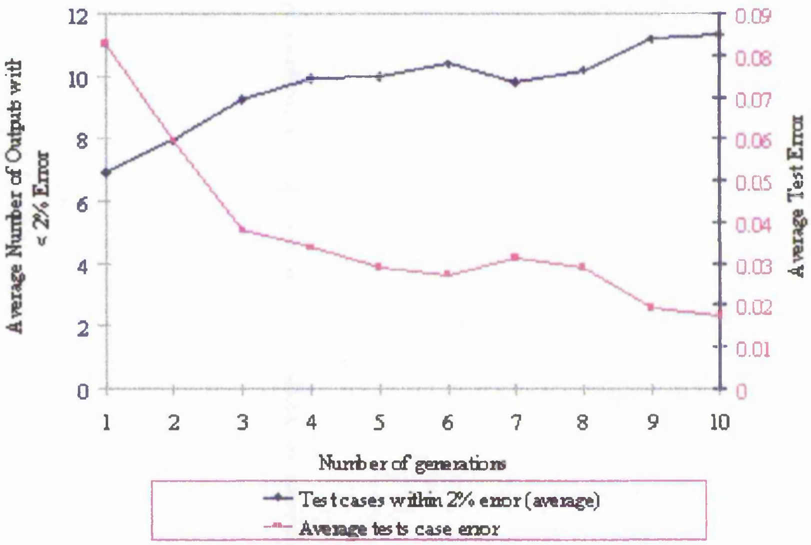


Figure 3.22 GEN-NEU Training Results for $K_{H\beta}$

Performance of the networks within the GA approach are determined by the average error between the network and actual results to the test set, giving a general fitness of the network.

To remove any uncertainty in the results that could originate from effects caused by the initial random weights at the commencement of training, the resultant network is tested individually. The final network configuration obtained by GEN-NEU comprises of 3 input elements, 2 hidden layers with 22 elements in the first, 12 in the second layer and a single output. A sigmoid transfer function is implemented and the network was trained for 5120 passes of the training set.

These results are presented in Figure 3.23 and Table 3.3. Figure 3.23 illustrates the proximity of the target and actual outputs from the network. It can be seen the general proximity of the two outputs are very close. Most importantly the proximity of the outputs to the test cases derived from interpolated curves, that network had never been presented during training are extremely close with a maximum error of 2.13 %. These cases are identified by dashed boxes surrounding the points in the figure and signify that the network is capable of interpolation, producing similar results to a designer.

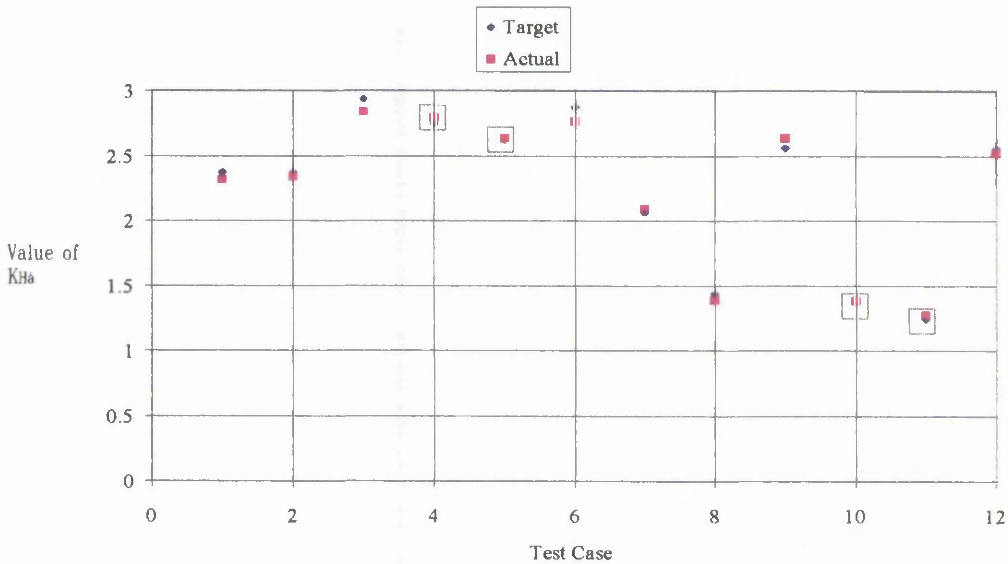


Figure 3.23 Test Results from $K_{H\beta}$

Input			Output				
				GEN-NEU		Professional II	
Pinion Diameter	Gear Accuracy	Facewidth Ratio	Target $K_{H\beta}$	Actual $K_{H\beta}$	% Error	Actual $K_{H\beta}$	% Error
9	3	1.6	2.3750	2.3195	2.3353	2.0707	12.8109
105	6	1.6	2.3750	2.3421	1.3860	2.3255	2.0828
207	9	1.6	2.9375	2.8429	3.2215	2.7464	6.5057
10	6	1.4	2.7500	2.7978	1.7369	2.6559	3.4213
70	7	1.4	2.6250	2.6350	0.3827	2.6813	2.1459
37	7	1.2	2.8750	2.7654	3.8137	2.7600	4.0001
159	8	1.2	2.0625	2.0982	1.7331	2.3495	13.9158
35	3	0.8	1.4375	1.3900	3.3028	1.3437	6.5260
71	8	0.8	2.5625	2.6374	2.9219	2.6392	2.9919
51	5	0.6	1.3750	1.3865	0.8384	1.4031	2.0433
320	7	0.6	1.2500	1.2766	2.1315	1.1092	11.2626
121	9	0.4	2.5625	2.5227	1.5513	2.2278	13.0623
				Average % Error	2.1129		6.7307

Shaded cells represent test cases derived from an interpolated curve the network was not presented during training.

Table 3.3 Comparison of GEN-NEU and Professional II Test Results

The performance of the network trained from the GEN-NEU program is verified by training the same network configuration under the same conditions with the commercial package Professional II. Table 3.3 contains the results and shows that the performance of the GEN-NEU network is slightly better than those obtained from Professional II indicating that the network configuration is in the region of an optimum. The variation in results can be attributed to the initial random connection weights values.

3.2.3.4 Concluding Remarks on GEN-NEU

The GEN-NEU approach to aid the training of backpropagation networks has proven to be capable of achieving successful results for a number of applications and input/output applications. As a result the laborious process of trial and error and the need for rules to determine a successful network configuration has been removed from the training process.

The implementation of the GEN-NEU approach into a program provides a tool that eases network training both while developing the Intelligent Integrated System and, most importantly once complete. GEN-NEU enables modification to information within the system to be performed by someone other than the system designer or with little ANN expertise. Therefore, GEN-NEU provides a facility for the easy alteration of knowledge, a process not easily performed with an expert system.

The inclusion of the training period into the GAs search has the added beneficial effect of preventing ‘over training’. Over training decreases the networks ability to predict and generalise outputs based upon the training data. As deviation from the test data increases when the network loses its ability to generalise, the fitness of the genomes causing this phenomenon will decrease. The result of the decreased fitness is to reduce the occurrence of this training period in the search.

Although the training data is not included within the GA the effect of the amount of training data is compensated for by both the training period and network topology. As the training period is flexible and both small and large network topologies can be generated a combination of these factors are adjusted to model the data.

GEN-NEU is a general approach, which can be used with variations of the backpropagation training technique and other supervised ANNs that require adjustment of several indirectly related factors.

3.3 Production Rule based Systems

3.3.1 General Structure and Process

The production system comprises three components; a database, a rule base and a rule interpreter. The database contains information about the problem, possible solution and related subjects, while the rule base consists of a set of rules, relating to the problem. The database is in the form of a series of text files, structured into fields that can be accessed by a search engine. Information is retrieved or deposited to a position that corresponds to parameters identified to the search engine by the rules. These rules are sometimes called production rules and represent general or specific knowledge about the problem. This knowledge is of a well defined structure, comprising of an action or process which is activated from a particular situation. The general form of these rules is that of an IF...THEN statement. i.e. IF x THEN z

or IF x & y THEN z

A connection is made when activating the rules between the production rule and the database in a variety of forms, depending upon the purpose of the rule. The mentioned database contains the information about the problem, therefore, the rules compare the information within the database with the situation and takes action based upon it, as in the Figure 3.24 below.

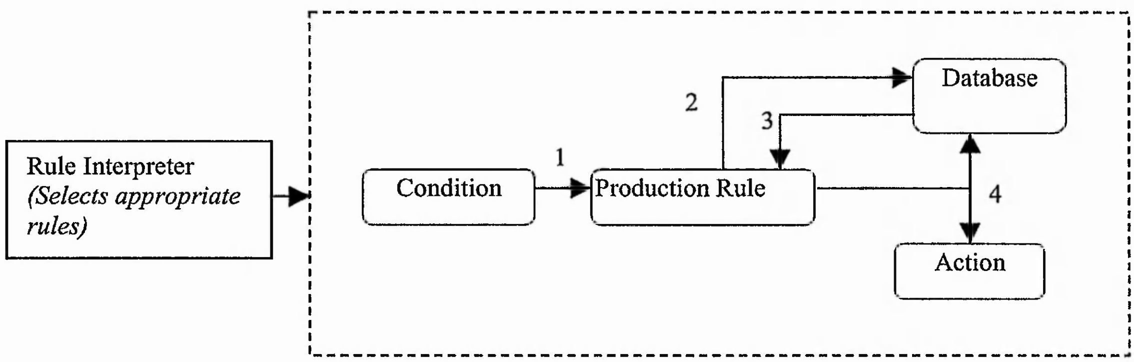


Figure 3.24 Production Rule Components and Action

An example of the Production Rule Based process is for the identification of the number of transmission stages that are required for a parallel arrangement transmission with 8 to 1 speed reduction.

The rule interpreter will select an appropriate rule to determine the number of stages for a transmission system for the given conditions. The arrangement and the speed reduction form the conditions of the rule. A rule that complies with these conditions is then activated and its actions performed.

i.e IF *orientation is parallel* AND *ratio > 7* AND *ratio <= 30*

THEN *transmission has 2 stages*

The action of this rule is to set the number of transmission stages that will be required for the speed reduction. The action is contained within the action (THEN) segment of the IF.. THEN statement and the reaction stored for reference. The resulting action for theses conditions is to identify that 2 transmission stages will be required.

Activation of the relevant rules are determined by the rule interpreter. In a typical production rule system the rule interpreter cycles through the rule base, comparing the conditions of the rules with the information from the database. Once a match is found the rule is activated and the action is invoked. The result of this action could modify the database or activate a process, depending upon the rule.

There are two methods of rule interpretation and selection that can be employed by the rule interpreter. These methods refer to the aims that drive the selection, either data-driven OR the goal-driven. Each method selects rules from the rule base, differing only by their perspectives.

<u>*Data-Driven*</u> selection, also known as non-deterministic selection or forward chaining, comprises of three steps:

1. Evaluate the conditions in all the rules within the rule base and assemble a smaller sub set which is applicable to the problem.
2. If no rules match, terminate the process. Otherwise select a rule from the sub set at random.
3. If the goal has been met, terminate with success, else repeat step 2.

Goal-Driven selection is the second method of rule selection employed by rule interpreter. This approach performs the data-driven process in reverse, as the sub set of rules, applicable to the problem, are selected by considering their solution instead of their condition. It is for this reason that the process is termed goal-driven.

The use of either method of rule selection is dependant upon three factors; the application to which the production rule system is applied, the form of the knowledge held within the rule and the information available. For example, if the solution to multiple conditions is required, the data-driven approach is best suited. However, if the action is known and conditions or cause is required the goal-drive approach is most appropriate. These rule types are illustrated below in Figure 3.25.

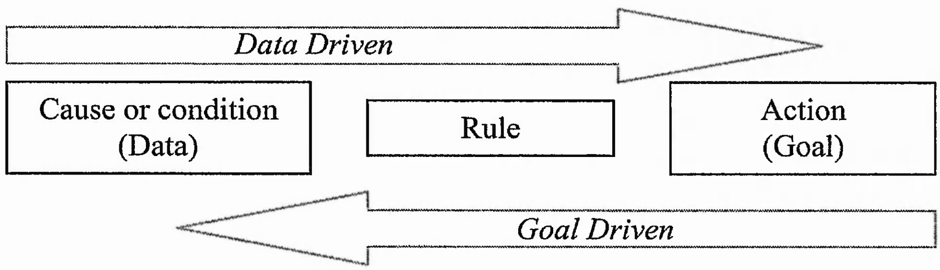


Figure 3.25 Orientation of Rule Types

3.3.2 Production Rules Applied to the IIS

The production rule based system, in its true sense has not been used, but instead structure has been imposed within the system to direct purpose and increase efficiency. The production rules have been used for a duel purpose. Firstly, to structure and control the development of a product through the design process and secondly, as a source of knowledge retention and relating to different elements of design.

The knowledge contained within the rules does not form a separate rule base in the sense of a conventional expert system sense. Instead the knowledge is 'a collection of simple facts together with general rules representing some universe of discourse', Frost (1986), in the form of a series of production rules constructing a complex cognitive system.

The use of production rules and a control procedure presents a number of advantages over a conventional expert shell and knowledge base for this application.

- The use of production rules does not require an external shell therefore, simplifying and speeding up the system.
- The design process can be expressed in a structured fashion.
- Certainty values and firing methods do not require manipulating.
- Interfacing with the rest of the system is simpler.

Within the project the data-driven approach has been adopted. This is due to the form of the design process adopted for the project. As the design progresses information is accumulated, describing the finished product. Thus, the conditions are known and the relative action or solution is required.

The production rules are in two forms. The control rules which are concerned with controlling the progress of the design (if a condition is met perform a task) and information rules which contain specific data or information about a particular design configuration or property. The activation of the second type of rule is governed by the controlling rules.

The control rules hold the knowledge relating to the designs progress. These rules structure the design process and control its development, forming the inference engine, activating the appropriate information rule depending on the circumstances applied and the stage of the design.

Example

IF *facewidth ratio is not defined by user* THEN *activate facewidth information rules.*

The control rules not only control the information rules they also control the activation of other modules within the design systems.

Example

IF *gear is through hardened* THEN *activate ANN module 1.*

The information rules contain information in the form of numerical values, design features or simple equation that encapsulate the design information.

Examples

IF *the gear is double helical and its heat treatment is nitrided and its mounting is symmetric* THEN *maximum facewidth ratio is 1.4.*

IF *heat treatment is case carburised* THEN

$$max.\ number\ of\ teeth = \left(\frac{11}{9}\right) gear\ ratio - 37.33$$

The information rules contain specific information relating to an area in the design process and are grouped together in sub-sets. This grouping aids the modification and maintenance of the knowledge stored in the rule base.

The use of production rules for both control and knowledge representation forms the production rule based system employed within the project. A system of this form can be molded and fashioned to suit the individual purpose of the application while maintaining a general structure. This enables a combination of AI techniques to be integrated with one another, thus forming the hybrid intelligent system and increasing efficiency and direction of design.

3.4 Multimedia Interfaces

Multimedia, in the form of graphical user interfaces (GUI) has been used throughout the IIS to present and extract information to and from the user. These interfaces generally provide a link with the user to receive information, describing the design specification of the product that the user wishes to produce or present information to aid decision making based upon knowledge that the IIS could not encapsulate.

Multimedia Development

The development of multimedia interfaces is split into three broad stages: design, layout and production. The design and layout stages require approximately equal time and resources. It is more efficient to spend time on establishing the purpose and structure of the interface form the concept if it prevents confusion for the user of the end product

The design stage begins by defining the purpose of the interface. Defining the interface's purpose enables its structure to emerge, identifying related parameters and orders that they should be extracted from or presented to the user. The order of the parameters is a critical factor in the design of the interface. The sequence of extracting information from or presenting it to the user should be ordered in such a way the user is not required to concentrate too hard. An effective interface should be simple and easy. This first step is crucial to the success of the interface, as the rest of the development is built upon it. To aid the structure of the interface and help it achieve its purpose a storyboard of the application is created. The size and complexity of this storyboard depends upon the application. The storyboard is a route map of the interface, just as the flowchart is used for conventional linear programming. Within the storybook the purpose of the interface is broken into sections. These sections generally relate to individual screens or with large applications tasks, which contribute to the final goal of the interface. For each section the aim is defined, the parameters used within and linked with past and future sections noted. The storyboard used for the development of the GUIs within this project uses the table in Figure 3.26 below.

<table>
<tr><th rowspan="2">Section /Frame</th><th rowspan="2">Aim/Task</th><th rowspan="2">Linked Frames</th><th rowspan="2">Parameters Used</th><th colspan="2">Linked Parameters</th><th rowspan="2">Graphics, Video and Sound</th></tr>
<tr><th>Previous</th><th>Next</th></tr>
<tr><td rowspan="2"></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="6">Comments</td></tr>
<tr><td rowspan="2"></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="6">Comments</td></tr>
</table>

Figure 3.26 Multimedia Storyboard

In addition to the layout of the screen the use of graphics should be exploited. Immense amounts of information can be presented graphically, allowing the presentation of information to be simplified, but more importantly the human brain can recognise and classify images better than raw data. Therefore, a series of images with the option for detailed information is an effective means of aiding the user to make complex decisions.

The final stage of the multimedia interfaces development is its production of the source code. Provided the storyboard has been created with sufficient information within it, relating to purpose of the application and the parameters, generation of the finished interface should be simplified and the finished product clear and concise.

The concept that should be considered at each stage of an interface's development is that the best interfaces are those that require no explanation.

3.5 Discussion and Summary of AI Techniques and Multimedia Used for the Intelligent Integrated System

The techniques discussed within this chapter have been described with regard to their technical application to the intelligent integrated system. The purpose of the techniques, the technical aspects and customised development are discussed. The two main topics discussed are the application of evolutionary programming and artificial neural networks. This is due to their relatively recent introduction to the field of design, in comparison to rule based systems and multimedia.

The application of GA to the project has lead to the development of a combination mutation operator in an attempt to maximise performance. This operator combines both the standard bit transfer operator with a total genome operator (developed for the project), in an attempt to increase GA performance. Additionally analysis of the effects of population size, genome size and construction, fitness function and convergence have lead to the greater understanding of the evolutionary process and how it can be applied to design applications. The analysis identified five main points when developing a GA application;

i. Correct identification of the target goals of the optimisation is essential
ii. Minimising the size of the genome through the correct use of encoding techniques can simplify the search and thus reduce computation expense
iii. A population of sufficient size is required to ensure that global optimums are found
iv. Repeatability is reliant upon achieving the same result repeatedly, thus the rusult attained can be considered the global optimum. This installs confidence in the results, necessary for industry

v. Use of fitness scaling functions ensures that pressure is constantly applied to the search and aid repeatability by maintaining an unevenly weighted roulette wheel throughout the search

The application of the BP network to the project lead to an attempt to find a means of easing the training process. Analysis of the effects that the topology, training data, training period, transfer function and learning and momentum coefficients have upon the performance of the network was conducted. This resulted in a flow chart that can aid a trial and error approach to training, but no general rule to aid in training being found. For example, a rule for determining the number of elements and network structure irrespective of the application. However, the lack of training rules lead to the development of the GEN-NEU program. This novel program removes the necessity for training rules and replaces it with an adaptive search process controlled by a GA. The result is a BP network training tool that requires no knowledge relating to network training, only requiring the input and output structure, training and test data and the number of generations to limit the search. This tool thus aids in the easy modification and update of knowledge that is to be encapsulated within the intelligent integrated design system, one of the project's objectives.

CHAPTER 4

INTELLIGENT HYBRID APPROACH TO DESIGN AND INTELLIGENT INTEGRATED SYSTEM DEVELOPMENT

The intelligent hybrid approach developed within this project incorporates the design process from concept design to the complete generation of assembly drawings. Within the approach the process of design has been defined with regard to automation and encapsulation of the knowledge through a hybrid of AI techniques, conventional programming and multimedia.

The approach is of a general form, applicable to the engineering design process for all products. The following sections describe the approach and its core components, together with the combination of AI techniques, CAE and multimedia and their methods of applications.

4.1 System Approach

The intelligent hybrid approach draws together all the design stages, knowledge and expertise required to perform the design process for a product all within a single environment. The single environment is the key to the approach, allowing the product design to progress from stage to stage without the duplication of information, repetitive requests for the designer and dependence upon several experts in different fields of engineering. The approach combines all these elements together through the integration of AI techniques and multimedia within the system forming an intelligent integrated system (IIS). The combination of AI and multimedia provide a hybrid medium for the extraction and encapsulation of knowledge with the intent of replacing, guiding or tutoring the designer using the finished system.

The strategy for developing the IIS consists of three inter-linked parts. The first is concerned with the design process and the stages the product must pass through during its development. The second relates to the knowledge acquisition and categorising process for identifying the components of the integrated system and implementing them to the design process. The final part relates to the modular structure of the system, ensuring ease of knowledge alteration.

4.1.1 Design process

According to Pugh's (1990) Total Design Activity model and French's (1985) model, which are widely accepted by most researchers and designers in the area of Engineering Design, the design procedure can be modelled into 5 basic design stages: i) identification of need, ii) specification or requirement, iii) concept generation, iv) detailed design and v) manufacture. Identification of need is not considered as a design stage in the same sense as the other stages with regard to developing a product. This is due to this stage being the realisation that there is a requirement for a product to perform an application. It is therefore not a design stage with respect to the IIS, but is regarded as a starting point for the purpose of this approach. The remaining design stages are considered individually within the IIS as they contribute to the final product, forming stages in its development. However, the design process is more complicated than just proceeding from one design stage to the next. Knowledge and experience is required about what designs are possible and the procedures and techniques required too arrive at feasible results. Acquisition of knowledge, expertise and information that enables these decisions to be made represent the vast majority of problems when developing a design. This knowledge is typically based upon the experience, judgement and personal preference of the designer, attributes which are all difficult to quantify.

The design process that the IIS follows is very similar to the majority of traditional design procedures and incorporates their fundamentals. All designs start from an identification of a requirement that progresses into the generation of a basic solution or concept. This concept in turn develops through further consideration and modification into a final detailed solution or design of a product, as illustrated in Figure 4.1.

Although the IIS design process follows the traditional basic approach, the basic approach does not clearly indicate the form of the actual expertise, knowledge and information needed to transform the initial requirement into the final product design. This consideration, with respect to the conventional design process, is assumed to be performed by the designer. However when incorporating the design process into a computer system, opinions, rules of thumb, experience and common sense, which forms the majority of the design process, cannot be assumed to have been incorporated and thus requires greater consideration.

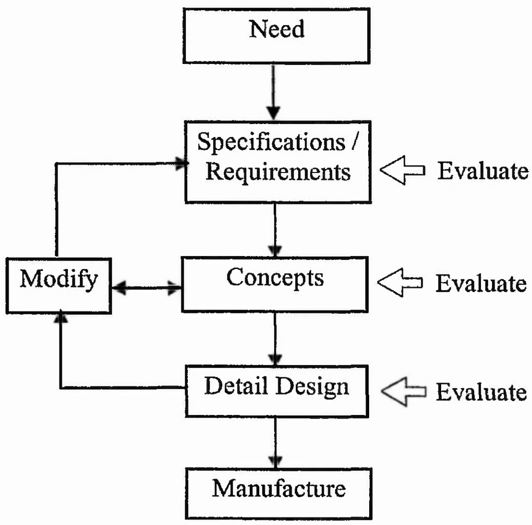


Figure 4.1 General Design Process

The computerised design strategy developed, within this project, takes into account these attributes and incorporates their effect on the design into the design process. Firstly, an analysis of the problem is performed, bearing in mind the needs that initiated the design process, producing the problem statement. This helps to generate general solutions that act as building blocks that combine to form the concept designs for variations of the initial problem statement. Information about the design considerations and limitations regarding the combination of the building blocks is accumulated in preparation for the system development.

The design process is followed through to the detail design and modification stages, where information is accumulated and design avenues investigated. Figure 4.2 illustrates the generalised, basic design analysis procedure, which can be applied to the design process of any product.

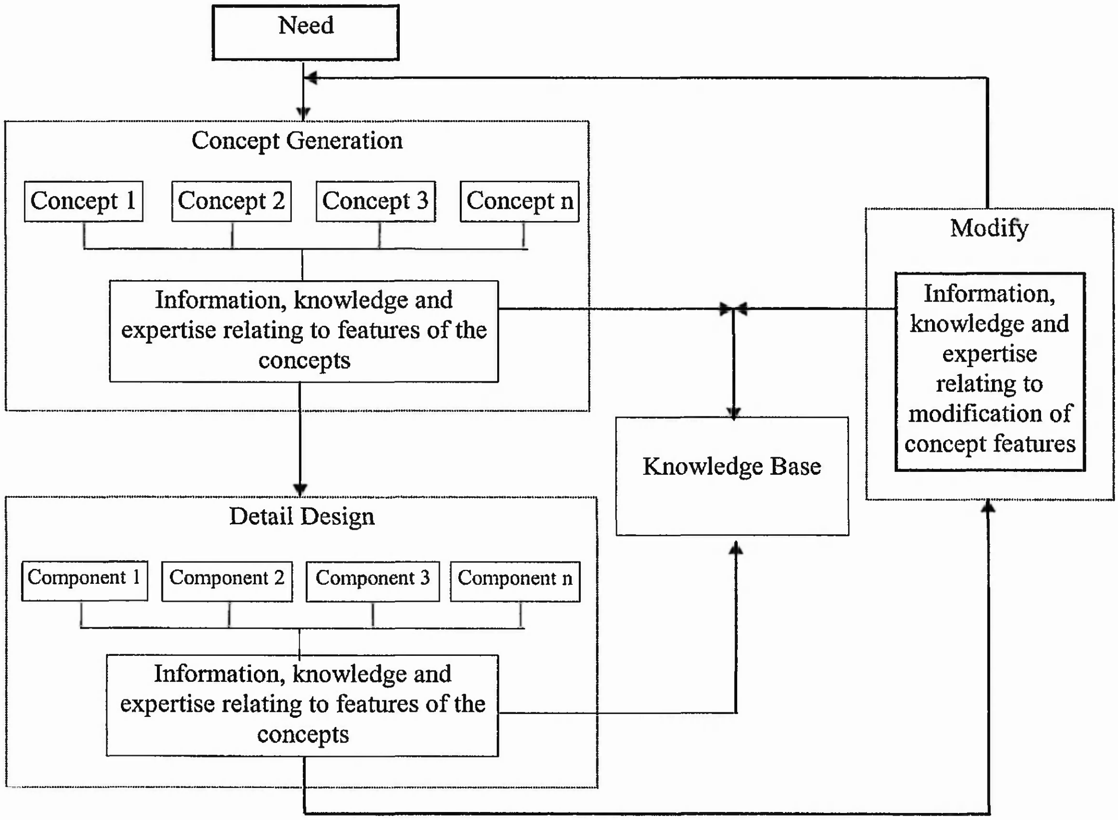


Figure 4.2 Design Procedure Analysis

4.1.2 System Development

During the development of the system, three main areas are progressed through: specification definition, generation of solutions and analysis of solutions. The initial starting position, (the identification and analysis of the general problem to determine the product to be designed) has been performed within the Design Process section of the methodology, section 4.1.1, when the design need is identified and problem statement produced.

Specification definition

Definition of specifications describing the design are determined on two levels: the general specifications, which relate to the requirements of the design and its purpose, and detailed specifications, which define actual physical properties. The specifications are determined at two different stages of the design analysis, but can be updated or modified throughout the process. The general specifications are derived directly from the problem statement considering the requirements and qualities that the final product will posses for all eventualities. These are defined at the outset of the design process, describing the purpose of the product from which the concepts are generated. The detail specifications are determined throughout the analysis and detail design derived from the concept, providing specific technical information relating to the individual application of the product.

Generation of Concepts

Concepts are derived from the basic concept designs that the final designs are based upon or built up from. These concepts form the building blocks that the design will comprise and should cover all eventualities the system will encounter. However their structure should be limited, providing only the rough outline. The final conceptual design should combine one or more general concepts integrated together.

Analysis of Solutions

Analysis of Solutions forms the main area through which the development of the system will pass, involving the analysis of possible solutions in order to identify and establish the knowledge required and the variations the designs development route could take. The structured acquisition of this information is illustrated in Figure 4.3.

Analysis covers the components and sub-assemblies that comprise a solution, whether for the conceptual or the detailed design stages. Sub-assemblies are in turn broken down to their composite components before detailed analysis is performed. Each component is then analysed, indicating the requirements necessary to enable complete design while the various areas of expertise are identified.

Once a design process of the component requiring knowledge has been identified the knowledge relating to it is categorised into one of three forms, well defined, data intensive and ambiguous. Categorisation of the knowledge will determine the means of encapsulation within the system.

- *Well defined* knowledge is normally encoded into production rules as for a known circumstance a particular action or result is given. e.g. If A then B.

- *Data intensive* knowledge can be encapsulated in two forms depending upon the raw data. If the original data is available the most appropriate means of encapsulation is within a database. However if the original information is not available or incomplete ANNs provide suitable storage capabilities.
- *Ambiguous* knowledge, which is based upon experience, is encapsulated within production rule form for simple, limited rules of thumb. However if the data is extensive, jumbled, incomplete and requiring additional judgement based upon similar circumstances the ANNs provide an excellent form of encapsulation. Additionally when neither of these techniques can cope, multimedia is used to help prompt the designer for the appropriate solution.

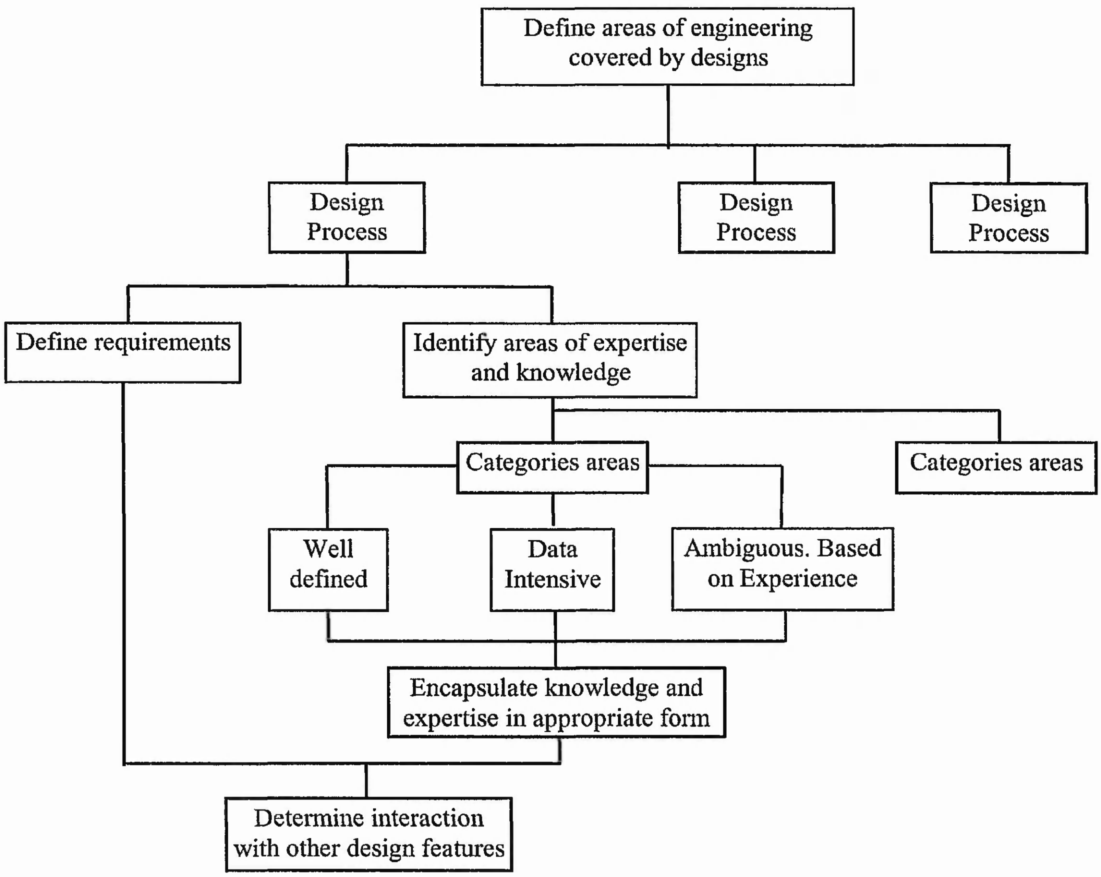


Figure 4.3 Knowledge Analysis Process

During the identification of the knowledge required, identification of requirements will be established. These requirements are either the result of decisions made by previous stages of the process or specifications required from the user.

Once the design route, knowledge, requirements and means of encapsulation are established, the development of the IIS may proceed.

4.1.3 Modular Structure of the Integrated System

The integrated system adopts a modular approach to its structure. A module is a self-contained stage or element of the design process that performs a part of the designs development. The modular structure enables areas of knowledge and expertise to be modified, replaced or removed from the system without consulting the system developer or requiring intricate knowledge of all the modules purpose within the entire system. Together with the advantage of easy modification the modular approach allows for simpler integration of external stand-alone design packages, which forms the essence of the IIS system.

The modular structure and approach is illustrated in Figure 4.4. From the figure it can be seen that a hierarchy of control develops as the system develops. This aids the structure of the system and provides levels of responsibility, with the lower levels being controlled and answerable to the upper levels.

The central controller structures the designs progress at the highest basic level, determining which design stage to initiate (either for the first time or for redesign and modification) and to define the final design. The modular approach is maintained throughout the system being implemented to different levels as and when required for each stage.

The conceptual design stage adopts the modular approach to structure the knowledge and information for different aspects of the concepts. Areas of knowledge relating to different concepts and methods of encapsulation are spilt into separate modules to allow ease of modification and initial system development. Section 5.2 details how this stage has been implemented for the design of a mechanical transmission system.

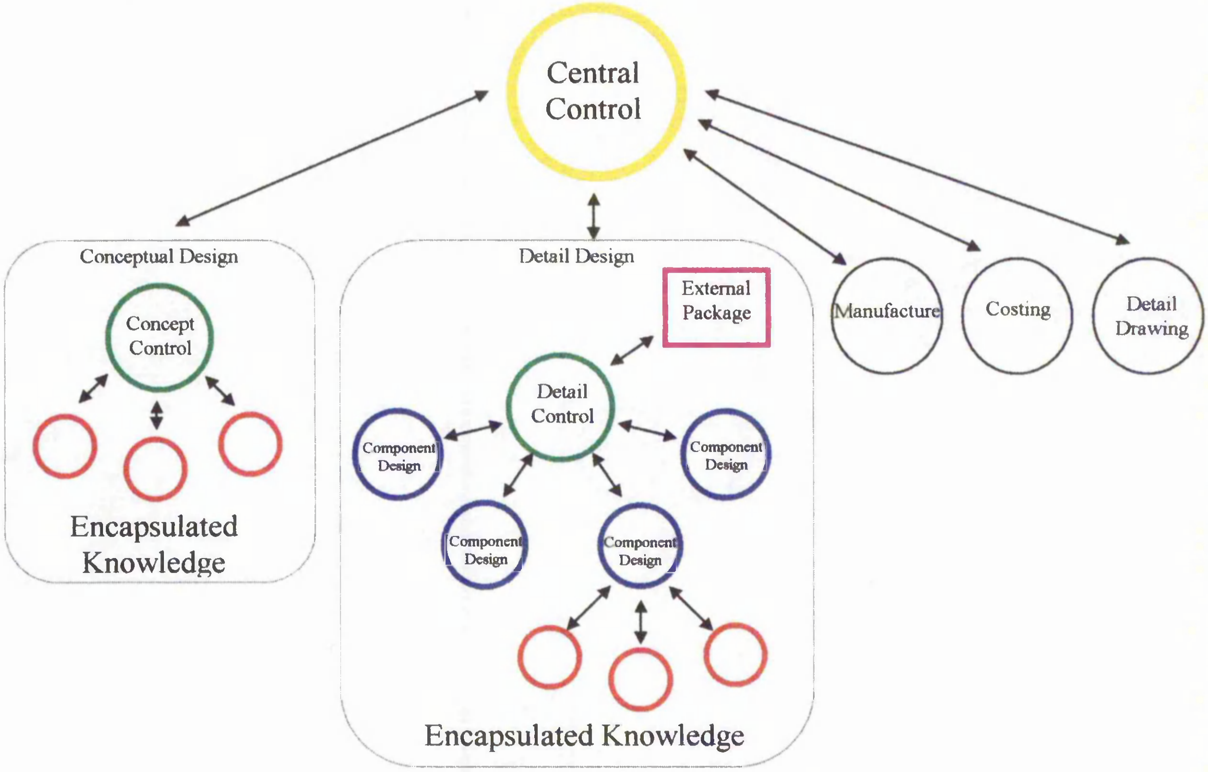


Figure 4.4 Modular Structure of IIS

Detailed design exploits the modular approach to the full, forming many sub-designs relating to components within the overall design. Each sub-design module encapsulates the knowledge required in separate knowledge modules locally. This stage of the product design process is extremely information intensive and represents the area of design where advances in materials, manufacturing processes and analysis techniques will require update and modification if the system is not to become obsolete.

The detailed design is controlled by its own control module, initiating the appropriate design module or external package depending upon the progress of the design. The results of each design module are evaluated by the detail design controller which passing the necessary information back to the central control when detailing is complete or when re-evaluation of the concept is necessary. Each of the design modules performs its task individually, calling upon knowledge modules relating to the task. These modules may take the form of encoded tables, ANNs or production rules.

The final design is achieved by modification and iteration of the design until the design meets the requirements of the problem statement while complying with specifications and limitations.

4.1.4 Implementing IIS Design Methodology

Implementation of the design methodology to develop a design system combines all three sections of the methodology simultaneously at periods and follows the complex path illustrated in Figure 4.5.

Breakdown and interaction of method parts :

- Part 1, the definition of the design process covers stages 1, 2, 3 and 5 in the diagram
- Part 2, the development of the IIS process consists of stages 2, 3, 4, 5 and 6.
- Part 3, definition of modular structure requires interaction with the other two sections and covers stages 2 through 7 in Figure 4.5.

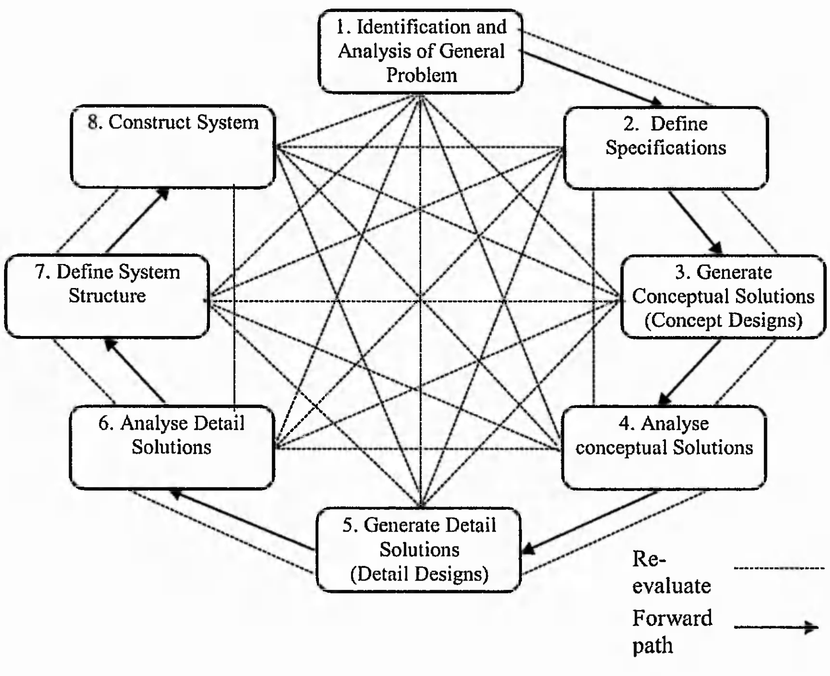


Figure 4.5 Interaction of Development Stages

With respect to Figure 4.5, the development of the system proceeds in a clockwise direction, progressing through the various stages. However re-evaluation of the system may be performed at any point in the process, with modifications to any lower stage being permitted. There-evaluation routes are indicated by dashed connecting lines

The final system is developed in stage 8, but as with all the stages below, will arrive at the final product, only after continuos update and modification.

The interconnection of the data and techniques used to generate the design forms the data transfer structure. The data relating to the various stages of the design is encapsulated in several forms and therefore the ability of each technique to interact with each other is essential for the IIS. The environment in which the system exists must therefore allow and help interface with all data manipulation techniques and external influences including those of both the user and existing design packages.

Figure 4.6 illustrates how the IIS approach combines the modular structure and methods of expertise representation. Additionally the software techniques that are employed are indicated together with the controlling structure of the IIS. It can be seen from the figure that the modular approach is maintained by the stage controllers together with the control routes that are represented by the connecting lines between controllers

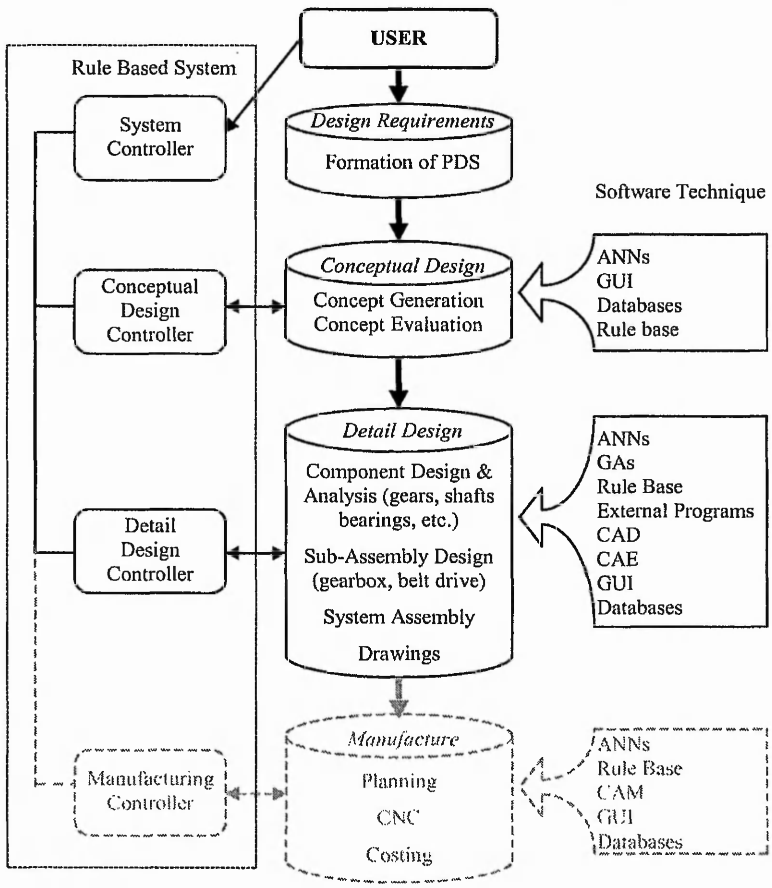


Figure 4.6 IIS Approach and Environment

The grey illustrates how the manufacturing stage of Pugh's total design process can be integrated within the IIS.

4.2 Intelligent Integrated System Controllers

The function of the System Controller is to structure the design process, determining the next process to be performed and assessing the current state of the design. The controllers guide the design through the relevant stages of the design, implementing redesign as and when required. The control of the design process is performed by the System Controller. This controller is at the top of the hierarchy and guides the designs development from one design stage to the next, assessing if problems are occurring with the design and implementing complete redesign if required. The System Controller delegates control of the design process to Stage Controllers upon their activation and regains control upon completion of the design stage control prior to assessment and activation (or re-activation) of the designs next stage.

The Stage Controllers are sub-controllers with respect to the System Controller, positioned in the next step down the control hierarchy. When activated the Stage Controller interfaces between the design modules and the design database. The Stage Controller continually assesses the designs current situation, determining which design module to activate then retrieving and preparing the information that is required for the task. Upon completion of the module the status of the design is re-assessed by the controller, which in turn updates the design database.

The Stage Controllers are the means by which the IIS combines the various aspects of the design process, enabling the external stand alone packages to be combined with purpose built programs. The controllers thus form a single environment in which information and knowledge can be readily transferred between separate stages of a designs development without interruption.

The controllers comprise of a combination of conventional programming and production rules, which guides the designs development. This combination of programming techniques is most suitable to the IIS application, as the development of the design requires structure and reasoning together with a hierarchy of control. These qualities can easily be represented using this combination. The conventional programming portion of the controller provides the interface through which the system will run, enabling the activation of the rule base corresponding to the progress of the design. In turn the appropriate design module is activated and the preparation of information to and from the database performed.

4.3 Design Modules

A design module typically comprises of a design process that determines the type of component or its geometric and physical properties. These modules form the physical building blocks of the system, just as the conceptual designs form the building blocks of the product design. Within each design module the knowledge and information necessary to perform its individual task is contained. It is this modular approach that allows the information and expertise that is contained within the system to be easily modified. This is achieved by modifying the information that is contained within that individual module. Provided the interface with the relevant Stage Controller is maintained the performance from the remainder of the system will not be affected by the modification.

The essence of the design module is illustrated in Figure 4.7 below.

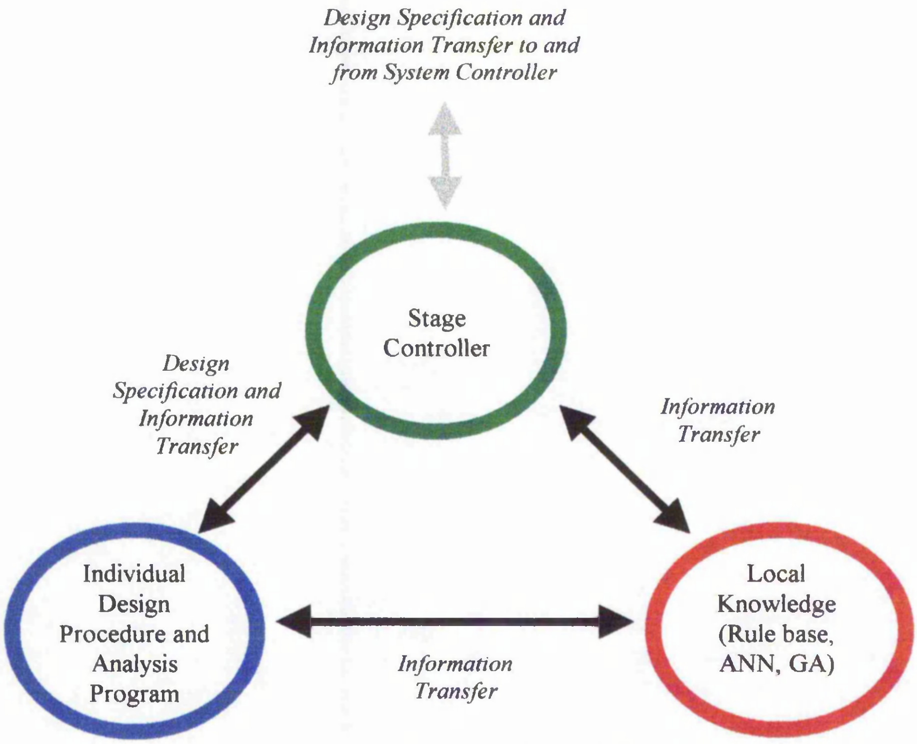


Figure 4.7 Basic Design Module Process

The modules structure generally comprises of three parts; transfer and preparation of relative design specifications and solutions, calculation and analysis procedures and finally, the extraction and application of knowledge. The amalgamation of these parts forms the design module, emulating the design process of the expert designer, (generation of specification, develop the designs and applying knowledge and expertise as required).

4.3.1 Information Transfer

Information transfer provides the vital link between the Stage controller of the IIS and the design module. Thus forming the medium through which the specifications of the design are transferred. The transferred information comprises constraints that determine the characteristics, qualities and performance of the component design. Upon completion of module the properties of the finished component design are returned to the Stage Controller, which in turn updates the product's design and re-assesses its progress.

During the development of the IIS identification of appropriate information relating to the product design is essential. This information should describe the component succinctly, establishing the essential qualities of the design. This forms the general specification, common throughout the design. Additional information is then included to describe a particular type of component. The structuring of the information enables modifications to the system to be performed with greater ease, thus increasing the versatility and robustness of the approach.

The transfer process is achieved through the use of swap files. The files use the ASCII format, enabling robust and reliable transfer of data between programs developed in a variety of computer languages, thus ensuring compatibility of the various design packages and integrating the module within the single environment of the IIS.

4.3.2 Calculation and Analysis

The procedures that the designer will conduct to produce the components design is performed by a program. The program structures the components development in a similar fashion to the Stage Controller, activating calculation procedures to determine the properties of the component based upon the information received form the Stage

Controller. This program extracts knowledge from the various forms of storage and manipulates it together with the specification generating the components properties.

In the case of the integration of an external package the part of the module is replaced by the package. Thus the purpose of this section will be to activate the package once the information from the Stage Controller has been prepared. Upon completion of the package the result is retrieved and prepared for return to the Stage Controller.

4.3.3 Knowledge Representation within the Design Module

Representation of knowledge is achieved through the use of AI techniques, databases and GUIs. During the development of the design module the knowledge required to perform the task is accumulated and assessed according to its form and purpose. Categorising the design knowledge into its relevant group; well defined, data intensive and ambiguous (as mentioned in section 4.1.2) aids in determining the appropriate method of encapsulation. It is at this point of the systems development that a combination of techniques is often used, enabling the weaknesses of one technique to be compensated for by the strengths of another.

4.4 Design Optimisation

To enhance the quality and performance of the design of the individual component, refinement of an initial design is required to exploit its full potential. This is achieved through optimisation. The optimisation process can be performed by a variety of techniques, such as Newton-Raphson, hill climbing, gradient descent and simulated annealing. However these techniques are limited when the modification of multiple parameters is required and when the constraints of the search are complex.

The AI technique, evolutionary programming (EP) incorporates a genetic algorithm, (an adaptive search technique) capable of modifying several parameters simultaneously in search of a global optimum. This process therefore displays the necessary properties required for component design and has been applied within the IIS to improve the quality of design. The use of EP provides additional benefits for the IIS with regard to the retention and formulation of design knowledge. EP is explained in greater detail in section 3.1 and Appendix A.

4.4.1 Evolutionary Process for Design

The application of EP allows the search parameters (parameters to be modified) to be defined in terms of the goals to be achieved. This is achieved by the use of fitness functions (section 3.1.5).

Prior to the application of EP to the optimisation of the design, the goal of the optimisation must be established and the parameters to be modified determined. Stepping back from the problem and studying the effects of various parameters upon the goal enables the search parameters of the GA to be defined with greater confidence in their results. The investigation does not require a vast amount of knowledge to be accumulated about the application, (this will partly be dealt with by the GA as discussed in the next section), but sufficient information about the application is required to define the goals and parameters to be adjusted. Therefore the initial stages of applying EP to design optimisation are:

1. Identify the goals to be achieved for the final design.
2. Identify limitations of the design with regard to failure or known reduction in performance.
3. Determine the parameters to be modified by the search process.
4. Determine form and limitations of parameter values.

These four steps aid in the successful application of the EP to the IIS and design.

4.4.2 Knowledge Representation Using the Evolutionary Process

The intention of applying the evolutionary process is to remove the necessity for detailed analysis of the inter-related effects resulting from the adjustment of parameters, which would require the advice and input from an expert in the particular field. The GA provides the medium through which a search of the numerous configurations can be performed with the minimum of external guidance. Using the evolutionary process it is possible to modify multiple factors without constraints or inter-relational rules. Relation rules will be considered (for the purpose of this project) as rules that relate modifications of one parameter to an adjustment of one or more related parameters by a definable amount. These are generally in the form of production rules. For example, modification to parameter A requires a modification to parameter B by amount k if greater than 10. The generation of such rules requires expertise and understanding of a particular application, sometimes to a level that is not available or practical. Instead the process only requires fitness criteria which defines the desired result and possible limits upon the search

4.5 Design Retrieval and Selection

A standard design will have several variants or derivations, each modified slightly to perform the desired task in a different way. For example selection between the various types of bearings. Selection of the appropriate design variant, for a particular application requires the expertise and experience of a designer, who uses his judgement to identify the appropriate design. This selection takes into account many factors ranging from physical constraints to aesthetics to personal experiences and preferences. Design retrieval is therefore a complex, difficult to define process, requiring selection based upon tangible and non-tangible decisions. For example, several variations of standard bearings are available, such as deep groove ball bearings, angular contact and spherical bearings. Additionally these variants have their own variants including seals, lubrication and performance variations. Selection of the correct bearing for the application can result in an increase in the performance of the design, its life and collectively the quality of the final product.

For the IIS to produce designs of high quality, comparable with those generated by conventional processes, it is necessary for the system to be capable of selecting variants of a basic component design. Selection of components variants are finishing touches to a design, transforming it from a standard, elemental design into a design of quality that is tailored for the design application. Therefore the design retrieval process must be incorporated within the IIS. This requires a means of encapsulating the designer's experience, preferences and judgment in a form that can integrate with the IIS.

In the search for a method of design retrieval, both rule based and neural network approaches have been considered. These approaches have been used for the encapsulation of expertise many times, and possess the qualities necessary to perform the task.

Production rules within the rule base comprise of an action for a given condition. To generate the set of rules that describe the process a designer performs would require detailed information relating to the selection procedure and the exact criteria for each selection. Accumulating all the necessary knowledge would be time consuming, complex and laborious, as a rule must exist to describe all eventualities in order to prevent the process from failing to produce a solution. A rule base of this size would be difficult to modify, due to its size and complexity. Also as the rules are fixed the production rule approach to design retrieval does not possess the capability to adapt and refine its selection to meet the requirements of a particular application of the IIS.

ANNs have previously been reported to of been used for the retrieval of designs and components by Peters (1992), Escobedo et al (1993) and Caudell et al (1994), as mentioned in the literature review. However, these examples of design retrieval differ to the application dealt with in this project. They are concerned with selection of designs based upon similarities in geometry, while within the project designs will be selected based upon their ability to fulfill its requirements.

The application the networks will be applied to select the appropriate variation of the design based upon the requirements of the product and the purpose the design is being set to. Using ANNs the necessity to evaluate the selection process with regard to the experiences and preferences is removed, due to their ability to construct their own rules and relations.

4.5.1 Design Retrieval and Selection Method Applied Within the IIS

The design selection process, as mentioned, is a complicated process based upon knowledge that cannot be quantified easily. Therefore, due to the lack of structure and the fact that the selection of designs based upon experience and preference a rule based approach would result in an incomplete and inflexible encapsulation medium. Additional difficulties that would arise from modification to the rule base suggest that this technique would not be appropriate for this application. However the ability of the ANNs to develop their own rules and heuristics based upon examples make their use in this application ideal. Therefore the process of design retrieval / selection will be performed by an ANN approach.

4.5.1.1 Selection of the ANN for Design Retrieval

The type of ANN used to select the design variant, must posses two essential qualities. Firstly, it must be capable of categorising the requirements and allocate a design to this category. Secondly, the ANN must have a structure that can be modeled to suit the IIS and retain this structure regardless of the circumstances during training or application.

Categorisation, with respect to design selection, involves the identification of a suitable component with respect to the design requirements and physical constraints imposed by the system. Allocating network inputs to an output category is a quality that many network paradigms are capable of and been applied to. The most commonly used networks for this application are the Adaptive Resonance Networks, i.e., ART1 and ART2 by Kumara & Kamarthi (1992), Peters (1992), Escobedo et al (1993) and Caudell et al (1994) and self organising feature map networks as used by Kiernan et al (1995). This is probably due to the way in which the networks adjust, matching a combination of input parameters with an output. In the case of the ART networks a new output category (or class) is created if no suitable output exists and have proved that they are well suited to classification. The backpropagation network has also been used successfully to identify suitable outputs with respect to the input requirements, both within the project and by other researchers, Pan *et al* (1997).

The selected network must have a structure that can be predefined and remain constant thereafter. This is essential if the process is to interface successfully with the IIS as the inputs and outputs from the network must correspond to the requirements of the system and the components to be chosen from. The ridged structure required by the system therefore means that the ART network is not suitable for this application due to the dynamic nature of the classification that if performs, (creating patterns instead of relating to the closest). Thus an ART network would experience difficulties interfacing with the IIS. The backpropagation network with its ridged structure enables this essential interfacing with the IIS as the pattern classification process will relate the input to the nearest output pattern corresponding to a design.

The network used for this application will be the multi-layer perceptron trained with the backpropagation technique, as this network fulfills the two main criteria. In addition, the network is also capable of interpreting continuous values as well as binary.

4.5.1.2 Application of Backpropagation Network to Design Retrieval and Selection

Prior to the application of the network to the design retrieval, three points must be considered. These points comprise the stages of development that the process must pass through which will be unique to each application. The three points include:

1. Interfacing with the IIS.
2. Identification of the correct parameters / requirements that will form the inputs to the network.
3. Accumulation of the training data for the network.

Interfacing

Interfacing with the IIS will consist of the input pattern being prepared at a pre-process stage and the results decoded and transferred back through a post-processor. In the pre-processor the information relating to the design regarding its location in relation to other components, physical constraints and purpose are extracted from the IIS design database and prepared forming the input pattern.

For the output to link with the IIS each element in the output layer of the network will correspond to a design variant. The successful design will be determined by the element with the highest output value. The post-process stage will identify the design variant to be retrieved and transfer its type back to the IIS design database.

The process follows the diagram in Figure 4.8 below.

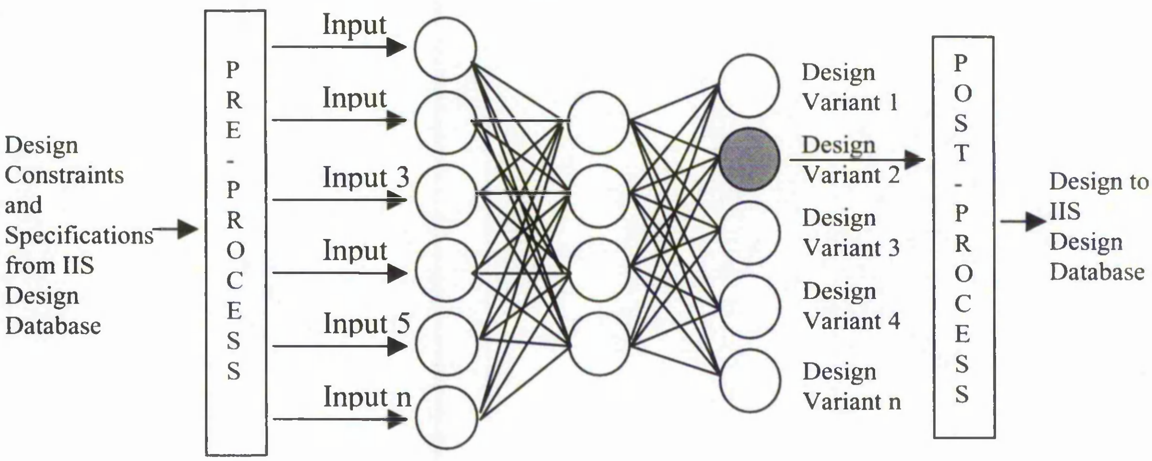


Figure 4.8 Interface of the Retrieval Network with the IIS

Identification of Input Parameters

Identifying the input parameters to the network is a critical stage of the development of the retrieval network. The success and effectiveness of the process is dependant upon these parameters, therefore they must represent the various aspects that affect the designers decision making process.

The parameters must reflect the points that the designer will take into consideration when selecting designs. The following guidelines will help identify the areas, which will be taken into account by the design expert:

1. *Product Design Specification-* The PDS describes the goals and aims that the final design must fulfill. Therefore some of these will probably directly influence the selection of the design variant.
2. *Location and Arrangement-* Positional requirements that would require specialised or modified geometry of the standard design.
3. *Dynamic Effects-* Physical effects acting upon the components performance, caused by neighbouring or related components.
4. *Other-* These parameters, if any, relate to idiosyncrasies of the component and its application(s) within the product design. These parameters may also relate to the range of products that the system is being used to design. For example, a standard motor used for a range of audio cassette decks.

Not all the guidelines above need to identify parameters to form inputs to the network. The guides are intended to help identify elements that affect the decision process by causing the system developer and design expert in the particular field to analyse their actions.

Identification of the parameters will be a dynamic process and probably require modification during the initial stages of the information accumulation. However during training of the network the inputs that have little or no influence upon the outcome will be identified by the training paradigm and the connections to the corresponding elements reduced in strength. The network can thus aid in the identification of the important parameters by analysis of the connection weights after training. This is a reverse application of backpropagation networks to that used by Vaughn (1996). Although too much information can be handled by the network, excessive amounts will cause convergence problems. However, to little will not correctly describe the process. It is therefore a good policy to present more rather than less.

Note that the parameters are to be supplied to the network from the IIS design database. Therefore the parameters must exist before they can be presented to the network, thus aiding in their identification.

Accumulation of Training Data

The quality of the design selection and in turn the IIS is largely dependant upon the information and knowledge that the network is trained with. If the knowledge is too vague, erratic and contradicting the network could have difficulty converging during training, thus making poor component selections. Accumulation of expertise is therefore a critical process and must be conducted by a method that will truly represent the design process, while not requiring the expert to constrain the selection process with regard to preferences and instinct.

The most appropriate method of assessing an experts opinion of best suited designs to the design specification and requirements is by presenting the design to the expert and noting the designs selected. However two methods of presenting the situations to the expert are applicable.

Firstly, generate a large variety of situations that will cover a broad range of applications the IIS will be set and record the resultant solutions. This method will produce the information required, but does however require an expert, (preferably several) with extensive knowledge in the particular field. Networks trained with this knowledge would be trained prior to use with the IIS. Therefore this knowledge will relate to the retrieval of the designs in general, as the accumulation of the knowledge is performed separate from the IIS.

The second method of accumulating the knowledge to train the network is by incorporating the entire training process within the system. The principle is to temporarily replace the design retrieval process to be performed by the ANN with a Graphical User Interface (GUI). The GUI will display the design specifications and requirements and the design options to the designer that is using the system. Based upon the specifications that are presented, the designer can select the appropriate design offered. The resultant solution and the specifications are recorded and appended to a training file, which will form the training data for a network. Generation of data using this method removes the laborious process of generating solutions for randomly generated situations, as used in the first method, but more importantly moulds the design selection to the application that the IIS is being used for. For example, instead of general knowledge about bearings and their applications, the knowledge will relate to the range of products that a particular manufacturer produces.

4.5.1.3 Network Training Process

As mentioned earlier, the training data is accumulated from the initial usage of the IIS, where the design parameters and results are recorded. During this stage of the IIS development, the designer is presented with the parameters that are felt to influence the selection of a design variant. The parameters are presented through the use of a graphical user interface (GUI). At the same time a choice of design variants are also presented, enabling the designer to identify more easily between the parameters and the selection. Figure 4.9 gives an indication of what the designer will be presented with.

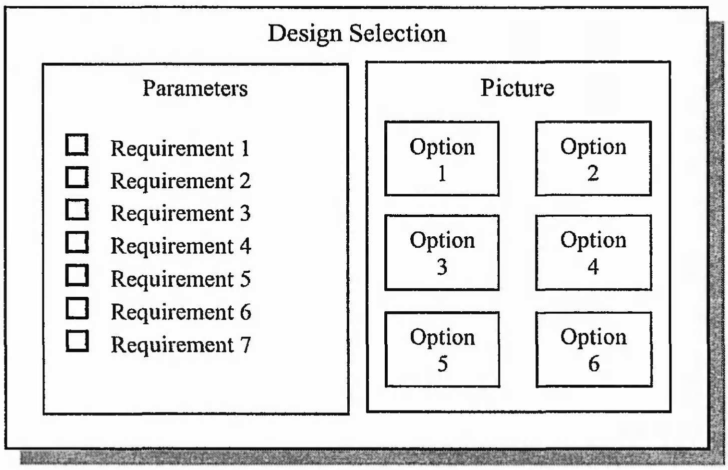


Figure 4.9 Design Selection Interface

Presenting the designer with a picture or diagram of the options makes the identification much easier as images are easier to relate to. This is due to the phenomena of images prompting people to recollect experiences and information in greater detail.

Training of the network is performed automatically using the GEN-NEU approach and program described in section 3.2.3. This program will remove the necessity for expertise in the field of ANNs regarding the parameters of the network paradigm that affect training.

In order not to slow down the response of the IIS while the training data is being accumulated and the networks trained, training of the networks will be conducted after the IIS has completed its current design. This is due to the amount of time that is required to train the networks.

Before the network replaces the GUI the network must prove that it can select designs based upon the same knowledge that the designer calls upon. The proof is conducted in two stages. Firstly, the network is trained and tested with examples accumulated via the GUI. Once the network has passed this test, the network will predict the design variant that the designer will select. If the network can successfully predict five selections (a nominal figure) in a row the network will replace the designer within the IIS for the selection of this design. The number of correct tests is determined by the system developer and should represent the importance of the correct component selection. However, during these tests the training file has been continuing to expand. If the network fails, the network is retrained to include the new examples, and the process repeats. Application of this approach is described in section 5.6.

4.6 Comments on IIS Approach Structure and Knowledge Encapsulation

The system approach comprises of three main parts, relating to the design process and strategy, the process of acquiring and representing knowledge and the structure that the integrated system will take. Each of these parts must be considered simultaneously when developing the system as the effects of one are transferred throughout the system development. The design process forms the general guide to the systems development, enabling the areas of knowledge that are require to generate the product. This knowledge is assessed to determine its form and the best method of encapsulation. Depending upon the purpose of the knowledge and the form of encapsulation, the structure of the system is determined, thus affecting to the systems development and structure.

Constant consideration of the form of encapsulation and the ease in which the knowledge can be modified is aided by the modular approach that the approach promotes. Thus the modular structure of the system benefits the ease of incorporating external packages within the single environment and enables the system to be updated by the user, (ensuring that the system does not become obsolete) by the replacement of the relevant module.

The use of a hybrid of AI techniques used for the encapsulation of knowledge within the IIS allows the weaknesses of one technique to be compensated for by the application of a different technique with strength in that area. This unique application thus allows the design process to be modelled closer to the process that a designer would perform. This is most evident when production rules and ANNs are considered. The production rules, as mentioned are an excellent method of encapsulating well defined, logical knowledge. However if the knowledge is incomplete of based upon previous example that are not exactly the same the production rule experience difficulty in determining a solution. The use of ANNs for this situation does not experience these difficulties in the same manner. As they are trained by example and able to interpret new examples based upon previous experiences a solution may be formed. The combination of these two AI techniques allows the logical and creative elements of the design process to be captured and used by the system. As these elements would be used by a designer.

The two applications of the AI techniques, EP and ANNs, developed for this project illustrate how the modular approach can be used to perform individual elements of the design process. The two applications illustrate two important elements of the design process, the combination of which within a single design environment is unique to this project.

CHAPTER 5

INTELLIGENT INTEGRATED SYSTEM FOR THE DESIGN OF MECHANICAL POWER TRANSMISSION SYSTEMS

The IIS has been applied to the design of mechanical transmission systems. The mechanical transmission system has been chosen as it is a common product with a vast amount of information available to base multiple designs upon

The actual design of the mechanical transmission system also provides an excellent structure to the design process. This is due to the finished product design comprising of several different components, therefore, allowing the modular approach toward the detail design of components to be clearly structured. The complexity of the component arrangements allows the iterative procedure of the design process to be exploited, in the endeavour to produce a design that assembles correctly. The general concepts of the conceptual design provide an excellent example of the conceptual building blocks that the IIS approach considers when generating conceptual designs. For example the basic transmission components of gears, belts and chains together with the basic arrangements of perpendicular, cross and parallel constitute the conceptual building blocks. The information relating to the design of the mechanical transmission system is summarised in the section 5.1. The identification of the information prior to system development allows the appropriate form of encapsulation to be determined.

5.1 Application Description

5.1.1 Identification of Problem and Design Specification

An intelligent computer-aided system is required to design general purpose mechanical power transmissions. The final design will transmit power and rotary motion between two fixed points in space. The design will be reliable, employ standard engineering techniques and comply with appropriate safety and performance standards. The design may be internally or externally housed but always fixed relative to the inputs and outputs.

The specification that the design will be required to conform to covers the performance of the design and the manufacturing process. These specifications cover all possible applications within the range of the problem statement. The following are the major initial specifications covering all applications of the transmission and form the general product design specification (PDS):

Transmission Performance

1. Orientation of Input to Output.
2. Reduction Ratio.
3. Ratio Accuracy.
4. Transmission Smoothness.
5. Transmission Efficiency.
6. Load Distribution.
7. Weight.
8. Size.

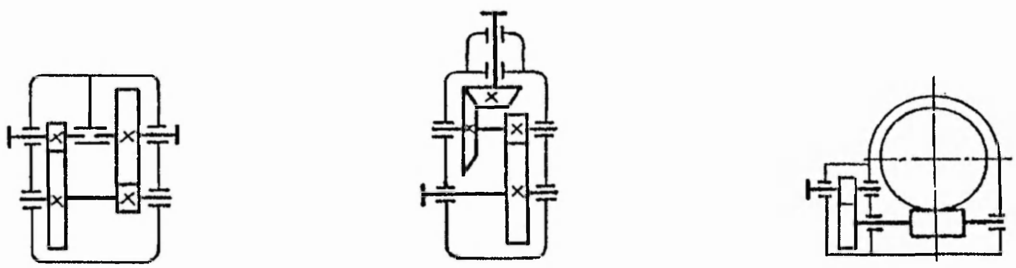
Manufacturing Considerations

1. Cost.
2. Manufacturing Accuracy.
3. Ease of Manufacture.
4. Ease of Maintenance and Assembly.

These specifications comply with those used by Su (1990) in his work. Use of the same specifications will allow comparisons to be drawn upon the approach developed within this project and the application's results.

5.1.2. Concepts and Possible Solutions

At completion of the concept stage the design is to be sufficiently developed to enable detail design to commence. In the case of the mechanical power transmission the concept is required to define the orientation, number of stages and the types of components and their arrangements. Figure 5.1 gives examples of concepts.



Two Stage Parallel

Three Stage Cross & Parallel

Two Stage Perpendicular & Parallel

Figure 5.1 Power Transmission Concepts

Orientation of Input and Output Shafts

The transmission's purpose is to transfer power and motion from one point in space to another while maintaining the same orientation to the input or altering the outputs geometric relationship to the input. This leads to three possible orientations of the design, parallel, cross and perpendicular, as shown in Figure 5.2 below.

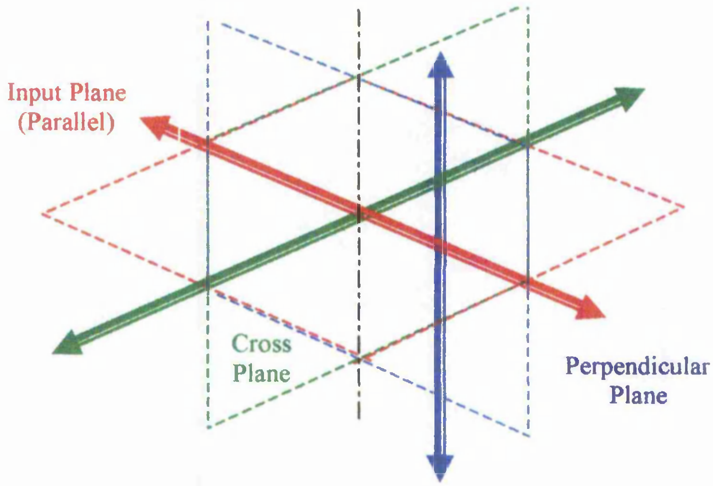


Figure 5.2 Orientation of Transmission

Parallel Orientation Parallel orientation is when the input and output are in the same plane. Two types of parallel arrangement are covered in this project, input-output offset and input/output in-line.

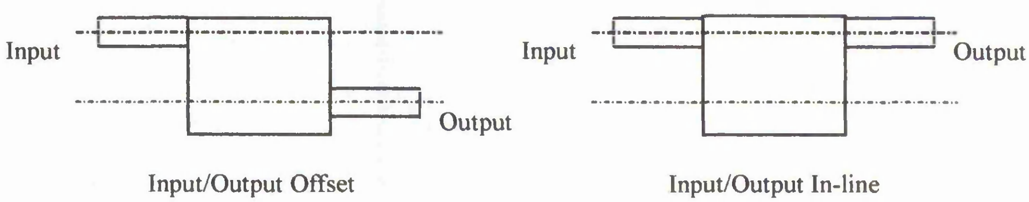


Figure 5.3 Parallel Arrangements

Cross Orientation Cross orientation uses a bevel gear to alter the axis of rotation as shown below in Figure 5.4. The gear is used for the first stage of the speed reduction train. This is due to lower forces being generated at higher speeds, therefore, enabling the size of gear to be reduced to a minimum. Size is particularly important for bevel gears as they are more difficult to produce, maintain and are more expensive as size increases, relative to parallel gears.

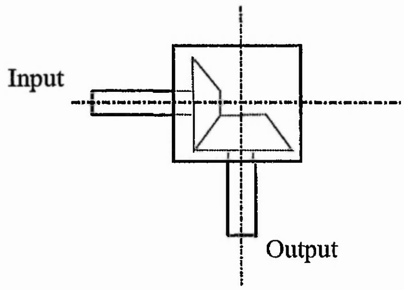


Figure 5.4 Cross Arrangement

Perpendicular Orientation Perpendicular orientation uses a worm and gear to alter the axis of rotation. The worm and gear are used for the first stage of the speed reduction train for similar reasons to the bevel gear, except that the input and output are not on the same plane.

The ratio achieved with a worm gear drive is large compared to that of the bevel gear and is therefore, suitable for large speed ratios. However, as the size ratio of the worm to gear is so large that speed ratios of less than 5:1 cause meshing problems.

This method of transmission has the characteristic of smooth power transmission, due to the continuous engagement of the worm and wheel.

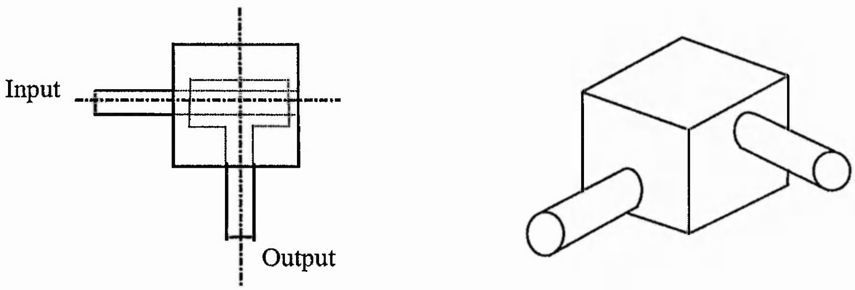


Figure 3.5 Perpendicular Arrangement

5.1.3 Types of Transmission Components

The selection of the correct components contributes dramatically to the performance of the transmission and therefore, the success of the design in meeting the specifications stipulated. Figures 5.6a and 5.6b show the basic components considered at the conceptual design stage. However, the basic principles developed in this research are also applicable to other types of components.

Figure 5.6a Flexible Transmissions

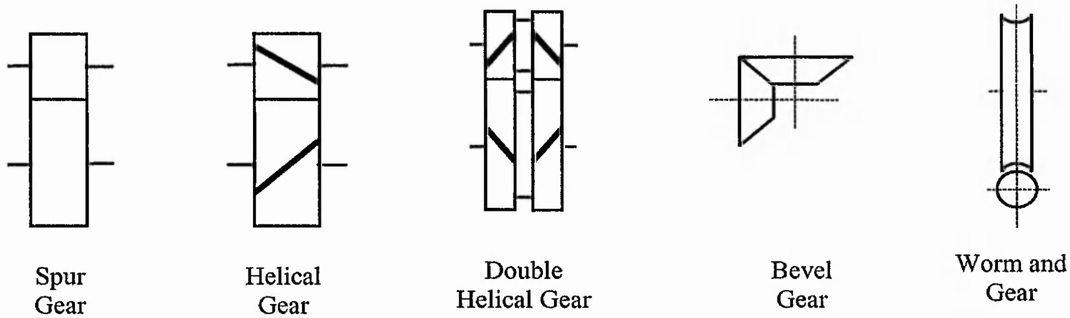


Figure 5.6b Gear Transmissions

Component Selection

The knowledge that is encapsulated within the system relating to the components refers to the relative performance of each component to one-another in their ability to satisfy the specifications identified in the general PDS, in section 5.1.2. The ratings range from 0 (representing unsuitable for the specification) to a maximum value (representing ideally suited). The maximum value can vary for different applications depending upon the expert that is providing the knowledge and information. The relative performance of the components to satisfy the specifications are determined by the design expert and constructed into a design matrix, as illustrated in Table. 5.1.

	Specifications									
Basic Units	A	B	C	D	E	F	G	H	I	J
Double Helical Gear	**3**	4.5	3	**2.5**	3	**2**	2.5	3	2.5	2.5
Helical Gear	**3**	4	3	**3**	3	**2.5**	3	2	3	3
Spur Gear	**3**	3	3	**3**	3	**3**	3	3	3	3
Belt	**1**	6	2	**4**	4	**3.5**	2	1.5	3	1
Chain	**2**	2	2	**4**	4	**3**	2.5	1.5	3	1
Worm Gear	**3**	5	1	**2**	2	**1.5**	6	2	2.5	8
Bevel Gear	**3**	2.5	3	**3**	3	**2**	2.5	2	2	2.5

Table 5.1 Basic Transmission Component Matrix (Range 0 to 8)

Key :-				
	A	Ratio Accuracy	F	Cost
	B	Transmission Smoothness	G	Weight
	C	Transmission Efficiency	H	Load Distribution
	D	Ease of Maintenance and Assembly	I	Manufacturing Accuracy
	E	Ease of Manufacture	J	Size

The suitability of a design to a set of specifications is found by summing all the values in the matrix that relate to both the design and the identified specifications. This summing process is repeated for all the designs. The design with the largest total represents the most suitable solution to the specification. The resultant design totals provide a relative comparison of the suitability of each design to the specification. Thus an output pattern is created corresponding to an input.

Example

Specifications A, E & F have been selected in the basic unit matrix.

Design	**Suitability Values**			**Total**
	A	**E**	**F**	
Double Helical Gear	3	2.5	2	7.5
Helical Gear	3	3	3	9
Spur Gear	3	3	3	9
Belt	1	4	3.5	8.9
Chain	2	4	3.5	**9.5**
Worm Gear	3	2	1.5	6.5
Bevel Gear	3	3	2	8

Design *Chain* has the highest total value with 9.5 and is therefore, the most suitable solution to maintain and allow an accurate ratio accuracy, allow for ease of maintenance and assembly while being cheap to produce.

The design matrix provides a means of representation and the component knowledge relating to their suitability to perform the task, but does not allow for exceptions to the rule. These exceptions derive directly from the design expert and represent many years of experience and personal preference.

5.1.4 Concept Arrangements

The arrangement of the components within a concept is determined from the PDS and a component hierarchy. Knowledge relating the suitability of an arrangement with respect to the PDS consists of a combination of physical limitations, which are well-defined rules and the expert designer's judgement and expertise. The physical limitations relate to the selection of components required to achieve the orientation and number of stages, while the design expertise is formulated in the same manner as for the components, via a design matrix.

The component hierarchy also constitutes a combination of physical limitations and common sense to form a well-defined rule. The rule states that once a component has been used in a design, any additional stages after this point may not use components higher in the hierarchy. This hierarchy follows the ordering illustrated in Figure 5.7.

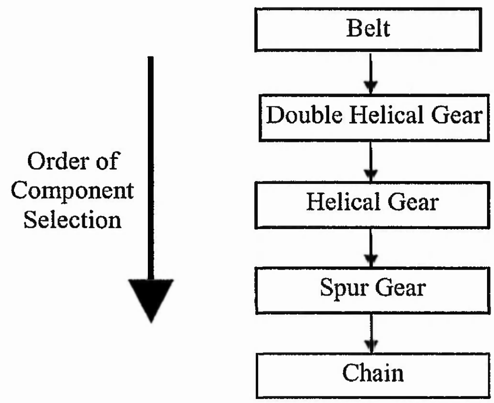


Figure. 5.7 Component Hierarchy

Components higher in the hierarchy are not used once lower components have been used within previous transmission stages. This is due to benefit achieved from components higher in the hierarchy being cancelled out by components from lower in the hierarchy used in earlier transmission stages. For example, if a helical gear were used after a spur gear to reduce noise, no benefit would be achieved as the spur gear would already generate considerable noise.

5.1.5 Categorise Conceptual Design Knowledge

Categorisation of the knowledge, selection of means of encapsulation and determination of requirements are carried out with the aid of Table 5.2 below. The purpose of the table is to order and simplify the allocation process while allowing the identification of related groups of knowledge.

Number	Type of Knowledge	Encapsulation means	Reason	Requirements	Interaction with
1	Input/Output Orientation	User	Basic Specification	PDS	None
2	Type of component in first stage	Production rule	Well defined	Orientation	1
3	Number of stages	Production rule	Well defined	Ratio and Orientation	1 & 4
4	Selection of arrangement dependant on PDS	ANN	Data intensive but with exceptions to the rule based on experts experience	PDS & number of stages and orientation	5 & 6
5	Selection of component dependant on PDS	ANN	Data intensive but with exceptions to the rule based on experts experience	PDS	6
6	Component hierarchy	Production rule	Well defined	Number of present stage and type of previous component if applicable	1, 2 & 5

Table 5.2 Categorisation and Relation of Conceptual Design Knowledge

5.1.6 Scope of the Detail Design

The range of the concepts have been defined providing important constraints upon the scope of detailed sub-assemblies and the components that are required to complement the general designs. These are listed to identify the areas where detailed analysis and information is required.

Detailed information about each of these components is given within their respective design descriptions in section 5.4 to prevent duplication of information.

Areas of design

Pulleys and belts
Chains and chain wheels
Parallel gears (spur, double and single helical)
Bevel Gears
Worm and gear
Bearings
Shafts
Housing
Spacers

Detail design within each of the above areas and their interaction with others is important the crucial as this will have a major effect on the validity of the performance of the final design. As the design process is iterative, modifications to one component will have a knock on effect on other related components.

For the purpose of this project, where the primary aim is to validate and develop the intelligent hybrid approach in the form of the IIS, the complexity of design at the detailed design stage will be limited to the parallel power transmissions. Thus detailed component design knowledge will be restricted to shafts, spur and helical gears, belt drives, bearings and their assembly.

5.1.7 Parallel Gear Design

Parallel gears relate to helical and spur gears and should be designed to comply with recognised professional standards. Thus ensuring that failure does not occur within the life and normal application of the power transmission. The standard used for the design is the British Standard, BS 436 part 3 (1986).

Knowledge with relation to Gears

Gear design has been studied for over 100 years and is an extensively researched area. Therefore, knowledge relating to the identification of characteristics, properties and dimensions is extensive.

Design of the gears is based on the design guide ESDU 88033, which is derived from BS 436 part 3. The basic dimensions and characteristics of the gear are determined with respect to the bending and contact stresses acting on the gear teeth.

The gears are designed with consideration to the relating factors that affect the performance of the gear.

- *Material and treatment*- determines the strengths and limits of the gear, e.g. maximum contact stress.
- *Manufacture route* - accuracy of dimensions and surfaces often dependant on material and treatment.
- *Safety factors* - limits on safety, often dependent on application.
- *Physical requirements* - loads to carry, number of teeth, etc.
- *Standardisation* - standard values observed by industry.

The sources of the information relating to these areas can be found in appendix B

5.1.8 Categorise Parallel Gear Design Knowledge.

Once the factors that contribute to the gears design and performance have been identified, the appropriate means of encapsulating the relevant knowledge are established in conjunction with their requirements and dependencies. These are presented in Table 5.3 below, again aiding the development of the design modules structure.

The knowledge and means of encapsulation have been established in preparation for the construction of the design modules. The information necessary to structure the IIS has also been established in both general terms for the overall structure and detail for the conceptual and detailed designs.

Number	Knowledge relating too.	Encapsulation means	Reason	Requirements	Direct interaction with
1	Gear ratio	Production rule dependant calculation	Well defined	Number of stages & overall ratio	IIS
2	Material, Heat Treatment and manufacturing route	User via GUI	Data intensive with minimal specification	User	
3	Safety Factors	ANNs	Original data not available and interpolation required	Gear ratio and Orientation	1
4	Application Factor	User via GUI	Requires input from user and offers medium to present choices	Application	
5	Maximum Stresses	Production rule dependant calculation	Equation	Heat Treatment and Life	IIS & 2
6	Facewidth ratio (centre distance not stated)	Production rule	Multiple specification	Gear type, mounting and heat treatment.	IIS & 2
7	Facewidth ratio (centre distance stated)	Calculation	Equation	Application factor, gear ratio, torque, diameter.	1, 4 & 8
8	Diameter	Calculation	Equation	Application factor, gear ratio, torque, facewidth ratio,	1, 4, 6 & 7
9	Pitch Accuracy	ANN	Original data not available and interpolation required		2
10	Lead Accuracy	ANN	Original data not available and interpolation required		2
11	Module	Calculation	Equation		1 & 8
12	Module Selection	Data base	Standard values	Calculated value	11

Table 5.3 Categorisation and Relation of Knowledge

5.2 System Control

Control of the IIS for the design of mechanical power transmission systems is structured into levels of responsibility and control. The higher the level the greater the responsibility and control. This structure is explained in section 4.1.3. Within the application of the IIS the system controller is responsible for the overall structure of the designs development, activating the appropriate design stage controller as required. Activation of the appropriate stage controller is determined from analysis of the designs status. Monitoring the status identifies if the design is following the standard path of conceptual design followed by detailed design or if redesign is required. Upon the need for redesign a series of production rules will determine the appropriate course of action, activating the appropriate stage controller. At this point the relative information is extracted from the IIS design database to enable effective redesign.

Once control and responsibility has been passed to a stage controller, development of the design is structured by that controller. The purpose of the stage controller is similar to the system controller, determining the appropriate design modules to activate, extracting and updating information to and from the IIS design database and taking appropriate action for redesign.

5.3 Conceptual Design Control

Overview

The conceptual design stage for the design of a mechanical power transmission system briefly follows the following sequence of events. The user is prompted for the initial PDS, from which the concept will be generated. The general arrangement and the number of transmission stages are defined using a combination of production rules and ANNs. Once the arrangement has been defined the type of power transmission between stages are determined. This is achieved again by the use of production rules and a series of ANNs. The output from the conceptual design module will be the conceptual design, comprising of the arrangement, number of stages and the methods of transmission within each stage. This overview is encompassed in Figure 5.9.

The conceptual design module consists of a combination of ANNs and rule bases developed within a C++ program. The rule bases cover the situations where clear decisions can be made one way or the other, while the ANNs are used for hazy, vague situations.

The use of the rule bases and ANNs are controlled by a Stage Controller, which executes the design process, calling upon both rule bases and ANNs as and when required. The controller also acts as a pre and post processor to the ANNs, preparing the inputs from the user interface then analysing and displaying the results. Figure 5.8 below illustrates how the Conceptual Stage Controller coordinates the conceptual stage.

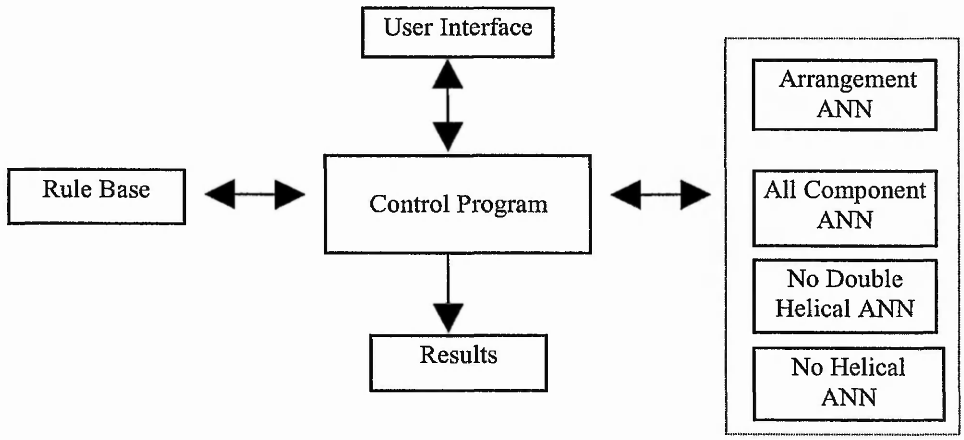


Figure 5.8 Conceptual Stage Controller Schematic

The design program addresses the problem in a similar manner to a designer. Just as a designer will split the problem up and address each section in turn, the program splits the design into two areas, arrangement and component selection. Each of these areas has a number of rules and ANNs that work together to produce the result. The division of the problem is illustrated within Figure 5.9.

The first stage of the design process is to determine the transmission system arrangement. This is performed by the arrangement network. Before the network can produce a suitable design its input pattern must be generated. A combination of rules and user input are used to generate the input pattern. The rule base determines part of the networks input pattern, corresponding to the orientation and the number of stages, while the remainder consists of the user defined design specifications. From the input pattern presented, the ANN produces a suitability rating of real numbers ranging from 0 to 1 for all the possible transmission system arrangements. Based on the rating results, the control program ranks the designs in order of suitability, i.e. the concept with the highest rating is ranked as the most suitable. The top ranked solution and the next two ranked alternatives are then presented to the user. If the user considers the recommended solution not to be satisfactory either of the alternatives can be chosen to override the system.

For example in Figure 5.11 the user is presented with the option of selecting between the following parallel arrangements; 1) Two stages with no casing, 2) One stage with casing and 3) Two stage with casing and input/output offset. The order corresponds to the suitability of the arrangement to satisfy the specification, determined by the ANN, but the final decision is left to the user.

Once the design arrangement has been finalised the components will be determined using both the rule base and a combination of ANNs. The rule base has two tasks at this point in the designs development, selection of non-parallel components and applying the component hierarchy to determine the correct ANN to use for parallel components. Selection of non-parallel components by the rule base is due to their application being well defined and simple to encode within this form. Determining the correct component ANN uses the component hierarchy, illustrated in Figure 5.7, which is transformed into a series of rules. The action of each rule is the activation of a component ANN. The component ANNs determine the type of transmission most suitable for each transmission stage with respect to the specification. These ANNs encapsulate the sections of the component hierarchy, resizing the hierarchy by cropping off the higher components depending on the purpose of the ANN. The ANNs contain either all the parallel components, the double helical gear removed or both the double and single helical gears removed. These components are removed by permanently setting the respective output elements to zero during training, effectively removing them from the design while keeping a uniform architecture to the ANNs. This uniformity is necessary for the control program to analysis of the results.

When the conceptual design has been completed the control program displays the finished conceptual design to the user and transfers the results to the system database. Upon completion control of the design process is returned to the system controller in the higher layer.

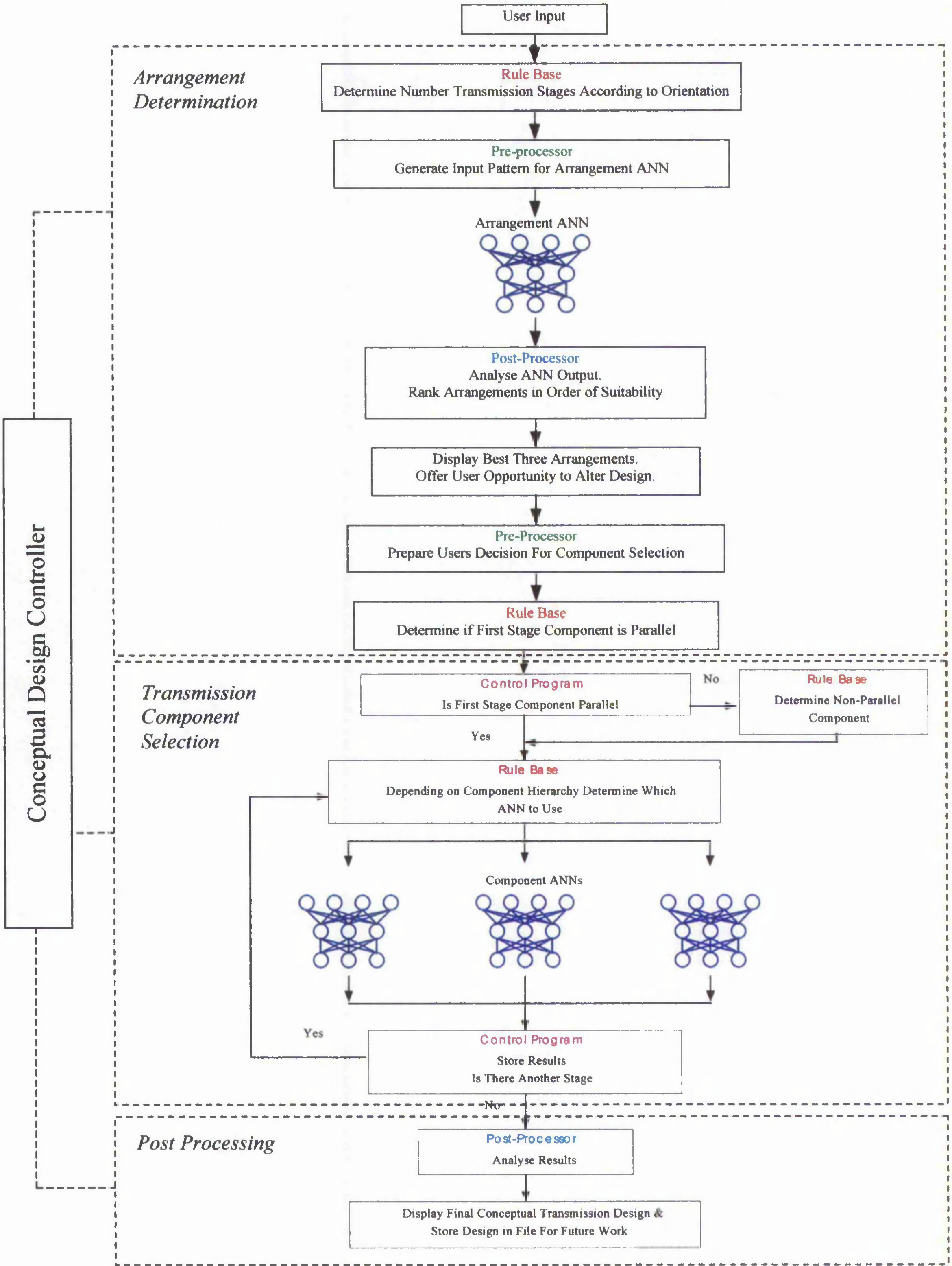


Figure 5.9 Schematic of Conceptual Design Process

5.3.1 User Interface

Information required for the design to proceed that cannot be encapsulated via the AI techniques or conventional methods is requested from the user. These requests are generally in a form that prompts the correct range of responses from the user. For the conceptual design these requirements comprise of its reduction ratio, orientation and the specifications that the finished design must comply with, such as size, weight, efficiency, etc. The full list of specifications is identified in section 5.1.2.

As discussed in section 3.4 the graphical user interfaces (GUIs) should provide all the information that the user will require in a form that is obvious, simple and clear. To achieve this, the GUIs are constructed using Visual Basic. The interfaces provide a professional appearance to the system enabling the familiar Windows environment to be exploited. Communication with the GUIs is achieved by the use of transfer files that can be simply modified if the information to be transfer needs to change due to modifications of the IIS. Examples of the interface are shown in Figures 5.11 and 5.12 and in section 6.

5.3.2 Rule Bases

The rule bases consist of three parts: determining the number of stages within the power transmission system, selection of relevant ANNs to conform with the component hierarchy and selection of the first stage components. The rule bases communicate with the control program throughout the whole design process. A typical example of a set of rules is shown below. These rules are used to determine the number of stages required for parallel orientation. The selection is dependant on the speed reduction *ratio*.

If *ratio* <4 then arrangement is *one stage parallel*

If *ratio* > 4 & ≤ 7 then arrangement is either *one stage parallel* or *two stage parallel*

If *ratio* > 7 & ≤ 30 then arrangement is *two stage parallel*

If *ratio* > 30 & ≤ 100 then arrangement is *three stage parallel*

5.3.3 ANNs and Training

The ANNs to be used for the purpose of selecting components and arrangements depending upon the PDS are feed-forward Multilayer Perceptrons trained with the backpropagation technique, more commonly known simply as backpropagation networks.

This type of network has been selected for two main reasons:

- *fixed structure* enabling a standard structure to be established within the design system for easy modification and update of knowledge.
- *success in pattern recognition* for other applications, increasing the probability of success for their application within the project.

Training data has been prepared for the networks using an excel program which performs the suitability calculations described in section 5.1.4, then scales the results between 0 and 1, matching the results against the PDS requirements. Exceptions to the rule are now added to the training data, replacing the calculated output pattern. The results are matched against the specifications, forming the training data similar to the form below.

0	1	0	1	0.2	1	0.35
Input Pattern				Output Pattern		

This data would represent 4 inputs to the network and 3 outputs.

Networks for concept design

Four networks are used within the concept stage. Three for component selection and one for the arrangement.

Component networks

Three networks are required to incorporate the hierarchy. Each network is trained with variations of the same data with either none, one or two of the outputs permanently set to zero. Setting the elements to zero effectively removes the element from the network while maintaining a standard network structure.

Arrangement network

The arrangement network incorporates the PDS, orientation and number of stages in its decision making process. The number of stages that are required are first defined by a series of production rules. However, a degree of ambiguity is introduced when the parameters of the production rule conditions overlap giving more than one resultant solution. The structure of the networks is such that the design specification forms the ANNs input, while the concepts represent the output. Figure 5.10 illustrates this relationship.

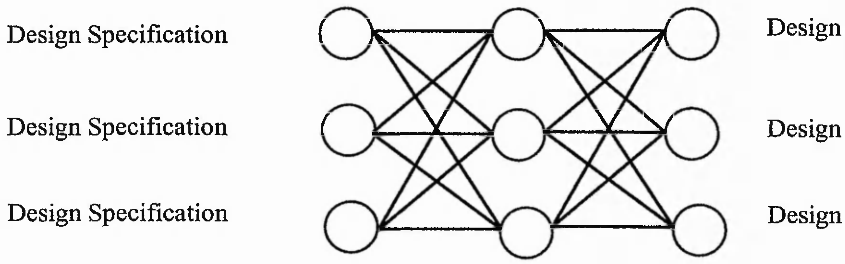


Figure 5.10. Fully Connected Network

Training

Training of the ANNs was performed using a training program developed for the conceptual design module. This program allows the architecture of the networks (known as the topology) to be defined. Additionally when extending the training period of an existing network the learning and momentum coefficients of the backpropagation error equation are modified to represent annealing. Training of the networks is performed by modifying the topology and coefficients. The effect of these modifications is to increase the performance of the network and reduce the error between the target and actual outputs. These networks were trained prior to the development of the GEN-NEU program and formed part of the investigation into the Backpropagation training discussed in section 3.2.3

Correct training of the ANNs is essential as they contain the knowledge and expertise of the design system. However, with ANNs it is common and expected that the output from the network will produce an error when compared with the target output. Therefore, a means of determining when the error is sufficiently small must be defined. The method used for the network is to compare the ranking of the outputs. Provided the ranked order of the network output matched that of the test case, the network had passed.

5.3.4 Example of Concept Design

Design a power transmission system to meet the following specifications:

- the input/ output shafts are parallel to each other
- speed reduction 5 : 1
- ratio accuracy important
- transmission efficiency
- able to distribute load

The input to the system and the arrangement and alternative designs from the system are shown in Figure 5.11. The final conceptual transmission design is shown in Figure 5.12.

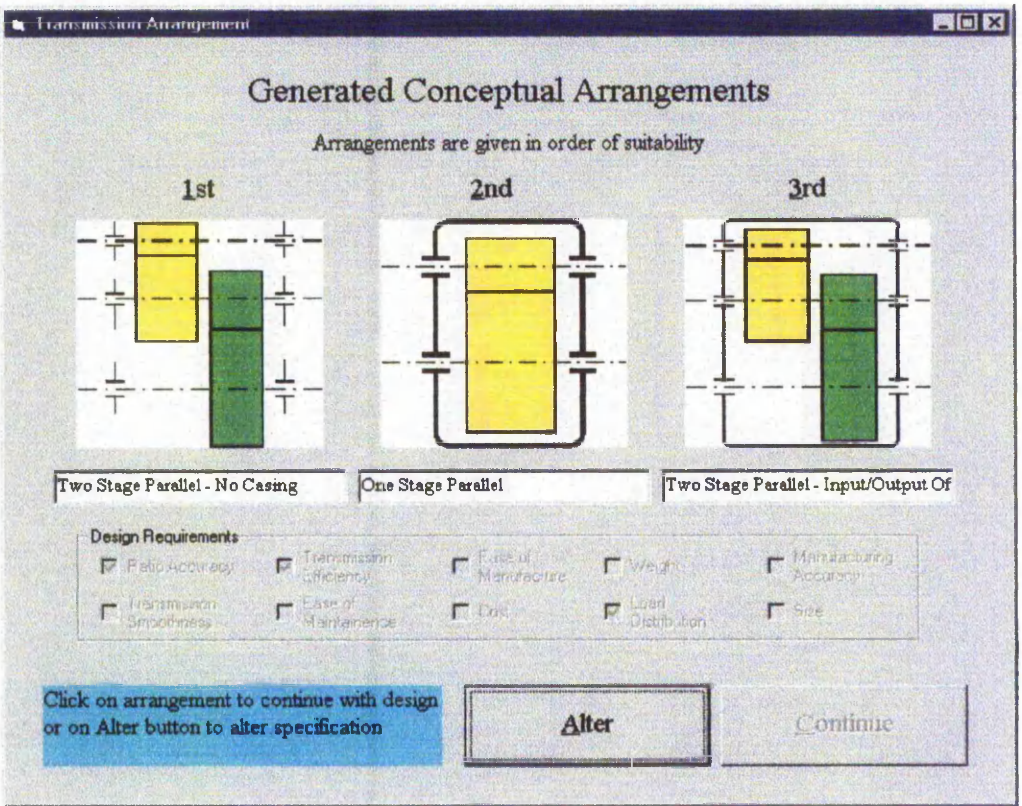


Figure 5.11 Screen Dump Illustrating Design Specification and Corresponding Arrangement Designs

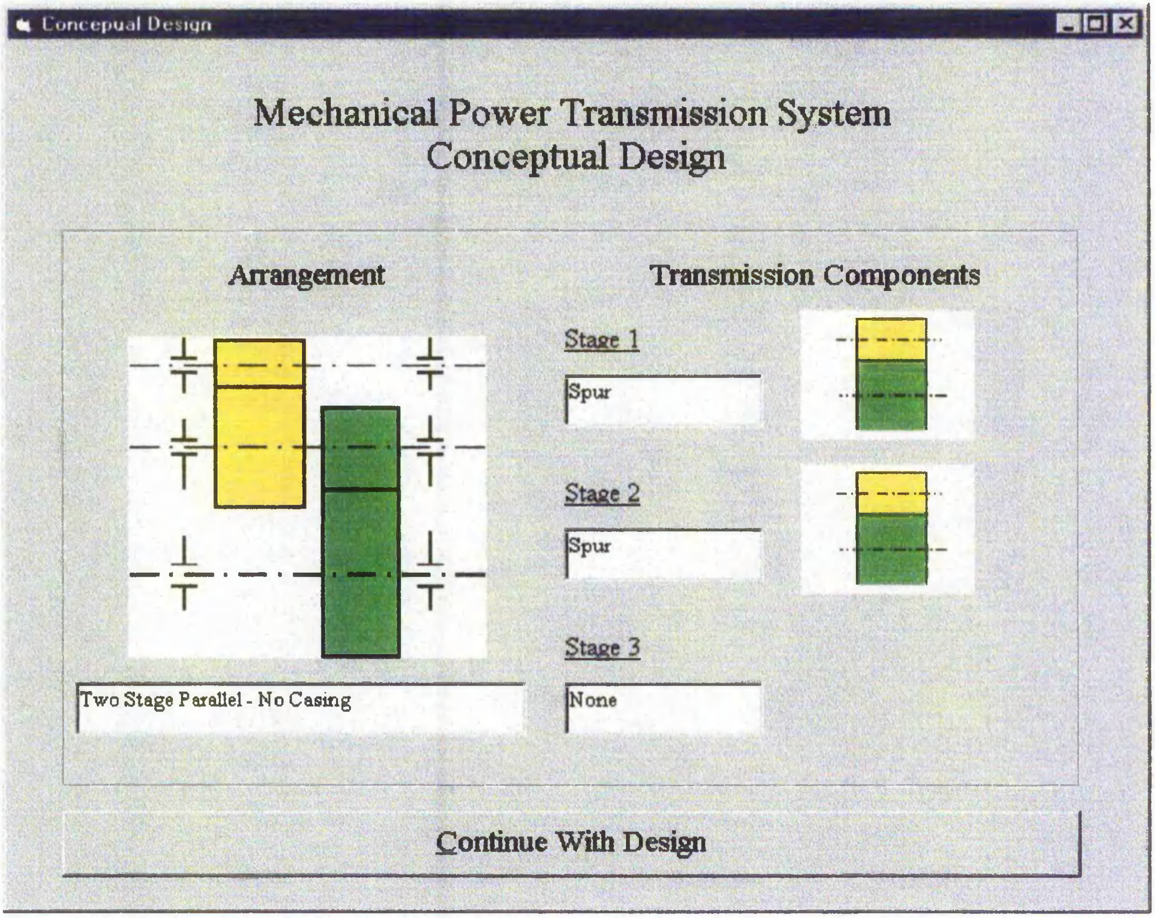


Figure 5.12 Screen Dump Illustrating the Resultant Finished Conceptual Design

Figure 5.12 gives the solution to the requirements stated earlier. The resultant conceptual design comprises of a two stage parallel arrangement with no solid casing (gears open) using spur gears for the type power transfer for both the first and second stage.

To ensure that the conceptual design module has encapsulated the knowledge correctly the design is repeated manually using the same rules and design matrices. The component in the first stage of a parallel transmission is selected to evaluate the system, which corresponds to a Spur Gear. The component for the same position within a transmission system is derived from the relevant design matrix and the process and results given below in Table 5.4. From the normalised values it is indicated that either the Double Helical Gear or the Spur Gear are equally the most suitable for the application. Therefore, the systems selection of component is viable selecting the more common of the two options.

Design	Specification A	B	C	D	E	F	G	H	I	J	Total	Normalised Value
Double Helical	**3.0**	0.0	**3.0**	0.0	0.0	0.0	0.0	**3.0**	0.0	0.0	9.0	1.0
Helical	**3.0**	0.0	**3.0**	0.0	0.0	0.0	0.0	**2.0**	0.0	0.0	8.0	0.889
Spur	**3.0**	0.0	**3.0**	0.0	0.0	0.0	0.0	**3.0**	0.0	0.0	9.0	1.0
Belt	**1.0**	0.0	**2.0**	0.0	0.0	0.0	0.0	**1.5**	0.0	0.0	4.5	0.5
Chain	**2.0**	0.0	**2.0**	0.0	0.0	0.0	0.0	**1.5**	0.0	0.0	5.5	0.611
Worm	**3.0**	0.0	**1.0**	0.0	0.0	0.0	0.0	**2.0**	0.0	0.0	6.0	0.666
Bevel	**3.0**	0.0	**3.0**	0.0	0.0	0.0	0.0	**2.0**	0.0	0.0	8.0	0.889

Specifications correspond to Figure 5.12.

Table 5.4 Hand Calculation of Most Suitable Component

The training data corresponding to this specification would be in the form below

Input Pattern (specification A to J in binary form, either selected or not selected)

1 0 1 0 0 0 0 1 0 0

Output Pattern (suitability of each component from double helical to bevel gear)

1.0 0.889 1.0 0.5 0.661 0.666 0.889

The design from the conceptual module complies with that derived from the hand calculations, therefore, indicating that the system is valid.

The hand calculation has identified that both the double helical and spur gears are equally suitable components for the application and therefore, an additional decision is required. However, the network has made the decision between the two and the result is the more common and standard design of the two, the spur gear.

5.4 Detail Design

Overview

The detail design of a mechanical power transmission system follows the following over simplified chain of events. Control of the design is transferred to the detail design controller from the system controller together with the detail information of the conceptual design. The conceptual design information includes the detailed list of components together with their position within the transmission system. Initial designs for the components are generated comprising of rough dimensions, material and manufacturing properties and force analysis. Once the rough, initial design of the transmission system has been generated, detail design modules are activated to perform detailed design of the components. Detail design of the components is performed in an iterative fashion if redesign is required. The detail design controller identifies components that have been modified and re-activates the design modules of the components that are affected by these changes. Throughout the design process relevant design steps, such as force analysis and the assembly are continually updated to ensure that the components design modules receive current information based upon the latest situation.

The Detail Design Control module approaches the process by performing design in two parts, the initial and the detail design. The initial design sets up the component database, identifies the components to be included within the database and produces initial dimensions and physical characteristics (such at material) of the components and their assembly. This process has been developed to allow for the modification of the component designs through its structure and the use of data tables. Once the components required to complete each transmission stage in the design have been identified and initiated the detailed design of the components can commence. An initial force analysis of the design is performed to enable the initial, rough design to be generated. Thus identifying where reaction forces occur together with initial magnitudes and directions.

The detail design process is of an iterative nature, where redesign of components is performed when physical or performance related components are altered. The detail design progresses through the stages of the transmission and shafts of the design, performing redesign and activating the appropriate design module as required. This process is repeated until all the components within the transmission complete. Once the detail design has been completed information relating to the components and assemblies are contained within the IIS database ready for the generation of the CAD model and drawings.

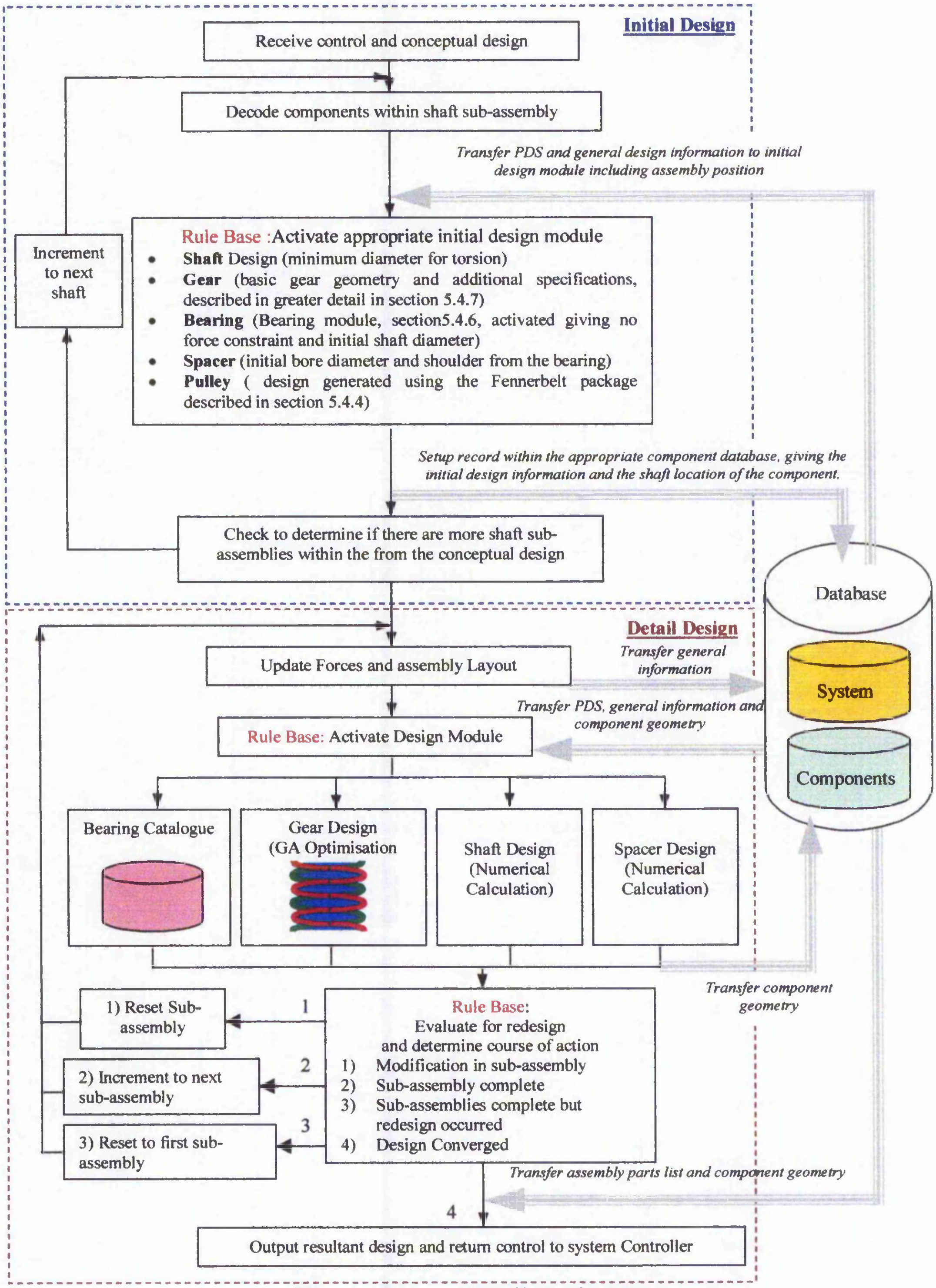


Figure 5.13 Detail Design Structure

In the eventuality that a solution is not achieved, interaction with the designer/ user is required. This interaction is either in the form of redefining the conceptual design, where control of the designs development is returned to the Conceptual Stage Controller or allows various parameters of the detailed design to be altered manually, thus enabling the current design to continue through to completion.

The detail design maintains the modular structure. The structure again comprises multiple modules that perform the detail designs of the different components, e.g. gear pair, shaft, etc. Activation of the relevant module is controlled via the detail design control program, which analyses the conceptual design and sequences the design activities throughout the design process. The sequence of activities is continually evaluated and modified to ensure correct alignment of connected components and that failure will not occur. Figure 5.13 illustrates this structure.

5.4.1 Component Assembly Knowledge

Identification of the components within the design, their position and interaction with one another is an essential process that must be performed. The identification has two major effects over the design process. Firstly, identification of the components allows the structure of the detail design process to be determined. Secondly, the interactive effects of the components within the arrangement has a direct effect on both the dimensions and properties of the components and the detailed design process. Further discussion of these factors will help clarify this point.

Identification of the components

For the detail design process to proceed it is essential to establish the type and number of components to be designed. The assembly of these components will form the final design and relates directly to the conceptual design that has been developed / chosen. A problem arises with regard to the method of generating the parts list. The method should primarily be capable of identifying the necessary components to satisfy the conceptual design and specification, while allowing easy modification of the knowledge, ensuring the design system does not become obsolete.

							Encoded Value																						
	Stage1		Stage2		Stage3		1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768	65536	131072	Arrangement	Transmission Component			Sub-Assembly
	gear	belt	gear	belt	gear	belt	belt2	belt1	pulley2	pulley1	spacer2	spacer1	bearing2	bearing1	gear2	gear1	In/Out	shaft7	shaft6	shaft5	shaft4	shaft3	shaft2	shaft1		Stage 1	Stage 2	Stage 3	Encoded Value
1 Stage Parallel	1											1	1	1		1	1							1	0	2	99	99	132832
											1		1	1	1		1					1							34256
																													0
																													0
		1						1		1			1	1			1				1				0	3	99	99	17610
									1				1	1			1			1									9412
																													0
																													0
2 Stage Parallel	1		1									1	1	1		1	1							1	3	2	2	99	132832
											1	1	1	1	1	1							1						66544
											1		1	1	1		1					1							34256
																													0
		1	1					1		1			1	1			1				1				3	3	2	99	17610
									1			1	1	1		1			1										4836
											1		1	1	1		1					1							34256
																													0
		1		1				1		1			1	1			1				1				3	3	3	99	17610
							1		1	1			1	1				1											2253
									1				1	1			1			1									9412
																													0
2 Stage Parallel - (No Casing)	1		1									1	1	1		1	1							1	7	2	2	99	132832
											1	1	1	1	1	1							1						66544
											1		1	1	1		1					1							34256
																													0
		1	1					1		1			1	1			1				1				7	3	2	99	17610
									1			1	1	1		1			1										4836
											1		1	1	1		1					1							34256
																													0
		1		1				1		1			1	1			1				1				7	3	3	99	17610
							1		1	1			1	1				1											2253
									1				1	1			1			1									9412
																													0
3 parallel	1		1		1							1	1	1		1	1							1	8	2	2	2	132832
											1	1	1	1	1	1						1							33776
											1	1	1	1	1	1						1							33776
											1		1	1	1		1						1						67024
		1	1		1			1		1			1	1			1				1				8	3	2	2	17610
										1		1	1	1		1			1										4840
											1	1	1	1	1	1							1						66544
											1		1	1	1		1					1							34256
		1		1	1			1		1			1	1			1				1				8	3	3	2	17610
							1		1	1			1	1				1											2253
									1			1	1	1		1			1										4836
											1		1	1	1		1					1							34256
		1		1		1		1		1			1	1			1				1				8	3	3	3	17610
							1		1	1			1	1				1											2253
								1	1	1			1	1				1											2254
									1				1	1			1			1									9412

Table 5.5 Identification of Components and Corresponding Shafts

The information required includes the selection of basic component types (i.e. spur gear, bearing, spacer, etc.). This information has to be presented in a way that can be manipulated and integrated throughout the detail design process. The solution applied to this project is to represent the information in tabular form that is converted into a data file for permanent record. The use of spread sheets aids in their development, as demonstrated in Table 5.5.Table 5.5 contains information relating to the formation of the conceptual design into the detailed design. In the table the columns correspond to the components that the final design will comprise, while the rows represent the shafts that they will assemble on. The shafts are the key to the creation of sub-assemblies. Assemble of the components requires a common datum that is not dimension based (as dimensions have not been calculated at this point of the design). The shafts provide this datum. The appearance of a '1' in the cell at the intersection of a component column and a shaft row indicates that the corresponding component and shaft are in the same sub-assembly.

The table is divided into three sections. The left of the table the conceptual designs are broken down into arrangement and transmission components, i.e. 1 stage parallel, 2 stage parallel, etc. and either gear or belt drive. The middle section contains the combinations of components that form the sub-assemblies. The encoded values for each component is given at the top of each column. The final section, on the right of the table, encodes the information contained within the table into a form that the system can interpret.

The table is set-up using a spread sheet, allowing modifications to the information contained within to automatically update the encoded information and thus the sub-assemblies generated by the detail design module of the IIS.

Component Information Encoding

The information about the components and their arrangement is encoded in two forms: stage transmission type and associated components on the shaft. The stage transmission type refers to the basic method that the power is transferred at each stage. These are represented by codes relating to the type as shown in Table 5.6. Initially, during development, the basic methods include parallel gear trains and belt drives.

Transmission Type	Code
Non	0
Double Helical Gear	1
Single Helical Gear	2
Spur Gear	3
Belt	4
Chain	5
Worm and Gear	6
Bevel Gear	7

Table 5.6 Identification Values for the Transmission Components

Component encoding is based on binary representation. Each column has a value and the corresponding component value, which is $\log_2$ of the code, (illustrated below in Table 5.7.)

Component	Pulley	Spacer	Bearing	Gear	Shaft
Column Number	1	5	8	10	16
Component Value ($\log_2$)	1	16	128	512	32768

Table 5.7 Component Codes

The encoded component value forms a unique number telling the detailed design controller which components to include in the design and aids identifying the design module to activate.

Additional tables are used to provide a medium through which information and knowledge can be extracted from the expert and supplied to the system. These tables provide information about the position of a component within a shaft assembly and the forces that act upon the design, (Tables 5.8 and 5.9 respectively.)

	2		4		6		8		10	
Shaft	Belt1		Pulley1		Spacer1		Bearing1		Gear1	
	Position	Mount	Position	Mount	Position	Mount	Position	Mount	Position	Mount
1 Gear in	0	1	0	1	2	1	2	1	3	1
2 Gear lay	0	1	0	1	1	1	1	1	2	1
3 Gear out	0	1	0	1	0	1	1	1	0	1
4 Pulley in	0	1	5	1	0	1	2	1	0	1
5 Pulley out	0	1	0	1	0	1	2	1	0	1
6 Pulley-belt	0	1	0	1	2	1	2	1	3	1

Table 5.8 Component Location

Shaft Number	Shaft Description	Number of Diameters	Change Diameter	Diameter											
				1		2		3		4		5		6	
				H&V	A	H&V	A	H&V	A	H&V	A	H&V	A	H&V	A
1	Gear Input	6	4	0	0	1	0	1	1	0	0	0	0	1	0
2	Gear Lay	5	3	1	0	1	1	0	0	1	-1	1	0	0	0
3	Gear Output	6	3	1	0	0	0	0	0	1	-1	1	0	0	0
4	Pulley Input	5	3	0	0	1	0	0	0	1	0	1	0	0	0
5	Pulley Output	5	3	1	0	1	0	0	0	1	0	0	0	0	0
6	Pulley-Gear	5	3	1	0	1	0	0	0	1	0	1	1	0	0
7	Pulley Lay	5	3	1	0	1	0	0	0	1	0	1	0	0	0

Table 5.9 Reaction Forces

The reaction force table identifies the positions within the assembly that reaction forces will be acting and their initial directions. Thus allowing the initial force analysis to be conducted. The appearance of a '1' indicates that a force can occur at that location on the respective shaft. **H&V** represent horizontal and vertical, while **A** represents axial forces.

Encoding the relationships between components in this form replaces the necessity for intricate rules. Thus, modification to the assembly knowledge can be performed by altering the contents of the respective table then updating its corresponding file in the system.

5.4.2 Initial Design Process

The purpose of the initial design is to provide a starting point from which the detail design of the components and assembly stem. The initial design determines the layout of the components, initialises the component database and develops initial dimensions and physical constraints such as centre distances, speed ratios and reaction forces.

The initial design is based upon the concept design determined at the conceptual design stage. The arrangement and components are retrieved from the database. The transmission assembly is then broken down into stages of the transmission that are identified by the type of shaft used to mount the transmission components. (Each transmission stage constitutes a speed reduction stage.) By reference to the component table, Table 5.5, an encoded representation of the components required for this stage is retrieved. The encoded information is decoded a component at a time, building up the assembly of components required for this stage and shaft. The position of the components within the stage, relative to the shaft is determined from the position table, Table 5.8. It is essential to position the components correctly to ensure that the correct forces are applied to each components as the design of that component in performed. Additionally the position ensures that the final design will be realistic. Force analysis throughout the initial design process is essential to provide the information necessary to perform the component design modules. Identification of where the forces act and their initial direction is established at the same time as the positions of the components are determined. Thus allowing inter-related characteristics of the design to be identified and aid assembly.

Once the layout of the components within a stage, their relations and the identification of forces acting upon them have been established, initial dimensions of the components are determined using initial design modules and procedures. The continued use of modules allows for the adaptation of this initial application to the design of transmission systems and the information within. If the general design of a component is drastically modified the relevant modules can be replaced with the new alternative.

After the initial designs for all the components within the assembly have been performed the layout of the design is updated in preparation for the detailed design process. This ensures that the components will mesh together and assemble correctly. For example the width of the casing and therefore, the length of all the shafts within a gearbox will be dependant upon the size of the longest shaft in the assembly. This shaft will have a knock-on effect throughout the assembly to ensure alignment of the gears. The process is shown in Figure 5.13.

5.4.3 Detail Design Process

Upon the completion of the initial design the Detail Design Control module performs the detail design.

As mentioned previously the detail design is conducted in an iterative fashion. The detail design starts with the first stage of the transmission, performing the detailed design of the components and updating the component database as the iterations continue. If a modification to one of the components has been performed the associated components are reassessed. Once there are no more modifications to the components the design moves onto the next stage. This process is repeated until the final stage of the design results in no modifications or errors.

Upon completion of all the stages the design is again updated to ensure that all the components mesh and locate correctly. If it is found that a modification to one or more of the components is required the process is repeated again. Figure 5.14 provides a schematic of the detailed design process.

In the event of the design persistently requires modification the designer/ user is consulted and presented with the problem. At this point the designer/ user has a choice of three options:

1. modify the problem area manually, after which the current solution will form the final design
2. initiate the conceptual design again with modified specifications (the same specification can be used and an alternative concept selected)
3. re-activate the detailed design process if the designer/ user feels that the design will eventually converge.

When the design converges upon the final design the component list, assembly details and geometric information is transferred to the drawing module.

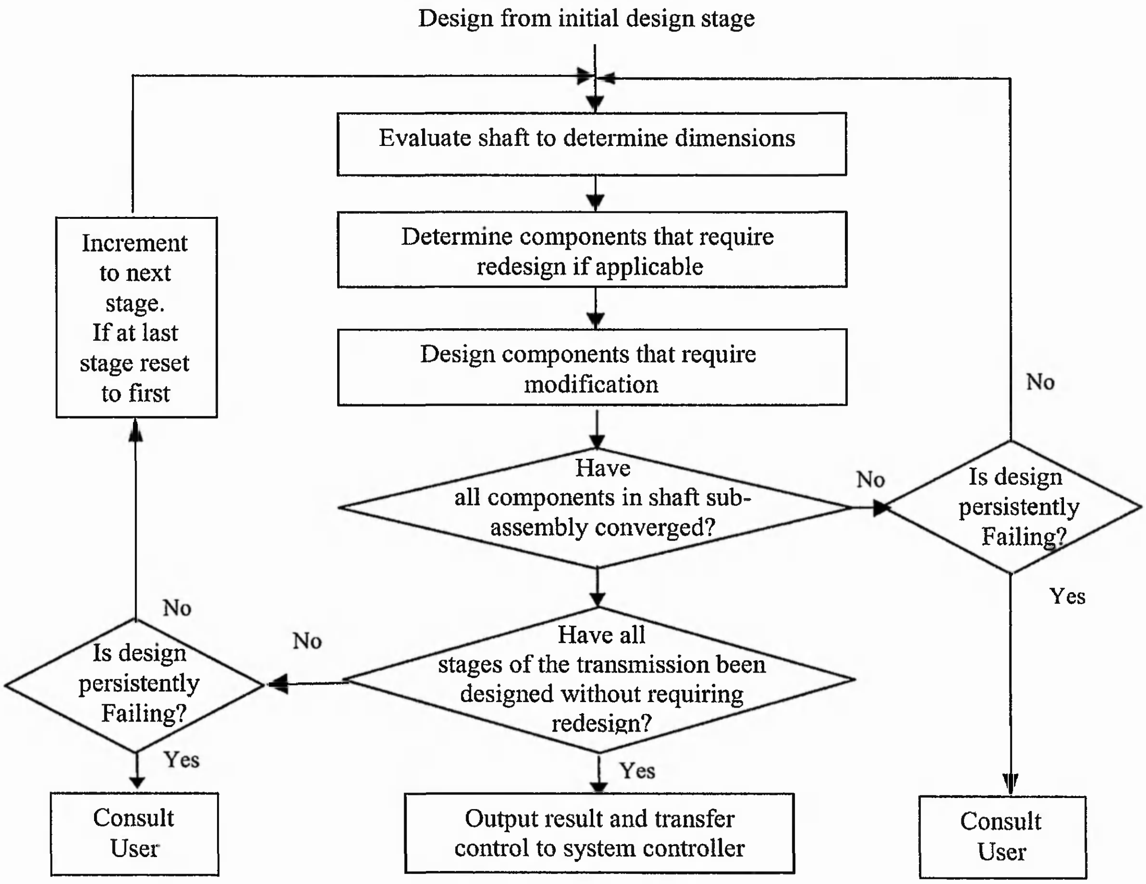


Figure 5.14 Schematic of Detail Design Stage

5.4.4 Pulley Module

Integration of external design packages forms an important part of the IIS. Many design packages exist, the majority of which are confined to a particular part of the total design and are designed to work independently. These packages typically require multiple, pre-determined inputs to be manually supplied to complete their tasks. To integrate the external package within the IIS requires the removal of this independence so that the input and output from the package can be linked to the IIS.

FennerBelt is an independent design package for the design of pulleys and belts and has been integrated into the IIS for mechanical power transmission design.

5.4.4.1 FennerBelt External Package

The FennerBelt package is an external package for the design and selection of pulleys and belts. The package calculates the forces resulting from design specifications from the user, then selects appropriate belts and pulleys from the Fenner electronic catalogue (1990). The result is then presented the solutions to the user in order according to one of three preferences: lowest cost, nearest speed, lowest dynamic load or size. The results from which are then displayed for the user to select a solution. Once the user is satisfied with the selected pulley/ belt drive, full details relating to the solution and its application, both technical and price wise, is presented to the user on the screen with the option to send to printer.

The package appears to adopt a modular structure, allocating different tasks of the pulley/ belt design to separate programs. These programs are linked by a central program that transfers information between programs via files. The structure of the package is as in Figure 5.15.

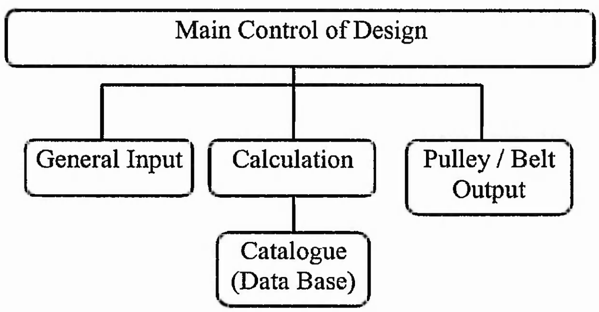


Figure 5.15 Modular Structure of Programs within FennerBelt Package.

Information required to perform the calculations, from which the selection of the pulleys and belts are derived, is requested from the user via one of several input programs. Each program relates to different stages of the design process. The information required to perform the calculations are mainly entirely dependant upon the user. However, there is a facility within the calculation program that analyses the information received and indicates if certain specifications will cause problems. For example, if a service factor is given outside the standard range, it is either corrected by the program or an alternative may be entered by the user. Upon completion of the specification information it is stored in a binary file for the calculation program to retrieve. (A binary file is used instead of text to aid in the data transfer.)

The calculation program retrieves the specification and analyses it to ensure that problems are not likely to arise as mentioned. Calculations are then performed to determine the properties the pulley and belt that must first satisfy then retrieve all the viable solutions from the Fenner catalogue, stored within a database. These solutions are then stored in a binary transfer file for the output program.

The suitable solutions to satisfy the design specifications are formulated ready for selection. The output program retrieves the solutions and corresponding information and ranks them in order according to the preference of the user. Either cost, size, lowest dynamic end load or nearest ratio to specification and presents them to the user. The user now selects the solution from the list and the relative information is transferred to a printer.

5.4.4.2 Integration of FennerBelt with Multimedia

A multimedia front-end that combines tutorials on belt design with the preparation of the specification information for the FennerBelt package has been created. The multimedia front-end has two functions:

- it provides tuition and information through diagrams and text relating to the design principles of belts, pulleys and some of their applications.
- it replaces the input module within the FennerBelt package with a user friendly front end that can provide instruction and tuition on the different requirements.

The multimedia front-end, developed using the LINKWAY multimedia creation package, forms a shell that co-ordinates and structures the pulley/ belt design process.

The specifications are input from the user via the multimedia front end, where explanations of the specifications are available upon request. Upon the completion of the specifications they are stored in a file for transfer to the next stage. The transfer file is then converted into the form required by the FennerBelt calculation program, CALWEDGE. This input information is required in the same format as that created by the FennerBelt input program. To achieve this the file is converted from text to binary and of the necessary structure with the aid of a Pascal program.

The calculation and output programs are now executed to perform the design process and display the results. The schematic in Figure 5.16 illustrates the structure of the integrated program.

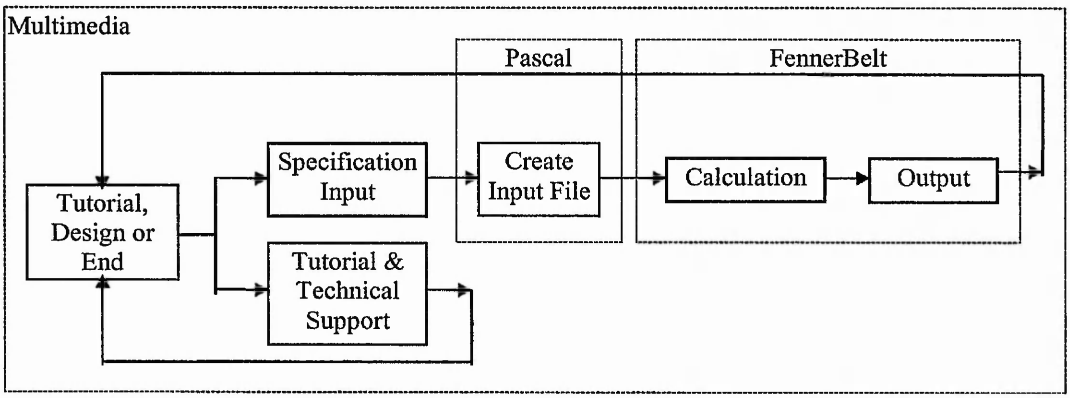


Figure 5.16 Schematic of Integration Between FennerBelt and Multimedia

5.4.4.3 Integration of FennerBelt and multimedia with IIS

Integration of the FennerBelt package with the IIS will allow the design of pulleys/ belts to be performed by a commercially validated system, extracting the solutions from a catalogue of standard parts. The addition of the multimedia front-end adds a user-friendly dimension to the FennerBelt package, providing technical support for non-expert users of the IIS. However, as LINKWAY is difficult to modify and obsolete the multimedia front end has been replaced by a version created with Visual Basic. The Visual Basic front end enables the Windows environment to be exploited and also increases the continuity of the IIS design through the use of a common GUI environment. In addition LINKWAY has difficulty running on networked computers, thus limiting the use of the IIS on computers running Windows NT.

The modular approach is maintained throughout the integration as this maintains the structure of both the IIS and the FennerBelt package while aiding in the development, stage by stage. The design of the pulley is based on the information currently held within the detail design module database, ensuring continuity throughout the design. This information is stored within a transfer file, prior to the commencement of the pulley design.

Upon activation of the module, control of the design process is passed to the multimedia front-end where either technical support may be accessed or the design may commence. Upon commencement, information relating to the design is retrieved from the transfer file and relayed to the user through the front-end, aiding the generation of the remainder of the specifications requested. Once the user is satisfied the specification file is created and converted into the form required by CALWEDGE. CALWEDGE then calculates the properties of the design and selects appropriate solutions from the electronic catalogue described in section 5.4.4.2

A display design selection and storage program is activated upon completion of CALWEDGE by the front-end. This program displays the results from CALWEDGE as described in section 5.4.4.2, then stores the information in a file for transfer to the IIS, where it is stored in the central database.

The design process is controlled through the multimedia interface, which either activates internal procedures for the technical support and information retrieval or initiates external programs for conversion of information, calculation of properties and design selection, display and storage. These stages of the design are structured and connected as in Figure 5.17.

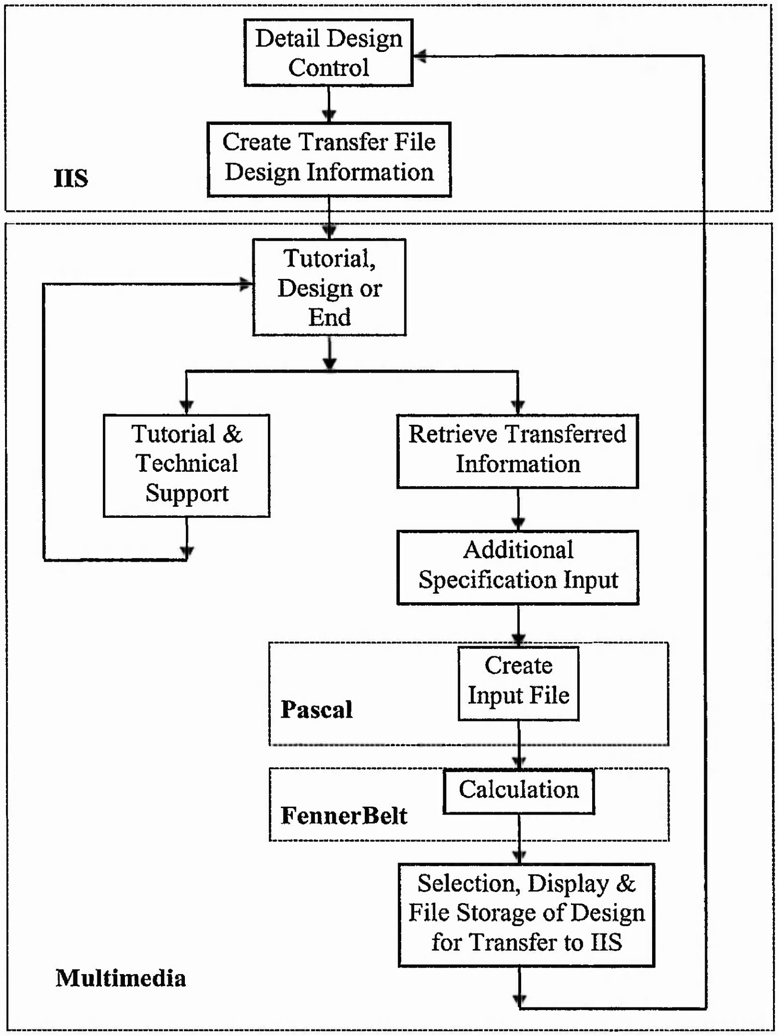


Figure 5.17 Integration of FennerBelt with IIS (Schematic)

Figures 5.18, 5.19 and 5.20 illustrate the user interfaces of the pulley and belt design module.

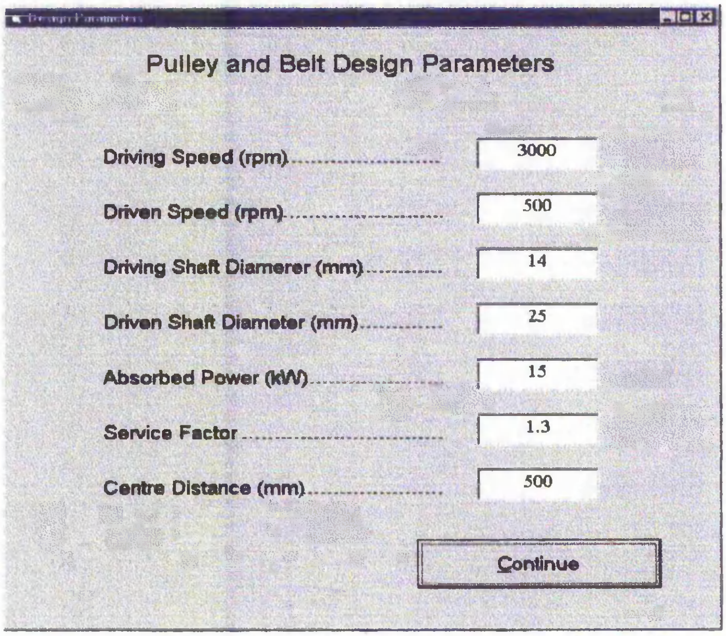


Figure 5.18 Belt Drive Specifications.

Specifications are extracted from IIS except for service factor and centre distance. Service factor is determined from table presented to the user upon selection of the specification, while the centre distance is entered directly from the user.

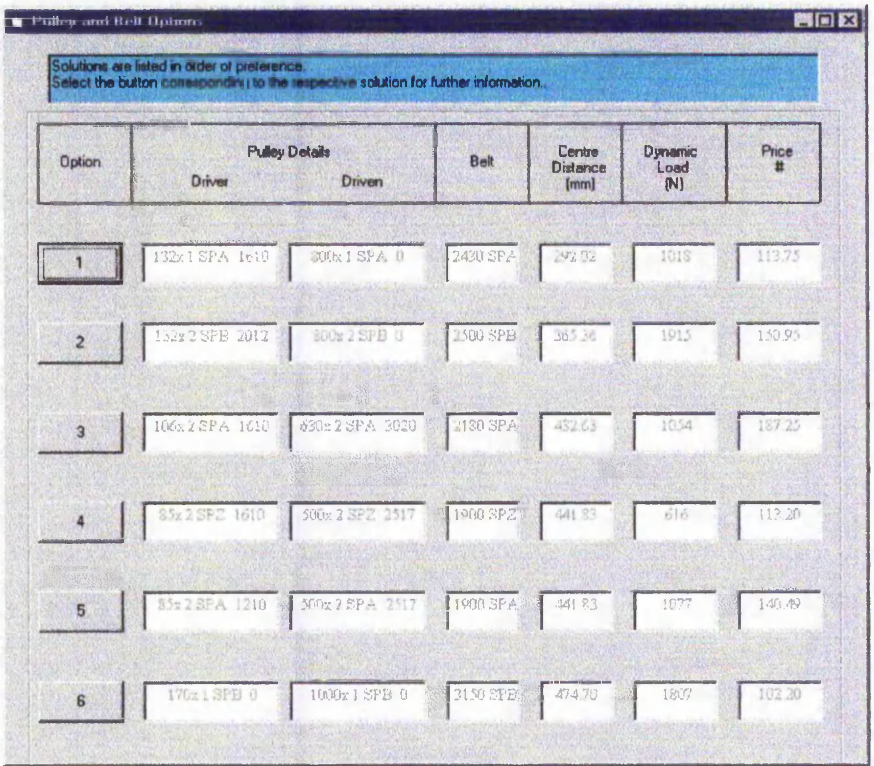


Figure 5.19 Pulley and Belt Options Determined by the FennerBelt Package.

Top six options are presented in order of preference. Options in Figure 5.19 are ordered with preference upon size of pulley. Upon selection of an option detailed information about the selection is presented to the user, as illustrated in Figure 5.20. If the user does not consider this option suitable another option can be selected.

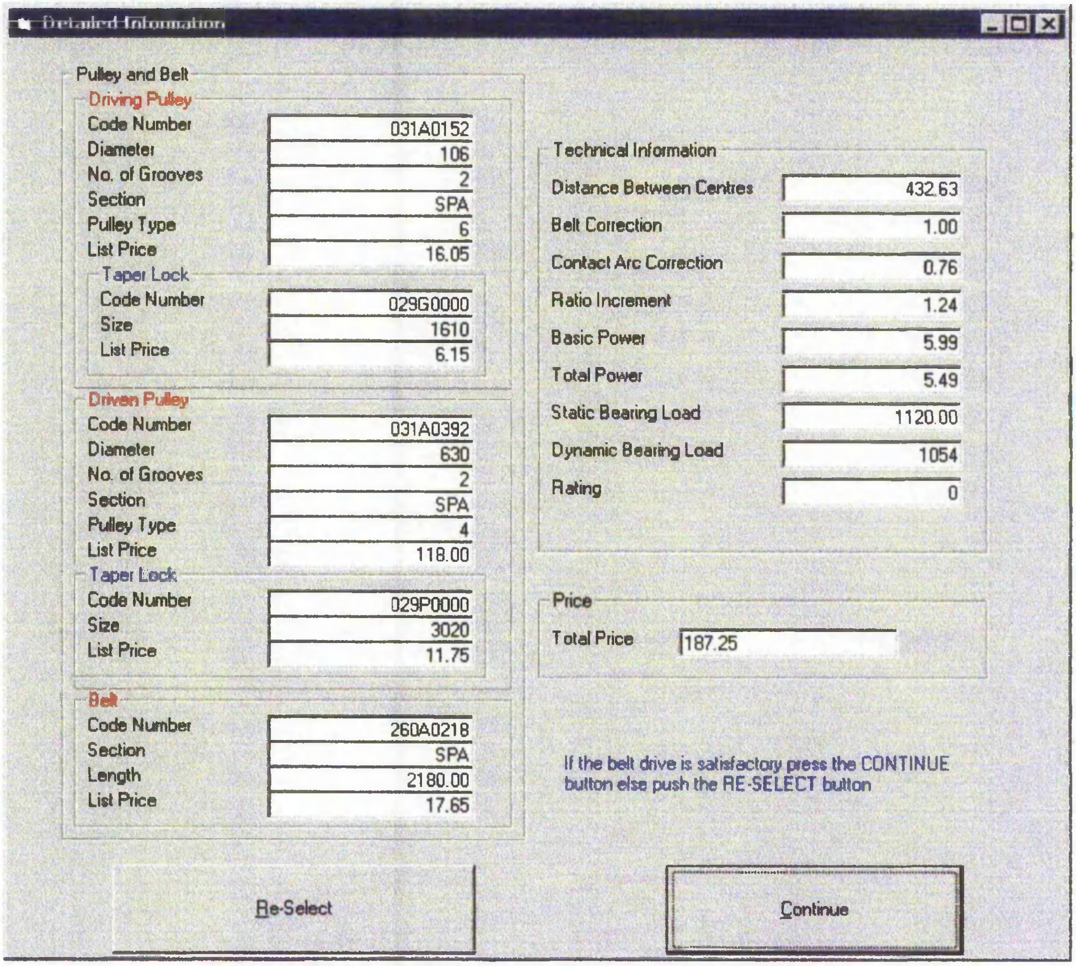


Figure 5.20 Detailed Information on Option 3.

5.4.5 Shaft Module

Within the shaft module a program has been developed to perform the analysis and calculations for the design of a transmission shaft. This program takes the form of a simple design aid that is frequently found in design office to simplify repetitive tasks. The module is developed to demonstrate the simple process of linking an external program with the IIS, thus integrating it within the single environment.

The program defines the dimensions of a shaft according to its purpose and the components that will be located on it, generating the diameters of the shaft to ensure that failure does not occur due to shear, bending or excessive deflection.

Information describing the purpose and forces acting upon the shaft are presented to the program from the detailed design module of the IIS through a data transfer file. This file contains a description of the shaft, comprising the number of diameters, lengths of the diameters, forces acting upon the shaft, the power to be transmitted and material properties. This information fully describes the purpose of the shaft.

The calculation procedure of the program generates the diameters of the shaft based upon the torsion and bending moments acting at each change of diameter. The calculated dimensions of the shaft are checked for deflection, ensuring that the deflection of the shaft will not cause meshing problems for gears mounted upon it. Throughout the design a worst case scenario approach has been applied as an added safety factor. This has been achieved by applying the forces acting upon the shaft at their most extreme positions. The calculation process is explained in greater detail in Appendix B

Upon completion of the shaft design the dimensions of the diameters are compared with those of the existing design to ensure that failure will not occur. If the new values exceed the previous the design is updated and a flag is raised to indicate that the design of other components may require modification or redesign.

5.4.6 Bearing Module

The bearing module is an analysis interface program combined with an electronic catalogue of standard bearings. Upon activation the information transfer file is generated from the Stage Controller containing the forces, mounting constraints and acting upon the bearing together with the type. The appropriate section of the catalogue is scanned for suitable solutions to the constraints stipulated, (bore diameter, speed, load). If a solution is not found that meets the specification the geometric constraints are adjusted until a suitable solution is found that can withstand the performance requirements. Upon location of a suitable solution the information relating to the bearing is transferred to the Stage Controller. Figure 5.21 illustrates the process.

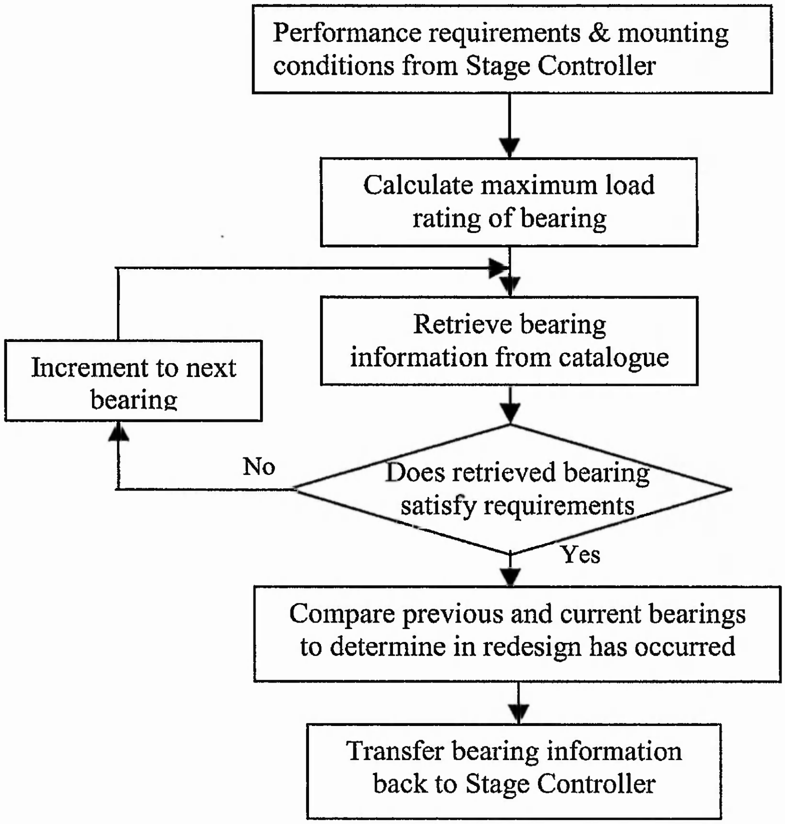


Figure 5.21 Schematic of Bearing Module

5.4.6.1 Interface of Bearing Module

The information received form the Stage Controller, via the information transfer file, comprises of two parts: performance and assembly information and the previous solution. Due to the iterative nature of the design process the bearing module will be activated several times as the design is updated. Therefore, after the first activation details of the previous bearing will be available, together with the current performance requirements and mounting constraints.

Performance and mounting information

The performance requirements of the bearing, which form the selection constraints, include: the forces acting upon the bearing from the transmission system (both radial and axial), the intended life of the transmission system and the maximum rotational speed. Additionally the type of bearing is given to localise the search of the catalogue. The mounting conditions that the bearing is to comply with are the shaft diameter the bearing is mounted upon and the minimum length of this diameter.

Previous solution

Information of the previous solution is returned to the module from the bearing database if available. This provides a means of checking to determine if the design process is settling down and converging upon a solution.

Upon retrieval of the previous solution it is returned to the Stage Controller through the information transfer swap file, thus maintaining the standard structure of the IIS. Within the information returned to the Stage Controller a flag is included that represents change in the bearing indicating to the controller that related components might require redesign.

5.4.6.2 Performance Calculation

From the performance information received from the IIS, which relates to the application of the bearing, the load ratings of the bearing can be calculated. These calculations are dependent upon the type of bearing, the minimum required life and the forces acting upon it. The calculation for each type of bearing is slightly different, thus the type of bearing required is necessary to allow correct calculation. These equations and method of calculation have been extracted and performed as defined by the bearing manufactured SFK and are available in their general catalogue (1989). Once the load ratings have been calculated all the information required for the scan of the catalogue is available, (minimum bore, minimum speed and minimum dynamic load rating).

5.4.6.3 Bearing Catalogue Search

The scan of the catalogue is performed as a simple scan of a database. The database of bearings is stepped through until a solution is found that complies with the specification stipulated and the load ratings calculated. The highest load rating of the bearings on the shaft is taken as the condition to be met. Thus standardisation of bearings within the design of the transmission system is encouraged and provides a safety factor for the smaller bearing. If a solution is not found after scanning the appropriate section of the database the bore constraint is increased. This modification allows the selection of bearings with higher load rating characteristics to be selected. However, redesign of the shaft will be required, but this will be indicated by new bearing being different to the previous.

The electronic catalogue was created using a GUI developed for this application and contains information extracted from the SKF General Catalogue. More information about this interface can be found in Appendix B.4.

5.4.7 Initial Gear Design Module

The initial gear design module again adopts the modular approach to its structure, consisting of a combination of rule, ANNs, numerical calculations and Graphical User Interface (GUI). The structure of the design is coordinated by a central control program that activates the appropriate rule, ANN or GUI dependant on the status of the design. The control program acts as an interface between the different modules, coordinating the development of the design governed by the results obtained from previously activated modules and specifications defined by the user. Figure 5.22 below shows how the modules interface with one another via the control program.

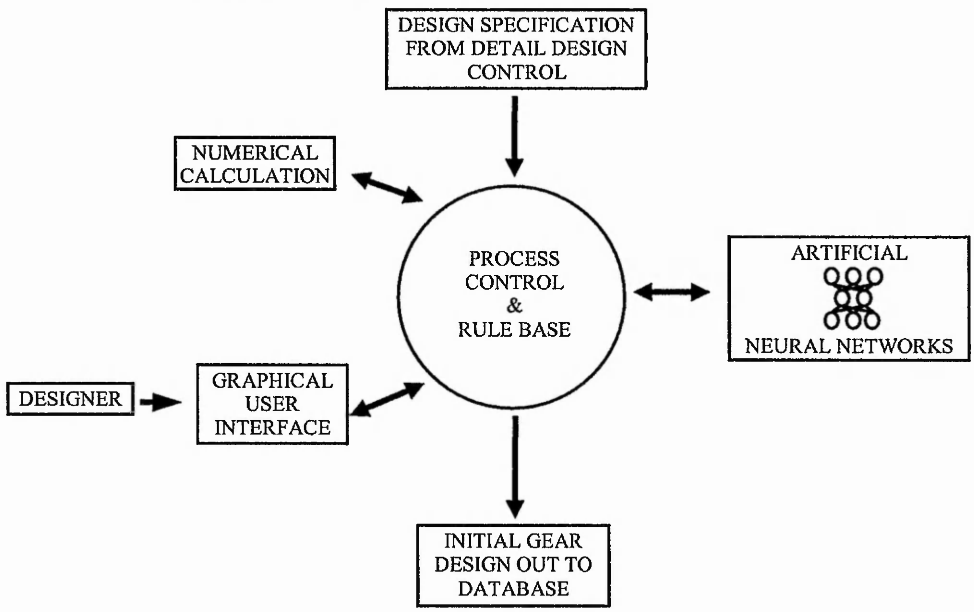


Figure 5.22 Gear Design Module.

The system is developed using a combination of methods and represents a truly integrated approach. The control program is encoded in C++ and forms the centre of the system calling relevant modules when required. ANNs, Numerical Calculation and the Rule Base modules are also encoded in C++ and directly activated by the control program. The GUI is developed in Visual Basic, taking advantage of the graphical capabilities and user interface facilities. The database at present consists of an ASCII file for storage and transfer of information extracted from the central data base which is updated and expanded as the design progresses by interrogation procedures embedded within the control program. The design system follows an iterative process, refining a number of parameters, dimensions and features of the design until the design is correct.

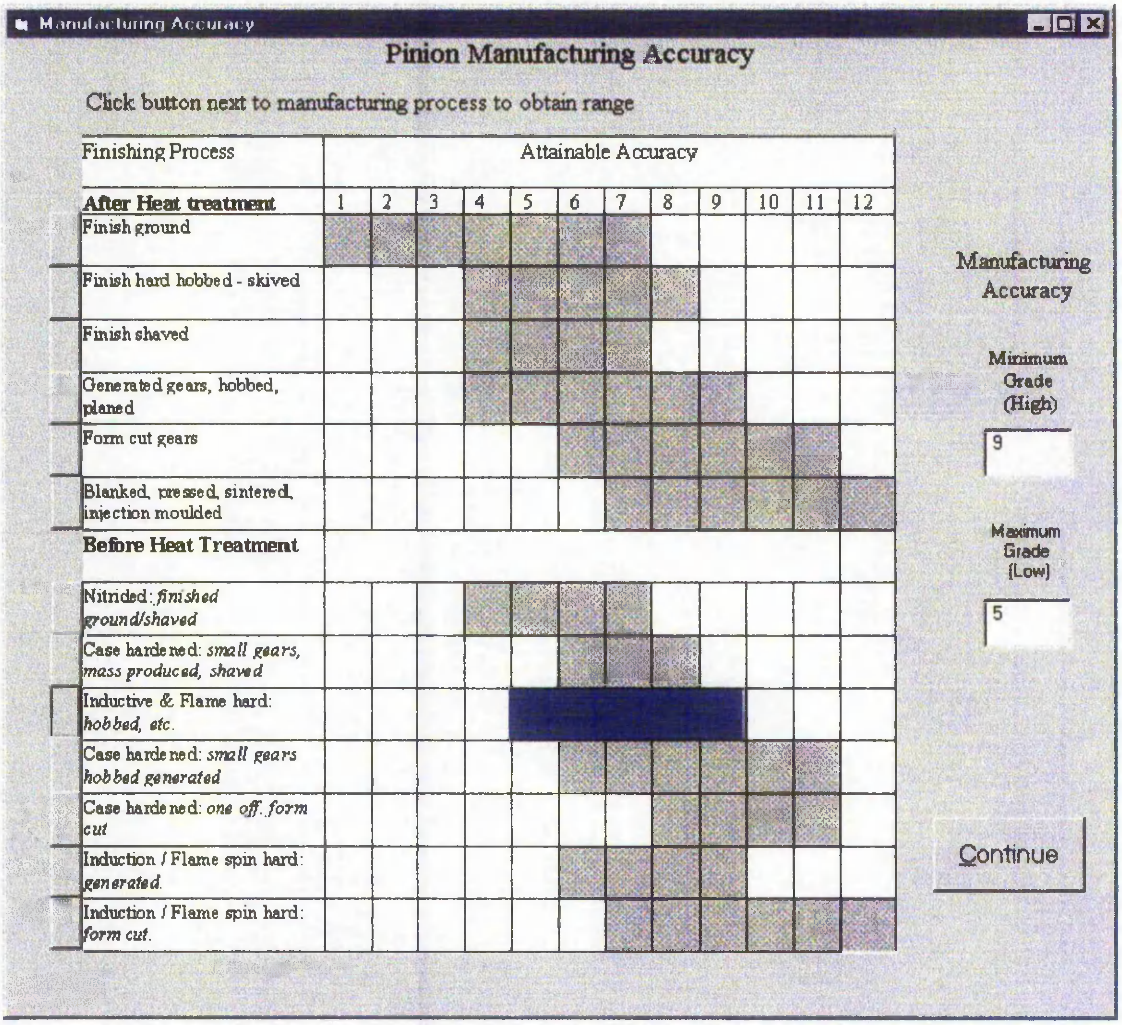


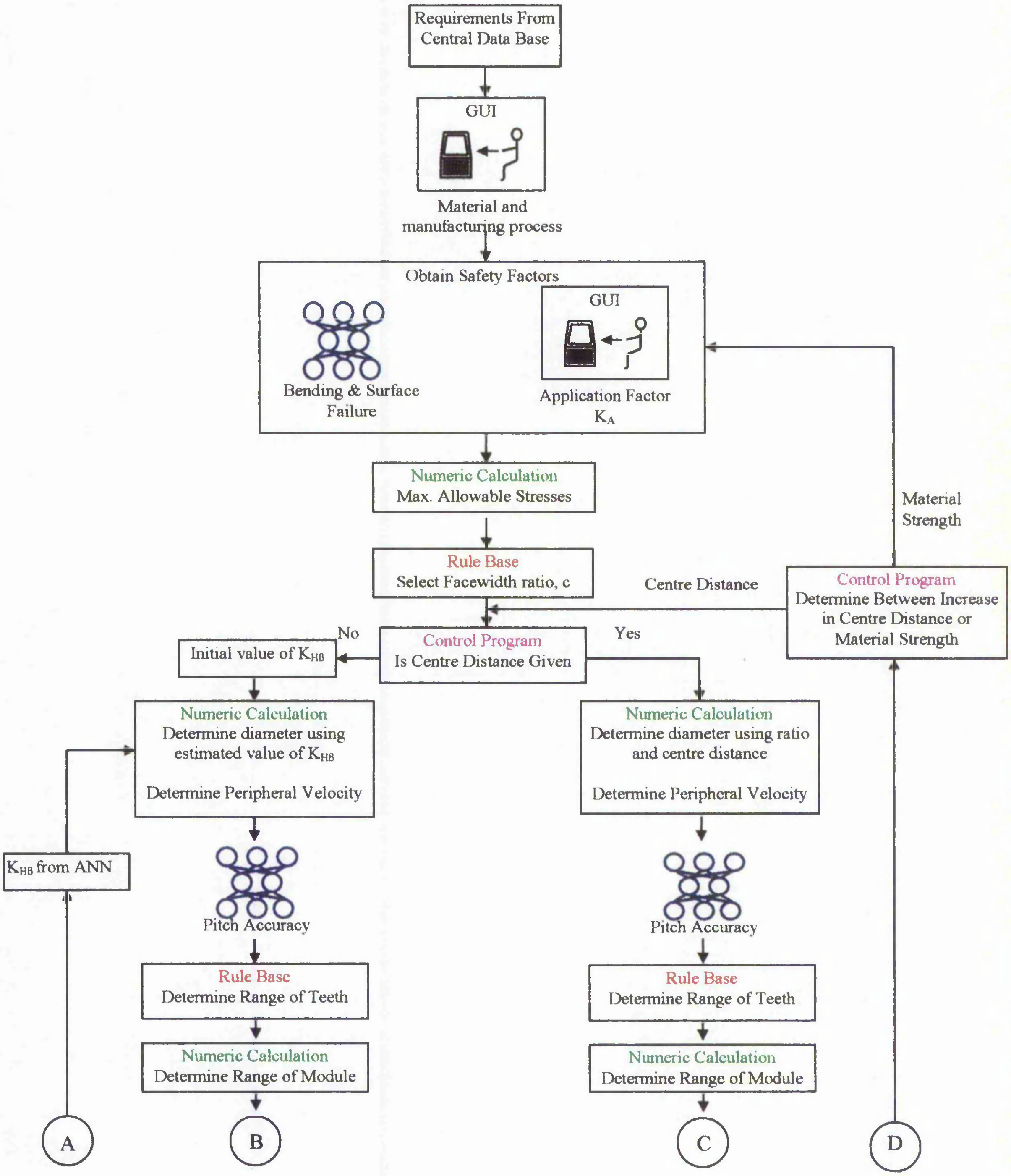
Figure 5.23 GUI to aid the selection of the Manufacturing Process (Technical information taken from ESDU88033)

Upon activation, information about the required design is supplied to the gear module from the Detail Design Controller. This information is a combination of the user's specifications and parameters defined at the conceptual design stage and by other design modules within the transmission design system. Additional information about the gear application and desired performance is required before the system proceeds with the initial and final design of the gears including: the application factor, materials, heat treatment, manufacturing process, initial pressure and helix angles and tooth modifications in the form of end relief or crowning. This information is of the form that is impractical or impossible (at present) to program and is therefore, extracted from the designer via the GUI. The GUI serves two functions within the system. Firstly, it displays the relevant information about a factor of the design in a clear and convenient manner, as illustrated in Figure 5.23. These interfaces remove the time consuming process of wading through design manuals and specifications to locate the relevant information. Also examples of typical usage are available to aid in the selection of the correct criteria. The second function that the GUI performs is that it gives the designer a sense of control over the final design. This allows for variations from the standard design. For example, the selection of materials and manufacturing processes or the application factor.

Once the gear specifications have been obtained the geometric design of the gear is performed. The design process is shown in Figure 5.24, utilising a combination of production rules, ANNs and numerical calculations.

The production rules have two functions. Firstly, to structure the design process forming the design controller where they are used to iterate the design sequence to achieve optimum dimensions by making decisions, modifying factors of the design to achieve a feasible solution. Secondly, the production rules are used for the allocation of the correct value to a parameter, dependant upon existing parameters of the design, e.g. facewidth ratio dependant upon gear arrangement within the transmission system, the material properties and chosen manufacturing process. The rules are used to encapsulate knowledge about the design process that is tangible and well definable, forming the backbone of the design system.

The addition of ANNs to the gear design process provides an advantage that is not available in the conventional gear design packages. The ANNs within the system are used to encode information that is of a graphical form, Figure 5.25 for example. Many designers accumulate information from a variety of sources often in graphical form. The information held within the graph may be extremely helpful to the design of a successful component but the original data that the graph is constructed from may be unobtainable. In this circumstance an ANN trained directly from the graph forms a desirable solution allowing the manual design process to be followed. The ANNs used are capable of encapsulation information of either a simple and complex nature. Three networks are used to represent the information contained within four graphs. These graphs can be found in Appendix B. Activation of the appropriate network is governed by the design controller.



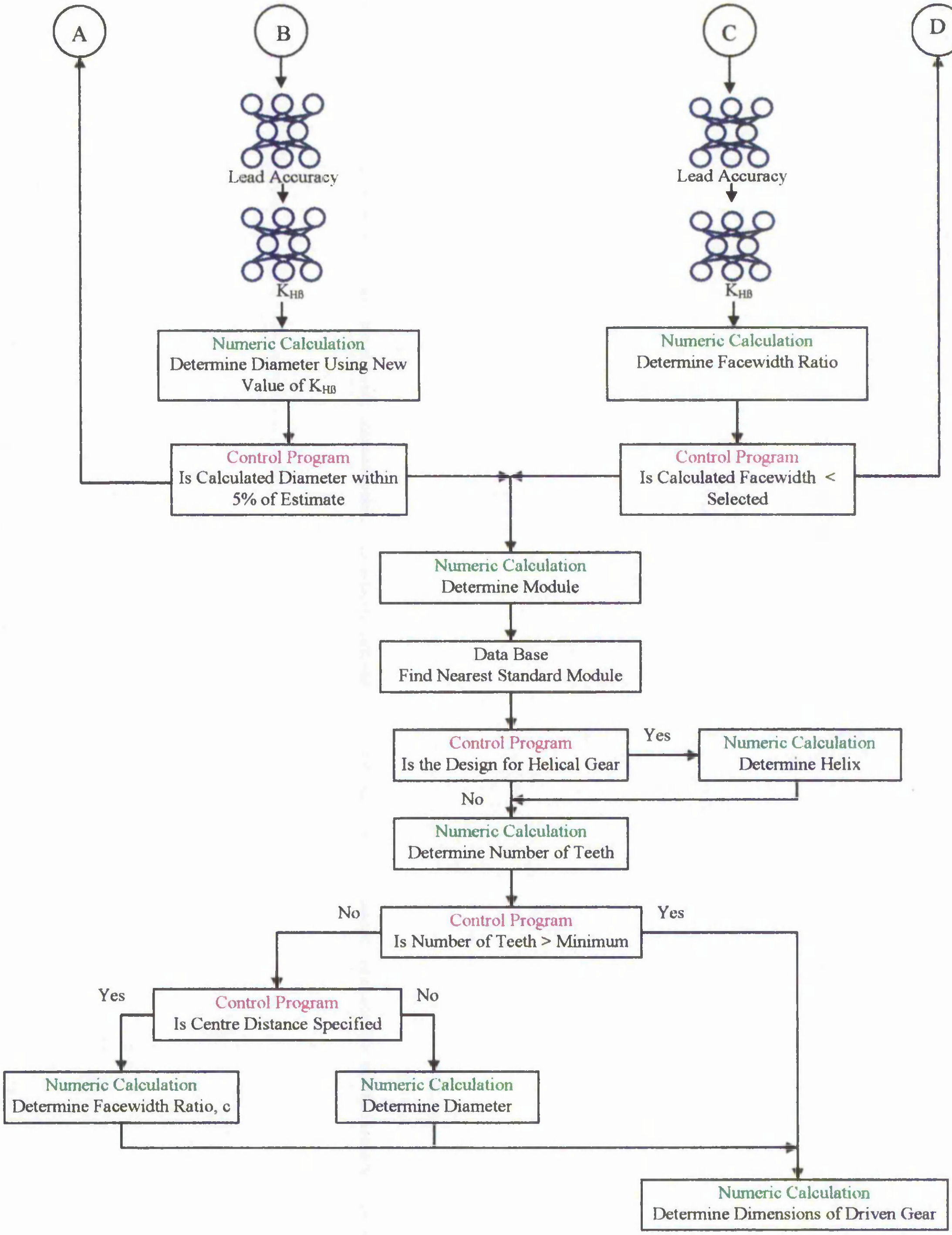


Figure 5.24 Schematic of Gear Design

Activation of the appropriate procedure, which houses the numerical calculation, is governed by the control system. The numerical calculations fall into two classes. The first class relates to the design of the gear geometry and properties, while the other checks the final design. The gear geometry calculations (derived in Appendix B) provide the tangible features of the design, e.g. dimensions and number of teeth. Final check of the design is performed to BS 436 part 3 (1986) to ensure that failure, in terms of bending and contract strength, does not occur. If it is identified that failure does occur the reason for failure is identified and modifications to the gear specification are made by the production rules prior to redesign.

Once it has passed BS 436 part 3 then the final design and control is passed to the Detail Design Controller for continuation of the transmission design.

5.4.7.1 Rule Encapsulated Knowledge

The knowledge is stored in a series of production rules constructing a complex cognitive system. The production rules used within the system are a series of conditions and actions in the form of IF...THEN statements where the entire collection of rules forms the knowledge base. The firing of a rule is determined by the control program, which acts as an inference engine.

As discussed in section 3.3.2 the production rules are used for two purposes, design progress control and specific information. Within the gear design module the rules relating to the progress control logically fire sets of information rules or activate calculation or ANN procedures, thus structuring the designs development. The conditions for these rules are generated either by the design itself as it develops or are supplied by the user or detail design controller prior to the activation of the design module. Below are examples of the control rules with conditions generated from the design and supplied by the IIS.

IF *calculated number of teeth* > *calculated maximum*
THEN *increase tooth module to next standard size.*

IF *centre distance is defined*
THEN *structure the design based upon centre distance.*

The information rules encapsulate information relevant to a specific condition. The information takes the form of either a value of an equation that generates a value. The examples below illustrate the forms.

IF *gear type is spur*
THEN *tooth form factor is 3.4.*

IF *centre distance is not defined*

THEN $d = Z_H Z_E Z_\varepsilon \left(\frac{2000 T_H}{c \sigma_H^{\;2}} \left(\frac{u+1}{u} \right) K_A K_v K_{H\alpha} K_{H\beta} \right)^{\frac{1}{3}}$ Equation B.25

The production rules are thus used to encapsulate knowledge that is well defined, where and action can be directly related to a specific condition.

5.4.7.2 Data Preparation and Application of ANNs

Within the detailed design of a gear, ANNs have been applied to the encapsulation and generalisation of information conventionally contained within a graphical design aid. Such as in Figure 5.25.

These design aids are a common means of representing the relationship between multiple design parameters. However, these relations are often obtained experimentally making a calculable relation difficult and complex to construct if the original data is unavailable. ANNs are used to implement the knowledge into the design system. Backpropagation ANNs are used for two reasons:

i. they are capable of representing complex information and interpolating between values.
ii. the defined input/ output structure of the network suits the modular approach to the system structure. Thus allowing the information retained by the ANN to be easily modified by removing the existing ANN and replacing it with the update.

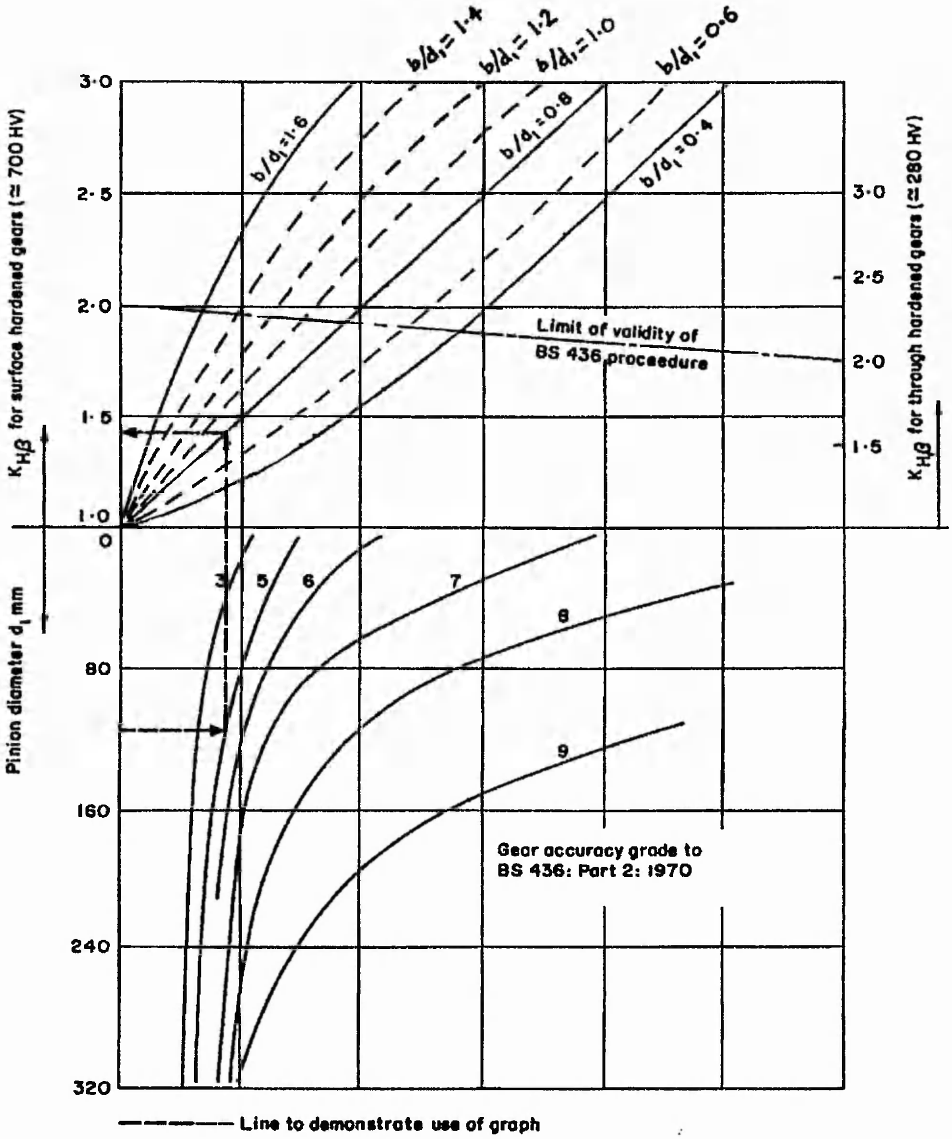


Figure 5.25 Graphical Design Aid for $K_{H\beta}$ to be Represented by Network

Information from the graphs is extracted and constructed into the training data. The input pattern to the network consists of the minimum information required to use the graph, while the output is the resultant information obtained from the graph. Use of the above diagram and corresponding ANN is discussed in section 3.2.3.3.

Table 5.10 contains the input and output pattern structures required by the four ANNs used for the initial gear design.

ANN purpose	**Input Pattern and Number of Elements**	**Elements in Layer**		**Transfer Function**	**Output Pattern and Number of Elements**
		1st Layer	**2nd Layer**		
Face load factor for contact stress, $K_{H\beta}$	1. Pinion diameter 2. Gear accuracy grade 3. Facewidth ratio	10	2	*tanh*	1. Face load factor for contact stress
Pitch Accuracy	1. Peripheral velocity 2. Gear Type	18	20	*tanh*	1. Pitch Accuracy (Upper) 2. Pitch Accuracy (Lower)
Lead Accuracy	1. Module 2. Facewidth	10	5	*tanh*	1. Lead Accuracy
Safety Factor	1. Probability of failure 2. Gear accuracy grade	6	7	*tanh*	1. Safety factor for bending stress, S_H

Table 5.10 Initial Gear Design ANN Input/ Output Patterns

Generation of the training data must account for two requirements of the BP network.

i. the training data is taken from across the entire scope of the graph to ensure generalisation by the network within this area. This is due to the BP networks poor capabilities for extrapolation.

ii. the amount of training data should be sufficient to allow the network to generalise for the entire area, but not too great as to cause over training.

Together with the data from the graph, additional data derived by interpolation may be required to increase the networks ability to predict as expected. Prediction expected by the network is a demanding task as prediction in two or more dimensions may be required, as in Figure 5.26.

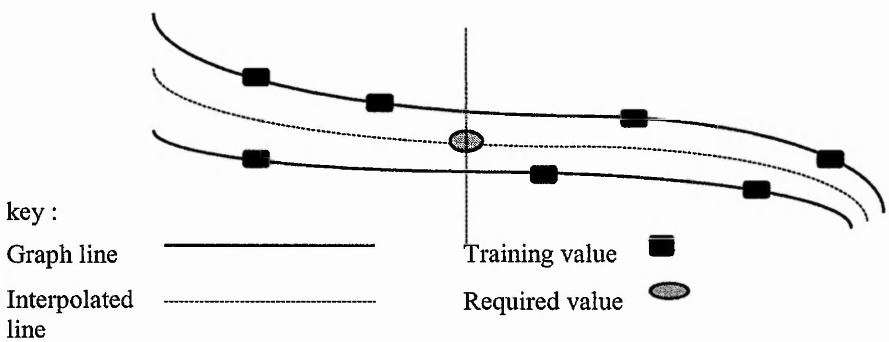


Figure 5.26 Example of Prediction

Training of the networks is a difficult process as several factors affect the performance of the network and is discussed in section 3.2. One of the factors is the creation of adequate training data. The person generating the data (either the system designer or a user wishing to modify information contained) should generate sufficient data to generally cover the desired scope initially. The amount can be increased if training is unsuccessful. The remaining factors relate to the network and the training process. These make network training difficult. For this reason the training aid, GEN-NEU, described in section 3.2.3 was developed.

5.5 Gear Design Optimisation Using Genetic Algorithms

The optimisation of a gear requires the adjustment of several factors that affect the performance of the gear. These include the general dimensions and properties, such as the material, number of teeth, pitch circle diameter and tooth facewidth together with more detailed modifications such as shifting the tooth's profile. Optimisation of the gears performance is achieved by the modification of these parameters, however, as the effects of the parameters are both directly and indirectly related, the search area to obtain an optimum is immense. The EP approach described in section 4.4 has been applied to the optimisation of this component design.

The design optimisation module acts in the same fashion as for the other design modules. The stage controller prepares the necessary information for the design within a data transfer swap file. The detail design is performed and the result transferred back to the stage controller. Upon completion the returned design information is checked for modification and redesign.

5.5.1 Identification of Problem and Parameters

Prior to the application of the GA to the solution of the problem, the optimisation criteria and parameters must be identified.

The problem. The optimisation process will adjust parameters that define the characteristics of the gear to fulfil the following criteria:

1. Achieve the minimum facewidth and module while complying with BS 436 part 3 -not exceeding the permissible bending and contact stress on the teeth.
2. Bending stresses within both the pinion and wheel gear will be approximately equal.
3. Contact ratio is to be maximised in order to reduce vibration and noise.

4. Speed ratio is to be maintained.
5. Centre distance of gear pair to be maintained for fixed centre distance and minimised for variable.

These criteria form the basis for the fitness functions, which determine the success of the configuration of parameters and therefore, the probability of this configuration progressing to the next stage of the search process.

Optimisation Parameters Before the GA can be applied to the problem it is advisable to establish the purpose of the various parameters. This enables the fitness functions to be sculpted to suit the application, reducing convergence criteria and ensuring that only critical parameters are encoded into the genome.

Descriptions of the main parameters that affect gear performance are listed below:

Facewidth

The load that is transferred through the gear pair will be distributed across the width of the tooth (the facewidth). Therefore, the smaller the facewidth the greater the pressure acting on the tooth, resulting in a higher stresses. The facewidth also affects the contact ratio of helical gears, enabling more teeth to be engaged simultaneously, A small facewidth also contributes to an decrease in the sensitivity to uneven distribution of the load across the tooth width.

Module

The module represents the ratio of the pitch circle diameter to the number of teeth on the gear and therefore, defines the size of the teeth. The lower the module, the smaller the teeth and thus the higher the stresses acting on them for the same power transfer.

Addendum Modification Coefficient,(profile shift, x)

Addendum modification relates to the shifting of the tooth's involute profile. The profile shift is the movement of the line of action (pitch circle diameter, PCD) from the reference circle by the amount $x.m$. This movement allows adjustments to be made between the mating gears in order to maintain a centre distance between the gears while a particular module is maintained, this is known as modified centre-distance gearing. Or the shift can reduce the minimum number of teeth due to its higher load bearing capacity.

Modification to the tooth profile is possible to manufacture using the same tools as standard to gears by adjustment to the set-up.

Shifting the profile can be positively outward from the gears centre or negatively toward the centre. Figures 5.27a to 5.27d demonstrate the movement of the profiles.

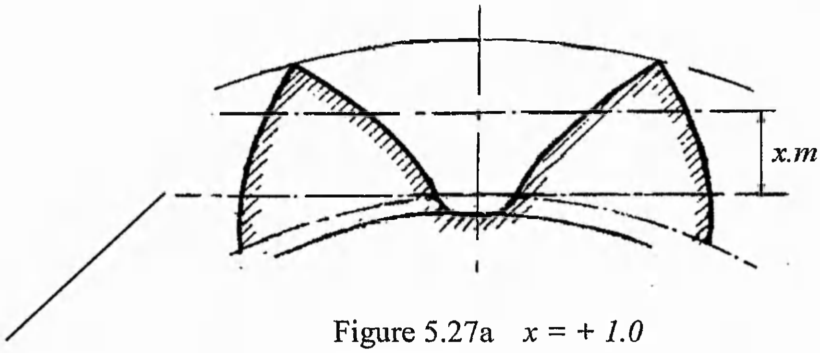


Figure 5.27a $x = +1.0$

(excessive modification, leads to reduced contact ratio)

Datum Line

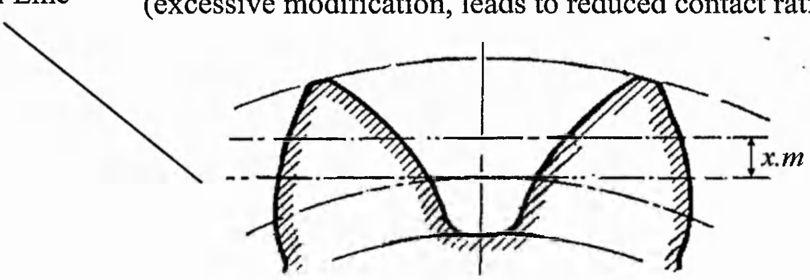
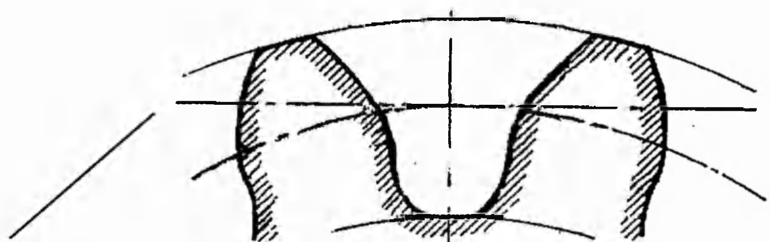


Figure 5.27b $x = +0.5$

(reduced undercutting)



Datum Line

Figure 5.27c $x = 0$

(standard Gear)

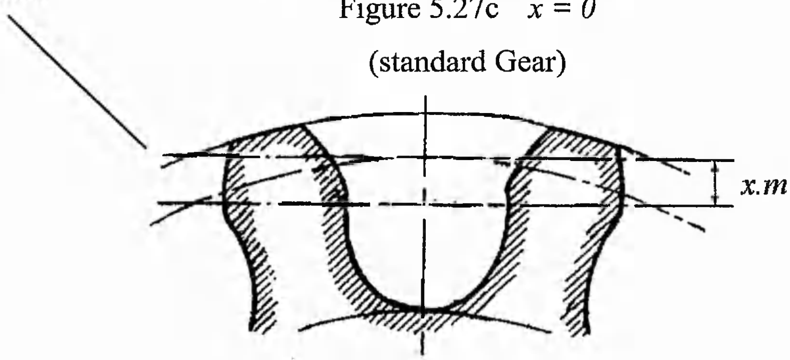


Figure 5.27 $x = -0.5$

(negative profile shift, requiring undercutting)

Profile shift for standard centre-distance gearing. With this type of modification the centre-distance remains the same as for the standard gears and is not affected by the profile shift. Thus the positive shift of the one gear (usually the pinion, as it has the lower number of teeth and will therefore, reduce the necessity for undercutting) is equal to the negative shift of the mating gear, $x_1 + x_2 = 0$. This is generally suitable for high speed ratios where the tooth forces are lower, however, for lower speeds the forces in the wheel gear may exceed permissible limits due to undercutting caused by negative shift. Figure 5.28a demonstrates this form of modification.

Profile shift for modified centre-distance gearing. This type of modification allows for a fixed centre-distance to be achieved when different to the standard. Again the profile shift of the pinion gear is positive, but the shift of the wheel will compensate for the difference in centre-distance, $x_1 + x_2 \neq 0$. This effect is demonstrated in Figure 5.28b.

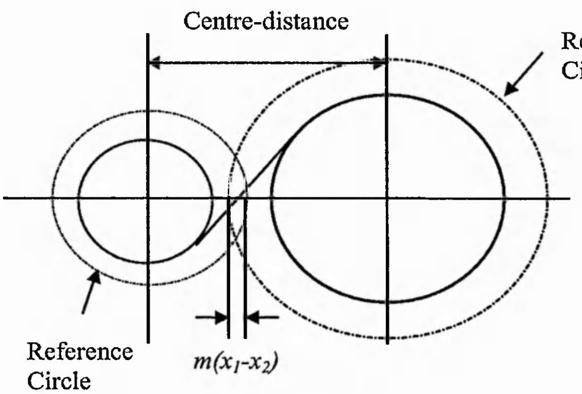


Figure 5.28a
Negative modification

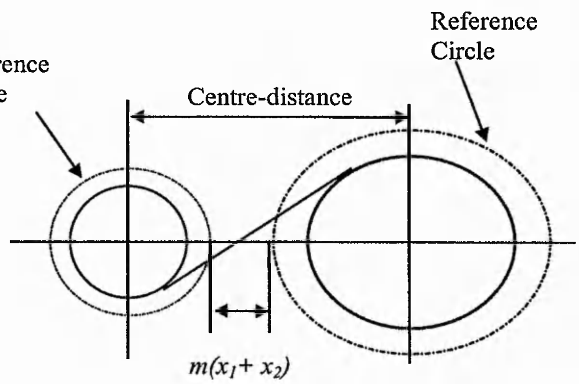


Figure 5.28b
Positive modification

Addendum Coefficient, ha^*

The addendum coefficient, ha^*, defines the length of the tooth in terms of the module, as illustrated in Figure 5.29. The addendum (h_a) is taken as the region from the reference circle to the tip of the tooth for standard profiles. For standard gears the pitch and reference circles are coincident, but for modified teeth the pitch circle is shifted by the amount of profile shift.

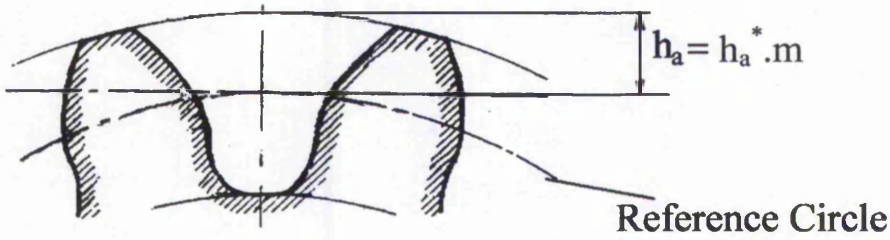
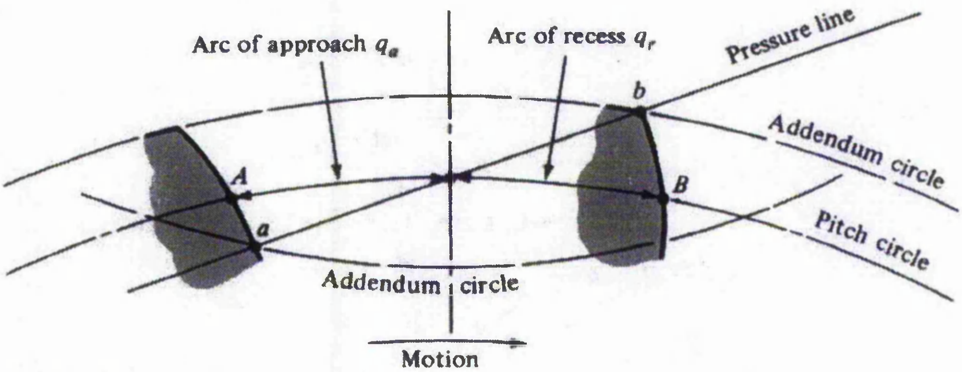


Figure 5.29 Addendum Position for Standard Gear.

Modifying the addendum affects both the contact ratio of the gear pair and the bending stress acting upon the tooth, both of which increase with an increase in h_a. With the increase in tooth length the thickness of the tooth's crest will decrease, as the involute profile must be maintained. This can lead to weakening of the tip and cause failure due to excessive load. Additionally lengthening of the tooth will result in increased velocities and sliding, thus increasing wear. Therefore, modifying the tooth's length can have multiple conflicting effects.

Contact Ratio

For motion transfer between teeth to be continuos at lease one pair of teeth must be in contact at all times to carry the load and during some portions of the cycle two or more pairs will be engaged at the same time. Engagement of more than one pair of teeth distributes the load and allows smoother transmission. Figure 5.30 below illustrates the line of contact that the transferred load acts along.



Arc of Action, $q_0 = q_a + q_r$

Figure 5.30. Arc of Action

The parameter that measures this aspect of the gear tooth's action is called the contact ration, ε. The contact ratio comprises of two components, ε_α and ε_β. The first component takes into account the geometry of the gear pairs tooth profiles and the mating line of action between them. This action involves the transverse pressure and helix angles together with addendum modification and profile shift. This component of the contact ratio comprises of the effect from the pinion, ε_1 and the effect of the wheel, ε_2.

$$\varepsilon_\alpha = \varepsilon_1 + \varepsilon_2 \qquad \text{Equation 5.1}$$

where $$\varepsilon_1 = \frac{z_1}{2 \cdot \pi}\left(\tan\alpha_{a1} - \tan\alpha_t'\right)$$

and $$\alpha_a = \cos^{-1}\left(\frac{d_b}{d_a}\right) \quad \text{and} \quad \alpha_t' = a\cos\left(\frac{d_2}{2} \cdot \frac{(d_1 + d_2)}{a'.d_2}\right)$$

Definition of the variables can be found in Appendix B.

The method of evaluating the working transverse pressure angle, α_t' has been modified from the original equation in MAAG (1990) to take account of the addendum modification coefficients, which are frequently applied within the optimisation process. The modification marginally alters the true centre distance, but not sufficient to cause problems for the optimisation, while simplifying the calculation procedure. Proof of the modification to the equation can be found in Appendix B

The second components of the contact ratio takes into account the overlap of the teeth due to the helix angle and facewidth. If the gear is a spur gear this will have no effect upon the overlap. The overlap is defined by equation 5.2.

$$\varepsilon_\beta = \frac{b \cdot \sin\beta}{\pi \cdot m} \qquad \text{Equation. 5.2}$$

The total contact, ε_γ ratio is obtained by combining its two components, as in equation 5.3 below.

$$\varepsilon_\gamma = \varepsilon_\alpha + \varepsilon_\beta \qquad \text{Equation. 5.3}$$

Effects of the Addendum Modification and Profile Shift on the Contact Ratio

Modifying the tooth profile by either shifting the profile or by adjusting the length of the addendum, will have a direct effect upon the Contact ratio. As previously mentioned the greater the contact ratio, the better the distribution of the load and transfer of power.

Therefore, it is desirable to find the optimum contact ratio by the adjustment of the addendum and the line of contact. Figures 5.30a to 5.30c below demonstrate their effects upon one gear pair

Number of teeth,	pinion : 22	wheel	: 66
Pressure Angle	: 20°	Module	: 5 mm

The combined effects of these adjustments on the contact ratio are complicated and therefore, a search for the optimum also becomes complicated and tedious as the solutions and search area increases. Combine the effects of the helix angle, module and number of teeth and the evaluation of the parameters becomes increasingly complex.

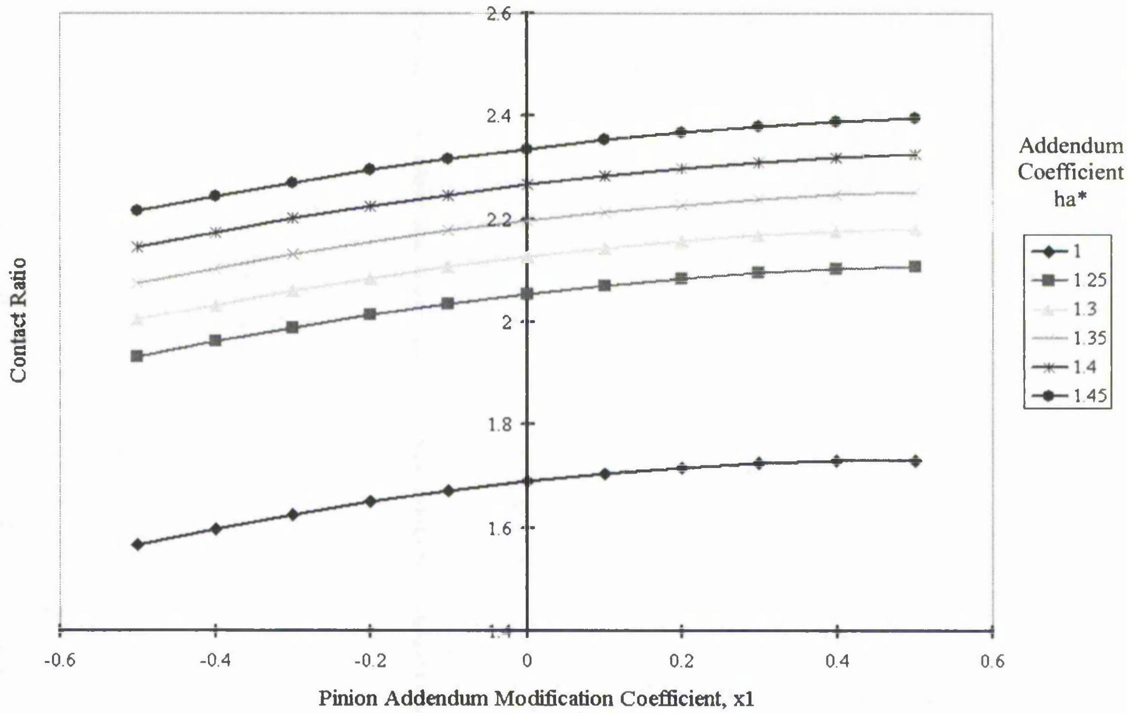


Figure 5.31a Effects of Profile Shift and Addendum Modification on Contact Ratio for Standard Centre-Distance ($x_1 + x_2 = 0$)

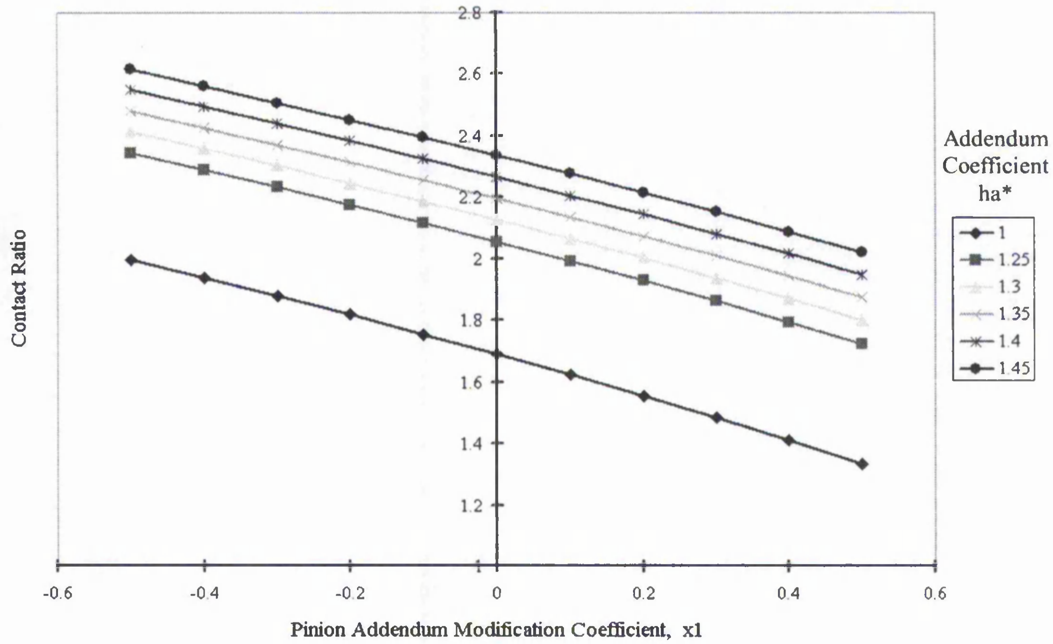


Figure 5.31b Effects of Profile Shift and Addendum Modification on Contact Ratio for Modified Centre-Distance (Changing x_1 only)

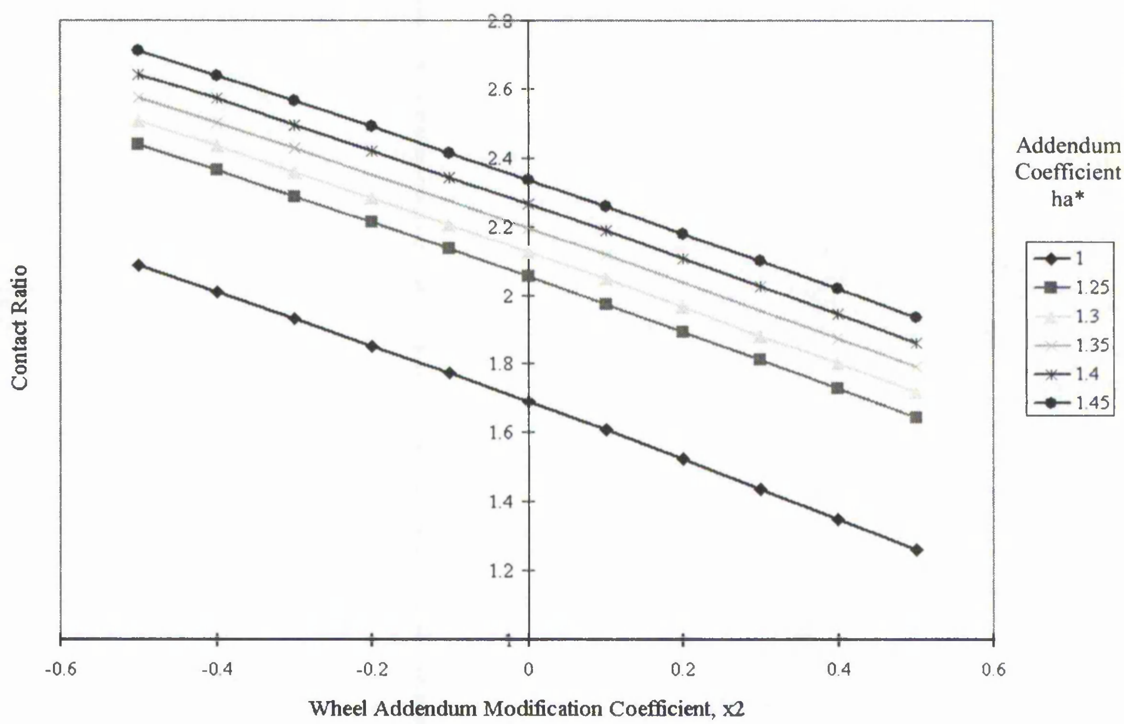


Figure 5.31c Effects of Addendum Modification Coefficient and Addendum Modification on Contact Ratio for Modified Centre-Distance (Changing x_2 only)

Rack Tip Radius

The rack tip radius is related to the root radius of the generated tooth and is used during calculation of the tooth's performance, due to the complexity of the calculation of the actual tooth's root radius. The root radius is positioned within the clearance region of the dedendum between the base circle and the involute of the tooth. Decreasing the root radius will have the effect of increasing the local stresses in this area, thereby forming possible failure. Therefore, as the rack tip radius and the root radius are related modifying the rack tip radius will alter the performance of the tooth while providing a feature that my be easily modified and measured during manufacture.

Pressure Angle, α

The pressure angle represents the angle at which the force is transmitted between the teeth. The angle is measured from the tangent of the pitch circle, illustrated in Figure 5.30. In this research, the angle at which the force is transmitted has been limited to one of four standard values, 17½°, 20°, 22½° and 24°. Changing the angle has a variety of effects upon the tooth and the stresses acting upon it. Increasing the pressure angle makes the tooth thicker at the base and increases the radii of curvature at the pitch line. The effects of this will be to improve the bending strength and enable greater load to be carried before the contact stress exceeds permissible limits. An additional effect of the increasing the pressure angle is that it reduces the contact ratio.

Helix Angle, β

The helix angle is the inclination of the tooth to the direction rotation. The range of the angle is practical up to a limit of 45°. Increasing the angle has the effect of increasing the axial load possibly to a point where the deflection of the shaft is intolerable or the size of bearings will be too great. However, increasing the helix angle has the effect of increasing the contact ratio, due to the number of teeth that are engaged increasing, thus increasing the overlap, ε_α.

Interference

Modifications to the geometry of the gear teeth will cause the meshing of the gear pair to alter. Due to excessive adjustment of parameters and incorrect combinations, interference problems can arise. Interference occurs when the point of connection between the pinion and wheel is not on the involute profile, i.e. the tip of the wheel connects with the root radius of the pinion tooth. The affects of interference can be fatal to the design, causing teeth to crack and fail, therefore, interference is not permitted in the design of a gear pair.

Interference can be determined in a number of ways, however, for the purpose of this research it is identified through two methods. Firstly, when the tip of the tooth projects beyond the length of path of addendum contact, g_{a2}. The condition is modelled in equation 5.4.

$$g_{a2} \leq a \cdot \sin \alpha_t' \qquad \text{Equation 5.4}$$

where: g_{a2} = the length of path of addendum contact for the wheel

a = centre distance

α_t' = working transverse pressure angle

the second method is to express interference in terms of the contact parameter, k, where the limiting condition is: $k \leq 1.0$

Details of these conditions can be found in Appendix B

5.5.2 Knowledge Representation Using the Evolutionary Process

The combination of effects from the modified parameters, upon the optimisation of a gear pair, is an example of knowledge representation. The GA approach is able to compensate for the knowledge required to adjust the parameters to obtain an optimum, provided the fitness criteria correctly defines the desired result. Determining the amount and type of profile shift is one area that the GA is being used to replace knowledge, especially when modification to the tooth length is simultaneously taking place. Unknown to the EP there are two types of profile shift resulting in different effects upon the tooth, the reasons for these shifts are mentioned in the previous section. The types fall into two main categories: $x_1+x_2=0$ and $x_1+x_2\neq0$. Additionally for $x_1+x_2\neq0$ there are four permutations of x_1 and x_2 being positive and negative. The general rule is that x1 is kept positive to reduce the bending stresses and to avoid undercutting of the pinion. Within the application the parameter x_1 is not restricted to being positive. The reason for this is to establish if the search can simulate the same reasoning as just stated, therefore, demonstrating the evolutionary principle in action.

5.5.3 Application of the Genetic Algorithm

Application of the GA to the gear problem is performed in two steps. Firstly, the creation of the genome and fitness functions, (these model the problem), and secondly, the characteristics of the GA process that control the search process. These characteristics are addressed with regard to the gear application within this section and affect the population size and crossover and mutation operators.

5.5.3.1 Genome Composition

The parameters that correspond to the genes of the genome have been identified in section 5.5.1. These genes combine to form either a seven or nine gene genome which are represented in Figures 5.32a and 5.32b below.

Gene Number

1	*2*	*3*	*4*	*5*	*6*	*7*
(4 bit)	*(4 bit)*	*(4 bit)*	*(2 bit)*	*(5 bit)*	*(3 bit)*	*(4 bit)*
width	module	addendum modification	pressure angle	helix angle	rack tip radius	profile shift x1

Figure 5.32a Genome Composition for Fixed centre Distance

Gene Number

1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
(4 bit)	*(4 bit)*	*(4 bit)*	*(2 bit)*	*(5 bit)*	*(3 bit)*	*(4 bit)*	*(4 bit)*	*(3 bit)*
width	module	addendum modification	pressure angle	helix angle	rack tip radius	profile shift x1	profile shift x2	pinion teeth

Figure 5.32b Genome Composition for Variable Centre Distance

These parameters have the effect of altering the geometry, performance and contact ration, but for the purpose of this project the exact relationships have not been determined. As mentioned in section 5.5.2, it is the purpose of the GA to find the combination of parameters that will produce the optimum design. However, some consideration is required with regard to the method of encoding the relevant information into the genome.

Two configurations of genome have been constructed for the two different types of gear design; fixed centre distance and variable centre distance. As can be seen in Appendix B, the optimisation of the fixed centre distance, the number of teeth for the pinion and wheel are determined by the value of the module and the speed ratio (speed ratio remaining constant). The difference in centre distance is compensated for by the profile shift of the teeth. For the profile shifts to be determined the amount that one of the profiles will be shifted must be known. The value of the pinion profile shift, x_1, is included within the genome allowing the value of the wheels shift, x_2, to be calculated. The process is defined in Appendix B.

For the variable centre distance, the relation between the pinion and the wheel is limited to the speed ratio. Due to the lack of relationships the number of teeth and amount of profile shift on the teeth cannot be calculated, therefore, these parameters are included within the genome forming another dimension to the search.

The genome has been constructed so that both fixed and variable centre distance gear designs can use the same genome. Parameters that are used within both designs form the first section of the genome, while the two extra genes, (x_2 and z_1), required for the variable centre distance are located at the end. This structure allows the addition parameters to be ignored when not required. Their permanent inclusion within the genome will not affect the GAs process when being ignored as during crossover the position of the transferred bits is maintained. Therefore, the contents of these genes cannot exchange places other genes and affect the search.

The method of encoding the information within the genes is dependant upon the affect that the parameters has upon the design. These considerations include limits, set values and resolution. Taking these into consideration the genome has been constructed as follows:

Gene 1 uses the percentage difference approach with a 28 % deviation. Giving a wide search area for the first tier of the optimisation, which is halved for each subsequent tier. Thus increasing the resolution.

Gene 2 uses the predefined list, allowing 7 positions movement either side of the start-point. As the module has a dramatic affect upon the design, 7 positions movement will give sufficient search space for the optimum. If extra is required, it will be achieved in the second tier of the optimisation.

Gene 3 uses the percentage deviation approach providing a range of ±0.35 about the initial value. The resolution of 0.05 is doubled for each subsequent tier of the optimisation. This range is sufficient for the tooth addendum coefficient as the standard value is 1.0, thus the range allows for both shorter and longer teeth to be catered for. No emphasis is placed on the longer teeth to determine if the fitness functions are working correctly. Teeth with higher contact ratios should be developed and thus longer teeth.

Gene 4 uses the predefined list, allowing the movement of 4 positions. This covers the full range of standard pressure angles.

Gene 5 uses direct encoding applied to a starting set-point. The helix angle can vary 28 degrees about the set-point with a resolution of 4 degrees for the first tier of the optimisation, which is doubled for the second but with half the range.

Gene 6 uses the direct approach providing a range of values from 0.05 to 0.4with a resolution of 0.05. It is possible to cover the full range viable values within the search. Limits are imposed if the values are not viable, (the limit is given in Appendix B which is influenced by the pressure angle).

Genes 7 & 8 use the direct modified encoding approach. The value is scaled to give values between –0.7 and 0.7 with a accuracy resolution of 0.1 about an initial set-point. The resolution is doubled and the range each subsequent tier of the search. The total range covered by the search spans all practical values that would be applied to the design.

Gene 9 uses direct encoding approach applied to a starting set-point. The number of teeth for the pinion gear can vary by up to 3 either side of the set-point. This range is sufficient as the number of teeth has a dramatic affect upon the gear design. The pinion gear is used within the genome as the wheel design is based upon the pinion.

Limits have been set on several parameters to prevent them from producing values that will result in certain failure of the design or undesirable values. Table 5.11 contains the limited parameters and their restrictions.

Parameter	**Upper Limit**	**Lower Limit**
Module	Position 34 within list	Position 0 within list
Addendum Modification Coefficient (x1 and x2)	1.0	$x_{min} = \frac{\text{No. teeth}}{\text{Min. No. teeth.}}$ or -1.0 if x_{max} is greater
Helix Angle (angle limited)	User defined (typically<45°)	0
Helix Angle (force limited)	√2.Tangential force	0

Table 5.11 Parameter Limits.

5.5.3.2 Fitness Functions

The fitness function, as mentioned earlier, has a dramatic effect upon the convergence of the search and the parameters contained within genome's that are transferred through the generations. Therefore, the selection of criteria that comprise the fitness function must mirror the desired characteristics of the gear that are to be optimised by the search. The fitness criteria can be classified into two categories: *rating criteria* and *conditional criteria*, which apply either a gradient or step function respectively to determine the level of fitness. The rating criteria fitness functions produce an individual fitness rating based upon the value of the genome with respect to the function limits representing the proximity of the genome to the target. The conditional criteria determines if the genome can pass or fail. If the result from the conditional fitness function is fail the total fitness of the genome is set to 0. The total fitness of the genome is generated by summing the individual fitness values for each criteria. The criteria used for the gear optimisation are listed in Tables 5.12 and 5.13.

Rating Criteria	**Max. Fitness (100)**	**Min. Fitness (0)**
Gear Facewidth	Smallest Value	Largest Value
Centre Distance	Smallest Value	Largest Value
Equal Stress	Smallest Value	Largest Value
Contact Ratio	Largest Value	Smallest Value
Speed Ratio Accuracy	Smallest Difference	Largest Difference

Table 5.12 Rating Criteria

Fitness Conditional Criteria	**Pass**	**Penalty**
Contact Stress (Pinion)	≤ Permissible Stress	> Permissible Stress
Contact Stress (Wheel)	≤ Permissible Stress	> Permissible Stress
Bending Stress (Pinion)	≤ Permissible Stress	> Permissible Stress
Bending Stress (Wheel)	≤ Permissible Stress	> Permissible Stress
Tip Crest	≤ 0.3 module	< 0.3 module
Helix Angle	Ft>√2.Fa	Ft<√2.Fa
Length of path of Addendum Contact (Wheel)	$g_{a2} \leq a \cdot \sin \alpha_t'$	$g_{a2} > a \cdot \sin \alpha_t'$
Contact Parameter, k	≤ 1.0	> 1.0

Table 5.13 Conditional Criteria

Gear Facewidth – Fitness criteria (Minimise dimension)

The fitness is evaluated by comparison of all the facewidths within the population. As the objective is to decrease the width, the highest fitness is given to the smallest value and visa versa. The fitness for the remainder of the population is determined linearly along a line between the two extremes as in Figure 5.33a.

Centre Distance – Fitness criteria (Minimum value)

The fitness for the centre distance is evaluated in two different ways depending upon whether the design has been specified as fixed or variable centre distance.

For fixed centre distance the fitness criteria is driven by the difference between the optimised designs centre distance and the specified target. The fitness function is set to minimise the difference by applying an inversely proportional function, giving a higher fitness rating, the smaller the centre distance.

The fitness function for a design with a variable centre distance is driven by an emphasis to reduce the centre distance to the smallest in the search. Again an inversely proportional function is used, giving the maximum fitness rating to the smallest centre distance, then proportionally lower ratings until zero is given to the greatest centre distance.

These functions are illustrated in Figures 5.33b and 5.33c respectively.

Tooth Tip (Crest) Thickness – Fitness condition

A side-effect from the modification of the tooth's profile is that the thickness of the tooth's tip can become increasingly smaller, as shown in Figure 5.33a. This is caused by a combination of lengthening of the tooth and shifting the profile, which can causes the tip to become weak and susceptible to extremely high stresses and failure. To prevent this happening, a fitness function is introduced to represent this side-effect during the evolution of an optimum solution. The crest thickness is monitored and a limit of 0.3 times the module set as a minimum. The limit set as the threshold of a step fitness function, which is shown in Figure 5.33d. This fitness profile gives a null fitness value if the limit is exceeded or a positive value if not. By using the step function the fitness function requires no extra information about the effects that the profile has upon the design, interceding only when the design increases the possibility of failure setting the fitness value is set to zero, resulting in the genomes ejection from the population.

Success of gear – Fitness condition

For the gear train to succeed the actual contact and bending stresses for both the pinion and wheel must be below the permissible levels. Therefore, if the permissible stress is exceeded the fitness for the relative criteria must represent a failure of design, equal zero. The original development of this fitness criteria considered the values from the entire population according to the profile in Figure 5.33e. However, as the permissible stresses are not constant throughout the population it is not truly representative. Therefore, the pass or fail profile in Figure 5.33f has been adopted to represents the fitness of each genome irrespective of the rest of the population.

Equal Stresses – Fitness criteria (Minimum Difference)

The fitness function to encourage the characteristics that cause the bending stresses in both the pinion and wheel gears, uses the absolute difference between the two to determine the fitness value. Evaluation of the fitness is performed in a similar manner to the geometry, the fitness is inversely proportional to the stress difference. Example the minimum difference is rewarded with a high fitness. Figure 5.33g illustrates the profile.

Speed Ratio Accuracy – Fitness criteria (Minimum Difference)

The fitness function acts in a similar manner to that for equal stresses. The emphasis is placed upon maintaining the speed ratio of the gear pair as close to the design's specification as possible. Therefore, higher fitness ratings are given to the designs with ratios of little difference to the target. Calculation of the ratio takes into account the effects of profile shift, which can increase of decrease the ratio without altering the number of teeth. The function profile is illustrated in Figure 5.33h.

Helix Angle – Fitness condition

A fitness condition is introduced for helical gears to limit the search to within the boundary stated in the design specification. The helix angle is constrained by either a maximum axial force or a maximum angle. The function used applies a step profile using either a relation between the limiting axial force, Fa and the calculated tangential force, Ft as the threshold or the specified angle. If Fa is used as the constraint the threshold is taken as $Fa = Ft/\sqrt{2}$. If $Fa > Ft/\sqrt{2}$ the design has failed due to the helix angle exceeding 45° otherwise the design is passed. 45° is taken as the limit due angles exceeding this are not being generally used in industry. If the maximum angle is stated in the design specification, this value forms the threshold, causing failure of the design if the helix angle exceeds it. The profiles are illustrated in Figures 5.33i and 5.33j.

Contact Ratio – Fitness criteria (Maximum)

The fitness criteria is to achieve the maximum, as this will lead to decreased vibration and increased distribution of power transfer. The fitness is taken relative the population with the highest value obtaining the maximum fitness and the lowest value zero. the remaining populations fitness are proportional to the extremes as illustrated in Figure 5.33k.

Interference– Fitness condition

The fitness condition to deter interference applies a penalty to the fitness rating of the genome if the limit of the path of addendum contact is exceeded or if the contact parameter exceeds 1.0. The function is illustrated in Figure 5.33l. Provided the conditions are met and interference does not occur, the fitness condition has no affect on the fitness rating. However, if the conditions are not met each function will impose a penalty to decrease the overall fitness of the genome.

The total fitness value for the genome is obtained by summing all the rating criteria values. This value is then scaled using the combined fitness scaling function. In section 5.5.4.1 this function was found to give the best results.

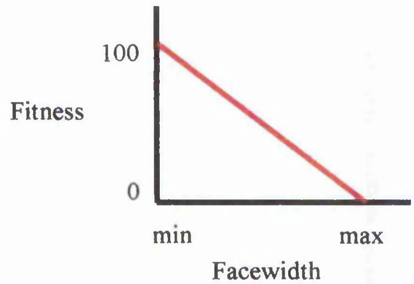


Figure 5.33a
Gear Facewidth

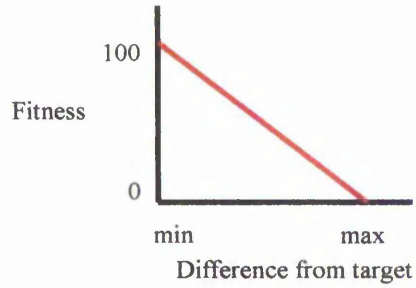


Figure 5.33b
Centre Distance Fixed

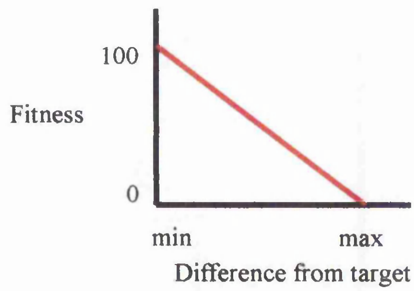


Figure 5.33c
Centre Distance Variable

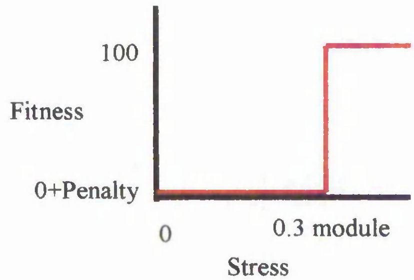


Figure 5.33d
Tooth Crest

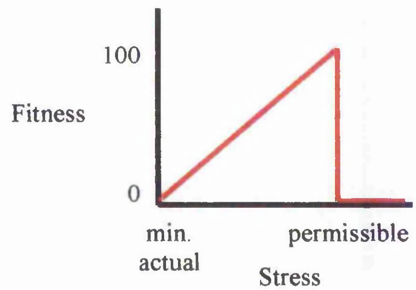


Figure 5.33e
Stress (Not True Representation)

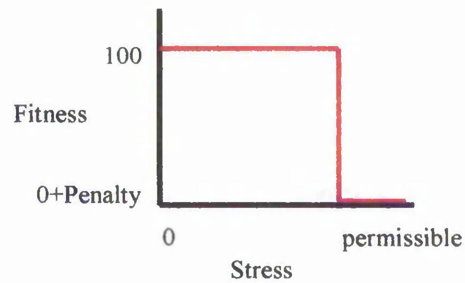


Figure 5.33f
Stress (Individual)

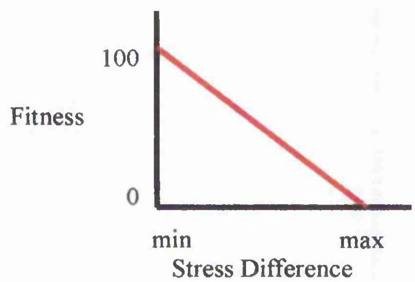


Figure 5.33g
Equal Stresses

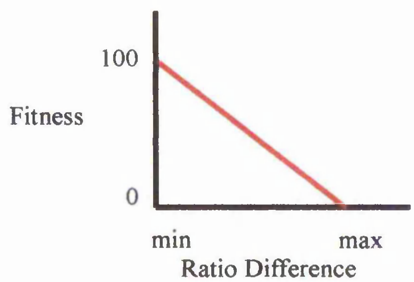


Figure 5.33h
Speed Ratio Accuracy

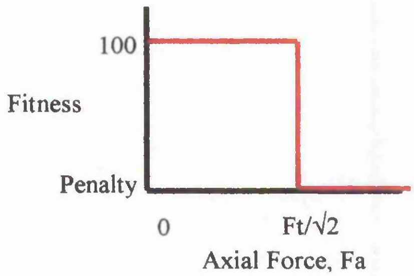


Figure 5.33i
Helix Angle (Axial Force)

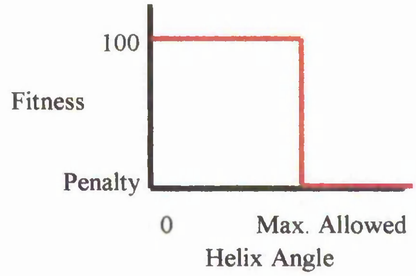


Figure 5.33j
Helix Angle (User Defined)

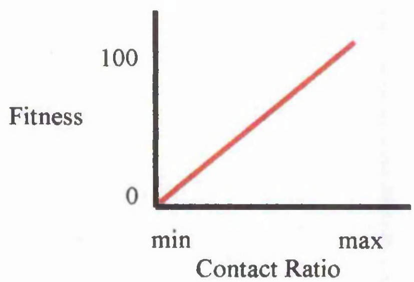


Figure 5.33k
Contact Ratio

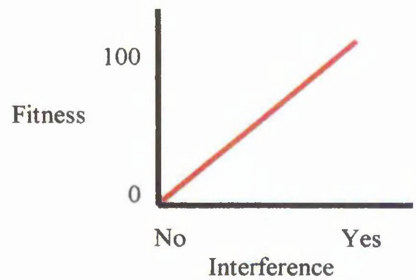


Figure 5.33l
Interference

5.5.3.3 Genetic Algorithm Controlling Factors

The controlling factors of the GA process are the size of the population and the rate at which crossover and mutation occurs. Defining these factors is sometimes difficult (as mentioned in sections 3.1.1 and 4.4.3.3) because a definite rule does not exist to determine their values. The values are therefore, initially set based upon intuition and the gear designer's experience then adjusted according to their performance and the success of the search. The initial values for this application were taken as 75 generations for a population of 75 genomes. These are arbitrary values from which the final values evolved. The actual values for the final process were determined from analysis of the results described in the next section.

Mutation

Mutation plays an important role within the application of the GA. The mutation operator is the Gene Mutation approach, described in section 3.1.4. Standard bit mutation is performed at a probability rate of 5 % for genome selection then randomly transformed within the selected genome. This process is performed continuously throughout the search. The second operator, gene mutation, is performed only during the initial period of the search. Gene mutation is performed to introduce a new genome to the population, thus increasing the resolution and area covered by the search until convergence begins. The initial period has been set as until half the number of genomes required for convergence have converged.

$$\therefore \qquad \text{initial period} = \frac{\text{convergence level}}{2}$$

Convergence

The convergence level is set at a high level, 60 %. This is due to the method of determining when the search has arrived at the optimum solution. As mentioned in section 3.1.1 convergence can be relative to either the decoded information within the genome or the structure of the genes. Selection of which type to use is dependant upon the importance of reducing the time taken and if limits have been applied to values. As the time taken is an important factor for the inclusion of this process within the IIS and limits have been imposed on parameters, convergence will be dependant upon the decoded information within the genome.

5.5.4 Test of the Gear Optimisation

The optimisation process is evaluated by performing numerous tests to check the controlling factors of the GA, the repeatability and robustness of the process and the accuracy of the results. The first category of tests relate to the GAs controlling factors. These concentrate upon the improving the speed and repeatability of the GA process. The factors that have the greatest influence on the performance are the population size, mutation operator and the fitness scaling function. The second category compares the results from identical GA configurations for three different design applications. These applications formed three case studies including two helical gears and one spur. The final set of tests evaluate the final gear designs produced to determine if the optimisation process is behaving as expected and that the resultant design has increased in performance as the optimisation has progressed.

The test cases cover a range of applications that the process will be capable of producing solutions. The range covers spur and helical gears and light and high power transmissions. The initial starting designs for each test case have been checked with the BS 436 part 3 program to provide a datum to compare the results.

5.5.4.1 GA Performance

Performance of the GA has been measured against the computational expense and the level of repeatability of results.

Computational Expense

As discussed in section 3.1 the GA process is computationally expensive. Therefore, for the GA to be a viable AI tool in the IIS, the time taken to achieve a resultant design must be reduced as far as possible. Initial tests with GAs experienced computational times of greater than 10 hours using a Pentium 166 MHz. This length of time is unacceptable. Three main factors that influence the rate of population convergence have been identified: the fitness scaling function, size of population and the mutation operator. Using test case 1 from Appendix C as an initial design, the effects of these three factors have been investigated.

As the actual process time taken (measured in seconds or minutes) for the GA to converge upon a solution will vary between computers an alternative scale to measure the computational expense has been used. Computational expense of the GA is measured by the number of generations required before the population converges. This scale is independent on processor capability enabling the performance of the GA to be concentrated upon, ignoring processor performance. Figures 5.34a to 5.34c.illustrate the effects of these factors upon computational expense.

From Figure 5.34a it is clear that using the fitness rating direct from the fitness function is unsuitable for allocating the probability of reproduction, when compared to fitness scaling. During the testing process, the testing of this function was suspended after the population had been increased to 600 genomes. This was due to the poor trend that can be seen to immerge indicating that any further increase in the population size will result in unacceptable computational expense before convergence occurs.

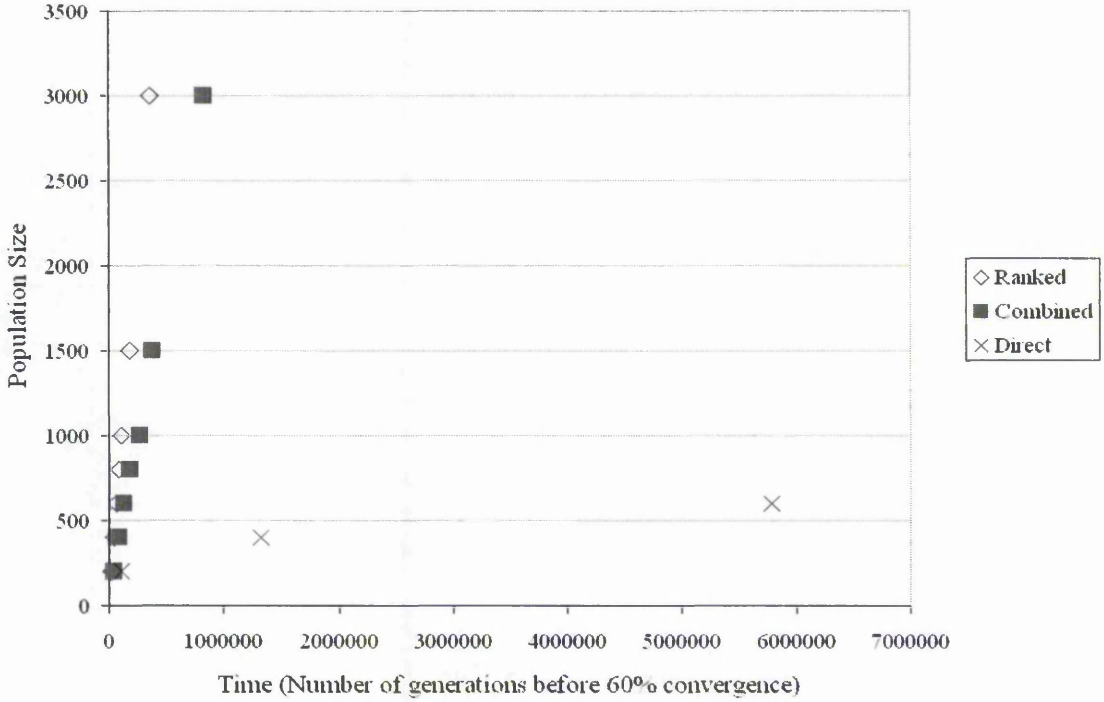


Figure 5.34a Relation of Population Size to Computational Expense using the Bit Mutation Operator.

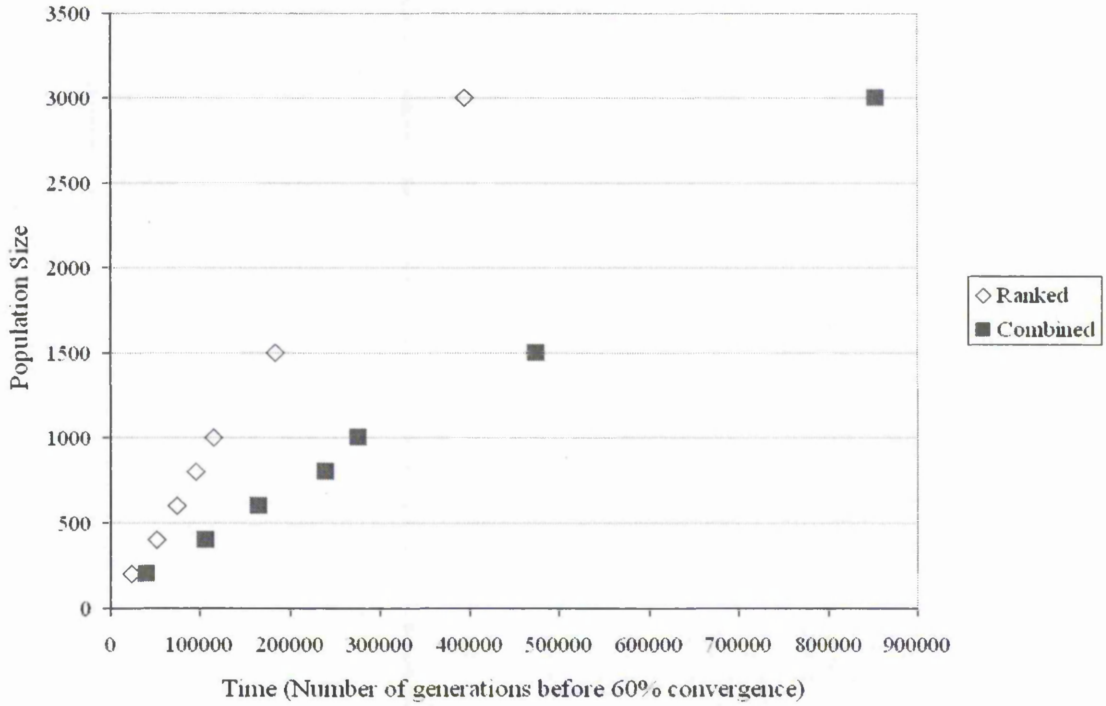


Figure 5.34b Relation of Population Size to Computational Expense using Gene Mutation Operator.

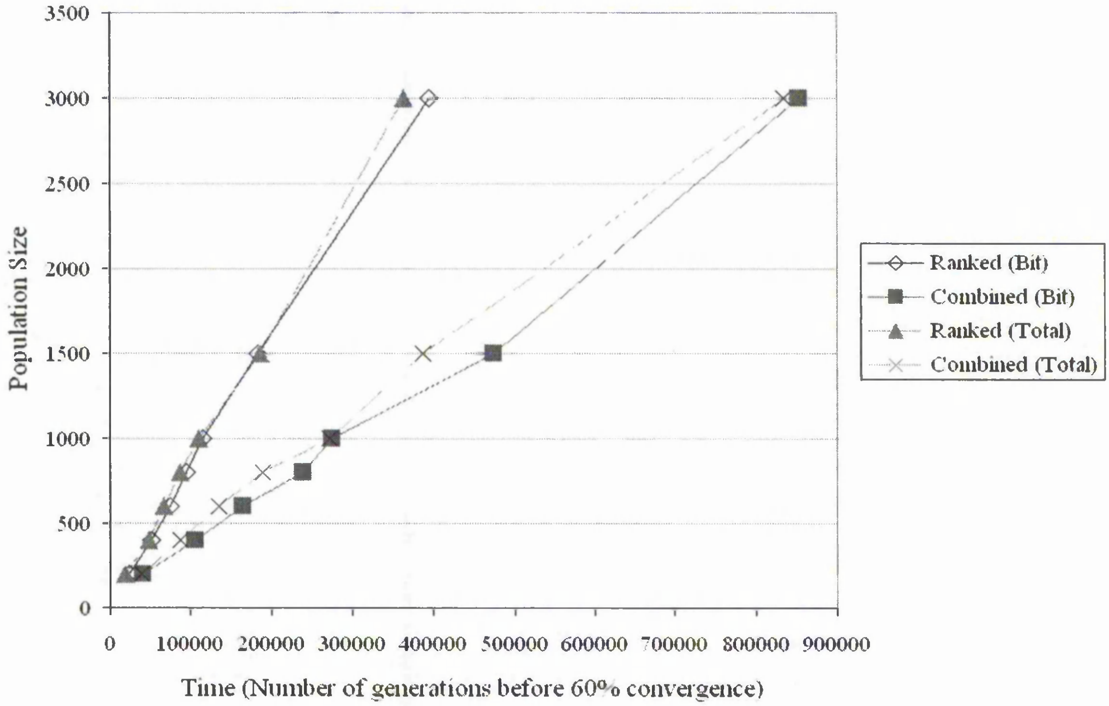


Figure 5.34c Comparison of Mutation Operators and Their Effect on Computational Expense

The results from the tests, presented in Figures 5.34a to 5.34c, suggest that the Ranked fitness function, (described in section 3.1.5.2) achieves convergence of the generation up to approximately 50% quicker that the combined function. Additionally it can be seen in Figure 5.34c that the mutation operators have only a slight influence on the rate of convergence, although the combined function did give the better performance of the two operators.

Repeatability

Repeatability of results is an essential quality that the GA process must posses if it is to be integrated into the IIS. This is due to the iterative process that the detail design module performs, redesigning until no modification to the components occurs. Initial tests of the GAs repeatability indicated that dramatic improvements were required. Hence, the GA factors that influence the search, fitness scaling function, mutation and particularly, population size have been tested to determine their effect on repeatability.

Repeatability of the results also gives an indication of the success that the search is having at achieving a global optimum and if the process working correctly. If the same results are obtained repeatedly for a number of identical searches it is evident that the optimum has been achieved. Thus, the results given in Figures 5.34a to 5.34c are the average of 5 identical tests, the results from which are given in Appendix C.

Using the test case 1 the GA factors have been plotted against the level of repeatability of results attained. Figures 5.35a and 3.35b illustrate trends in the level of performance.

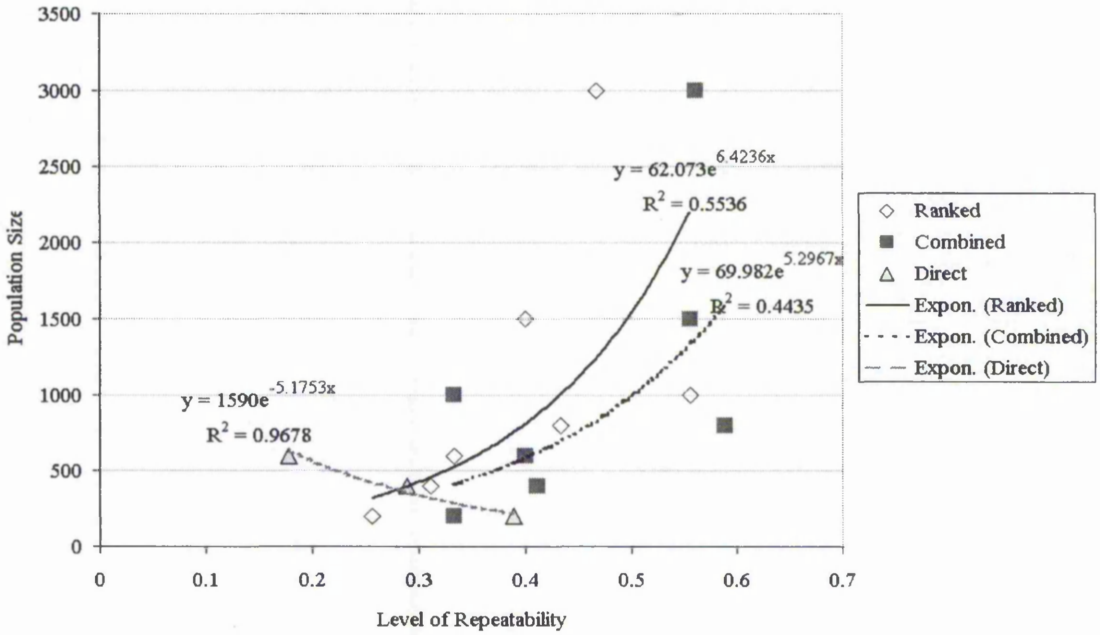


Figure 5.35a Relation between Population Size and Average Level of Repeatability using the Bit Mutation Operator

As can be seen from Figures 5.35a and 5.35b the population size has a dramatic effect on the levels of repeatability that can be achieved. With the exception of the direct fitness scaling function, the levels clearly increase as the population increases. This is as expected, due to a more comprehensive search being performed. An additional observation is that the combined fitness scaling function tends to attain higher levels of repeatability when used with the gene mutation operator.

The lines of best fit that have been superimposed on Figures 5.35a and b emphasise the relationship between the population size and the level of repeatability attainable. Exponential models were used to illustrate the trend as these gave the best fit. Although the R^2 values are low, indicating a poor representation of the data, the tend lines reaffirm that to attain high levels of repeatability the population size must be increased. However, this is at the cost of greater computational expense.

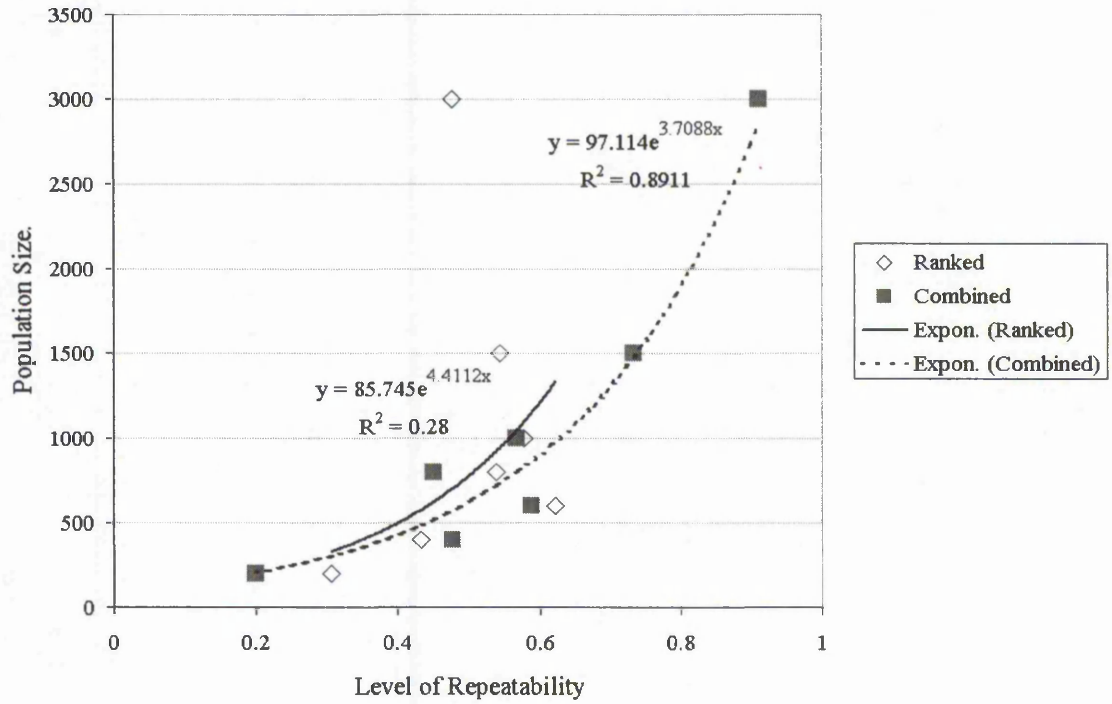


Figure 5.35b Relation between Population Size and Average Level of Repeatability using the Gene Mutation Operator

The performance results suggest that a combination of a population of approximately 1000 genomes using the gene mutation operator with the ranked fitness function will achieve the best results with respect to time. However, with regard to the level of repeatability this combination will only give a level of 0.55 (which corresponds to 55%). From Figure 5.35b it can be seen that much higher levels of repeatability can be achieved using a larger population and the combined fitness scaling function, but the time will be quadrupled. As repeatability of the results is more important than computational expense a trade-off is made. A population size of 2000, the combined fitness scaling function and the gene mutation operator will give a repeatability level of 0.8 (80%), which is acceptable.

5.5.4.2 Optimisation Process

To ensure that the optimisation process is conducting its search correctly, achieving the goals defined, parameter traces have been generated to trace the GA process. The parameter traces record the results of the fittest or most common genome in each generation of the search. The general trend of each of the traces will illustrate if the GA is achieving its goals.

During the evaluation of the GA optimisation the importance of each fitness function has been set by considering the effect that the function will have on the performance of the resultant design. To enable the importance of each fitness function to be compared against the others, the importance is set relative to the size of the population. For example if the fitness function is considered very important the output of the fitness function is scaled between 0 and the number of genomes in the population. Similarly if the fitness function is not considered important or is having too great an effect on the search the output of the fitness function is scaled between 0 and a proportion of the number of genomes in the population. For the purpose of testing the GA process, the range of the fitness functions for equal stress, centre distance, and contact ratio have been set to a maximum output of the value of the population, while the facewidth is set the half. This is due to the facewidth fitness function relating solely to a single parameter.

Parameter traces have been produced for 3 test cases. Figures 5.36a to 5.36d illustrate the traces for case 1, while traces for cases 2 and 3 can be found in Appendix C.

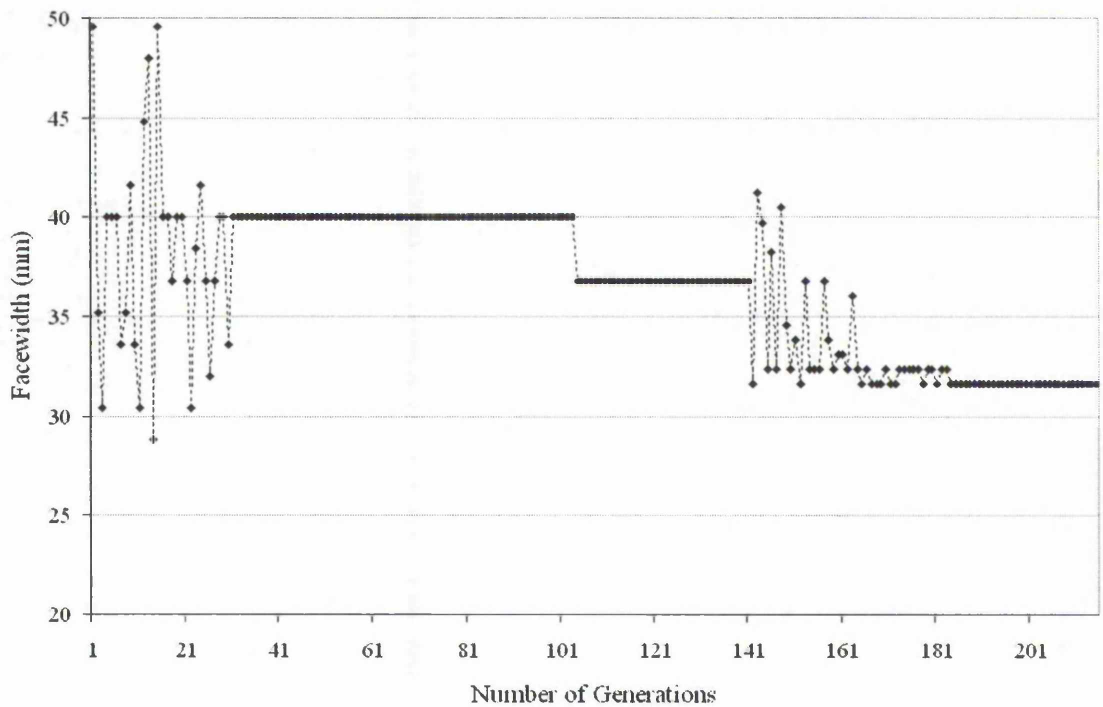


Figure 5.36a Trace of Facewidth (Test Case 1)

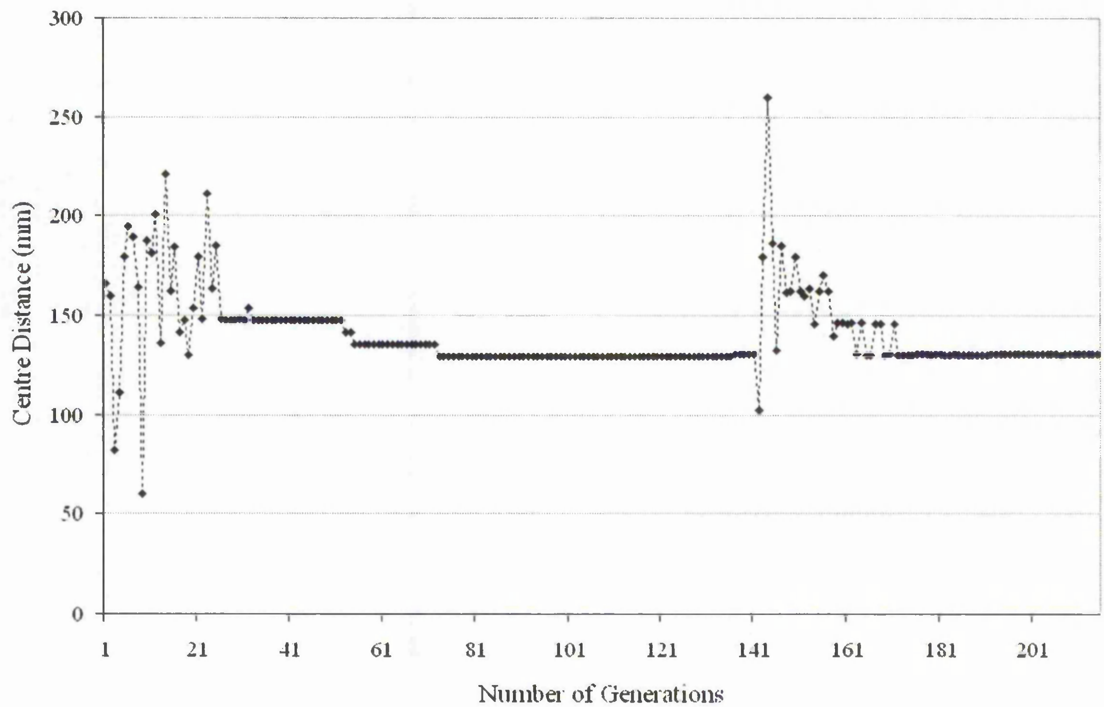


Figure 5.36b Trace of Centre Distance (Test Case 1)

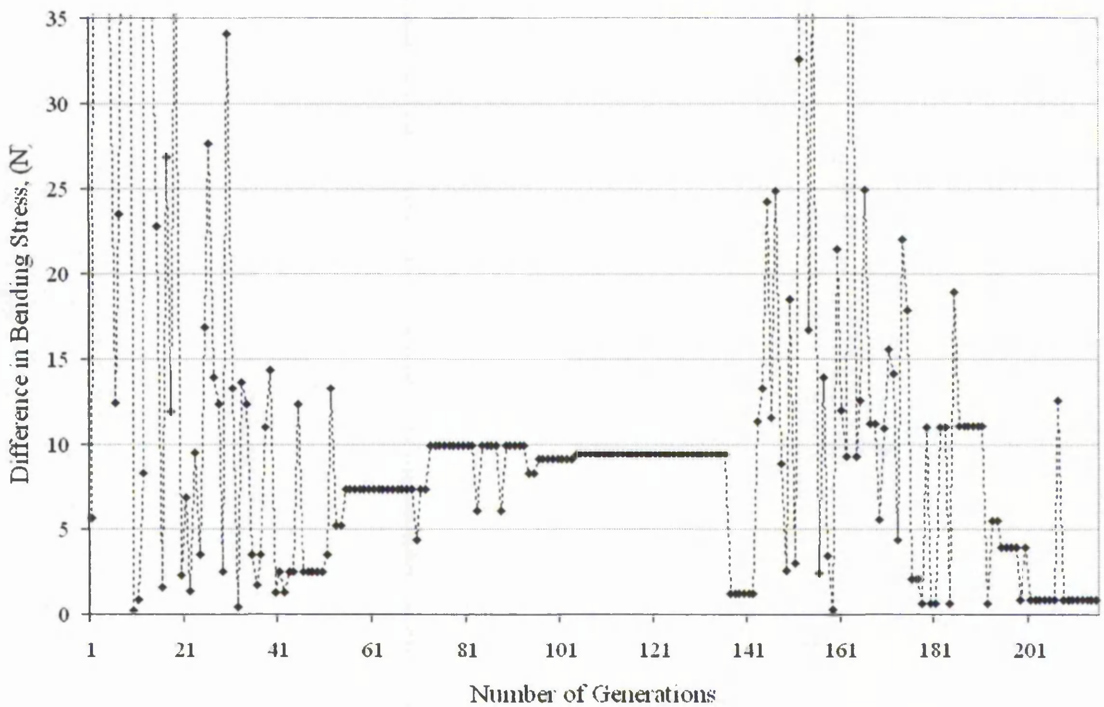


Figure 5.36c Trace of Difference in Bending Stress (Test Case 1)

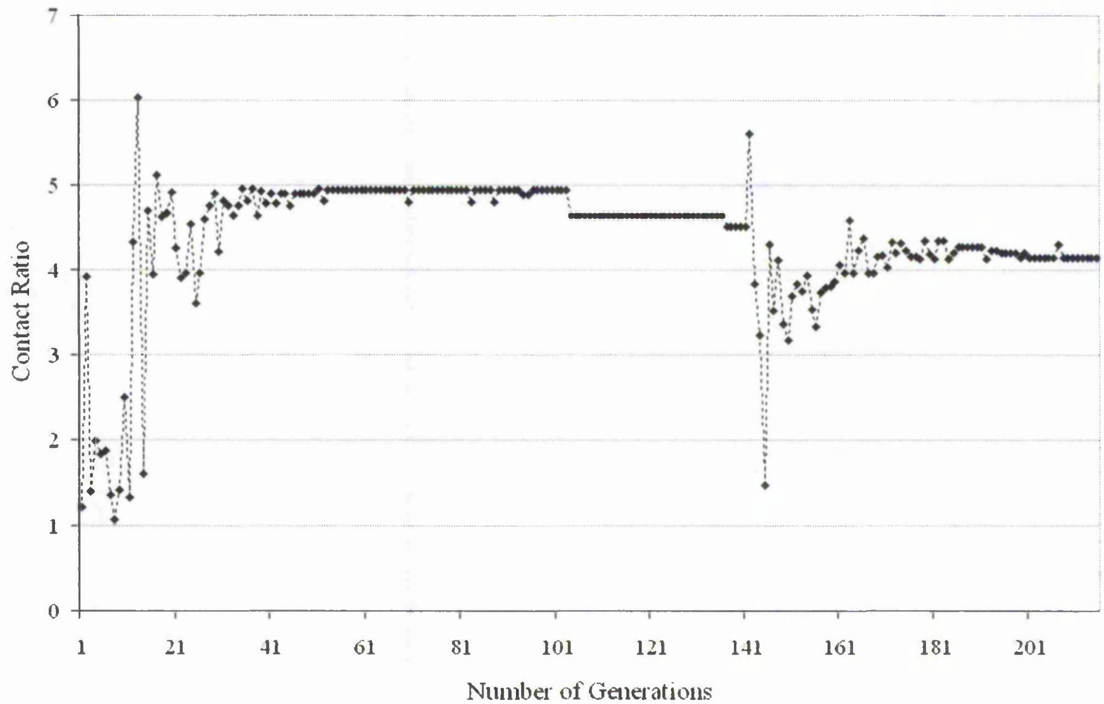


Figure 5.36d Trace of Contact Ratio (Test Case 1)

Each of the traces suggest that the optimisation process is performing correctly. In each figure the two tiers of the search are distinctly clear, starting with oscillations about a general trend toward the desired goal until the population begins to converge on its solution. In Figures 5.36a and 5.36c the second tier of the search has lead to additional decreases in facewidth and difference in bending stress.

The parameter traces for the two additional test cases have displayed similar trends, indicating that the process is working correctly and capable of achieving solutions for a broad range of gear applications.

5.5.4.3 Robustness and Versatility of GA

To ensure that the GA process for the optimisation of gears is versatile The performance of the resultant designs are evaluated for all the case studies. The case studies cover a broad range of applications, hence, testing the robustness of the process to achieve a solution under varied conditions. In addition the quality of the solutions (measured in terms of increase in gear performance) to the test cases will establish if the process has been generalised or tailored to suit one application. The results to the 3 test cases as presented in Tables 5.14 to 5.16.

Geometry	Initial	Test Case 1	Improvement
Module, (mm)	2.00	2.00	-
Alpha, (Deg.)	20.00	20.00	-
Beta, (Deg)	15.00	35.00	-
Pinion Teeth	24	21	-
Facewidth	40.00	31.65	20.88%
Pinion Addendum Coefficient	1.000	1.200	0.425
Wheel Addendum Coefficient	1.000	1.200	0.425
Pinion Rack Tip Radius, (mm)	1.000	1.518	-
Wheel Rack Tip Radius, (mm)	1.000	1.518	-
x1	0.000	0.400	-
x2	0.000	0.400	-
Center Distance (mm)	120.0	106.6	11.17%

Table 5.14a Test Case 1. Gear Geometry

Performance	Initial	Test Case 1	Improvement.
Contact Stress Pinion (% below permissible)	-15.36	1.99	-
Contact Stress Wheel (% below permissible)	-8.44	7.87	-
Bending Stress Pinion (% below permissible)	27.22	68.87	-
Bending Stress Wheel (% below permissible)	29.98	69.14	-
Equal Stresses (% difference)	2.84	0.41	2.43 %
Contact Ratio	3.29	4.15	0.86

Table 5.14b Test Case 1. Gear Performance

Geometry	Initial	Test Case 1	Improvement
module, (mm)	2.00	1.75	-
Alpha, (Deg.)	20.00	20.00	-
Beta, (Deg)	0.00	0.00	-
Pinion Teeth	22	19	-
Facewidth	120.00	74.30	38.1%
Pinion Addendum Coefficient	1.000	1.225	0.225
Wheel Addendum Coefficient	1.000	1.225	0.225
Pinion Rack Tip Radius, (mm)	1.56	1.162	-
Wheel Rack Tip Radius, (mm)	1.56	1.162	-
x1	0.000	0.150	-
x2	0.000	-0.501	-
Center Distance (mm)	132.00	130.1	1.44%

Table 5.15a Test Case 2. Gear Geometry

Performance	Initial	Test Case 1	Improvement.
Contact Stress Pinion (% below permissible)	36.13	0.19	-
Contact Stress Wheel (% below permissible)	51.35	23.97	-
Bending Stress Pinion (% below permissible)	62.94	5.52	-
Bending Stress Wheel (% below permissible)	77.91	38.21	-
Equal Stresses (% difference)	8.84	1.69	7.15 %
Contact Ratio	0.91	2.03	1.12

Table 5.15b Test Case 2. Gear Performance

Geometry	Initial	Test Case 1	Improvement
module, (mm)	8.00	8.00	-
Alpha, (Deg.)	20.00	22.50	-
Beta, (Deg)	18.00	35.00	-
Pinion Teeth	19	17	-
Facewidth	125.00	94.60	24.32%
Pinion Addendum Coefficient	1.150	1.250	0.425
Wheel Addendum Coefficient	1.150	1.250	0.425
Pinion Rack Tip Radius, (mm)	4.00	3.239	-
Wheel Rack Tip Radius, (mm)	4.00	3.239	-
x1	0.000	0.250	-
x2	0.000	0.200	-
Center Distance (mm)	304.0	275.6	5.25%

Table 5.16a Test Case 3. Gear Geometry

Performance	Initial	Test Case 1	Improvement.
Contact Stress Pinion (% below permissible)	-11.28	1.08	-
Contact Stress Wheel (% below permissible)	-4.89	6.76	-
Bending Stress Pinion (% below permissible)	64.89	49.01	-
Bending Stress Wheel (% below permissible)	69.82	49.37	-
Equal Stresses (% difference)	12.84	0.30	12.54 %
Contact Ratio	1.80	3.47	1.67

Table 5.16b Test Case 3. Gear Performance

As can be seen from Tables 5.14b, 5.15b and 5.16b all the optimised gear designs created comply with BS436 part3 for contact and bending stress. This fact is particularly important considering that the initial starting designs for test cases 1 and 3 began with design failures. In addition the bending stresses within the pinion and wheel teeth have almost been equalised, thus encouraging equal wear on both teeth.

For all the test cases, the greatest increase in performance has been for the contact ratio. The increase in contact ration has been achieved in several ways for all the test cases. Increasing the contact ration for a helical gear is relatively easy to achieve by the increase of the helix angle. However, additional modifications to the addendum coefficients (length of teeth), profile shift and number of teeth have been performed to extend the contact ration beyond the level of standard designs. This fact is illustrated with test case 2, the pinion gear. As there is no helix angle increases in contact ration are achieved by the modifications previously mentioned. Table 5.15b illustrates that the contact ratio has been doubled to 2.03

Reduction of the size of the gear pair through the facewidth and the centre distance has been successful, achieving excellent result in achieving their goals. Facewidth has been reduced by up to 38.1% while the centre distance has improved by up to 11.17%, thus the process encouraged smaller gears.

From the results, in Tables 5.14a to 5.16b, it has been shown that the optimisation process achieved the goals set, producing high quality gear designs.

5.5.5 Gear Optimisation Module

The gear optimisation process is developed into a design module for the IIS. The initial design required as the starting point for the optimisation process comprises of the existing gear design within the IIS database. This design is either a previously optimised design requiring redesign or the design generated by the initial gear design module, section 5.4. Together with the initial design the application of the gear is transferred. Information relating to the application forms two rolls. The first is to supply information required to calculate the tooth stresses. The second is to set up limits to the critical conditions and goals for fitness functions. The values of these limits and goals refer to the assembly and performance requirements of the transmission design.

The GA optimisation process forms the basis for the structure of the design module, as illustrated by the shaded area in Figure 5.37 below.

The process commences with the set-up of the population size, convergence parameters, fitness function goals and critical condition limits and the initial design. The population is then filled with random genomes, forming the initial population. The genes within the genome are decoded, as described in section 5.5.3.1, and the gear geometry adjusted to comply with the new parameters. The geometry is checked to ensure that no limits have been exceeded. If a limit is exceeded a penalty is attached to the genome which will be imposed on its fitness rating.

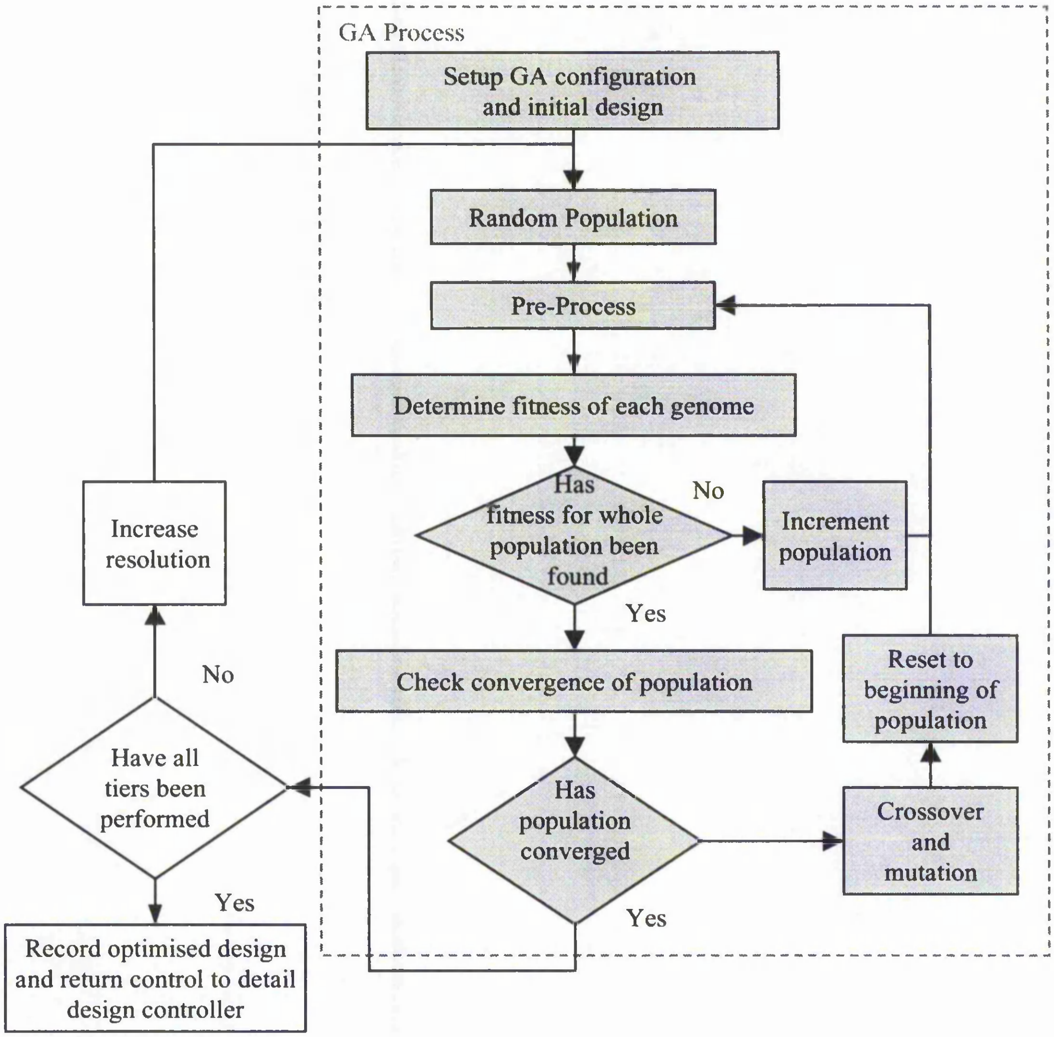


Figure 5.37 Schematic of Gear Optimisation Module

The performance of the genome is determined, with respect to stress, contact ratio and speed ratio. These values are assigned to the genome for use during the calculation of the fitness rating, once the performance of all the genomes in the population has been determined. Details of the performance characteristics that are considered are given in section 5.5.3.2.

Once the performance of all the genomes in the population has been determined, the fitness rating for each genome relative to the population is calculated. Penalties that have been awarded to genomes during the pre-process stage or resulting from the performance are imposed on the resultant fitness rating. The fitness rating is then scaled using the combined scaling function, described in section 3.1.5.2, to increase the ability of the GA to converge upon the optimum.

The population is checked to determine if it has converged upon a solution. If convergence has not occurred crossover and mutation of the population is performed to create the new generation. The process is repeated until the population converges on the same solution.

At this point the basic GA process has finished, however, this module utilises the cascade procedure, as described in section 3.1.2. The resolution of the decoding process during per-process is increased and the optimisation process repeated. However, the initial gear design that formed the base for the search is replaced by the solution obtained from the first tier.

Upon completion of the second tier of the optimisation the resultant solution is taken as the optimised gear design. The solution is compared to establish if the gear design has significantly altered from the initial design used by the module. If parameters of the design that can alter the assembly of the gear within the transmission design, such as diameter and width, have altered redesign of associated components will be necessary.

The resultant gear design is transferred to the component database and control ad an indication if redesign is necessary is returned to the detail design controller.

5.5.6 Discussion of GA for Gear Optimisation

Presently there are several methods for the optimisation of gears including the use of GAs (Abersek et al., 1996). Abersek's application optimises some of the parameters that this application addresses, as do the other gear optimisation techniques. However, they do not consider modification of the tooth's profile and encourage the increase of the contact ratio for reduced vibration and noise as this application does.

In addition to parameter adjustment, when applicable, the selection of the type and magnitude of profile shift is determined. According to Niemann (1978), the magnitudes of the profile shift should be biased towards the pinion wheel being positive in order to avoid undercutting. From the results in Tables 5.14a, 5.15a and 5.16a it is shown that the GA has obeyed this rule, indicating that the fitness criteria do correctly define the desired result. This may be due to the limiting factors of permissible and actual stress, which form one of the main fitness criteria. The GA optimisation therefore, allows designs to be improved using the desired result as the target. Thus intricate knowledge about the effects of individual parameters on either the final design or each other is not required.

The traditional GA problems of computational expense and repeatability have been addressed in this section, with acceptable results achieved. The combination of the Gene mutation operator and the combined fitness scaling function allows a population of sufficient size to be used that comprehensively covers the search space without excessive computational expense, thus increasing the repeatability of results. As the repeatability increases, confidence in the results and the ability of the GA to achieve an optimum also increases. This is important as the GA is considered by industry as a 'black box' technology, in a similar way that ANNs are considered. However, the results of the test cases show that the GA process can be reliable and produce high quality gear designs.

A byproduct of the GA research has been the development of a separate design package solely for the optimisation of performance gears. The package, called OPTGEAR, combines the GA gear optimisation module developed for the IIS with a GUI. The GUI performs the pre and post-process operations required by the gear module which are performed by the detail design controller within the IIS. Using the GUI the user provides the initial design to be optimised. This design does not need to be viable with respect to stress (as shown by case studies 1 and 3) as the design only forms a starting point for GA process, the parameters of which will be modified to ensure that the design does not fail due to stress. The initial gear is then optimised according to the design goals, which the user can adjust to suit each application. Adjustment is achieved by graphically altering the importance assigned to each of the fitness functions.

As the package is aimed at industry (currently several companies are have shown considerable interest) repeatability of results is essential. As illustrated in Figure 5.34a, b and c repeatability is not guaranteed, therefore, to overcome this problem the GA process is repeated a number of times. Once complete the results are compared and the most frequent solution taken as the final design. The process of repeating the gears optimisation has provided a high rate on repeatability.

The package has been developed for two reasons. Firstly, so that the benefits to gear design provided by the GA approach to design improvement are available to industry without the necessity for the Total Design system. Secondly, the quality of the results achieved will help increase industrial awareness to GAs for design and illustrate their possibilities. An example of the OPTGEAR package and instructions are given in Appendix E.

5.6 Design Selection and Retrieval

Within the detailed design stage of the IIS for the design of mechanical power transmission systems, design retrieval and selection has been applied to the selection of two different components: gears and bearings. These applications will demonstrate the procedure for the application and development of this technique, described in section 4.6. The two components illustrate the approach toward both the selection and retrieval of components and are applied at different stages of the detailed design.

5.6.1 Gear Design Retrieval

The approach is applied to the identification and selection of gear variations from the basic form. The selection is based upon the design specification, physical properties and intended application of the gear. Selection of the design is performed after the properties of the gear have been determined. The appropriate gear for the application is selected from a choice of designs and retrieved prior to the generation of the machine and assembly drawings.

5.6.1.1 Gear Selection Criteria and Designs

The main considerations for the selection of the appropriate gear type relate to the performance requirements of the component, the application of the transmission system and the design specifications that directly relate to the component. The last consideration may appear obvious, but for the purposes of identifying the input parameters for an ANN all parameters considered must be clearly defined. Explicit identification of all relative parameters is crucial, as non- inclusion within the input pattern will result in no relation being formed within the network. Therefore, all parameters that will affect the selection process are listed and their significance determined. The table below contains all the parameters that could affect the selection process together with a value corresponding to the influence that they exert over the selection.

The influence ratings from the three experts[1] are averaged to determine their influence on the design selection and therefore, if they should be presented to the ANN. The influence of the parameter upon the design selection is banded into one of four categories: directly related, important, considered, not considered. Tale 5.18 below illustrates the bands.

Parameter	Type	Influence Rating (0 – 5)			Mean
		Expert 1	Expert 2	Expert 3	
Power	Specification	4	5	5	4.67
Application (Application Factor)	Specification	3	2	2	2.33
Type (Pinion or Wheel)	Application	5	5	5	5.00
Diameter	Determined	5	5	4	4.67
Ease of Assembly and Maintenance	Specification	2	1	2	1.67
Ease of Manufacture	Specification	4	3	3	3.33
Cost	Specification	4	4	4	4.00
Weight	Specification	3	3	2	2.67
Load Distribution	Specification	1	2	1	1.33
Facewidth	Determined	4	5	4	4.33
Size	Specification	4	5	4	4.33
Speed	Specification	2	4	5	3.67

Table 5.17 Considered Parameters for the Selection of Gears and Their Significance

Influence Category	Directly Related	Important	Considered	Not Considered
Rating	5 – 3	3 – 1	1 – 0	0

Table 5.18 Influence Category

As mentioned in section 4.5, the network is capable of establishing if an input has no relation or influence to the output pattern. Therefore, even if a parameter has been given a rating of less than one, it will be included in the input pattern to the network. However, parameters with a rating of 0 will not be included, thus simplifying the input pattern.

The value of the parameter determines the screen position and size on the GUI for design selection and acquisition of the training data. The position of the parameter is important as the initial users of the IIS will base their selections on the information presented to them. Therefore, the more prominent parameters will draw the most attention and consideration. The prominent positions vary for each display, but generally constitute

[1] Engineers within the Department of mechanical and Manufacturing Engineering, Nottingham Trent University

the top of lists, centre of screen or on their own. To prevent overcrowding of the screen, parameters that fell into the bottom of the important group that could not fit on the initial screen and parameters within the 'consideration group' will be displayed on an additional screen that the user must select. Figure 5.38 illustrates the selection criteria and GUIs for the selection of spur gears.

Figure 5.38 User Interface for Gear Selection and Collection of Training Data (Significant Parameters)

5.6.2 Bearing Type Selection

The bearing selection is performed at the beginning of the detail design stage. The selection process is performed at this point as the type of bearing to be extracted from the bearing catalogue will cause dramatic effects on the transmission design throughout the iterative process. The selection process determines the type of bearing to be used for the individual transmission stages of the design. The selection procedure is conducted in the same manner as for the gear selection above, requiring the initial users of design system to select the appropriate bearing from the selection presented to them via the GUI. The GUI used is illustrated in Figure 5.38 and is of a similar format as that for the gear selection.

5.6.2.1 Bearing Selection Criteria and Designs

The criteria that the selection of the bearing type is based upon relate heavily to the design specification and the position within the assembly. The identification of the parameters that influence the selection procedure have been determined by experts within the field of bearing application[2] and general user's[3]. The parameters have been collated and can be found in Table 5.19 below. These parameters form the information and specification that will be presented to the user and direct the ANN.

Parameter	Type	Influence Rating (0 – 5)			Mean
		Expert 1	Expert 2	Expert 3	
Power	Specification	4	5	5	**4.7**
Speed	Determined	3.7	5	4	**4.2**
Mounting Position on Shaft	Determined	1	0	1	**0.7**
Bore Diameter	Determined	0	0	0	**0**
Axial Loads	Determined	2.9	5	5	**4.3**
Ease of Manufacture	Specification	0	0	0	**0**
Cost	Specification	2.9	5	4	**4**
Ease of Assembly and Maintenance	Specification	3.7	3.4	4	**3.7**
Application (Heavy, light, etc)	Specification	3.6	NA	4	**2.9**
Width	Determined	3.7	NA	4	**2.6**
Life	Specification	3.9	5	4.7	**4.5**
Weight	Specification	4.3	3	4.2	**3.8**
Lubrication	Determined	NA	3.9	2	**2**
Relative Size (capacity v room)	Determined	4	3.25	3.9	**3.7**
Noise	Specification	3.6	NA	3	**2.2**

Table 5.19 Considered Parameters for the Selection of Bearings and Their Significance

The parameters that are presented to the used via the GUI are a combination of direct values and elements of the product design specification. The parameters are arranged in the GUI as for the gear, significant values and prominent positions, top of list and centre of the screen, while parameters of minor consideration are grouped together to make room for the others. The presentation of the information is illustrated in Figure 5.39.

[2] Current and past Application Engineers for SKF Ltd.
[3] Engineers within the Department of Mechanical and Manufacturing Engineering, Nottingham Trent University

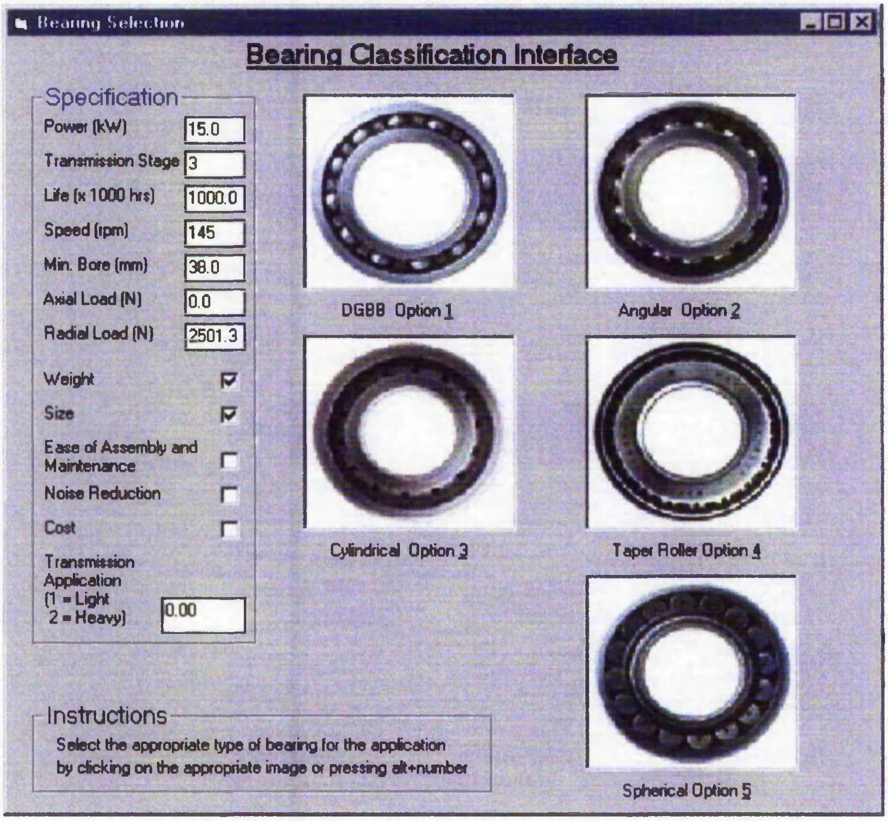


Figure 5.39 User Interface for the Collection of Training Data and Bearing Selection.

Due to the point of the design process that the selection of the bearings occurs, (initial design stage), information relating to the dimensions of the application is limited to initial values. The performance characteristics and general specification are available during the initialisation of the detail design stage and therefore, included. However, external forces and dimensions are rough values, but any deviation from the final value will not be sufficient to alter the class of bearing.

5.6.2.2 Training Data Generation for the Selection of Bearings

The training information for the bearing selection is generated in two stages. Firstly, selection data extracted from the SKF bearing catalogue (1989). This will form the initial base of the training data. The second method of generating the training data is through the use of the GUI as for the gear in section 5.6.1. The GUI method allows parameters, specifications and criteria that has been identified by experts, (collated in Table 5.19) and are specific to the mechanical transmission design system that have been developed to contribute to the selection. Thus allowing the knowledge within the system to be formed to suit the individual application.

The networks are trained by the method described in section 4.5

5.6.3 Test of Design Retrieval Approach

The test data that the design retrieval network is initially tested with comprised of estimated values that the IIS for mechanical power transmissions could produce. The test data comprised both analogue and discrete data to test the ability of the network to handle patterns comprising combinations of these data types.

The input data was generated based upon ranges of random values and aimed at the selection of gear designs. Table 5.20 lists the input restrictions, while the options for the selection of gears, illustrated in Figures 5.39, correspond to selection categories.

Input Parameter	Minimum Value	Maximum Value	Increment
Power, Kw	0	100	Continuos
Transmission Stage	1	3	Continuos
Centre Distance, mm	0	1000	Continuos
Ease of Maintenance and Assembly	0	1	Discrete
Ease of Manufacture	0	1	Discrete
Cost	0	1	Discrete
Weight	0	1	Discrete
Load Distribution	0	1	Discrete
Size	0	1	Discrete

Table 5.20 Input Pattern

The output pattern was created by allocating points, for certain values and qualities of the input specification, to the design category that deserved them. The allocation and the reasons were constructed into a series of rules, which are represented in Table 5.21 below. The rules are not exact in their knowledge, but only form a method of testing the ANN approach for design retrieval. The selected design corresponds to the design with the highest value. Where several are equally well suited the lowest category will win.

Within the training data the effect of introducing redundant inputs has been reproduced. These inputs, as mentioned in section 4.5.1.3, represent the scenario of excessive input data form the IIS, which the designer is not basing the design selection upon.

Condition	Design 1	Design 2	Design 3	Design 4	Design 5
Power >= 75	1	1	0	0	0
50 < Power < 75	1	1	1	0	0
50 < Power < 75	1	1	1	1	0
10 < Power < 50	1	1	1	1	0
Power < 10	1	1	1	1	1
Stage = 1	1	0	0	0	0
Stage > 1	1	1	1	1	1
Centre Distance < 75	1	0	0	0	0
<= 75 Centre Distance < 150	1	1	0	0	0
<= 150 Centre Distance < 500	0	1	1	1	0
Centre Distance > 500	0	0	1	1	1
Ease of Maintenance and Assembly	1	1	1	0	0
Cost	1	1	1	0	0
Weight		1	1	1	1
Load Distribution	-	-	-	-	-
Size	1	0	0	0	0

Table 5.21 Selection Rules

5.6.3.1 Training of the Network

The network was trained with a training set of 50 examples tested with a separate, original test set of 10 examples. Both sets of data were generated with pseudo random values within the ranges of the restrictions in Table 5.20 by a computer program to prevent any additional, external influences upon the design selection. Training of the network was performed using the GEN-NEU network training program. The parameters of the GEN-NEU program were set for a small population size (20), but with a high mutation rate (10%), thus allowing for a wide search for an optimised performance network while reducing the time taken.

5.6.3.2 Network Results

The network was trained with the data described above. The result from the test was that the network was predicting the selection from the test set (simulating the designer when within the system) with reasonable success, 60% of the outputs correct. These results are given in the table below, Table 5.22.

Target Bearing Type	Chosen Bearing Type	Success	Target Bearing Type	Chosen Bearing Type	Success
1	2	✗	1	5	✗
2	2	✔	2	2	✔
1	1	✔	1	1	✗
2	5	✗	3	3	✔
1	1	✔	1	1	✔

Table 5.22 Test Results for the Selection Network

5.6.4 Conclusions on Design Selection Retrieval

The selection process has shown that the procedure can learn the design mannerisms of the user. However, the data that the network was trained with represented a very small area of preference. This will influence the results obtained. The training data is very limited but the network is beginning to generalise the selection process. The process has therefore, demonstrated the desired qualities; perform new selections based on similar or previous cases.

The results are sufficiently successful for this process to be combined within the IIS, however, the networks will only replace the user once sufficient training information has been collected.

CHAPTER 6

VALIDATION OF APPLICATION

The IIS for the design of a mechanical power transmission system is tested to ensure that the system is producing feasible results. Production of feasible results will therefore, indicate the success of the intelligent hybrid approach.

The IIS has been completed for the design of parallel transmission systems, therefore, this type of transmission will form the test case. Throughout the development of the design the results will be evaluated, allowing the performance and decision making process to be observed. The output from the system throughout the designs development is accessed, where possible, with reference to a similar design produced by an existing knowledge based system (KBS), Su 1990.

6.1 Test Case

To evaluate the performance of the system the same test used by Su (1990) has been used. The comparison with Su's results will allow the IIS results and decision making processes to be compared with a knowledge based system, thus validating advantages stated of the hybrid approach.

The test will comprise of the design a transmission system that meets the following conceptual criteria:

Power Transmitted	15 kW
Input Speed	1450 rpm
Speed Reduction Ratio	10 : 1
Orientation of input to output	Parallel
Life of transmission	1000000 hrs

Further design specifications will be identified as the design progresses.

6.2 IIS Design

The analysis of the transmission system test case is performed in two parts, conceptual design and detailed design. This allows the stages of the design process to be illustrated and clarification of the decision-making processes.

6.2.1 Conceptual Design

The basic product design specification (PDS) is supplied to the IIS by the user via the GUI. The PDS comprises the physical requirements, orientation and characteristics of the intended design. The user inputs them where prompted in Figures 6.1, 6.2 and 6.3. These three categories of information, illustrated in Figure 6.1, represent the minimum required to perform the conceptual design.

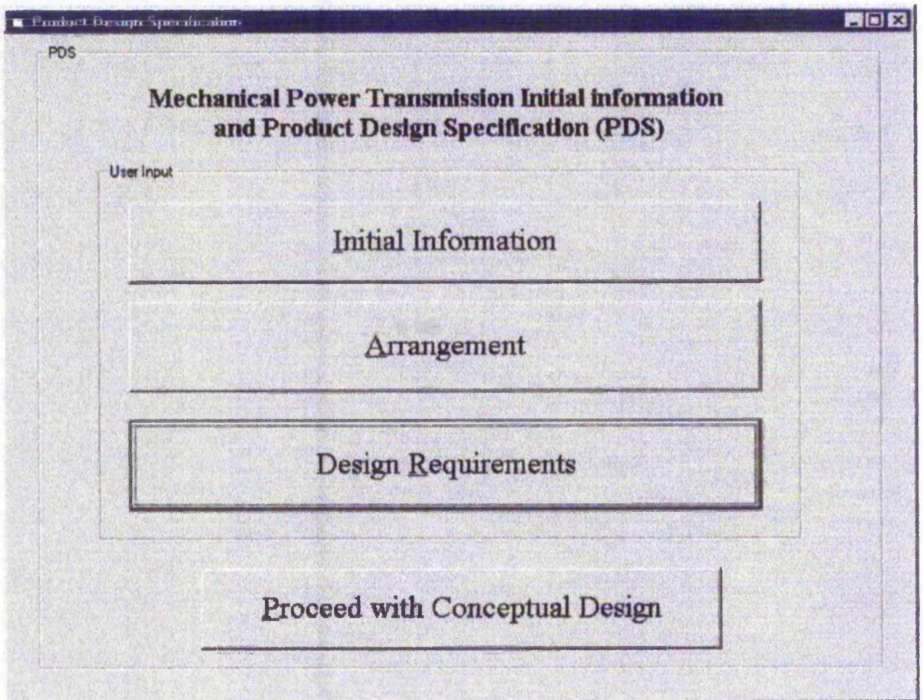


Figure 6.1 PDS Categories GUI

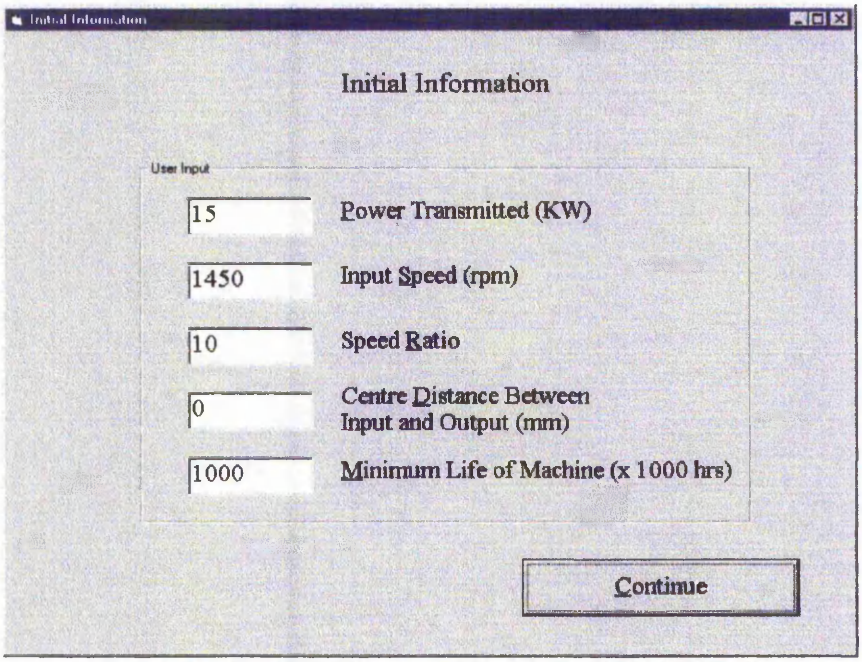


Figure 6.2 Initial Performance Information GUI

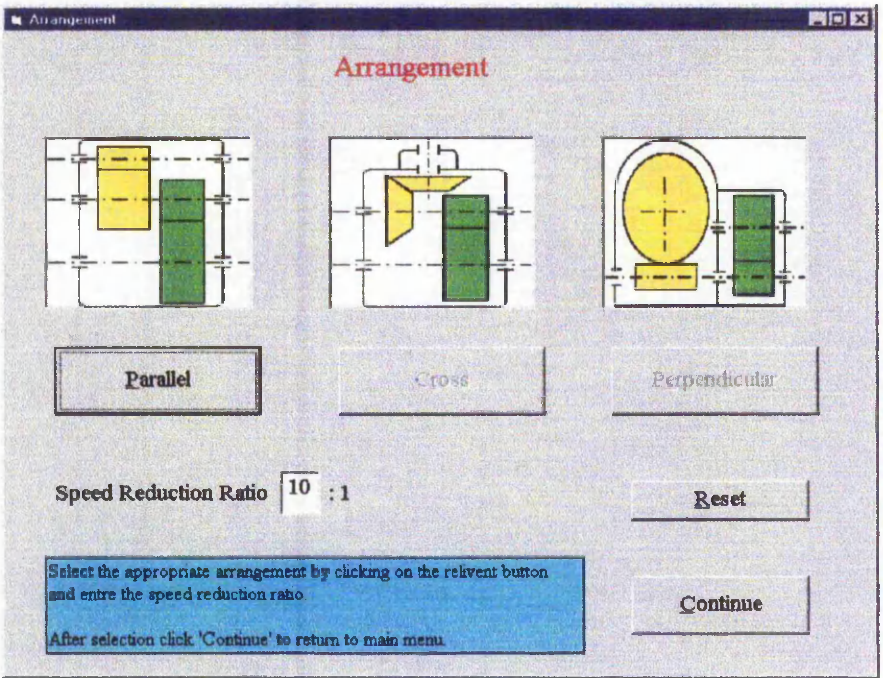


Figure 6.3 Design Orientation GUI

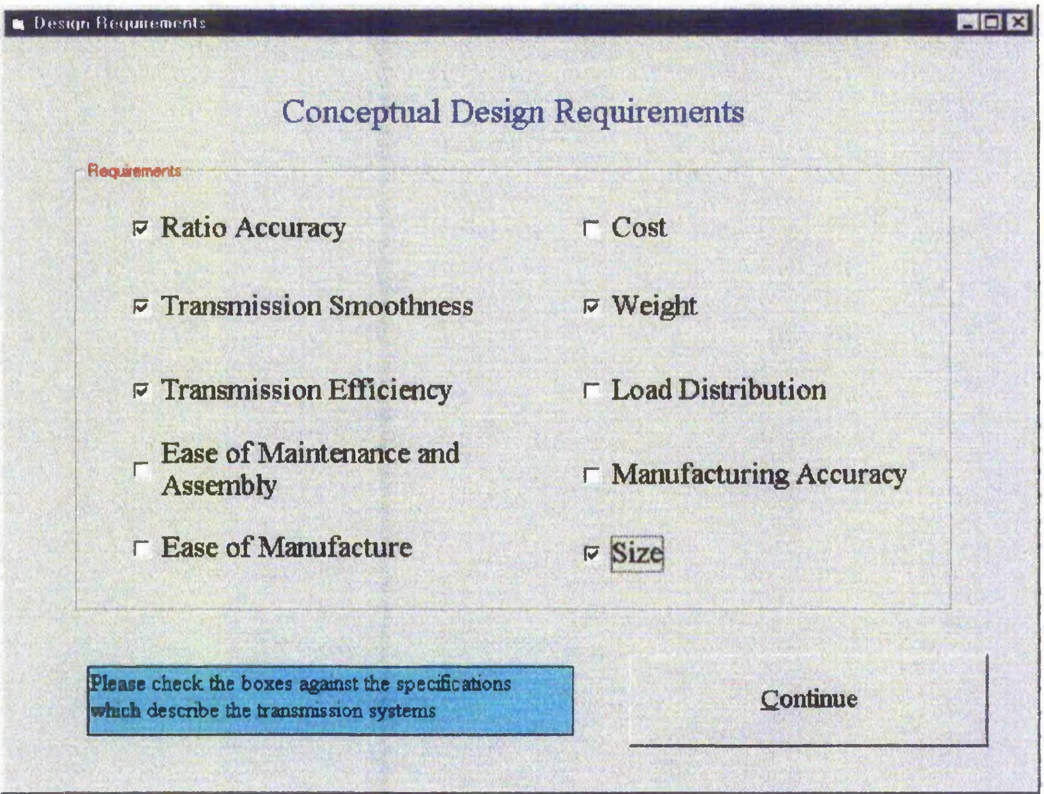


Figure 6.4 Conceptual Design Requirements GUI

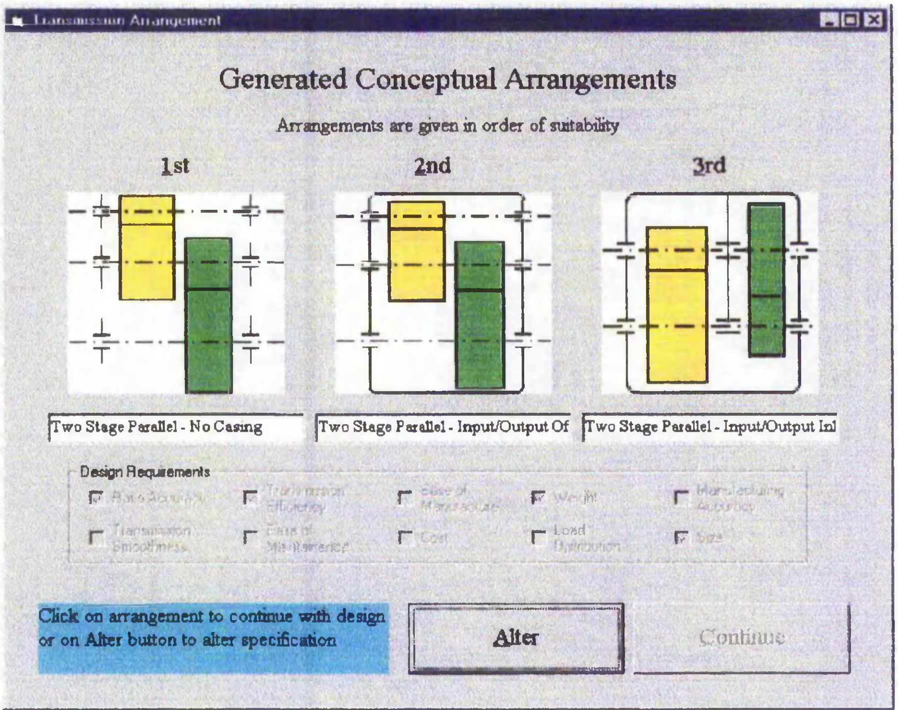


Figure 6.5 Generated Conceptual Arrangements GUI

The requirements indicated in Figure 6.4 relate to:

- Accuracy of the transmission ratio to be as high as possible
- Transmission smoothness and low noise
- High transmission efficiency between input and output
- Design weight to be considered (small)
- Size of the transmission to be considered (small)

All the information necessary to commence the conceptual design has now been supplied by the user, the basic requirements have been specified. This corresponds to the first stage of the design approach, *Requirements*.

Using the ratio and the orientation of the input to the output shafts, the number of transmission stages is determined from the production rule:

IF *ratio* > *7* AND *ratio* $\leq$ *30* AND *orientation is parallel* THEN *2 stages are required*

The number of stages completes the input pattern required by the arrangement network, which gives the conceptual arrangement of the design.

Input pattern to network:

1	*Parallel Orientation*
0	*Cross Orientation*
0	*Perpendicular Orientation*
0	*1 Stage*
1	*2 Stages*
0	*3 Stages*
1	*Ratio Accuracy*
1	*Transmission Smoothness*
1	*Transmission Efficiency*
0	*Ease of Maintenance and Assembly*
0	*Ease of Manufacture*
0	*Cost*
1	*Weight*
0	*Load Distribution*
0	*Manufacturing Accuracy*
1	*Size*

Output pattern from ARRANGEMENT *network*

0.000137211	*1 Stage Parallel*
0.00009915	*1 Stage Cross*
0.00000078	*1 Stage Perpendicular*
0.598549	*2 stage Parallel, input/output offset*
0.547031	*2 stage Parallel, input/output in line*
0.00034845	*2 Stage, Cross then Parallel*
0.00004469	*2 Stage, Perpendicular then Parallel*
0.999574	*2 Stage Parallel, no gearbox casing*
0.00066137	*3 Stage Parallel*
0.00033414	*3 Stage, Cross the Parallel*

From the output pattern the order of suitability is extracted. Emphasis is placed on the top 3 most suitable options, which are presented to the user in the GUI form shown in Figure 6.5. From the output pattern however, it is clear that the 2 Stage Parallel with no gearbox casing is the most suitable. (The no gearbox casing refers to a gear train, which constitutes part of a larger assembly). This option is chosen by the user forming the conceptual arrangement of the design.

Transmission Component Selection

The arrangement is checked to determine the orientation of the first transmission stage. As the orientation is parallel, for which 5 types of component are available, the production rules cannot determine the type of transmission required. Thus the Component ANNs are used as described in section 5.1

The input pattern is prepared and presented to the ANN.

Input pattern to Component Network containing all components

1	*Ratio Accuracy*
1	*Transmission Smoothness*
1	*Transmission Efficiency*
0	*Ease of Maintenance and Assembly*
0	*Ease of Manufacture*
0	*Cost*
1	*Weight*
0	*Load Distribution*
0	*Manufacturing Accuracy*
1	*Size*

Output pattern from Component Network

0.977055	*Double Helical*
0.999910	*Single Helical*
0.936947	*Spur*
0.747440	*Belt Drive*
0.591291	*Chain*

The most suitable component selected by the network is the single helical gear. The rules governing the component hierarchy use this component to select the appropriate network for the next transmission stage. As the component was a single helical gear the double helical gear cannot be used in the design, therefore, the network that excludes this component is used for the second stage. The input pattern is presented to the network.

Input pattern to Component Network containing all components

1	*Ratio Accuracy*
1	*Transmission Smoothness*
1	*Transmission Efficiency*
0	*Ease of Maintenance and Assembly*
0	*Ease of Manufacture*
0	*Cost*
1	*Weight*
0	*Load Distribution*
0	*Manufacturing Accuracy*
1	*Size*

Output pattern from Component Network

0.000000	*Double Helical*
0.999976	*Single Helical*
0.945091	*Spur*
0.769338	*Belt Drive*
0.600129	*Chain*

From the output pattern the single helical gear, again, is the most suitable solution and thus forms the transmission component of the 2nd stage.

As there are no more stages to the transmission design, the component selection is complete. The conceptual design is presented to the user prior to the detail design stage of the product development. The conceptual design comprising the arrangement and transmission components is illustrated in Figure 6.6.

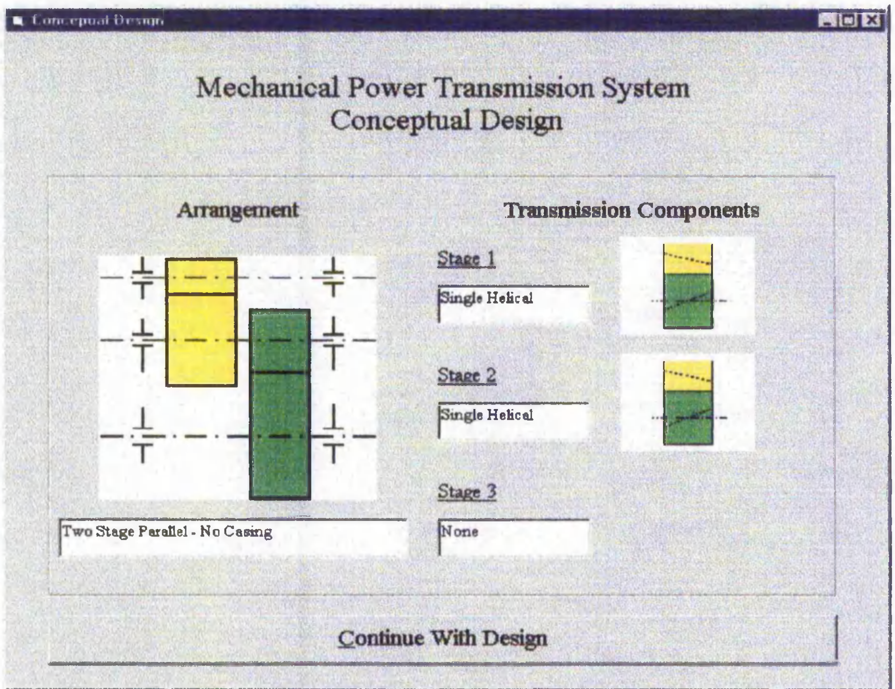


Figure 6.6 Conceptual design GUI

6.2.2 Detail Design Stage (Initial)

Control of the design is passed to the detail design of the IIS together with the conceptual design. The design is converted into an encoded representation of the design using the data extracted from Table 5.5. This gives the following encoded values for the sub-assemblies the final design comprises of.

Shaft 1 Sub-assembly	132832
Shaft 2 Sub-assembly	66544
Shaft 3 Sub-assembly	34256

These sub-assemblies comprise the following components:

Shaft 1 Sub-assembly	1 Input Shaft, 2 bearings, 1 spacer and 1 gear
Shaft 2 Sub-assembly	1 Lay Shaft, 2 bearings, 2 spacers and 2 gears
Shaft 3 Sub-assembly	1 Output Shaft, 2 bearings, 1 spacer and 1 gear

Using the production rules the split of the speed ratio between the stages of the transmission is calculated for 2 stages according to Niemann (1978):

Stage 1, $U_1 = 0.8U^{\frac{2}{3}}$ $= 3.71321{:}1$

Stage2, $U_2 = \dfrac{U}{U_1}$ $=2.69308{:}1$

where: U = Total speed ratio

U_1= 1st stage ratio

U_2= 2nd stage ratio

From the speed ratios the shaft speeds are determined:

Shaft 1 rpm, RPM_1=1450

Shaft 2 rpm, RPM_2=390.5

Shaft 3 rpm, RPM_3=145

Initial gear pair (Stage 1)

To perform the Initial Gear designs additional information is required for each gear pair. Figures 6.7 to 6.15 illustrate the system aided input of the information.

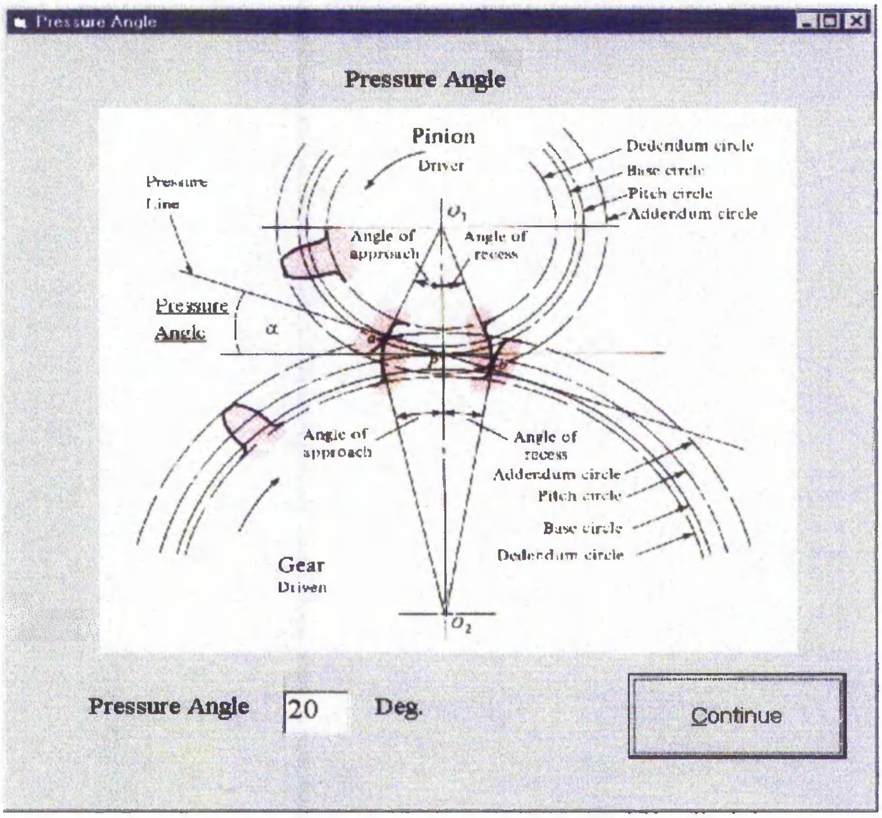


Figure 6.7 Initial Pressure Angle GUI

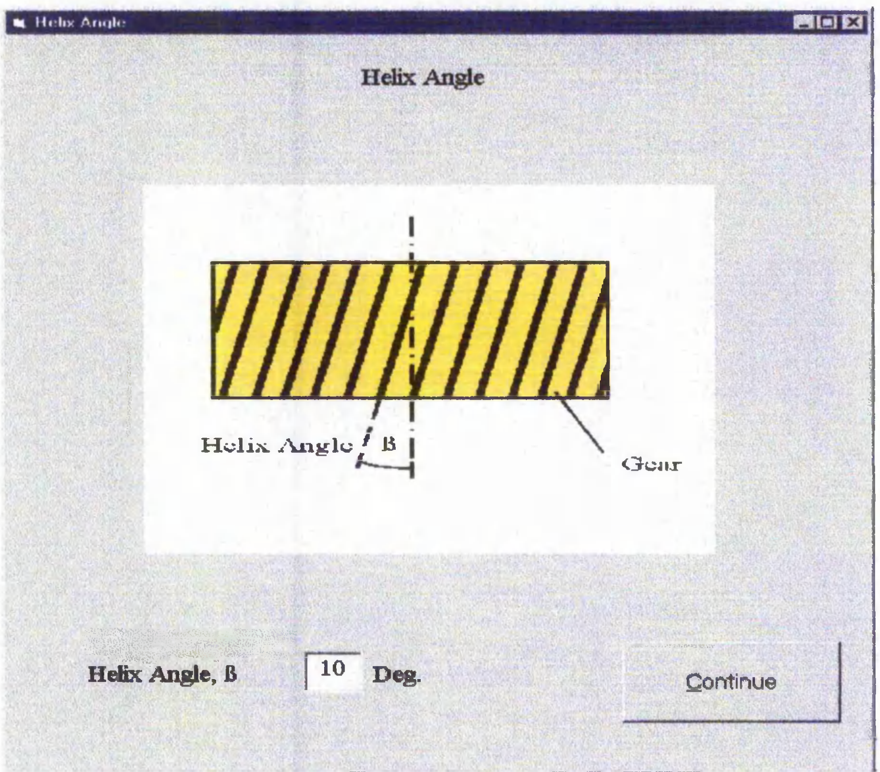


Figure 6.8 Initial Helix Angle GUI

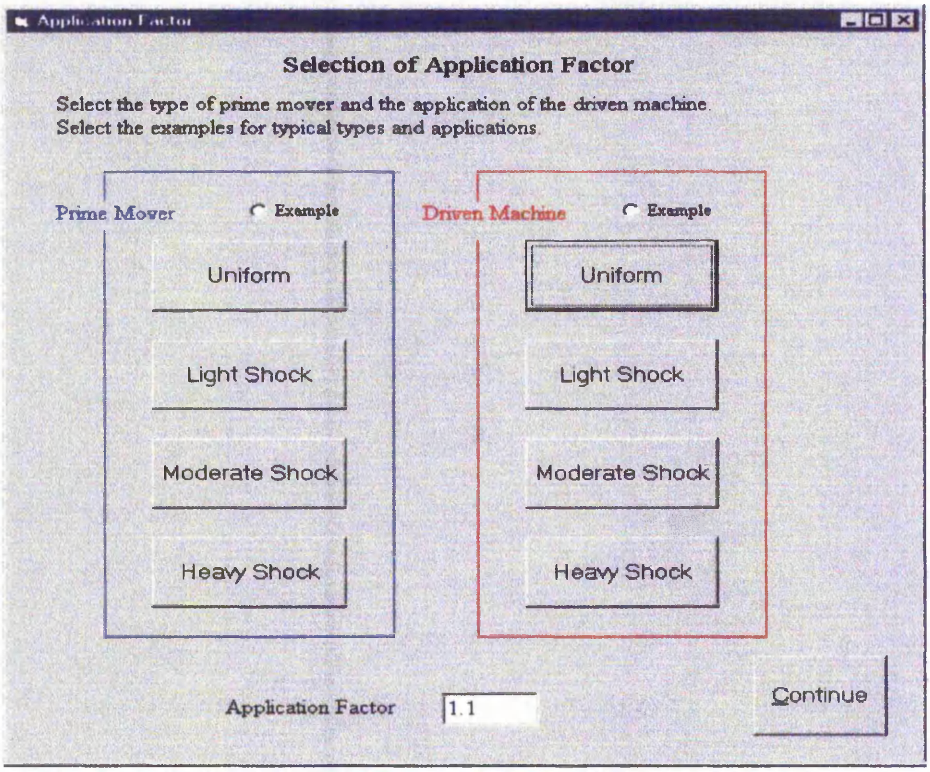


Figure 6.9 Application Factor GUI

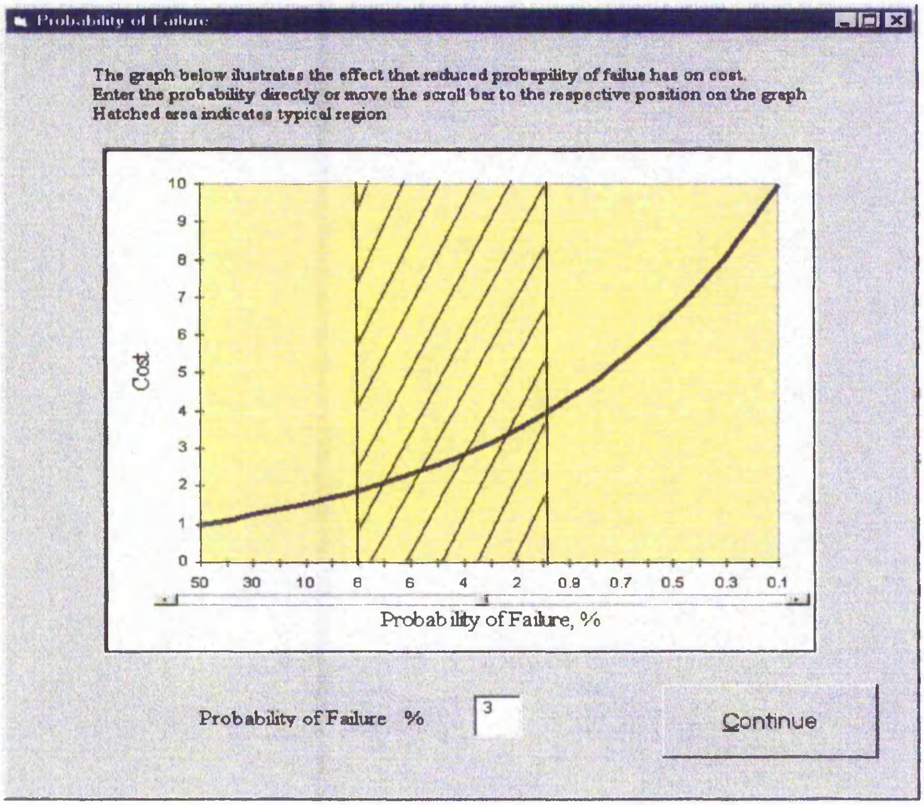


Figure 6.10 Probability of Failure GUI

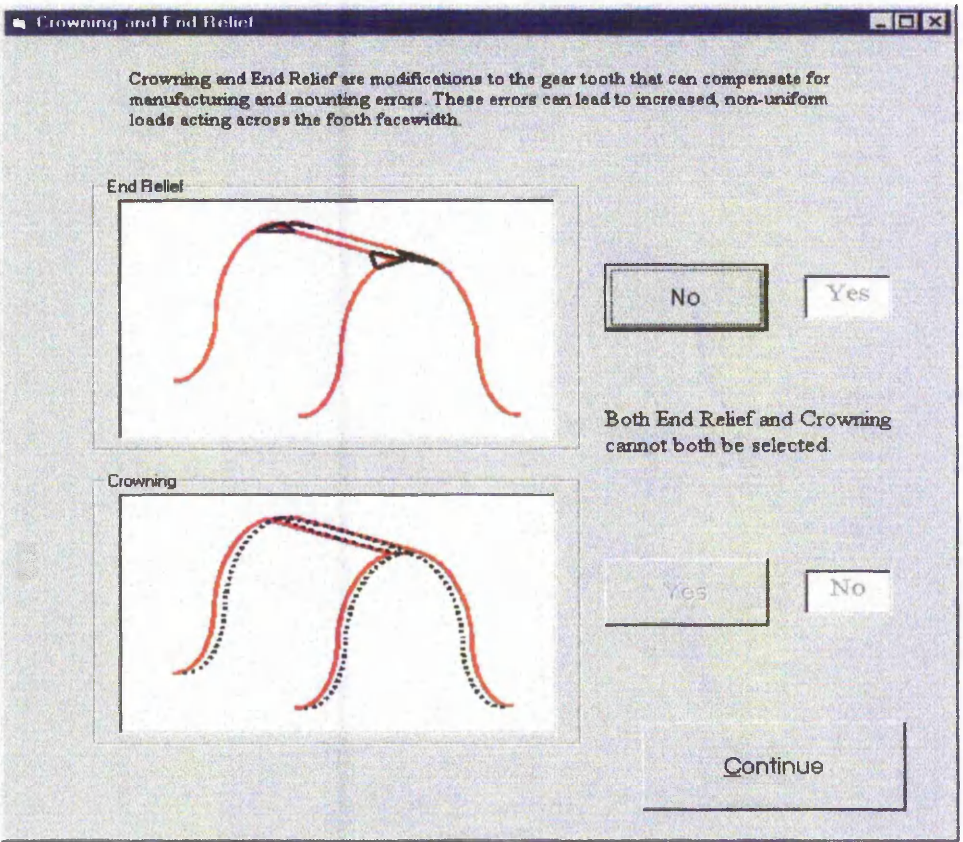


Figure 6.11 Tooth Modification GUI

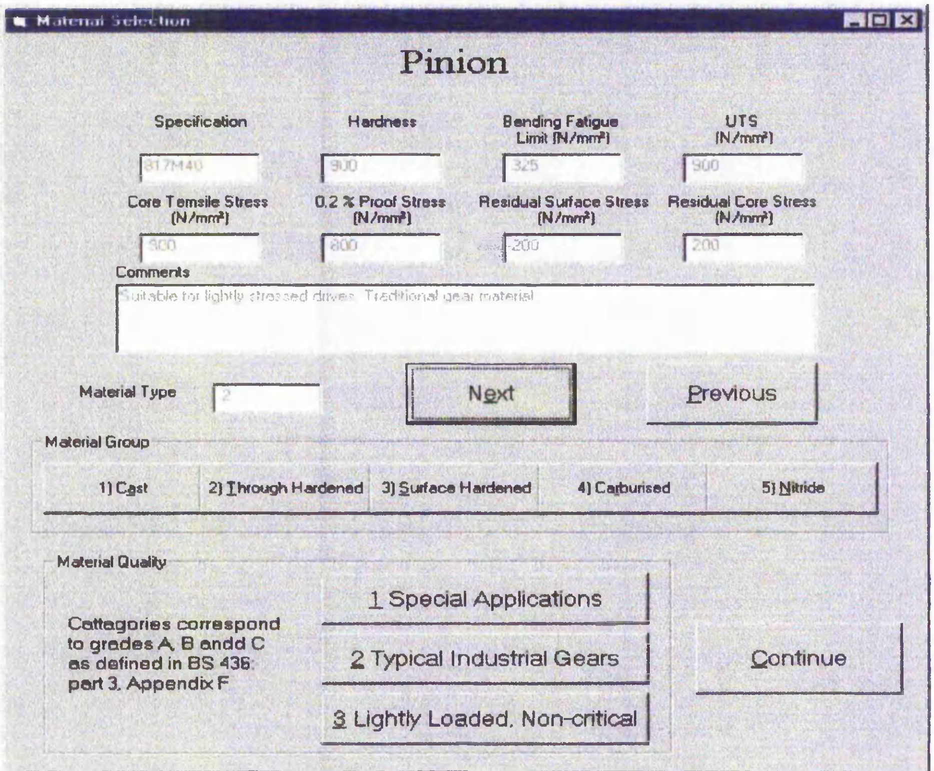


Figure 6.12 Pinion Material GUI

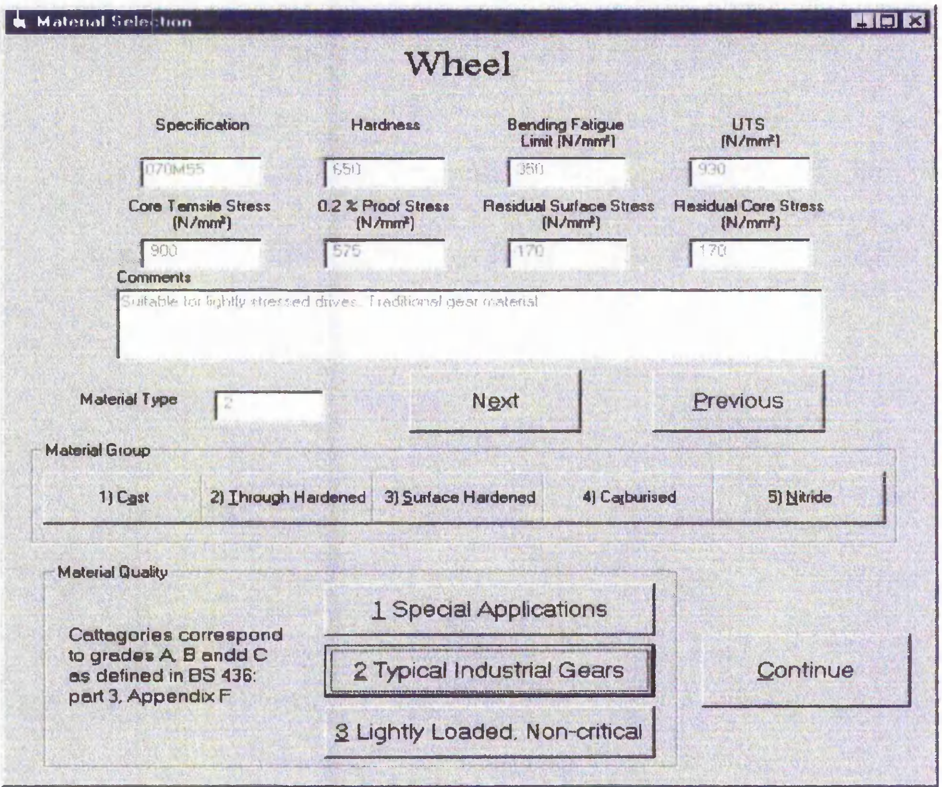


Figure 6.13 Wheel Material GUI

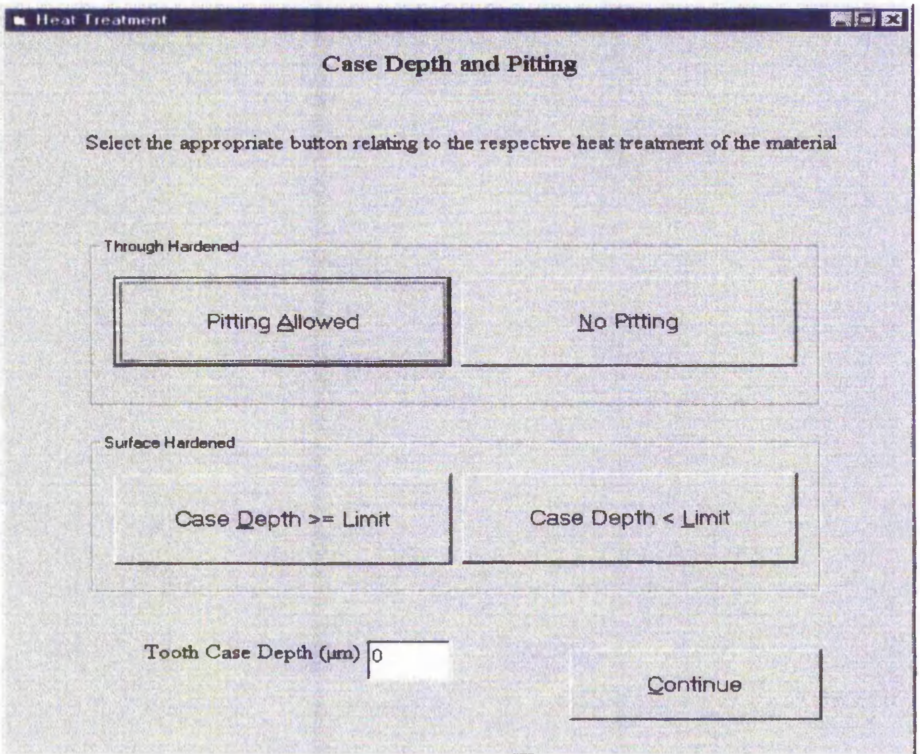


Figure 6.14 Case Depth and Pitting GUI

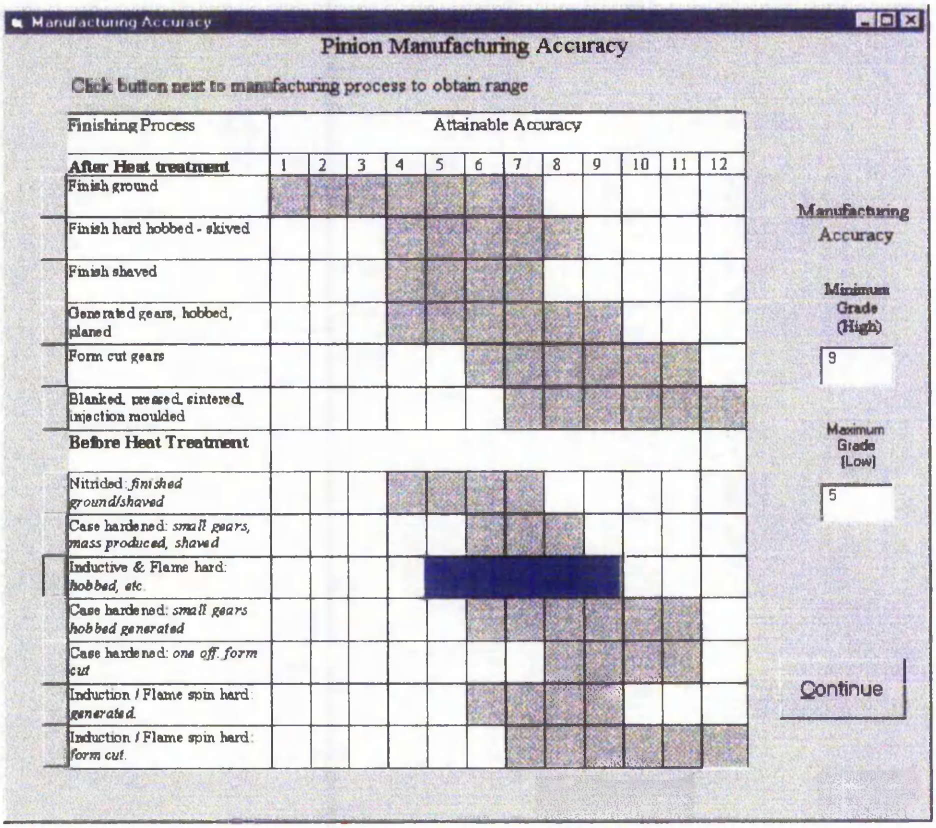


Figure 6.15 Manufacturing and Hardness Process GUI

Initial Gear Calculation (Stage 1)

Using the information obtained above, the initial design of the gear pair is derived using the gear design module described in section 5.4.7.

Gear Parameter	Value
Number of Teeth, Pinion	26
Number of Teeth, Wheel	96
Module	2.0 mm
Pressure Angle	20°
Helix Angle	45.0°
Facewidth, Pinion	128.68 mm
Facewidth, Wheel	128.68 mm
Addendum Modification Coefficient, Pinion	0.0
Addendum Modification Coefficient, Wheel	0.0
Addendum Coefficient	1.0
Tooth Clearance	0.25 * Module
Pitch Accuracy, Pinion	5
Pitch Accuracy, Wheel	5
Safety Factor S_H	1.04195
Safety Factor S_F	1.45712
Facewidth Ratio	1.4

The initial performance of this design is calculated using the British Standard 436 part 3 (1986) and the geometric calculations found in Appendix B.

Parameter	Value
Permissible Contact Stress, Pinion	331.6
Actual Contact Stress, Pinion	1226.0
Permissible Contact Stress, Wheel	327.6
Actual Contact Stress, Wheel	1226.0
Permissible Bending Stress, Pinion	531.6
Actual Bending Stress, Pinion	475.4
Permissible Bending Stress, Wheel	524.2
Actual Bending Stress, Wheel	519.0
Contact Ratio	15.53

Initial Bearing (1st Shaft Sub-assembly)

Initial information that the bearing selection is based upon is indicated in Figure 6.16. This information forms the input information to the bearing type selection ANN. For the purpose of the validation the ANN has not been set to replace the designer, allowing a comparison of its performance to be established.

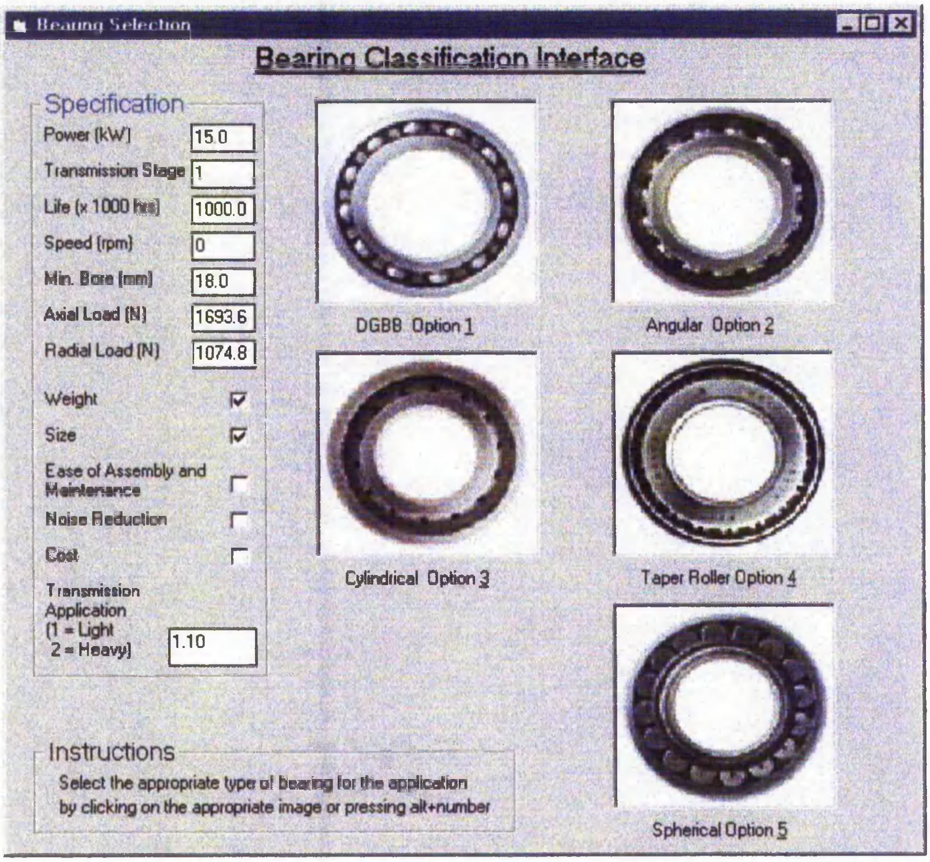


Figure 6.16 Stage 1 Bearing Type Selection using GUI

From the GUI the Deep Groove Ball Bearing was selected. This selection was based upon the radial force, light application and a small initial bore. The axial load has been taken into consideration, but at this magnitude will not impair the chosen bearings performance.

The bearing selection made by the ANN is illustrated below and indicates that the Deep Groove Ball Bearing should be used for the design. The ANN used has been trained with basic selection information, (as the IIS has not currently been fully developed in an industrial environment).

0.941073	*Deep Groove Ball Bearing*
0.809517	*Angular Contact Ball Bearing*
0.775967	*Cylindrical Rolled Bearing*
0.781107	*Taper Roller Bearing*
0.778499	*Spherical Roller Bearing*

Using the bearing catalogue and the initial bearing information the bearing is selected based upon the bore diameter, the speed, and the load rating. The load rating calculated from the life, speed and resultant force is 47589 N. The minimum bearing capable of satisfying these conditions is the 6309, the dimensions of which are given below.

Bore diameter	45mm	Code Number	6309
Outer diameter	100 mm	Dynamic Load	52700 N
Width	25 mm	Max. Speed	6700 rpm

To standardise sub-assembly design the same type of bearing is used at both positions of the shaft. The size or model of the bearing can however, differ, as selection from the bearing catalogue is individual to each application. For the initial bearing design the second bearing in the shaft sub-assembly is set to the same as the first. This is due to identical initial conditions. (These will change as the design develops).

Initial Gear Pair (Stage 2)

The additional specification definition of the material, manufacturing process and performance for the 2nd stage gear pair will be set to the same as 1st stage. The option to alter all the specification exists, but for the purpose of the validation and to allow comparison with the KBS using the same specification. Therefore, the GUI inputs are the same as in Figures 6.7 to 6.15

The critical dimensions and properties of the 2nd stage gear pair derived from this information are given below.

Gear Parameter	Value
Number of Teeth, Pinion	27
Number of Teeth, Wheel	72
Module	3.0 mm
Pressure Angle	20°
Helix Angle	45.0°
Facewidth, Pinion	161.32 mm
Facewidth, Pinion	161.32 mm
Addendum Modification Coefficient, Pinion	0.0
Addendum Modification Coefficient, Wheel	0.0
Addendum Coefficient	1.0
Tooth Clearance	0.25 Module
Pitch Accuracy, Pinion	5
Pitch Accuracy, Wheel	5
Safety Factor S_H	1.04195
Safety Factor S_F	1.45712
Facewidth Ratio	1.4

The initial performance of this design is calculated using the British Standard 436 part 3 (1986) and the geometric calculations found in Appendix B.

Parameter	Value
Permissible Contact Stress, Pinion	526.7
Actual Contact Stress, Pinion	539.4
Permissible Contact Stress, Wheel	510.3
Actual Contact Stress, Wheel	539.4
Permissible Bending Stress, Pinion	517.4
Actual Bending Stress, Pinion	70.7
Permissible Bending Stress, Wheel	507.6
Actual Bending Stress, Wheel	72.2
Contact Ratio	13.15

Initial Bearing (2nd Shaft Sub-assembly)

From the GUI the Deep Groove Ball Bearing was selected. This bearing selection is supported by the ANN. The resultant initial bearing for the lay shaft is given below.

Bore diameter	70mm	Code Number	6314
Outer diameter	150 mm	Dynamic Load	104000 N
Width	35 mm	Max. Speed	4500 rpm

Initial Bearing (3^{rd} Shaft Sub-assembly)

From the GUI the Deep Groove Ball Bearing was selected. This bearing selection is supported by the ANN. The resultant initial bearing for the lay shaft is given below.

Bore diameter	70mm	Code Number	6314
Outer diameter	150 mm	Dynamic Load	104000 N
Width	35 mm	Max. Speed	4500 rpm

6.2.3 Detail Design Stage (Refinement)

Refinement of the components that design comprises is performed. The refinement is performed using the detail design modules, either generating the component designs totally using the current conditions or optimising and adapting existing designs to account for related design modifications.

As the design contains single helical gears, the GA gear optimisation module is activated by the IIS, to improve the gears designs. The optimisation uses a GA to achieve the results, which in turn uses fitness criteria to access its performance, as described in section 5.5.4. Setting the fitness criteria will thus describe the desired characteristics of the resultant gears within the transmission system. The importance of each criteria is set using the GUI in Figure 6.17 below.

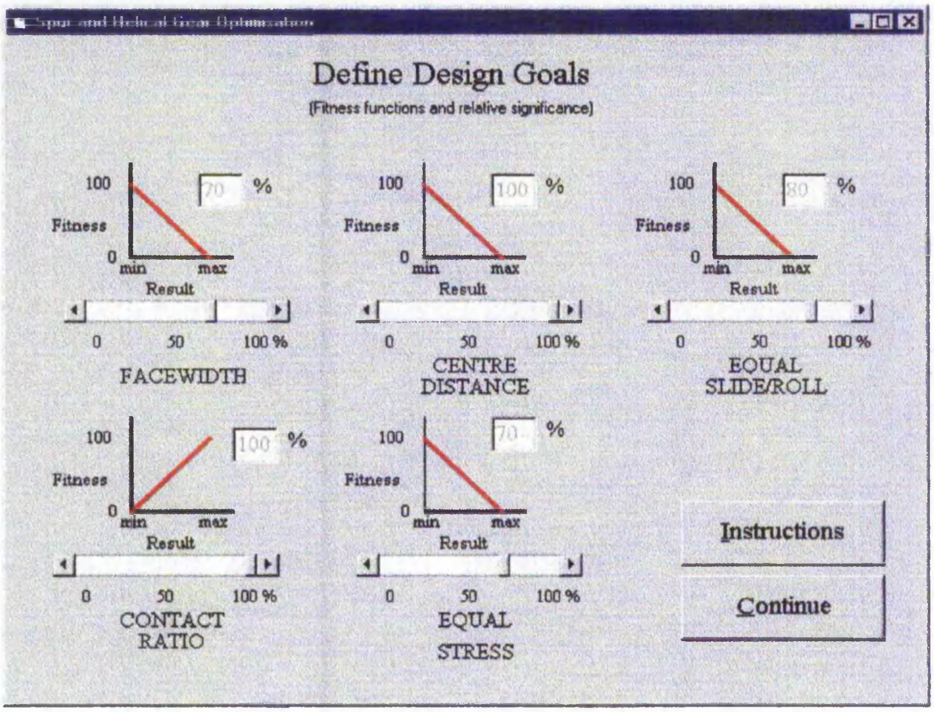


Figure 6.17 Optimisation Goals GUI (Importance of optimisation criteria)

Emphasis is placed on the minimisation of the centre distance and the increase of the contact ratio, while the facewidth, slide/roll ratio and equalisation of the bending stresses are not considered as important. The effects of these settings can be seen in the resultant gear designs in section 6.2.4.

The detail design stage performs the sub-assembly design of each shaft iterating until the design has converged, requiring no redesign. At this point all the geometrical information about the components is held in the respective component databases.

For the gear designs the final step, the selection of the standard part, is performed. Using the GUI the user selects the appropriate design option.

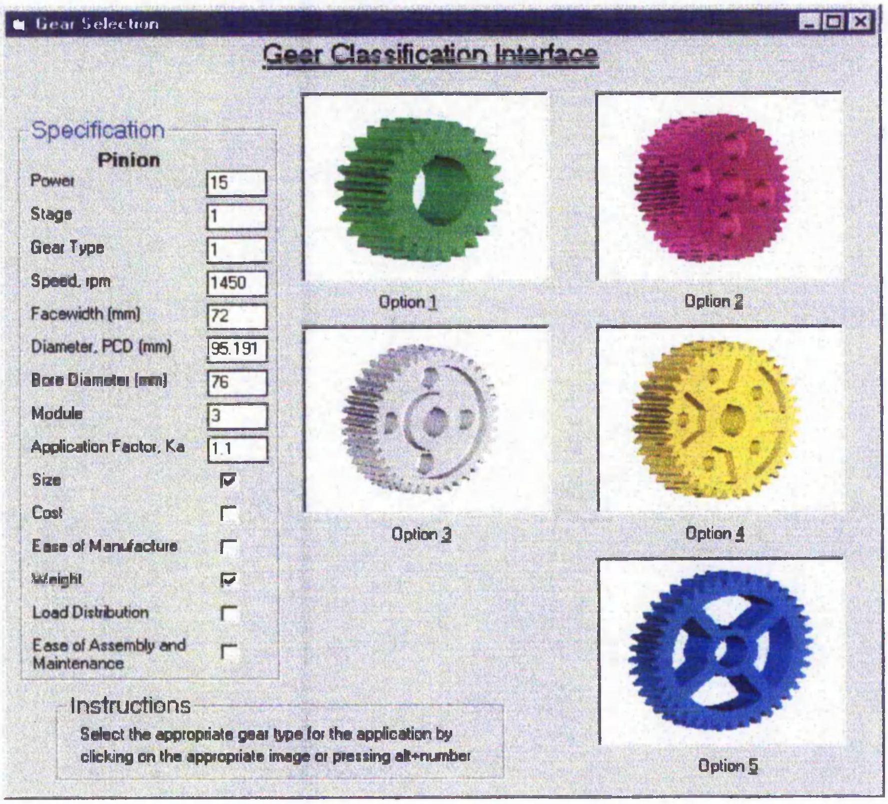


Figure 6.18 Pinion Gear Selection, 1st Stage

Figures 6.18 to 6.21 illustrate the GUIs presented to the user for the test case, defining the final specifications of the gears.

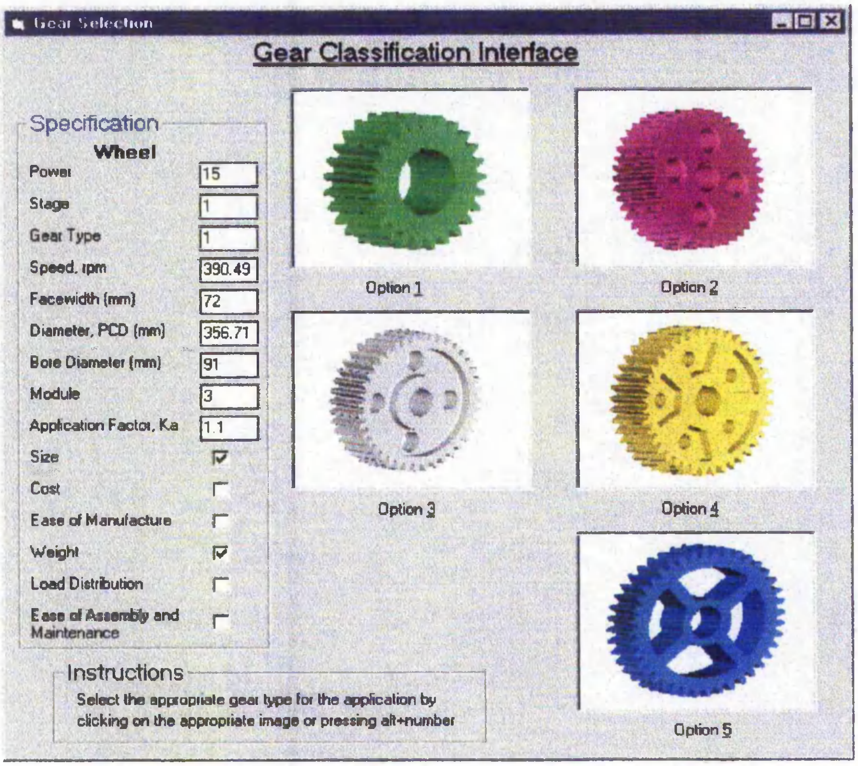


Figure 6.19 Wheel Gear Selection, 1st Stage

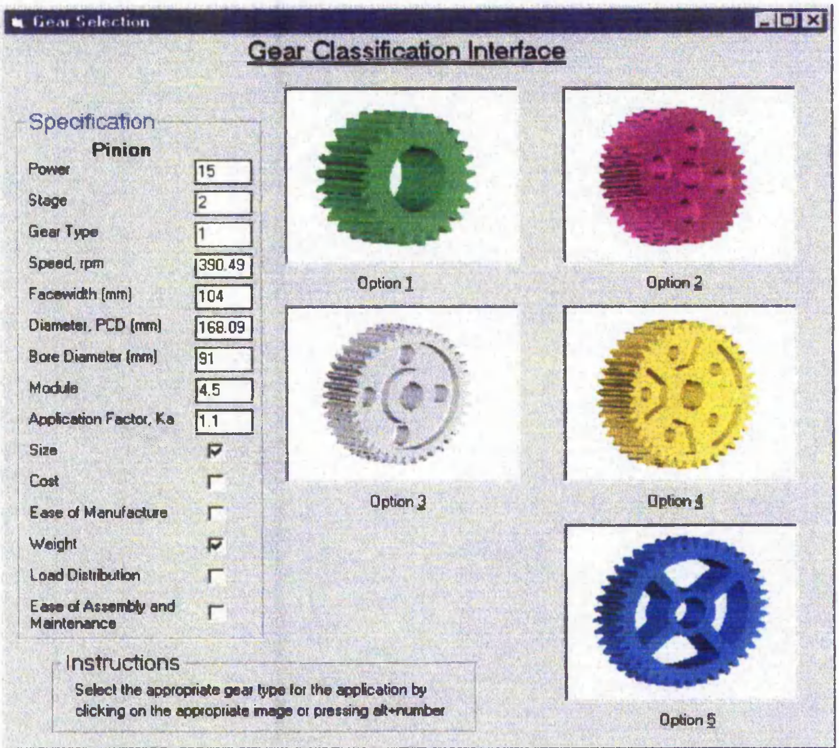


Figure 6.20 Pinion Gear Selection, 2nd Stage

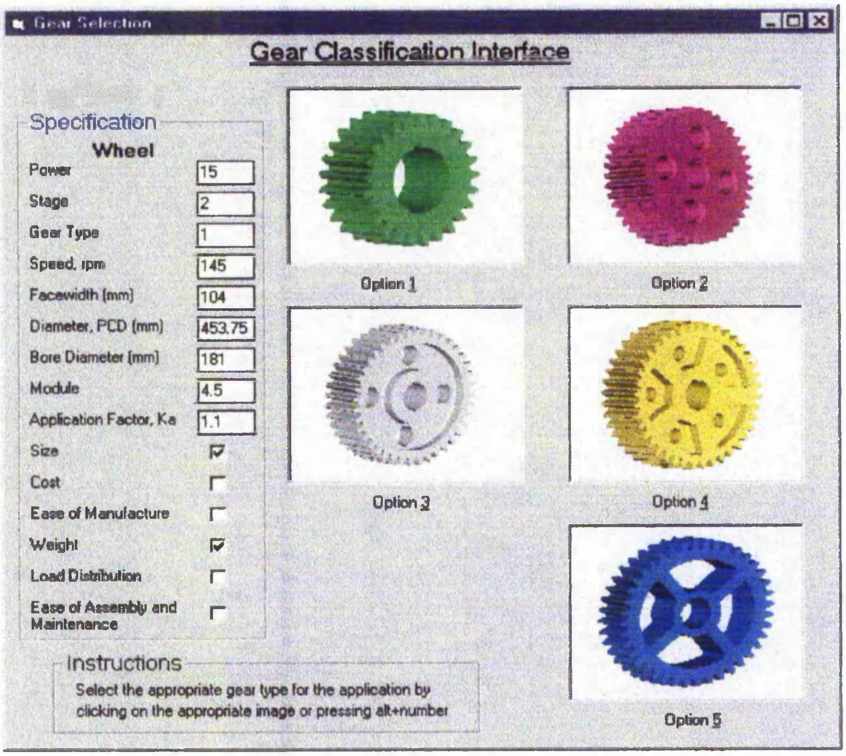


Figure 6.21 Wheel Gear Selection, 2nd Stage

The types of gear selected for each position can be found together with the gear design in section 6.2.4.

6.2.4 Component Information of Final Design

Shafts

Shaft Type	Input					
Length (mm)	64.0	98.5	72.0	10.0	175.0	33.0
Diameter (mm)	32.0	65.0	76.0	80.0	76.0	65.0
Shaft Type	Lay					
Length (mm)	101.5	72.0	10.0	104.0	107.0	
Diameter (mm)	80.0	91.0	95.0	91.0	80.0	
Shaft Type	Output					
Length (mm)	42.0	133.0	10.0	104.0	108.5	126.0
Diameter (mm)	170.0	181.0	185.0	181.0	170.0	63.0

Spacers

Position	2	1	5	5
Bore Diameter	65.0	80.0	80.0	170.0
Length	65.5	62.5	68.0	66.5

Bearings

Bearing Location	2	6	1	5	1	5
Type	1	1	1	1	1	1
Bore Dia. (mm)	65.0	65.0	80.0	80.0	170.0	170.0
Outer Dia. (mm)	140.0	140.0	170.0	170.0	260.0	260.0
Width (mm)	33.0	33.0	39.0	39.0	42.0	42.0
Code Number	6313	6313	6316	6316	6034	6034
Max. Speed (rpm)	4800	4800	3800	3800	2200	2200
Dynamic Load Rating (N)	92300	92300	124000	124000	168000	168000
Inner Shoulder (mm)	76.0	76.0	91.0	91.0	181.0	181.0
Outer Shoulder (mm)	129.0	129.0	159.0	159.0	249.0	249.0

Gears

Type	1	1	1	1
Number of Teeth	26	97	31	83
Module (mm)	3.0	3.0	4.5	4.5
Facewidth (mm)	72.0	72.0	104.0	104.0
Pressure Angle (deg.)	17.5	17.5	17.5	17.5
Helix Angle (deg.)	35.0	35.0	35.0	35.0
Tip Diameter (mm)	104.2	365.4	183.3	469.0
Tooth Depth (mm)	11.9	11.9	19.5	19.5
Bore Diameter (mm)	76.0	91.0	91.0	181.0
Rack Tip Radius (mm)	0.900	0.900	1.125	1.125
Addendum Coefficient	1.50	1.50	1.65	1.65
Addendum Modification Coefficient	-0.004	0.200	-0.201	-0.201
Permissible Contact Stress (N/mm)	344.4	371.9	334.1	352.1
Actual Contact Stress (N/mm)	343.3	343.3	317.4	317.4
Permissible Bending Stress (N/mm)	490.5	498.0	489.3	492.7
Actual Bending Stress (N/mm)	48.7	48.6	55.5	55.5
Contact Ratio	6.32	6.32	6.19	6.19
Transverse Contact Ratio	1.94	1.94	1.97	1.97

Assembly

Centre distance	Total	360.2 mm		
between shafts	1st Stage	151.6 mm	2nd Stage	208.6 mm

6.4.5 Design Drawings

Below, in Figure 6.22, is the section view of the general assembly for the two stage gear transmission. No casing is included as the transmission will form a sub-assembly of a larger design, within which it will be housed.

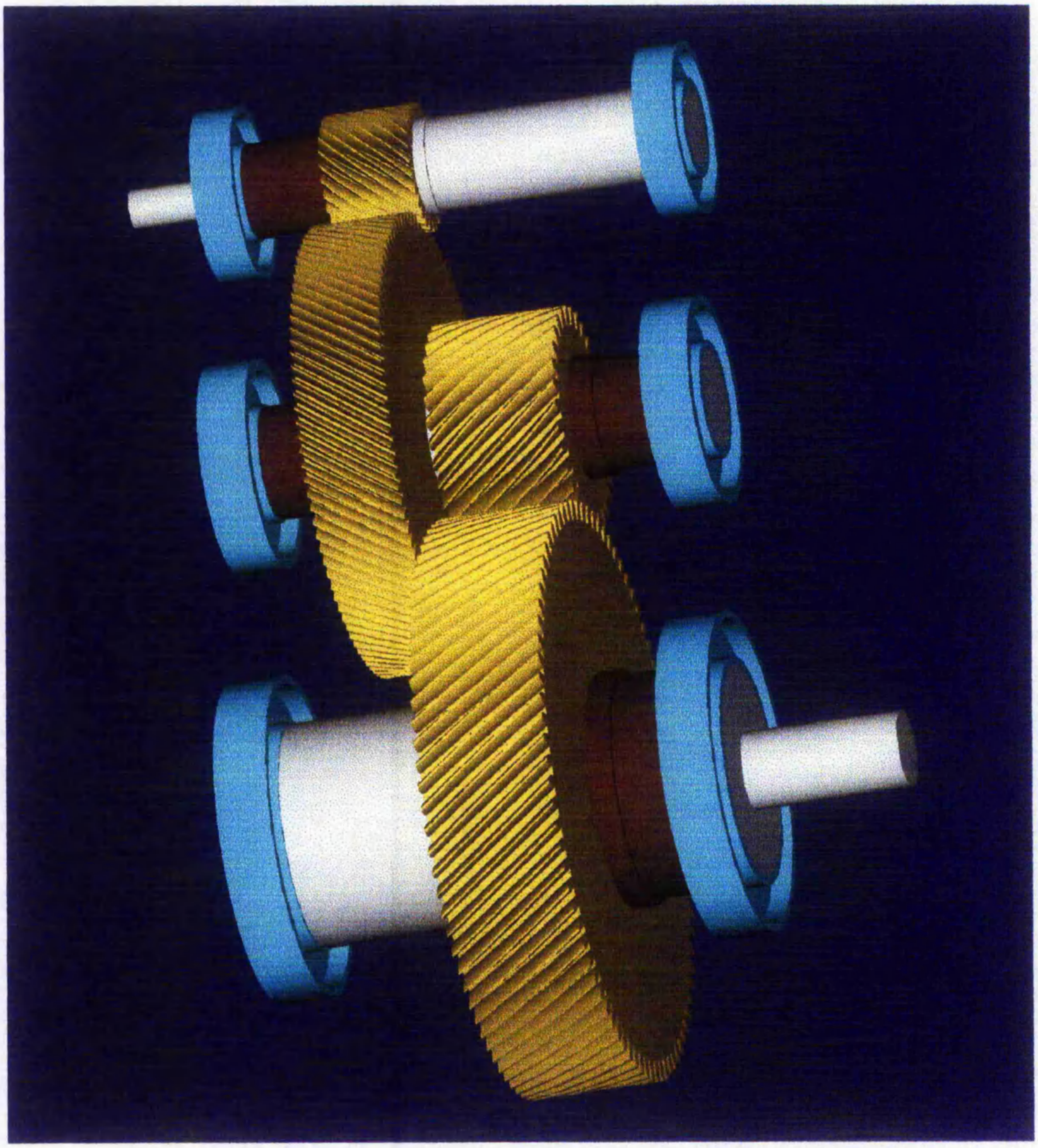


Figure 6.22 Assembly of Resultant Design (ProENGINEER Model)

6.3 Knowledge Based System Design

The transmission design used as the comparison for the IIS was generated using the knowledge based system developed by Su (1990). This system is based on similar knowledge as the IIS, thus allowing the performance of the IIS to be evaluated. The design given below uses the same PDS as the IIS design and the same or similar specifications whenever possible.

For the same PDS and specification the KBS generated the following design:

Arrangement	2 Stage Parallel with no fixed casing
1st stage transmission	Helical Gear
2nd stage transmission	Helical Gear

Centre distance

Total	470mm		
1st Stage	219mm	2nd Stage	250mm

Shaft Diameters

Input	30mm	Output	40mm

Gears

1st Stage	Ratio		3.944
	Module		2.5mm
	Facewidth		100 mm
	Helix Angle		12.898 deg.
	Number of Teeth	Pinion	36
		Wheel	135
	Addendum Mod. Coef.	Pinion	0.0
		Wheel	-0.14
2nd Stage	Ratio		2.667
	Module		3.5mm
	Facewidth		130 mm
	Helix Angle		13.00 deg.
	Number of Teeth	Pinion	38
		Wheel	101
	Addendum Mod. Coef.	Pinion	-0.02
		Wheel	0.12

Bearings

Shaft 1	Bearing Code	6008
	Outer Diameter	68mm
	Bore Diameter	40mm
	Width	15mm
Shaft 2	Bearing Code	6008
	Outer Diameter	68mm
	Bore Diameter	40mm
	Width	15mm
Shaft 3	Bearing Code	6310
	Outer Diameter	110mm
	Bore Diameter	50mm
	Width	27mm

Using the OPTGEAR package the performance of the gears designed by the expert system for the two stages of the transmission system have been determined.

Gear	Stage 1		Stage 2	
	Pinion	Wheel	Pinion	Wheel
Permissible Contact Stress (N/mm)	598.8	540.8	569.1	517.3
Actual Contact Stress (N/mm)	517.5	517.5	491.1	491.1
Permissible Bending Stress (N/mm)	551.7	1177.3	552.9	1178.1
Actual Bending Stress (N/mm)	135.7	145.0	120.4	126.2
Total Contact Ratio	4.69	4.69	4.18	4.18
Transverse Contact Ratio	1.71	1.71	1.52	1.52

6.4 Comparison of Designs

The comparison made between the two design systems will concentrate upon the resultant designs generated, (conceptual and detailed). This is due to variations in the design paths used by each system to achieve each design.

Conceptual Design

The conceptual design generated by both the IIS and the KBS match. Both the arrangement of the transmission system and the primary transmission components selected are the same.

Detail Design

The coincidence of the IIS and the KBS designs allows the detail design of the IIS to be scrutinised, enabling direct comparison.

Comparisons between the designs are limited to the assembly, gear designs and selected bearings. Limitation is due to the availability of the KBS design information.

Assembly

The centre distance between the input and output shafts for the IIS design is 360.2 mm compared with 470.0 mm for the KBS design. This represents a 23.4% reduction.

Gear Design

Stress The gear designs for both the IIS and the KBS comply with the British Standard BS 436 part 3 for bending and contact stress. This is essential for the transmission system to be valid, as failure to comply would result in failure of the gear teeth.

It can be seen from the contact stresses calculated for the 2^{nd} stage initial gear design that the gear is in failure. However, the purpose of the initial gear design is to generate a rough design that the optimisation module can improve upon. Therefore, failure at this point is not critical. Upon completion of the design the optimisation module has corrected this error, resulting in a successful high performance gear for the 2^{nd} stage.

Equalisation of the bending stresses within a gear pair has been identified as a desirable characteristic. This characteristic has been displayed in both the designs. For the 1^{st} stage the KBS, using the commercial gear design package DU436, achieved a 6.8% deviation while the IIS, using the GA gear optimisation module achieved a deviation of 0.2%. The 2^{nd} stage results improved upon these with a 4.8% deviation for the KBS design and no deviation for the IIS. Both designs thus illustrate the stress equalisation characteristic, with exceptional results being achieved by the IIS design.

Contact Ratio A high contact ratio is a desirable feature for a gear pair, however, increased contact ratio should not be at the expense of increasing stresses beyond their design limits. (This has already been proven not to be the case for this gear pair). As the gear pair are single helical the obvious means of increasing the contact ratio is by increasing the facewidth and the helix angle, thus increasing the overlap ratio. Another subtle method is by increasing the addendum coefficient and thus the length of the teeth. From the transverse contact ratio for the KBS and the IIS gear designs it is shown that the IIS's ratio is higher, (1.94 compared to 1.71 for the KBS design for the 1st stage and 1.97 compared to 1.52 for the 2nd). As this method is independent from the overlap ratio it will also be effective on spur gears.

The total contact ratio for the IIS design exceeds the KBS designs. This has been achieved despite decreases in facewidth.

Size Size is an important characteristic of gear design, with a reduction in size being preferable. Size includes centre distance and facewidth, both of which are smaller for the IIS designs. The facewidth and centre distance for the IIS are 31.4% and 30.8% smaller respectively for the for the 1st stage and 5.5% and 16.5% smaller for the 2nd. These reductions are the result of the GA gear optimisation module and the iterative redesign process encouraged by the IIS.

Bearings

The bearing selections made by the user via the GUI and the ANN both match the KBS's design. Both systems selected Deep Groove Ball Bearings although Angular Contact Ball Bearings were a close alternative.

The bearings selected by the IIS were larger than the KBS's design. This has been accredited to the reduction in the centre distances, which have produced higher forces within the transmission system. Thus illustrating that the IIS has continually updated and modified the design to ensure that failure will not occur at any point of the design.

6.5 Remarks on the Application of the IIS

To validate the IIS the resultant final design produced does not need to be identical to that comparison design. However, the conceptual designs must coincide as they are both derived from the same design knowledge, although it is held and manipulated using different AI techniques. This has been shown to be true for the mechanical transmission system application. The success has thus shown that ANNs and production rules can be combined and that the resultant hybrid is capable of retaining design preferences and making decisions based upon this knowledge.

Within the detailed design stage, wider interpretation of the design knowledge has been expected. Therefore, variations between the final designs from the IIS and the KBS do not indicate a problem or failure of the system. Variations in the designs can be accredited to the use of different design modules and the knowledge that is retained within them. Thus, the validation of the IIS will be made not by the variation of the designs, but if the performance of the design has been increased or at least equaled. With respect to this criteria the IIS for mechanical transmission system design has been successful, also indicating successful combination of AI techniques has been achieved and that the intelligent hybrid approach to integrated design is feasible.

CHAPTER 7

DISCUSSION AND CONCLUSIONS

This project has developed an approach that integrates a combination of artificial intelligence (AI) techniques with the engineering design process to improve design quality and automation. The resultant intelligent hybrid approach has identified and addressed three main features, the design process, the design knowledge and the development of a computerised design system. The conventional design process is adapted to structure the development of a design identifying communication routes between stages and paths for redesign. Once the design process is established the knowledge and expertise to achieve the design is analysed to determine the most effective method of encapsulation. Selection of the appropriate method is dependant upon the form of the knowledge (well defined, data intensive or experience based and incomplete) and the ability of each method at handling the form. The computerised system utilises a modular structure, breaking down the system into the stages of the design process and the design tasks contained within. The modular structure enables the combination of AI techniques with different ways of presenting results and the integration of separate design packages within a single common environment. The single environment of the intelligent integrated system (IIS) provides the medium for the transfer of common information, thus increasing the continuity of design and removing the necessity for repeated input of design specifications.

During the development of the approach and the IIS, the application of ANNs has resulted in an investigation into the training process of backpropagation neural networks. The intention of the investigation was to establish a relationship between the performance of the network and the configuration of the network and its training process. The conclusion of the investigation was that the best method of improving the performance of a backpropagation was trial and error. The lack of relationships and the need for a process to aid the time consuming task of network training lead to the integration of a genetic algorithm (GA) into the training process. Use of the GA to adaptively adjust the training period, type of transfer function and network topology allows the information contained within the ANNs to quickly be modified. Development of the GEN-NEU program, increases the performance of the resultant networks in comparison to networks trained by the manual trial and error process, while reducing the level of ANN expertise required to train the networks. Thus, information within the IIS that is contained in ANNs can easily be modified, preventing the system becoming obsolete.

The fundamental aims of the intelligent hybrid approach are to utilise a combination of AI techniques to model the complicated process of design and integrate its various stages within a single environment. The combination of AI techniques led to the utilisation of a GA for design improvement. The improvement of gear designs was the application it was used for, proving very successful.

High performance designs have been achieved by the GA without the necessity for a gear design expert. Analysis of the resultant designs show that with the aid of parameter limits and penalties, highlighting design failures, the performance of a design can be improved based upon the desired goals. The success of the GA for spur and helical gear design and industrial interest has resulted in the development of a separate design package, called OPTGEAR. This package utilises the gear design module used by the IIS for mechanical transmissions and combines a graphical user interface to perform pre and post process operations. An example of the package can be found in Appendix E.

7.1 Development and Features of the Intelligent Hybrid Approach and IIS

The literature review revealed that previous research has been undertaken into the topic of application of AI into computer aided design systems. The majority of these systems have been ES or KBS, both of which have produced valid results, however, they displayed inherent problems. Due to their structure the systems display difficulties emulating ill-structured knowledge, predicting actions and allowing modification and the update of information. ANNs have also been used within the design process, but only applied to specific areas, producing unstructured results that make their integration into a system difficult. This is not to say that these techniques are not useful for design, but merely that each technique has shortfalls. This research combines these techniques into a system that benefits from the strengths of each of the AI techniques, offering new possibilities to the simulation of expertise within a system. The rules offer structure and logic while the ANNs offer both structure and flexibility depending upon their application. This combination of techniques allows rigid rules and circumstances to be dealt with by rule bases and databases, (as with previous systems) while taking advantage of the networks ability to encapsulate and manipulate knowledge that is difficult to quantify or structure. Based on the review, this unique combination of techniques has not been applied to Total Design or an integrated computer aided design system, thus making this research not only necessary but also novel.

A design approach has been developed to construct a system that combines and applies AI techniques to the design process. This approach structures the design process, the acquisition of knowledge and the structure of the system. All three of which are considered simultaneously when developing the system due to their inter-linked relationship. The development of the intelligent hybrid approach has combined advanced design methodologies. In particular Pugh's (1990) Total Design model has been incorporated within an approach, orientated around the identification and categorisation of design knowledge and its encapsulation within a computerised system. It is this emulation of knowledge that adds intelligence to the system, enabling decisions to be made, mimicking those expected from a designer or expert in a particular area of design.

Development of the approach and the implementation of the IIS for the design of mechanical power transmission systems has progressed simultaneously. This has resulted in the categorisation and knowledge identification techniques, described in Chapter 4, being developed, providing a valuable aid to future systems. Throughout the development of the system and approach, two factors form the main considerations with respect to knowledge and the intelligence of the system:

i. What information/ knowledge is required?

   From this consideration the information about a stage in the design is scrutinised and broken down so that factors of the design that are conventionally overlooked are analysed, such as starting points and designers preferences. These are decisions undertaken by designers every day without conscious consideration and forms part of their expertise.

ii. How the information/ knowledge is to be captured?

   Identifying the form of the knowledge is not considered a stage in any conventional design methodology. This is due to the designer concentrating on what he knows, not how he knows it. However, within a computerised system the form of the knowledge will govern the method of encapsulation. Therefore, categorising the knowledge is essential for the systems development.

Consideration of these two factors is essential throughout the development of the system as they directly influence its structure.

One of the aims of the project is to combine different AI techniques, to exploit the advantages of each and compensate for the weaknesses in others. The use of production rules is well established and many previous works have used them to represent well-defined knowledge in the form of IF *condition* THEN *action*. Indeed this project uses them for the same purpose. However, they do not allow for noisy incomplete information or make an allowance for variations in conditions. If no rule meets the condition exactly no action is taken. This is one of the situations where the use of ANNs has proven helpful due to their ability to cope with incomplete or noisy information, always providing an output. Hence, combining these AI techniques provides a creative quality to the problem solving capacities of the system.

The detail design stage is, as mentioned, the most information intensive stage of the design process, requiring a combination of engineering principles to generate component dimensions and physical properties to ensure that failure will not occur and optimum design is achieved. Information about the relations between components within the assembly is essential to ensure redesign and modification of adjoining components, resulting in continuity of design. Redesign is a costly and time consuming process, therefore, introducing intelligence into the iterative design process within the system helps speed up the process. This is achieved by combining intelligent procedures within the rule base with identified modifying factors and relations between components. The result is the ability to act upon a multitude of different situations, enhancing the redesign process.

The use of ANNs within the system has offered alternatives to conventional methods of encoding knowledge and information. New applications are continuously being found for ANNs that indicates their full potential has not yet been established, particularly in the field of design. Therefore, particular attention, within the project, has been given to the application of ANNs to determine which of their properties will improve the computerisation of expertise within the design process.

7.2 Findings from the Application of ANNs for Design

Within the IIS ANNs have been used for prediction, classification and cased based selection and the retention of complex, ill structures and incomplete information. These applications have provided a cross-section of the uses that backpropagation ANNs can be applied to with respect to engineering design.

The application of the ANNs to conceptual design has provided a novel feature similar to an experienced designer's reasoning process, which is based upon previously similar situations. The networks are not only capable of making decisions based upon similar training cases, but can additionally select between solutions that are equally matched when evaluated by the weighted matrix method. An example can be found in section 5.3.4. Upon analysis it appears that the network will select the most common output with respect to the training data. This is due to two reasons:

i. The manner in which the post processor interprets the output from the network. The most suitable output is that with the highest value. This would have no effect upon the output if calculated from the weighted matrix as the results will be exact. With the ANN an error between the target and actual outputs will exist. It is from this error that the network is capable of deciding between two or more equally suited designs.

ii. The output from the network is dependant upon the connective weights. If the connective weights are small the output will be low and vice versa. Therefore, if two output elements, relating to two components, were given the same inputs the output element with the stronger connections would give the greater value of the two. The strength of the connections is determined during the training period. The greater the frequency and value of an output during training the stronger the connection to that output element. Similarly the opposite is true and is proven by the networks trained with outputs permanently set to zero. Therefore, if an output has a more frequent higher value during training it stands to reason that the strength of the connections will subsequently be stronger.

This property of the networks adds a feature to the system that displays characteristics of case based reasoning together with prediction, thus encouraging the use of standard parts, which is largely desired in modern design and engineering.

The ANNs also play an important role in knowledge representation in gear design. The detail gear design demonstrates how knowledge relating to a multitude of areas, ranging from specific knowledge about gear geometry to material properties, is required to perform the component designs. This range of knowledge consists of different types. Therefore, the best method of encapsulating knowledge within the system is by a combination of techniques. The application of the ANNs to the detail gear design has allowed the system to represent information that may have been accumulated over many years by a designer. This knowledge is frequently in the form of graphs contained within product catalogues that allow easy calculation of parameters, but the original data upon which the graphs are created is not available. The ANNs also offer the ability to predict results in a similar manner to interpolation, allowing the information that has been extracted from the graphs to be interpreted as a designer would.

7.3 Findings from the Application of Genetic Algorithms

GAs have been applied to both the training of backpropagation ANNs and design optimisation within this project. Although the applications are different, the same principles have been employed, thus illustrating the versatility of GAs.

The design optimisation process, that evolutionary programming has been applied to, has provided a means of improving the quality of the design without the necessity for extensive, detailed knowledge about the effects of modifying various parameters and dimensions. This is not to say that the use of GAs will remove the necessity for detailed knowledge in design, but does allow design performance to be improved without an expert in the particular area. For example the effect of the tooth profile shift on the equalisation of the bending stresses in the pinion and wheel, (as mentioned in section 5.5).

The GA adjusted the profile shift parameters to match this element of detailed gear design knowledge without specific instruction (in the form of rules) to guide it. The goal was achieved by the correct identification and application of the fitness function that described the goal. Therefore, it is the correct identification of goals, in the form of the fitness functions, that guides the search, the effect of which govern the success of the optimisation process. Detailed knowledge about a design process and the effect that modifications to design parameters have upon performance can be considered non-essential when using the GA approach. Instead it is the clear identification of the desired resultant design that is important.

During the development of the GA optimisation of gear performance additional observations were made and commented on in sections 3.1 and 5.5. These observations corresponded to the set up of the fitness functions, the scaling of relative fitness values, the size of population and the effect of the mutation operator. A summary of the observations follow:

It was found that the set-up of the fitness functions and the range of the search that they are applied to dramatically effects the repeatability of GAs results. It was found that applying local limits (extreme fitness values within the current population) continuously applies pressure on the search to converge upon the optimum. The alternative, global limits from the entire search loose their resolution when convergence of the population begins as the resolution of the fitness function is reduced.

GAs are renowned for being computationally expensive (Gen and Cheng, 1997 and Goldburg 1989). During the development of the GA applications within this project it has become evident that the set-up of the genomes, the population and the fitness functions collectively influence the performance of the search and the time taken to achieve an optimum. One of the main factors to affect the time taken by the search is the population. Determining the population size is performed by trial and error, but as discussed in section 5.5 the level of repeatability achieved is the main consideration. Figures 5.34a and 5.34b showed that the levels of repeatability increased with an exponential characteristic, relative to the population size, thus 100% repeatability cannot be practically achieved. The population size needs to be of an adequate size to ensure that the search area is comprehensively covered, while too large a population will increase the computational time unnecessarily for little increase in repeatability. The use of the combination of mutation operators has proved to help reduce the convergence time. This unique application of bit transfer and total genome mutation allows the population size to be reduced. This is due to the total genome mutation operator having the effect of introducing new population members at the beginning of the search, aggressively altering the search area. Then once the population starts to converge this operator is removed and the bit transfer operator subtly introduces variations to the search, allowing convergence to occur. This causes the search to gradually converge while taking into account trends of the search itself, mimicking an annealing effect similar to the momentum term of the ANN backpropagation paradigm. The level at which the total genome mutation operator is deactivated has been taken as half the convergence level. This relationship was used during development of the gear performance application and proved to be successful.

7.4 Multimedia and Integration of External Package

Multimedia and GUIs provide a facility that allows knowledge and information relevant to the design task to be presented clearly and conveniently to the designer at the correct point of the design. Hence, time consuming searches for the appropriate information are prevented. Combining the clear display of information with suggestions and prompts towards appropriate solutions or inputs provides an intelligent aspect to the GUI.

Integration of the FennerBelt package, described in section 5.4.4, required the analysis of the initial program before integration was possible. Incorporating existing packages within the IIS may lead to reduced development time and offer features otherwise unavailable. Integration with the FennerBelt package has incorporated the experience of the Fenner designers and their catalogue of standard parts within the system, together with a user-friendly environment.

7.5 Conclusions

The intelligent hybrid approach in this project has been successfully developed and tested by the application of the IIS for the design of mechanical transmission systems. Although the application is a prototype, the methodologies of the approach have been tested, resulting in new applications for AI techniques for engineering design.

Conclusions drawn from the work conducted for this project are:

i The literature review indicated that an approach or system that combines the advantages of multiple artificial intelligence techniques does not exist. Further more approaches and systems that integrate the total design process form concept to manufacture into a single system, incorporating intelligence within the system have not been found. Therefore, this research contributes toward meeting a requirement for greater integration in modern engineering.

ii The modular structure of the IIS allows the knowledge contained to be modified without disruption to the entire system.

iii The type of applications to which the AI techniques (production rules, GAs and ANNs) are suited to have been established. Production rules are best suited to well-defined knowledge, GAs enable a design to evolve without relational design knowledge and ANNs are capable of retaining discontinuous and ambiguous data. ANNs play a large part in the approach, which could increase as the full limits of their abilities and applications have not yet been established. When the application of the GA to the improvement of the gear design is also considered, it becomes evident that the use of AI techniques to enhance design performance is an area of engineering that will increase the general quality of design.

iv Computerisation and integration of the stages of the design process offers a solution to the industrial problem with respect to manpower. The aim of reducing the time and labour intensity of the design process has been met by the IIS. An additional feature is the ability of the employer to retain a designer's expertise after he has left the company.

v The ANNs have displayed the characteristic of performing decisions based upon known information and assumptions. The known information, which formed the training data is returned from the ANN as it was presented. However, when two options cause a conflict the ANN has shown the ability to decide between the options without external prompting. This is due to the higher re-occurrence of one of the options during training, forming stronger connections within the ANN. The result is to encourage standardisation within designs.

vi The application of GAs to component design has shown that this technique has the capability to substantially improve performance and quality of design.

vii The use of the ANNs in conjunction with the GUIs for design selection allows the IIS to be tailored to suit the environment it is used in.

CHAPTER 8

FURTHER WORK

The core of the Intelligent Hybrid Approach, developed within the project, allows for expansion with regard to the final stage of a products design, manufacture. The manufacturing stage of the design process is equally as complex as either the conceptual or detailed design stages, requiring expertise, technical information and knowledge in a variety of areas directly relating to the manufacture of the products design. With regard to the IIS, manufacturing should not be considered as just the final stage of the design process, but as a stage within the iterative cycle of the design's development. Faults with the design should be identified and corrective measures implemented at either the conceptual or detailed design stages. The manufacturing stage then becomes enveloped within the IISs single environment, completing Pugh's Total Design model (1990). The IIS will then encompass the complete product design process from concept to finished product, within a single environment.

The manufacture stage would exploit the advantages and capability of the various AI techniques together with CAE packages and multimedia in a similar fashion as for the conceptual and detailed design stages. Lessons learned and the techniques developed for this project with regard to the combination and application of the various AI techniques would increase the viability of a manufacturing stage for the IIS. The field of engineering manufacture comprises many areas, the most relevant to the IIS being planning, machining and costing. These areas currently provide popular topics for research into improving processes with the aid of AI techniques. A brief review conducted of some of the work in these areas concluded that it would be feasible to develop a manufacturing module for the IIS.

8.1 Outline of the Manufacturing Module

The manufacturing module would be constructed in the same manner as the conceptual and detail design modules. Maintaining a modular structure to enable the systems knowledge to be modified if required, preventing the system from becoming obsolete. This characteristic of the IIS is particularly important to the manufacturing stage of product design, as production processes and machine capabilities are continually changing. Within the manufacturing module the production process would be broken down into the three significant areas. Each area forms a separate module, exploiting the hybrid combination of techniques used to emulate the necessary expertise and knowledge required to perform the respective tasks.

8.1.1 Planning

Planning and scheduling the type and sequence of manufacturing processes required to produce the finished design of the product is an area of manufacture that extensive effort, development and research has been performed in. This ranges from the material requirements planning (MRP) to operation sequencing and utilisation of machining resources. Within this field of engineering the use of AI techniques has been widely used, including rule based systems, GAs and ANNs. The techniques of GAs and ANNs could make a considerable contribution to the IHS. Kumar and Srinivasan (1996), Lee and Kim (1995) and Feng et al (1997) among others have applied GAs to the job scheduling and planning demonstrating that this AI technique is capable of performing this task. However, integration with the IIS will require the manipulation of the GAs fitness constraints for each application and the preparation of the input parameters, typically through the combination with a rule base. Rule bases have been used to perform the job scheduling process for a number of years demonstrating acceptable results with this technique, but as with all rule based approaches, modification of the knowledge that is contained within can be difficult, conflicting with the easy modification goal of the IIS. Therefore, this approach would not be encouraged for development.

8.1.2 Machining

Machining is one of the core areas of the manufacturing stage of a products development. A multitude of processes and methods are applied to manufacture a product and with current computer numerical controlled (CNC) techniques the processes are becoming increasingly automated. The CNC processes would be the main manufacturing techniques that would be implemented within the manufacturing module. This is due to their inclusion within many computer aided design tools, such as the NCPost application within ProENGINEER (1997). The application produces the tool paths for the cutters based upon the machining drawings that are created at the end of the detailed design stage. For the use of NCPost within the IIS, the NCPost application of ProENGINEER would have to be linked to the system through an interface developed with ProTOOLKIT. ProTOOLKIT is the ProENGINEER customisation toolkit that allows third-party users to interface and control functions within the ProENGINEER environment.

8.1.3 Costing

Within the IHS the costing of a product is an integral part of the approach but is a process that if performed at the manufacture stage of a design is purely for the purpose of defining the finished cost. The resulting value having little affect upon the finished design. Several approaches have been applied to the costing of products that implement AI techniques including the use of ANNs and production rules. These are in addition to the analytical processes.

8.2 Component and Assembly Drawings

The development of the drawing interface will link the visualisation of the finished products design to the IIS. The essential dimensions of the components and the assembly layout have been established. An interface between the IIS and a parametric CAD package would therefore, enable the generation of the final design, forming a valuable communication tool. Additionally the automatic creation of the drawings would enhance the IIS and is the next step in the development of the Total Design process. Interfacing would remove the need for duplicated input and transfer of dimensions together with a reduction in error due to miss-reading.

The CAD package ProENGINEER is a suitable system to interface with the IIS. ProENGINEER is a parametric system that would allow refinements to component design to be contained within the drawings. These refinements would be in the form of parametric relations, which can scale fillets, chamfers and other finishing details to the critical dimensions generated by the detailed design stage of the IIS.

The interface with the IIS would be achieved using the customisation language/kit of ProENGINEER, ProTOOLKIT. This facility allows the functions of ProENGINEER to be utilised autonomously by an external program.

8.3 Assembly Components, Perpendicular and Cross Orientation

The IIS for the design of mechanical transmission systems developed as part of this research currently performs the detail design for parallel arrangements of the input and output shafts. Development of the design modules for seals, shaft closures, bevel gears and worm drives would complete the system, enabling the detail design of all the conceptual designs to be performed.

Throughout the development of the design modules the application of combinations of AI techniques can be employed. This will provide further opportunities to develop new applications of ANNs and GAs in the field of engineering.

8.4 Design Library

To prevent repetition of designs, a design library module could be introduced to the IIS. The purpose of the module would be to evaluate the current design and compare it with previously generated designs. If a match is made, the previous design is extracted from a library, thus preventing repetition of drawings, tooling and part numbers.

8.5 Tribology

Lubrication has a dramatic effect on the life and performance of a power transmission system. Therefore, a module to evaluate the final design with respect to lubrication and friction would enable design failures due to these reasons to be identified and corrected before the expensive manufacturing process begins.

The further work described within this section indicates that this project forms a base for ongoing research, with the potential to enhance the design process.

REFERENCES

Abersek, B., Flasker, J. and Balic, J., 1996, Expert System for Designing and Manufacturing of a Gear Box, *Expert Systems with Applications*, **Vol. 11(3)**, pp 397-405.

Amirikian, B. and Nishimura, H., 1994, What Size Network is Good for Generalisation of a Specific Task of Interest ?, *Neural Networks*, **Vol. 7**, No. 2, Elsevier Science Ltd., pp 321-329.

Archer, L. B., 1965, *Systematic Method for Designers*, The Design Council, London.

Beale, R. and Jackson, T., 1990, *Neural Computing – An Introduction*, Adam Hilger, Bristol, England.

Bullock, G. N., Denham, M. J., Parmee, I. C. and Wade, J. G., 1995, Developments in the use of the Genetic Algorithm in Engineering Design, *Design Studies*, No. 16, Elsevier Science Ltd, pp 507-524.

BS 436: Part 3, 1986, *Method of Calculation of Contact and Root Bending Stress Limitations for Metallic Involute Gears*, British Standards Institution.

Caudell, T. P. and Dolan, C. P., 1989, Parametric connectivity: Training a constrained network using genetic algorithms, *Proceedings of the Third International Conference on Genetic Algorithms and Their Applications*, pp 370-374.

Caudell, T. P., Scott, D. G., Escobedo, R. and Anderson, M., 1994, NIRS: Large Scale ART-1 Neural Architectures for Engineering Design retrieval, Neural Networks, Vol. 7, Elsevier Science Ltd, pp 1339-1350.

Clancey, W. J., and Shortliffe, E. H., 1984, *Readings in Medical Artificial Intelligence: The First Decade*, MA: Addison-Wesley.

Counsell, M. J., 1997, Design Principles for Mechatronic Systems, *International Workshop on Engineering Design*, Lancaster University, England.

Cross, N. (Eds), 1984, *Developments in Design Methodology*, John Wiley & Sons, Chichester, England.

Cross, N., 1990, *Engineering Design Methods*, John Wiley & Sons Ltd., Chichester, England, ISBN 0-471-92215-3.

da Silva., J. C. and Dawson, D., 1997, The Development of an Expert System for Hydraulic Systems Design Focusing on Concurrent Engineering Aspects, *International Conference on Engineering Design*, Tampere, 19th-21st August.

da Silva., J. C. and Cheung, H. F., 1997, Computer Support Environment Applying Object Orientated Methods for Hydraulic System Design, *Lancaster International Workshop on Engineering Design*.

Escobedo, R., Smith, S. D. G. and Caudell, T. P., 1993, A Neural Information Retrieval System, *International Journal of Advanced Manufacturing Technology* **Vol. 8**, Springer-Verlag London Limited, pp 269-274.

ESDU 88033, 1988, *The Design of Spur and Helical Involute Gears*, Engineering Science Data Unit plc, London,

Feng. W, Burns. G. R. and Harrison. D. K., 1997, Using a Hybrid Genetic Algorithm to Optimise a Production Plan, *Proceedings of the Thirteenth National Conference on Manufacturing*, 9th-11th September,pp 129-133.

FennerBelt, Software Package, 1994, J. H. Fenner Ltd., Hull, England.

French, M. J., 1985, *Conceptual Design for Engineers*, Design Council, London.

Frost, R. A., 1986, *Introduction to Knowledge Based Systems,* Collins Professional and Technical Books, London.

Gen, M. and Cheng, R., 1997, *Genetic Algorithms and Engineering Design*, John Wiley & Sons, Inc. New York, ISBN 0-471-12741-8.

Goldberg, D., 1989, *Genetic Algorithms in Search, Optimisation and Machine Learning*, Addison Wesley.

Hebb, D. O., 1949, *The Organisation of Behaviour*, John Wiley & Sons, New York.

Hopfield, J. J., 1984, Neural Networks and Physical Systems with Emergent Collective Computational Abilities, *Proc. Natl. Acad. Sci. USA* **Vol. 79**, pp 2554-2558.

Hubka, V. and Eder, W. E., 1996, *Design Science – Introduction to the Needs, Scope and Organization of Engineering Design Knowledge*, Springer-Verlag, New York, ISBN 3-540-19997-7.

Hush, D. R. and Horne, B. G., 1993, Progress in supervised neural networks: What's new since Lippmann ?, *IEEE Signal Processing Magazine*, January, pp 8- 39.

Kaftanoglu, B., Ulugul, E. and Carkoglu, N., 1995, Computer-Aided Optimal Design Strategy of Power Transmission Systems, *Annals of the CIRP*, **Vol. 44**, No. 1, pp 83-86.

Kiernan, L., Kambhampati, C. and Mitchell, R. J., 1995, Using self Organising Feature Maps for feature selection in Supervised Neural networks, *Proc. IEE Int. Conf. Artificial Neural Networks*, Cambridge, pp 195-200.

Kumar. N. S. H. and Srinivasan. G., 1996, A genetic algorithm for job shop scheduling – A case study, Computers in Industry, 31, Elsevier Science B.V., pp 155-160.

Kumara, S. R. T. and Kamarthi, S. V., 1992, Application of Adaptive Resonance Networks for Conceptual Design, *Annals of the CIRP*, **Vol. 41** No. 1, pp 213-216.

Lapedes, A. and Farber, R., 1987, Non-linear signal processing using neural networks: Prediction and System Modelling, Los Alamos National Laboratory report LA-UR-87-2662.

Lee, C. Y. and Kim, S. J., 1995, Parallel Genetic Algorithms for the earliness-tardiness job scheduling problem with general penalty weights, *Computers in Engineering*, Elsevier Science Ltd., pp 231-243.

LINKWAY, Version 2.0, Software Package, IBM, Portsmouth, England.

Lippmann, R. P., 1987, An Introduction to Computing with Neural Nets, *IEEE ASSP Magazine*, April, pp 4-22.

MAAG, 1990, *MAAG Gear Book*, MAAG Gear Company Ltd., Switzerland.

Maniezzo, V., 1994, Genetic Evolution of the Topology and Weight Distribution of Neural Networks, *IEEE Transactions on Neural Networks*, **Vol. 5** No. 1, pp 39-53.

McCulloch, W. S. and Pitts, W., 1943, A Logical Calculus of the Ideas Imminent in Nervous Activity, *Bulletin of Mathematical Biophysics,* **Vol. 5**, pp 115-133.

Miller, G. F., Todd, P. M. and Hedge, S. U., 1991, Designing Neural Networks, *Neural Networks*, **Vol. 4**, pp 53-60.

Minsky, M. and Papert, S., 1969, *Perceptrons*, Cambridge, MA: MIT Press.

Niemann, G., 1978, *Machine Elements: design calculation in mechanical engineering, Vol. 2: Gears*, Springer-Verlag, New York.

Pahl, G. and Beitz, W., 1988, *Engineering Design - a systematic approach*, Wallace, K., (Eds.), Design Council London, Springer-Verlag.

Pan, P. Y, Cheng, K. and Harrison, D. K., 1997, A Neural-fuzzy Approach to the Selection of Journal Bearings, *Proceedings of the Thirteenth National Conference on Manufacturing*, 9^{th}-11^{th} September, Glasgow, pp 427-431.

Pandiarajan, V. and Dwivedi, S. N., 1993, Feature Based Expert Process Planning CAD System in Concurrent Engineering, *Manufacturing Systems*, **Vol. 22**, No. 1, pp 65-70.

Patnaik, S. N., Coroneos, R. M. and Hopkins, D. A. 1997, A Cascade Optimisation Strategy for Solution of Difficult Design Problems, *International Journal of Numerical Methods in Engineering*, **Vol. 40**, pp 2257-2266.

Peters, T. J., 1992, Encoding Mechanical Design features for Recognition via Neural Nets, *Research in Engineering Design – Theory, Applications and Concurrent Engineering*, **Vol. 4**, Springer-Verlag, New York Inc., pp 67-74.

Professional II Version 1.92, Software Package, Neuralware Inc.

Pugh, S., 1990, *Total Design: Integrated Methods for Successful Product Engineering*, Addison-Wesley Publishing Company.

Rosenblatt, R., 1959, *Principles of Neurodynamics*, Spartan Books, New York.

Rowlands, H., 1996, A Hybrid Approach for Optimum Design using a Genetic Algorithm, a Neural Network and the Taguchi Method, *Proceedings of ACEDC '96, PEDC*, University of Plymouth, pp 205-221.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J., 'Learning Internal Representations by Error Propagation', in D. E. Rumelhart & J. L. McClelland (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations. MIT Press, 1986.

Sharpe, J. E. E., 1995, Computer Tools for Integrated Conceptual Design, *Design Studies*, **Vol. 16**, Elsevier Science Ltd., pp 471-488.

Shtud, A. and Zimerman, Y., 1993, A Neural-network-based Approach for Estimating the Cost of Assembly Systems, *International Journal of Production Economics*, **Vol. 32**, Elsevier Science Publishers B. V., pp 189-207.

SKF, 1989, *General Catalogue*, Carl Gerber GmbH, Germany.

Su, D. and Forgie, R. J., 1989, Development of an Expert System Approach to an Engineering Design Procedure, *Proceedings of the IMechE, Engineering Design*, **Vol. 2**, pp 975-989.

Su, D., 1990, *Knowledge-Based Integrated System Approach for and Engineering Design Procedure*, Ph.D. Thesis, University of Strathclyde.

Su, D., 1993, Application of a Knowledge Based, Integrated-system Approach to the Conceptual Design of a Power transmission System, in 'Quality Through Engineering Design', W. Kuo (Editor), Elsevier Science Publishers B. V., pp 417-427.

Su, D., 1994, Gearbox KBIS: A Prototype Knowledge-based Integrated System for Gearbox Design, *Proceedings of the International Gear Conference*, Newcastle upon Tyne, pp 201-206.

Swift, K. G. and Doyle, R., 1987, Design for Assembly in CAD Using Knowledge-Based Systems, *Engineering Designer*, July / August, pp 12-14.

Takeuchi, J. and Kosugi, Y., 1994, Neural Network Representation of Finite Element Method, *Neural Networks*, **Vol. 7**, No. 2, Elsevier Science Ltd., U.S.A., pp 389-395.

Udo, G. J. and Gupta, Y. P., 1994, Applications of Neural Networks in Manufacturing Management Systems, *Production Planning and Control*, **Vol. 5**, No. 3, Taylor and Francis Ltd., pp 258-270.

Vaughn. M. L., 1996, Interpretation and knowledge discovery for multilayer perceptron network: Opening the black box, *Neural Computing and Applications*, **Vol. 4**, Spinger-Verlag London Ltd., pp 72-82.

Visual Basic, Version 4.0, Software Package, Microsoft Corporation, USA.

Wang, Z., Di Massimo, C., Tham, M. T. and Morris A. J., 1994, A Procedure for Determining the Topology of Multilayer Feedforward Neural Networks, *Neural Networks*, **Vol. 7**, No. 2, Elsevier Science Ltd., pp 291-300.

Wu, B., 1992, An Introduction to Neural Networks and their Application in Manufacturing, *Journal of Intelligent Manufacturing*, **Vol. 3**, Chapman & Hall, pp 391-403.

Woodward, J. A. J. and Corbett, J., 1990, An Expert System to Assist the Design for Manufacture of Die Cast Components, *Engineering Designer*, May / June, pp 14-17.

APPENDIX A

ARTIFICIAL INTELLIGENCE TECHNIQUES

A.1 Genetic Algorithm

The GA is an adaptive search technique that forms a type of machine learning, which derives its process from a simplification of the natural evolutionary process. The natural process comprises, in its simplest form, of DNA (deoxyribonucleic acid) chains called chromosomes, combining through reproduction forming offspring to pass on their characteristics to the next generation. The offspring in turn reproduce and repeat the cycle and pass on the characteristics. The selection of the chromosomes is based upon the fitness or success of each chromosome, according to Darwinian survival of the fittest. Therefore, the fitter the chromosome, the greater its chances of reproducing and passing on its characteristics to the next generation. The chromosome comprises of a series of building blocks called genes. These genes contain individual characteristics or pieces of information about the parent chromosome. It is the recombination of the parents' genes that transfers individual characteristics from generation to generation, provided its fitness is high.

Similarly to the natural process the GA simulates the chromosome with a chain of information, containing the characteristics of a solution to the problem. These chains are referred to as genomes within the GA and comprise genes that hold the information relating to the parameters to be optimised. These genes correspond to the characteristics of the solution which are to be passed to the next generation. The genes comprise of a sequences of bits which represent the information in encoded form (as illustrated in Figure A.1 below). The methods of encoding may vary, but in most cases they are based upon binary notation. Encoding to Base 2 will be the encoding method used to explain the encryption procedure. Values are converted into binary, and the string of bits will in turn form the gene, which combine to form the genome. Constructing the genome from multiple genes allows many parameters to be modified simultaneously despite compound inter-related effects.

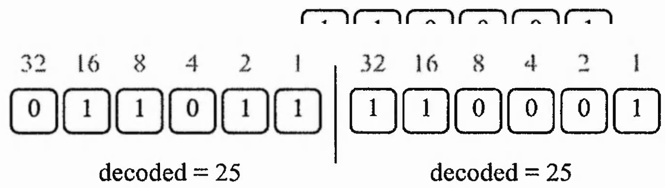


Figure A.1 Two Gene Genome Encoded to the Base 2

The process of combining genomes through reproduction is termed crossover when related to GAs. The process consists of splitting the genomes at one or more points, known as the crossover points, and swapping the segments between parents. Determining the number and position of crossover points can be achieved by a number of methods. The factors that determine which methods are chosen are the application the GA is being applied too and the construction of the genome. The actual position is generally determined at random along the genome, but can be fixed to ensure that genes are not altered. However, the fixed method will not increase the search area and is therefore not truly adaptive. The number of crossover points can be either single or multi-point.

Single point crossover takes a single point and swaps the second half of the genomes to form the offspring, as shown in Figure 3.2a. This method allows complete genes to pass from one generation to the next without alteration, which is good, with respect to the transfer of strong genes, but does not increase the search area and can therefore lead to a local optimum. This can be compensated for, to a degree, by increasing the population. An additional characteristic is that if the genome contains more than two genes the optimisation process will take longer or would terminate without achieving its goal.

Multi-point crossover uses two or more points to divide up the genome. The location of these points can be fixed, with similar problems as the single point. Additionally the number of points can be pre-determined or random. The combination of the two random features for crossover is generally the best and most robust method. Dividing the genes at any point allows for greater modification to the values which will decrease as the search converges on a solution, and the genomes within the population become identical. Once convergence has been reached, crossover will no longer have any effect upon the evolution of the genome. Figure A.2b illustrates the method.

0 1 1 0 0 1 0 1 0 1 1 1 0 1 0 1 0 0 1 0

0 1 1 0 1 1 0 1 **0 1 0 0 1 0 1 0 1 0 0 1**

0 1 1 0 0 1 0 1 0 1 0 0 1 0 1 0 1 0 0 1

Figure A.2a Single Point Crossover

0 1 1 0 0 1 **0 1 0 1 1 1 0** 1 **0 1 0 0 1** 0

0 1 1 **0 1 1** 0 1 0 1 0 0 1 **0** 1 0 1 0 0 **1**

0 1 1 0 1 1 0 1 0 1 1 1 0 0 0 1 0 0 1 1

Figure A.2b Multi-Point Crossover

The selection of the parent genomes is dependant upon their relative strength or fitness within the population. Therefore, the calculation of the fitness is a critical stage within the GA, and has the greatest effect upon guiding the search. The fitness is the measure of how successful the information encoded within the genome has been toward achieving an optimum solution. The calculation that determines the fitness of the parameters within the genome is termed the fitness function and can consist of a single or combination of calculations, depending upon the optimisation requirements. It is these requirements that govern the type and nature of the fitness function, which is unique to the application the GA is being applied to. From a review of current applications and their use within the project it has been found that there are no rules for constructing the fitness functions. Therefore, the aims and requirements of the search must be clearly identified and the functions modeled to simulate these goals.

Selection of the parent genomes for crossover is based upon the survival of the fittest. Therefore, the fitter the genome, the greater the chance of reproduction and passing on its genes to the next generation. However, there are variations of this process. Selection of the parents is either by direct selection of the fittest portion of the population or via a probabilistic method.

The direct method takes the top proportion (a predefined section of the fittest genomes) of the population and combines them by crossover to produce the next generation. Selection of the parents within the top proportion is made at random. Additionally, genomes may be allowed to pass through directly to the next generation. This process rapidly hones in on an optimum solution, but tends to lead to super-convergence upon a local optimum. This is due to genomes that contain fit genes, but have an overall unfit bias being rapidly removed from the population. Therefore, this method is not recommended when searching for a global optimum.

The probabilistic method determines the probability of reproduction for each genome based upon its fitness in relation to the rest of the population. The selection process is based upon the associated probability of the genomes and a random factor. The process is often compared to a roulette wheel as this best illustrates it. The greater the probability, the larger the genomes proportion of the wheel, as shown below in Figure A.3. It is therefore, possible for a gene to become a parent more than once, passing on its strong characteristics to more of the next generation.

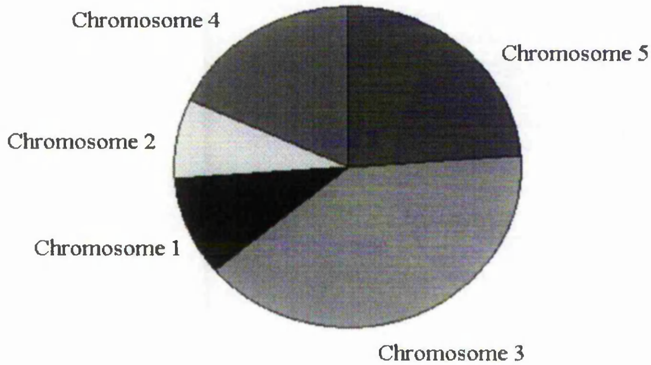


Figure A.3. Parent Selection Roulette Wheel.

As with nature, random factors are introduced into the population in the form of mutations. The purpose of the mutation is to increase the search area and prevent local optimisation by introducing a genome comprising of random values. The mutation will only aid the search, due to the fitness functions. If the information held within the genome does not produce a high fitness value, it is probable that the genes contained within it will be removed from the population within a few generations. The introduction of the mutated genome is probabilistic, generally between 10 to 1 % according to Goldberg (1990). The setting of the level is not quantifiable due to the random nature of the search and is set through trial and error and dependant upon the method of limiting the number of generations.

Another factor of the GA that has a dramatic effect upon the optimisation route and success is the size of the generation. The size of the population represents the number of search cases that are being performed with combinations of parameter values. Therefore, the larger the population the more comprehensive the cover of the search area. However, there must be a limit to the maximum and minimum sizes of the population. Calculable methods for determining the population size for any application have not been found and are therefore, set through trial and error and with consideration of two factors; computational speed and minimum cover. The maximum size of population has no limit, except for the limitations of the computational speed, the larger the population the more calculations to perform and the longer the convergence, due to the wider spread of initial values. Therefore, minimum cover should be the factor to determine the population size. The minimum cover takes into consideration the number of optimisation parameters and their ranges. The outcome of which, will be a population size that will offer enough variation within the initial genes to adequately cover the search area.

The basic GA process is summarised in Figure A.4, and shows how the above mentioned features combine within an iterative process.

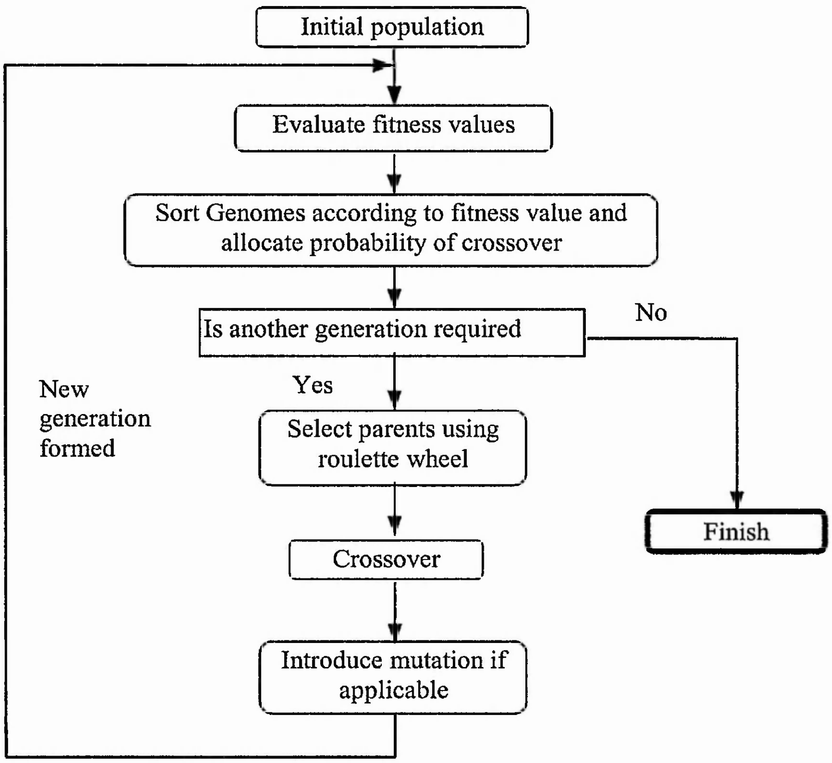


Figure A.4. GA Process

A.2 Artificial Neural Networks

Many types of ANNs have been developed for a variety of applications. Table A.1 below describes some of the more commonly used networks together with their type of typical application. This is not a comprehensive list but does include the ANNs considered for application to this project.

Network Type	Description	Application
Perceptron	The perceptron comprises of a single element, which is capable of simple classification. Due to the limited capabilities of the Perceptron can only recognise exact training patterns.	Generally not used but can perform limited simple classification
Multi-layer Perceptron	The multi-layer perceptron network comprises of one or more layers of perceptron elements between the input and output. The extra layer increases connectivity of the network enabling multiple classification. Training is generally performed using the backpropagation technique which determines the error between the training pattern and the output and propagates the error back through the network, modifying the connective weights .	Network is capable of classification and prediction for both continuous and discrete representations
Hopfield	The Hopfield model is used as an auto-associative memory to store and recalling images. Images are stored by calculating a corresponding weight matrix. Thereafter, starting from an arbitrary configuration, the memory will settle on exactly that stored image, which is nearest to the starting configuration in terms of Hamming distance. Thus given an incomplete or corrupted version of a stored image, the network is able to recall the corresponding original image.	Most appropriate when exact binary representations are possible.
Blotzmann Machine	The Boltzmann machine is a stochastic version of the Hopfield model, whose network dynamics incorporate a random component in correspondence with a given finite temperature. Starting with a high temperature and gradually cooling down, allowing the network to reach equilibrium at any step, chances are good, that the network will settle in a global minimum of the corresponding energy function. This process is called simulated annealing.	Displays many of the qualities of the Hopfield network and additionally can have hidden units, which allow, given enough units, to learn arbitrary functions

Kohonen	The Kohonen network is a self-organising feature map comprising of two layers, an input and output. The output layer is normally arranged in a two-dimensional array with each element connected to all the inputs. The feature mapping can be thought of as a nonlinear projection of the input pattern space on the neurons array that represents features of the training data.	Network is suitable for unsupervised clustering of data and conversion of data from high to low dimensional space.
ART1	This paradigm has the ability to plastically adapt when presented with new input patterns, while remaining stable at previously seen input patterns.	Best suited to the classification of binary representations, not requiring an omnipotent trainer
ART2a	Similar to the ART1 network except that it has the capacity of handling continuous data beyond the range of 0 to 1.	Extensively used in the field of vision recognition and clasification

Table A.1 Artificial Neural Networks

A.3 Multi-Layer Perceptron and Backpropagation

The multi-layer perceptron (MLP) is an advance on the perceptron as described by Minsky and Papert in Perceptrons (). To explain the MLP it is first necessary to explain the single perceptron. The perceptron is a single artificial neuron, which can be likened to the basic biological neuron. Figure A.5 shows the similarities between the biological and artificial neurons.

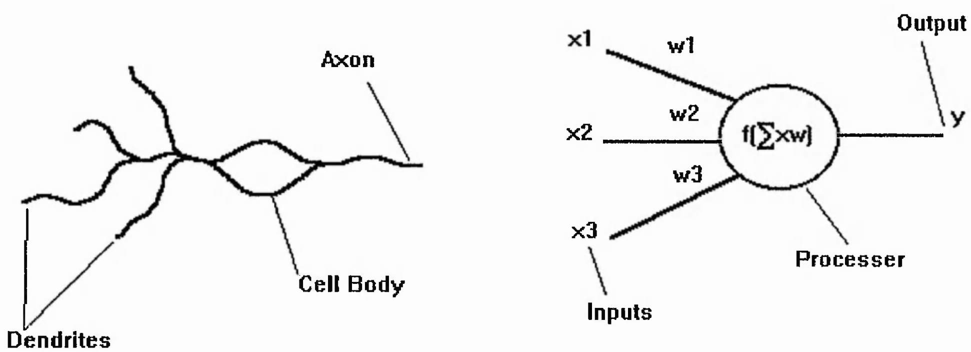


Figure A.5 Biological and Mathematical Neurons

A simplified explanation of how a biological neuron functions is as follows. The biological neuron consists of dendrites, a cell body and an axon. Synaptic connections at the ends if the dendrites have varying chemical strengths, that are summed together at the base of the dendrite to form the input to the cell body. The cell body in turn activates producing an output dependant upon the input that is transmitted out along the axon.

The artificial neuron simulates this procedure mathematically. Inputs enter the artificial neuron via connections and each connection carries a multiplying factor, called a weight, (the weight may be likened to the synaptic strength). The product of the input and the weight forms the input for that connection and the sum of all the connections inputs form the input to the artificial cell body. A transfer function is performed upon the input to obtain the output from the perceptron.

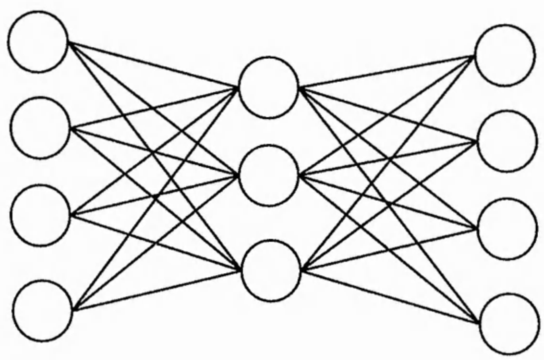


Figure A.6 Multi-layer Perceptron and Connections

The MLP, as demonstrated in Figure A.6, is a artificial neural network consisting of multiple perceptrons grouped into layers, with each layer connected to the next. The output of each perceptron is connected to every element in the next layer, therefore, the outputs from perceptrons in one layer become the input to the next. For multiple layers there are three types of layer; the input, hidden and output layers. The input layer is as it suggests, the layer where the external input is presented to the network. Likewise the output layer is where the networks output is produced. The hidden layers add extra connectivity to the network, increasing its capabilities and performance.

The connection weights between the perceptrons contain the networks information and characteristics. Therefore, adjustment of the weights will alter the information and characteristics contained. Adjustment of the weights so that the network performs a certain task is called training. The training process for the MLP is the back-propagation technique.

Backpropagation

The invention of the Backpropagation (BP) technique is associated with Rumelhart () and also with Parker () who introduced a similar algorithm at the same time. Backpropagation, one of the training techniques used to alter the connection weights in a feed forward network is used with the MLP. The BP technique is a supervised process which adjust the weights to obtain the desired output from a given input.

The input is propagated through the network as a function of the summed product of the inputs and weights.

$$x_j^{[s]} = f\sum_i w_{ji}^{[s]} x_i^{[s-1]}$$
$$= f\left(I_j^{[s]}\right)$$

where :

$x_j^{[s]}$ = current output state of j^{th} neuron in layer[s]

$w_{ji}^{[s]}$ = weight of connection joining i^{th} neuron in layer [s-1] to j^{th} neuron in layer [s]

$I_j^{[s]}$ = weighted summation of inputs to j^{th} neuron in layer [s]

Back-propagation of the error.

It is highly improbable that the network will obtain the desired output from the initial connection weights. Therefore, an error will exist. Suppose that the network has a global error function, E, associated with it. This global error comprises of many local errors which are directly dependant upon the connection weights and outputs from the local neuron. Suppose that the relationship is differentiable, the critical parameter of the function that is propagated back through the layers of the network is defined by $e_j^{[s]} = \dfrac{\partial E}{\partial I^{[s]}}$ which can be considered as a measure of the local error at neuron j in level s.

The association between the local error and a particular error in the next layer, toward the output layer, is the important component in the training process. The association is given by equation A which is the solution to the relation in equation A.1

$$e_j^{[s]} = f'\left(I_j^{[s]}\right)\sum_k\left(e_k^{[s=1]} w_{kj}^{[s+1]}\right) \qquad \text{equation A.1}$$

This indicates that the transfer function used for the network must also be included in the error equations. equations A.2 and A.3 are the local error equations for a sigmoid and hyperbolic tangent functions respectively.

sigmoid function $f\left(I_j^{[s]}\right)=\dfrac{1}{1+e^{\left(-I_j^{[s]}\right)}}$

$$e_j^{[s]}=x_j^{[s]}\left(1-x_j^{[s]}\right)\sum_k\left(e_k^{[s=1]}w_{kj}^{[s+1]}\right) \qquad \text{equation A.2}$$

tanh function $f\left(I_j^{[s]}\right)=\dfrac{e^{I_j^{[s]}}-e^{-I_j^{[s]}}}{e^{I_j^{[s]}}+e^{-I_j^{[s]}}}$

$$e_j^{[s]}=\left(1+x_j^{[s]}\right)\left(1-x_j^{[s]}\right)\sum_k\left(e_k^{[s=1]}w_{kj}^{[s+1]}\right) \qquad \text{equation A.3}$$

As there are no elements above the output layer the effect of the higher levels, $\sum_k\left(e_k^{[s=1]}w_{kj}^{[s+1]}\right)$,is replaced by the difference between the target and actual output from the network $\left(d_i-y_i\right)$. Equations A.4 and A.5 become:

$$e_j^{[s]}=x_j^{[s]}\left(1-x_j^{[s]}\right)\left(d_i-y_i\right) \qquad \text{equation A.4}$$

$$e_j^{[s]}=\left(1+x_j^{[s]}\right)\left(1-x_j^{[s]}\right)\left(d_i-y_i\right) \qquad \text{equation A.5}$$

Any transfer function producing a steady curved output can be used within the network. The sigmoid produces an output between 0 and 1, while tanh forms a similar curve between 1 and –1. Figure A.7 illustrates the functions.

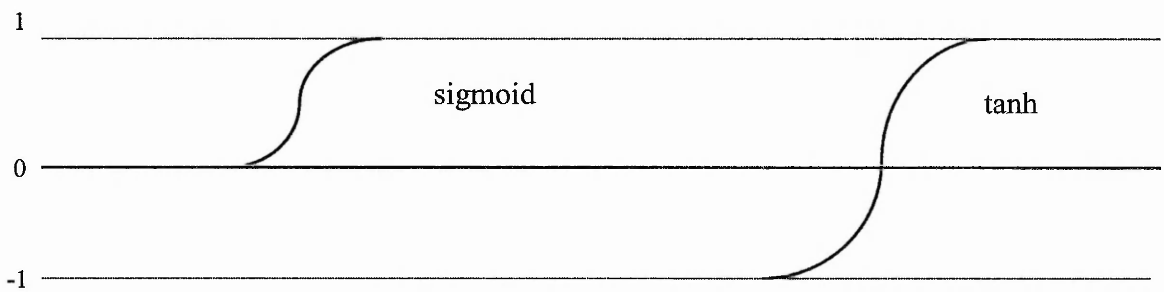


Figure A.7. Transfer Functions

To train the network to produce the desired outputs for the input, the global error of the network must be reduced. As the error is associated to the connection weights they should be modified to reduce the global error. The adjustment of the weights will be based upon the knowledge of the local errors which may be obtained from equations A.4 to A.5 above.

The amount the weights are to increment or decrement is governed according to the local errors using the gradient decent rule as follows :

$$\Delta w_{ji}^{[s]} = -lcoef\left(\frac{\partial E}{\partial w_{ji}^{[s]}}\right)$$

where $\frac{\partial E}{\partial w_{ji}^{[s]}} = -e_j^{[s]}x^{[s-1]}$

$$\therefore \Delta w_{ji}^{[s]} = lcoef.e_j^{[s]}x_i^{[s-1]} \qquad \text{equation A.7}$$

The term *lcoef* is the learning coefficient, which limits the gradient of the error decent. Adjustment of the learning coefficient throughout training may prevent resonance and settling for a local instead of global minimum.

An additional variation to the standard algorithm is the introduction of a momentum term. A problem of the delta weight equation is that it assumes that the adjustment of the error is linear. The momentum is a proportion of the previous delta weight, which is transferred through to the present. The momentum term allows for a small learning coefficient while not increasing the training period dramatically. Introduction of the momentum term alters equation A.7 to :

$$\Delta w_{ji}^{[s]}(t) = lcoef.e_j^{[s]}x_i^{[s-1]} + momentum\Delta w_{ji}^{[s]}(t-1) \qquad \text{equation A.8}$$

where t indicates the time period.

The new value of the connection weight becomes equation A.9:

$$w_{ji}^{[s]}(t) = w_{ji}^{[s]}(t-1) + \Delta w_{ji}^{[s]}(t) \qquad \text{equation A.9}$$

APPENDIX B

B.1 Gear Geometry Calculations for GA Optimisation of Spur and Helical Gears

Addendum Modification Coefficient, *x*

$1 \leq x \leq x_{min}$

Upper limit, x_{max} 1.0 is the maximum, generating pointed teeth. Excess of 1 will cause failure of the tip geometry.

Lower Limit, x_{min} The lower limit is determined by equation B.1 obtained from MAAG (1990). The limit avoids cutter interference and has been altered to include the allowance for lengthened teeth.

$$x_{\min} = \frac{z}{z_g} \qquad \text{equation B.1}$$

where: $$z_g = \frac{2 \cdot \text{Addendum Coeficient}}{\sin^2 \alpha}$$

Rack Tip Fillet Radius, ρ_{fP}

A maximum limit is applied to the rack tip fillet radius, preventing a fillet radius greater that a full fillet radius being generated. The limit is defined in equation B.2, obtained from MAAG (1990).

$$\rho_{fP} \leq \frac{C_P}{1 - \sin\alpha} \qquad \text{equation B.2}$$

where C_P is taken as 0.25 (typical standard value)

Reference Diameter, *d*

$$d = \frac{z \cdot m}{\cos\beta} \qquad \text{equation B.3}$$

Addendum (without profile shift), h_a

$$h_a = m \cdot \text{Addendum Coefficient} \qquad \text{equation B.4}$$

Tooth Addendum (including profile shift)), h_a^*

$$h_a^* = m(\text{Addendum Coeficient} + x) \qquad \text{equation B.5}$$

Tangential Tip Diameter, d_a

$$d_a = \frac{d + 2(h_a + xm)}{\cos\beta} \qquad \text{equation B.6}$$

Tangential Tooth Depth, h

Derived from Figure B.1 $\quad h = \dfrac{h_a + h_f}{\cos\beta}$

Where h_f is the dedendum of the tooth. To allow meshing with the next gear the, length of the dedendum equals the addendum + clearance.

Therefore $\quad h\dfrac{2h_a + C_P}{\cos\beta} \qquad$ equation B.7

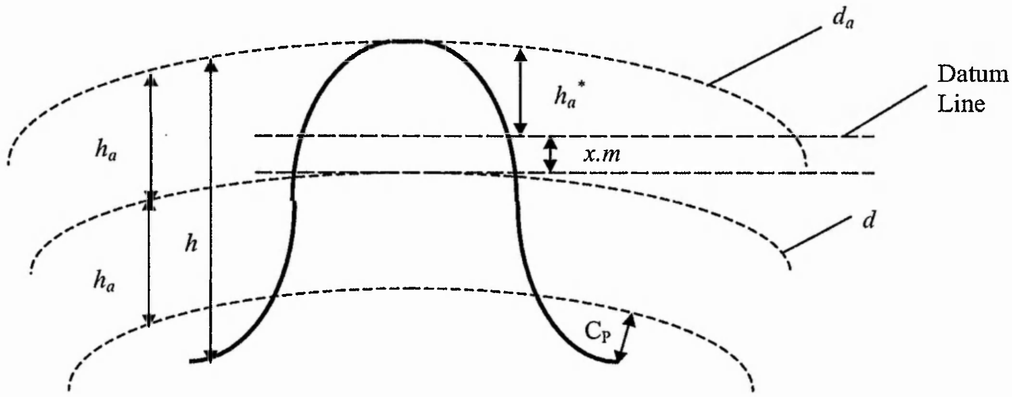


Figure B.1 Basic Tooth Geometry

Calculation of number of teeth on wheel, (Variable centre distance only)

To maintain speed ratio the ratio of the number of teeth must remain constant. Therefore:

number of teeth on wheel gear = number of teeth on pinion * speed ratio.

or

$$z_2 = z_1 \cdot ratio \qquad \text{equation B.8}$$

(z_2 is rounded to the nearest integer).

N.B: Profile shift and length of teeth have no effect on the speed ratio

Calculation of teeth and profile shift for fixed centre distance

The fixed centre distance and speed ratio are to be maintained throughout the calculation. The number of teeth for the pinion and wheel gears are determined, based upon the tangential diameters of the gears. The difference in centre distance is then compensated for with the profile shift.

Pinion Gear Teeth

$$\text{Centre Distance, } a = \frac{\text{Tangential Reference Pinion Diameter} + \text{Tangential Reference Wheel Diameter}}{2}$$

$$a = \frac{z_1 m + z_2 m}{2\cos\beta} \qquad \text{equation B.9}$$

$$a = \frac{m z_1 (1 + ratio)}{2\cos\beta}$$

therefore

$$z_1 = \frac{2\cos\beta \cdot a}{(1 + ratio)} \qquad \text{equation B.10}$$

Round to nearest integer.

Wheel Teeth

From equation B.9 $\quad a = \dfrac{m(z_1 + z_2)}{2\cos\beta}$

Therefore

$$z_2 = \left(\frac{2\cos\beta \cdot a}{m}\right) - z_1 \qquad \text{equation B.11}$$

Round to nearest integer.

Difference in Centre Distance and Profile Shift of Wheel

A difference in centre distance will require modification of the teeth profile to compensate. First the difference in centre distance, *DCD,* must be found.

$$DCD = a - \text{Centre distance from } z_1 \text{ and } z_2 \text{ without profile shift}$$

$$DCD = a - \left(\frac{z_1 m + z_2 m}{2\cos\beta}\right) \qquad \text{equation B.12}$$

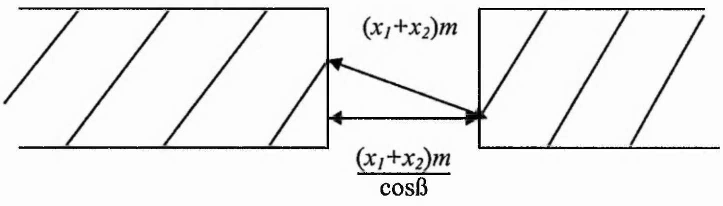


Figure B.2 Difference in Diameter Compensated for by Profile Shift

Applying the profile shifts to equation B.12 as shown in figure B.2 above the difference in centre distance can be reduced to zero. The value of x_1 will be taken from the GA Genome allowing x_2 to be calculated.

$$DCD = a - \left(\frac{(z_1 + 2x_1)m + (z_2 + 2x_2)m}{2\cos\beta} \right)$$

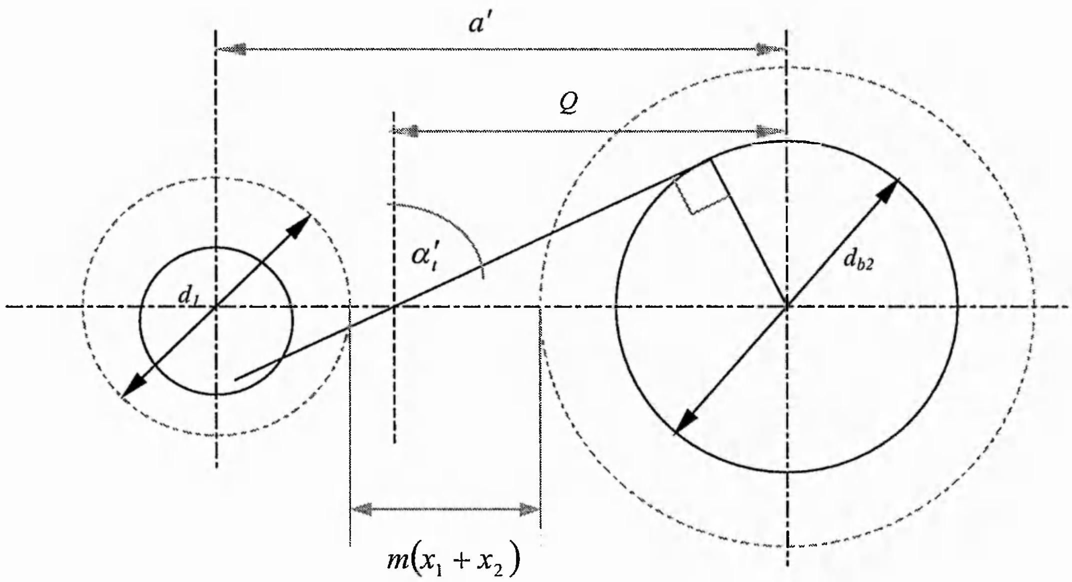
Therefore rearranging
Therefore rearranging

$$x_2 = \frac{\left(\frac{2}{m}(a - DCD) \right) - (z_1 + z_2) - 2x_1}{2} \qquad \text{equation B.13}$$

The conditions of $x_1 + x_2 = 0$ and $x_1 + x_2 \neq 0$ can apply depending on the value of x_1 that is received from the Genome and the difference in centre distance.

Proof of modified Working Transverse Pressure Angle, α'_t allowing for Profile Shift

Modification of transverse working pressure angle, α_t' to account for profile shift. This modification is based upon derivation of α_t' from MAAG. The formula derived in MAAG (1990) does not allow for the variation to the centre distance.



Key: a' modified centre distance; d reference diameter; d_b base diameter

Figure B.3. Geometry of Tooth Profile Shift with Respect to Centre Distance

$$a' = Y(d_1 + d_2)$$

where Y is the centre distance modification which takes into account the profile shift.

$$Y = \frac{a'}{(d_1 + d_2)}$$

length of $Q = Yd_2 = \dfrac{a'.d_2}{(d_1 + d_2)}$

$\therefore$ from trigonometry $\qquad \dfrac{\text{O}}{\text{H}} = \sin(\text{x})$

$$\sin\left(90 - \alpha_t'\right) = \frac{d_2}{2} \cdot \frac{1}{Q} = \frac{d_2}{2} \cdot \frac{(d_1 + d_2)}{a'.d_2}$$

$$\sin(A - B) = \sin A \cos B - \cos A \sin B$$

$$\sin 90 = 1 \text{ and } \cos 90 = 0$$

$$\sin\left(90 - \alpha_t'\right) = \cos \alpha_t'$$

thus $\qquad \alpha_t' = a\cos\left(\dfrac{d_2}{2} \cdot \dfrac{(d_1 + d_2)}{a'.d_2}\right)$ equation B.14

Contact Ratio

Total Contact Ratio, ε_γ $\qquad \varepsilon_\gamma = \varepsilon_\alpha + \varepsilon_\beta$ equation B.15

Transverse Contact Ratio, $\varepsilon\alpha$ $\quad \varepsilon_\alpha = \varepsilon_1 + \varepsilon_2$ equation B.16

Addendum Contact Ratio of Pinion, ε_1 $\qquad \varepsilon_1 = \dfrac{z_1}{2\pi}\left(\tan \alpha_{a1} - \tan \alpha_t'\right)$ equation B.17

Addendum Contact Ratio of Wheel, ε_2, uses equation B.17 except with wheel references.

Overlap Ratio, $\varepsilon\beta$ $\qquad \varepsilon_\beta = \dfrac{b \cdot \sin \beta}{p}$ (only affects helical gears) equation B.18

Interference

Contact beyond the line of action (point of tangency of the line of action with the base circle of the mating gear). For the tip[of the wheel this means:

$$g_{a2} \le a \cdot \sin \alpha_t' \qquad \text{equation B.19}$$

where $$g_{a2} = \frac{d_{b2}}{2} \tan\alpha_{a2}$$

Contact parameter for the tip of the pinion tooth at the end of its path of contact,

$$k_E = \frac{u+1}{u}\left(1 - \frac{\tan\alpha_t'}{\tan\alpha_{a1}}\right) \quad \text{equation B.20}$$

Contact parameter for the tip of the wheel tooth at the end of its path of contact,

$$k_A = (u+1)\left(1 - \frac{\tan\alpha_t'}{\tan\alpha_{a2}}\right) \quad \text{equation B.21}$$

To avoid interference the values of both k_E and k_A must be <1

B.2 Technical Information for Initial Gear Design

Ratio Split

The ratio to be covered by the gear train with two of more stages requires to be split. This split is not equal, as the ration is smaller for the lower stages. This is to keep the gear sizes to a minimum due to lower forces. This ratio split has been defined by Nieumann (1978) as:-

for two stage

$u_1 = 0.8(u)^{2/3}$

for three stage

$u_1 = 0.6(u)^{4/7}$

$u_2 = 1.1(u)^{2/7}$

Where u = overall ratio

u_1= ratio in first stage

u_2= ratio in second stage

Actual contact stress

This forms the basis for the gear design as the gear dimensions and qualities are determined to ensure that the gear performs at the maximum permissible stress, therefore optimising the design while working within safety limits. The actual contact stress as defined by BS 436 part 3 is in equation B.15 and B.16

$$\sigma_H = \frac{Z_N \sigma_{H\lim}}{S_{H\min}} \quad \text{equation B.22}$$

$$\sigma_H = Z_H Z_E Z_\varepsilon \sqrt{\frac{F_{Ht}}{bd} * \frac{(u+1)}{u} * K_A K_v K_{H\alpha} K_{H\beta}}$$ equation B.23

where $F_{Ht} = \dfrac{2000 T_H}{d}$

The diameter of the gear is one of the main design unknowns and has a massive effect on the other features of the gear Therefore equation B.23 is rearranged to give the diameter.

$$d = Z_H Z_E Z_\varepsilon \sqrt{\frac{2000 T_H}{b \sigma_H^{\ 2}} \left(\frac{u+1}{u} \right) K_A K_v K_{H\alpha} K_{H\beta}}$$ equation B.24

The width of the gear is another factor that is unknown, but from the mounting and the material is ratio to the diameter can be determined from BS 436 where they have been experimentally determined. Therefore the diameter can how be determined by equation B.25

$c = \dfrac{b}{d}$ therefore

$$d = Z_H Z_E Z_\varepsilon \left(\frac{2000 T_H}{c \sigma_H^{\ 2}} \left(\frac{u+1}{u} \right) K_A K_v K_{H\alpha} K_{H\beta} \right)^{\frac{1}{3}}$$ equation B.25

Zone factor for contact stress, Z_H

Accounts for the influence of tooth flank curvature at the pitch point and is calculated from the simplified equation B.26 for standard gears

$$Z_H = 2 \sqrt{\frac{\cos \beta_b}{\sin(2\alpha_t)}}$$ equation B.26

Contact ratio factor, Z_ε

Accounts for load sharing influences and differs for spur and helical gears.

For spur $Z_\varepsilon = \sqrt{\dfrac{4 - \varepsilon_\alpha}{3}}$

For Helical $Z_\varepsilon = \sqrt{\dfrac{1}{\varepsilon_\alpha}}$

Take ε_α as 1.5 for first estimate. This will ensure that undercutting is not necessary.

Facewidth ratio

Facewidth ratio is the ratio of the width of the gear to the diameter, ($c = b/d$) and determine the load distribution across the gear face. This ration can be calculated if the diameter of the gear, power transmitted, stress limit and various factors relating to material properties and application.

Alternatively if the centre distance between shafts is not known c may be determined from the material, location and heat treatment.

Gear Type	Facewidth ratio, c = b/d Gear mounting relative to bearings	
	Symmetric	Asymmetric
Spur and Helical		
Hardness < 180 HB	< 1.0	< 1.0
Hardness > 180 HB	< 1.0	< 1.0
Induction or Case Hardened	< 1.0	< 0.9
Nitrided	< 0.8	< 0.6
Double Helical Gears		
Induction or Case Hardened	< 2.0	< 1.6
Nitrided	< 1.4	< 1.1

Table B.1. Facewidth Ratio

If the calculated facewidth ratio is greater than that in Table B.1 change material to increase strength or increase centre distance.

Safety Factors, S_F and S_H

The safety factors, S_F and S_H, for bending and surface stress respectively are derived from Figure B.4 which gives their values dependant upon the reliability expected and the accuracy of manufacture. As would be expected as either the reliability or accuracy increases the safety factors decrease.

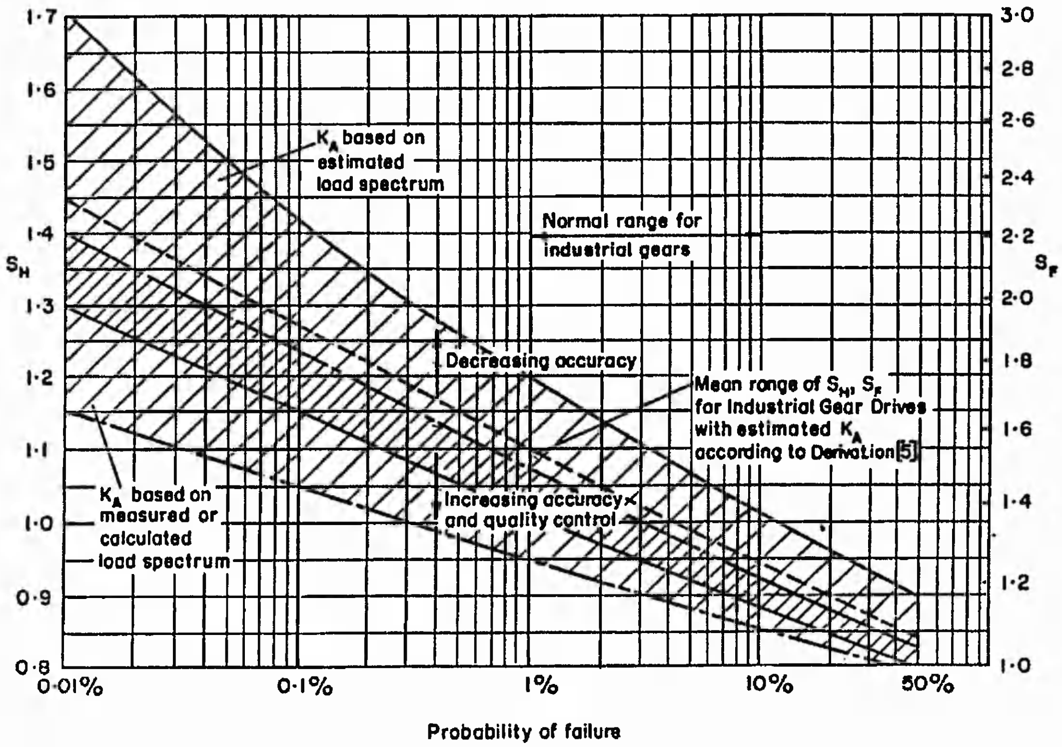


Figure B.4. Safety Factors for Bending and Contact Stress

Gear Accuracy

Gear accuracy is the dimensional and surface accuracy to which the gear is produced. The lower the grade, the smaller the tolerances and therefore lower grades can compensate for tighter safety factors and minimise gear size. However tighter tolerances are more expensive to achieve and the cost to increase accuracy to the next grade is not constant but increases as the grades get lower. The cost of increasing quality from grade 8 to 7 will cost between 2 and 3 %, while increasing from 5 to 4 will cost between 5 and 10 %.

Gear accuracy plays an important and intricate part of the gear design and requires three factors to be taken into consideration when determining its grade, typical attainable accuracy grade, pitch and profile accuracy grade and the lead accuracy grade.

Typical attainable accuracy grade.

These are the accuracy grades attainable from different forms of finish machining processes. Table B.3 gives the accuracy grades as defined by BS 436 part 2.

Finishing Process	Attainable Accuracy											
After Heat treatment	1	2	3	4	5	6	7	8	9	10	11	12
Finish ground												
Finish hard hobbed - skived												
Finish shaved												
Generated gears, hobbed, planed												
Form cut gears												
Blanked, pressed, sintered, injection moulded												
Before Heat Treatment												
Nitrided: *finished ground/shaved*												
Case hardened: *small gears, mass produced, shaved*												
Inductive & Flame hard: *hobbed, etc.*												
Case hardened: *small gears hobbed generated*												
Case hardened: *one off. form cut*												
Induction / Flame spin hard: *generated.*												
Induction / Flame spin hard: *form cut.*												

Table B.3. Attainable Accuracy

Pitch and Profile Accuracy

The accuracy grade relates to the dimensional machining tolerances for the pitch and profile as described in BS 436 part 2. and demonstrated in Figure B.5 for the profile.

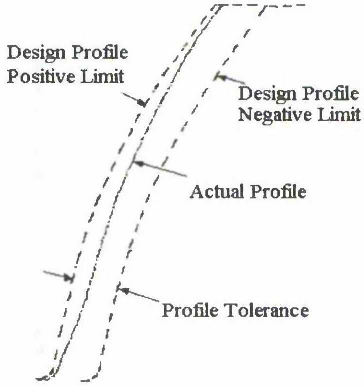


Figure B.5. Profile Accuracy

The accuracy of the pitch and profile is dependant upon the peripheral speed of the gears. The greater the speed the more accurate the gears must be. This relation is given graphically in Figure B.6, but as can be seen may give a range.

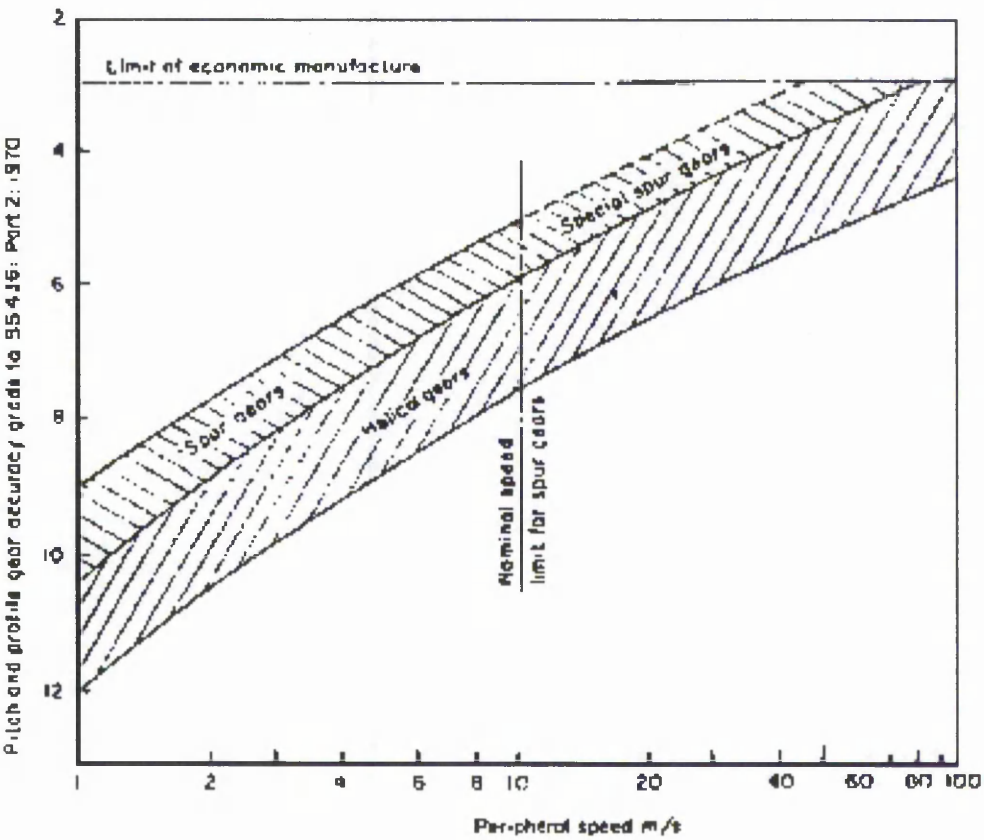


Figure B.6 Pitch Profile

Lead Accuracy

The correct lead accuracy will give an even load distribution across the facewidth of the tooth by ensuring that lead errors are small in comparison to the elastic deformation. Deformation is dependant upon the load applied, material and the module, therefore the lead accuracy will also be dependant upon the facewidth as this disperses the load, material for strength, and the module as this will govern the cross section of the tooth. Figure B.7 below graphically represents the relationship, derived from experimental data.

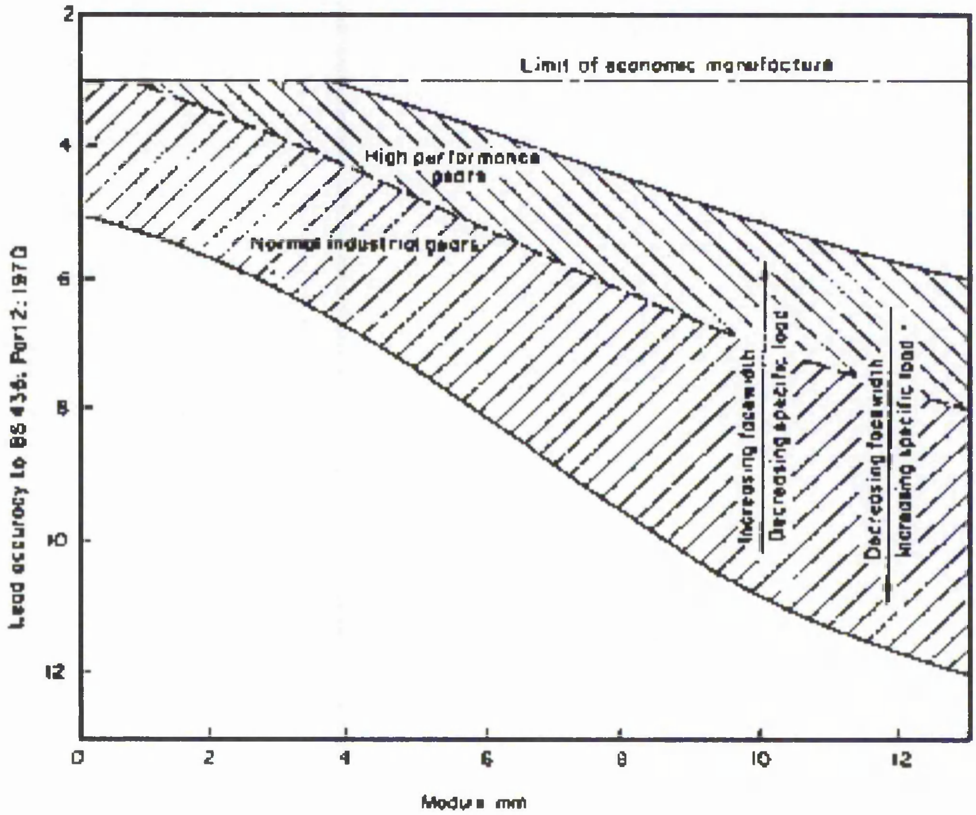


Figure. B.7. Lead Accuracy

The determined values of pitch and profile and lead accuracy grades must fall within the machinable grades obtainable from the manufacturing process. If the highest grade is beyond the manufacturing process modifications to the design are required.

Face Load Factor for Contact Stress (Hertizian Pressure), $K_{H\beta}$

The Face load factor for contact stress, $K_{H\beta}$, accounts for the increase in local load due to distributions of load caused by pinion shaft bending and torsional deflections, misalignment due to manufacturing tolerances.

This safety factor is dependant upon gear accuracy, specific load on tooth and the face width of the tooth, but for preliminary sizing, $K_{H\beta}$ is given as a function of the pinion diameter, gear accuracy and facewidth ratio and graphically represented in Figure B.8. The load has previously been accounted for in the calculation of the pinion diameter.

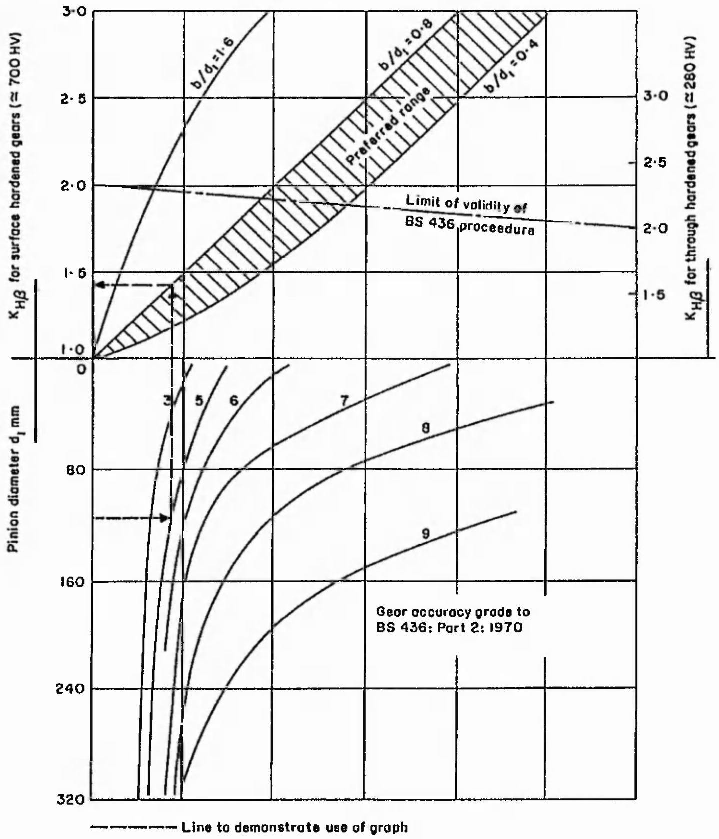


Figure B.8. Face Load Factor for Contact Stress

B.3 Shaft Design

Diameter calculation considering torsion and bending

$$Diameter = \left[\left(\frac{16}{\pi s_s \text{x}10^6}\right)\sqrt{(k_b M_b)^2 + (k_t M_t)^2}\right]^{\frac{1}{3}}$$

where

k_t, torsional safety factor for combined shock and fatigue	=	1.5
k_b, bending safety factor for combined shock and fatigue	=	2.0
s_s, stress for commercial shaft steel with keyway	=	40 MN/m^2
M_t, torsional moment		
M_b, bending moment		

B.4 Bearing Calculation

Bearings are selected depending upon 3 categories: bore diameter, maximum speed and dynamic load rating. The limits of the first two are comparisons between the requirements and the data in the SKF catalogue. The third category is calculated from equation B.27, which is widely used in industry.

$$L_{10} = \left(\frac{C}{P}\right)^p \qquad \text{equation B.27}$$

Where L_{10}	Life of bearing for 90% reliability (million revolutions)
C	Dynamic load rating (N)
P	Equivalent dynamic bearing load (N)
p	exponent of the life equation

Ball Bearing

$p = 3$

$P = F_r$ when $F_a/F_r \le e$

$P = XF_r + YF_a$ when $F_a/F_r > e$

e, X and Y taken from the SKF catalogue and are dependent upon the application, axial load and static load rating.

Angular Contact Ball Bearings

$p = 3$

$P = F_r$ when $F_a/F_r \leq 1.14$

$P = 0.35F_r + 1.57F_a$ when $F_a/F_r > 1.14$

Cylindrical Roller Bearing

$p = 10/3$

$P = F_r$ when $F_a/F_r \leq e$

$P = 0.92F_r + Y \cdot F_a$ when $F_a/F_r > e$

e and Y taken from the SKF catalogue and are dependent upon the application, axial load and static load rating.

Spherical Roller Bearing

$p = 10/3$

$P = F_r + Y_1 F_a$ when $F_a/F_r \leq e$

$P = 0.67F_r + Y_2 \cdot F_a$ when $F_a/F_r > e$

e, Y_1 and Y_2 taken from the SKF catalogue and are specific to each bearing.

Taper Roller Bearing

$p = 10/3$

$P = F_r$ when $F_a/F_r \leq e$

$P = 0.4F_r + Y \cdot F_a$ when $F_a/F_r > e$

e and Y taken from the SKF catalogue and are specific to each bearing.

APPENDIX C

GENETIC ALGORITHM RESULTS

C.1. Methods of Fitness Scaling and their Effect on the Reproductive Selection Process

Percentage Fitness Ratings Direct from Fitness Functions

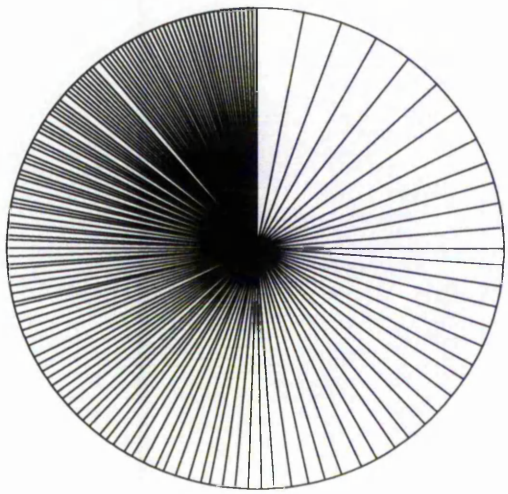


Figure C.1a. 1 Generation. Direct Percentage Fitness

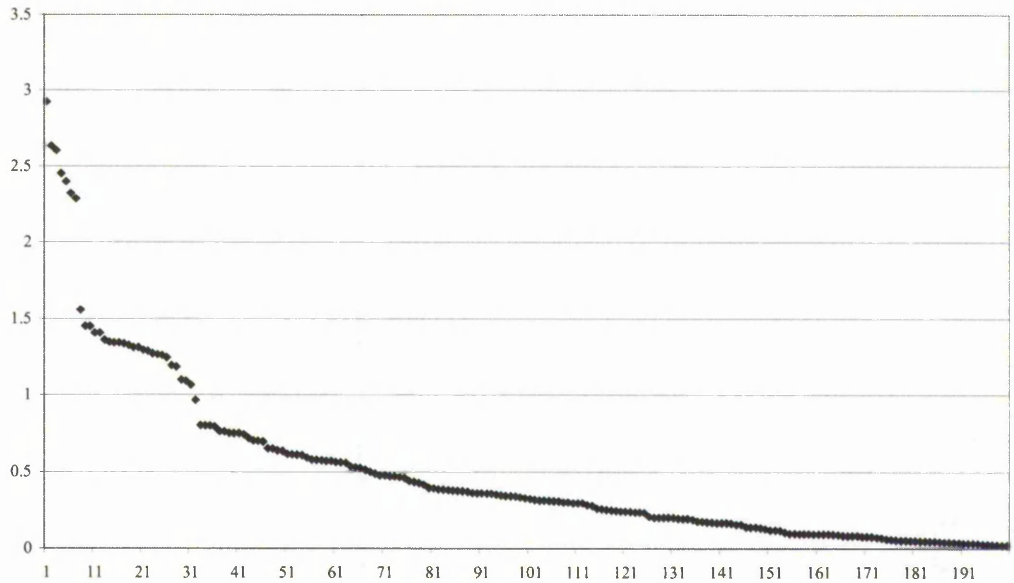


Figure C.1b. 1 Generation, Roulette Wheel. Direct Percentage Fitness Values

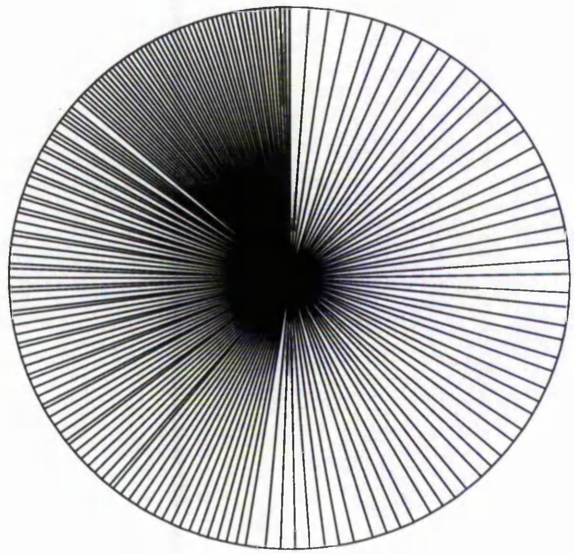


Figure C.2a. 3 Generations, Roulette Wheel. Direct Percentage Fitness

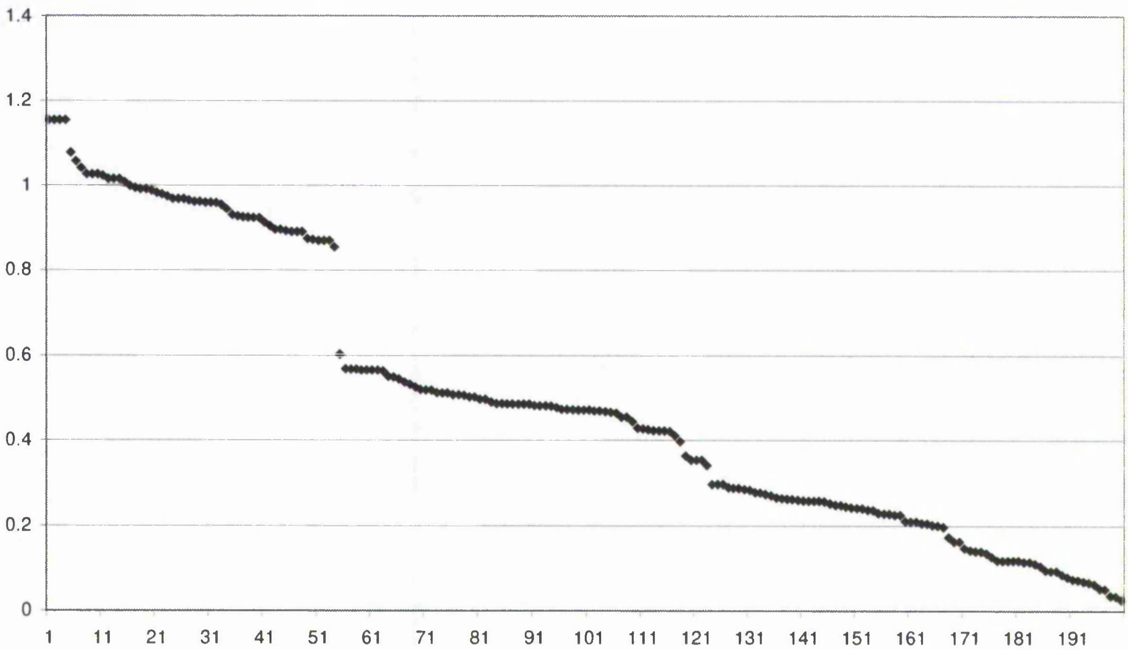


Figure C.2b. 3 Generations. Direct Percentage Fitness Values

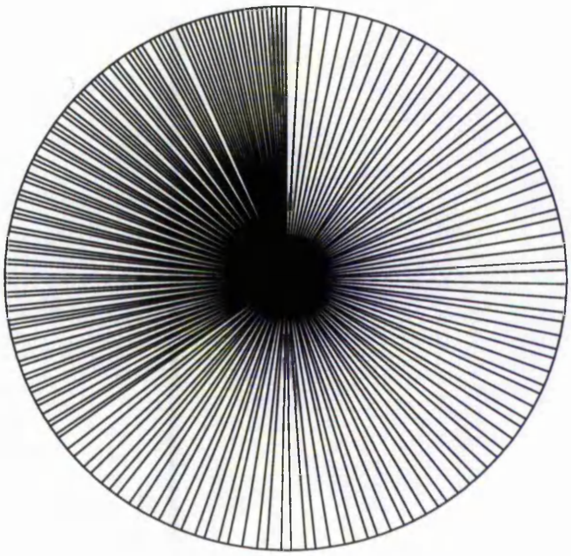


Figure C.3a. 5 Generations, Roulette Wheel. Direct Percentage Fitness

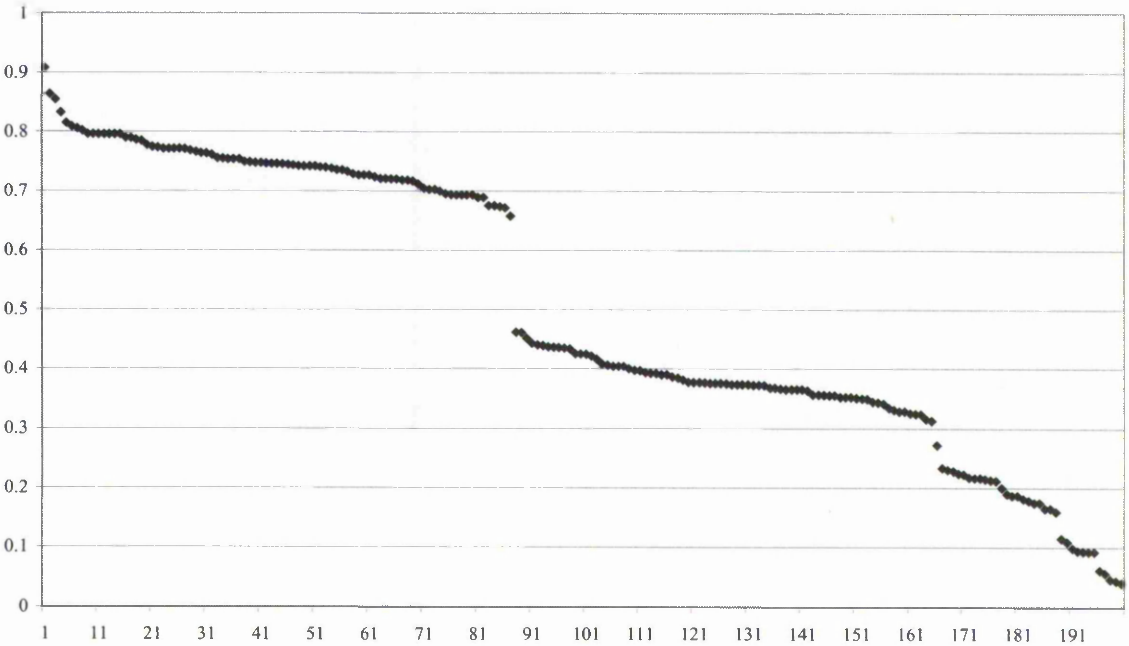


Figure C.3b. 5 Generations. Direct Percentage Fitness Values

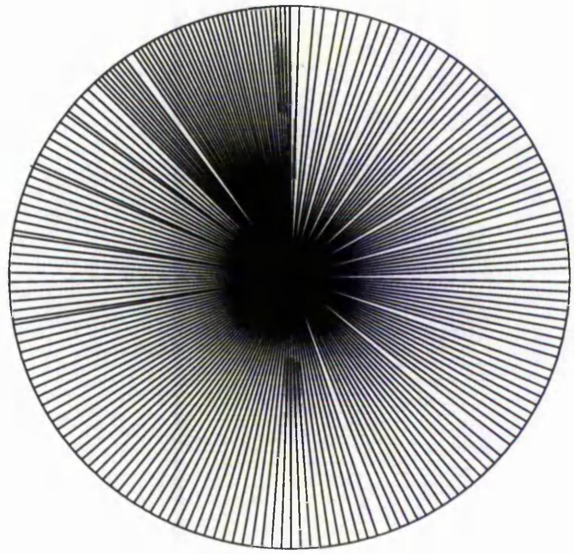


Figure C.4a. 10 Generations, Roulette Wheel. Direct Percentage Fitness

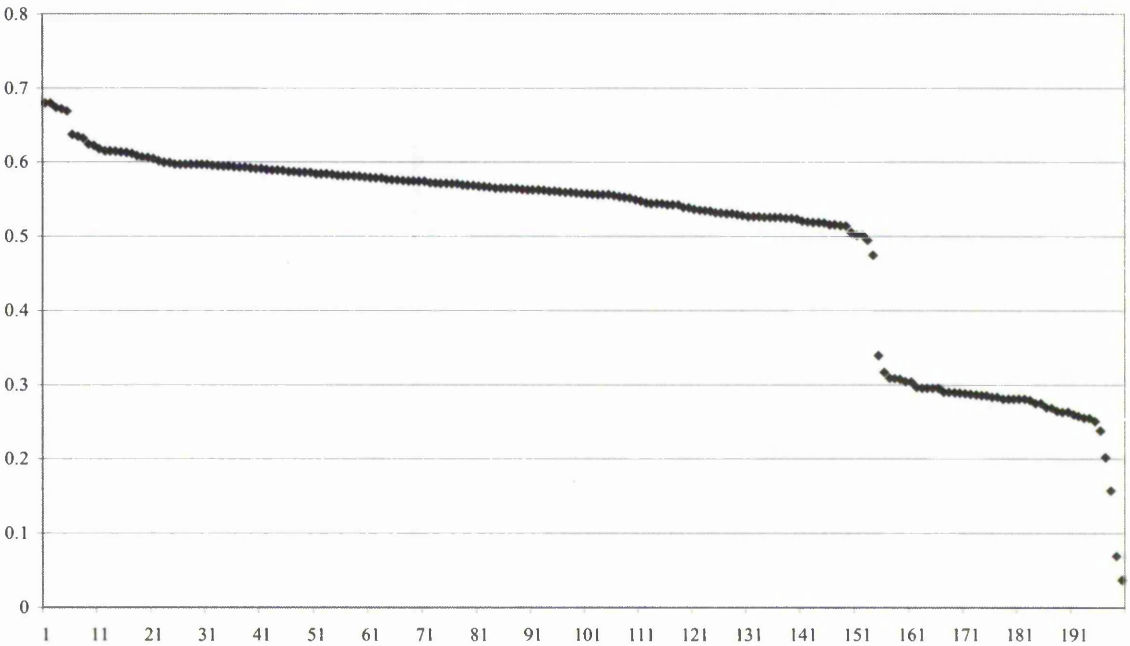


Figure C.4b. 10 Generations. Direct Percentage Fitness Values

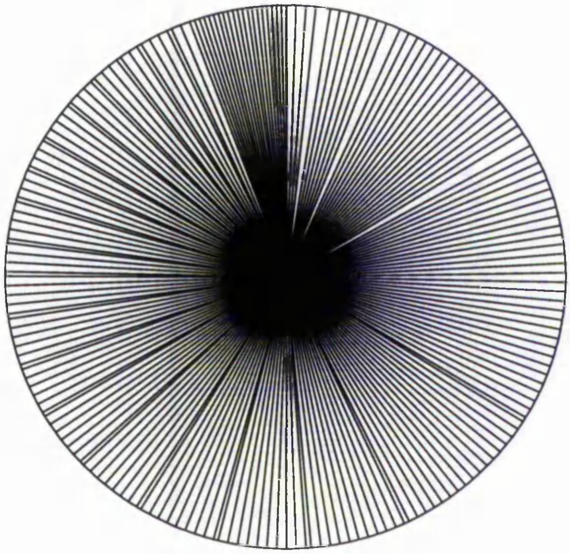


Figure C.5a. 50 Generations, Roulette Wheel. Direct Percentage Fitness

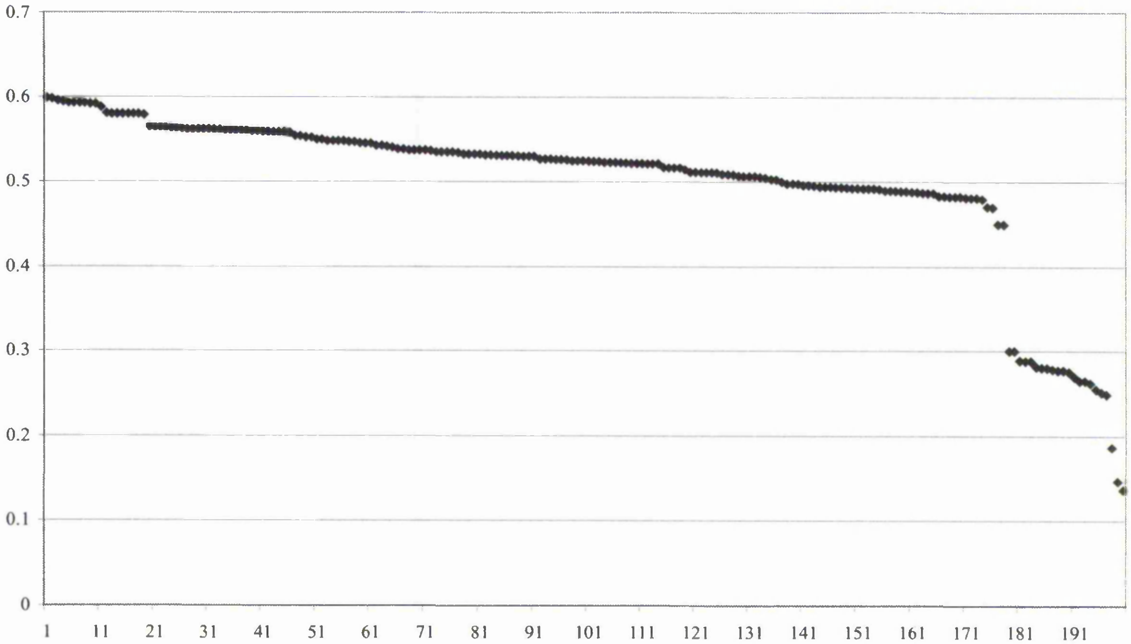


Figure C.5b. 50 Generations. Direct Percentage Fitness Values

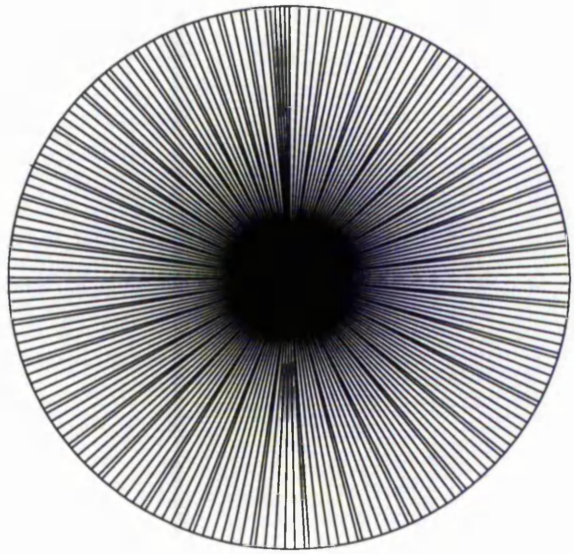


Figure C.6a. 100 Generations, Roulette Wheel. Direct Percentage Fitness

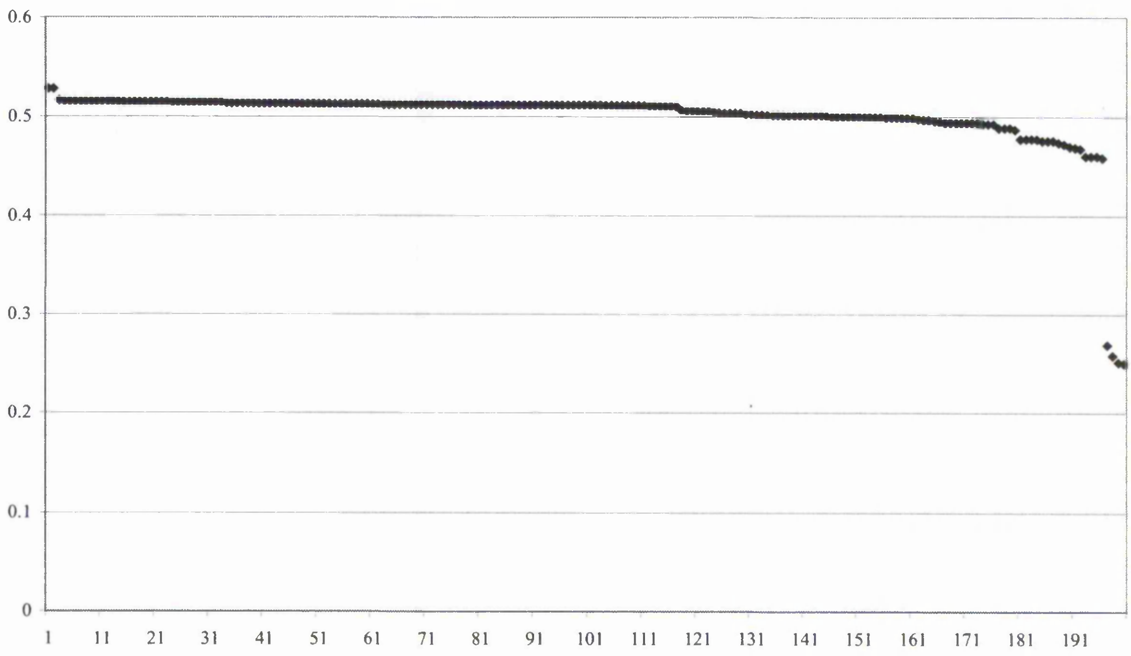


Figure C.6b. 100 Generations. Direct Percentage Fitness Values

Fitness Ratings Applying Linear Ranking Function

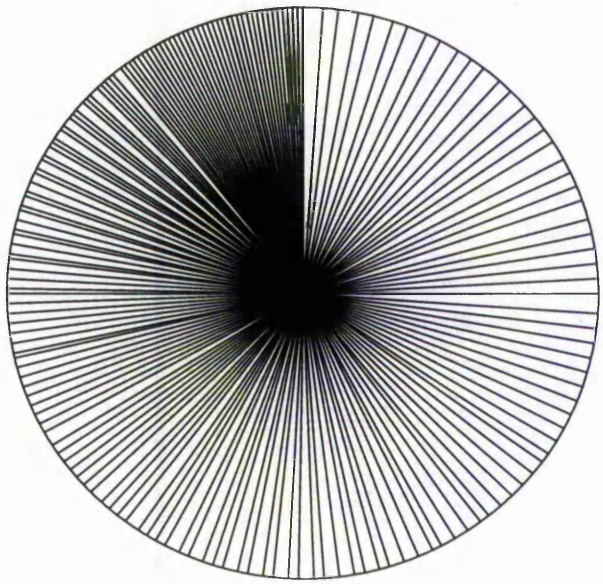


Figure C.7a. 1 Generation, Roulette Wheel. Linear Ranked Fitness

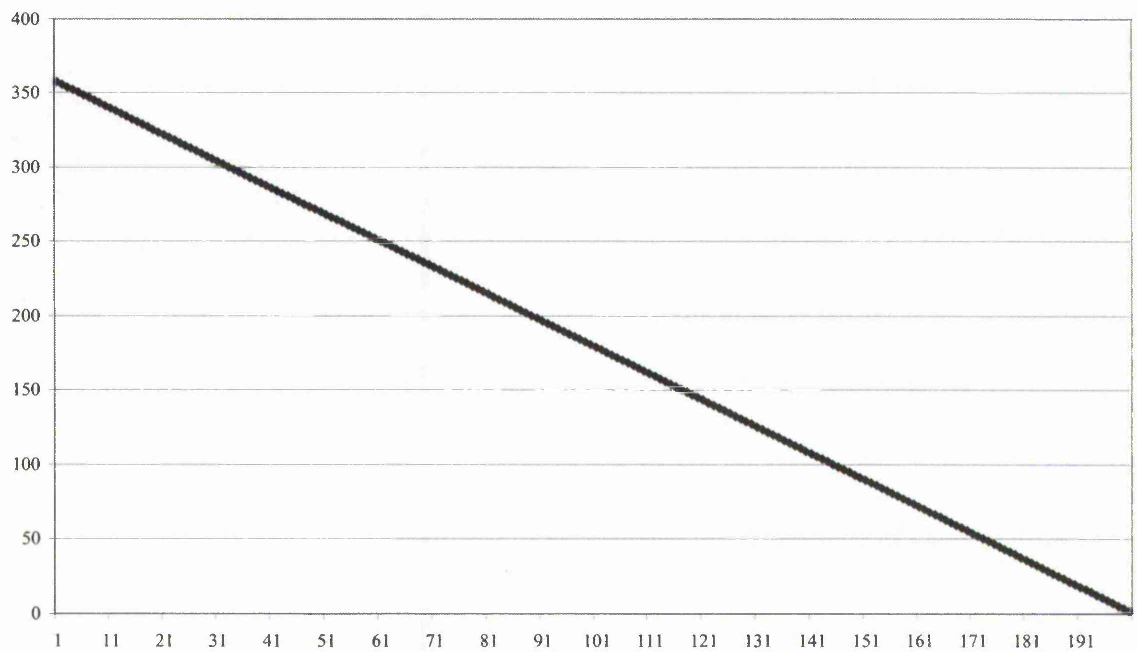


Figure C.7b. 1 Generation. Linear Ranked Fitness Values

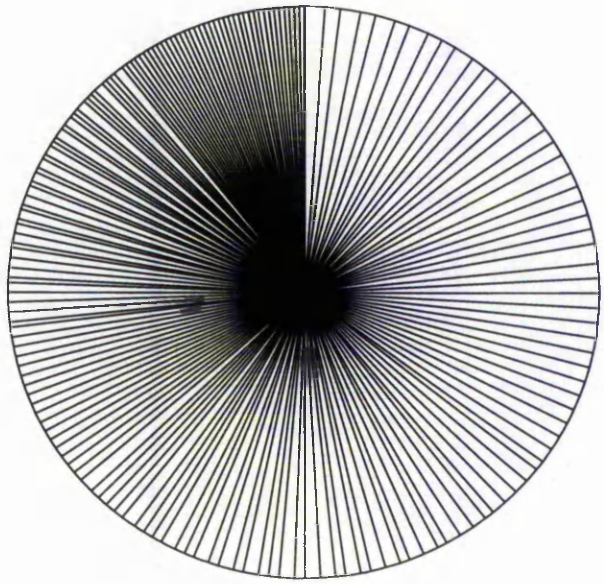


Figure C.8a. 3 Generations, Roulette Wheel. Linear Ranked Fitness

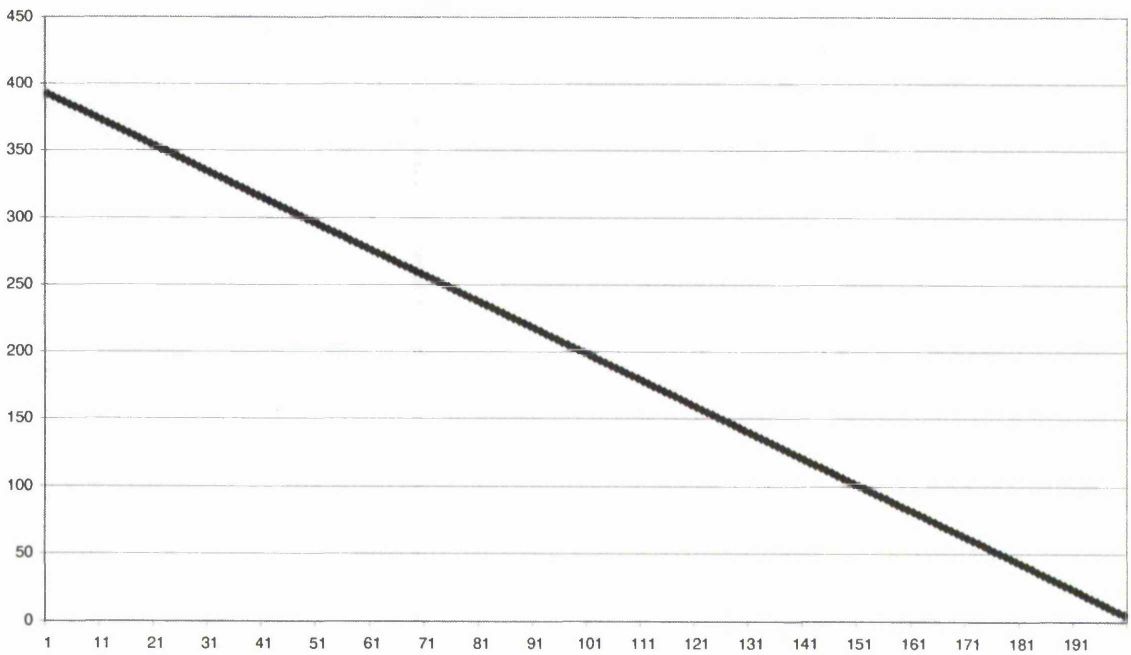


Figure C.8b. 3 Generations. Linear Ranked Fitness Values

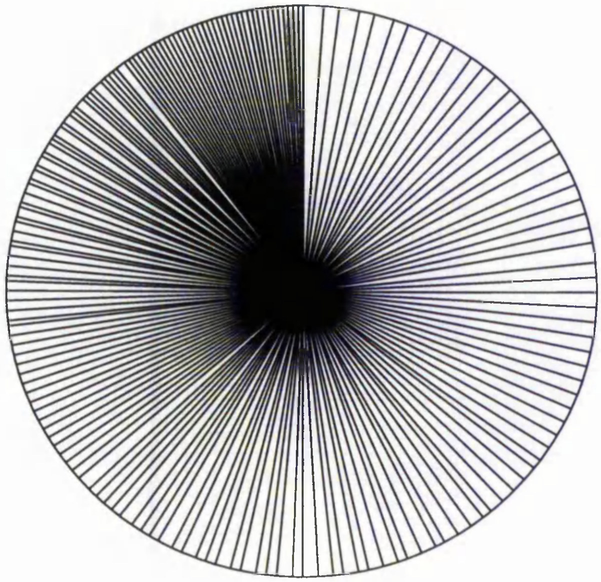


Figure C.9a. 5 Generations, Roulette Wheel. Linear Ranked Fitness

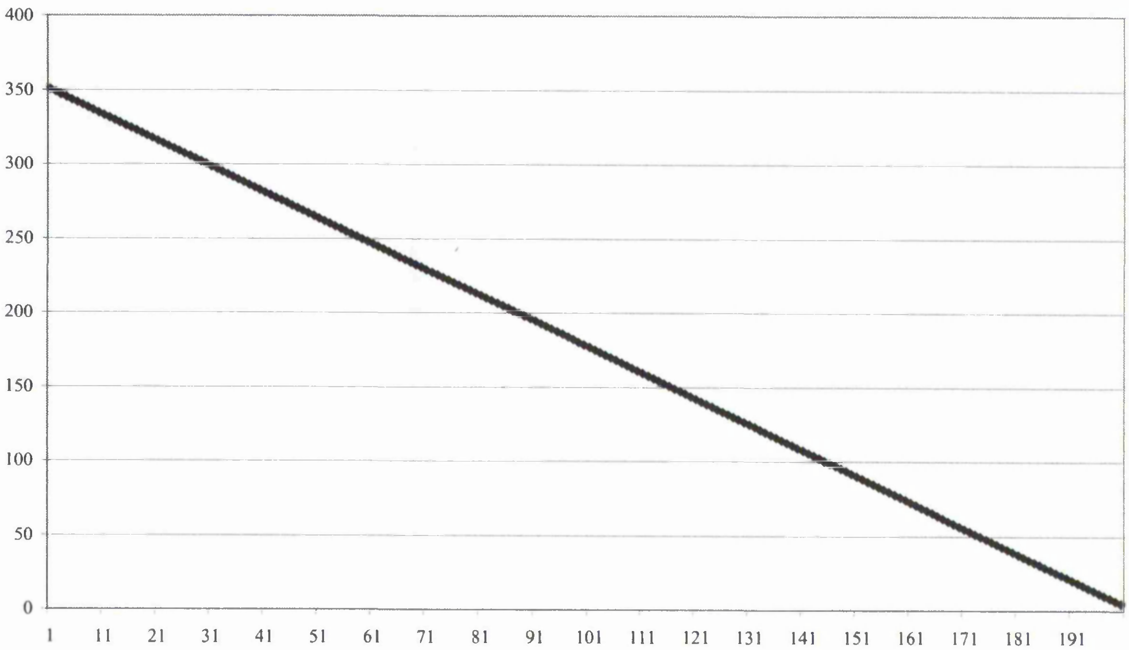


Figure C.9b. 5 Generations. Linear Ranked Fitness Values

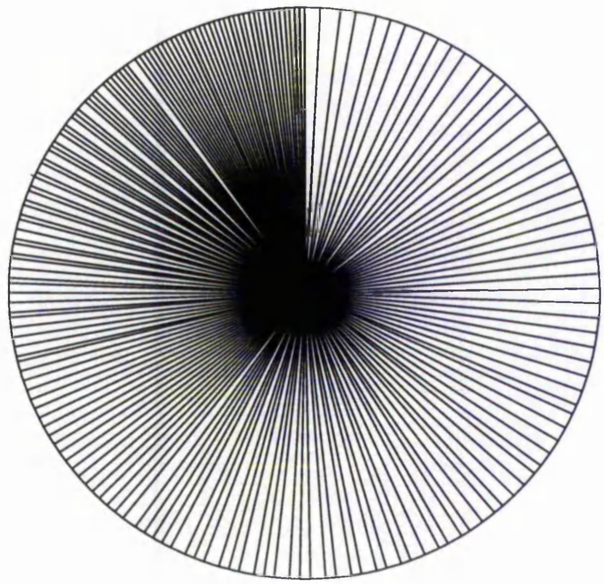


Figure C.10a. 10 Generations, Roulette Wheel. Linear Ranked Fitness

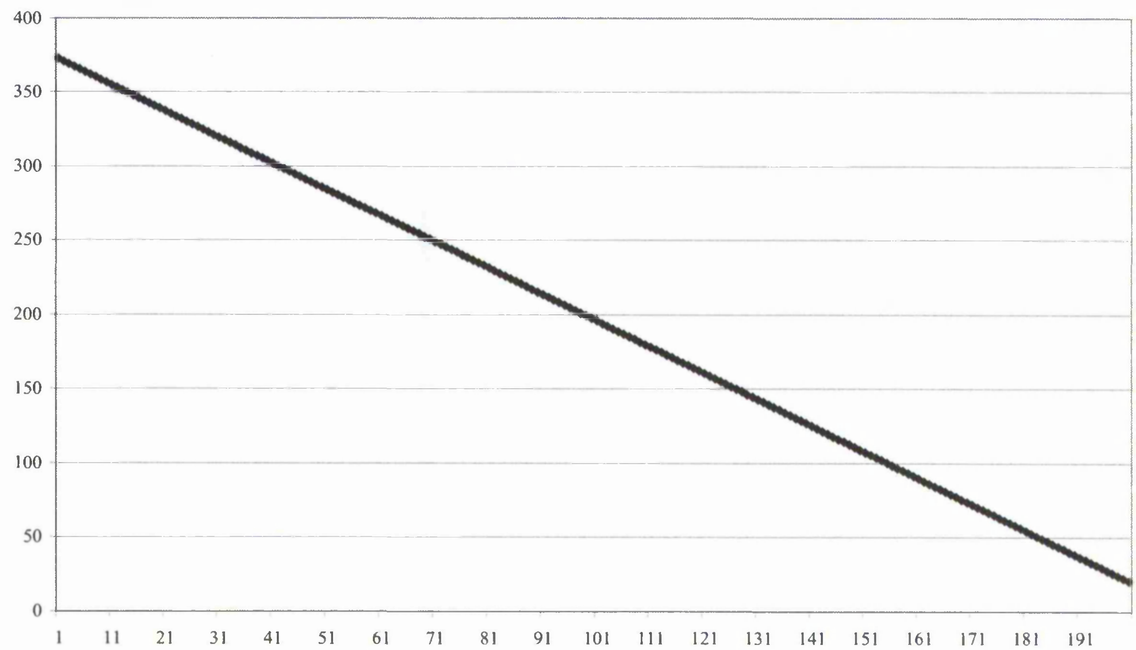


Figure C.10b. 10 Generations. Linear Ranked Fitness Values

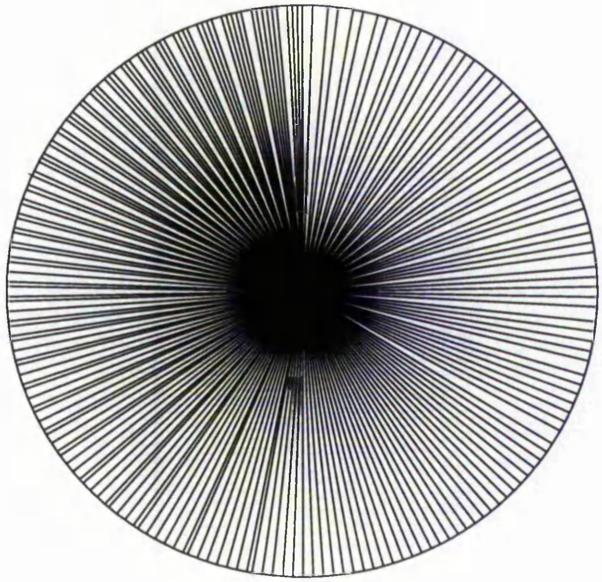


Figure C.11a. 50 Generations, Roulette Wheel. Linear Ranked Fitness

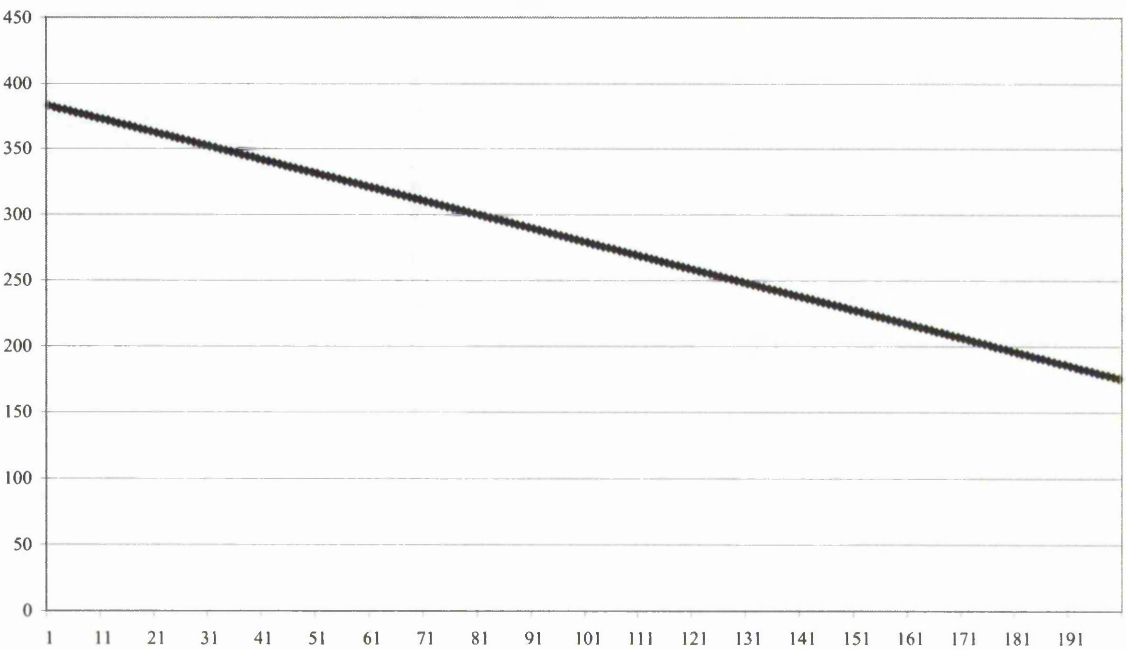


Figure C.12b. 50 Generations. Linear Ranked Fitness Values

C.2 Initial Gear Designs for Optimisation

	Case Study		
	1	**2**	**3**
Power (kW)	40	10	2100
Rpm	1450	960	1450
Life (x1000 hrs)	114943	25000	25000
Variable Centre Distance	0	0	0
Module (mm)	2	2	8
Pressure Angle (Deg.)	20	20	20
Helix Angle (Deg.)	15	0	18.6167
Number of Pinion Teeth	24	22	19
Number of Wheel Teeth	96	110	58
Facewidth Pinion (mm)	40	120	125
Facewidth Wheel (mm)	40	122	125
Pinion Tip Diameter (mm)	53.693	48	180.708
Wheel Tip Diameter (mm)	202.773	224	501.292
Pinion Tooth Depth (mm)	4.5	4.5	19.2
Wheel Tooth Depth (mm)	4.5	4.5	19.2
Pinion Pitch Accuracy	5	6	5
Wheel Pitch Accuracy	5	4	5
Pinion Flank Surface Roughness μm	1	2.4	0.8
Wheel Flank Surface Roughness μm	1	2.4	0.8
Pinion Root Surface Roughness μm	3	3.2	3.2
Wheel Root Surface Roughness μm	3	3.2	3.2
Pinion Hob Addendum (mm)	2.5	2.5	11.2
Wheel Hob Addendum (mm)	2.5	2.5	11.2
Pinion Hob Tip Radius Coefficient	0.5	0.78	2.4
Wheel Hob Tip Radius Coefficient	0.5	0.78	2.4
Pinion Addendum Coefficient	0	0	0.27
Wheel Addendum Coefficient	0	0	-0.27
Crowning	0	1	0
Load Distribution Factor	0	0	0
Gear Offset (mm)	60	0	80
Shaft Span Between Bearings (mm)	200	1	300
Shaft Diameter, Pinion	60	130	80
Material Type, Pinion	2	1	2
Material Type, Wheel	2	1	2
Material Quality, Pinion	2	3	2
Material Quality, Wheel	2	2	2
Hardness Process, Pinion	1	1	1
Hardness Process, Wheel	1	1	1
Surface Hardness, Pinion (HV)	700	825	825
Surface Hardness, Wheel (HV)	700	825	825

Effective Case Depth, Pinion (μm)	0.32	0.32	1.3
Effective Case Depth, Wheel (μm)	0.45	0.32	1.3
Ultimate Tensile Strength, Pinion (MN/m²)	2250	2130	2130
Ultimate Tensile Strength, Wheel (MN/m²)	2250	2130	2130
Core Tensile Strength, Pinion (MN/m²)	950	1000	1000
Core Tensile Strength, Wheel (MN/m²)	950	1000	1000
Surface Residual Stress, Pinion (MN/m²)	-400	-400	-400
Surface Residual Stress, Wheel (MN/m²)	-400	-400	-400
Core Residual Stress, Pinion (MN/m²)	240	240	240
Core Residual Stress, Wheel (MN/m²)	240	240	240
Yield Strength for Bending Stress Pinion (MN/m²)	300	300	300
Yield Strength for Bending Stress Wheel (MN/m²)	300	300	300
0.2% Proof Stress, Pinion (MN/m²)	300	300	300
0.2% Proof Stress, Wheel (MN/m²)	300	300	300
Lubrication Viscosity @ 40 °C (cSt)	168	303	303
Application Factor	1	1	1
Min. Surface Safety Factor	1.1	1	1
Min. Bending Safety Factor	1.4	1.4	1.4
Pitting	1	1	1
End Relief	1	0	1
Max. Axial Force (N)	0	0	0
Max. Helix Angle (Deg.)	35	35	35

C.3 Results of the GA application to Gear Design and Optimisation

C 3.1 Use of the Bit Mutation Operator

Populatio Size	200					Repeat Occurance				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	1.75	2.50	5.50	2.25	2.50	0	1	0	0	1
Pressure Angle	24.00	24.00	24.00	24.00	24.00	4	4	4	4	4
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Nunber of Teeth (Pinion)	29.00	28.00	22.00	29.00	22.00	1	0	1	1	1
Facewidth	58.37	37.60	34.50	43.20	47.10	0	0	0	0	0
Addendum Coeficient	0.575	0.775	0.825	0.925	0.850	0	0	0	0	0
Rack Tip Radius Coeficient	0.421	0.421	0.421	0.421	0.421	4	4	4	4	4
Addendum Modification Coeficient (Pinion)	-0.202	-0.051	-0.101	-0.101	-0.102	0	0	1	1	0
Addendum Modification Coeficient (Wheel)	-0.700	-0.401	-0.602	-0.652	-0.700	1	0	0	0	1
Normalised Repeatability						0.389	0.361	0.389	0.389	0.417
Average Normalised Repeatability						0.389				

Table C.1 200 Genome Population using Direct Fitness Bit Mutation Operator

Populatio Size	400					Repeat Occurance				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.75	2.50	2.00	2.00	2.50	0	1	1	1	1
Pressure Angle	24.00	24.00	24.00	24.00	22.50	3	3	3	3	0
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	27.00	26.00	27.00	28.00	25.00	1	0	1	0	0
Facewidth	27.52	36.61	49.25	51.20	43.90	0	0	0	0	0
Addendum Coeficient	0.675	0.650	0.850	0.825	0.800	0	0	0	0	0
Rack Tip Radius Coeficient	0.421	0.421	0.421	0.250	0.405	2	2	2	0	0
Addendum Modification Coeficient (Pinion)	-0.101	-0.102	-0.101	-0.102	0.000	1	1	1	1	0
Addendum Modification Coeficient (Wheel)	-0.500	-0.500	-0.602	-0.602	-0.401	1	1	1	1	0
					Normalised Repeatability	0.333	0.333	0.361	0.278	0.139
					Average Normalised Repeatability	0.289				

Table C.2 400 Genome Population using Direct Fitness Bit Mutation Operator

Populatio Size	600					Repeat Occurance				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.75	2.50	2.50	0.00	0.00	0	1	1	0	0
Pressure Angle	24.00	24.00	22.50	0.00	0.00	1	1	0	0	0
Helix Angle	35.00	35.00	35.00	0.00	0.00	2	2	2	0	0
Number of Teeth (Pinion)	27.00	24.00	26.00	0.00	0.00	0	0	0	0	0
Facewidth	30.72	34.40	34.40	0.00	0.00	0	1	1	0	0
Addendum Coeficient	0.725	1.275	0.850	######	######	0	0	0	0	0
Rack Tip Radius Coeficient	0.421	0.421	0.405	######	######	1	1	0	0	0
Addendum Modification Coeficient (Pinion)	-0.051	0.050	0.000	0.000	0.000	0	0	0	0	0
Addendum Modification Coeficient (Wheel)	-0.401	-0.702	-0.401	0.000	0.000	1	0	1	0	0
					Normalised Repeatability	0.139	0.167	0.139	0.000	0.000
					Average Normalised Repeatability	0.089				

Table C.3 600 Genome Population using Direct Fitness Bit Mutation Operator

(Tests stopped after 3 repeats due to excessive computational expense)

Populatio Size	200				Repeat Occurance				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.75	2.25	3.00	2.75	2.25	1	1	0	1	1
Pressure Angle	22.50	24.00	22.50	24.00	24.00	1	2	1	2	2
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	25.00	26.00	22.00	28.00	24.00	0	0	0	0	0
Facewidth	31.65	38.53	30.27	27.07	42.66	0	0	0	0	0
Addendum Coeficient	1.200	1.250	0.725	0.875	1.025	0	0	0	0	0
Rack Tip Radius Coeficient	0.405	0.421	0.405	0.421	0.421	1	2	1	2	2
Addendum Modification Coeficient (Pinion)	0.000	0.000	-0.152	-0.051	-0.151	1	1	0	1	1
Addendum Modification Coeficient (Wheel)	-0.801	-0.750	-0.750	-0.500	-0.952	0	1	1	0	0
Normalised Repeatability						0.222	0.306	0.194	0.278	0.278
Average Normalised Repeatability						0.256				

Table C.4 200 Genome Population using Ranking Function and Bit Mutation Operator

Populatio Size	400				Repeat Occurance				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.25	2.50	2.50	2.75	2.75	0	1	1	1	1
Pressure Angle	22.50	24.00	24.00	24.00	24.00	0	3	3	3	3
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	29.00	28.00	27.00	24.00	26.00	0	0	0	0	0
Facewidth	31.65	28.58	31.65	37.15	27.52	1	0	1	0	0
Addendum Coeficient	0.650	0.675	0.875	1.075	0.800	0	0	0	0	0
Rack Tip Radius Coeficient	0.300	0.350	0.300	0.421	0.421	1	0	1	1	1
Addendum Modification Coeficient (Pinion)	-0.102	-0.102	-0.102	0.000	-0.102	3	3	3	0	3
Addendum Modification Coeficient (Wheel)	-0.500	-0.500	-0.650	-0.601	-0.602	1	1	0	0	0
Normalised Repeatability						0.278	0.333	0.361	0.250	0.333
Average Normalised Repeatability						0.311				

Table C.5 400 Genome Population using Ranking Function and Bit Mutation Operator

Populatio Size	600				Repeat Occurance				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.75	2.50	2.50	3.00	2.75	1	1	1	0	1
Pressure Angle	24.00	24.00	24.00	24.00	24.00	4	4	4	4	4
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	25.00	28.00	27.00	26.00	26.00	0	0	0	1	1
Facewidth	31.65	28.90	34.40	24.77	27.52	0	0	0	0	0
Addendum Coeficient	0.900	0.775	1.125	1.000	0.800	0	0	0	0	0
Rack Tip Radius Coeficient	0.421	0.350	0.350	0.421	0.350	1	2	2	1	2
Addendum Modification Coeficient (Pinion)	-0.102	-0.152	0.000	0.000	-0.102	1	0	1	1	1
Addendum Modification Coeficient (Wheel)	-0.700	-0.702	-0.602	-0.500	-0.602	0	0	1	0	1
					Normalised Repeatability	0.306	0.306	0.361	0.306	0.389
					Average Normalised Repeatability	0.333				

Table C.6 600 Genome Population using Ranking Function and Bit Mutation Operator

Populatio Size	800				Repeat Occurance				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	3.00	3.00	2.50	3.50	3.00	2	2	0	0	2
Pressure Angle	24.00	24.00	24.00	24.00	24.00	4	4	4	4	4
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	25.00	24.00	26.00	22.00	26.00	0	0	1	0	1
Facewidth	26.14	26.14	31.65	24.77	26.14	2	2	0	0	2
Addendum Coeficient	0.900	0.775	0.800	0.925	0.925	0	0	0	1	1
Rack Tip Radius Coeficient	0.421	0.400	0.350	0.421	0.421	2	0	0	2	2
Addendum Modification Coeficient (Pinion)	-0.102	-0.102	-0.102	0.000	-0.102	3	3	3	0	3
Addendum Modification Coeficient (Wheel)	-0.700	-0.600	-0.600	-0.501	-0.700	1	1	1	0	1
					Normalised Repeatability	0.500	0.444	0.361	0.306	0.556
					Average Normalised Repeatability	0.433				

Table C.7 800 Genome Population using Ranking Function and Bit Mutation Operator

Populatio Size	1000					Repeat Occurance				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	3.00	2.75	3.00	2.75	2.75	1	1	0	1	1
Pressure Angle	24.00	24.00	24.00	24.00	24.00	4	4	4	4	4
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	25.00	25.00	25.00	26.00	26.00	2	2	2	1	1
Facewidth	26.14	31.65	26.14	28.90	28.90	1	0	1	1	1
Addendum Coeficient	0.900	0.900	0.900	0.800	0.800	2	2	2	1	1
Rack Tip Radius Coeficient	0.421	0.421	0.421	0.350	0.350	2	2	2	1	1
Addendum Modification Coeficient (Pinion)	-0.102	-0.102	-0.102	-0.102	-0.102	4	4	4	4	4
Addendum Modification Coeficient (Wheel)	-0.700	-0.700	-0.700	-0.602	-0.602	2	2	2	1	1
Normalised Repeatability						0.611	0.583	0.583	0.500	0.500
Average Normalised Repeatability						0.556				

Table C.8 1000 Genome Population using Ranking Function and Bit Mutation Operator

Populatio Size	1500					Repeat Occurance				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	3.50	3.00	2.50	3.00	3.50	1	1	0	1	1
Pressure Angle	24.00	24.00	24.00	24.00	24.00	4	4	4	4	4
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	22.00	25.00	27.00	24.00	23.00	0	0	0	0	0
Facewidth	24.77	26.14	26.14	26.14	24.77	1	2	2	2	1
Addendum Coeficient	1.025	1.025	1.275	0.900	0.850	1	1	0	0	0
Rack Tip Radius Coeficient	0.421	0.400	0.421	0.421	0.421	3	0	3	3	3
Addendum Modification Coeficient (Pinion)	0.000	-0.002	0.000	-0.102	0.000	2	0	2	0	2
Addendum Modification Coeficient (Wheel)	-0.601	-0.551	-0.751	-0.700	-0.401	0	0	0	0	0
Normalised Repeatability						0.444	0.333	0.417	0.389	0.417
Average Normalised Repeatability						0.400				

Table C.9 1500 Genome Population using Ranking Function and Bit Mutation Operator

Populatio Size	3000					Repeat Occurance				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	3.00	2.50	3.00	2.50	3.00	2	1	2	1	2
Pressure Angle	24.00	24.00	24.00	24.00	24.00	4	4	4	4	4
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	25.00	27.00	24.00	27.00	26.00	0	1	0	1	0
Facewidth	24.77	26.14	26.14	26.14	24.77	1	2	2	2	1
Addendum Coeficient	0.900	1.325	0.775	1.325	0.925	0	1	0	1	0
Rack Tip Radius Coeficient	0.421	0.421	0.400	0.421	0.421	3	3	0	3	3
Addendum Modification Coeficient (Pinion)	-0.102	0.000	-0.102	0.000	-0.102	2	1	2	1	2
Addendum Modification Coeficient (Wheel)	-0.700	-0.801	-0.602	-0.801	-0.700	1	1	0	1	1
Normalised Repeatability						0.472	0.500	0.389	0.500	0.472
Average Normalised Repeatability						0.467				

Table C.10 3000 Genome Population using Ranking Function and Bit Mutation Operator

Populatio Size	200					Repeat Occurance				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.25	3.00	2.25	3.50	3.00	1	1	1	0	1
Pressure Angle	24.00	24.00	24.00	22.50	24.00	3	3	3	0	3
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	26.00	26.00	25.00	22.00	26.00	2	2	0	0	2
Facewidth	38.40	25.34	48.00	28.90	28.22	0	0	0	0	0
Addendum Coeficient	0.750	0.675	0.975	0.675	0.650	0	1	0	1	0
Rack Tip Radius Coeficient	0.421	0.421	0.421	0.405	0.421	3	3	3	0	3
Addendum Modification Coeficient (Pinion)	-0.152	-0.151	0.000	0.000	-0.102	0	0	1	1	0
Addendum Modification Coeficient (Wheel)	-0.702	-0.650	-0.500	-0.301	-0.500	0	0	1	0	1
Normalised Repeatability						0.361	0.389	0.361	0.167	0.389
Average Normalised Repeatability						0.333				

Table C.11 200 Genome Population using Combined Scaling Function and Bit Mutation Operator

Populatio Size	400				Repeat Occurance				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	3.00	3.00	3.00	3.00	2.50	3	3	3	3	0
Pressure Angle	24.00	24.00	24.00	24.00	24.00	4	4	4	4	4
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	26.00	27.00	27.00	24.00	28.00	0	1	1	0	0
Facewidth	24.77	24.77	24.77	27.52	28.90	2	2	2	0	0
Addendum Coeficient	0.925	0.675	1.125	0.800	0.775	0	0	0	0	0
Rack Tip Radius Coeficient	0.421	0.421	0.421	0.421	0.350	3	3	3	3	0
Addendum Modification Coeficient (Pinion)	-0.102	-0.101	0.000	0.000	-0.152	0	0	1	1	0
Addendum Modification Coeficient (Wheel)	-0.700	-0.500	-0.602	-0.351	-0.702	0	0	0	0	0
Normalised Repeatability						0.444	0.472	0.500	0.417	0.222
Average Normalised Repeatability						0.411				

Table C.12 400 Genome Population using Combined Scaling Function and Bit Mutation Operator

Populatio Size	600				Repeat Occurance				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.50	3.00	3.00	2.75	3.50	0	1	1	0	0
Pressure Angle	24.00	24.00	24.00	24.00	24.00	4	4	4	4	4
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	28.00	26.00	28.00	26.00	23.00	1	1	1	1	0
Facewidth	28.90	24.77	24.77	27.52	24.77	0	2	2	0	2
Addendum Coeficient	0.775	0.800	0.900	0.925	1.050	0	0	0	0	0
Rack Tip Radius Coeficient	0.350	0.421	0.421	0.421	0.421	0	3	3	3	3
Addendum Modification Coeficient (Pinion)	-0.152	-0.102	-0.102	-0.102	0.000	0	2	2	2	0
Addendum Modification Coeficient (Wheel)	-0.702	-0.602	-0.650	-0.700	-0.602	0	1	0	0	1
Normalised Repeatability						0.250	0.500	0.472	0.389	0.389
Average Normalised Repeatability						0.400				

Table C.13 600 Genome Population using Combined Scaling Function and Bit Mutation Operator

Populatio Size	800					Repeat Occurance				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.75	3.00	3.50	3.00	2.75	1	1	0	1	1
Pressure Angle	24.00	24.00	24.00	24.00	24.00	4	4	4	4	4
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	29.00	26.00	22.00	26.00	26.00	0	2	0	2	2
Facewidth	24.77	24.77	24.77	24.77	27.52	3	3	3	3	0
Addendum Coeficient	0.850	0.800	0.750	0.800	0.800	0	2	0	2	2
Rack Tip Radius Coeficient	0.400	0.421	0.421	0.421	0.350	0	2	2	2	0
Addendum Modification Coeficient (Pinion)	-0.102	-0.101	-0.102	-0.102	-0.102	3	0	3	3	3
Addendum Modification Coeficient (Wheel)	-0.602	-0.602	-0.602	-0.602	-0.602	4	4	4	4	4
					Normalised Repeatability	0.528	0.611	0.556	0.694	0.556
					Average Normalised Repeatability	0.589				

Table C.14 800 Genome Population using Combined Scaling Function and Bit Mutation Operator

Populatio Size	1000					Repeat Occurance				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.75	2.75	3.50	3.00	3.00	1	1	0	1	1
Pressure Angle	24.00	24.00	24.00	24.00	24.00	4	4	4	4	4
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	26.00	29.00	23.00	27.00	25.00	0	0	0	0	0
Facewidth	27.52	24.77	24.77	26.14	26.14	0	1	1	1	1
Addendum Coeficient	0.800	1.150	0.850	0.875	0.900	0	0	0	0	0
Rack Tip Radius Coeficient	0.350	0.421	0.421	0.400	0.421	0	2	2	0	2
Addendum Modification Coeficient (Pinion)	-0.102	-0.051	0.000	-0.102	-0.102	2	0	0	2	2
Addendum Modification Coeficient (Wheel)	-0.602	-0.702	-0.401	-0.650	-0.700	0	0	0	0	0
					Normalised Repeatability	0.306	0.333	0.306	0.333	0.389
					Average Normalised Repeatability	0.333				

Table C.15 1000 Genome Population using Combined Scaling Function and Bit Mutation Operator

Populatio Size	1500	Repeat Occurance								
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.75	3.00	3.00	3.00	3.50	0	2	2	2	0
Pressure Angle	24.00	24.00	24.00	24.00	24.00	4	4	4	4	4
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	26.00	26.00	26.00	26.00	23.00	3	3	3	3	0
Facewidth	27.52	24.77	26.14	24.77	24.77	0	2	0	2	2
Addendum Coeficient	0.800	0.925	0.925	0.925	0.850	0	2	2	2	0
Rack Tip Radius Coeficient	0.350	0.421	0.421	0.421	0.421	0	3	3	3	3
Addendum Modification Coeficient (Pinion)	-0.102	-0.102	-0.102	-0.102	0.000	3	3	3	3	0
Addendum Modification Coeficient (Wheel)	-0.602	-0.700	-0.700	-0.700	-0.401	0	2	2	2	0
Normalised Repeatability						0.389	0.694	0.639	0.694	0.361
Average Normalised Repeatability						0.556				

Table C.16 1500 Genome Population using Combined Scaling Function and Bit Mutation Operator

Populatio Size	3000	Repeat Occurance								
Test Number	1	2	3	4	5	1	2	3	4	5
Module	3.00	3.00	3.00	3.00	3.00	4	4	4	4	4
Pressure Angle	24.00	24.00	24.00	24.00	24.00	4	4	4	4	4
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	26.00	25.00	25.00	25.00	27.00	0	2	2	2	0
Facewidth	24.77	26.14	24.77	26.14	24.77	2	1	2	1	2
Addendum Coeficient	0.800	0.900	1.100	0.900	1.175	0	0	0	0	0
Rack Tip Radius Coeficient	0.400	0.421	0.421	0.421	0.421	3	3	3	3	3
Addendum Modification Coeficient (Pinion)	-0.102	-0.102	0.000	-0.102	0.000	2	2	1	2	1
Addendum Modification Coeficient (Wheel)	-0.602	-0.700	-0.602	-0.700	-0.652	1	1	1	1	0
Normalised Repeatability						0.556	0.583	0.583	0.583	0.500
Average Normalised Repeatability						0.561				

Table C.17 3000 Genome Population using Combined Scaling Function and Bit Mutation Operator

C 3.2 Gene Mutation Operator

Population Size	200					Repeat Occurrence				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.50	2.50	2.00	2.00	2.25	1	1	1	1	0
Pressure Angle	24.00	22.50	22.50	22.50	22.50	0	3	3	3	3
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	3	0	3	3
Number of Teeth (Pinion)	22.00	22.00	21.00	21.00	21.00	1	1	2	2	2
Facewidth	31.65	37.60	35.78	54.27	33.02	0	0	0	0	0
Addendum Coefficient	1.275	0.925	1.225	1.175	1.175	0	0	0	1	1
Rack Tip Radius Coefficient	0.421	0.405	0.405	0.405	0.405	0	3	3	3	3
Addendum Modification Coefficient (Pinion)	-0.202	-0.700	-0.002	-0.102	-0.102	0	0	0	1	1
Addendum Modification Coefficient (Wheel)	-0.051	-1.000	0.250	0.250	0.200	0	0	1	1	0
Normalised Repeatability						0.167	0.306	0.278	0.417	0.361
Average Normalised Repeatability						0.306				

Table C.18 200 Genome Population using Ranking Function and Gene Mutation Operator

Population Size	400					Repeat Occurrence				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.00	2.00	2.00	2.00	2.25	3	3	3	3	0
Pressure Angle	22.50	22.50	22.50	22.50	24.00	3	3	3	3	0
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	23.00	23.00	21.00	21.00	20.00	1	1	1	1	0
Facewidth	27.52	27.52	36.00	44.03	34.40	1	1	0	0	0
Addendum Coefficient	1.275	1.275	1.200	1.150	1.300	1	1	0	0	0
Rack Tip Radius Coefficient	0.405	0.405	0.405	0.405	0.421	3	3	3	3	0
Addendum Modification Coefficient (Pinion)	0.000	-0.002	0.048	0.000	-0.152	1	0	0	1	0
Addendum Modification Coefficient (Wheel)	0.300	0.300	0.300	0.300	0.048	3	3	3	3	0
Normalised Repeatability						0.556	0.528	0.472	0.500	0.111
Average Normalised Repeatability						0.433				

Table C.19 400 Genome Population using Ranking Function and Gene Mutation Operator

Population Size	600					Repeat Occurrence				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.00	2.00	2.00	2.00	2.00	4	4	4	4	4
Pressure Angle	22.50	22.50	22.50	22.50	22.50	4	4	4	4	4
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	23.00	21.00	22.00	23.00	23.00	2	0	0	2	2
Facewidth	30.27	33.02	30.27	26.14	27.52	1	0	1	0	0
Addendum Coefficient	1.300	1.275	1.250	1.325	1.300	1	0	0	0	1
Rack Tip Radius Coefficient	0.405	0.405	0.405	0.405	0.405	4	4	4	4	4
Addendum Modification Coefficient (Pinion)	0.049	0.150	0.050	0.048	0.048	0	0	0	1	1
Addendum Modification Coefficient (Wheel)	0.300	0.300	0.300	0.300	0.300	4	4	4	4	4
					Normalised Repeatability	0.667	0.556	0.583	0.639	0.667
					Average Normalised Repeatability	0.622				

Table C.20 600 Genome Population using Ranking Function and Gene Mutation Operator

Population Size	800					Repeat Occurrence				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.00	2.00	2.00	2.00	2.00	4	4	4	4	4
Pressure Angle	22.50	22.50	24.00	22.50	22.50	3	3	0	3	3
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	23.00	22.00	22.00	22.00	22.00	0	3	3	3	3
Facewidth	26.14	30.27	33.02	30.27	30.27	0	2	0	2	2
Addendum Coefficient	1.400	1.300	1.325	1.325	1.300	0	1	1	1	1
Rack Tip Radius Coefficient	0.405	0.405	0.421	0.405	0.405	3	3	0	3	3
Addendum Modification Coefficient (Pinion)	-0.002	0.000	-0.052	-0.052	0.000	0	1	1	1	0
Addendum Modification Coefficient (Wheel)	0.250	0.250	0.200	0.200	0.250	2	2	1	1	2
					Normalised Repeatability	0.444	0.639	0.389	0.611	0.611
					Average Normalised Repeatability	0.539				

Table C.21 800 Genome Population using Ranking Function and Gene Mutation Operator

Population Size	1000					Repeat Occurrence				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.00	2.00	2.00	2.00	2.00	4	4	4	4	4
Pressure Angle	22.50	22.50	22.50	22.50	22.50	4	4	4	4	4
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	23.00	23.00	22.00	23.00	22.00	2	2	1	2	1
Facewidth	26.14	27.52	30.27	27.52	30.27	0	1	1	1	1
Addendum Coefficient	1.350	1.275	1.300	1.325	1.300	0	0	1	0	1
Rack Tip Radius Coefficient	0.405	0.405	0.405	0.405	0.405	4	4	4	4	4
Addendum Modification Coefficient (Pinion)	0.048	0.000	0.000	-0.102	0.150	0	1	1	0	0
Addendum Modification Coefficient (Wheel)	0.250	0.300	0.250	0.250	0.300	2	1	2	2	1
Normalised Repeatability						0.556	0.583	0.611	0.583	0.556
Average Normalised Repeatability						0.578				

Table C.22 1000 Genome Population using Ranking Function and Gene Mutation Operator

Population Size	1500					Repeat Occurrence				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.00	2.00	2.00	2.00	2.00	4	4	4	4	4
Pressure Angle	22.50	22.50	22.50	22.50	22.50	4	4	4	4	4
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	23.00	22.00	21.00	23.00	22.00	1	1	0	1	1
Facewidth	26.14	30.27	31.65	27.52	30.27	0	1	0	0	1
Addendum Coefficient	1.375	1.275	1.300	1.375	1.300	1	0	0	1	0
Rack Tip Radius Coefficient	0.405	0.405	0.405	0.405	0.405	4	4	4	4	4
Addendum Modification Coefficient (Pinion)	-0.002	0.100	0.148	0.000	0.000	0	0	0	1	1
Addendum Modification Coefficient (Wheel)	0.250	0.300	0.300	0.250	0.250	2	1	1	2	2
Normalised Repeatability						0.556	0.528	0.472	0.583	0.583
Average Normalised Repeatability						0.544				

Table C.23 1500 Genome Population using Ranking Function and Gene Mutation Operator

Population Size	3000					Repeat Occurrence				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.00	2.00	2.00	2.00	2.00	4	4	4	4	4
Pressure Angle	22.50	22.50	20.00	22.50	22.50	3	3	0	3	3
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	24.00	23.00	23.00	24.00	23.00	1	2	2	1	2
Facewidth	24.77	26.14	26.14	24.77	27.52	1	1	1	1	0
Addendum Coefficient	1.350	1.325	1.300	1.425	1.375	0	0	0	0	0
Rack Tip Radius Coefficient	0.405	0.405	0.380	0.405	0.405	3	3	0	3	3
Addendum Modification Coefficient (Pinion)	-0.102	-0.102	-0.102	-0.102	-0.102	4	4	4	4	4
Addendum Modification Coefficient (Wheel)	0.250	0.250	0.250	0.250	0.349	3	3	3	3	0
Normalised Repeatability						0.639	0.667	0.500	0.639	0.556
Average Normalised Repeatability						0.600				

Table C.24 3000 Genome Population using Ranking Function and Gene Mutation Operator

Populatio Size	200					Repeat Occurance				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.50	2.00	2.50	2.00	1.75	1	1	1	1	0
Pressure Angle	24.00	22.50	17.50	24.00	24.00	2	0	0	2	2
Helix Angle	35.00	35.00	0.00	35.00	35.00	3	3	0	3	3
Number of Teeth (Pinion)	23.00	21.00	29.00	23.00	27.00	1	0	0	1	0
Facewidth	39.10	43.62	26.14	35.62	43.65	0	0	0	0	0
Addendum Coeficient	0.900	1.075	1.275	1.375	1.350	0	0	0	0	0
Rack Tip Radius Coeficient	0.421	0.405	0.357	0.421	0.421	2	0	0	2	2
Addendum Modification Coeficient (Pinion)	-0.102	-0.102	0.000	-0.251	-0.251	1	1	0	1	1
Addendum Modification Coeficient (Wheel)	-0.601	0.349	-0.151	0.000	0.349	0	1	0	0	1
Normalised Repeatability						0.278	0.167	0.028	0.278	0.250
Average Normalised Repeatability						0.200				

Table C.25 200 Genome Population using Combined Scaling Function and Gene Mutation Operator

Populatio Size	400					Repeat Occurance				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.00	2.00	2.00	2.00	2.00	4	4	4	4	4
Pressure Angle	22.50	22.50	22.50	22.50	22.50	4	4	4	4	4
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	23.00	23.00	22.00	21.00	24.00	1	1	0	0	0
Facewidth	39.90	30.27	44.03	33.02	39.90	1	0	0	0	1
Addendum Coeficient	1.375	1.200	1.150	1.300	1.425	0	0	0	0	0
Rack Tip Radius Coeficient	0.405	0.405	0.405	0.405	0.405	4	4	4	4	4
Addendum Modification Coeficient (Pinion)	-0.202	0.048	-0.251	0.050	-0.202	1	0	0	0	1
Addendum Modification Coeficient (Wheel)	-0.151	0.349	0.050	0.250	-0.101	0	0	0	0	0
Normalised Repeatability						0.528	0.472	0.444	0.444	0.500
Average Normalised Repeatability						0.478				

Table C.26 400 Genome Population using Combined Scaling Function and Gene Mutation Operator

Populatio Size	600					Repeat Occurance				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.00	2.00	2.00	2.25	2.00	3	3	3	0	3
Pressure Angle	22.50	22.50	22.50	22.50	22.50	4	4	4	4	4
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	21.00	22.00	22.00	21.00	22.00	1	2	2	1	2
Facewidth	34.40	30.27	30.27	27.07	31.65	0	1	1	0	0
Addendum Coeficient	1.300	1.300	1.300	1.300	1.200	3	3	3	3	0
Rack Tip Radius Coeficient	0.405	0.405	0.405	0.405	0.405	4	4	4	4	4
Addendum Modification Coeficient (Pinion)	0.050	-0.052	-0.052	-0.052	0.000	0	2	2	2	0
Addendum Modification Coeficient (Wheel)	0.250	0.250	0.250	0.200	0.349	2	2	2	0	0
Normalised Repeatability						0.583	0.694	0.694	0.500	0.472
Average Normalised Repeatability						0.589				

Table C.27 600 Genome Population using Combined Scaling Function and Gene Mutation Operator

Populatio Size	800					Repeat Occurance				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.00	2.00	2.00	2.00	2.25	3	3	3	3	0
Pressure Angle	22.50	22.50	24.00	22.50	22.50	3	3	0	3	3
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	23.00	22.00	23.00	23.00	21.00	2	0	2	2	0
Facewidth	27.52	30.27	30.27	27.52	30.27	1	2	2	1	2
Addendum Coeficient	1.325	1.250	1.350	1.325	1.225	1	0	0	1	0
Rack Tip Radius Coeficient	0.405	0.405	0.421	0.405	0.405	3	3	0	3	3
Addendum Modification Coeficient (Pinion)	-0.052	0.000	-0.102	-0.102	-0.002	0	1	1	1	0
Addendum Modification Coeficient (Wheel)	0.250	0.300	0.200	0.250	0.250	2	0	0	2	2
Normalised Repeatability						0.528	0.444	0.333	0.556	0.389
Average Normalised Repeatability						0.450				

Table C.28 800 Genome Population using Combined Scaling Function and Gene Mutation Operator

Populatio Size	1000					Repeat Occurance				
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.00	2.00	2.00	2.00	2.00	4	4	4	4	4
Pressure Angle	22.50	22.50	22.50	22.50	22.50	4	4	4	4	4
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	23.00	23.00	23.00	22.00	24.00	2	2	2	0	0
Facewidth	30.27	30.27	28.90	30.27	26.14	2	2	0	2	0
Addendum Coeficient	1.250	1.350	1.300	1.300	1.300	0	0	2	2	2
Rack Tip Radius Coeficient	0.405	0.405	0.405	0.405	0.405	4	4	4	4	4
Addendum Modification Coeficient (Pinion)	-0.052	-0.101	0.048	0.000	-0.052	1	0	0	0	1
Addendum Modification Coeficient (Wheel)	0.300	0.200	0.349	0.250	0.300	1	0	0	0	1
Normalised Repeatability						0.611	0.556	0.556	0.556	0.556
Average Normalised Repeatability						0.567				

Table C.29 1000 Genome Population using Combined Scaling Function and Gene Mutation Operator

Populatio Size	1500					Repeat Occurance			
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.00	2.00	2.00	2.00	2.00	4	4	4	4	4
Pressure Angle	22.50	22.50	22.50	22.50	22.50	4	4	4	4	4
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	23.00	23.00	23.00	23.00	23.00	4	4	4	4	4
Facewidth	26.14	27.52	26.14	27.52	26.14	2	1	2	1	2
Addendum Coeficient	1.375	1.300	1.375	1.300	1.375	2	1	2	1	2
Rack Tip Radius Coeficient	0.405	0.405	0.405	0.405	0.405	4	4	4	4	4
Addendum Modification Coeficient (Pinion)	-0.002	0.048	-0.002	0.048	-0.002	2	1	2	1	2
Addendum Modification Coeficient (Wheel)	0.250	0.300	0.250	0.300	0.250	2	1	2	1	2
					Normalised Repeatability	0.778	0.667	0.778	0.667	0.778
					Average Normalised Repeatability	0.733				

Table C.30 1500 Genome Population using Combined Scaling Function and Gene Mutation Operator

Populatio Size	3000					Repeat Occurance			
Test Number	1	2	3	4	5	1	2	3	4	5
Module	2.00	2.00	2.00	2.00	2.00	4	4	4	4	4
Pressure Angle	22.50	22.50	22.50	22.50	22.50	4	4	4	4	4
Helix Angle	35.00	35.00	35.00	35.00	35.00	4	4	4	4	4
Number of Teeth (Pinion)	24.00	24.00	24.00	24.00	23.00	3	3	3	3	0
Facewidth	24.77	24.77	24.77	24.77	24.77	4	4	4	4	4
Addendum Coeficient	1.425	1.425	1.425	1.425	1.400	4	4	4	4	4
Rack Tip Radius Coeficient	0.405	0.405	0.405	0.405	0.405	4	4	4	4	4
Addendum Modification Coeficient (Pinion)	-0.102	-0.102	-0.102	-0.102	-0.102	4	4	4	4	4
Addendum Modification Coeficient (Wheel)	0.250	0.250	0.250	0.250	0.349	3	3	3	3	0
					Normalised Repeatability	0.944	0.944	0.944	0.944	0.778
					Average Normalised Repeatability	0.911				

Table C.31 3000 Genome Population using Combined Scaling Function and Gene Mutation Operator

C 4 Parameter Traces for Test Cases 2 and 3

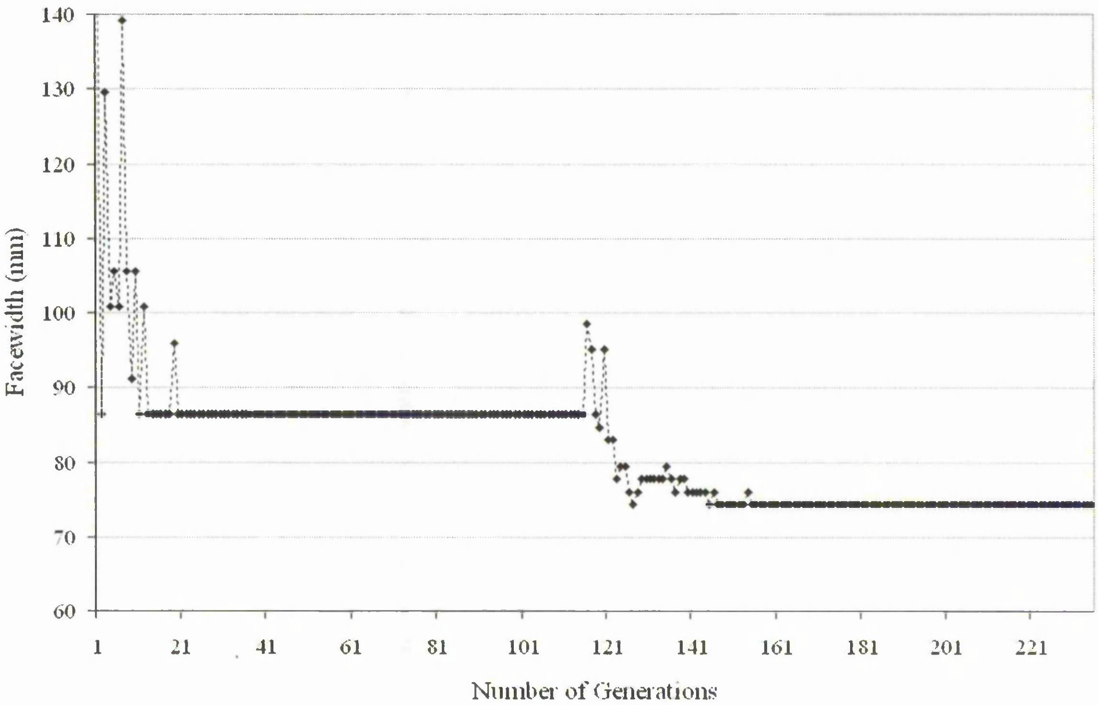


Figure C.13a. Test Case 2 Facewidth

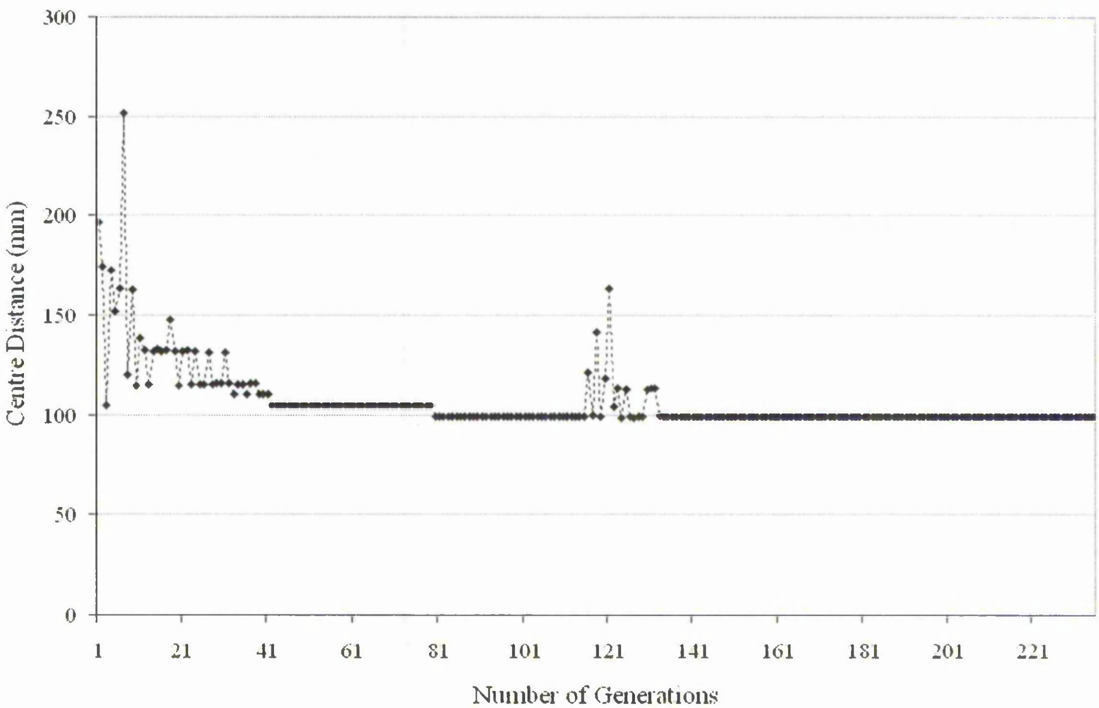


Figure C.13b Test Case 2 Centre Distance

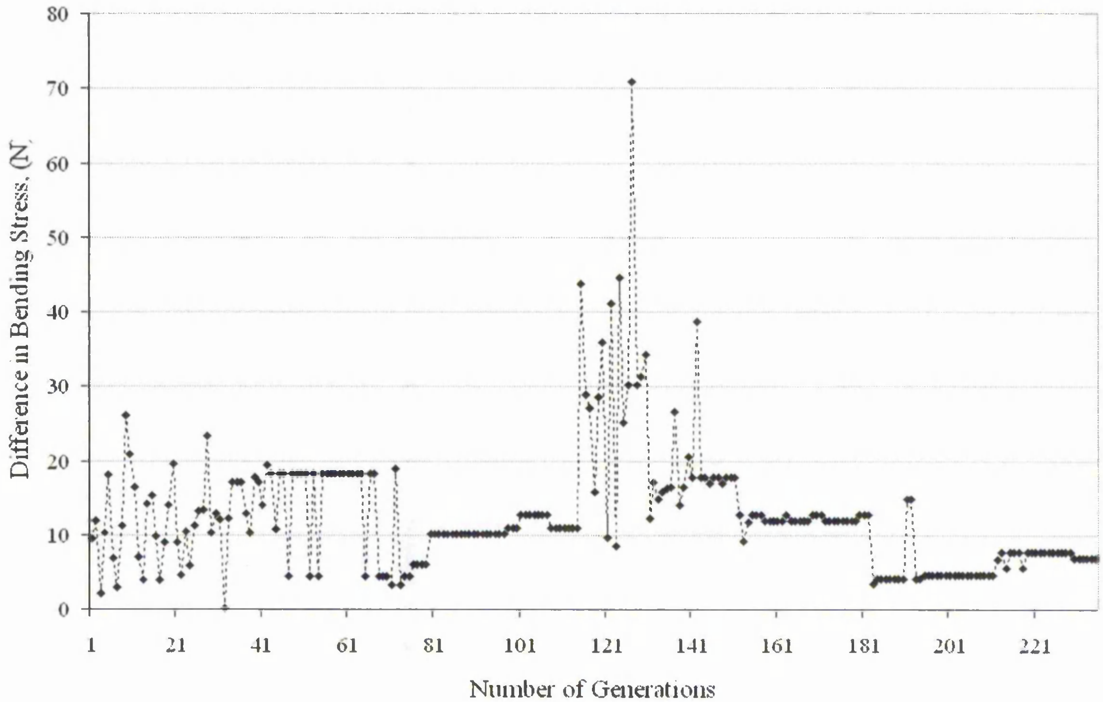


Figure C.13c Test Case 2 Equal Bending Stress

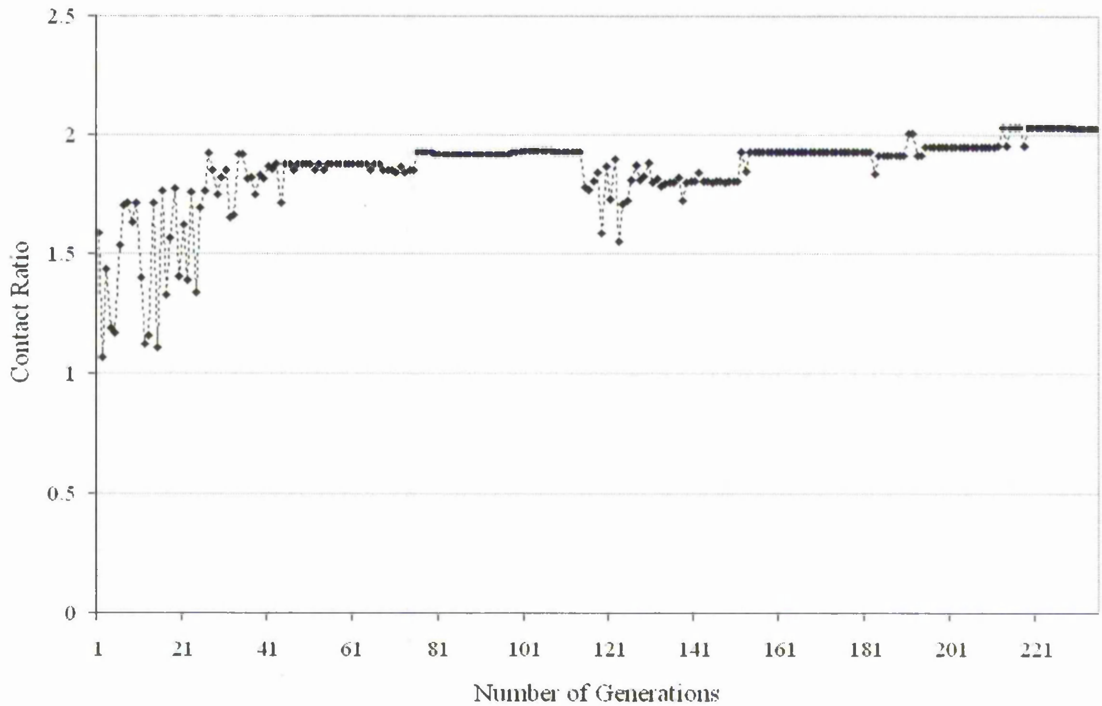


Figure C.13d Test Case 2 Contact Ratio

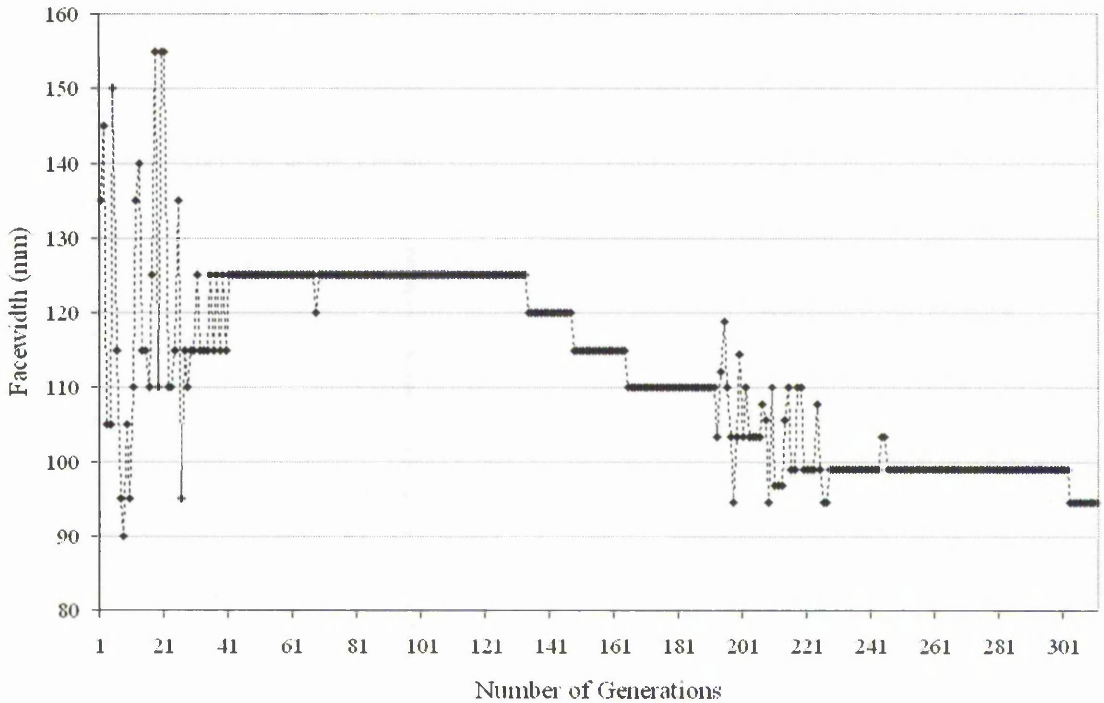


Figure C.14a Test Case 3 Facewidth

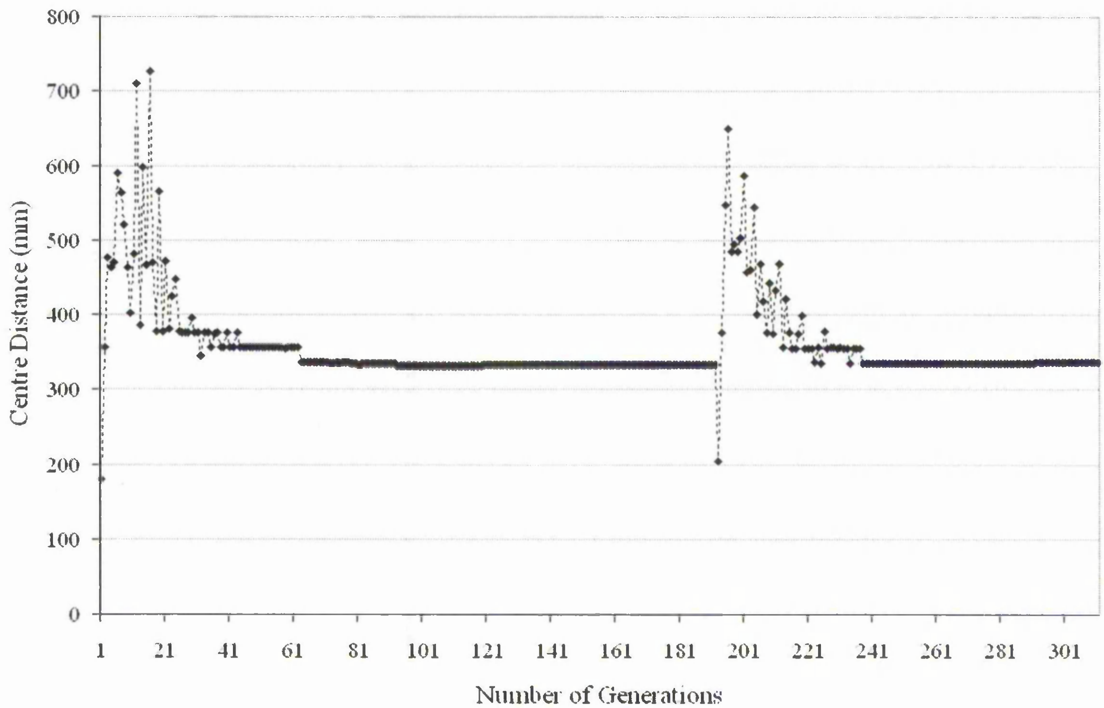


Figure C.14b Test Case 3 Centre Distance

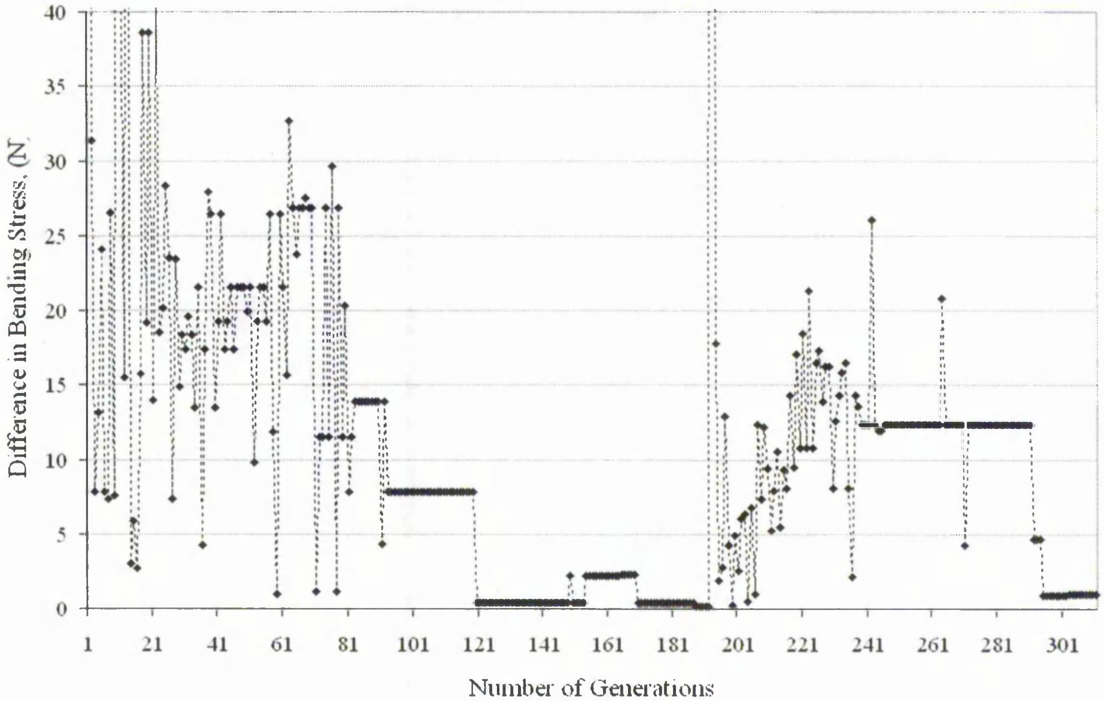


Figure C.14c Test Case 3 Equal Bending Stress

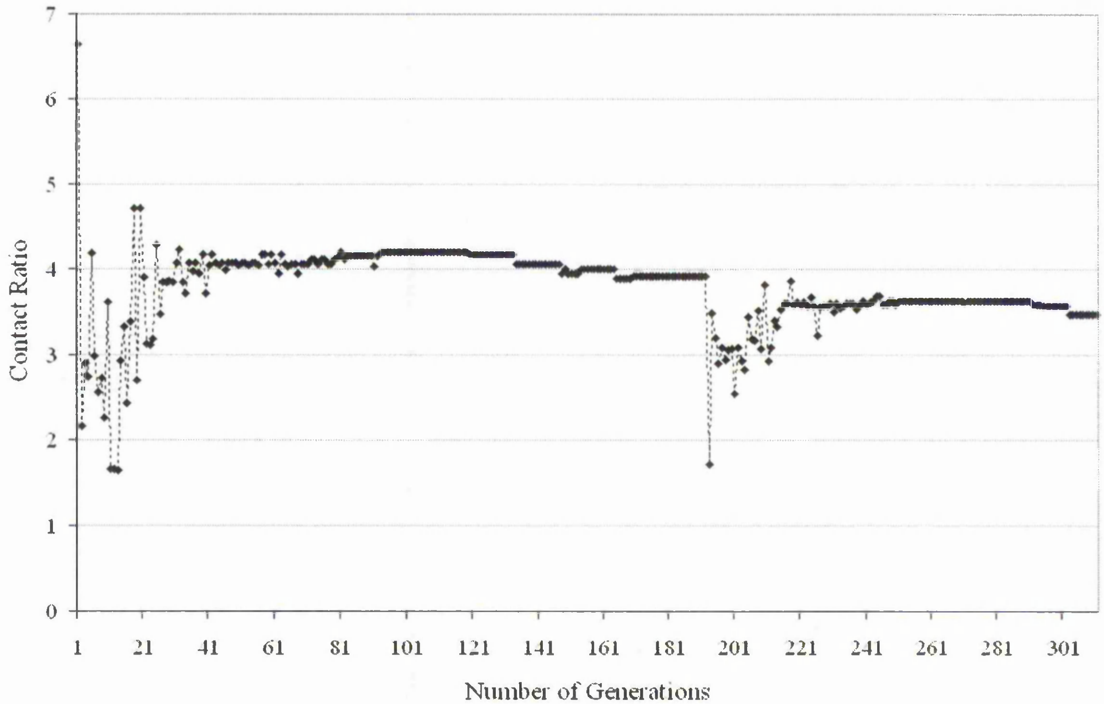
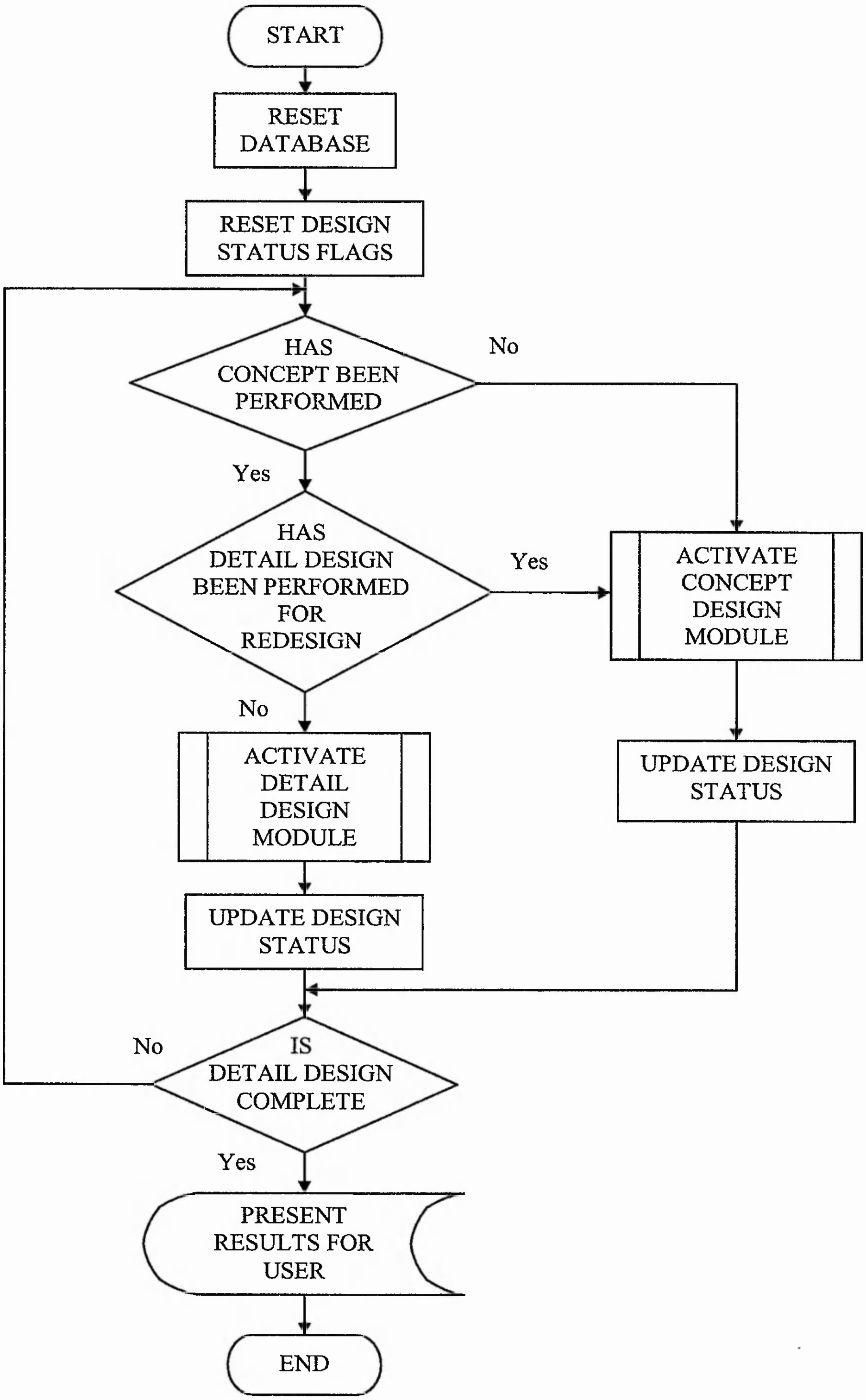


Figure C.14d Test Case 3 Contact Ratio

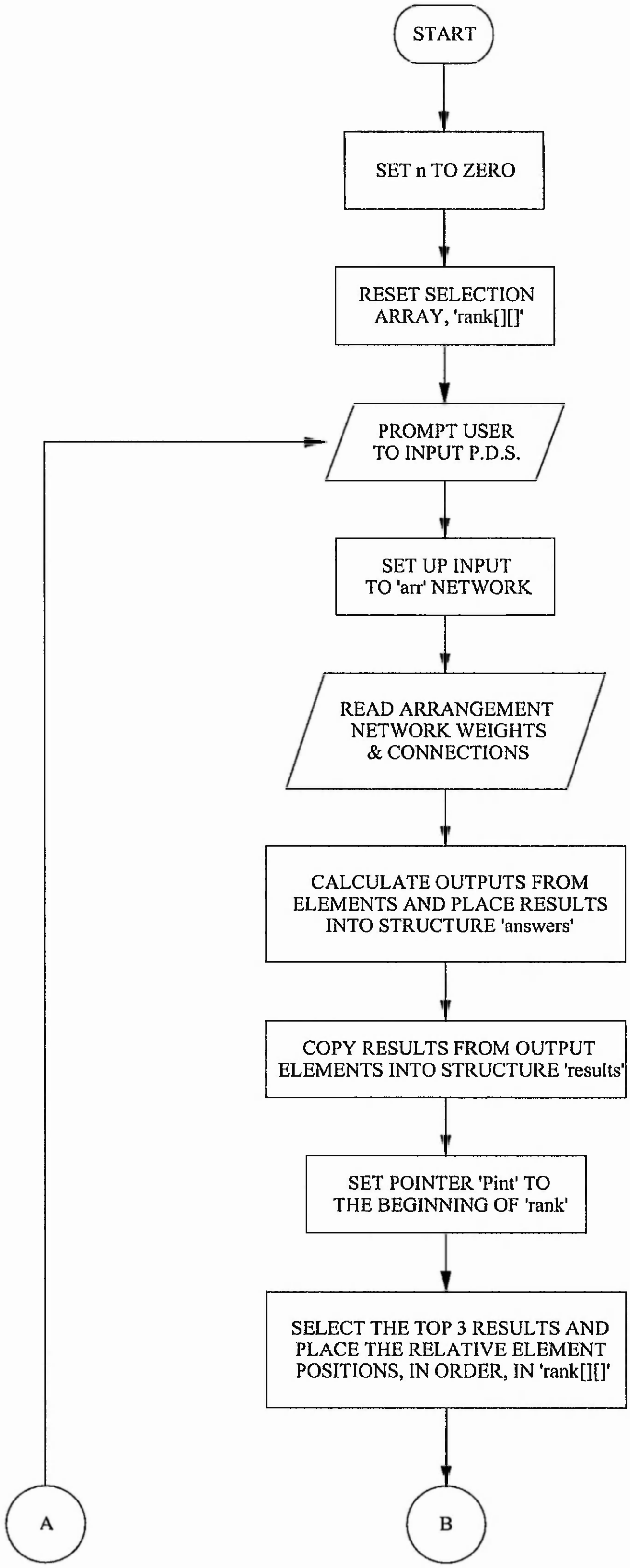
APPENDIX D

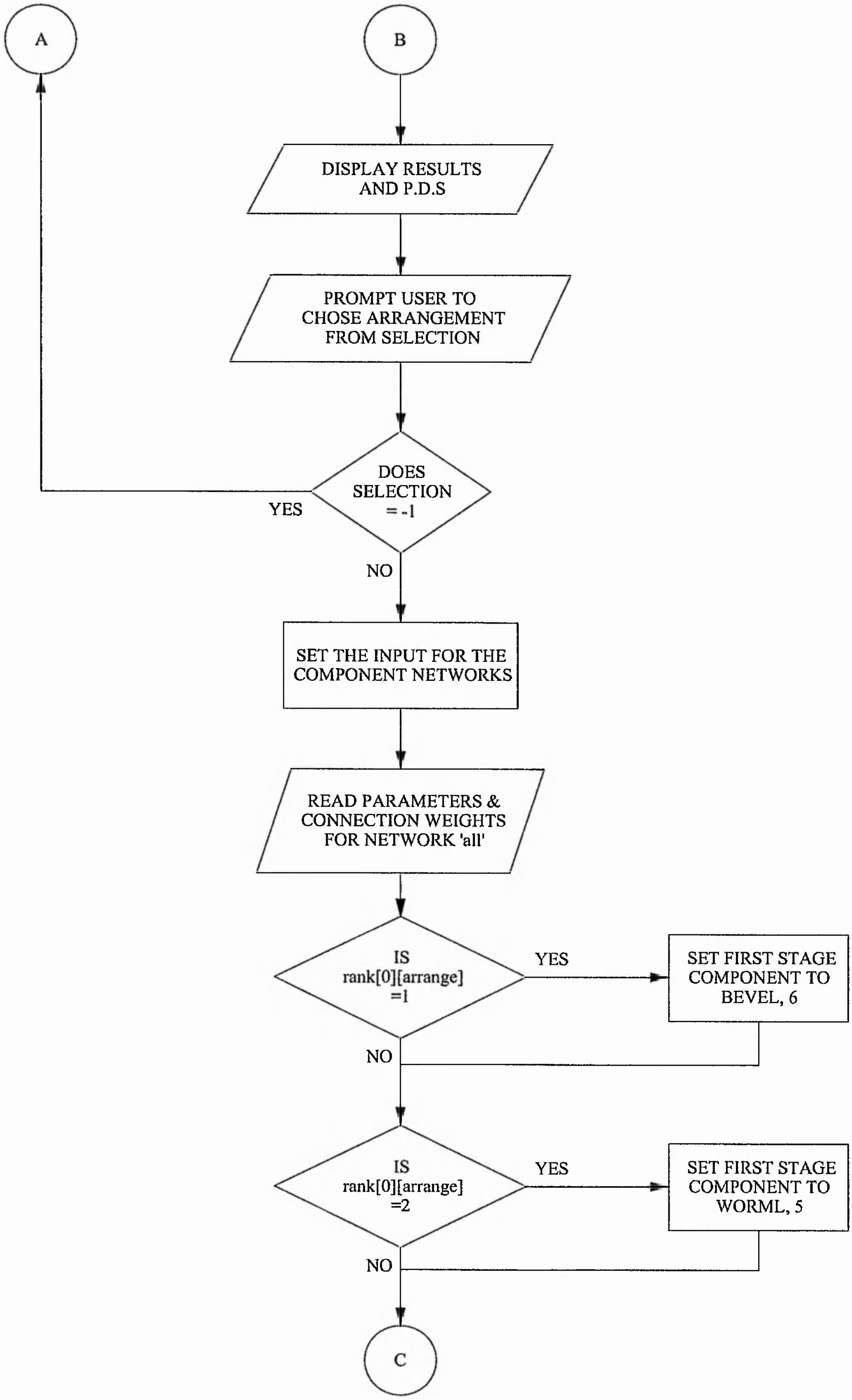
PROGRAM FLOWCHARTS

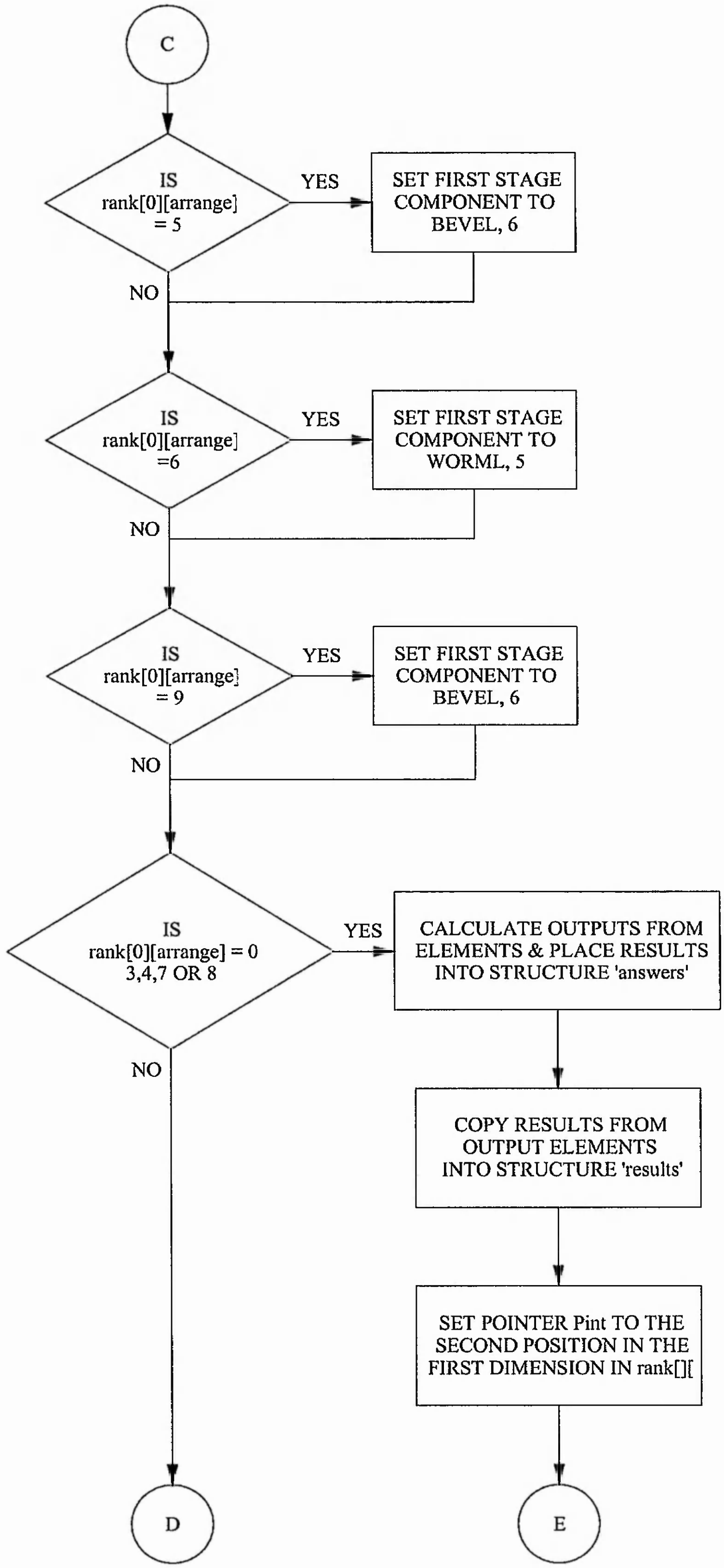
D.1. System Control Program

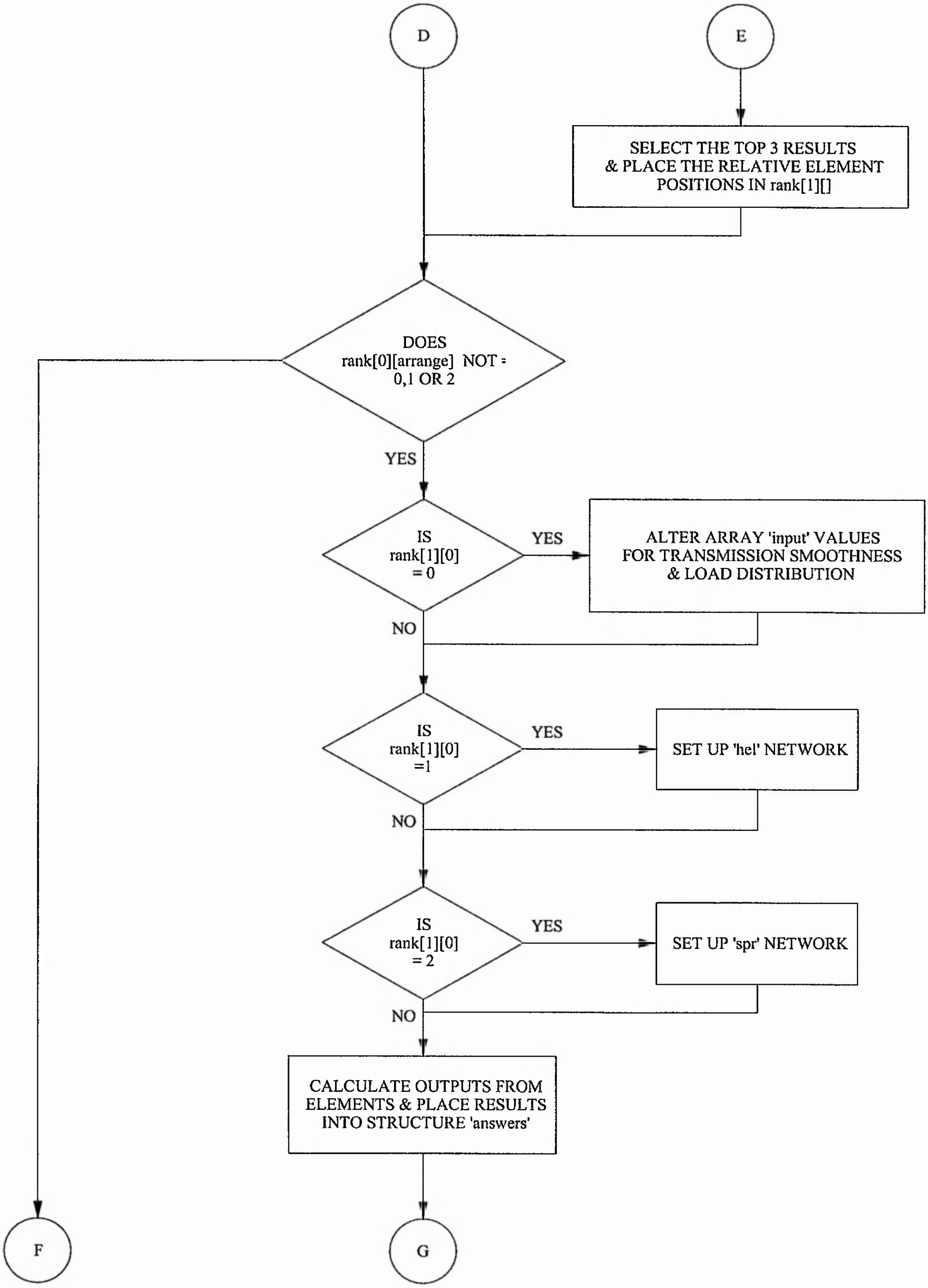


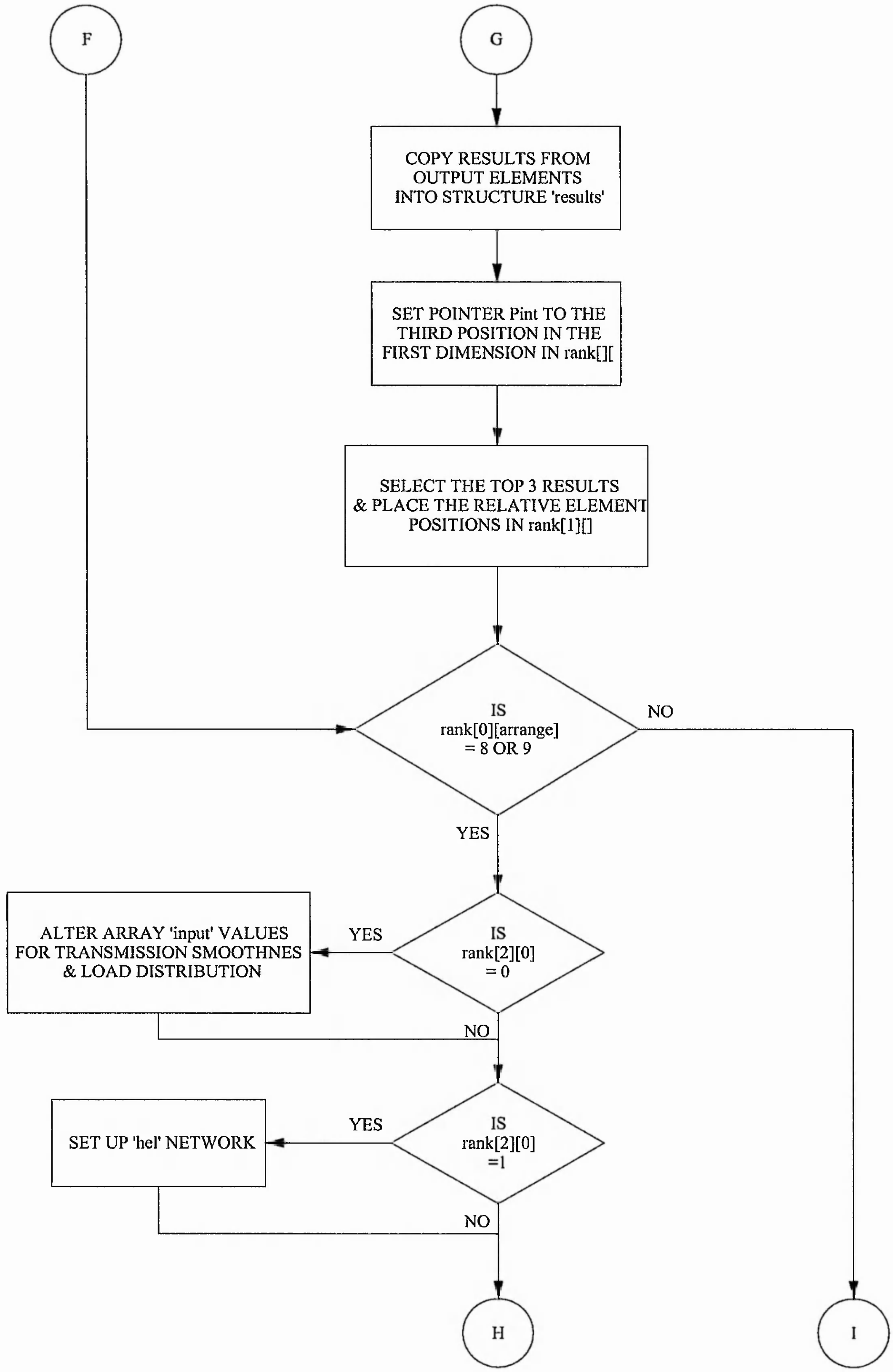
D.2. Conceptual Design Program

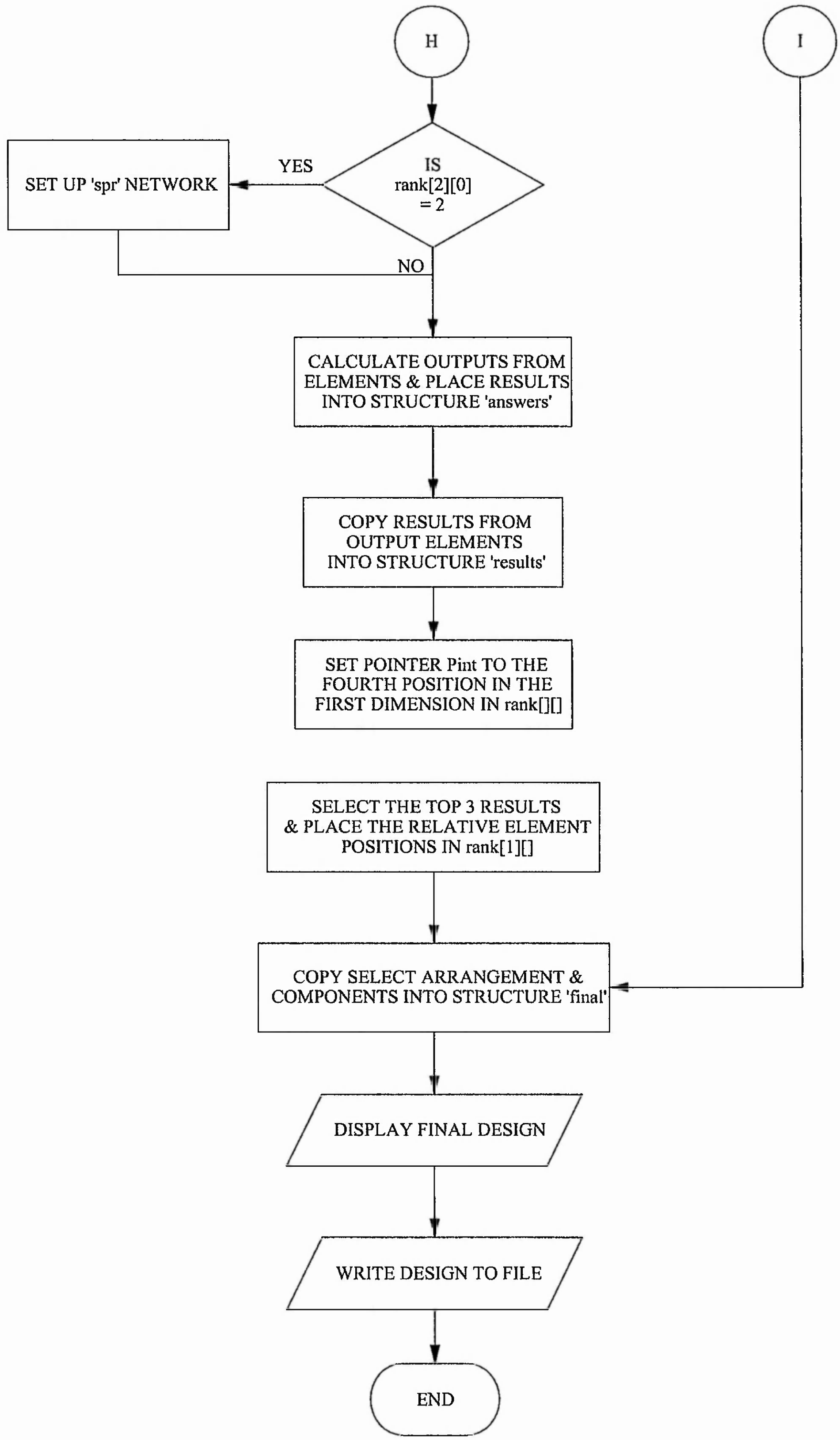



A
B
DISPLAY RESULTS
AND P.D.S
PROMPT USER TO
CHOSE ARRANGEMENT
FROM SELECTION
DOES
SELECTION
= -1
YES
NO
SET THE INPUT FOR THE
COMPONENT NETWORKS
READ PARAMETERS &
CONNECTION WEIGHTS
FOR NETWORK 'all'
IS
rank[0][arrange]
=1
YES
SET FIRST STAGE
COMPONENT TO
BEVEL, 6
NO
IS
rank[0][arrange]
=2
YES
SET FIRST STAGE
COMPONENT TO
WORML, 5
NO
C

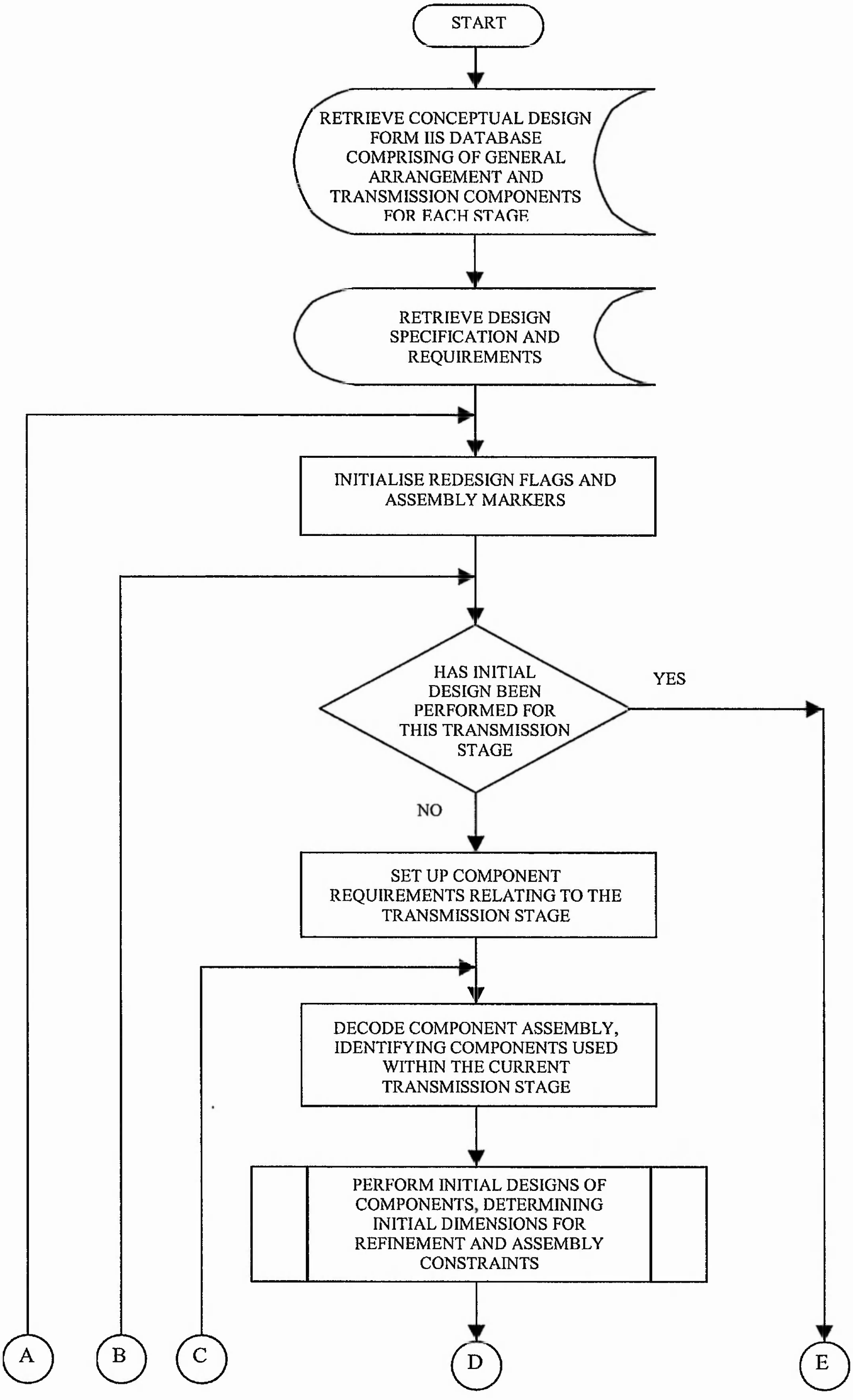

C
IS
rank[0][arrange]
= 5
YES
SET FIRST STAGE
COMPONENT TO
BEVEL, 6
NO
IS
rank[0][arrange]
=6
YES
SET FIRST STAGE
COMPONENT TO
WORML, 5
NO
IS
rank[0][arrange]
= 9
YES
SET FIRST STAGE
COMPONENT TO
BEVEL, 6
NO
IS
rank[0][arrange] = 0
3,4,7 OR 8
YES
CALCULATE OUTPUTS FROM
ELEMENTS & PLACE RESULTS
INTO STRUCTURE 'answers'
NO
COPY RESULTS FROM
OUTPUT ELEMENTS
INTO STRUCTURE 'results'
SET POINTER Pint TO THE
SECOND POSITION IN THE
FIRST DIMENSION IN rank[][
D
E

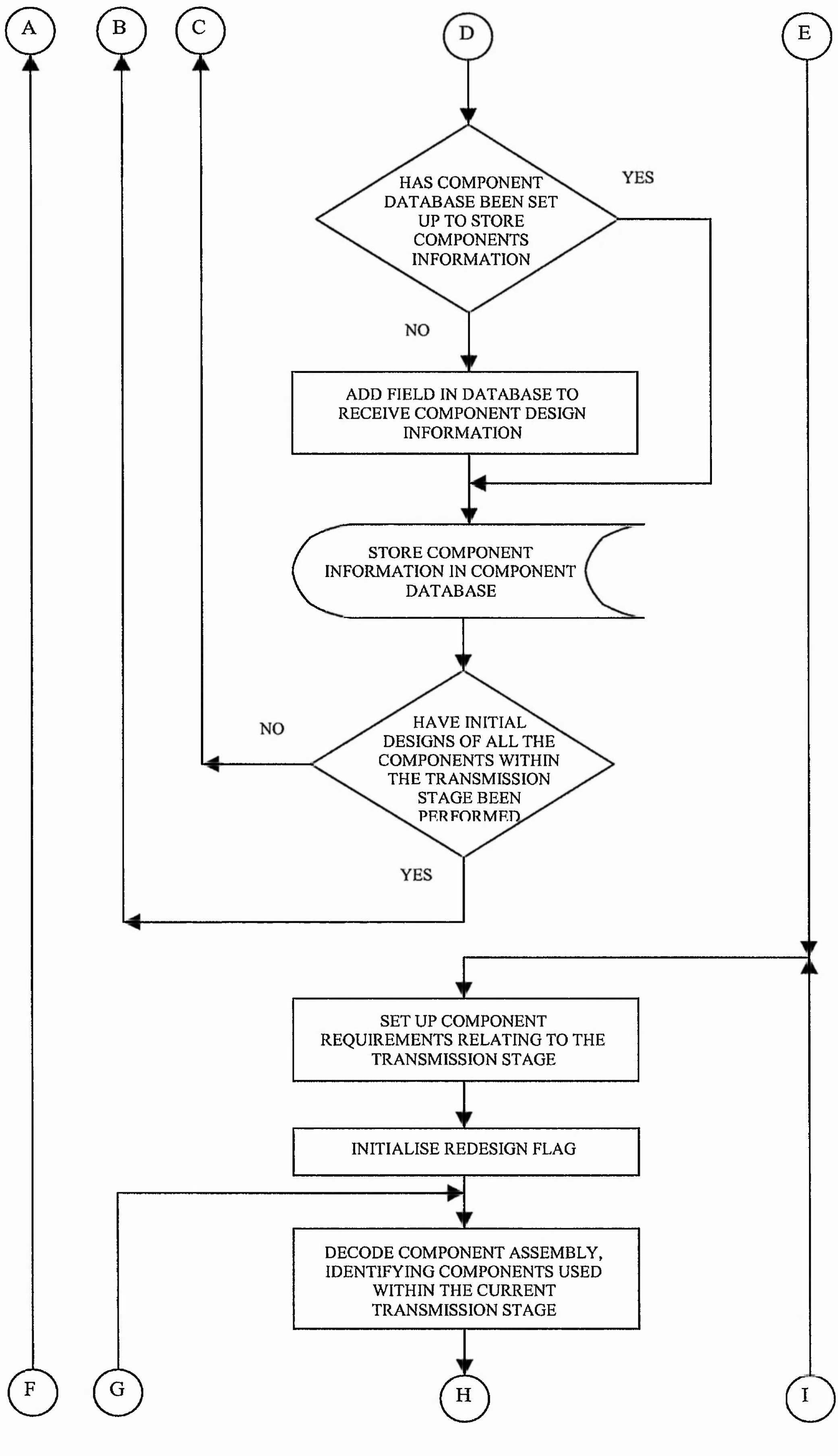

D
E
SELECT THE TOP 3 RESULTS & PLACE THE RELATIVE ELEMENT POSITIONS IN rank[1][]
DOES rank[0][arrange] NOT = 0,1 OR 2
YES
IS rank[1][0] = 0
YES
ALTER ARRAY 'input' VALUES FOR TRANSMISSION SMOOTHNESS & LOAD DISTRIBUTION
NO
IS rank[1][0] =1
YES
SET UP 'hel' NETWORK
NO
IS rank[1][0] = 2
YES
SET UP 'spr' NETWORK
NO
CALCULATE OUTPUTS FROM ELEMENTS & PLACE RESULTS INTO STRUCTURE 'answers'
F
G

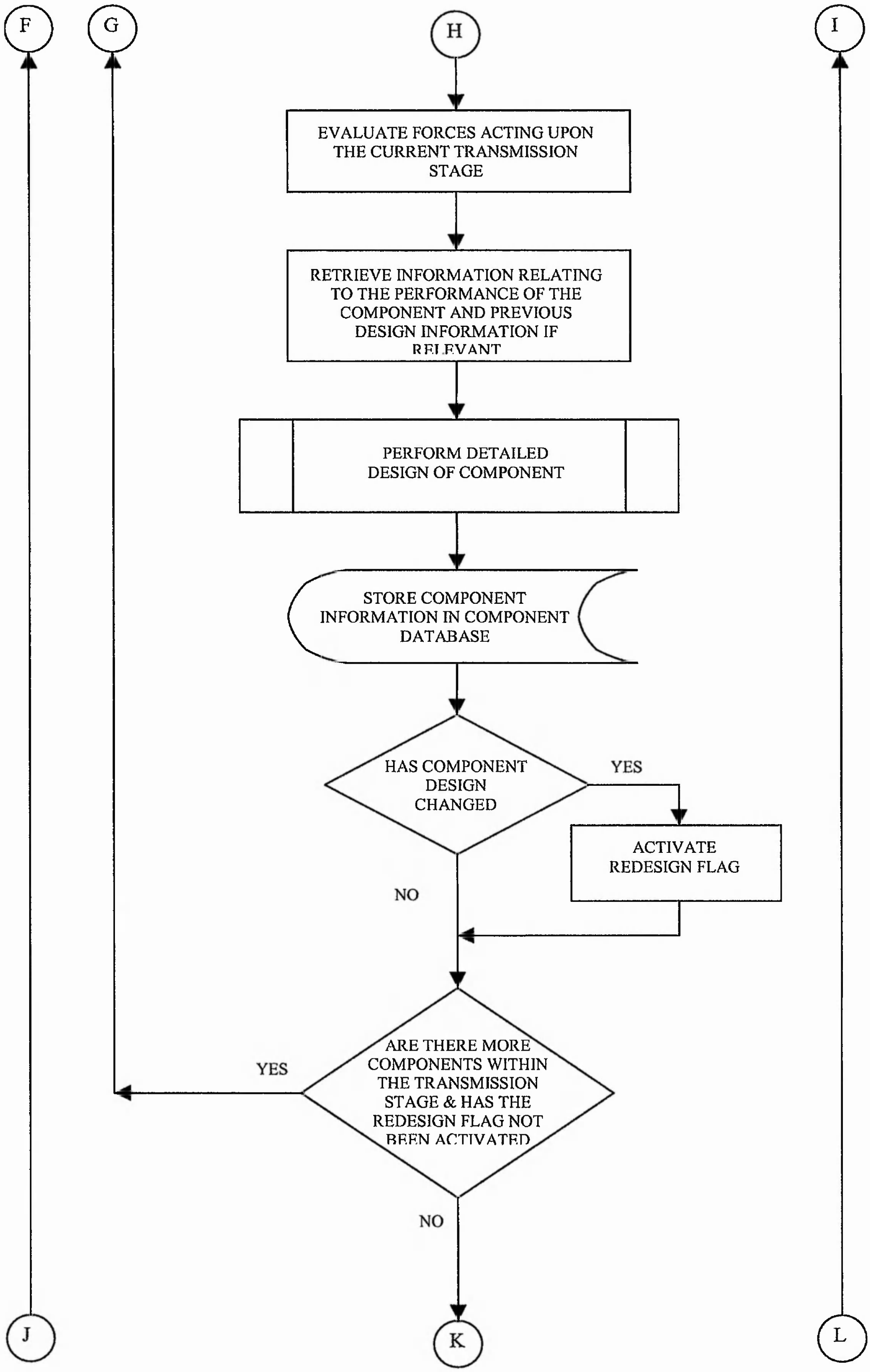

F
G
COPY RESULTS FROM OUTPUT ELEMENTS INTO STRUCTURE 'results'
SET POINTER Pint TO THE THIRD POSITION IN THE FIRST DIMENSION IN rank[][
SELECT THE TOP 3 RESULTS & PLACE THE RELATIVE ELEMENT POSITIONS IN rank[1][]
IS rank[0][arrange] = 8 OR 9
NO
YES
ALTER ARRAY 'input' VALUES FOR TRANSMISSION SMOOTHNES & LOAD DISTRIBUTION
YES
IS rank[2][0] = 0
NO
SET UP 'hel' NETWORK
YES
IS rank[2][0] =1
NO
H
I

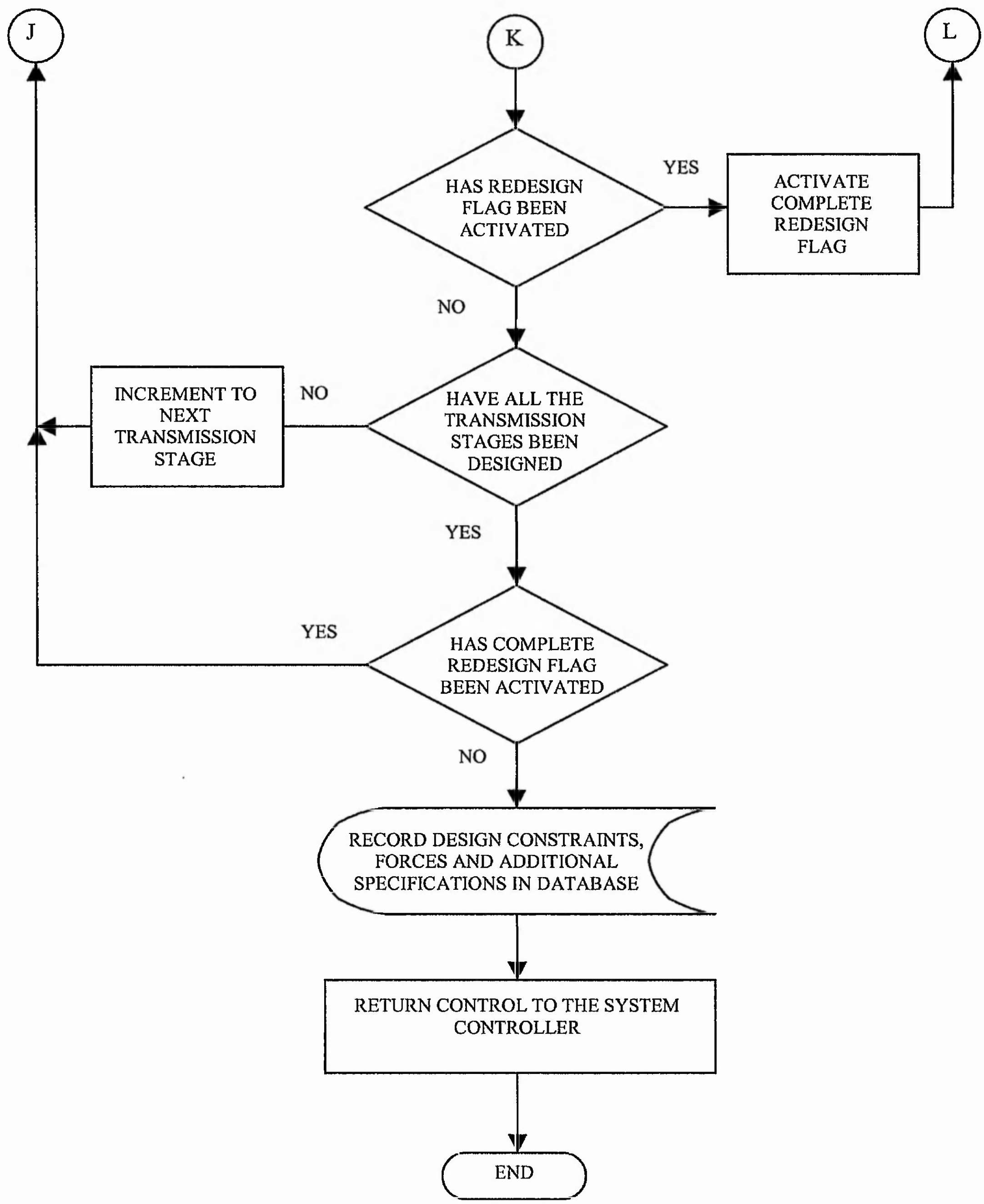

H
I
IS
rank[2][0]
= 2
YES
SET UP 'spr' NETWORK
NO
CALCULATE OUTPUTS FROM ELEMENTS & PLACE RESULTS INTO STRUCTURE 'answers'
COPY RESULTS FROM OUTPUT ELEMENTS INTO STRUCTURE 'results'
SET POINTER Pint TO THE FOURTH POSITION IN THE FIRST DIMENSION IN rank[][]
SELECT THE TOP 3 RESULTS & PLACE THE RELATIVE ELEMENT POSITIONS IN rank[1][]
COPY SELECT ARRANGEMENT & COMPONENTS INTO STRUCTURE 'final'
DISPLAY FINAL DESIGN
WRITE DESIGN TO FILE
END

D.3. Detail Design Control Program

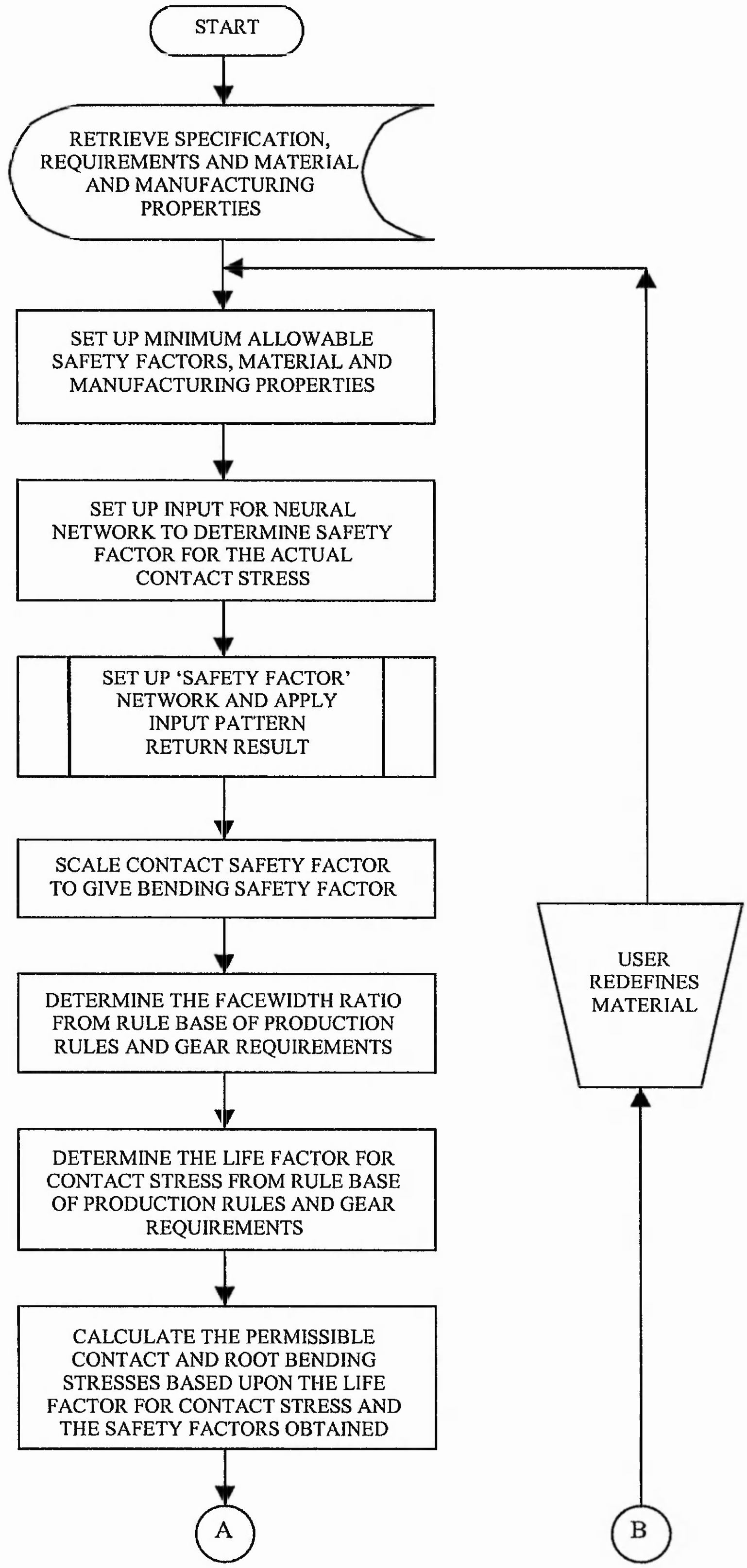


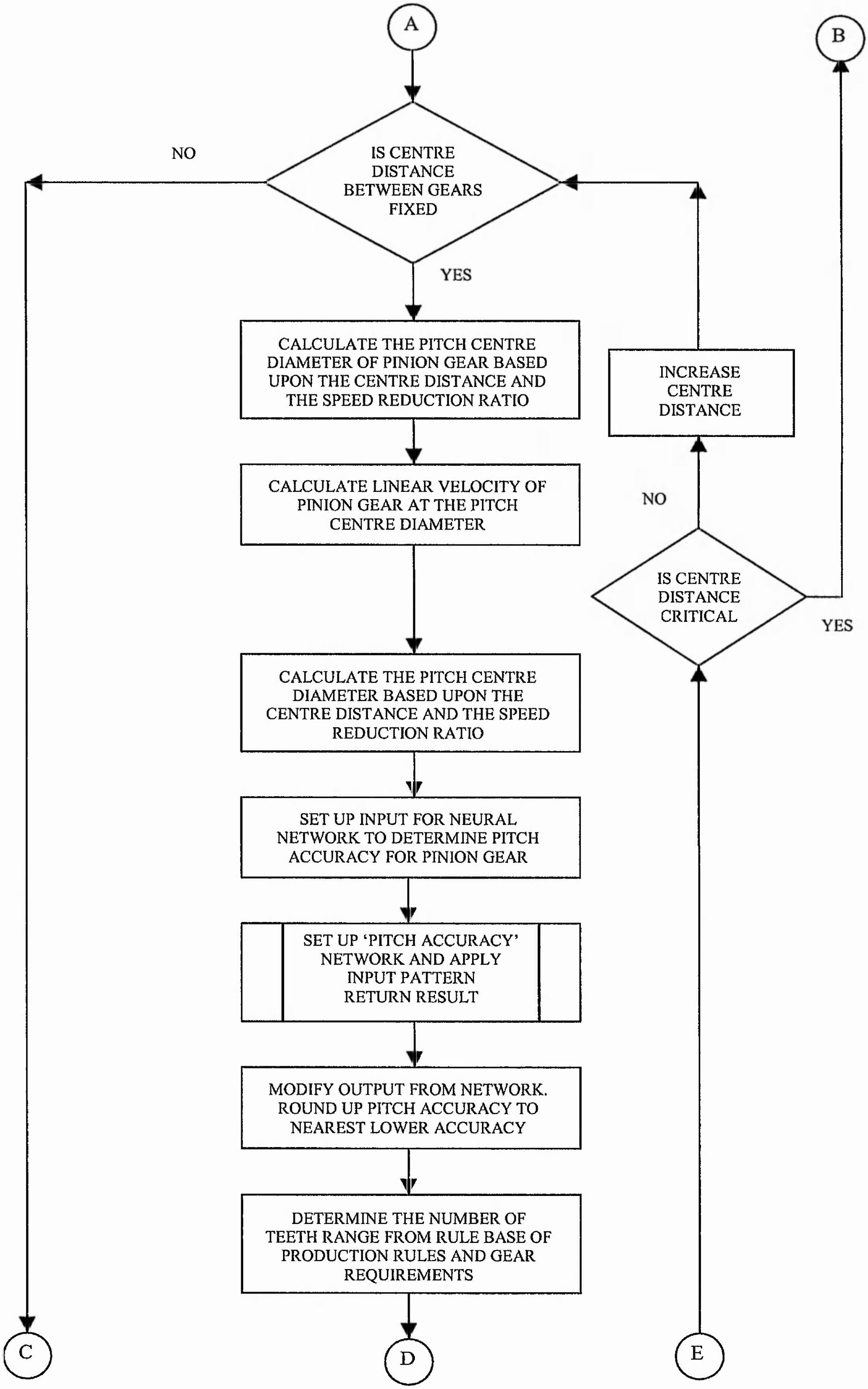

A
B
C
D
E
HAS COMPONENT DATABASE BEEN SET UP TO STORE COMPONENTS INFORMATION
YES
NO
ADD FIELD IN DATABASE TO RECEIVE COMPONENT DESIGN INFORMATION
STORE COMPONENT INFORMATION IN COMPONENT DATABASE
NO
HAVE INITIAL DESIGNS OF ALL THE COMPONENTS WITHIN THE TRANSMISSION STAGE BEEN PERFORMED
YES
SET UP COMPONENT REQUIREMENTS RELATING TO THE TRANSMISSION STAGE
INITIALISE REDESIGN FLAG
DECODE COMPONENT ASSEMBLY, IDENTIFYING COMPONENTS USED WITHIN THE CURRENT TRANSMISSION STAGE
F
G
H
I

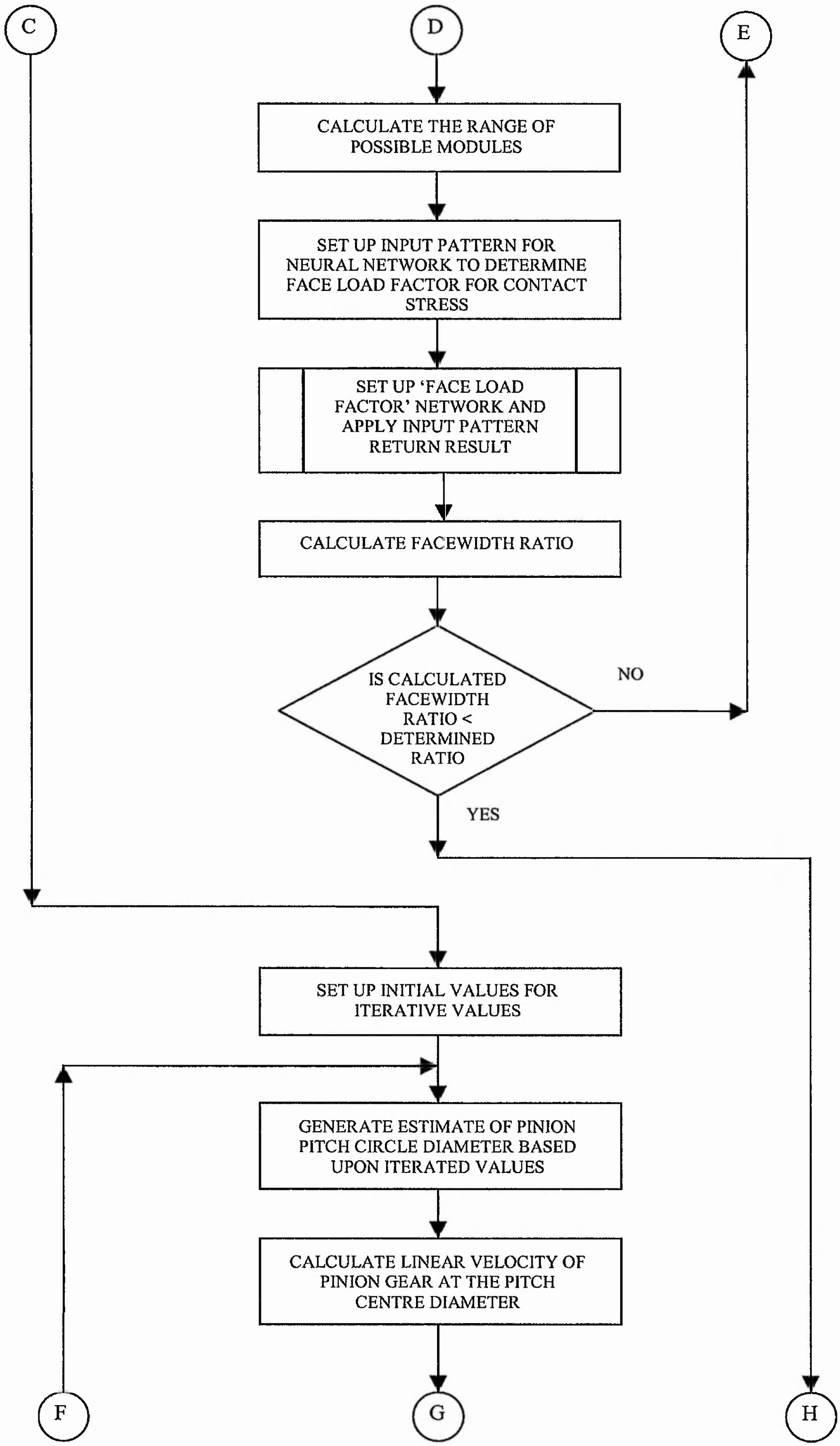

F
G
H
I
EVALUATE FORCES ACTING UPON THE CURRENT TRANSMISSION STAGE
RETRIEVE INFORMATION RELATING TO THE PERFORMANCE OF THE COMPONENT AND PREVIOUS DESIGN INFORMATION IF RELEVANT
PERFORM DETAILED DESIGN OF COMPONENT
STORE COMPONENT INFORMATION IN COMPONENT DATABASE
HAS COMPONENT DESIGN CHANGED
YES
ACTIVATE REDESIGN FLAG
NO
ARE THERE MORE COMPONENTS WITHIN THE TRANSMISSION STAGE & HAS THE REDESIGN FLAG NOT BEEN ACTIVATED
YES
NO
J
K
L

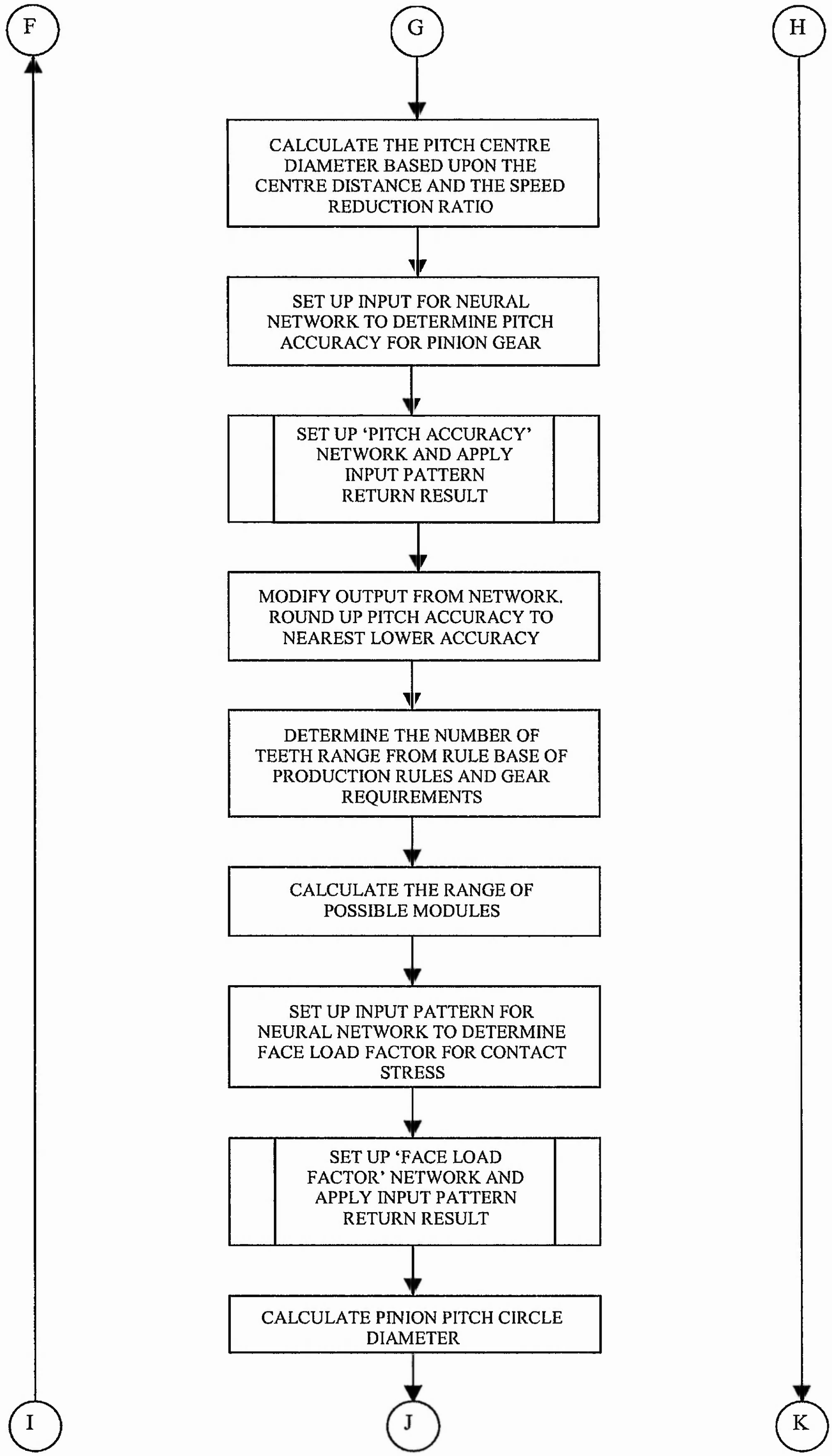

J
K
L
HAS REDESIGN FLAG BEEN ACTIVATED
YES
ACTIVATE COMPLETE REDESIGN FLAG
NO
HAVE ALL THE TRANSMISSION STAGES BEEN DESIGNED
NO
INCREMENT TO NEXT TRANSMISSION STAGE
YES
HAS COMPLETE REDESIGN FLAG BEEN ACTIVATED
YES
NO
RECORD DESIGN CONSTRAINTS, FORCES AND ADDITIONAL SPECIFICATIONS IN DATABASE
RETURN CONTROL TO THE SYSTEM CONTROLLER
END

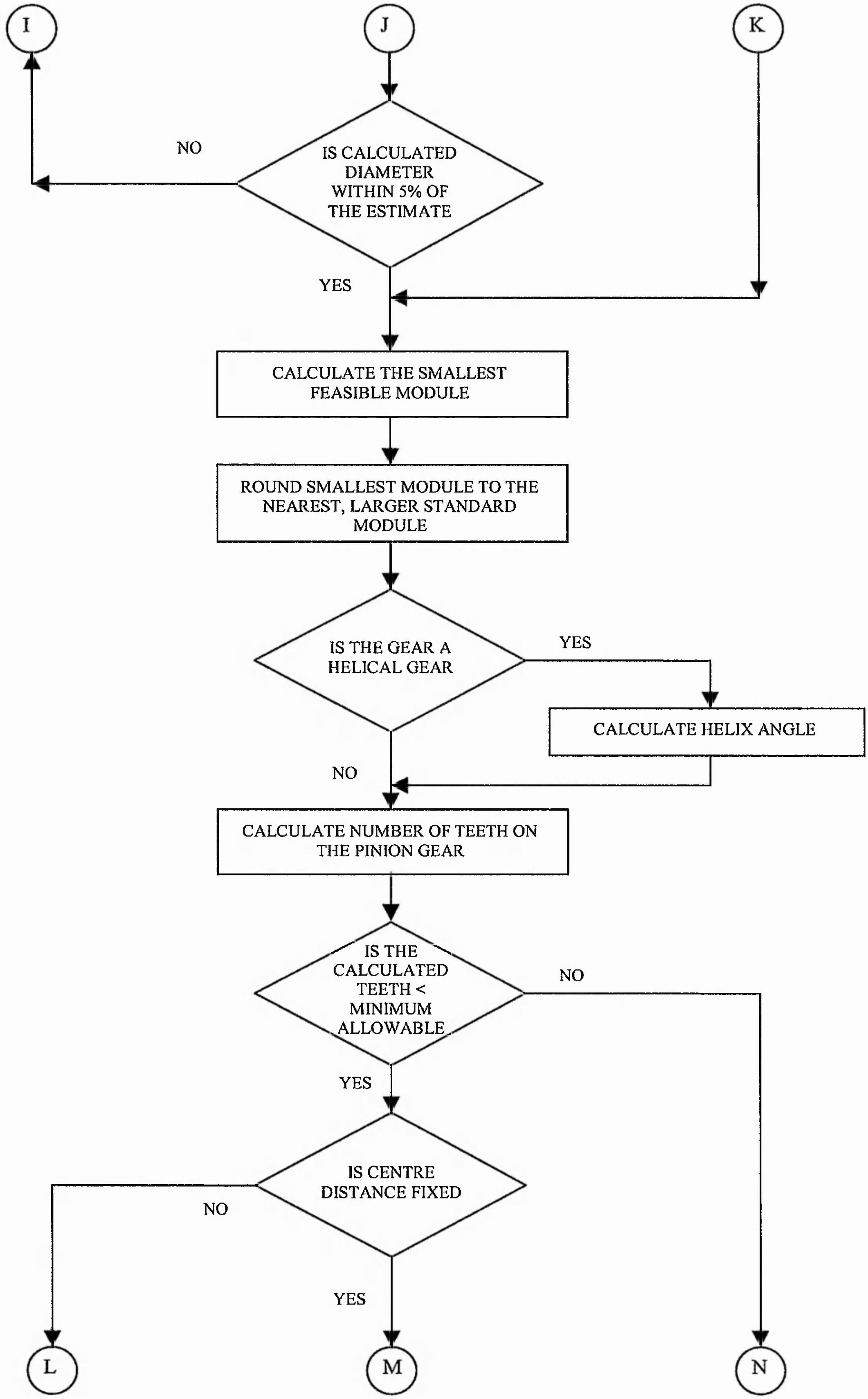
D.4. Initial Gear Design

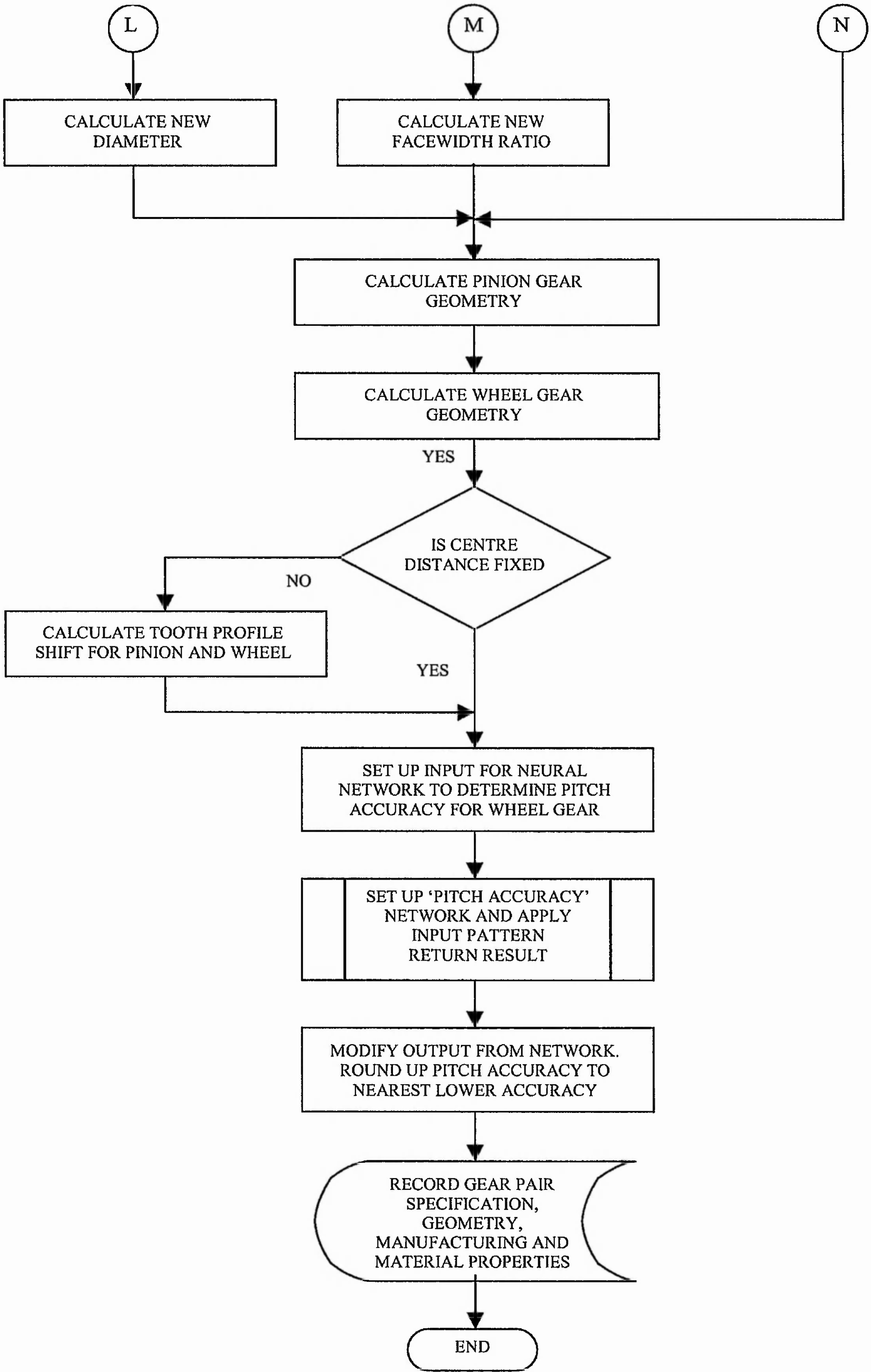



A
B
NO
IS CENTRE DISTANCE BETWEEN GEARS FIXED
YES
CALCULATE THE PITCH CENTRE DIAMETER OF PINION GEAR BASED UPON THE CENTRE DISTANCE AND THE SPEED REDUCTION RATIO
INCREASE CENTRE DISTANCE
CALCULATE LINEAR VELOCITY OF PINION GEAR AT THE PITCH CENTRE DIAMETER
NO
IS CENTRE DISTANCE CRITICAL
YES
CALCULATE THE PITCH CENTRE DIAMETER BASED UPON THE CENTRE DISTANCE AND THE SPEED REDUCTION RATIO
SET UP INPUT FOR NEURAL NETWORK TO DETERMINE PITCH ACCURACY FOR PINION GEAR
SET UP 'PITCH ACCURACY' NETWORK AND APPLY INPUT PATTERN RETURN RESULT
MODIFY OUTPUT FROM NETWORK. ROUND UP PITCH ACCURACY TO NEAREST LOWER ACCURACY
DETERMINE THE NUMBER OF TEETH RANGE FROM RULE BASE OF PRODUCTION RULES AND GEAR REQUIREMENTS
C
D
E

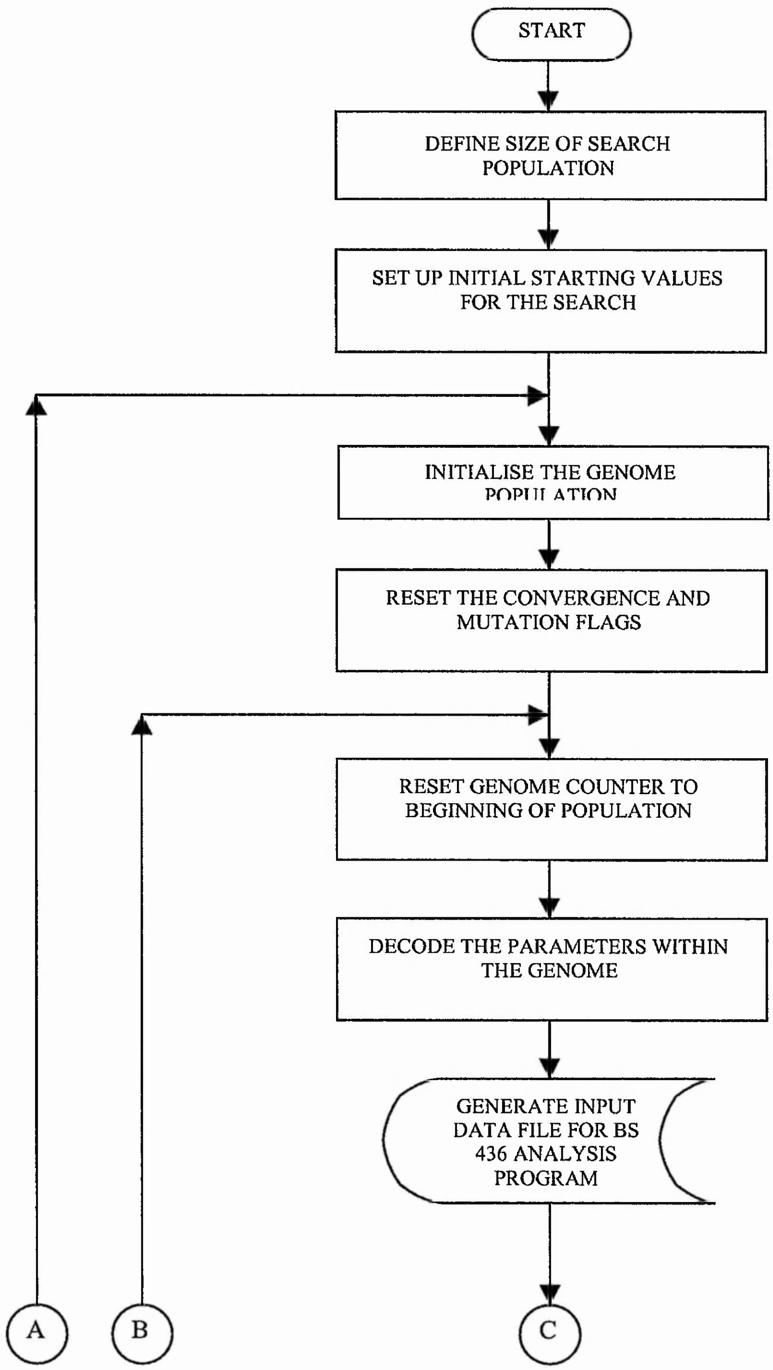

C
D
E
CALCULATE THE RANGE OF POSSIBLE MODULES
SET UP INPUT PATTERN FOR NEURAL NETWORK TO DETERMINE FACE LOAD FACTOR FOR CONTACT STRESS
SET UP 'FACE LOAD FACTOR' NETWORK AND APPLY INPUT PATTERN RETURN RESULT
CALCULATE FACEWIDTH RATIO
IS CALCULATED FACEWIDTH RATIO < DETERMINED RATIO
NO
YES
SET UP INITIAL VALUES FOR ITERATIVE VALUES
GENERATE ESTIMATE OF PINION PITCH CIRCLE DIAMETER BASED UPON ITERATED VALUES
CALCULATE LINEAR VELOCITY OF PINION GEAR AT THE PITCH CENTRE DIAMETER
F
G
H

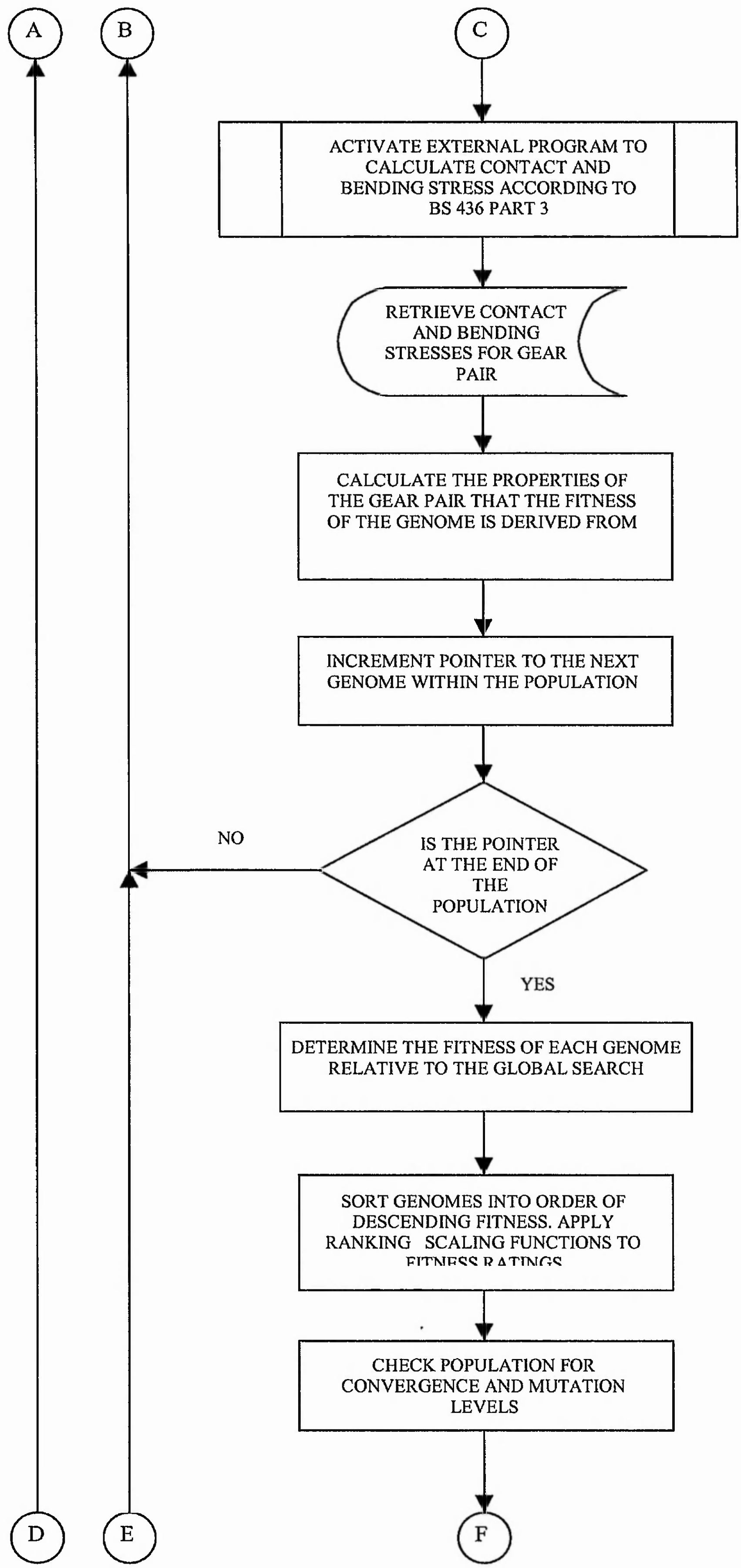

F
G
H
CALCULATE THE PITCH CENTRE DIAMETER BASED UPON THE CENTRE DISTANCE AND THE SPEED REDUCTION RATIO
SET UP INPUT FOR NEURAL NETWORK TO DETERMINE PITCH ACCURACY FOR PINION GEAR
SET UP 'PITCH ACCURACY' NETWORK AND APPLY INPUT PATTERN RETURN RESULT
MODIFY OUTPUT FROM NETWORK. ROUND UP PITCH ACCURACY TO NEAREST LOWER ACCURACY
DETERMINE THE NUMBER OF TEETH RANGE FROM RULE BASE OF PRODUCTION RULES AND GEAR REQUIREMENTS
CALCULATE THE RANGE OF POSSIBLE MODULES
SET UP INPUT PATTERN FOR NEURAL NETWORK TO DETERMINE FACE LOAD FACTOR FOR CONTACT STRESS
SET UP 'FACE LOAD FACTOR' NETWORK AND APPLY INPUT PATTERN RETURN RESULT
CALCULATE PINION PITCH CIRCLE DIAMETER
I
J
K

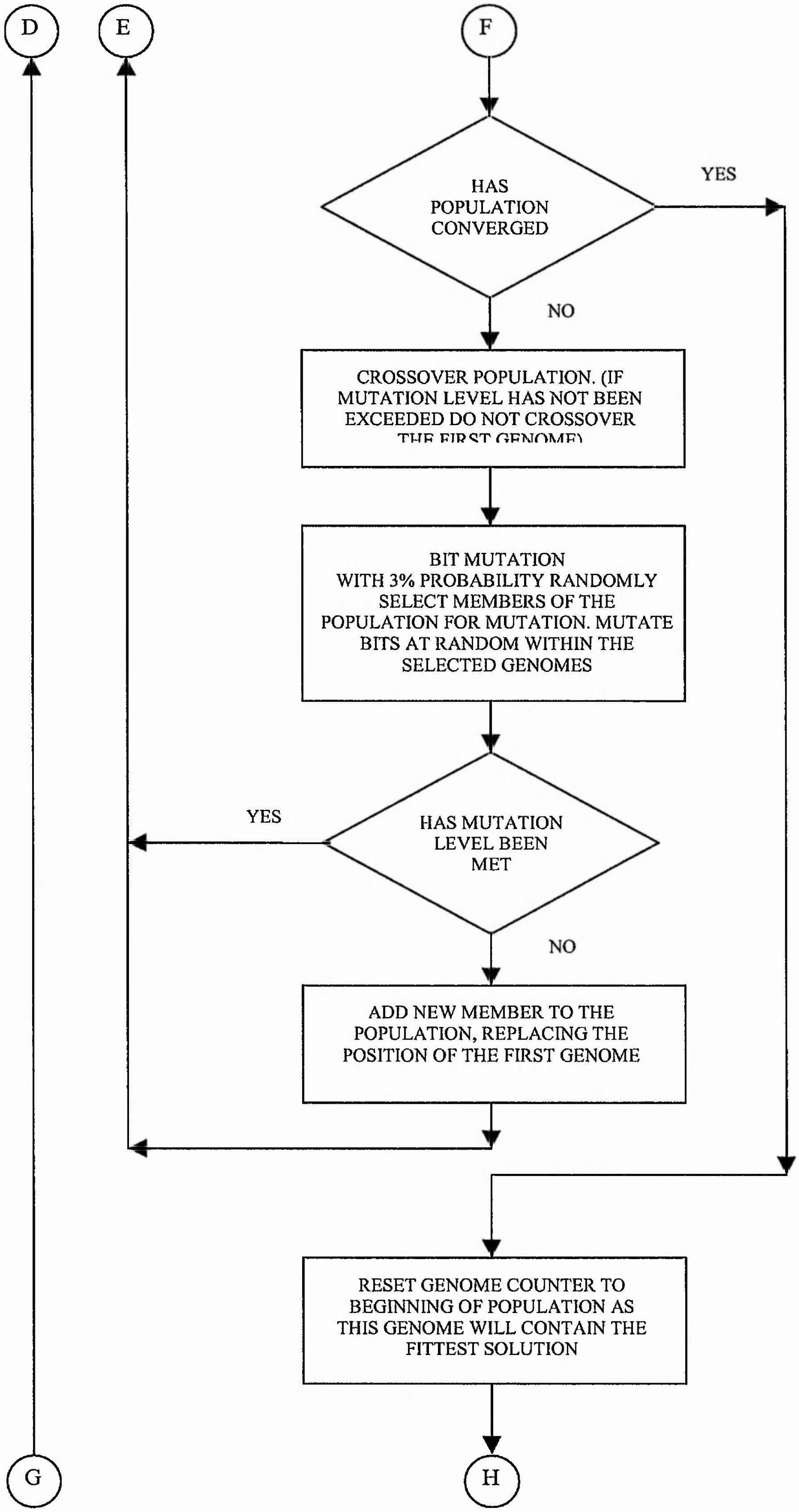

I
J
K
NO
IS CALCULATED DIAMETER WITHIN 5% OF THE ESTIMATE
YES
CALCULATE THE SMALLEST FEASIBLE MODULE
ROUND SMALLEST MODULE TO THE NEAREST, LARGER STANDARD MODULE
IS THE GEAR A HELICAL GEAR
YES
CALCULATE HELIX ANGLE
NO
CALCULATE NUMBER OF TEETH ON THE PINION GEAR
IS THE CALCULATED TEETH < MINIMUM ALLOWABLE
NO
YES
IS CENTRE DISTANCE FIXED
NO
YES
L
M
N

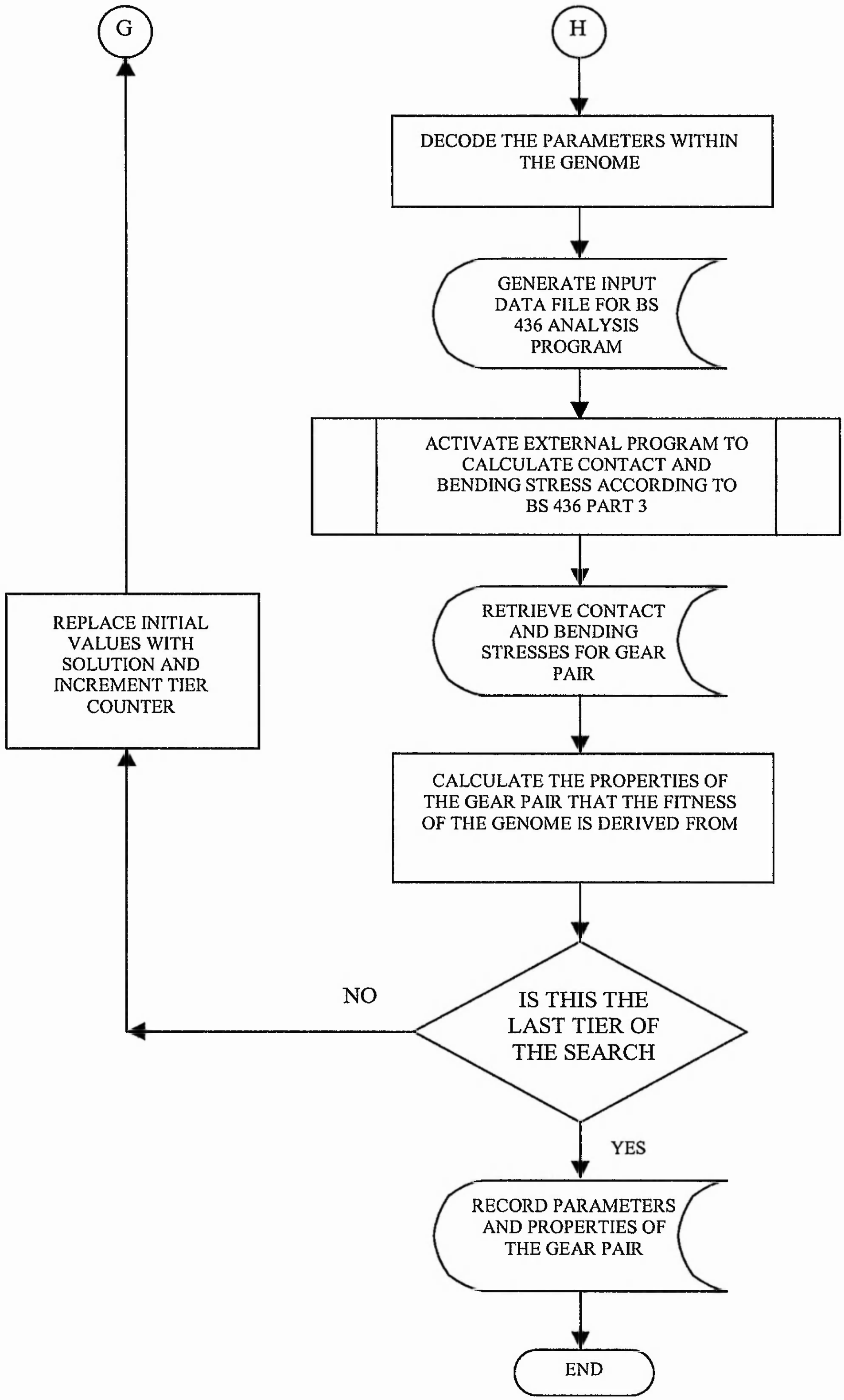

L
M
N
CALCULATE NEW DIAMETER
CALCULATE NEW FACEWIDTH RATIO
CALCULATE PINION GEAR GEOMETRY
CALCULATE WHEEL GEAR GEOMETRY
YES
IS CENTRE DISTANCE FIXED
NO
CALCULATE TOOTH PROFILE SHIFT FOR PINION AND WHEEL
YES
SET UP INPUT FOR NEURAL NETWORK TO DETERMINE PITCH ACCURACY FOR WHEEL GEAR
SET UP 'PITCH ACCURACY' NETWORK AND APPLY INPUT PATTERN RETURN RESULT
MODIFY OUTPUT FROM NETWORK. ROUND UP PITCH ACCURACY TO NEAREST LOWER ACCURACY
RECORD GEAR PAIR SPECIFICATION, GEOMETRY, MANUFACTURING AND MATERIAL PROPERTIES
END

D.5 Gear Optimisation

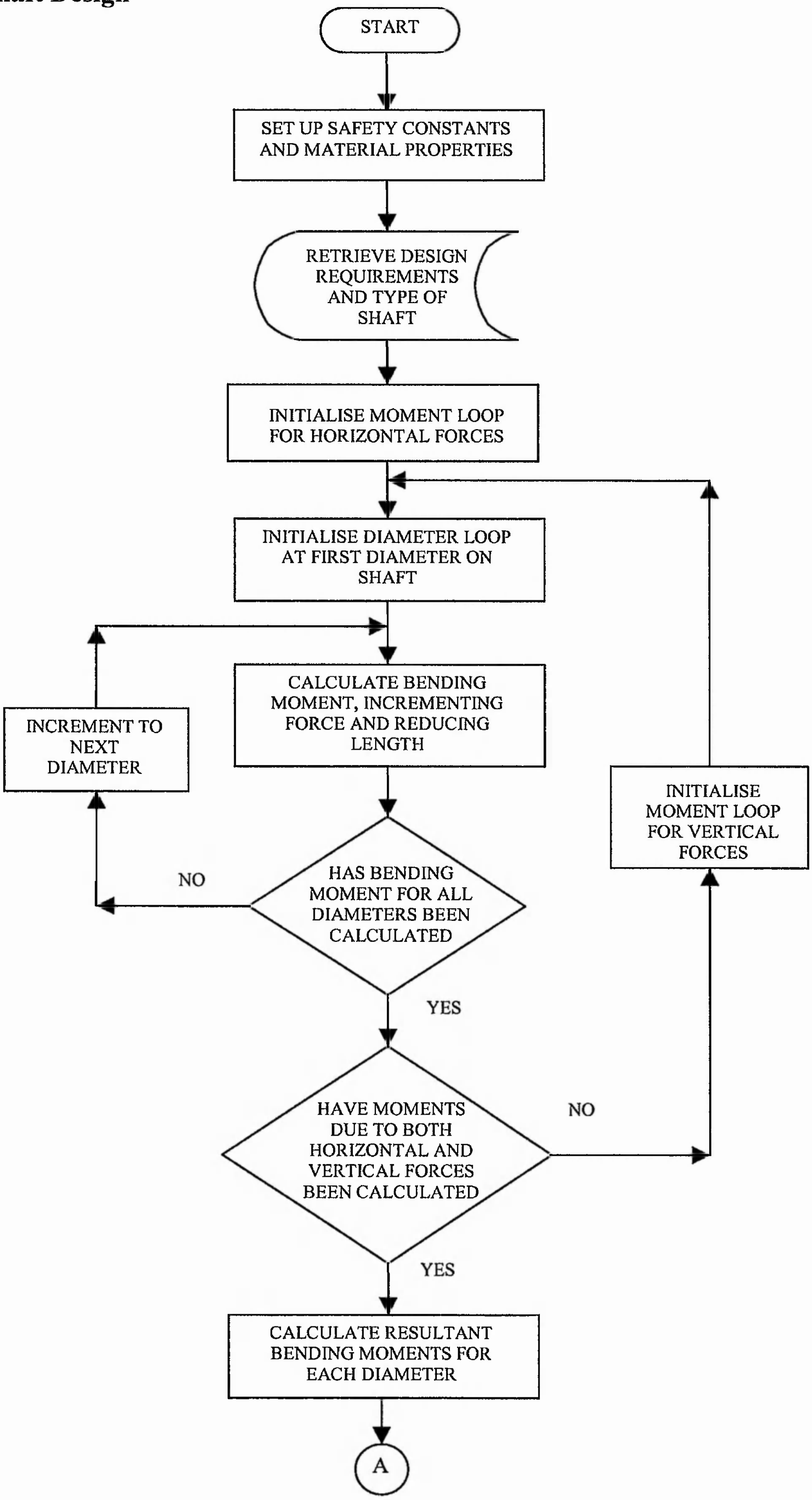


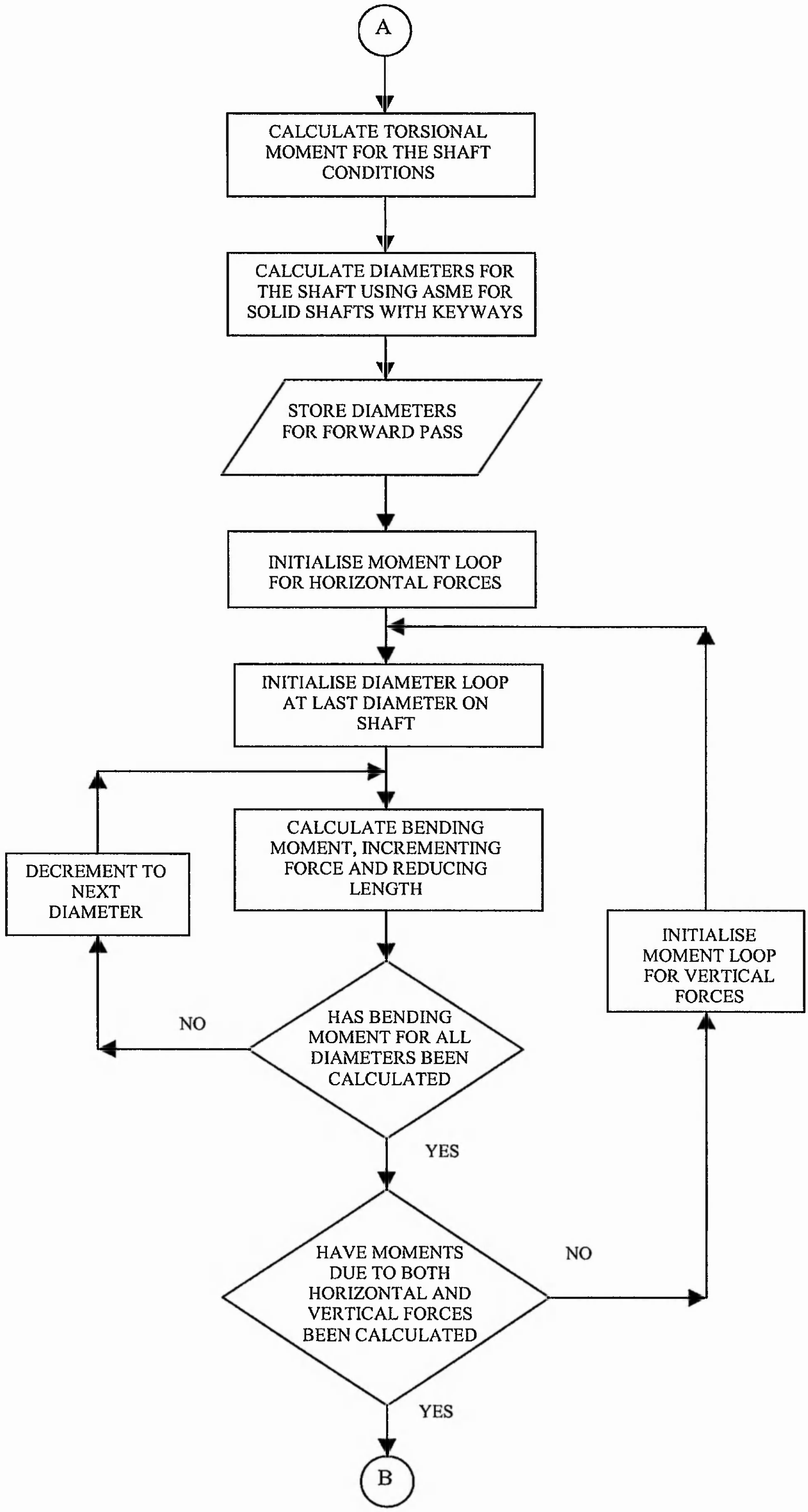

A
B
C
ACTIVATE EXTERNAL PROGRAM TO CALCULATE CONTACT AND BENDING STRESS ACCORDING TO BS 436 PART 3
RETRIEVE CONTACT AND BENDING STRESSES FOR GEAR PAIR
CALCULATE THE PROPERTIES OF THE GEAR PAIR THAT THE FITNESS OF THE GENOME IS DERIVED FROM
INCREMENT POINTER TO THE NEXT GENOME WITHIN THE POPULATION
NO
IS THE POINTER AT THE END OF THE POPULATION
YES
DETERMINE THE FITNESS OF EACH GENOME RELATIVE TO THE GLOBAL SEARCH
SORT GENOMES INTO ORDER OF DESCENDING FITNESS. APPLY RANKING SCALING FUNCTIONS TO FITNESS RATINGS
CHECK POPULATION FOR CONVERGENCE AND MUTATION LEVELS
D
E
F

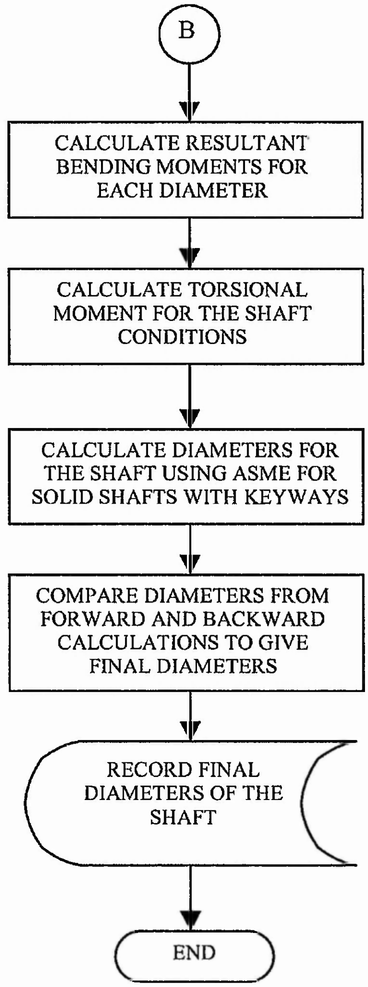

D
E
F
HAS POPULATION CONVERGED
YES
NO
CROSSOVER POPULATION. (IF MUTATION LEVEL HAS NOT BEEN EXCEEDED DO NOT CROSSOVER THE FIRST GENOME)
BIT MUTATION WITH 3% PROBABILITY RANDOMLY SELECT MEMBERS OF THE POPULATION FOR MUTATION. MUTATE BITS AT RANDOM WITHIN THE SELECTED GENOMES
YES
HAS MUTATION LEVEL BEEN MET
NO
ADD NEW MEMBER TO THE POPULATION, REPLACING THE POSITION OF THE FIRST GENOME
RESET GENOME COUNTER TO BEGINNING OF POPULATION AS THIS GENOME WILL CONTAIN THE FITTEST SOLUTION
G
H


G
H
DECODE THE PARAMETERS WITHIN THE GENOME
GENERATE INPUT DATA FILE FOR BS 436 ANALYSIS PROGRAM
ACTIVATE EXTERNAL PROGRAM TO CALCULATE CONTACT AND BENDING STRESS ACCORDING TO BS 436 PART 3
RETRIEVE CONTACT AND BENDING STRESSES FOR GEAR PAIR
CALCULATE THE PROPERTIES OF THE GEAR PAIR THAT THE FITNESS OF THE GENOME IS DERIVED FROM
IS THIS THE LAST TIER OF THE SEARCH
NO
REPLACE INITIAL VALUES WITH SOLUTION AND INCREMENT TIER COUNTER
YES
RECORD PARAMETERS AND PROPERTIES OF THE GEAR PAIR
END

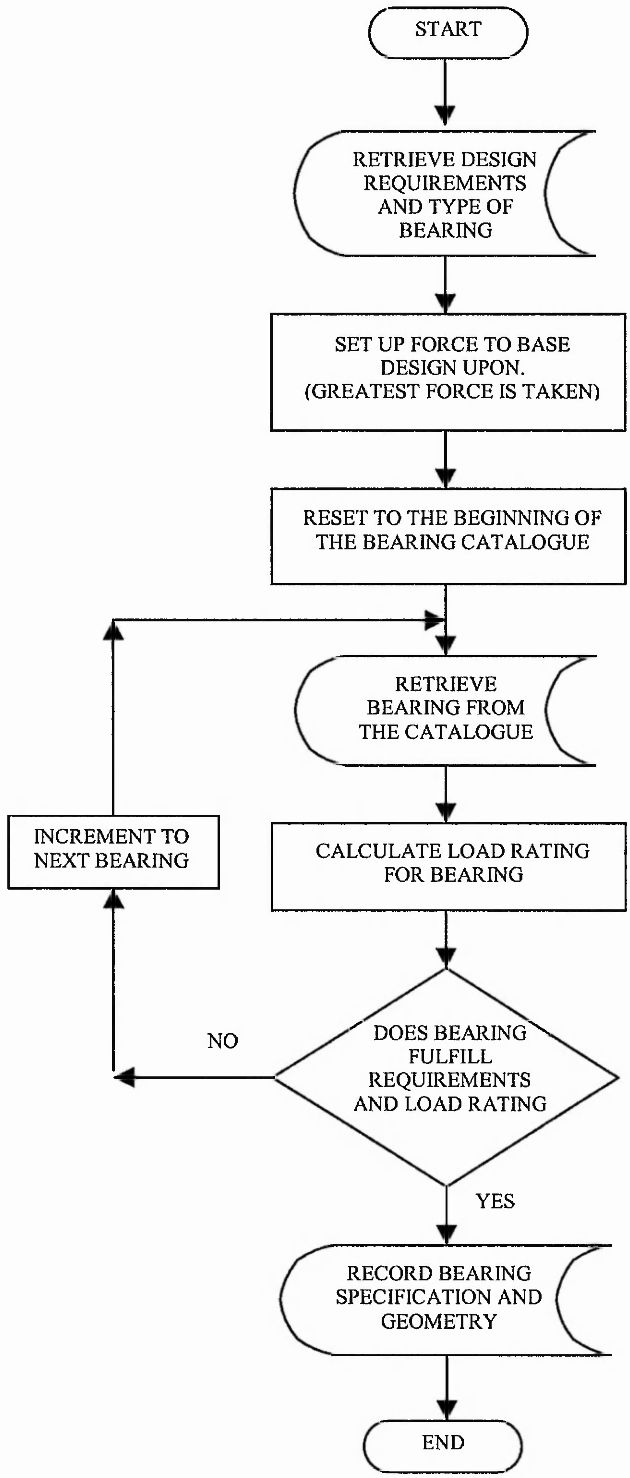
D.6 Shaft Design



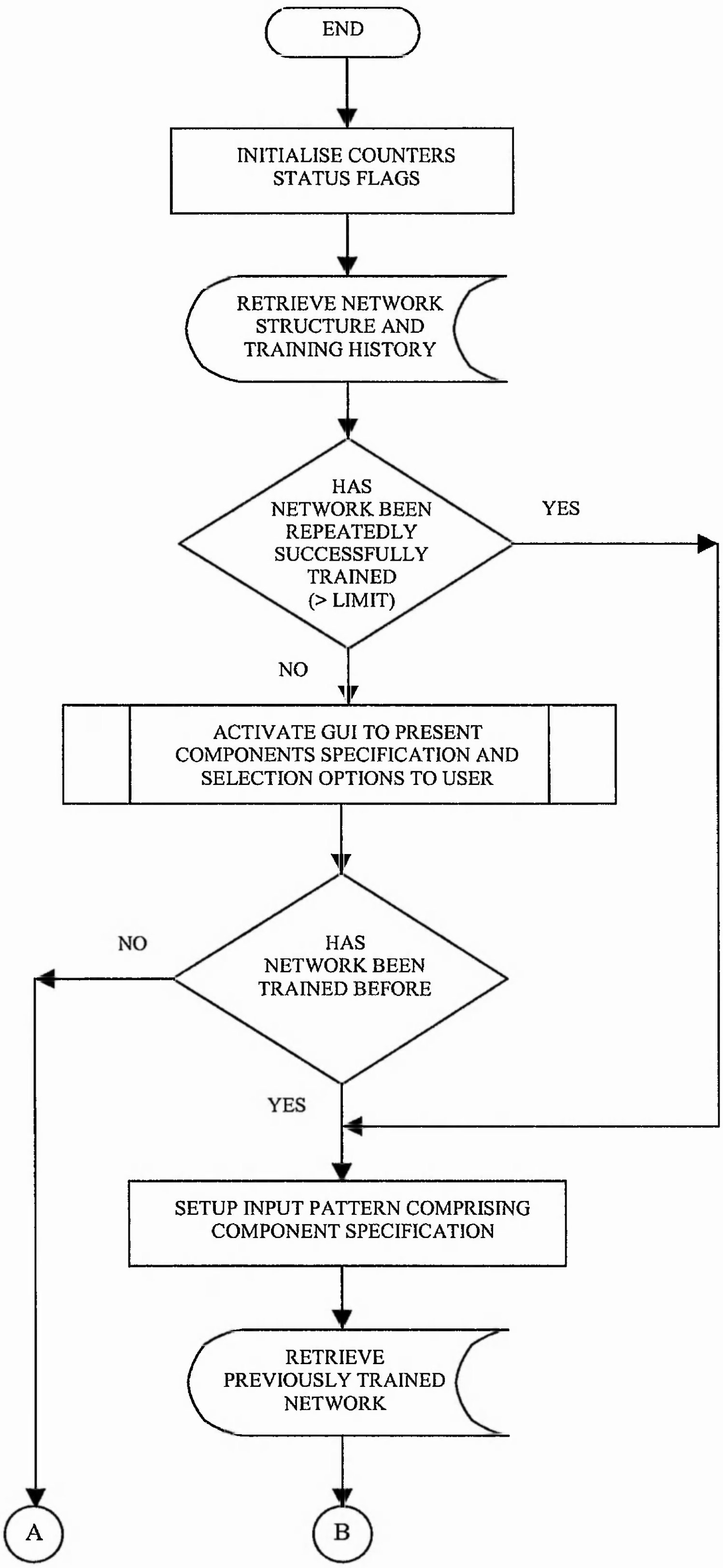

A
CALCULATE TORSIONAL MOMENT FOR THE SHAFT CONDITIONS
CALCULATE DIAMETERS FOR THE SHAFT USING ASME FOR SOLID SHAFTS WITH KEYWAYS
STORE DIAMETERS FOR FORWARD PASS
INITIALISE MOMENT LOOP FOR HORIZONTAL FORCES
INITIALISE DIAMETER LOOP AT LAST DIAMETER ON SHAFT
CALCULATE BENDING MOMENT, INCREMENTING FORCE AND REDUCING LENGTH
DECREMENT TO NEXT DIAMETER
HAS BENDING MOMENT FOR ALL DIAMETERS BEEN CALCULATED
NO
YES
INITIALISE MOMENT LOOP FOR VERTICAL FORCES
HAVE MOMENTS DUE TO BOTH HORIZONTAL AND VERTICAL FORCES BEEN CALCULATED
NO
YES
B

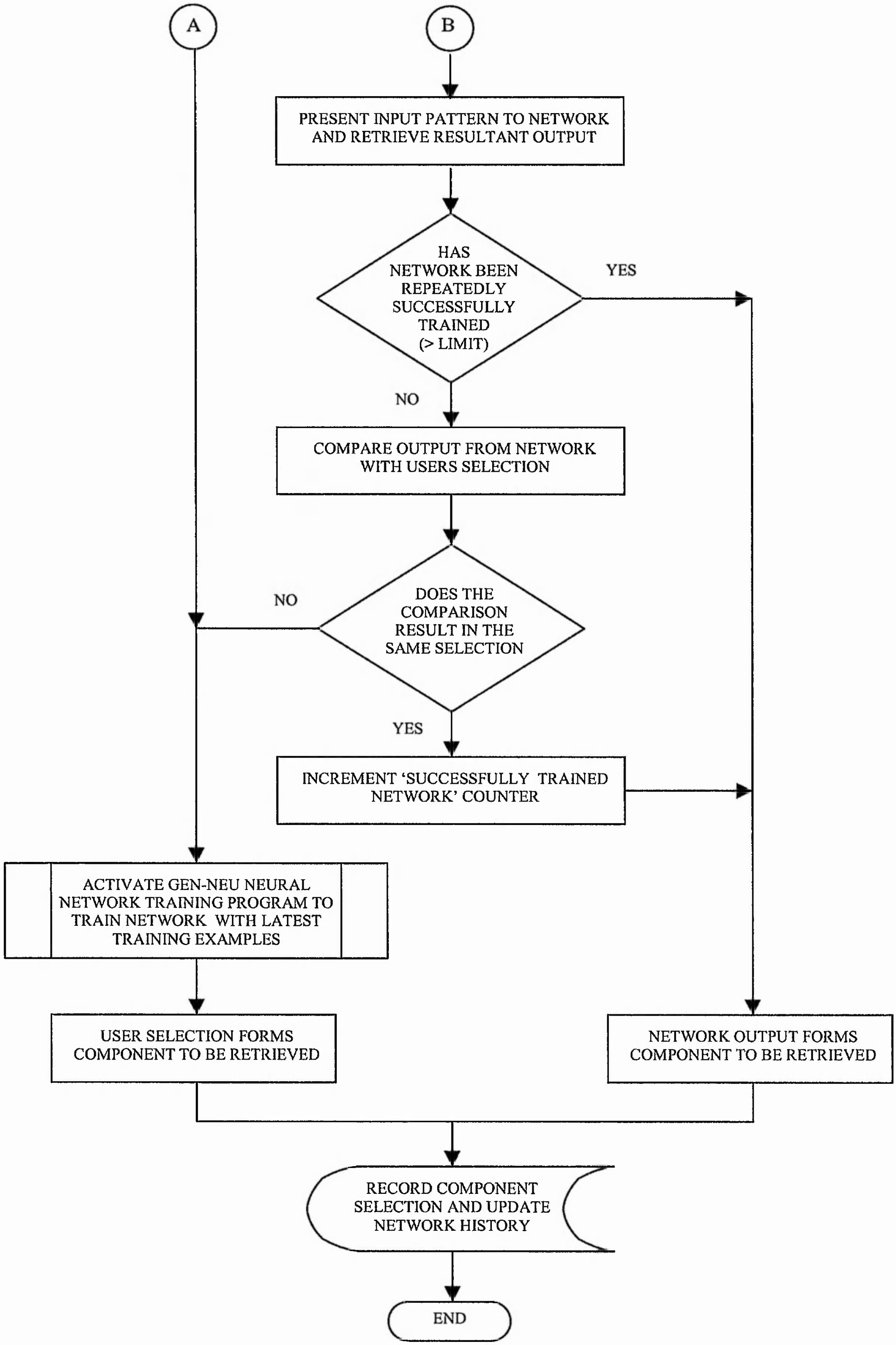

B
CALCULATE RESULTANT BENDING MOMENTS FOR EACH DIAMETER
CALCULATE TORSIONAL MOMENT FOR THE SHAFT CONDITIONS
CALCULATE DIAMETERS FOR THE SHAFT USING ASME FOR SOLID SHAFTS WITH KEYWAYS
COMPARE DIAMETERS FROM FORWARD AND BACKWARD CALCULATIONS TO GIVE FINAL DIAMETERS
RECORD FINAL DIAMETERS OF THE SHAFT
END

D.7 Bearing Selection Catalogue Program



D. 8 Design Selection and Retrieval




A
B
PRESENT INPUT PATTERN TO NETWORK AND RETRIEVE RESULTANT OUTPUT
HAS NETWORK BEEN REPEATEDLY SUCCESSFULLY TRAINED (> LIMIT)
YES
NO
COMPARE OUTPUT FROM NETWORK WITH USERS SELECTION
DOES THE COMPARISON RESULT IN THE SAME SELECTION
NO
YES
INCREMENT 'SUCCESSFULLY TRAINED NETWORK' COUNTER
ACTIVATE GEN-NEU NEURAL NETWORK TRAINING PROGRAM TO TRAIN NETWORK WITH LATEST TRAINING EXAMPLES
USER SELECTION FORMS COMPONENT TO BE RETRIEVED
NETWORK OUTPUT FORMS COMPONENT TO BE RETRIEVED
RECORD COMPONENT SELECTION AND UPDATE NETWORK HISTORY
END

APPENDIX E

GEAR OPTIMISATION PACKAGE
OPTGEAR

E.1. OPTGEAR Package

The OPTGEAR package is a self-contained GA optimisation program and user interface. The package utilises the gear optimisation process described in section 5.5, to perform the search for the design configuration that will give the maximum performance according to the desired goals.

The results of the GA optimisation are displayed to the user, giving both the current performance and relative performance to the initial design. Upon achieving satisfactory levels of performance, the configuration of the gear and its performance can be saved to the file destination of the user's choice. The schematic in Figure E.1 illustrates how the GUI relates to the Gear Optimisation module of the IIS.

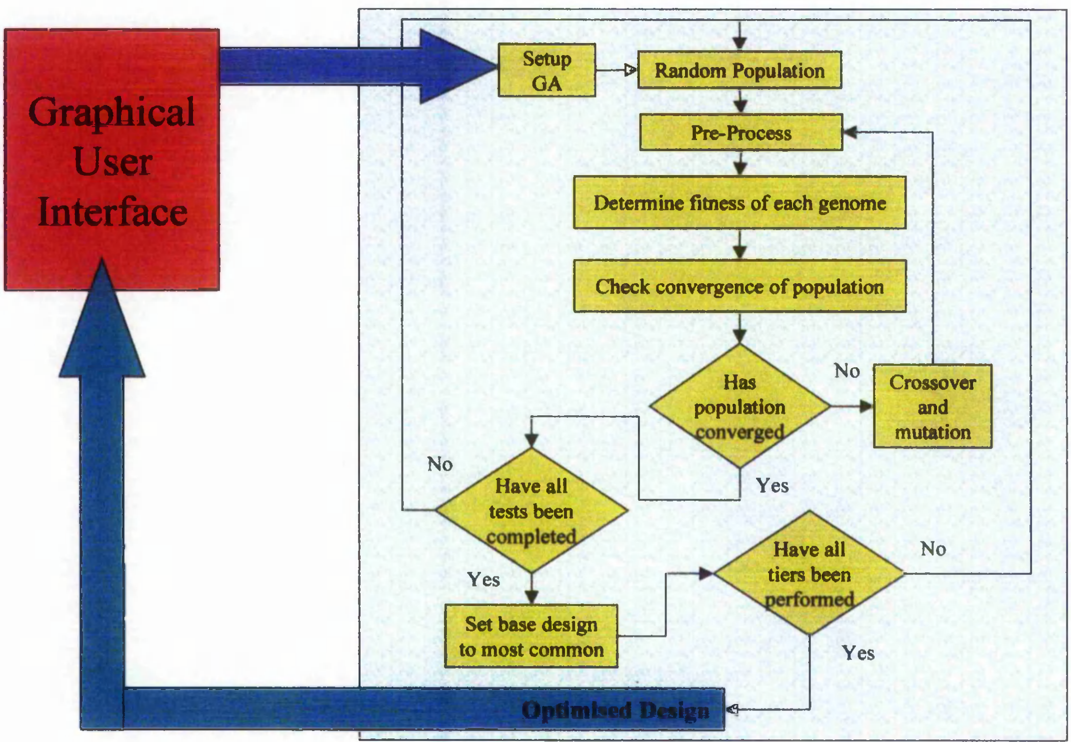


Figure E.1 Schematic of OPTGEAR Program

The following sections give instructions and an example to illustrate the package and its abilities.

E.2. OPTGEAR Instructions

The following Instructions are extracted from the OPTGEAR package.

OPTGEAR version 1.0

Spur and Helical Gear Optimisation Package

Created by M. Wakelam.

Copyright- 1998, The Nottingham Trent University, England.

Overview

This package applies a Genetic Algorithm (GA) to improve the performance of Spur and Helical gears. The GA conducts an adaptive search of various configurations of gear, derived from an initial rough design that must be supplied.

Requirements

Prior to the optimisation of the gear, two stages must be completed: define the target of the search (stage 1) and provide an initial design (stage 2).

Stage 1

The optimisation process can modify up to 9 parameters of the gear design including:

· Facewidth

· Module

· Pressure angle

· Helix angle

· Rack tip radius

· Addendum coefficient

· Addendum modification coefficients for both Pinion and Wheel

· Number of teeth on the Pinion

Selection of these parameters will include them in the optimisation process. Non selection will freeze the values to those defined in the initial design. The process is capable of optimising the performance of both variable and fixed centre distance gear pairs, however, for the fixed centre distance cases the Addendum Modification Coefficient for the wheel and the number of teeth on the pinion are calculated and not directly included in the search.
Thus their selection is not critical for fixed centre distance cases.

The aims of the optimisation are defined by the adjustment of the scales relating to the fitness functions. The fitness functions influence the final design that the package will produce. Bias toward one ore more of the goals will result in the final design emphasising this characteristic.
The goals include:
· Minimising facewidth
· Minimising centre distance (variable centre distance only)
· Reducing the difference in Bending Stresses between the Pinion and Wheels teeth
· Increasing contact ratio
. Reducing the difference in the tooth tip sliding ratio of the pinion and wheel
Initially the goals are set equal at their maximum. Bias is applied by reducing the significance of the goals of lower importance to the desired design.

The set-up of the GA may also be adjusted, however, it is suggested that the 'recommended' values are used. If a quick solution is required this can be achieved by reducing the population size (size of the search conducted) and the number of tests (number of times the optimisation process is repeated to ensure that the optimum is being achieved).

Stage 2
The initial design provides the starting point of the optimisation search. Here the basic configuration of the gear must be provided, including geometry, performance and material information. These are prompted for or default values are provided.

Once an initial design has been entered, it can be saved to any location for retrieval in the future.

Optimisation

Once the goals of the search and the initial design have been defined the optimisation process can proceed. This process can be time consuming depending upon the size of population defined and the number of tests.

Upon Completion

Upon completion the performance of the optimisation process can be displayed in graphical form. The paths of the goals are provided to allow the user to ensure that the solution has displayed repeatability.

The final design is displayed giving its major parameters and its performance, including the improvements on the initial design and the stresses acting upon the design. The final design is recorded in the text file RESULT.DAT but can be copied to any alternative location.

Failure of Program

If the program fails at the optimisation stage ensure that the initial design is realistic and does not include zero for dimensions, module, addendum coefficient and rack tip radius.

E.3. Example of Spur Gear Optimisation

The example below illustrates the pre and post-process stages of the gear optimisation that the GUI performs.

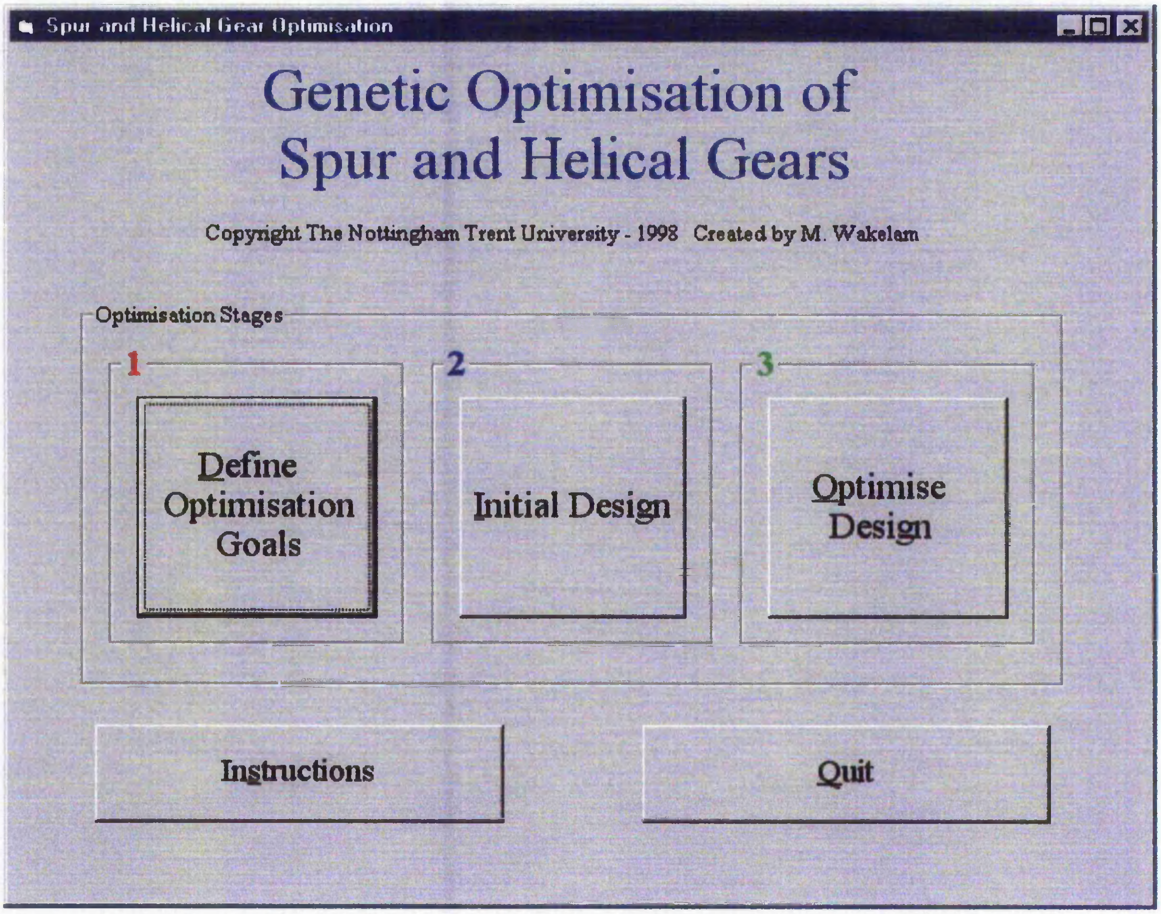


Figure E.2. OPTGEAR Front End

Figure E.2 illustrates the three steps of the optimisation process:

1. Goal Definition
2. Initial Design
3. Optimisation

Stages 1 and 2 must be completed prior to commencing the optimisation process. Failure to complete these steps will result in an error box telling the user which step has been missed.

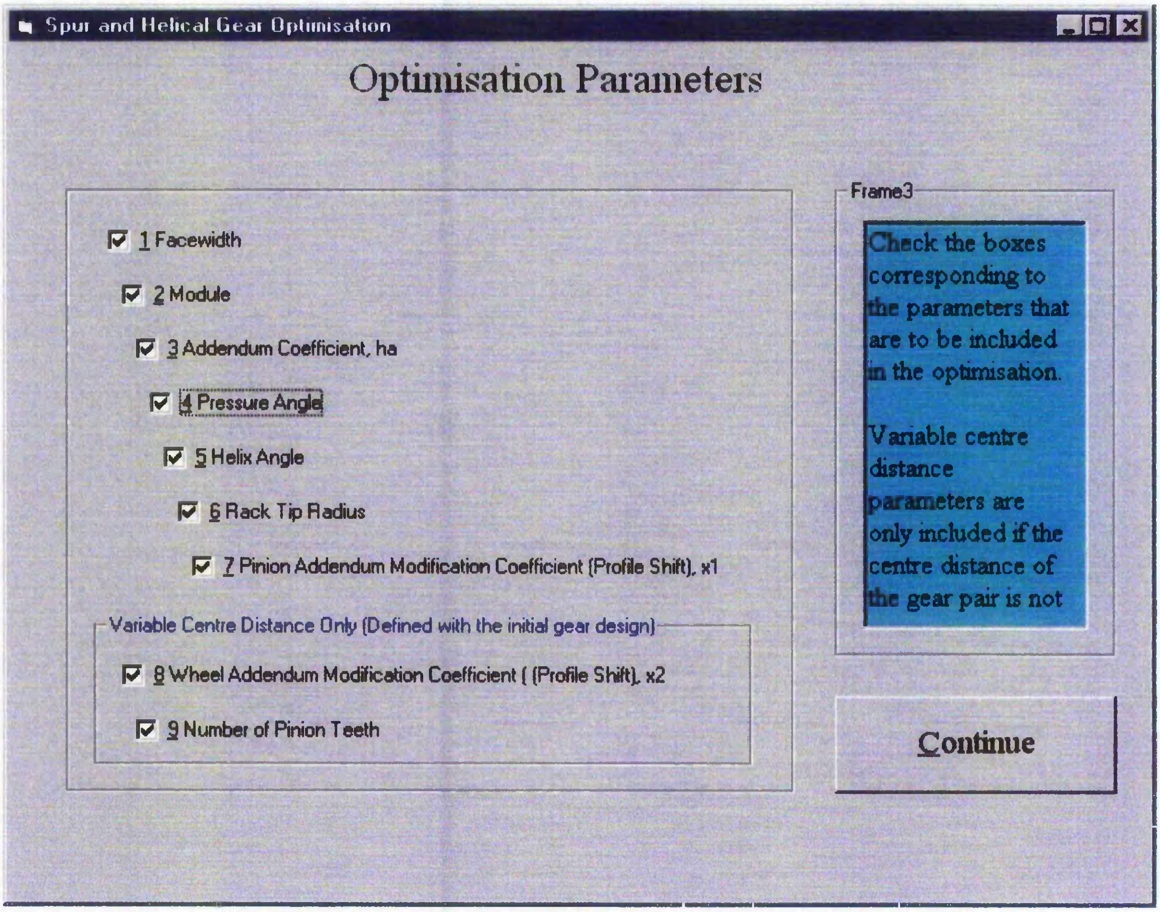


Figure E.3. Selection of Parameters to Include in the Search

The goal definition commences with the selection of the parameters to be adjusted. In Figure E.3, 9 parameters are shown which correspond to the genes that form the GAs genome. Selection of the parameters (indicated with a tick) includes that parameter in the search. Leaving the box blank removes the parameter from the search and maintains its initial design value constant.

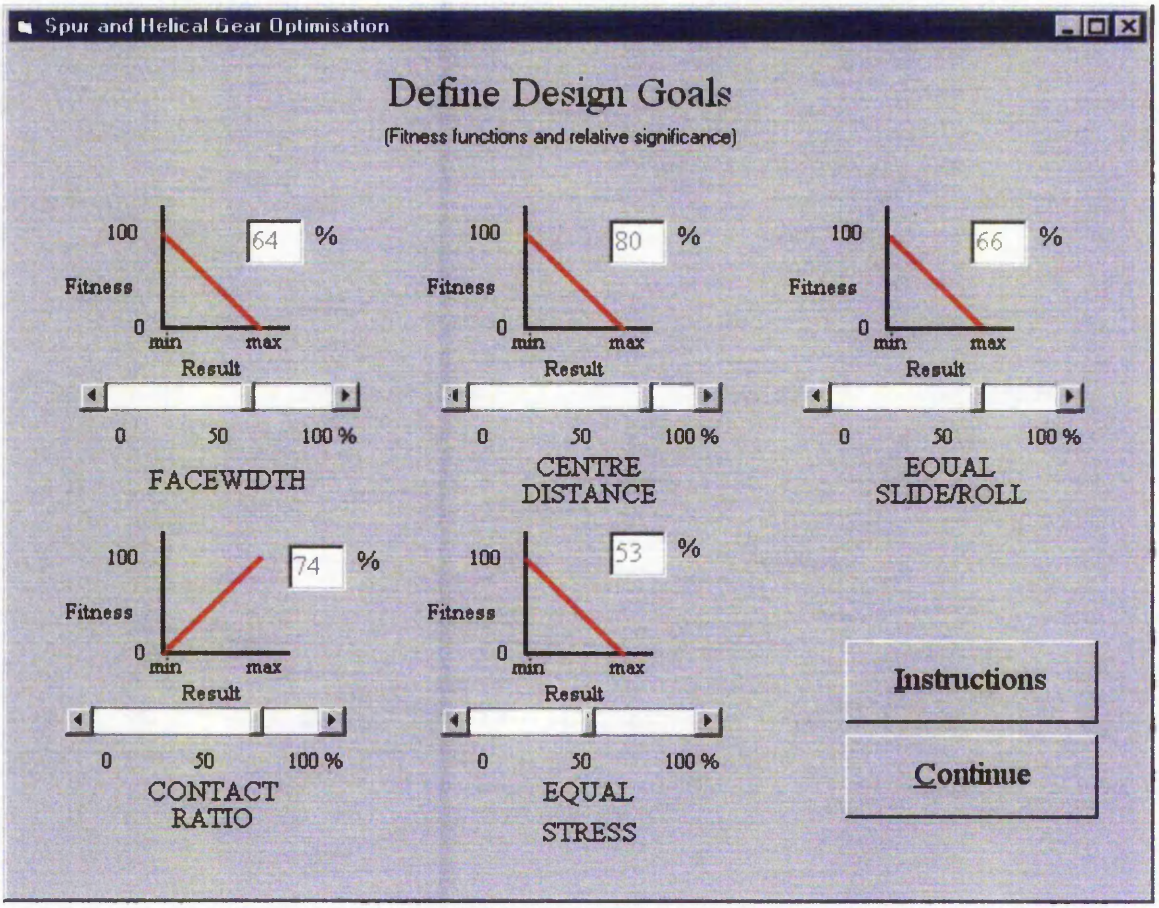


Figure E.4. Graphical Adjustment of the Optimisation Goals

The goals of the search are defined by adjusting the importance of the 5 fitness functions shown in Figure E.4. The importance of each goal is adjusted by moving the scroll bar either to the left to decrease or to the right to increase. The user alters the importance of the goals to model the desired design, thus allowing emphasis to be placed on individual goals.

The diagrams illustrate the fitness functions user by the GA and how the performance of the design, the goals and the fitness are related.

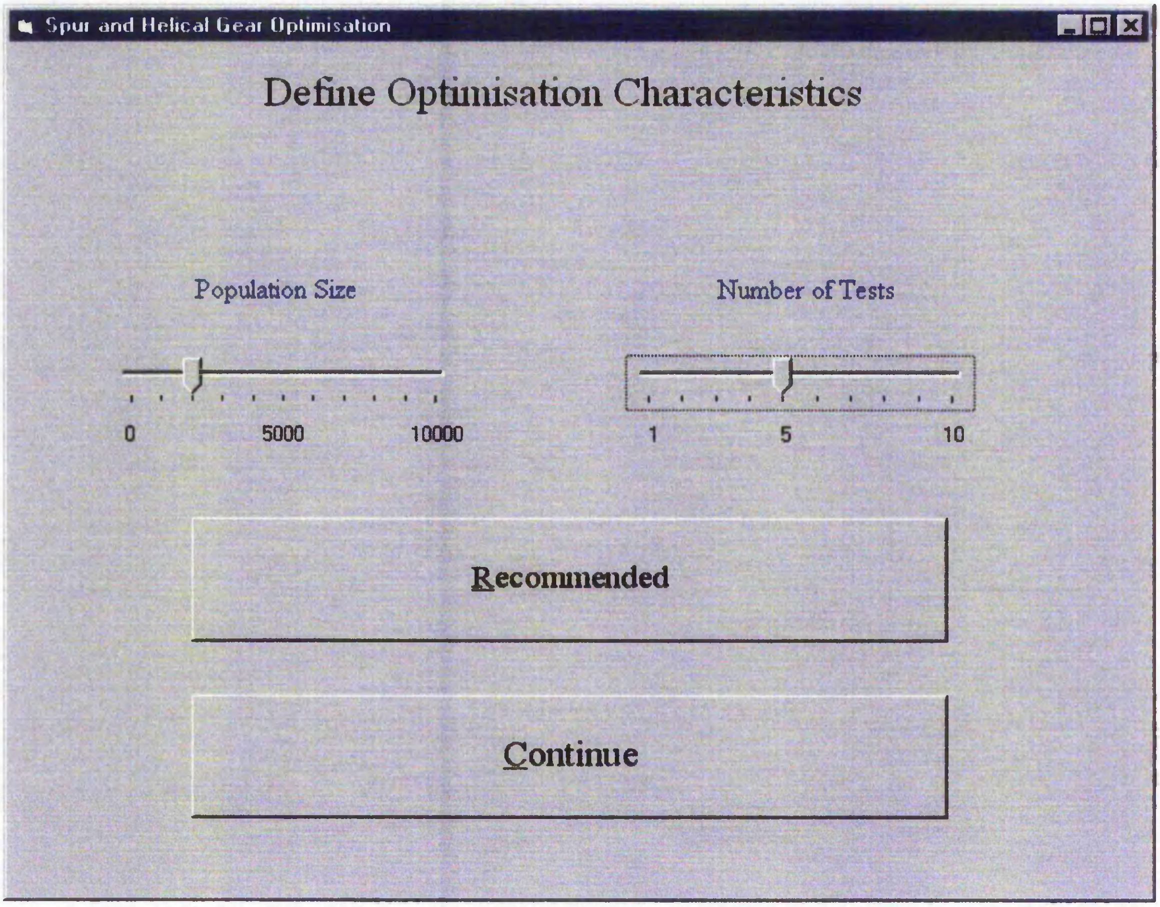


Figure E.5. GA Population and Repeatability Test

The characteristics of the GA are set by the user with the aid of the form illustrated in Figure E.5. The user can either set the population size and the number of times the optimisation process is repeated (to ensure repeatability) or can select the recommended settings, (population of 2000 genomes and 5 tests).

Increasing the population size increases the resolution of the search process with regard to the search area. This increase thus increases the ability of the GA to obtain the optimum solution. However, increasing the population increases the time taken for the process to converge upon a solution

The level of confidence in the solution increases as the number of tests increase. This is due to the increase in the number of solutions from which the most frequent solution is chosen. However, increasing the number of tests increases also increases the time taken to achieve the resultant design.

The package therefore provides 'recommended values' for initial use, which can be altered by the user if the solution is considered unsatisfactory.

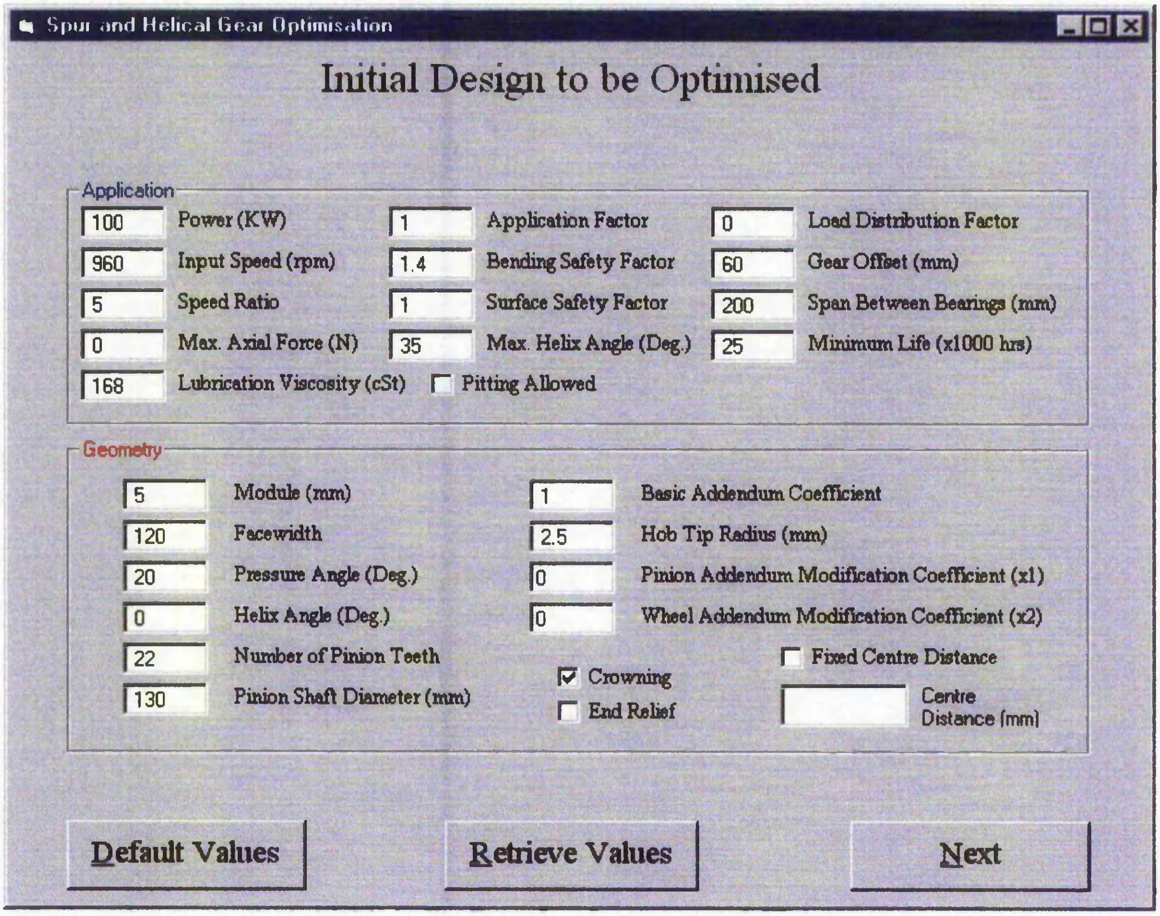


Figure E.6. Initial Geometrical Design

The geometrical information relating to the initial design is entered into the fields in Figure 6 above. This information represents the minimum required to perform the design. Failure to enter all information results in a warning message indicating the design is not complete. The default for the form is for variable centre distance. If fixed centre distance is required the fixed centre distance check box must be selected. Upon selection the centre distance field is enabled allowing its value to be entered. Once the centre distance is given the module and number of teeth are checked to ensure that the initial design is valid for this distance. If the module and number of teeth do not match the centre distance results in an error message until the calculated centre distance agrees with the stated. A small discrepancy is allowed and is compensated for by adjustment of the addendum modification coefficients.

The manufacturing and material properties are entered as in Figure E.7. The definition form shown in Figure E.8 explains the enumerate values that have been used.

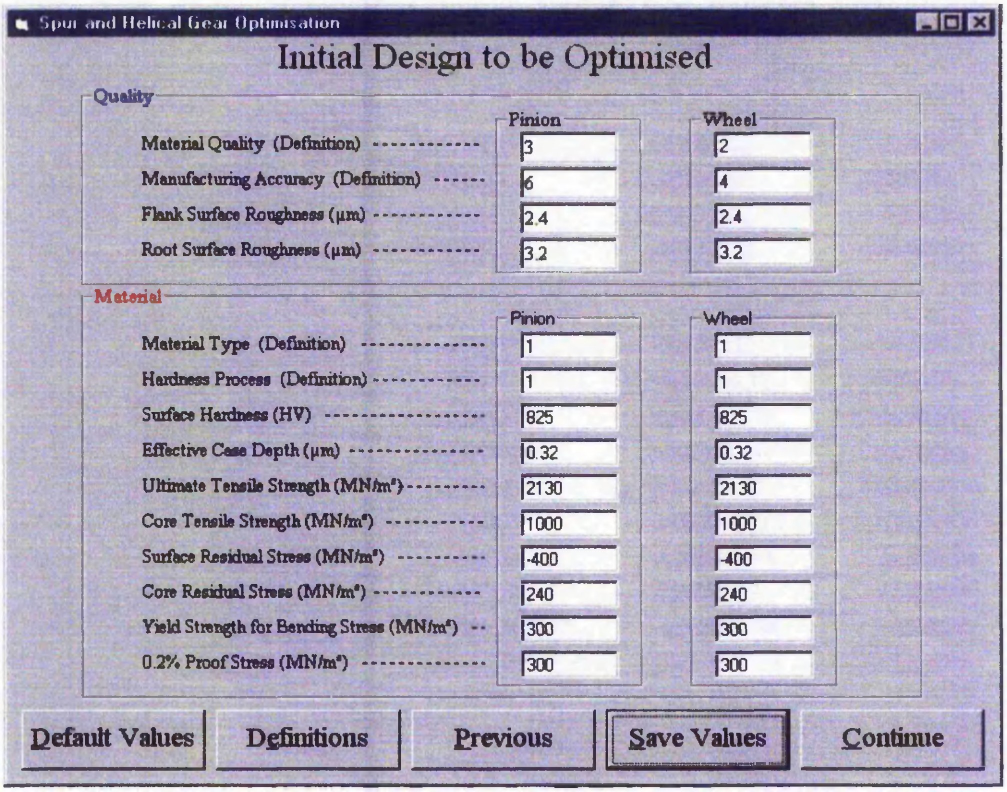


Figure E.7. Material and Manufacturing Properties

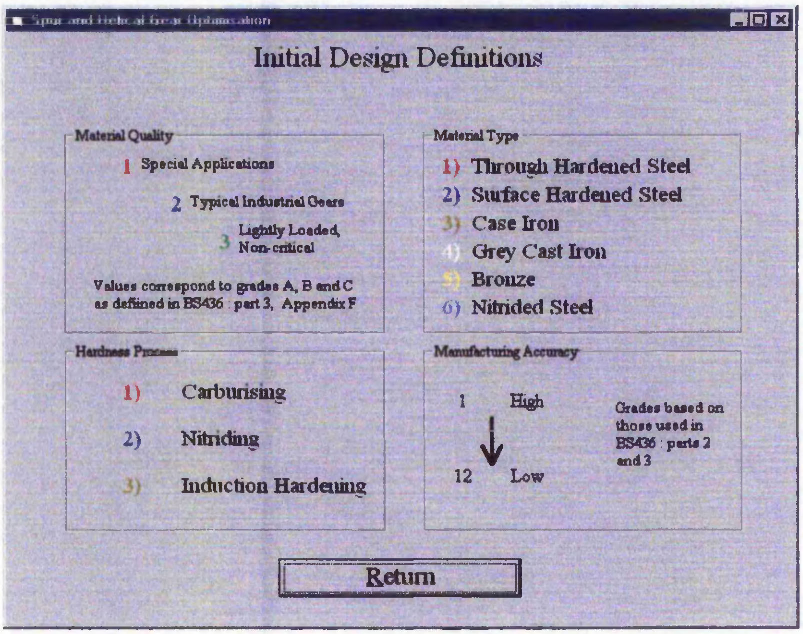


Figure E.8 Definitions of Material and Manufacturing Processes

Once all the fields have been filled the initial design is complete and ready for optimisation. At this point the design can be saved for reference or to repeat the process (for example using different goals)

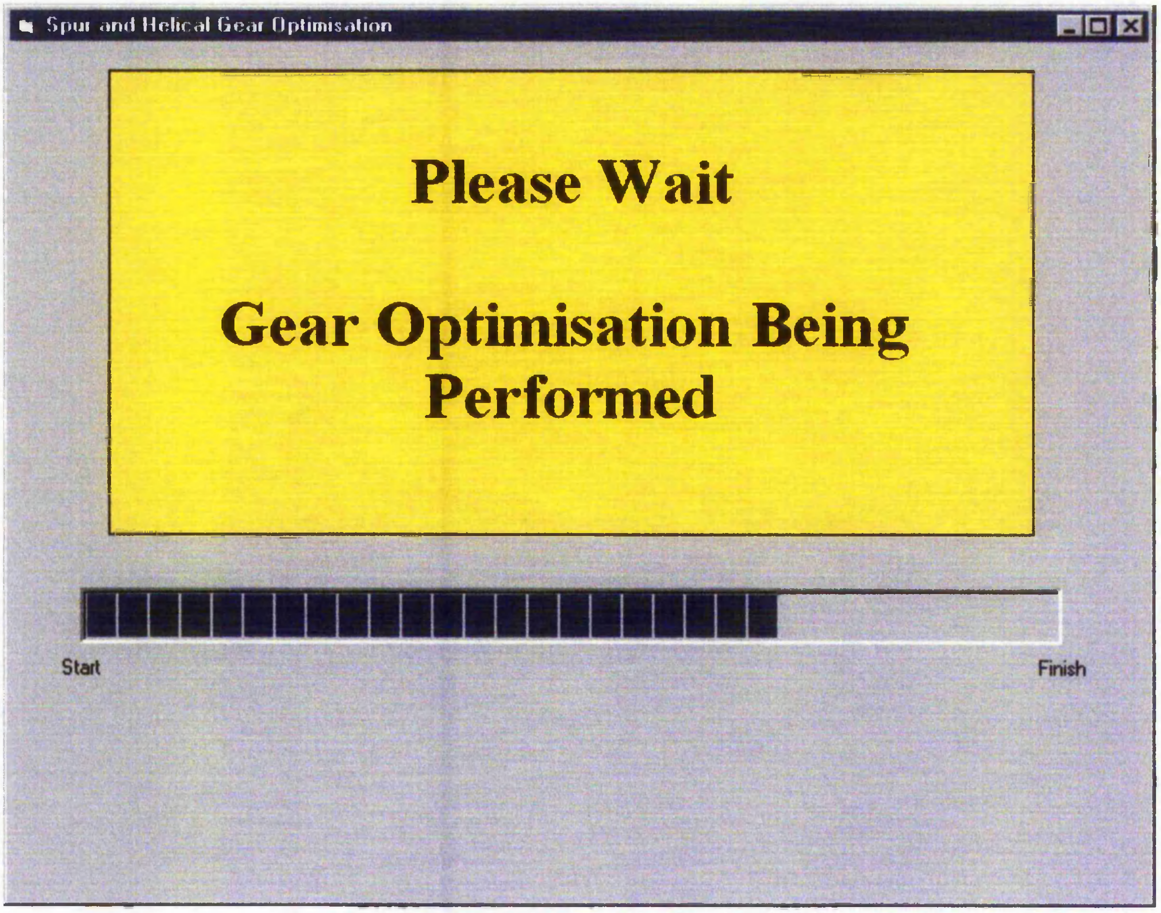


Figure E.9. Status of Optimisation Process

The time taken to perform the optimisation process varies and is dependant upon the population size and the number of times that the process is to be repeated. The status bar in Figure E.9 indicates the progress of the optimisation.

Once the GA process is complete the status bar is filled as in Figure E.10, and the results can be viewed

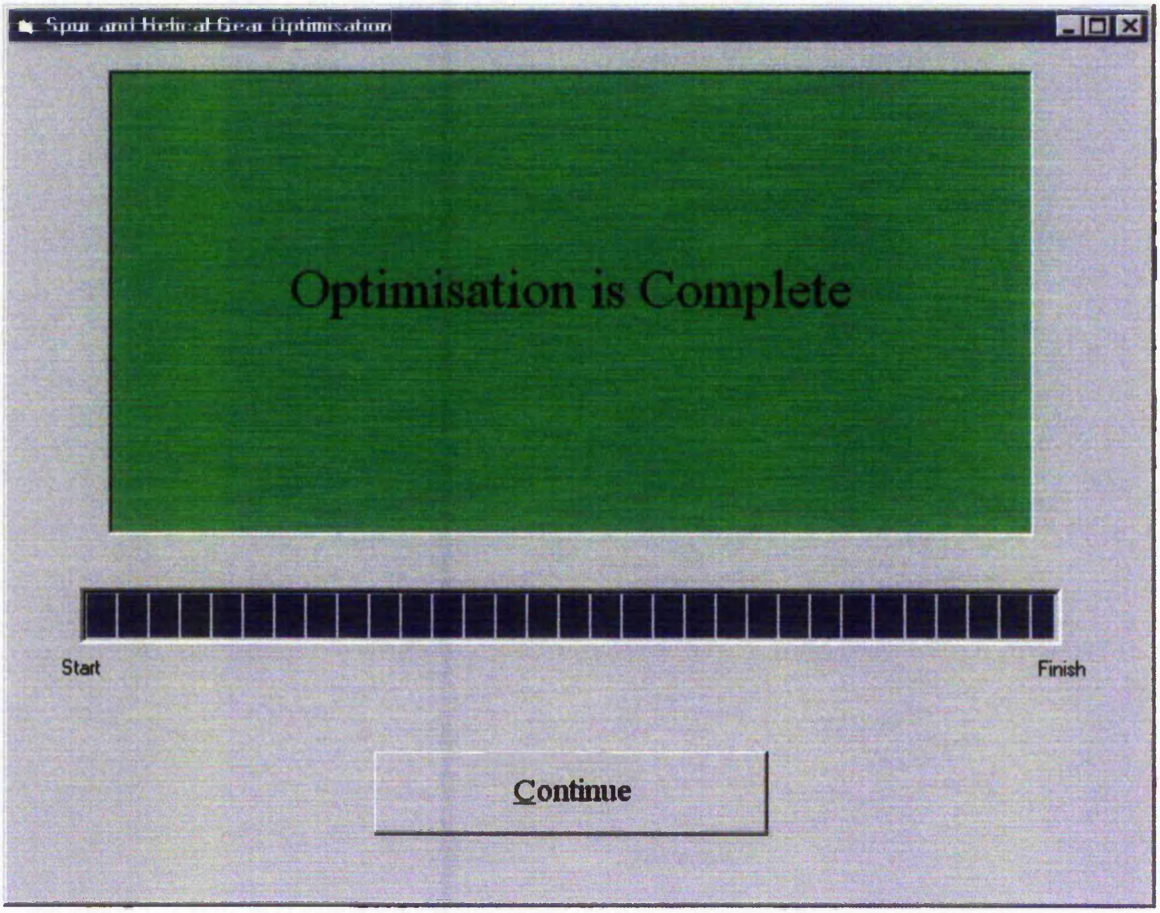


Figure E.10. Optimisation Process Complete

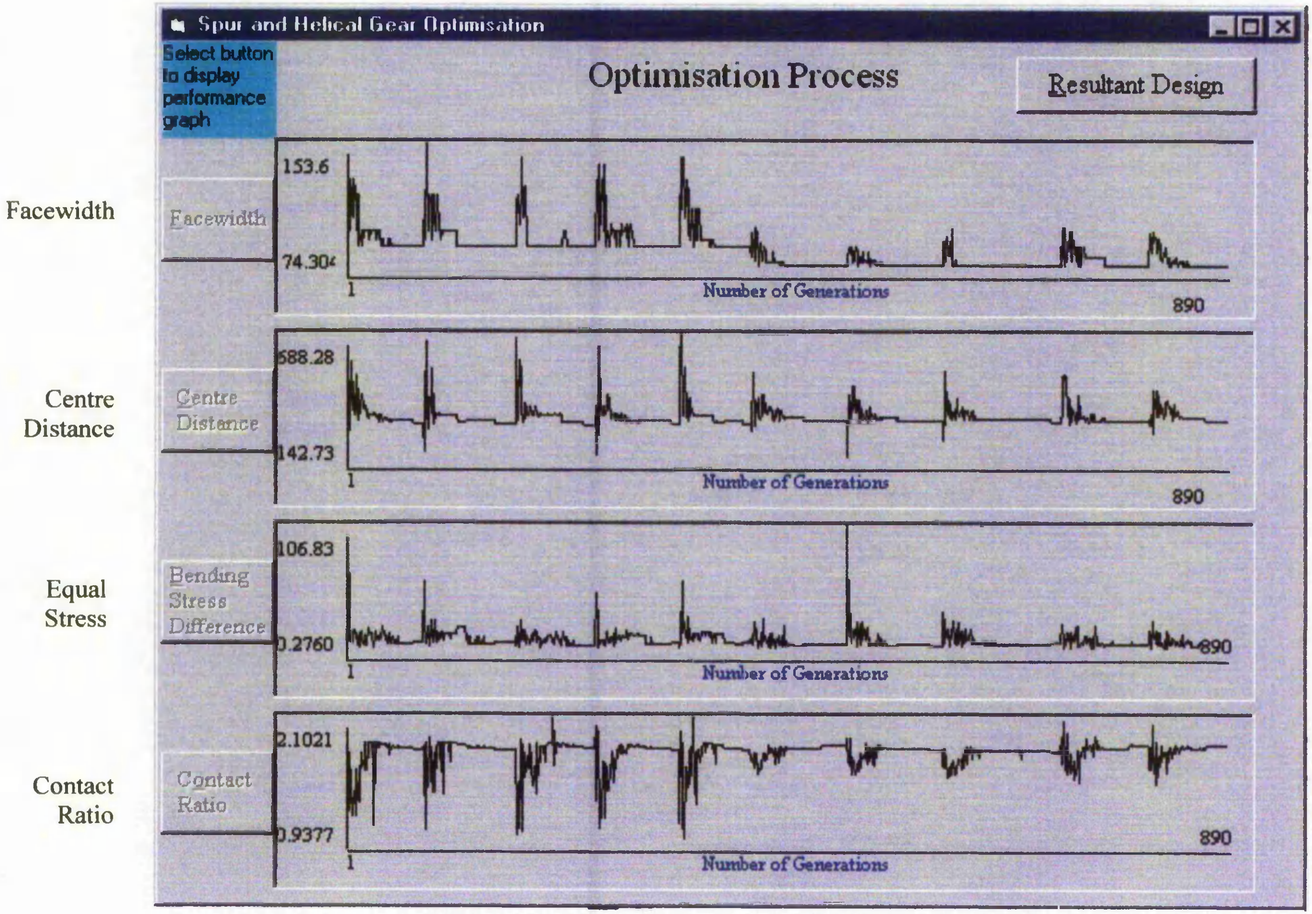


Figure E.11. Optimisation Process Performance

The path that the search has taken is illustrated in Figure E.11. The performance of the designs is given in the graphs, indicating the trend of the search and the levels of performance. The result traces also provide a means of evaluating the repeatability of the GA to achieve a solution to this design and thus indicates whether the fitness functions and population size are set correctly. Each spike represents the beginning of a test and as the trace levels out convergence is displayed.

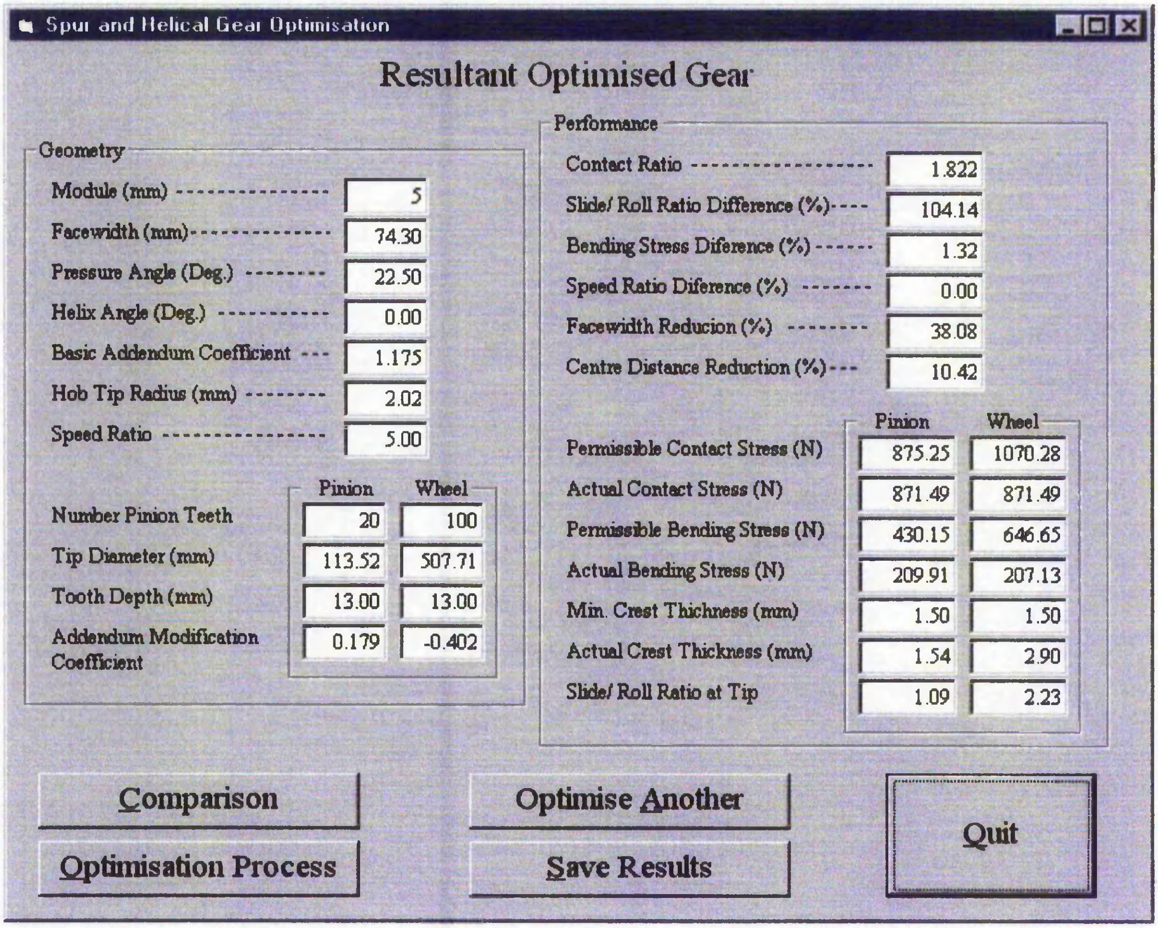
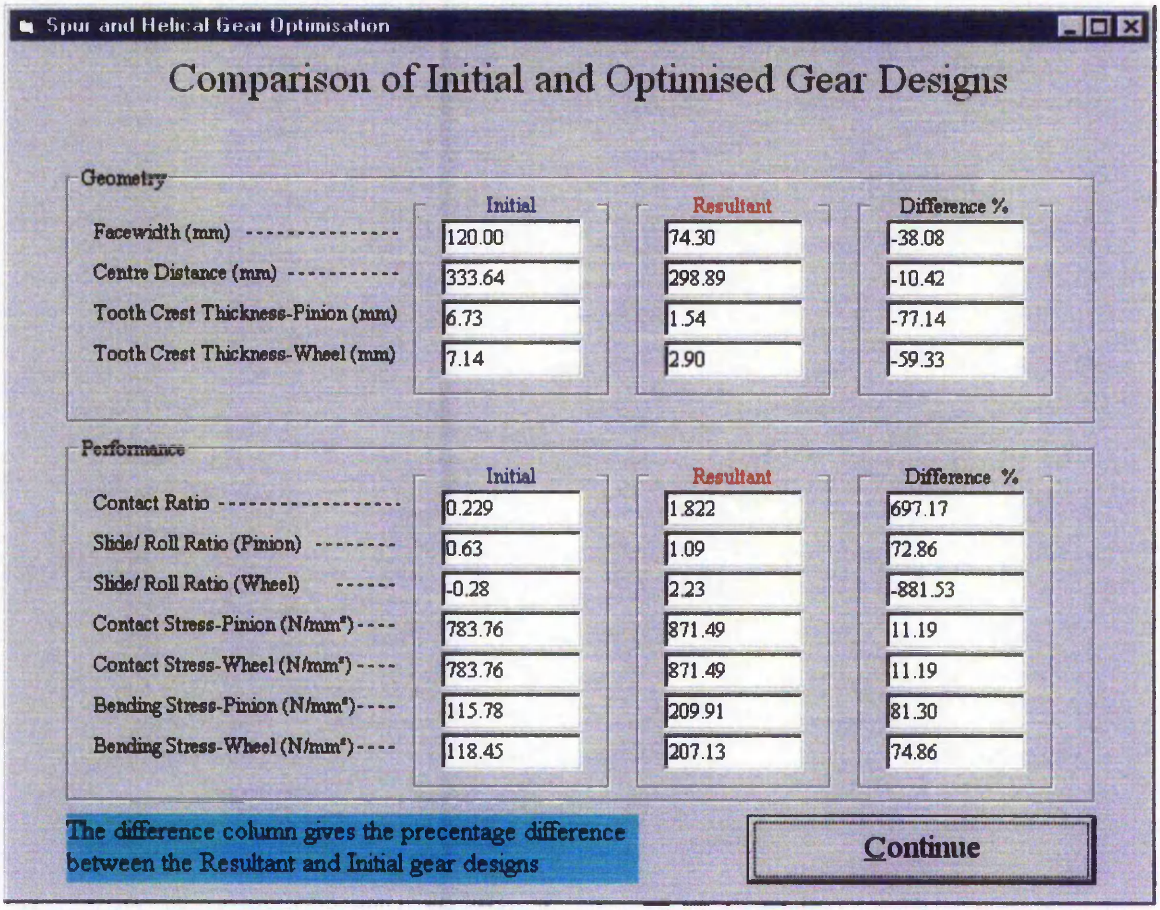


Figure E.12. Resultant Gear Design and Performance

The performance and critical dimensions of the final design are given in Figure E.12. As can be seen from the example the actual contact and bending stresses are below the permissible values indication that the design will not fail to BS436 part 3 (1986). Additionally the facewidth and the centre distance have been reduced and the bending stresses within the pinion and wheel teeth vary by 1.32%.

The addendum modification coefficient of the pinion is positive, ensuring that undercutting will not be required.

Full results of the design parameters can be saved in ASCII format to a destination of the users choice by choosing the 'Save As' button.



Spur and Helical Gear Optimisation

Comparison of Initial and Optimised Gear Designs

Geometry

	Initial	Resultant	Difference %
Facewidth (mm)	120.00	74.30	-38.08
Centre Distance (mm)	333.64	298.89	-10.42
Tooth Crest Thickness-Pinion (mm)	6.73	1.54	-77.14
Tooth Crest Thickness-Wheel (mm)	7.14	2.90	-59.33

Performance

	Initial	Resultant	Difference %
Contact Ratio	0.229	1.822	697.17
Slide/ Roll Ratio (Pinion)	0.63	1.09	72.86
Slide/ Roll Ratio (Wheel)	-0.28	2.23	-881.53
Contact Stress-Pinion (N/mm²)	783.76	871.49	11.19
Contact Stress-Wheel (N/mm²)	783.76	871.49	11.19
Bending Stress-Pinion (N/mm²)	115.78	209.91	81.30
Bending Stress-Wheel (N/mm²)	118.45	207.13	74.86

The difference column gives the precentage difference between the Resultant and Initial gear designs

Continue

Figure E.13. Comparison of Initial and Resultant Gear Designs

A supplementary form to display the relative performance of the resultant design to the initial design is illustrated in Figure E.13. The improvement in design helps indicate if the GA optimisation has achieved its goals and to what extent.

A RULE BASED AND ARTIFICIAL NEURAL NETWORK SYSTEM FOR CONCEPTUAL DESIGN

D Su, K Henthorn, K Jambunathan and M Wakelam

Department of Mechanical Engineering, The Nottingham Trent University. England.

ABSTRACT

A hybrid system of rule bases and Artificial Neural Networks (ANNs) has been developed for conceptual design. The design of the power transmission systems is used as a test case where the ANNs and rule bases work in conjunction to conduct the conceptual design. In this paper, after a brief introduction to the design of power transmission systems, the hybrid system is described with respect to the overall structure of the system, the rule bases, the ANNs architecture and training data preparation. The system exploits the ANNs advantages over Expert Systems of flexibility to alter knowledge and the ability to make ambiguous decisions. The An example is given which illustrates that the system is able to use the best qualities of the two Artificial Intelligence techniques.

Keywords: Artificial Intelligence, Artificial Neural Networks, Rule Based Systems, Conceptual Design, Power Transmission Systems.

1. INTRODUCTION

Generating design concepts is the most demanding stage of the design process. The designer must take into account the physical constrains, manufacturing processes, commercial aspects and practical knowledge when creating concepts.

At present the majority of conceptual designs are created manually requiring a designer with expertise in a particular field. The designer's expertise and knowledge has been built up over a number of years and is something only the designer possesses, and it is not always transferable to another person. This is where a problem arises. If the designer were to leave a company, for example, all the expertise and knowledge acquired is lost.

The solution is to develop a means of retaining expertise and knowledge indefinitely and make it generally available.

To meet with such a demand an Expert System approach was developed [1-4]. Although the approach has been successfully applied to design, it has been found that it is not flexible. Alteration of the knowledge bases after the system development is complete is almost impossible. ANNs on the other hand are different and can be adapted to be more flexible due to the inherent way in which knowledge is captured. The knowledge stored in an ANN system can be changed by training the systems ANNs using different training data thus allowing the knowledge to be changed without altering the system itself.

This paper applies Artificial Intelligence techniques to the conceptual design of a power transmission system. In particular rule bases and ANNs are used together to overcome problems within the conceptual design. The rule bases are used to handle the clearly defined knowledge, while the ANNs are used to deal with ill-defined knowledge.

ANNs form a major part of the design system. The advantages that they have over rules and design calculations are that they do not require the knowledge they contain to be explicitly described. The knowledge that the ANNs contain for this system appears to be of a random nature. However the ANN is able to learn the training data.

The ANNs selected for the system are multi layer perceptrons, incorporating the back propagation technique as described by Lippmann [5]. The back propagation technique has been implemented due to its previous successful use in simulating complex functions by Shtub and Zimerman [6] and Lou and Brunn [7]. Also as input and output patterns for training sets are available, a supervised network becomes an attractive option.

2. CONCEPTUAL DESIGN OF POWER TRANSMISSION SYSTEMS

Power transmission systems are found in many applications ranging from hydraulic, electrical to mechanical systems. This research is focused on mechanical systems.

The basic arrangement of a transmission system is determined by the orientation of its input and output shafts. Figure 1 gives schematic examples of the systems orientation.

The design of the power transmission system also includes the selection of components such as gears, belt drives, etc. The selection of the components are determined by two factors:

i. The suitability of a component to fulfil the design specification, (as described in section 5).
ii. The component hierarchy. The component hierarchy is a sequence that governs a rule. The rule states that once a component has been used in a design, any additional stages may not use components higher in the sequence. Figure 2 demonstrates the hierarchy.

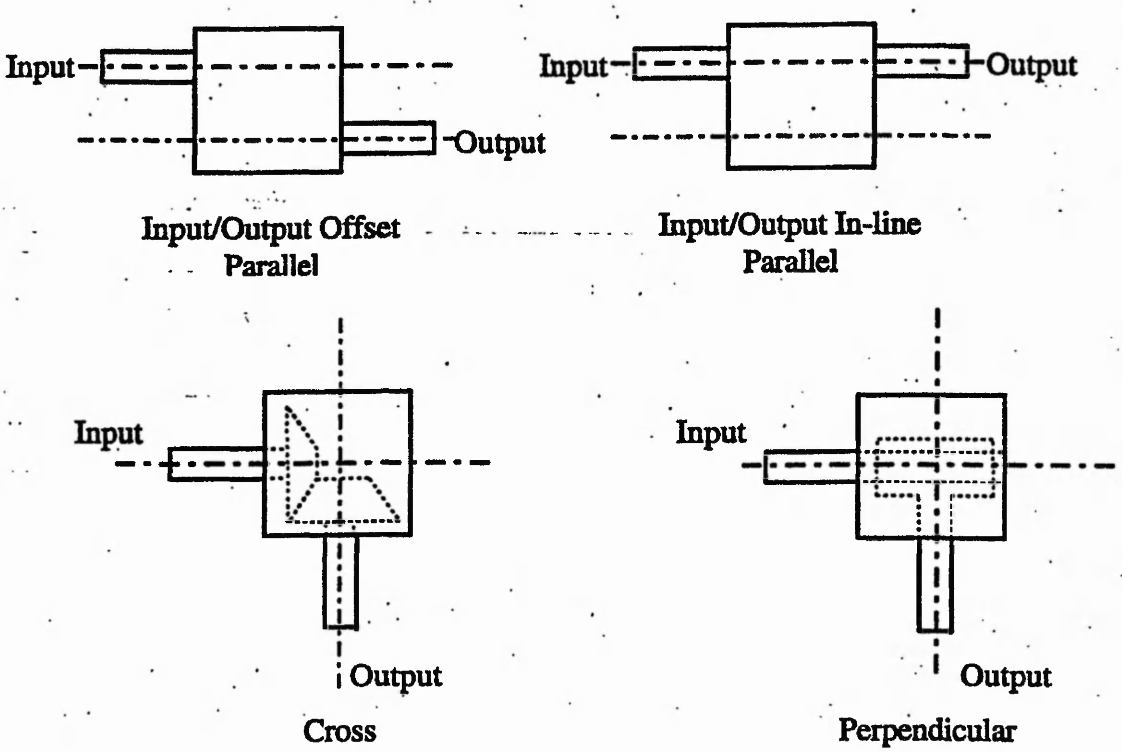


Figure 1. Schematic of System Orientation

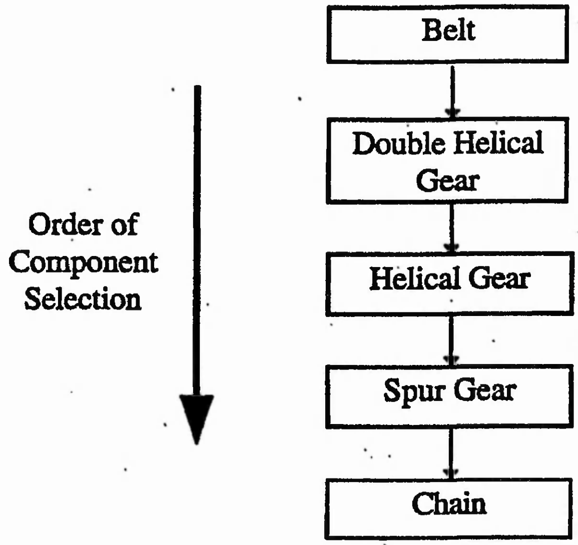


Figure 2. Component Hierarchy

3. OVERVIEW OF THE HYBRID SYSTEM

A hybrid system of ANNs and rule bases was developed using the C++ programming language. The rule bases cover the situations with clear decisions one way or the other, while the ANNs are used in ambiguous situations.

The use of the rule bases and ANNs are controlled by a control program which executes the design process, calling upon both the rule base and ANNs as and when required. The control program also acts as a pre and post processor, preparing the inputs from the user interface then analysing and displaying the results. Figure 3 below illustrates how the control program co-ordinates the design.

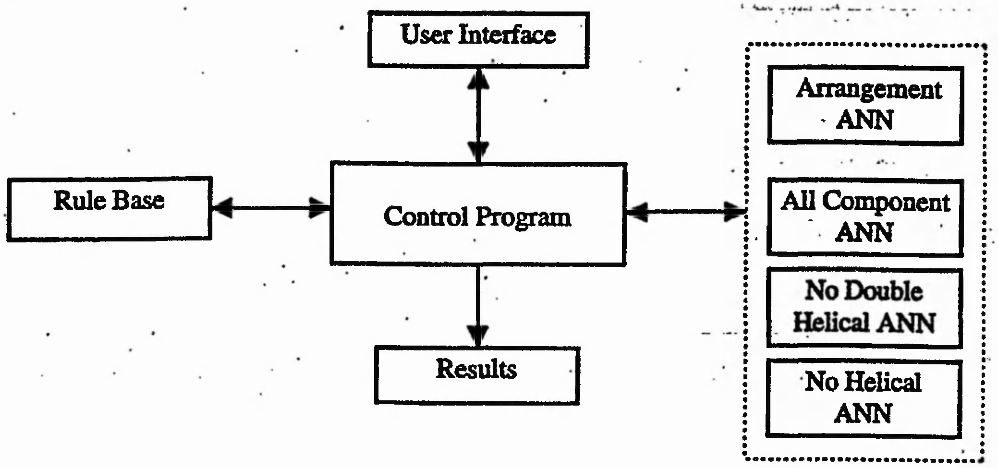


Figure 3. Design Program Schematic

The design program addresses the problem in a similar manner to a designer. Just as a designer will split the problem up and address each element in turn, the program splits the design into two areas, arrangement and component selection. Each of these areas has a number of rules and ANNs that work together to produce the result.

The first stage of the design process is to determine the transmission system arrangement. This is performed by the arrangement network. Before the network can produce a suitable design its input pattern must be generated. A combination of rules and user input are used to generate the input pattern. The rule base generates the first two parts of the input pattern that correspond to the orientation and the number of stages, while the remainder consists of the user defined design specifications. Using the input pattern, the ANNs produce a suitability rating for all the possible transmission system arrangements. The control program then analyses the results of the ANNs, ranks the designs obtained in order of suitability and then presents the best solution together with two alternatives to the user. The program also allows the user to change the decision if it is not considered satisfactory.

Once the design arrangement has been finalised the components will be determined using both the rule base and a combination of ANNs. The rule base selects the non-parallel components and determines the correct ANN to use. This is based upon the component hierarchy which is represented by three component ANNs. These three component ANNs are used as another method of splitting up the design process. The ANNs contain either all the parallel components, the double helical gear set to zero or both the double and single helical gears set to zero. Setting the components permanently to zero during training effectively removes them from the design while keeping a uniform architecture to the ANNs necessary for analysis of the results by the control program.

When all the elements of the design have been obtained the control program displays the finished conceptual design for the user and stores the results as a permanent record on file.

4. RULE BASES

The rule bases consist of three parts: determination of the number of stages within the power transmission system, selection of relevant ANNs to conform with the component hierarchy, and selection of the first stage components. The rule bases communicate with the control program throughout the whole design process. A typical example of a set of rules is shown below which are used to determine the number of stages required for parallel orientation. The selection is dependant on the speed reduction *ratio*.

If *ratio* <4 then arrangement is *one stage parallel*
If *ratio* $\geq$ 4 & $\leq$ 7 then arrangement is either *one stage parallel* or *two stage parallel*
If *ratio* $\geq$ 7 & $\leq$ 30 then arrangement is *two stage parallel*
If *ratio* $\geq$ 30 & $\leq$ 100 then arrangement is *three stage parallel*

5. ANNs AND TRAINING DATA PREPARATION

The training of the ANNs were performed using a training module developed for the design system by Wakelam [8]. This module allows the architecture of the networks to be created or adjusted and also the training period and learning coefficient to be defined. The development of the ANN architectures were by experimenting with various configurations, adjusting the number of processing elements and layers and analysing the results according to the training diagram shown Figure 4.

Testing, which is performed by another module [8], ensures the ANN is following the training set and producing the correct output to a series of input patterns. If testing results in unacceptable discrepancies between the desired and actual output from the ANN retraining is necessary. Correct training of the ANNs is essential as they contain the knowledge and expertise of the design system.

The training data sets for the ANNs were prepared based on the evaluation matrix approach for conceptual design developed by Su [4]. The matrices (see Figures 5 & 6) contain a series of values or weights at the intersection where the design and a design specification meet. The weights represent the suitability of a design to perform the design specification, ranging from complete unsuitability, 0, to ideally suited, maximum value within the range. The suitability of a design to a set of specifications is found by summing all the values in the matrix which relate to both the design and the identified specifications. This summing process is repeated for all the designs and the design with the largest total representing the most suitable to the specification.

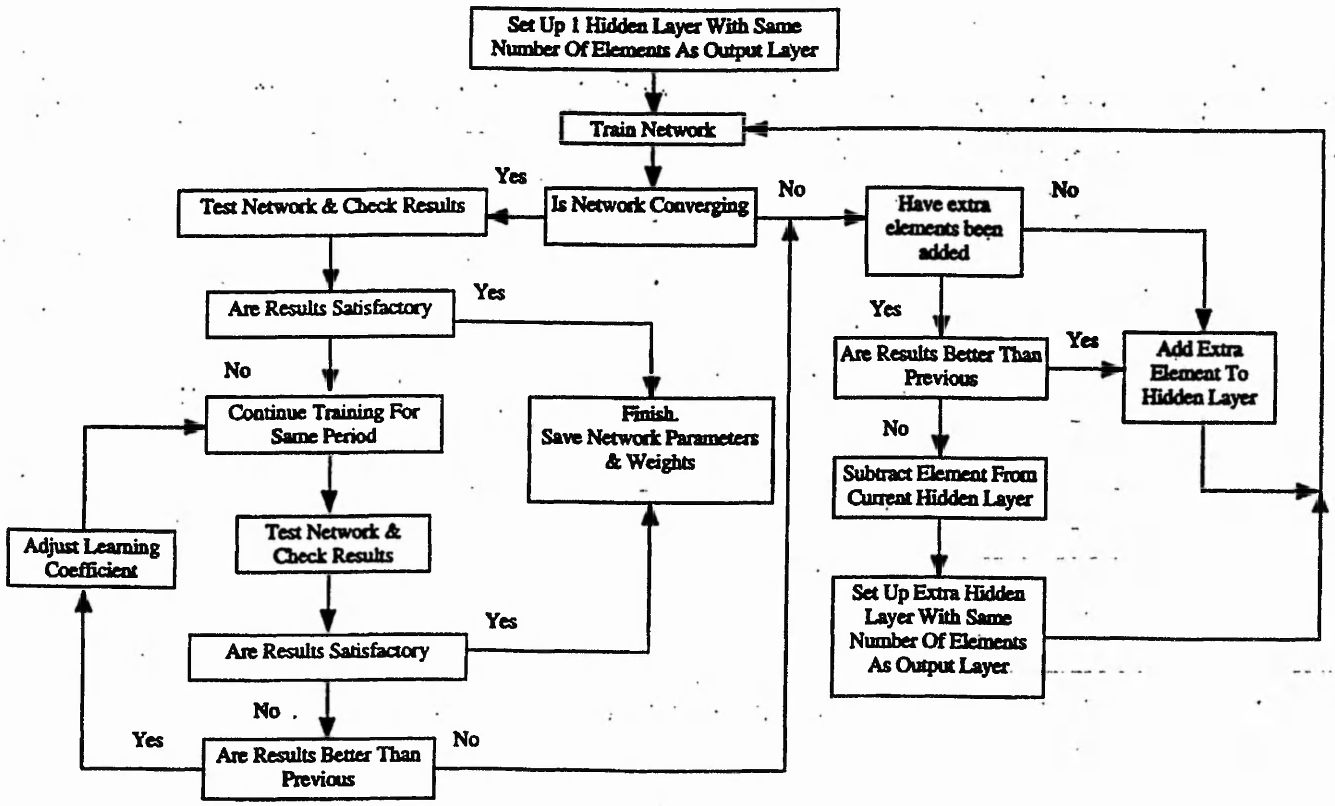


Figure 4. Training Diagram

Specifications

Arrangement Patterns	A	B	C	D	E	F	G	H	I	J
1 Stage Parallel	0	0	0	3.5	3.5	4.5	4	3	4	4
1 Stage Cross	0	0	0	3.5	3.5	3.5	3	3	3.5	3
1 Stage Perpendicular	0	0	0	3	2.5	2.5	3	3	3	3
2 Stage Parallel Offset	0	0	0	3	3	3	3	3	3	3
2 Stage Parallel In-line	0	0	0	2.5	3	2.5	3	3	2.5	3
2 Stage Cross	0	0	0	3	2.5	2.5	3	3	2.5	2.5
2 Stage Perpendicular	0	0	0	3	2	2	3	3	2	3
2 Stage, No Casing	0	0	0	5	5	5	5	5	5	5
3 Stage Parallel	0	0	0	3	3	2.5	2.5	3	2.5	2
3 Stage Cross	0	0	0	2.5	1	1	2	3	1.5	2

Figure 5. Arrangement Matrix (Range 0 to 6)

Specifications

Basic Units	A	B	C	D	E	F	G	H	I	J
Double Helical Gear	3	4.5	3	2.5	3	2	2.5	3	2.5	2.5
Helical Gear	3	4	3	3	3	2.5	3	2	3	3
Spur Gear	3	3	3	3	3	3	3	3	3	3
Belt	1	6	2	4	4	3.5	2	1.5	3	1
Chain	2	2	2	4	4	3	2.5	1.5	3	1
Worm Gear	3	5	1	2	2	1.5	6	2	2.5	8
Bevel Gear	3	2.5	3	3	3	2	2.5	2	2	2.5

Key :

A Ratio Accuracy
B Transmission Smoothness
C Ease of Maintenance & Assembly
D Ease of Manufacturing
E Cost
F Weight
G Load Distribution
H Manufacturing Accuracy
J Size

Figure 6. Basic Component Matrix (Range 0 to 8)

The design specification represents the ANNs input while the concepts represent the output as illustrated below in Figure 7.

The training data is created by the design specifications forming the input pattern while the normalised sum of the selected weights form the output pattern. The example below demonstrates the method.

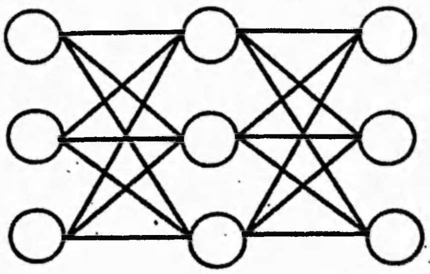


Figure 7.

Example. The transmission system design requires a component that will satisfy the specifications of smooth transmission, ease of manufacture and lightweight.

The specifications correspond to design specifications B, E & F that are highlighted in Figure 6. Table A below demonstrates how the training data is formed from the design matrix.

Table A. Training Data Table

Design	Specification										Total	Normalised Values
	A	B	C	D	E	F	G	H	I	J		
Double Helical Gear	0.0	4.5	0.0	0.0	3.0	2.0	0.0	0.0	0.0	0.0	9.5	0.7037
Helical Gear	0.0	4.0	0.0	0.0	3.0	2.5	0.0	0.0	0.0	0.0	9.5	0.7037
Spur Gear	0.0	3.0	0.0	0.0	3.0	3.0	0.0	0.0	0.0	0.0	9.0	0.6667
Belt	0.0	6.0	0.0	0.0	4.0	3.5	0.0	0.0	0.0	0.0	13.5	1.0000
Chain	0.0	2.0	0.0	0.0	4.0	3.0	0.0	0.0	0.0	0.0	9.0	0.6667
Worm Gear	0.0	5.0	0.0	0.0	2.0	1.5	0.0	0.0	0.0	0.0	8.5	0.6296
Bevel Gear	0.0	2.5	0.0	0.0	3.0	2.0	0.0	0.0	0.0	0.0	7.5	0.5556

From table A the following training data is obtained.

Input Pattern (in binary form, either selected or not selected)
0 1 0 0 1 1 0 0 0 0

Output Pattern
0.7037 0.7037 0.6667 1.0000 0.6667 0.6296 0.5556

6. EXAMPLE

Design a power transmission system to meet the following specifications:

- the input / output shafts are parallel to each other

- speed reduction 5 : 1
- ratio accuracy
- transmission efficiency
- able to distribute load

The input to the system and the arrangement and alternative designs from the system are shown in Figure 8. The final conceptual transmission system design, is shown in Figure 9.

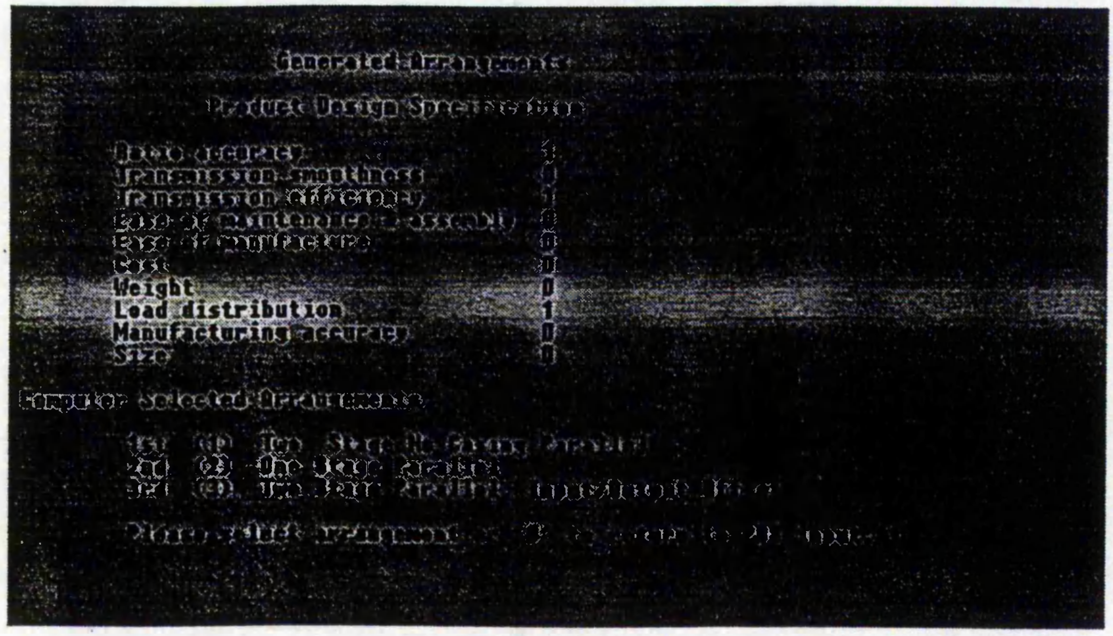


Figure 8 Screen Dump Illustrating Design Specification and Corresponding Arrangement Designs

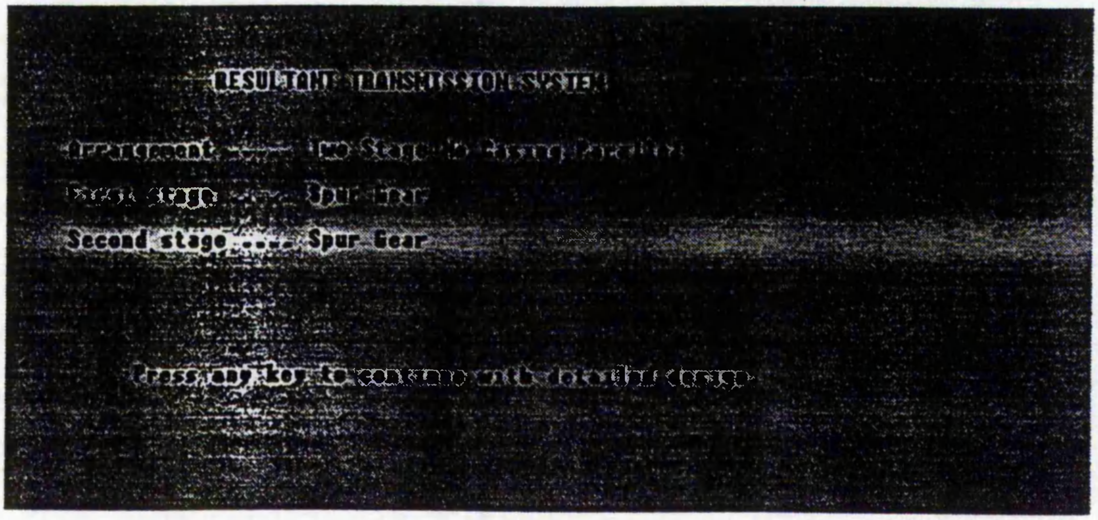


Figure 9. Screen Dump Illustrating The Resultant Finished Conceptual Design From The System

7. DISCUSSION AND CONCLUSIONS

From the example, in previous sections, the system has shown that it is capable of storing and applying design expertise and knowledge held within it to an application. The combination of the two Artificial Intelligence techniques have proven successful in overcoming some of the more ambiguous decision making situations such as when two or more designs are equally suited. This is the case in the example. From Figure 6 the design weights for both the Double Helical and Spur gear are the same for the design specification requested. As the designs are equally matched conventional methods of producing a design would require an additional decision to select the final design. The use of ANNs has removed this extra step as the decision between two or more equally suited designs will be made based on other designs encountered during training, appearing to giving the system a true sense of expertise.

During development of the system modifications to the knowledge contained within the ANNs were made. The modifications were successfully performed by retraining the relevant ANNs whilst not altering the system design [8], thus proving that the use of ANNs as a means of storing knowledge can solve the flexibility problem encountered by Expert Systems when modifying or updating a system.

References

[1]D. SU, 1993, *Application of a Knowledge Based, Integrated-System Approach to the Conceptual Design of a Power Transmission System*, Quality Through Engineering Design, W. KUO (Ed.), Elsevier Science Publications B. V., pp. 417-427.

[2]D. SU & R. J. Forgie, 1989, *Development of an Expert System Approach to an Engineering Design Procedure,* Proceedings of the Institution of Mechanical Engineers: Engineering Design (ICED89), Vol.2, Mechanical Engineering Publications Ltd., pp 975-989.

[3]D. SU, 1994, *Gearbox KBIS: A Prototype Knowledge-Based Integrated System for Gearbox Design*, Proceedings of the 1994 International Gear Conference, 7-9, September, Newcastle upon Tyne, UK, pp 201-205.

[4]D. SU & K. JAMBUNATHAN, 1994, *A Prototype Knowledge-Based Integrated System for Power Transmission Design*, Advancement of Intelligent Production, Eiji Usui (Ed.), Elsevier Science B.V., pp 45-50.

[5] R. P. LIPPMANN, 1987, *An Introduction to Computing with Neural Nets* IEEE ASSP Magazine pp. 4-22.

[6]A. SHTUB A & Y. ZIMERMAN, 1993, *A Neural-Network-Based Approach For Estimating The Cost Of Assembly Systems*, International Journal of Production Economics, Elsevier Science Publications B. V., 32, pp. 189-207.

[7]Y. F. LOU & P. BRUNN, 1995, *Using Artificial Neural Networks For Solving Engineering Problems*, Proceedings of the Thirty-First International MATADOR Conference, The Macmillan Press LTD, pp. 635-640.

[8]M. WAKELAM, 1995, *Development of a Rule Based and Artificial Neural Network System for Conceptual Design of Rotary Power Transmission Systems*, Internal Report No.MW01, Department of Mechanical Engineering, The Nottingham Trent University, England.

KNOWLEDGE REPRESENTATION INCORPORATING ARTIFICIAL NEURAL NETWORKS FOR DESIGN OF A MECHANICAL TRANSMISSION SYSTEM

M.Wakelam and D. Su

Abstract
An intelligent system has been developed and applied to the design of mechanical transmissions. Within the system, knowledge and expertise about the application has been encapsulated in several forms, including artificial neural networks. The applications to which the artificial neural networks are applied and the knowledge contained are described together with an automated training approach incorporating a genetic algorithm.

Introduction
The process of design, whether for an engineering purpose or any other, requires many stages relying heavily on information, knowledge and expertise in various fields. Manipulation of these factors enables the designer to develop a final product which will be capable of achieving the initial application. As the final product is entirely dependant on the design stages encountered during development, a design based upon the correct knowledge is essential. Often the design of a product may be similar to previous and future products, for example in a range of hand drills. Therefore accumulation of the knowledge for a design into a process where the relevant information and expertise is readily accessible would be desirable, saving both time and money. The design knowledge and information relating to the design can be expressed in many forms from tables to the manipulation of similar existing designs. This paper will concentrate on the use of artificial neural networks (ANNs) to express knowledge within a computerised design system, [1] & [2], currently being developed by a team at the Nottingham Trent University for mechanical transmissions.

Design System Overview
The design process is divided into stages, which represent phases the product passes through while the design is developing. These stages are sequential, with the output from one forming the input to the next. For simplicity the mechanical transmissions development has been divided into three stages, product design specification, conceptual design and detailed design. Each of these stages requires a combination of knowledge, expertise about the type of product being developed and the areas of expertise required to generate a successful design. As the stages of the system are different in nature, the knowledge required and manipulation methods are also varied, making allocation of knowledge to its relevant design stage recommended from the conception. The design stages may be classed into two types with respect to the representation of knowledge. The first class contains the product design specification and conceptual design. The knowledge and expertise necessary for these stages is of a creative nature, which specifies the design products requirements and generates concepts and solutions. The second class, the detailed design, is of an information retrieval and data manipulation form which develops the conceptual design into a tangible form, by the selection of materials, standard parts and calculation of dimensions.
The design system is divided into two stages which mimic the design process stages of concept and detail design. Product design specification is split into two types and implemented within the different design stages of the system. The first type of specifications cover the general requirements of the final product, e.g. power, ratio, etc. The second are specific to each component and only applicable if part of the concept design.
The conceptual and detail design stages of the system are embedded within a decision making structure, emulating the designer's expertise and enabling integration and correct matching of parts. Within the system structure the detailed design of the main components and their supporting parts is repeated until they are

capable of performing their functions without failure. Once the detailed design of each component has been completed it is checked to ensure failure does not occur.

Use of ANNs for Design

Artificial neural networks are used within the design system for two different applications, feature recognition and information storage and retrieval. Although the applications are different, multi-layer perceptron networks trained by back propagation are capable of achieving effective results for both applications.

Feature recognition. At the conceptual design stage the feature recognition capabilities of the ANNs are used to define the basic allocation of components and their arrangement according to which will best suit the design specification. Information contained within the networks is derived by a combination of means. The majority is based upon a weighted matrix as in fig 1, where the suitability of an arrangement of component to a specification is represented by the magnitude of its corresponding value, or weight. The remainder of the information to be help and reproduced by the network is in the form of 'exceptions to the rule', circumstances that do not agree with logical solutions, but either by past experience or by an expert's consideration will produce a successful solution. These exceptions to the rule are the main reason for implementing networks into the conceptual stage. The information relating to the exception may be simply added to the training data before commencement of training.

Information that will be held within the network may be represented equally well if not more precisely by a series of numerical calculations relating to design matrices in combination with production rules which covering any exceptions. However the network has a major advantage when a circumstance occurs which has never been experienced before. The network will, in effect, compare the new circumstance with other similar circumstances and match it to the most similar. This is similar to a designer drawing on previous experience to devise a solution that may be suitable and therefore adds an intelligent aspect to the design system.

	Specification				
Design Options	A	B	C	D	E
1	5	1	5	4	2
2	2	6	2	5	2
3	1	2	2	4	6

Fig 1 Design Matrix

Information storage and retrieval. A popular medium for representing information is by means of graphs or charts. This is due to the ease in which a large amount of information may be simplified into the form the designer may quickly use and retrieve. However manipulating graphical information by a computerised system poses difficulties and requires conversion into a suitable form. If additionally the original data is not available and must be taken from the graph or the data is empirically derived manipulation into a series of numerical functions may become complicated or impossible, even with mathematical tools such as MATLAB, without a great deal of expertise in mathematics. However the problem may be solved by the use of ANNs. The detailed design stage is the most information intensive stage of the design process and entails extensive use of information in graphical form. For the mechanical transmission application, some of the data contained within graphical form was unobtainable and therefore lead to the use of ANNs for representing the information contained within.

Back propagation (BP) networks have proven to be capable of emulating numerical functions and learn, from what appears to be, randomly scattered data to give the correct output to an input provided they are from within the same domain as the training data. It is for these purposes that the networks are used within the detailed design stage. Additionally the BP networks are capable of interpolating between curves on a graph with similar results to that a designer would achieve.

ANN Training

The basic principle of back propagation training is relatively simple. First present an input to the network and propagate through to obtain an output. Then derive the error between the networks and the target output and propagates the error back through the network, adjusting the connective weights between process elements. Repeat for every training example within training set then present entire set repeatedly until either convergence is achieved or training period has expired. However training a BP network can prove to be a time consuming process, requiring multiple attempts to obtain quality performance. Training difficulties are due to a number of factors effect the training of a BP network. The most predominant of these factors are the structure of the process elements or the topology of the network, length of training, transfer function, learning and momentum coefficients and the pass threshold. Successful training is achieved by manipulation of these factors. However different factors have different degrees of effect on the training process. From analysis of results taken during the training of two BP networks, one used for feature recognition (selection of a component) and the other to represent a graph it was found that the topology, length of training period and transfer function had the greatest effect upon successful training.

A number of observations were made in an attempt to develop a number of heuristics to reduce the number of training attempts with different configurations:

- Start with same number of elements in hidden layer as in there are inputs.
- Increase number of elements in layer until performance persistently decreases, then add extra layer with half the number of elements of the lower layer.
- Best performance is achieved with less than four hidden layers.

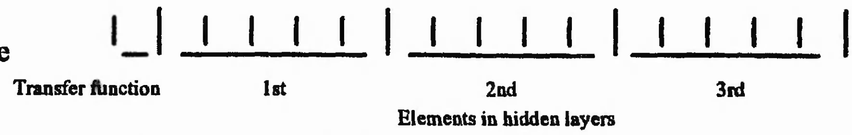
However these heuristics are vague and only found to be effective on the networks they were developed on. Although the factors that appeared to have the greatest effect on the performance of the network were the transfer function and the topology.

Genetic Algorithm Approach to Training

An adaptive search technique has been integrated with the ANN training process to aid the optimisation of the network topology and select the best transfer function for the application. The adaptive search technique implemented is that of a genetic algorithm (GA). GAs are based upon natural biological evolution. Just as living organisms evolved to cope with their environments by reproduction of the strong through generations, the genetic algorithm develops strings of parameters (called chromosomes), which characterise performance, by combining features of successful (or fit) chromosomes within a population to create the next generation. Unfit chromosomes are discarded from the population. As the generations progress the fitness of the chromosome should increase to a maximum. To ensure that a local maximum is not obtained a mutated chromosome is introduced into the population periodically, increasing the search area.

The chromosomes used to train the networks contain information about the transfer function and the network topology. This information is encoded in binary form as in fig 2. The structure of the chromosome allows for the selection of one of two transfer functions (sigmoid or tanh) and up to 3 hidden layers with a maximum of 31 process elements in each.

Fig 2. Chromosome Structure



The most important procedure when implementing GAs is the determination of fitness, as this governs which chromosomes reproduce and which perish. The fitness of the chromosomes is determined by the mean error between the ANN and target outputs after test. Lower mean errors suggest a higher level of fitness and therefore a better chance of the chromosome reproducing.

Discussion and Conclusion

Representation of knowledge by ANNs have enhanced the systems ability to generate new concepts and convert complicated information into a simpler form. The performance of the ANNs trained with the aid of the GA suggests that the GA approach to optimisation improves performance, removing the necessity for training heuristics. Automation of the training process also reduces the need for manual intervention which can be frustrating and time consuming.

References

[1]. D. Su, K.Henthorn, K. Jambunathan and M. Wakelam, 1996, *A Rule Based and Artificial Neural Network System for Conceptual Design*, International Journal of INGENIUM 96 (1), pp. 1-9.

[2]. Wakelam, K.Henthorn, K. Jambunathan and D. Su. 1996, *Integration of Multi-Media, Artificial Neural Networks and Rule Based Systems for Gear Design*, International Conference on Theory and Practice of Gearing, Izhevsk, Russia (forthcoming).

Printed and published by the IEE, Savoy Place, London WC2R 0BL, UK.

Integration of Multi-Media, Artificial Neural Networks and Rule Base Systems for Gear Design.

K. Jambunathan, M. Wakelam, K. Henthorn and D. Su.
Department of Mechanical Engineering, The Nottingham Trent University. England

Abstract
This paper applies an alternative approach to the computerisation of the design process for Spur and Helical gears. The computer system is designed to emulate the manual design process captures the designers' expertise and incorporate information that a designer has accumulated in the form of graphical design aids, but does not have the original source of data. A combination of knowledge representation techniques are employed to encapsulate both the designers' expertise and information relating to a successful design. Particular attention is paid to the use of Artificial Neural Networks for the representation of graphical information.

Introduction

Research into the design of involute spur gears has progressed in the last 100 years since the presentation of a paper by Wilfred Lewis on the design of gears taking into account the bending stresses acting on the teeth. However Lewis's initial breakthrough still forms the basis for the international BS, ISO and DIN gear standards today. The original Lewis formula has been developed and studies into the many factors that affect the performance of involute gears has created a vast amount of information, both theoretical and empirical, which direct the designer toward a possible optimum design. This information, in the form of numerous charts, diagrams, look up tables and calculations leads to a time consuming process that is open to wide interpretation between designers. To relieve the time consuming process of searching for the information computerised systems have been developed containing the relevant information required to form the design, removing the necessity for information searches and aiding the standardisation of parts. However these systems are difficult to develop without the original information which graphical design aids are created from. This paper addresses an alternative approach to the collection, reproduction and manipulation of conventional graphical design aids with the aid of Artificial Intelligence (AI). A computerised gear design system using this approach, combining the techniques of multi-media, Artificial Neural Networks (ANNs), production rules and numerical calculations has been developed by a research team at the Nottingham Trent University, forming part of an intelligent design system for a complete mechanical transmission.

Transmission Design System

The gear design system has been developed based on the observations of Niemann[1] and the ESDU 88033 design method [2], which itself is based on the British Standard for Spur

and Helical Gears [3]. The system designs the gear based upon tooth breakage and surface failure as these are the factors that the British Standard deals with.

The gear design system forms a module within an AI integrated system for the design of the entire transmission system. The transmission design system incorporates the design process from the initial concept to the finished drawings. A modular approach toward the design process is taken, breaking it down into the different elements and areas of expertise. Each module performs a different design task, e.g. the detailed design of a gear or shaft. This process is controlled via a central control program, which activates the relevant module dependant upon either the stage to which the design has progressed or if features of components clash and require redesigning. The control module iterates the design of each component until all the features of the transmission system are compatible with one another and checked to ensure failure does not occur. Fig. 1 shows a schematic of the transmission design system, demonstrating how the system is modularised.

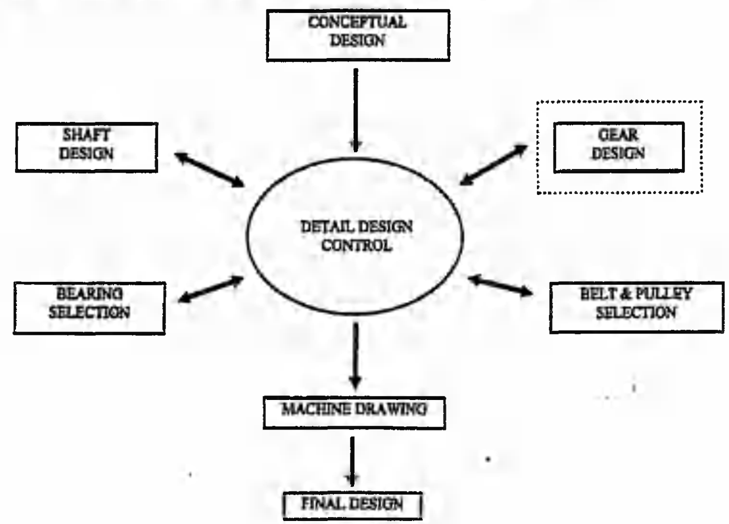


Fig. 1 Transmission Design System Schematic.

Gear Design System

The gear design system is activated by the Detail Design Controller. Once the module is activated control over the development of the gear design is passed to the process controller which structures the design process of the gear. The structure of this system is again of a modularised form, as shown in Fig 2.

Upon activation information about the required design is supplied to the gear module from the Detail Design Controller. This information is a combination of the users specifications and parameters defined at the conceptual design stage and by other design modules within the transmission design system. Additional information about the gears' application and required performance are required before the system proceeds with the initial and final design of the gear. This information is of the form that is impractical or impossible (at present) to be integrated into the intelligent features of the design system and is therefore extracted from the designer via the Graphical User Interface (GUI). The GUI serves two functions within the system. Firstly it displays the relevant information about a factor of the design in a clear and convenient manner, removing the time consuming process of wading through design manuals and specifications to locate the relevant information. Also examples of typical usage are available to aid in the selection of the correct criteria. Secondly the designer is given a sense of control over the final design which allows for variations from the normal, for example with respect to materials and manufacturing processes or the application factor with the aid of interfaces as in fig. 3.

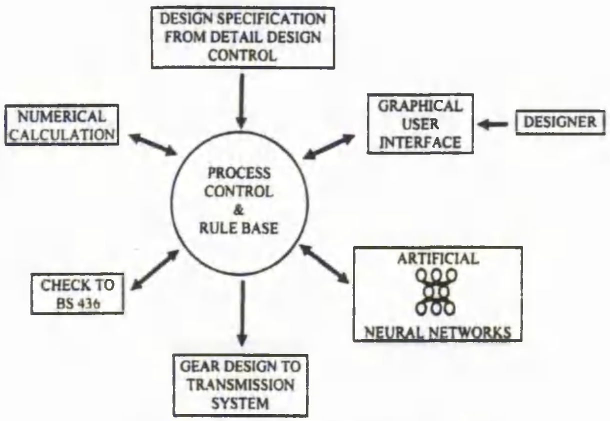


Fig. 2 Gear Design System Schematic.

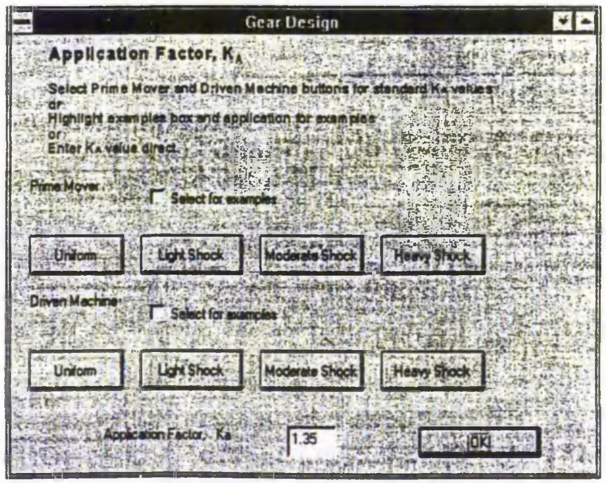


Fig. 3. Graphical User Interface - Specification of Application Factor, K_A

Once the gear specifications have been obtained the physical design of the gear is defined. This stage of the design process involves the integration of production rules, ANNs and numerical calculations. The production rules have two purposes. Firstly to structure the design process forming the design controller where they are used to iterate the design sequence to achieve optimum dimensions by making decisions, modifying factors of the design to achieve a feasible solution. Secondly for the allocation of the correct value to a feature, dependant upon existing features of the design i.e. facewidth ratio dependant upon gear arrangement within the transmission system, the material properties and chosen manufacturing process. The rules are used to encapsulate knowledge about the design process that is tangible and well definable, forming the backbone of the design system.

ANNs add a feature to the gear design process that is not available in the conventional gear design packages. The application ANNs have been applied to within the system is to encode information that is of a graphical form. Many designers accumulate information from a variety of sources often in graphical form. The information held within the graph may be extremely helpful to the design of a successful component but the original data that the graph is constructed from may be unobtainable. In this circumstance an ANN trained directly from the graph forms a desirable solution allowing the manual design process to be followed. The ANNs used are capable of encapsulation information of both a simple and complex nature. Activation of the appropriate network is governed by the control system which follows the simplified route shown in fig. 4.

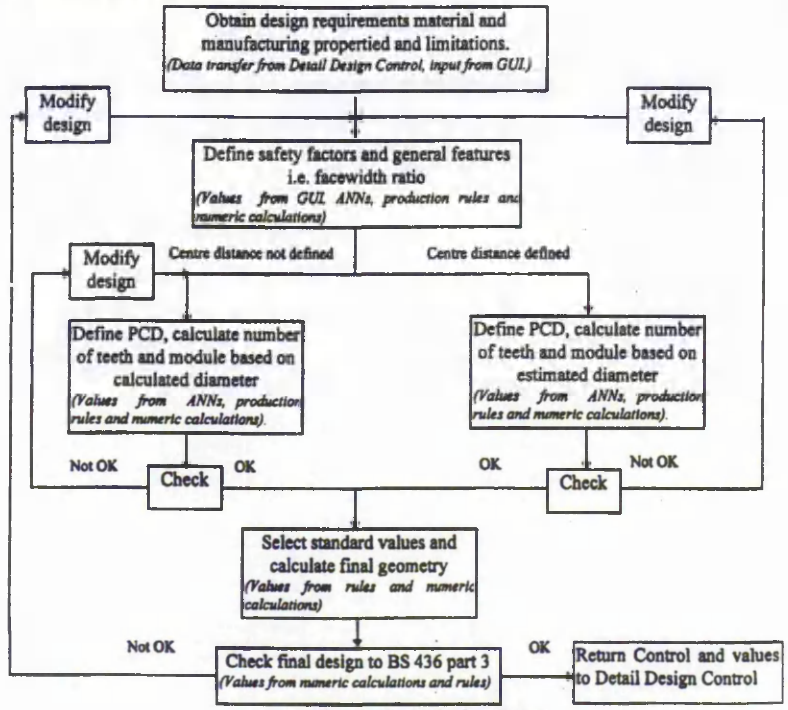


Fig. 4. Gear Design Route

Activation of the appropriate procedure, which houses the numerical calculation, is governed by the control system. The numerical calculations fall into two classes. The first class relates to the design of the gear geometry and properties, while the other checks the final design. The gear geometry numerical calculations, derived from [1],[2] and [3], provide the tangible features of the design e.g. dimensions and number of teeth. Final check of the design is performed to BS 436 part 3 to ensure that failure does not occur. If it is identified that failure does occur the reason for failure is identified and modifications to the gear specification are made by the production rules prior to re-design.

Once the final has passed BS 436 part 3 then the final design and control is passed to the Detail Design Controller for continuation of the transmission design.

Performance of ANNs

Performance of a network is dependant upon a suitable network type for the application being used and the training undertaken. The networks embedded within the design system are multi-layer perceptron trained with the back propagation technique. These networks have proven successful at ordering what appears to be random information, emulating a function to map the data and capable of interpolation within the limits of the training data. These properties are ideal for the design systems purpose.

Training data for the networks is based upon the information the designer would require to enable use of a graphical design aid and the output expected which forms the inputs to and output from the network respectively.

Fig. 5 shows a graphical design aid that has been incorporated into the design system while fig. 6 demonstrates the accuracy of the equivalent network.

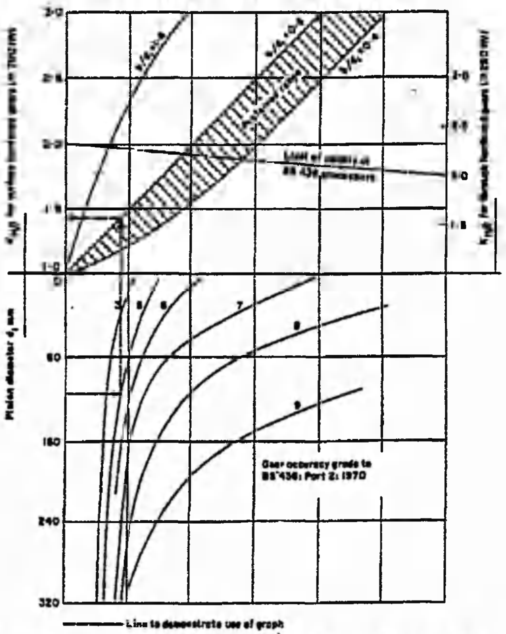


Fig. 5 Diagram for face load factor for contact stress $K_{H\beta}$ [3]

The network to represent $K_{H\beta}$ has been trained with a set consisting of 262 examples directly taken from fig. 5 and tested with 12 sets consisting of both information from the graph and interpolated values of the facewidth ratio, b/d_1. The results in fig. 6 demonstrate that the network is capable of representing the successfully well to be incorporated within the design. Fig. 6 only represents a small test of the information held by the trained network. Additional accuracy may be achieved by extending training of the network, incorporating an adaptive search approach to optimising the networks parameters[4].

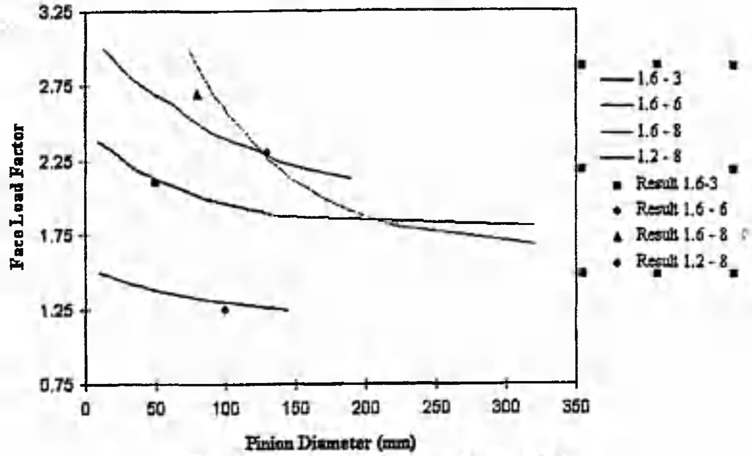


Fig. 6 Achieved performance from ANN

Discussion / Conclusion

The alternative approach to the computerisation of the design process for a gear closely resembles the manual procedure but without the time consuming information searches that occupy a large amount of a designers' time. Use of ANNs to represent graphical design aids within the design system allows, for example, an existing design processes within a company to be kept providing a smoother transition from manual to computerised methods. The gear design system is an example of successful integration of the knowledge representation techniques which may be applied to other information intensive design processes.

References

[1].G. Niemann, *Machine Elements Design and Calculation in Mechanical Engineering, Vol.2 Gears*, Springer-Verlag, 1978

[2].ESDU 1988, *The Design of Spur and Helical Involute Gears, Item 88033*, Engineering Science Data Unit International plc., London

[3].British Standards Institution, BS 436 part 3: 1986, *Spur and Helical Gears. Part 3 Methods for Calculation of Contact and Root Bending Stress Limitations for Metallic Involute Gears*,

[4].M. Wakelam and D. Su, 1996, *Knowledge representation incorporating artificial neural networks for design of a mechanical transmission system*,proceedings of the IEE Knowledge Discovery and Data Mining Colloquium, 17th-18th October 1996.

INTELLIGENT INTEGRATED SYSTEM FOR THE DESIGN OF POWER TRANSMISSION SYSTEMS

Daizhong Su   *Mark Wakelam*

Department of Mechanical and Manufacturing Engineering
The Nottingham Trent University, Burton Street, Nottingham, NG1 4BU, UK.
Tel: +44 115 941 8418 ext 2306, Fax: +44 115 948 6506, E-mail: mec3sud@ntu.ac.uk

Abstract This paper presents an intelligent integrated system (IS) approach and its application to the total design of power transmission systems. The approach fully integrates all stages of total design including formulation of product design specification, conceptual design, detail design and manufacture. This is achieved by using state of the art AI techniques to integrate various activities within design process. The IS blends a rule based system, artificial neural networks, multimedia and CAD/CAE/CAM packages into a single environment to provide a powerful design tool.

Key words Artificial intelligence Integrated design Mechanical transmission Computer aided design and manufacture

1 INTRODUCTION

Mechanical power transmission systems are built from drive units such as belts, chains and gear pairs, together with other relevant mechanical components. The process of total design includes formulation of product design specification, conceptual design, detail design and manufacture [1]. To design a power transmission system throughout the process is a tedious and time-consuming task. In order to reduce production cost and time-to-market, it is highly desirable to automate the process.

This research aims to develop an intelligent integrated system (IS) approach to integrate all the stages of total design, and to apply it to the design of power transmission systems. This is achieved by blending knowledge based systems (KBS), artificial neural networks (ANNs), genetic algorithms (GA), multimedia and CAD/CAM/CAE into a single environmentto provide a powerful design tool.

Both design integration and the application of AI in engineering are currently attractive research topics with a number of successful applications, however, integration of multi AI techniques with various CAD/CAE/CAM into the total design process have never been reported, hence, this research provides a novel approach to total design.

2 SYSTEM OVERVIEW

2.1 Software integration

The software technique and packages/programs involved in the IS and their relationship are shown in fig. 1. Integration of the various elements into a single environment is achieved via the *control* which is a KBS. The main elements of the IS are briefly described below:

The *KBS* plays a leading role interacting between the elements to control the design process. The KBS is a production rule based system developed using C++.

The *CAE* currently consists of numerical calculation programs and a commercial package, FennerBelt. The former are for strength analyses of shafts and gears, while the latter is for belt and pulley selection.

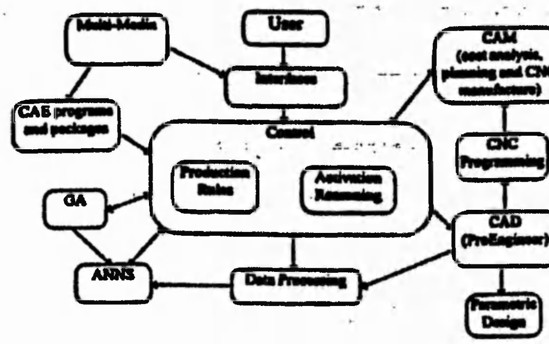


Fig. 1 Software integration

Artificial neural networks: four ANNs are developed to generate concepts in the concepual design phase, four ANNs are used for detail design to derive design coefficients, and another one is employed for structural design to retrieve existing design from databases.

Two *GA* programs are implemented: one for the optimisation of ANN architecture, the other for searching the best combination of gear design parameters.

The *Hypermedia* is applied in two places in the IS: the graphic user interface (GUI) and front end for the CAE package Fenner Belt. The GUI is used for the user to input the design parameters and to specify PDS, which is developed using Visual BASIC [8]. The front end is developed using a Hypermedia tool, Linkway [2], which provides facilities for the user to input data to FennerBelt, and to explain the terminology involved in the design.

CAD: ProEngineer is integrated into the IS to produce component and assembly drawings using parametric design method and to produce CNC programs for the designed components.

The *Data processing* includes (1) data preparation for training ANNs, for which a spread sheet method has been developed using EXCEL, (2) databases for the data used in the design including materials, bearings, etc. and (3) data transfer between packages/programs and the KBS.

The *CAM* consists of programs for cost analysis, manufacture process planning and CNC manufacturing.

2.2 Functions of the IS

As a prototype system, the IS has currently been developed to perform the following tasks:

a. *Formulation of product Design Specification (PDS)*. 12 PDS items, such as orientation of input/output shafts, manufacture cost, etc., are considered. It is possible to extend the system to aid more PDS items.

b. *Conceptual Design*. The concepts to be constructed by the prototype IS fall into the following range

- stages of the transmission: one, two or three;
- orientation of input/output shafts: parallel, cross and perpendicular;
- components at each stage of the transmission: seven types of components including gears, belts and trains;

c. *Detail Design*. The IS can conduct the detail design of gearboxes, belt drives or combination of both, including the following: gear strength analysis, bearing selection, shaft design, case design, belt and pulley selection, design optimisation, component and assembly drawings, and parametric design of components.

d. *Manufacture*. The IS can perform the following tasks: cost analysis, process planning for the manufacture of major components, and CNC programming for manufacture of shafts.

3 MAIN FEATURES OF THE SYSTEM

In this section, the design process and system control are described first, followed by the two key parts of the IS, conceptual design and gear design. Due to the restriction on the length of the paper, other tasks are only outlined in section 2.2 and their further details are given in [3].

3.1 Design process and system control

The design process is shown in Fig. 2. The product design spectifications are formulated first, then the IS creates all possible concepts to meet the specifications and selects the best concept for detail design. At the detail design stage, two types of sub-systems are considered, belt drives and gearboxes, from which a power transmission system is built; and the design of components such as gears, shafts, and bearings are also carried out. After completion of the detail design, the IS moves to manufacturing stage where CNC, planning and costing are to be conducted. The IS can also carry any relevant redesign task whenever it is necessary.

The whole process is fully integrated and controlled by a system controller (SC) and sub-controllers (SubC). The SC controls the overall process by communication with the SubCs, while a SubC controls the design activities within a design stage. As shown in Fig. 2, relevant software tools/programs/techniques are involved at each stage.

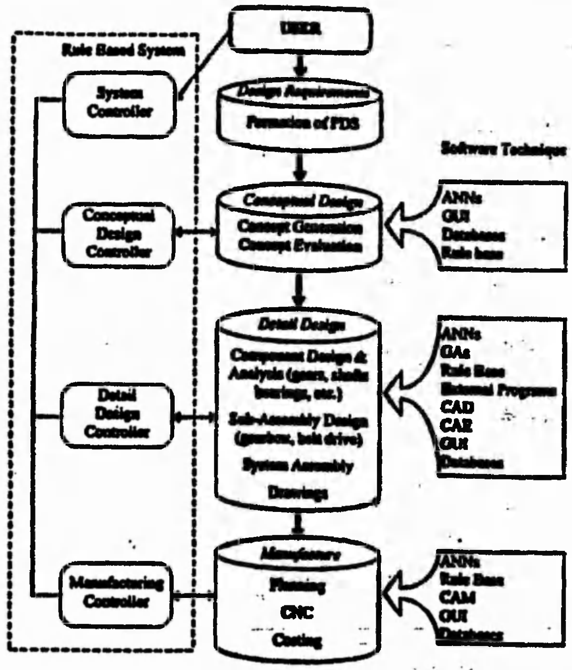


Fig. 2 Design process

The SC and SubCs are rule based systems comprising two types of rules: *control rules* and *information rules*.

The *control rules* hold the knowledge relating to the designs progress. These rules stucture the design process and control its development, forming the inference engine and activating the appropriate information rules depending on the circumstances applied and the design stage. For example

IF *facewidth ratio is not defined by user*
THEN *activate facewidth information rules*.

The control rules also control the activation of other modules within the design systems. For example

IF *gear is through hardened*
THEN *activate ANN module* 1.

The information rules contain information in the form of numerical values and design features or activate an equation that encapsulates the design information. For example

IF *the gear is double helical and its heat treatment is nitrided and its mounting is symmetric*
THEN *maximum facewidth ratio is* 1.4.
IF *heat treatment is case carburised*
THEN max. *number of teeth* $=\left(\frac{11}{9}\right)$ *gear ratio*-37.33

3.2 Conceptual design

The conceptual design module consists of a combination of ANNs and rule bases. The rule bases cover the situations with clear decisions one way or the other, while the ANNs are used in hazy situations.

The use of the rule bases and ANNs are controlled by a control program which executes the design process, calling upon both the rule base and ANNs as and when required. The control program also acts as a pre and post processor, preparing the inputs from the user interface then analysing and displaying the results. Fig. 3 below illustrates how the control program co-ordinates the conceptual design.

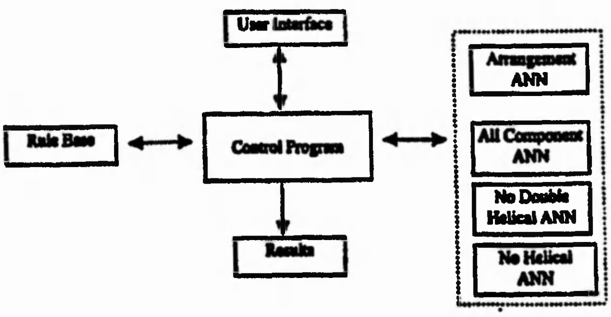


Fig. 3 Conceptual design module

A concept is formed by assembling component (s) into the position(s) of an arrangement, which is illustrated below:

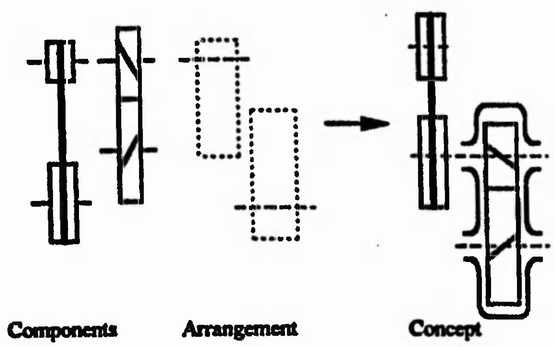


Fig. 4 Concept construction

The IS determines the transmission system arrangement first. This is performed by the arrangement ANNs. Before the ANNs can produce a suitable design its input pattern must be generated. A combination of rules and user input are used to generate the input pattern. The rule base generates the first two parts of the input pattern that correspond to the orientation and the number of stages, while the remainer consists of the user defined design specifications. Using the input pattern, the ANNs produce a suitability rating for all the possible transmission system arrangements. The control program then analyses the results of the ANNs ranks the designs obtained in order of suitability and then presents the best solution together with two alternatives to the user. The program also allows the user to change the decision if it is not considered satisfactory.

Once the design arrangement has been finalised the components will be determined using both the rule base and a combination of ANNs. The rule base selects the components regarding shaft orientations and determines the correct ANN to use. There are three component ANNs which contains either all the parallel components, the double helical gear removed or both the double and single helical gears removed.

When all the parameters of the design have been obtained the control program displays the finished conceptual design to the user and records the results in the database. Upon completion control of the design process is returned to the central control in the higher layer.

3.3 Gear design

Within the IS, the gear detail design consists of a combination of fules, ANNs, formulae, user interface and knowledge modules. The design is co-ordinated by a central control program that activates the appropriate rules, ANN or Graphical User Interface (GUI) module dependant on the stage of the design. Fig. 5 below shows how the modules interface with one another via the control rogram.

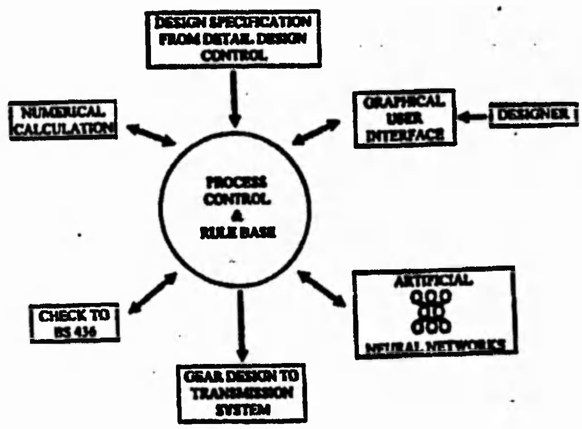


Fig. 5 Gear design module

Upon activation, information about the required design, such as power, input speed, gear ratio and centre distance, is supplied to the gear module from the Detail Design Controller. This information is a combination of the users specifications and parameters defined both at the conceptual design stange and by other design modules within the transmission design system.

Additional information about the gears application and requied performance, such as application factors, material and manufacture route, is required before the system proceeds with the initial and final design of the gear. Some of the information is of the form that is impractical or impossible (at present) to be integrated into the intelligent features of the design system and is therefore extracted from the designer via the Graphical User Interface (GUI).

Once the gear specifications have been obtained the IS goes through the process of gear sizing to determine the geometric parameters including module, number of teeth, diameters, helical angle, etc. The process is shownin Fig. 6.

After completion of the gear sizing process, the IS runs a program for the gear rating of contact and bending strength to BS436 part 3. If the design fails to the rating, the IS redesigns the gear pair until it passes the rating.

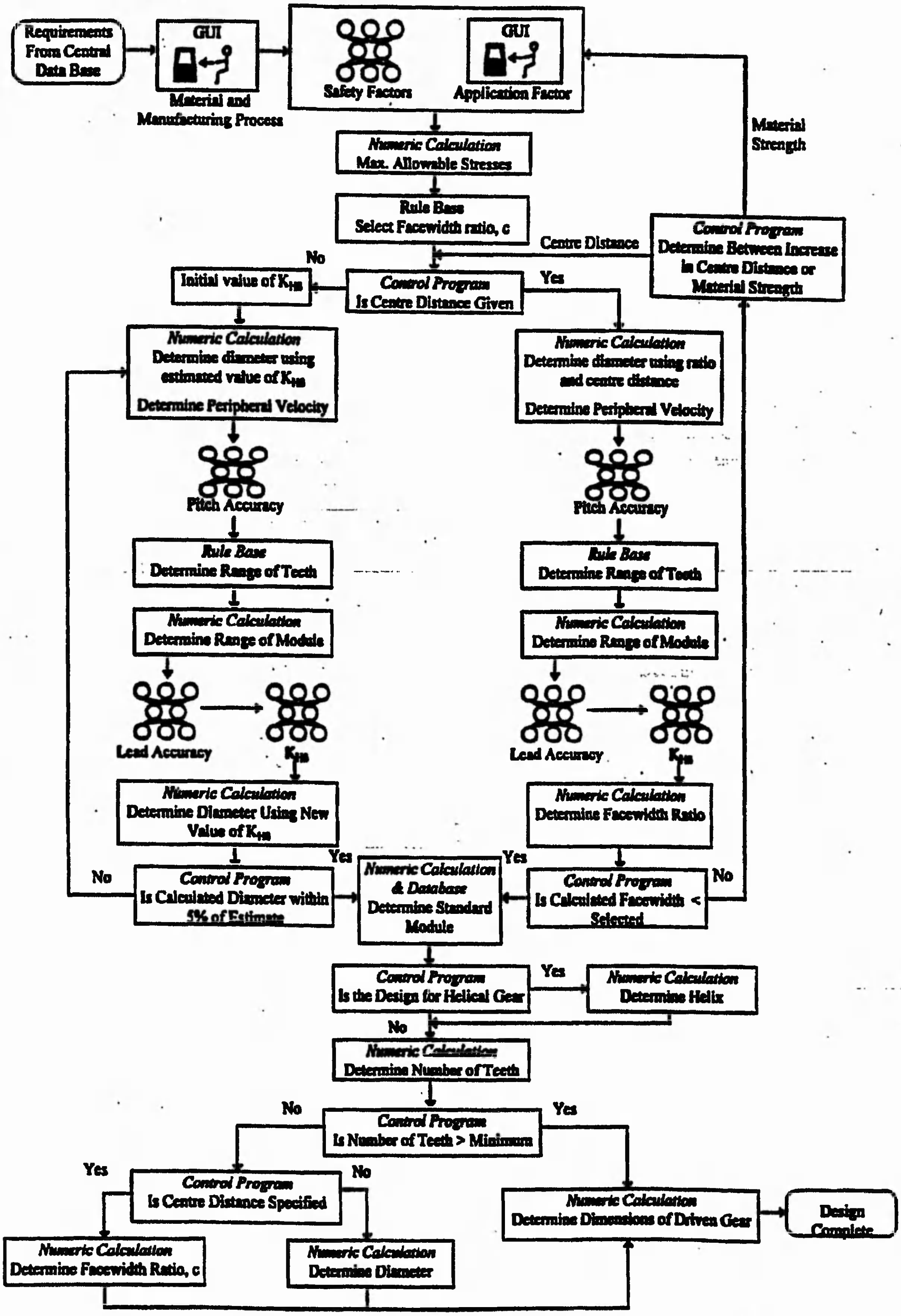


Fig. 6 Gear design process

Once the final design has passed BS 436 part 3, control of the process is passed to the Detail Design Controller for continuation of the transmission design.

The gear design process described above is conducted by the combination of production rules, numerical calculation and ANNs.

The production rules are used to control the design process and allocate correct values to the required parameters, which have been mentioned in the previous section 3.1.

The numerical calculations are carried out by sets of programs, which fall into two classes. The first class relates to the design of the gear geometry and properties, such as dimensions and number of teeth, while the other checks the design including the final check of the design to BS 436 part 3.

The ANNs add a feature to the gear design process that is not avaiable in conventional gear design packages. The ANNs are applied to encode information that is of a graphical form. As shown in fig. 6, four ANNs are applied to obtain the safety factors, pitch accuracy, lead accuracy and load factor $K_{H\beta}$. The graphs for the derivation of those parameters are for use in the manual design process, and normally the original data that the graph was constructed from are unobtainable, which makes it difficult to include them into a computer integrated program. However, the application of ANNs solved this problem. In this circumstance, the ANNs are trained using the data obtained directly form the graphs, which provides a desirable solution to encode the design process into the system.

4 DISCUSSIONS AND CONCLUSIONS

1 This research results in a computer integrated approach for total design, with its application for power transmission systems. The approach blends multi artificial intelligence (AI) techniques, together with CAD/CAE/CAM into a single environment to provide a novel design tool.

2 The IS approach is the further extension of the author's previous research in knowledge based integrated system (KBIS) approach for engineering design, which was first reported in [4] and further developed in [5, 6, 7]. The differences of the IS from KBIS are

(1) more AI techniques, i. e., artificial neural networks and genetic algorithms, are employed in the IS. This results in a system that benefits each AI techniques' strength and avoids their short fallings.

(2) more subsystems such as belt drives, and more total design stages such as manufacture have been included, hence, the total design route is completed

(3) more advanced software techniques such as multimedia and ProEngineer are employed.

(4) the IS has a structure that enable updating and modification of knowledge.

REFERENCES

1 S pugh, Total Design, Addison-Wiley, 1984

2 Linkway, Version 2, IBM

3 M Wakelam. Intelligent integrated system approach for total design. Report for transfer from MPhil to PhD, Department of Mechanical and Manufacturing Engineering, The Nottingham Trent University, Nottingham, January, 1997

4 Su Daizhong & R J Forgie. Development of an Expert System Approach to an Engineering Design Procedure. Proceedings, International Conference on Engineering Design, Harrogate, U. K., 22nd-25th August 1989: 975~989

5 Daizhong Su. Application of a Knowledge Based, Integrated System Approach to the Conceptual Design of a Power Transmission System. in: W. Kuo (ed.). Quality Through Engineering Design. Elsevie Science Publisher B. V., 1993: 417~427

6 Daizhong Su and K Jambunathan. A Prototype Knowledge-Based Integrated System for Power Transmission Design. in: Advancement of Intelligent Production. Eiji Usui (ed.), Elsevier Science B. V., 1994: 45~50

7 Daizhong Su, 1994b. Gearbox KBIS: a Prototype Knowledge-Based Integrated System for Gearbox Design. proceedings, 1994 International Gear Conference, 7-9, September, Newcastle upon Tyne, U. K.: 201~205

8 Visual BASIC, Version 4, Microsoft Corporation

Application of Artificial Intelligence into the Design and Manufacture of Mechanical Power Transmission Systems

K S Henthorn, M Wakelam, K Jambunathan, D Su
Department of Mechanical and Manufacturing Engineering, The Nottingham Trent University, Nottingham. NG1 4BU

ABSTRACT

Integration of the design and manufacture process into a single, continuous system ensures continuity of design. Combine expertise and reasoning through a combination of artificial intelligence techniques and the system becomes capable rapidly forming designs and manufacturing process information from knowledge indefinitely available. The system has been applied to the production of a mechanical transmissions, generating and analysing designs then linking with a CAD/CAM package to produce 3D models prior to manufacture.

1. Introduction

Due to the ever increasing economic and financial pressures imposed on industry, from competitors and advances in technology, time-to-market has become an increasingly important factor to be considered in the initial stages of a products life cycle. Additionally, increased mobility of the work force (the primary source of expertise and knowledge within a company) has developed a need for a method of retaining expertise within the company. These two factors have led to the development of a computerised system that reduces design time and automatically links with manufacture whilst incorporating expertise and knowledge in an intelligent manner. The traditional approach to the design process of concept, evaluation and redesign is still the most effective design route for generating an optimum design. Therefore a system that adopts the traditional approach, but is enhanced with the capability of encapsulating, manipulating and evaluating expertise, while structuring and co-ordinating the design and manufacturing process will form the solution.

Currently a multitude of methods are employed to perform the design stages and initial manufacture of a product. Each method performs its task effectively, but generally within a separate environment resulting in manual interaction and repeated specification of performance, a tedious and time wasting process. Integration between the various stages within the design phase of a products life cycle is an essential process in the attempt to develop the best solution.

Research into the development of systems to solve this problem has been conducted for a number of years. The continuing results of which are displaying increasing possibilities. A review of these systems [1] indicates that the majority are expert systems (ES) or knowledge based systems, producing valid results. However ES and KBS display an inherent problem. Due to their structure, the systems have difficulty evaluating situations previously not encountered and allowing for modification of structure or update of expertise. This inflexibility is one of the reasons for the development of this intelligent hybrid system (IHS) together with advantages that the combination of different artificial intelligence (AI) techniques can offer.

2. Intelligent Hybrid System

The hybrid system has been developed to combine the AI techniques of rule based systems, artificial neural networks (ANNs) and genetic algorithms (GAs) with multimedia and CAD/CAE/CAM packages within a single environment. The IHS employs the AI techniques to reduce the dependency upon the users expertise by manipulating knowledge installed within it for the generation of designs, make decisions and control the products development.

The system is structured to follow the product from its conceptual design through to manufacture and costing. Within each of these stages a combination of techniques are employed to control the progress of the products development by replicating expertise and acting upon installed knowledge. Heuristic rules offer structure and logic while the ANNs offer both structure and flexibility depending on the type, supervised or unsupervised. This combination of techniques allows rigid rules and circumstances to be dealt with by rule bases and databases, as with previous systems and takes advantage of the networks ability to encapsulate and manipulate knowledge that is difficult to quantify or structure.

Throughout the system a modular structure has been maintained and emphasised, yet always contained within the single environment. A modular approach has been developed as it naturally tends toward easier integration of existing packages, modification and combination of AI techniques. The modular structure is demonstrated in Figure 1.

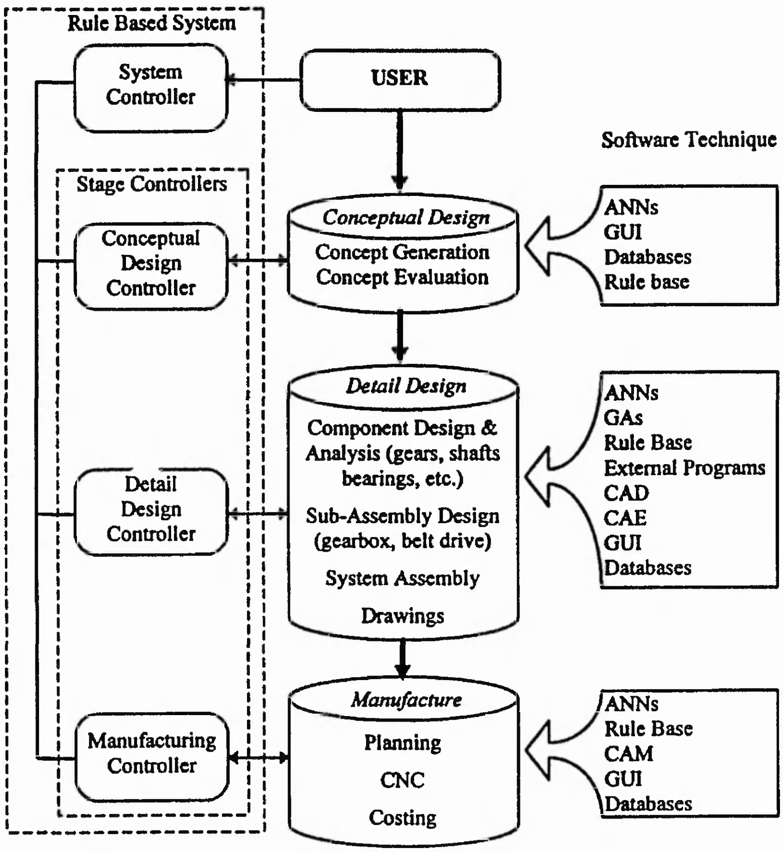


Figure 1: System Structure

The entire system is controlled by a rule base which forms the central control of the system. This rule base reacts to results from the various stages of the design and the progress of the products development to determine the next course of action. In this respect the system displays many of the characteristics of an ES. Control over each stage of the process is governed by a rule based system specific to that stage. These stage controllers control the progress of the product through that stage, activating the appropriate task module, supplying and storing the relevant information as and when required. Control of the process is returned to the system controller upon completion following the hierarchy of control within the system.. Throughout the system, interaction with the user is necessary to evaluate situations where specifications are required or to evaluate a situation when AI is inappropriate. The interface with the system is established via multi-media. The interfaces are designed to guide the user towards making a decision or entering and displaying options in a user-friendly form that are concise and simple.

At the conceptual stage, the specifications supplied by the user are analysed and by a combination of rules and ANNs viable solutions are generated [2]. This concept is then broken down into its main components within the detailed stage. Detailed analysis of the components is performed, ensuring that the design will not fail due to excessive force or interference with mating parts. Upon completion of the detail stage the relevant information necessary to generate the product has been established and checked for failure. However if after iteration of the design the design continually fails, the system controller reactivates the conceptual stage to enable another concept to be generated. Once the design stages have iterated to generate a viable solution, the system integrates and controls the CAD/CAM package, Pro/ENGINEER, to perform the visualisation and preparation of CNC machining codes for the design and its components. The connection between design and manufacture is achieved through an interface developed in Pro/DEVELOP, Pro/ENGINEER's customisation toolkit, through which the facilities of Pro/ENGINEER are made available. Dimensions, limits and specifications of the product that are determined at the design stages are applied to parametric parts that have been previously created. These parametric components are assembled to give a full 3D model of the product which may be modified if this is required. The link enables the manufacturing capabilities, such as machining, to be carried out.

3. Intelligent Techniques applied within the System

Each of the AI techniques employed within the IHS performs a task that exploits its advantages and compensates for short comes in others. This combination of techniques therefore forms a hybrid system that has the ability to compensate for limitations on single technique systems, such as the rigidity of a true ES. The different techniques display different qualities and are applied and best suited to different situations and applications. Therefore it is important to identify the nature of the decision and the characteristics of the different techniques to select the most suitable for a particular application.

3.1. Production Rules

The production rules used within the system are a series of conditions and actions in the form of IF...THEN statements where the entire collection of rules forms the systems knowledge base. Rules are applied when the knowledge is in a well-defined form, where a known reaction will be the result of a certain action or circumstance. The production rules are in two forms. Those concerned with controlling the progress of the design (if a condition is met then perform a task) and those containing specific data or information about a particular design configuration or property. The control rules hold the knowledge relating to the progress of the design. These rules structure the design process and control its development, forming the inference engine, activating the appropriate information rule depending on the circumstances applied and the design stage. The information rules are sets of simple rules containing information in the form of numerical values, design features or activate an equation that encapsulates the design information. Activation of the these rules are governed by the controlling rules.

3.2. Artificial Neural Networks

ANNs are used within the IHS to store and manipulate knowledge that is of an ill-defined manner or of a form that is impractical to encode by other means. The ANNs used within the system are supervised backpropagation networks. These networks require examples of the task they are to perform but once trained posses the capability of interpolation between training examples and decide between equally matched solutions. These capabilities allow for circumstances that have previously never been encountered and remove the necessity for secondary decision making.[2]

3.3. Genetic Algorithms

GAs are an adaptive search technique based upon the evolutionary process. These are used to aid the training of ANNs and implemented within the IHS at the design stage to increase the performance of the design and increase quality. Increasing the quality of design is an important feature of the system when the product is to be exposed within a competitive market. GAs are applied to the optimisation of values where a means of establishing their fitness can be determined, as the fitness is the controlling factor of the search. The initial values that constitute the genetic population comprise a band of values taken from the calculated starting point, together with a percentage of random mutations. The GA then evaluates the fitness of the gene, dependant upon the fitness function and grades it with respect to the rest of the population. The genes then reproduce, with the fitter genes having a higher probability of reproduction. The process is repeated until an optimum is obtained.

4. Mechanical Power Transmission System

The IHS has been applied to the design and manufacture of a mechanical power transmission. The system has been developed to generate the conceptual design, analyse the design and perform detailed analysis of the constituent components then generate a 3D CAD model of the final design ready for manufacture.

4.1. Conceptual Design

The conceptual stage is the creative stage of the products development. Within this stage the orientation and layout of the design is determined, together with the principal components for each of the transmissions stages [2]. Knowledge relating to this stage is encapsulated through the combination of production rules and ANNs. Rules govern the orientation and component hierarchy while the ANNs contain information about the characteristics of each component and makes decisions and selection based upon it.

4.2. Detail Design

Detail design is the most information intensive stage of the products design, requiring information about multiple component designs and their effects upon the product. Within the detail design of the transmission, the initial sizing, optimisation and checking of the constituent components that comprise the transmission are performed by separate modules. Such as the initial sizing and feature module for spur and helical gears [3]. Modularising the tasks allows for ease of modification without affecting the remainder of the design. Existing design packages form modules that integrate through swap files with the task controller, incorporating existing computerised expertise. Information relating to the components is stored and accessed to and from a central data base, ensuring that duplicate or obsolete data is not used, leading to errors.

Optimisation of the transmissions components is performed in two stages. Firstly through the iteration of the components designs until failure does not occur, performed by the task controller. Secondly using GAs to optimise dimensions and features of a component, such as the optimisation of gears, using the British Standard [4] as the fitness function.

4. 3. Design Completion and Linking for Manufacture

A library of parametric components are created which only require critical dimensions and properties from the detailed design stage to allow the complete component to be generated. For example the spur gear, in Figure 2, is generated requiring only the width, PCD, number of teeth, pressure angle and shaft diameter.

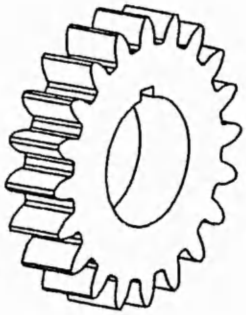


Figure 2: Parametric Component. Spur Gear

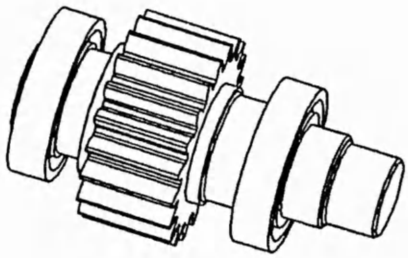


Figure 3: Parametric Input Shaft Assembly

Combining the components into assemblies and establishing limits and interference warnings allows the design to be generated with an extra degree of confidence. Figure 3 demonstrates the sub-assembly of the input shaft. Positioning of the components has been done so that modifications to components will have a knock-on effect to related components causing automatic modification to components and assembly. Modification to the components is made possible, allowing the user to exercise their judgement to have a direct effect upon the final design. All these facilities are available manually through Pro/ENGINEER, but using the Pro/DEVELOP link, the facilities are activated automatically, thereby decreasing the modelling time dramatically and ensuring consistency of design.

5. Discussion and Conclusions

Incorporation of design and manufacturing within a single environment and the transfer of common data encourages continuity of design. Continuity of information is achieved by the design sharing a common source of information, which prevents errors occurring during the transfer of information from task to task and stage to stage. Thus reducing the occurrence of expensive design failure due to incorrect calculations. An important feature of the IHS, as redesign at the point of manufacture can be extremely costly and increases time-to-market.

Combining AI techniques increases the possibilities that a computerised system can achieve. From the point of view of developing an intelligent integrated system, opportunities become apparent that will decrease the necessary level of the users expertise by increasing the decision making capabilities of the system. This retains expertise indefinitely, compensating for the departure of experienced personnel from the company or department. However this is not to say that an IHS will replace invaluable experience and expertise.

6. References

1. Wakelam M, Report for transfer from MPhil to PhD, Department of Mechanical and Manufacturing Engineering, The Nottingham Trent University, 1997.
2. Su D, Henthorn K S, Jambunathan K, Wakelam M, A Rule Based and Artificial Neural Network System for Conceptual Design, *International Journal of INGENIUM 96 (1)*, pp. 1-9, 1996.
3. Jambunathan K, Wakelam M, Henthorn K S, Su D, Integration of Multi-Media, Artificial Neural Networks and Rule Base Systems for Gear Design, *Proc. Int. Conf. Theory and Practice of Gearing Russia 4-6th Dec. 1996*, pp 463-468.
4. *British Standards Institution*, BS 436 part 3: 1986, Spur and Helical Gears. Part 3 Methods for Calculation of Contact and Root Bending Stress Limitations for Metallic Involute Gears.

DEVELOPMENT STRATEGIES FOR INTELLIGENT DESIGN INTEGRATION

M. Wakelam, D. Su, K. S. Henthorn and K Jambunathan
Department of Mechanical and Manufacturing Engineering
Nottingham Trent University, England

ABSTRACT

A design strategy towards enhancing the design process is described within this paper. The approach combines the stages of the design process from specification creation to detailed design within a single environment, utilising a combination of artificial intelligence techniques to simulate the decision making qualities and knowledge of expert designers. Successful development of such integrated systems will enable the automation of future designs with reduced dependency upon specialists.

INTRODUCTION

Total Design, as defined by Pugh (1990), 'is the systematic activity necessary, from the identification of the market / user need, to the selling of the product to satisfy that need'. The major elements of Total Design considered within the design system include:

- Specification: formulation of product design specifications.
- Conceptual design: concept generation and evaluation.
- Detail design: analysis, sub-assembly/ component design, assembly, engineering drawing and design retrieval.
- Manufacture: the manufacturing processes required to produce a product have been considered throughout the design approach developed, but have not currently been implemented at this stage of development.

Due to the numerous stages a design must pass through and the information and expertise that is required for each, the entire process can be laborious and time consuming. Therefore redesign, if required, toward the final stages of the design process can be costly if performed manually. Since the introduction of computers in the 1940s, engineers have been applying their rapid computational qualities to all aspects of engineering, including design. The large amounts of information required by the individual stages of the design process can be stored and rapidly retrieved and manipulated, giving a speedy response. However the application of computers for design has generally concentrated on the reduction of time taken to perform individual design tasks. Interactions between the tasks are generally not considered, together with their effects upon the form and structure of the design stages.

Integration of the design tasks and stages, forming a single complete design system appears to be the solution as design problems and redesign can be achieved relatively quickly and cheaply. Several integrated design systems have been developed, (Su (1993), Kaftanoglu

(1995), amongst others), however due to the shear diversity of design they have been limited to particular design areas.

Increasing research into the development of design systems has led to the incorporation of Artificial Intelligence (AI) to emulate the human decision making process. As the design process is not just a series of calculations, but also the application of experience to the selection and generation of solutions, the application of AI within the system will increase its performance and versatility.

Although AI has been implemented into the design process in the form of Expert Systems, sometimes referred to as knowledge-based or rule based systems, successfully, they have displayed limitations with regard to flexibility. Additionally the use of other AI techniques such as Artificial Neural Networks (ANNs) and genetic algorithms (GAs) have not been extensively investigated. Advantages that these techniques may offer and methods of integrating them into the design process need to be explored further to develop their full potential. Therefore a system that applies a blend of state of the art AI techniques for design would result in an intelligent design system with increased efficiency.

INTELLIGENT INTEGRATED SYSTEM (IIS) APPROACH

The IIS approach draws together all the design stages, knowledge and expertise required to perform the design process for a product within a single environment. The single environment is the key to the approach, allowing the products design to progress from stage to stage without the duplication of information, repetitive requests for the designer and dependence upon several experts in different fields of engineering. The approach combines all these elements together through the integration of AI techniques and multimedia within the system. The AI and multimedia when combined provide a hybrid medium for the extraction and encapsulation of knowledge with the intent of replacing, guiding or tutoring the designer using the finished system.

The design strategy for developing the IIS consists of three inter-linked parts. The first is concerned with the design process and the stages the product must pass through during its development. The second part relates to the knowledge acquisition and categorising process for identifying the components of the integrated system and implementing them within the design process. The final section relates to the modular structure of the system, ensuring ease of knowledge alteration.

Design Process

According to Pugh's (1990) *total design activity* model or French's model (1985), which are widely accepted by most researchers and designers, the design procedure can be modelled in 5 basic design stages: i) identification of need, ii) specification or requirement, iii) concept generation, iv) detailed design and v) manufacture. Identification of need is not considered as a design stage in the same sense as the other stages with regard to developing a product. This is due to this stage being considered as the realisation that there is a requirement for a product to perform an application. It is therefore not a design stage with respect to the IIS, but is regarded as a starting point for the purpose of this approach. The remainder of the design stages are considered individually within the IIS approach as they contribute to the final product, forming stages in its development. However the design process is more complicated than just proceeding from one design stage to the next. Knowledge and experience is required about what designs are possible and the procedures and techniques required to arrive at feasible results. Acquisition of knowledge, expertise and information that enables these decisions to be made present the vast majority of problems in developing a design. This knowledge is typically based upon experience, judgement and personal preference of the designer, attributes which are all difficult to quantify.

The design process that the IIS follows is very similar to the majority of traditional design procedures and incorporates their fundamentals. All designs start from an identification of a requirement that progresses into the generation of a basic solution, this in turn develops through greater consideration and modification into a final detailed solution or design of a product, as illustrated in Figure 1.

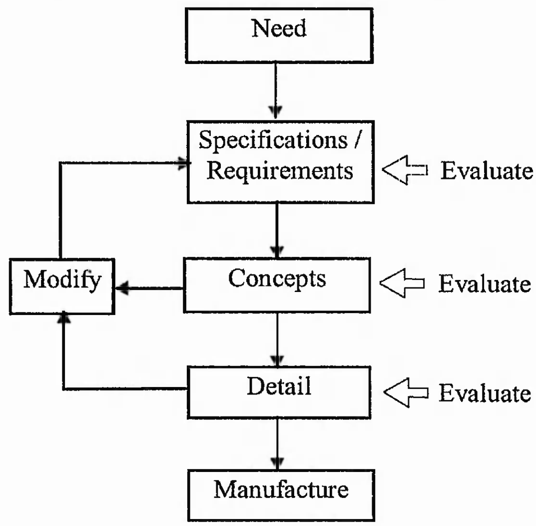


Figure 1: General Design Process

Although this design process follows the traditional basic approach it does not clearly indicate the form of the actual expertise, knowledge and information that is used to transform the initial requirement into the final product design. This consideration, conventionally is assumed to be performed by the designer. However when incorporating the design process into a computer system, opinions, rules of thumb, experience and common sense, which forms a major part of the design process, cannot be assumed to have been incorporated and thus requires greater thought.

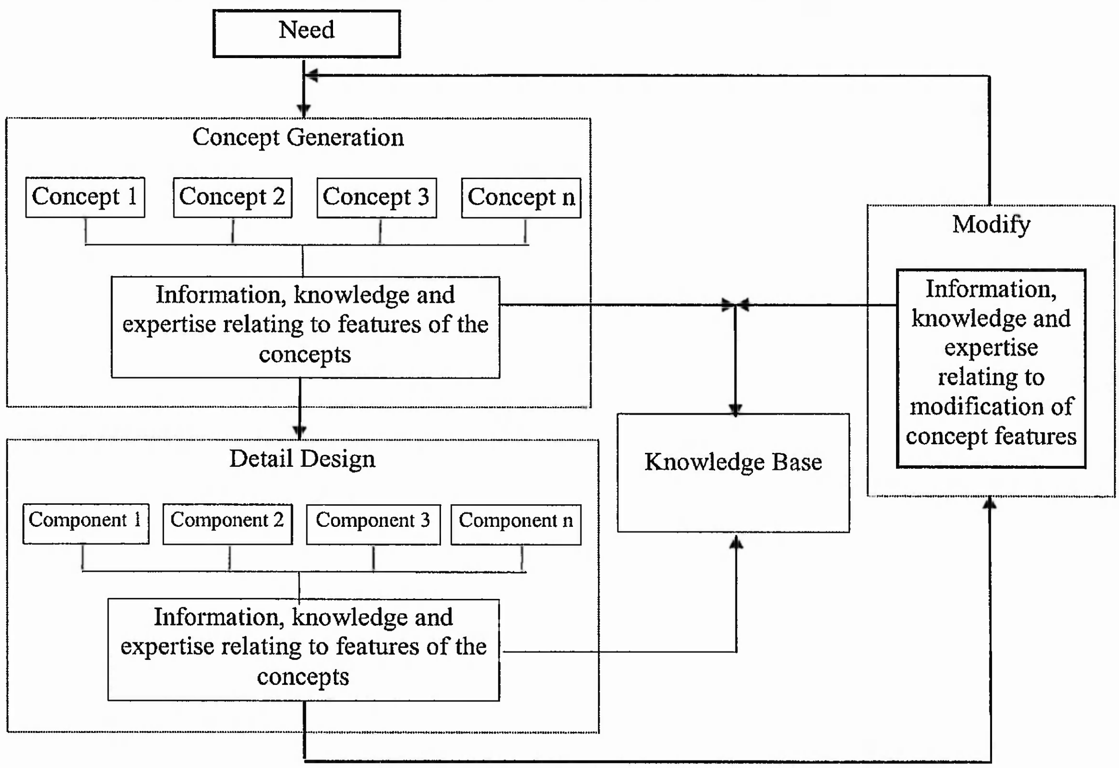


Figure 2: Design Procedure Analysis

The computerised design strategy developed takes into account these attributes and incorporates their effects into the design process. Firstly an analysis of the problem is performed, bearing in mind the needs that initiated the design process, producing the problem statement. This helps to generate general solutions that act as building blocks that combine to form the concept designs for variations of the initial problem statement. Information about the design considerations and limitations regarding the combination of the building blocks is accumulated in preparation for the system development.

The design process is followed through to the detail design and modification stages, where information is accumulated and design avenues investigated. Figure 2 illustrates the general, basic design analysis procedure, which can be applied to the design process of any product.

System Development

During the development of the system, three main areas are progressed through: specification definition, generation of solutions and analysis of solutions. The initial starting position, (the identification and analysis of the general problem to determine the product to be designed) has been performed within the *Design Process* section of the methodology when the design need is identified and problem statement produced.

Specification definition Definition of specifications describing the design are determined on two levels: the *general specifications*, which relate to the requirements of the design and its purpose, and *detailed specifications*, which define actual physical properties. The specifications are determined at two different stages of the design analysis, but can be updated or modified throughout the process. The general specifications are derived directly from the problem statement considering the requirements and qualities that the final product will posses for all eventualities. These are defined at the outset of the design process, describing the purpose of the product from which the concepts are generated. The detail specifications are determined throughout the analysis and detail design derived from the concept, providing specific technical information relating to the individual application of the product.

Generation of Concepts General Concepts are derived from the basic concepts that the final designs are based upon or built up from. These concepts form the building blocks that the design will comprise and should cover all eventualities the system will encounter. However their structure should be limited, providing only the outline. The final conceptual design should consist of a combination of one or more general concepts integrated together.

Analysis of Solutions Analysis of Solutions forms the main area through which the development of the system will pass, involving the analysis of possible solutions in order to identify and establish the knowledge required and the possible variations the development route could take. The structured acquisition of this information is illustrated in Figure 3.

Analysis covers the components and sub-assemblies that comprise a solution, whether for the conceptual or the detailed design stages. Sub-assemblies are in turn broken down to their composite components before detailed analysis is performed. Each component is then analysed individually, indicating the requirements necessary to enable complete design while the various areas of expertise are identified.

Once a design process of the component requiring knowledge has been identified the knowledge relating to it is categorised into one of three forms, well defined, data intensive and ambiguous. Categorisation of the knowledge will determine the means of encapsulation within the system.

- *Well defined* knowledge is normally encoded into production rules as for a known circumstance a particular action or result is given. e.g. If A then B.
- *Data intensive* knowledge can be encapsulated in two forms depending upon the raw data. If the original data is available the most appropriate means of encapsulation is within a

database. However if the original information is not available or incomplete ANNs provide suitable means to handle.

- *Ambiguous* knowledge, which is based upon experience, is encapsulated within production rule form for simple, limited rules of thumb. However if the data is extensive, jumbled, incomplete and requiring judgement based upon similar circumstances the ANNs provide an excellent form of encapsulation. Additionally when neither of these techniques can cope, multimedia is used to help prompt the designer for the appropriate solution.

Once the design route, knowledge, requirements and means of encapsulation are established, the development of the IIS may proceed.

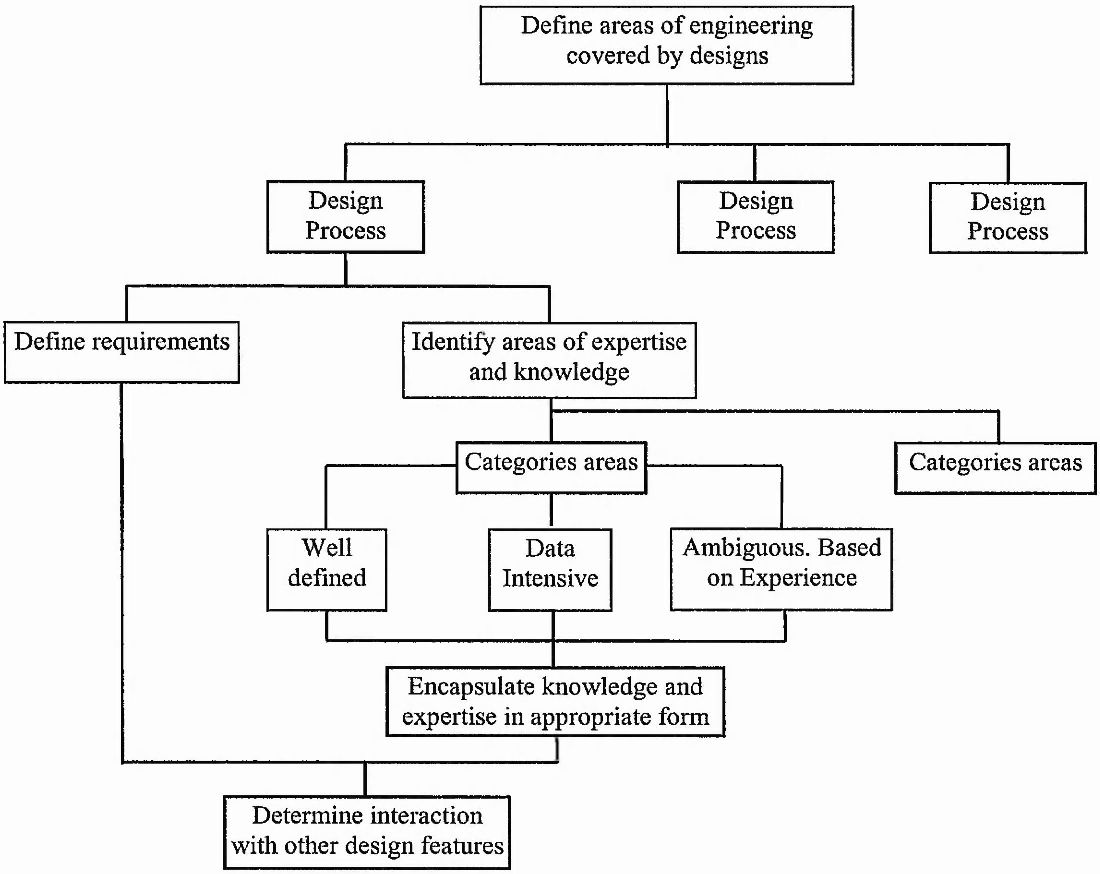


Figure 3: Knowledge Analysis Process

Modular Structure of the Integrated System

The integrated system adopts a modular approach to its structure. A module is a self-contained stage or element of the design process that performs part of the design's development. The modular structure enables areas of knowledge and expertise to be modified, replaced or removed from the system without consulting the developer or requiring intricate knowledge of the purpose and structure of each module within the entire system. Together with the advantage of easy modification the modular approach allows for simpler integration of external stand-alone design packages, which forms the essence of the IIS system.

The modular structure and approach are illustrated in Figure 4. It can be seen that a hierarchy of control develops as the system progresses. This aids the structure of the system and provides levels of responsibility, with the lower levels being controlled and answerable to the upper levels.

The central controller structures the designs progress at the highest basic level, determining which design stage to initiate (either for the first time or for redesign and modification) and to define the final design. The modular approach is maintained throughout the system being implemented to different levels as and when required for each stage.

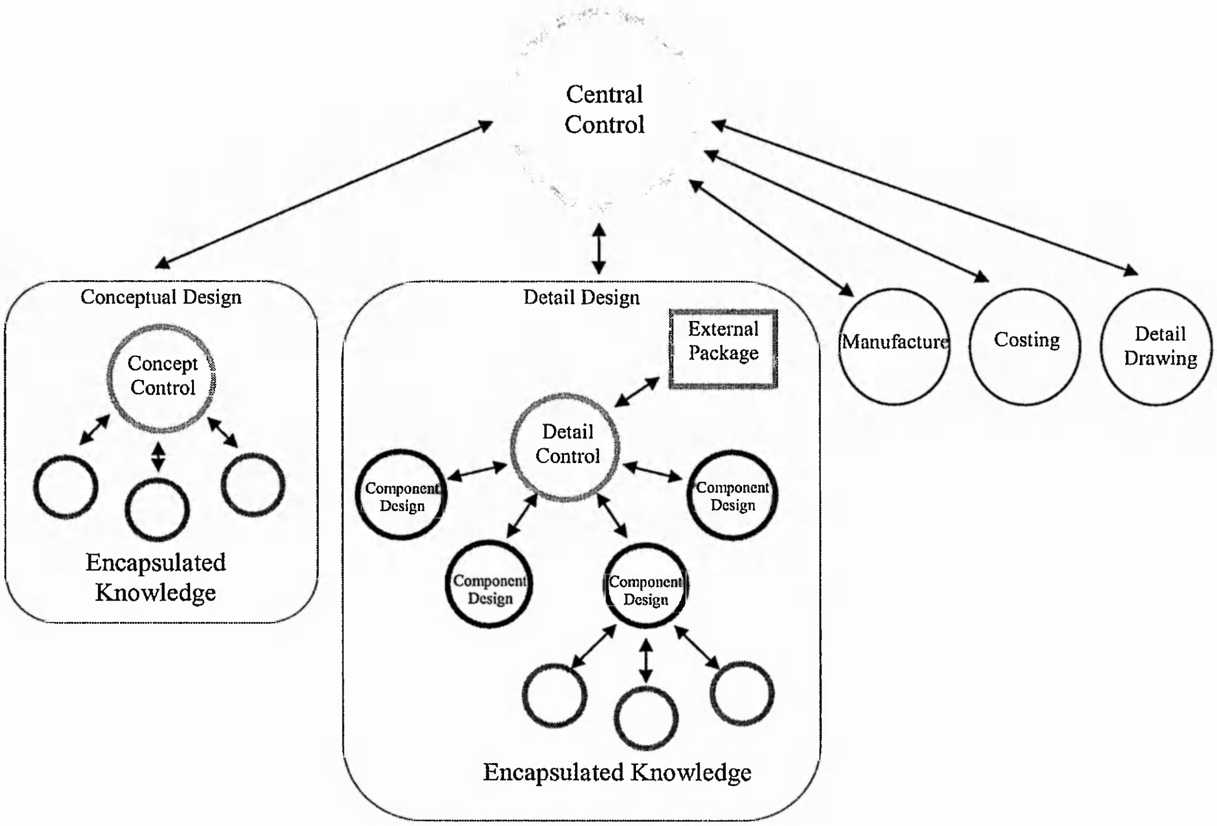


Figure 4: Modular Structure of IIS

The conceptual design stage adopts the modular approach to structure the knowledge and information for different aspects of the concepts. Areas of knowledge relating to different concepts and methods of encapsulation are spilt into separate modules to allow ease of modification and initial system development.

Detailed design exploits the modular approach to the full, forming many sub-designs relating to components within the overall design, each of which encapsulates the knowledge required in separate modules. This stage of the design process is extremely information intensive and represents the area of design where advances in materials, manufacturing processes and analysis techniques will require update and modification if the system is not to become obsolete.

The detailed design is controlled by its own control module, initiating the appropriate design module or external package dependant upon the progress of the design, evaluating results and passing the necessary information back to the central control when detailing is complete or when re-evaluation of the concept is necessary. Each of the design modules performs its task individually, calling upon knowledge modules relating to the task. These modules may take the form of encoded tables, ANNs or production rules.

The final design is achieved by modification and iteration of the design until the design meets the requirements of the problem statement while complying with specifications and limitations.

INCORPORATION OF AI TECHNIQUES

One of the main aims of the IIS is to integrate various AI techniques to encapsulate the decision making process of an expert designer. A combination is used to enable short-fallings of one technique to be supported by the strengths of another.

Artificial Neural Networks

Neural networks form an integral part of the intelligent approach being developed for design, capturing information and knowledge in a medium that produces responses to situations which emulate those of an expert. The ANNs are used to aid the modular structure of the system and enable easy alteration of the information held within them. Alteration of information within the system is achieved by replacing the relevant neural network with on trained to incorporate the new information. Within the IIS ANNs have been used for three applications, concept selection, information storage and manipulation, and component selection. This section will consider the applications that the ANNs are to be set to and determine the appropriate type.

Each network type has its own special qualities, however the Multi-Layer Perceptron trained with the Backpropagation training method (commonly known as the backpropagation network) displays the qualities of a network required by the IIS: classification, structure and prediction. The features of the network that make it suitable of the applications in the system are:

- set structure
- ability to interpolate between trained values
- prediction
- robustness, (its ability to always produce a solution)

The set structure of the network, caused by it being a supervised network, allows the network to integrate effectively within a structured system, where a pre-defined number of inputs and outputs are required. Without the constant structure of the network, modification of information and knowledge contained within them could cause major knock-on effects throughout the IIS.

Interpolation of data enables the system to evaluate situations not experienced before. The network output is based upon a comparison with similar situations, resulting in new solution if necessary.

Prediction is a feature of the network that the system uses extensively with regard to classification and suitability of designs. Its application, often in conjunction with interpolation, is required when a novel solution must be derived.

The network displays robustness to the presentation of incomplete data, always producing an output. This output may not be exact but should be within its desired region, thus demonstrating an advantage over rule based systems

Evolutionary Programming

Evolutionary programming (EP) is an area of AI that is primarily devoted to adaptive searches. Using EP it is possible to search for optimums when the situation prevents other techniques such as gradient decent and direct analytical discovery from being used, due to its ability to simultaneously modify multiple non-related parameters. These situations arise when the contour of the solution for several parameters is discontinuous.

The EP search space refers to the envelope within which the search is confined. This space can become difficult to envisage, as the application of the EP becomes increasingly more complex. The number of dimensions of the search space is dependant upon the number of parameters (one for each), therefore visualization becomes increasingly difficult after three parameters. The value of each parameter corresponds to a position within its dimension, the intersect of which will represent the solution. As the search converges, the intersection points will become increasingly closer together, until theoretically they will be coincident.

EP combines the fundamentals of the natural evolutionary process in the form of a genetic algorithm with structured knowledge relating to a combination of the problems' parameters and method of determining the success. The GA search technique is explained in its simplest form as an adaptive search technique, based upon the Darwinian survival of the fittest principle. The process evaluates the success of a set of possible solutions with regard to the goal of the search. The more successful of the solutions are combined to encourage the search to localise in on the optimum. The process is explained well in Gen and Chang (1997). The GA allows multiple parameters to be modified simultaneously, while the structured knowledge enables these modifications to be restrained within known physical or practical limits.

The EP additionally allows enhancements to be introduced to the optimisation process. These enhancements are in the form of tiring the process Wakelam (1998), or restrictions upon parameters as optimisation proceeds. These features of EP enable the refinement of the search area and prevent the parameters drifting into areas that are known to cause failure of the fitness functions. The application that EP has been implemented to within the IIS stretch the capabilities of the GA to the limits, optimising up to 9 parameters for the increased performance of spur and helical gears, Wakelam et al (1998).

The inclusion of EP within the IIS allows the quality of the final product design to be enhanced, in addition to the GA replacing the decision making and experience of the expert designer.

Production Rule Based Systems

Within the IIS the production rule based system, in its true sense has not been used, but instead structure has been imposed within the system to direct purpose and increase efficiency. The production rules have been used for a duel purpose. Firstly to structure and control the development of a product through the design process and secondly as a source of knowledge retaining and relating to different elements of design.

The knowledge contained within the rules does not form a separate rule base in the conventional expert system sense. Instead the knowledge is *a collection of simple facts together with general rules representing some universe of discourse* in the form of a series of production rules constructing a complex cognitive system.

The use of production rules and a control procedure presents a number of advantages over a conventional expert shell and knowledge base for this application.

- The use of production rules does not require an external shell therefore simplifying and speeding up the system.
- The design process can be expressed in a structured fashion.
- Certainty values and firing methods do not require manipulating.
- Interfacing with the rest of the system is simpler.

The data-driven approach to the formation of the rules has been adopted, structuring the rule around the conditions or situation and not the goal. This is due to the form of the design process adopted for the IIS. As the design progresses information is accumulated, describing the finished product. Thus the conditions are known and the action or solution is required.

The production rules are in two forms. The control rules which are concerned with controlling the progress of the design (if a condition is met perform a task) and information rules which contain specific data or information about a particular design configuration or property. The activation of the second type of rule is governed by the controlling rules.

The control rules hold the knowledge relating to the design's progress. These rules structure the design process and control its development, forming a 'pseudo' inference engine, activating the appropriate information rule depending on the circumstances applied and the stage of the design.

Examples

The control rules not only control the information rules they also control the activation of other modules within the design systems.

IF *previous stages transmission component was a spur gear* THEN *activate ANN that excludes double and single helical gears.*

The information rules contain information in the form of numerical values, design features or simple equation that encapsulate the design information.

IF *the transmission arrangement is 1 stage parallel* THEN *mounting is symmetrical.*

IF *shaft diameter is unkeyed* THEN *shaft design safety factor = 1.5*

The information rules contain specific information relating to an area in the design process and are grouped together in sub sets. This grouping aids the modification and maintenance of the knowledge stored in the rule base.

The use of production rules for both control and knowledge representation forms the production rule based system employed within the IIS. A system of this form can be molded and fashioned to suit the individual applications purpose while maintaining a general structure, which enables a combination of AI techniques to be integrated with one another, forming the hybrid intelligent system and increasing efficiency and direction of design.

CONCLUSION

The intelligent integrated system approach described within this paper draws together the different stages of the design process into a single environment, enhanced by the hybrid application of artificial intelligence. The modular structure to the system described ensures that its update and modification is easily possible together with the advantage of modularising knowledge. This increases the ease of combining multiple AI techniques and the future modification of knowledge, a quality lacking in simple expert systems.

The emphasis on the encapsulation of knowledge within the system allows knowledge and expertise to be made readily available to the user. A highly desirable quality with regard to increased employment mobility. This approach therefore forms a base for automation of the design process.

REFERENCES

French, M. J., 1985, *Conceptual Design for Engineers*, Design Council, London.

Gen, M. and Cheng, R., 1997, *Genetic Algorithms and Engineering Design*, John Wiley & Sons, Inc. New York.

Goldberg, D., 1989, *Genetic Algorithms in Search, Optimisation and Machine Learning*, Addison Wesley.

Kaftanoglu, B., Ulugul, E., Carkoglu, N., 1995, Computer-Aided Optimal Design Strategy of Power Transmission Systems, *Annals of the CIRP*, Vol. 44, No. 1, pp 83-86.

Pugh S, 1990, *Total Design: Integrated Methods for Successful Product Engineering*, Addison-Wesley Publishing Company.

Su, D., 1993, Application of a Knowledge Based, Integrated-system Approach to the Conceptual Design of a Power transmission System, *Quality Through Engineering Design*, W. Kuo eds., Elsevier Science Publishers B. V., pp 417-427.

Wakelam, M., Intelligent Hybrid Approach for Integrated Design, Ph.D. Thesis. (Submitted).

Wakelam, M., Su, D., Jambunathan, K., Henthorn, K. S., Wang, X., 1998, "Genetic Optimisation of Spur and Helical Gears", *Proceedings of Mechanics in Design*, Nottingham England, (Submitted).

GENETIC OPTIMISATION OF SPUR AND HELICAL GEARS

M. Wakelam, D. Su, K. Jambunathan, and K. S. Henthorn
Department of Mechanical and Manufacturing Engineering,
Nottingham Trent University, England.

X. Wang
Department of Mechanical Engineering,
Harbin Institute of Technology, PR China.

ABSTRACT

A novel approach for enhancing gear design is described using genetic algorithms (GA). The critical parameters, which govern the performance of a gear pair, are identified and the extent of their modification is established by the GA. The technical aspects of the GAs structure that determine the characteristics of the search mechanism are described. An alternative cascade application of the GA allowing the search to increase resolution without excessively increasing the search time is presented. Convergence of results for repeatability is demonstrated for the case of a pair of helical gears.

INTRODUCTION

The optimisation of gear design requires the adjustment of several factors that affect the performance of the gear. These include the general dimensions and parameters such as the material property, number of teeth, pitch circle diameter and tooth facewidth together with more detailed modifications such as shifting the tooth profile. Optimisation of the gear performance is achieved by modification of these parameters. However, as the effects of the parameters are both directly and indirectly related, the search area to obtain an optimum is large. Due to a lack of direct relationships between some parameters and the complexity of the search area, conventional search techniques, such as hill climbing and the Newton-Raphson method, would have difficulty in achieving a global optimum. A solution is to apply the adaptive search technique of a *Genetic Algorithm* (GA) to gear design optimisation, enabling multiple parameters to be simultaneously adjusted, homing in on a configuration that produces the best performance.

The optimisation process adjusts parameters that define the characteristics of the gear to fulfil the following criteria:

1. Achieve the minimum facewidth and module while complying with BS 436 part 3 -not exceeding the permissible bending and contact stress on the teeth.
2. Bending stresses within both the pinion and wheel gear will be approximately equal.
3. Contact ratio is to be maximised in order to reduce vibration and noise.
4. Speed ratio is to be maintained.
5. Centre distance of gear pair to be minimised for variable and maintained for fixed centre distances.

These criteria form the basis for the fitness functions which determine the success of the configuration of parameters and therefore the probability of this configuration progressing to the next stage of the search process. The fitness functions are defined in a later section.

Before the GA can be applied to the problem it is advisable to establish the purpose of the various parameters. This enables the fitness functions to be sculpted to suit the application, reducing convergence criteria and ensuring that only critical parameters are encoded into the genome. The critical factors considered are the width, module, pressure angle, helix angle, and number of teeth in the pinion wheel. These parameters have a profound effect on the performance of the gear pair. Additional parameters are included in the optimisation which subtly alter the performance of the gear. These include the addendum coefficient (determines the length of the teeth), the addendum modification coefficients for the pinion and wheel (shifts the profiles of the teeth along the involute, relative to the reference circle) and the rack tip radius (determines the tooth's root radius, a difficult value to determine). Adjustment of these 9 parameters allows the performance of the gear pair to be fine tuned to suit the desired goals.

APPLICATION OF THE GENETIC ALGORITHM

The GA optimisation process is applied in a cascade fashion, along a similar principle to that discussed by Patnaik et al. (1997). They discussed a project undertaken at the NASA Lewis Research Centre where a number of optimisation techniques were combined in series. Each technique in the series used the results from the previous as the starting positions. This procedure was concluded to be generally more successful and robust than any single technique tested when presented with a variety of problems.

The cascade procedure has been adapted and applied in this work. Using initial values generated from a basic gear design package as initial starting positions, base limiting conditions for the parameters are established. These initial values provide 'warm' starting values within the region of a solution, as opposed to totally random, 'cold' values that increase the search area dramatically to the point where the solution may not comply with practical or physical laws. The first tier of the optimisation is invoked using the 'warm' values to adjust the parameters in search of an global optimum, searching a broad range of values. The optimisation process continues until a limiting percentage of the genome population are identical. This limit has been determined by trial and error and taken as 70%. At this point the information encoded within the converged genome is decoded forming the solution to this tier and the intermediate 'warm' starting values for the next. The GA optimiser is initiated again with the solution from the initial optimiser, applying a narrower, more accurate band to the search. The increase in resolution localises the search in the region of the global maximum, enabling the accuracy of optimisation to be increased. Again the search is repeated until the limiting percentage of the population are identical, at which point the converged genome is decoded to form the final solution. Figure 1 illustrates the tiered procedure.

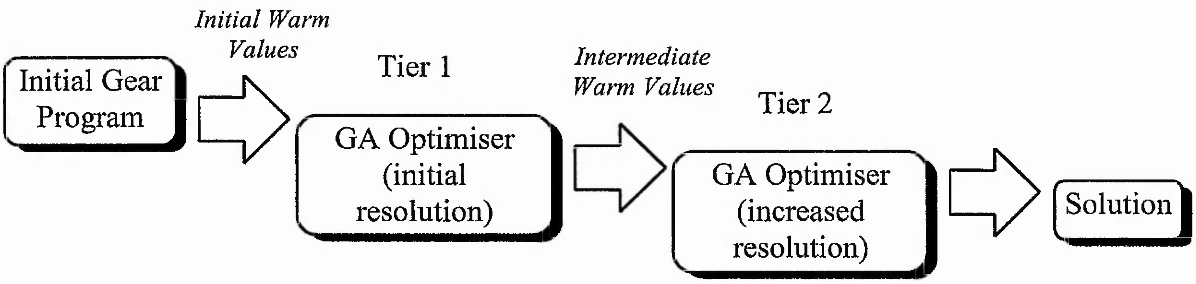


Figure 1: Cascade Procedure

Genome Encoding.

The method of encoding the information within the gene has a dramatic effect upon the performance of the optimisation process. Three methods of encoding are used, applicable to different circumstances: direct, percentage deviation of the value and position within a pre-defined list.

Direct. The direct method is the most straightforward way of encoding the value within a gene. The value is encoded directly into the gene either in its entirety or as a proportional value that is manipulated after decoding. For example, from a range of 0 to 31, dividing by 40 alters the range giving 0 to 0.775. If 0.8 is then added to the value the range is moved to between 0.8 and 1.575.

Percentage Deviation This method of representing the encoded value within each gene requires two stages of decoding to extract the information. This enables the value to be stored in a condensed fashion and is ideally suited for use with initial starting points or values. This method is an alternative to storing the entire value directly within the gene in binary form, as only a percentage deviation from an initial starting point is stored, thus enabling the size of the gene to be considerably reduced for large values. The principle of the percentage deviation method is as follows. The initial encoded values within the population cover a band of possible solutions about either side of an initial start-point. This is illustrated in Figures 2 and 3 below.

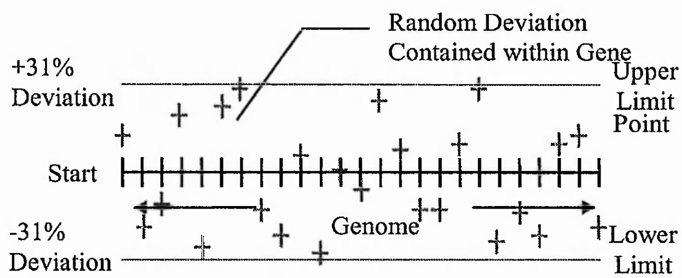


Figure 2: Deviation About Initial Start Point

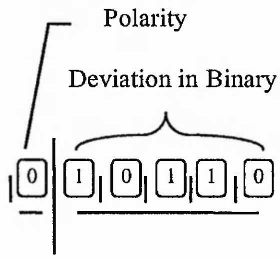


Figure 3: Structure of Gene

The encoded deviations are contained within binary strings. For example a five or seven bit length gives a range of 0 to 31 or 0 to 127 respectively. The range is then effectively doubled by the addition of an extra bit to indicate the polarity of the deviation, + or -. Decoding the gene and applying the percentage deviation to the start-point value produces the new value corresponding to the search. The upper and lower limits represent the bounds of the search area in which the parameters are retained.

Pre-defined List The third method of encoding information is by referring to a list of pre-defined values such as standard gear modules. A pointer is moved up or down the list from a reference point to identify the position that contains the information. The position of the pointer relative to the start-point is determined by the information contained within the gene. The decoded value from the gene relates to the number of positions within the list that the pointer is moved. Information is encoded within the gene using the same technique as for percentage deviation. For this method it is advised that the size of the gene be reduced. For example, instead of a five bit string giving a range of 0 to 31, a three bit string is used to give a 0 to 7 range. The smaller range enables the movement within the list to be restricted. Restriction of movement is often required due to the nature of lists of information which can change dramatically within the movement of a few places. Also the lists size may be relatively small in comparison to the range. Reducing movement within the list will thus reduce the necessity to truncate movement, ensuring the position is within its bounds. Additionally, this reduction helps restrict the size of the genes and therefore aids convergence by a reduction of the search area. The encoded value directly represents the number of spaces shifted up (+) or down (-) the list. The process is illustrated in Figure 4 below showing how a –5 deviation from the gene relates to the movement of the pointer relative to the start-point within the predefined list. The full extent of the deviation range and the list are also given in the figure, illustrating that the movement of the pointer is restricted within the bounds of the list.

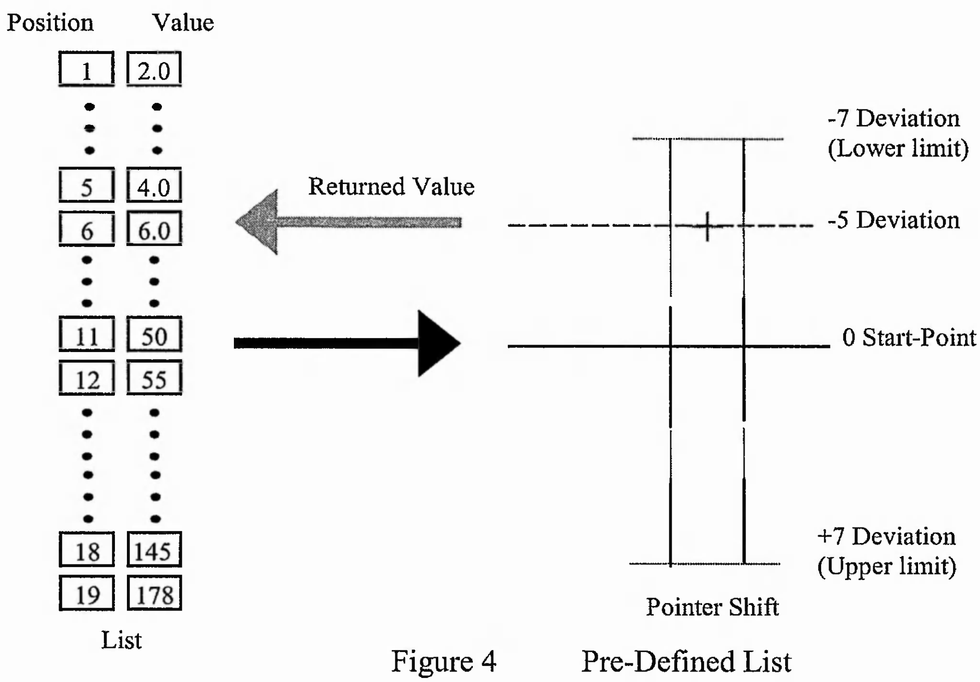


Figure 4 Pre-Defined List

Limits are imposed on the generated parameters to ensure that the optimisation process does not cause the values to drift into regions where the result would be invalid. These limits are maintained and implemented by production rules at the pre-process stage of the GA. The method of limiting the parameter values used is as follows. The value contained within a gene is first decoded then pre-processed to produce the value of the parameter required by the analysis stage of the GA. If this value exceeds the limits imposed the value is set to its relative limit. The action of constraining the value should not affect the GA process with regard to regeneration fitness levels, due to the method of encoding. As the encoded information is the deviation and the optimised value is the decoded resultant, the limit will produce the desired output provided the fitness for the limit is high. This characteristic of the gene that caused the high fitness will then be passed to the next generation.

Although the limiting process will not degrade the capability of the GA for producing an optimum solution it will retard convergence of the population. Convergence of the population towards a single solution can be retarded if different gene configurations produce the same value. This can be the case when modifying a decoded value without reference to the gene. This is due to a potentially high fitness rating being awarded to a gene that should be modified or removed from the population. Modification of the gene would be a complex process requiring detailed knowledge about the problem and the decoded parameter value. Therefore the use of a penalty system is used to penalise genes and in turn the genomes that require limitation.

Genome Construction.

The parameters that correspond to the genes of the genome have been identified earlier. These genes combine to form either a seven or nine gene genome which are represented in Figures 5 and 6 below.

Gene Number

1	*2*	*3*	*4*	*5*	*6*	*7*
(6 bit)	*(6 bit)*	*(5 bit)*	*(2 bit)*	*(6 bit)*	*(3 bit)*	*(8 bit)*
width	module	addendum modification	pressure angle	helix angle	rack tip radius	profile shift x1

Figure 5. Genome Composition for Fixed centre Distance

Gene Number

1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
(6 bit)	*(6 bit)*	*(5 bit)*	*(2 bit)*	*(6 bit)*	*(3 bit)*	*(8 bit)*	*(8 bit)*	*(3 bit)*
width	module	addendum modification	pressure angle	helix angle	rack tip radius	profile shift x1	profile shift x2	pinion teeth z1

Figure 6. Genome Composition for Variable Centre Distance

These parameters have the effect of altering the geometry, performance and contact ratio. It is the purpose of the GA to find the combination of parameters that will produce the optimum design. However, some consideration is required with regard to the method of encoding the relevant information into the genome.

Two configurations of genome have been constructed for the two different types of gear design; fixed centre distance and variable centre distance. Due to the method of calculation for, the optimisation of the fixed centre distance, the number of teeth for the pinion, $z1$ and wheel, $z2$ are determined by the value of the module and the speed ratio (speed ratio remaining constant). The difference in centre distance is compensated for by the profile shift of the teeth. For the profile shifts to be determined the amount that one of the profiles will be shifted must be known. The value of the pinion profile shift, $x1$, is included within the genome allowing the value of the wheels shift, $x2$, to be calculated.

For the variable centre distance, the relation between the pinion and the wheel is limited to the speed ratio. Due to the lack of relationships, the number of teeth and amount of profile shift on the teeth cannot be calculated from the existing parameters, therefore these parameters are included within the genome, forming additional dimensions to the search.

The genome has been constructed so that both fixed and variable centre distance gear designs can use the same genome. Parameters that are used within both designs form the first section of the genome, while the two extra genes, (x_2 and z_1), required for the variable centre distance are located at the end. This structure allows the additional parameters to be ignored when not required. Their permanent inclusion within the genome will not affect the GA process as they are ignored and during crossover the positions of the transferred bits are maintained, therefore never letting the contents of these genes affect the search.

The method of encoding the information within the genes is dependant upon the effect that the parameters has upon the design. These considerations include limits, set values and resolution. Limits have been set on several parameters to prevent them from producing values that will result in certain failure of the design or undesirable values. Table 1 contains the limited parameters and their restrictions.

Table 1: Parameter Limits.

Parameter	**Upper Limit**	**Lower Limit**
Module	Position 34 within list	Position 0 within list
Addendum Modification Coefficient (x1 and x2)	1.0	$x_{min} = \frac{\text{No. teeth}}{\text{Min. No. teeth.}}$
Helix Angle (angle limited)	User defined (typically<45°)	0
Helix Angle (force limited)	√2.Tangential force	0

FITNESS FUNCTIONS

The fitness function, as mentioned earlier, has a dramatic effect upon the convergence of the search and the parameters contained within genomes that are transferred through the generations. Therefore the selection of criteria that the fitness function comprise of must mirror the desired characteristics of the target design. Due to the nature of design two categories of function have been applied. These categories have been termed *fitness rating criteria* and *fitness conditional criteria* and apply either a gradient or step function to determine the level of fitness.

The gradient functions are used by the fitness rating criteria that determine the fitness of the non-critical design characteristics. These characteristics do not directly cause failure of the design, but do influence its performance and guide towards optimisation targets to be achieved. The range given for each rating criteria is from 0 to a maximum value, the greater the fitness, the higher the value. Figure 7 and Table 2 below illustrates the principle.

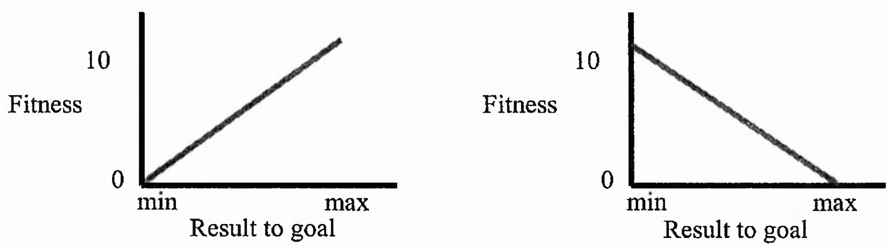


Figure 7: Gradient Fitness Functions

Table 2: Fitness Rating Criteria

Fitness Rating Criteria	Max. Fitness (max. Value)	Min. Fitness (0)
Result to goal 1	Smallest Value	Largest Value
Result to goal 2	Largest Value	Smallest Value

The second category of fitness functions relates to critical characteristics of the design that directly influence or cause failure of the design. These fitness functions form *conditional criteria* and apply a step function to the level of fitness. As failure to meet the requirements of the function will result in failure of the design, these fitness functions have the effect of causing the overall fitness value of the genome to be drastically reduced thus encouraging removal from the population. Figure 8 and Table 3 illustrate the principle.

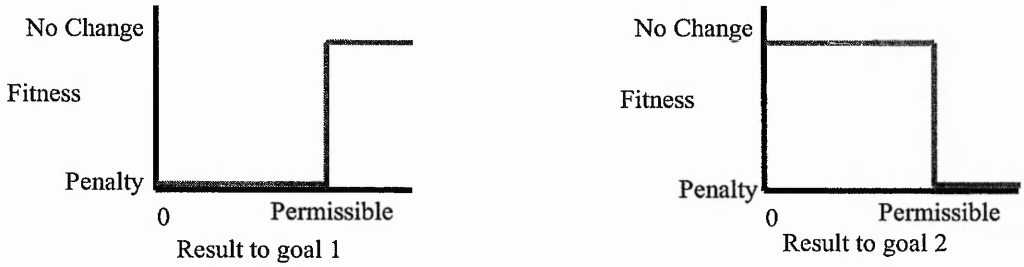


Figure 8: Step Fitness Functions

Table 3: Fitness Conditions

Fitness Conditional Criteria	Pass	Penalty
Result to goal 1	>= Permissible Value	< Permissible Value
Result to goal 2	<= Permissible Stress	> Permissible Stress

Penalties

The penalties that are awarded to the genome are the means by which the probability of the genome reproducing is reduced. Penalties are awarded for two reasons: parameter values when decoded that are outside their limits or if the design has failed to meet one or more of the critical *fitness conditional criteria*.

Penalties are applied to the overall fitness of the genome instead of immediately ejecting the genome from the population. Immediate ejection removes the complete genome from the population. Therefore possible fit characteristics of the genome can be permanently lost from the search. The use of penalties therefore allows the genome remain in the population for crossover and transfer its genes to the next generation, but with less probability of success. If unfit genes are transferred to the next generation, penalties will be awarded again and the probability of the genome to reproduction again reduced. The probability of the gene transferring through to a third generation is low and decreases as the generations increase, due to fitter genomes being encouraged to reproduce.

The method of applying the penalty to the genome is to scale the overall fitness. Summing the fitness values for each fitness criterion forms the overall fitness of the genome. The penalty is applied through a function that scales the fitness proportional to the number of penalties that have been awarded to the genes that the genome comprise. The value of the penalty increases as limits are imposed or conditional criteria are not met. Therefore the greater the penalty the greater the reduction in fitness.

Awarding penalties instead of ejecting the genomes that cause failure of design allows genomes that are just outside the feasible boundary to pass on fit genes to the next generation and very unfit genes to be ejected rapidly from the population. Figure 9 illustrates the feasible region of the search and the effect of the penalty function. The position of the single penalty genome is just outside the feasible region and therefore probably contains only genes that are fit and could aid the search. However the 3 penalty genome is well beyond the feasible region and therefore should be ejected from the population.

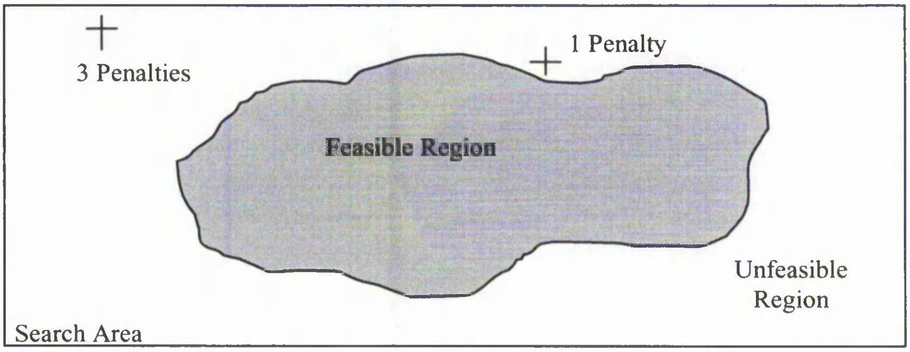


Figure 9: Effect of Fitness Penalties

Fitness Scaling

The fitness values applied to the roulette wheel often require modification, due to the possible close proximity of some fitness values. This close proximity of genome fitness increases as the population begins to converge causing only small deviations in the fitness. Therefore the fitness values for two different genes will cause little difference to the size of the respective roulette wheel segments. To overcome this problem scaling methods are applied to the fitness values to emphasise the relative fitness of the genome within the population.

Several methods of scaling exist, using the sorted fitness order of the genomes and the actual fitness value. Several of the methods are described by Gen and Chang (1997), including the linear ranking function which is the base of the function used in this work. The function combines the actual fitness value together with an element relating to its ranked position. The fitness is given by:

$$p_k = q_k + (k - popsise)\left(\frac{q}{1 - popsize}\right)$$

where : p_k = scaled fitness; q = fitness of highest ranked genome
k = ranked position; q_k = fitness direct from fitness function
$popsize$ = population size

This function is applied in conjunction with the deviation in fitness, thus providing greater resolution to the selection roulette wheel. The effect of this is to define a noticeable ranking order without encouraging super convergence, caused by excessively biased scaling of fit genomes.

GEAR OPTIMISATION

The optimisation process is evaluated by performing numerous tests to check the controlling factors of the GA, the repeatability and robustness of the process and the accuracy of the results. These include tests relating to the GA controlling factors: size of the population, the crossover and mutation levels and convergence criteria.

The optimisation has been set to achieve three goals which form the rating criteria to guide the search. These are in addition to the conditional criteria that will ensure failure of the design will not occur. The rating criteria cover:

1. Reduction of the gear pair size, (both centre distance and facewidth.)
2. Increase contact ratio
3. Equalise bending stresses within the teeth of the pinion and wheel

The case study given below is for a helical gear pair that is required to transfer 40 kW of power at 1450 rpm. The application of the gear is for moderate shock conditions which is represented by an application factor of 1.4. The results given below compare the performance of the initial gear design with that produced from the GAs optimisation.

Table 4: Comparison of Optimised Gear Pair Geometry with Initial Design

Geometry	Initial	Optimised	Difference
Module, (mm)	2.00	2.00	0
Alpha, (Deg.)	20.00	20.00	0
Beta, (Deg)	15.00	21.00	+6
Pinion Teeth	24	23	-1
Facewidth	40.00	35.78	-10.56%
Pinion Tooth Depth	4.5	5.3	+17.78%
Wheel Tooth Depth	4.5	5.3	+17.78%
Pinion Rack Tip Radius, (mm)	0.50	0.4	-0.1
Wheel Rack Tip Radius, (mm)	0.50	0.4	-0.1
x1	0.00	0.00	-
x2	0.00	0.30	-
Centre Distance (mm)	124.23	123.82	-0.33%

The geometric goals of the search have been achieved. The facewidth of the gear and the centre distance have been reduced by 10.56 % and 0.33 % respectively. The reduction in centre distance is not very dramatic, but as its increase would aid the other criteria in achieving their goals its reduction is a sign that the GA process is working.

One of the main targets of the optimisation was to increase the contact ratio of the gear pair. Its purpose is to reduce vibration and thus noise, fatigue and shock loads. This has been achieved, improving on an initial design that already has a high contact ratio. This can be seen from the results in Table 5, below.

As can be seen from the stresses acting within the gear teeth the improvements to the design have been achieved without reducing the margin of material safety. Yet the lengthening of the teeth by 17.78 %, (as shown by the increase in tooth depth) should increase the bending stress. In addition the bending stresses acting on the teeth are within 5.33 % of each other, (an improvement of 1.34 %) thus promoting equal wear and fatigue of the teeth. The GA has therefore modified several parameters to ensure that the design will not fail. The removal of failing designs from the search has thus encouraged stronger designs.

The optimisation process was repeated three times to ensure that the final solutions were the same and thus the process displayed repeatability.

Table 5: Comparison of Optimised Gear Pair Performance with Initial Design

Performance	Initial	Optimised	Difference
Contact Stress Pinion (% below permissible)	5.10	12.22	+7.12
Contact Stress Wheel (% below permissible)	10.79	17.48	+6.69
Bending Stress Pinion (% below permissible)	10.34	29.68	+19.34
Bending Stress Wheel (% below permissible)	16.84	27.1	+10.26
Equal Bending Stresses (% difference)	6.67	5.33	-1.34
Contact Ratio	3.29	3.95	+20.06

CONCLUSIONS

Presently there are several methods for the optimisation of gears including the use of GAs (Abersek et al., 1996). Abersek's application optimises some of the parameters that this application addresses, as do the other gear optimisation techniques. However they do not consider modification of the tooth's profile and encourage the increase of the contact ratio for reduced vibration and noise as this application does.

In addition to parameter adjustment, when applicable, the selection of the type and magnitude of profile shift is determined. According to Niemann (1978), the magnitudes of the profile shift should be biased towards the pinion wheel being positive in order to avoid undercutting. From the results it is shown that the GA has obeyed this rule, indicating that the fitness criteria do correctly define the desired result. This may be due to the limiting factors of permissible and actual stress, which form one of the main criteria from which the fitness of the design is determined. The GA optimisation therefore enables designs to be improved using the desired end result as the target. Thus intricate knowledge about the effects of individual parameters on either the final design or each other is not required.

REFERENCES

Abersek B., Flasker J. and Balic J., 1996, Expert System for Designing and Manufacturing of a Gear Box, *Expert Systems With Applications,* **11**(3), pp. 397-405.

Gen, M. and Cheng, R., 1997, *Genetic Algorithms and Engineering Design*, John Wiley & Sons, Inc. New York.

Niemann, G., 1978, *Machine Elements: design calculation in mechanical engineering, Vol. 2: Gears*, Springer-Verlag, New York.

Patnaik. S. N., Coroneos. R. M. and Hopkins. D. A., 1997, A Cascade Optimisation Strategy for Solution of Difficult Design Problems, *Int. J. Numerical Methods in Engineering*, **40**, pp. 2257-226

Evolutionary Aid for Training Backpropagation Neural Networks

M. Wakelam, D. Su*, K. Jambunathan and K. S. Henthorn
Department of Mechanical and Manufacturing Engineering,
The Nottingham Trent University, Burton Street, Nottingham, England. NG1 4BU

* corresponding author, Tel. (01159) 418418 Ext. 2306,
Fax (01159) 9486506, e-mail: daizhong.su@ntu.ac.uk

Abstract

The backpropagation network is a popular and versatile artificial neural network used for a multitude of applications, from prediction to behaviour modelling. Training these networks has traditionally been a tedious process requiring multiple attempts and a trial and error approach to improve performance. This paper describes an automated method of achieving the desired performance from a backpropagation network by the integration of a genetic algorithm into the training process. The method determines the structure and training period of the network, thus reducing the training time and the level of artificial neural network experience required.

Keywords

Backpropagation, artificial neural networks, genetic algorithms

1. Introduction

Training a backpropagation (BP) artificial neural network (ANN) requires not only expertise in the training process, but also a knowledge of the factors that affect the performance of the trained network. Primarily these factors include the training period, training data, network topology and the transfer function of the process elements. The settings of these factors affects the ability of the network to accurately simulate the problem it is attempting to model. Although several previous works have described methods for setting the values of these factors, for example Amirikian and Nishimura (1994) and Wang et al (1994), these methods have been developed for a limited range of applications. As the setup for each BP network is problem dependant these methods can only provide a guide to training, still requiring a trail and error approach to improve performance. Therefore a means of removing the expert element from the training process that is application independent will aid the development of BP networks, increasing their availability to more applications.

A training aid has been developed for BP networks requiring only the training data to describe the application. The training aid utilises an optimisation search technique to control the training process, removing the tedious manual, trial and error process. A number of search techniques were considered, such as hill climbing and the Newton Raphson method, but due to the magnitude of the search area and lack of direct relationships between the training factors a continuous search was concluded to be too time consuming. An adaptive search technique provides a solution that is capable of covering the search space without systematically considering every point, using only the resultant performance of the ANN to guide its progress. A genetic algorithm (GA) based approach has been developed which implements the optimisation capabilities of a

GENetic algorithm to define the factors that determine a BP NEUral networks performance (GEN-NEU).

Researchers have previously made efforts to apply GAs to the optimisation of BP networks. Caudell and Dolan (1989) used them to adjust the connective weights during the training process to improve the networks convergence, while Miller et al (1991) and Maniezzo (1994), have applied them to the optimisation of the network topologies. However adjustment of the connective weights has not provided significant improvements in network performance over the BP technique and therefore has not been used. Optimisation of the network topology on the other hand has produced more encouraging results, although only this training factor affecting the networks performance has been addressed.

The GEN-NEU approach takes into account three of the major factors that affect network performance: topology, transfer function and training period. Optimisation of these factors is performed simultaneously, considering their combined effects upon performance and convergence, making the optimisation more effective. However the training data is not included as a factor in the adaptive search. Although it has a profound influence over the training process and network performance the diversity of its form and nature prevents it from practically being encoded within the GA. The training data is therefore generated by the ANN developer and should comprehensively describe the task the ANN is being set.

2. Adaptive Optimisation Process of the GEN-NEU Approach

The basic principle of the GEN-NEU approach is to adjust the factors required by the BP training technique based upon the performance (fitness) of the network being trained. Gene sequences which produce high fitness levels combine with other fit sequences to form an optimum. Figure 1 demonstrates the process.

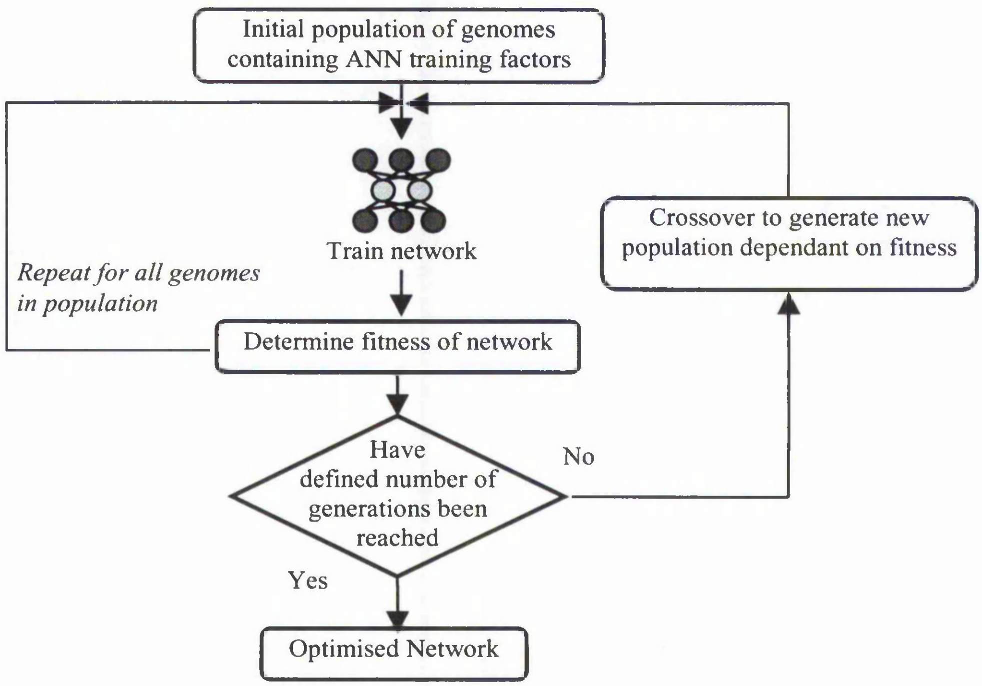


Figure 1. Basic GEN-NEU Process

The approach encodes the values for the factors that influence the performance of the network, including the transfer function, topology and training period into separate genes collectively forming a genome (a string of genes). Upon initiation the values contained within the genes are randomly set from values within the search space. The networks corresponding to the information contained within all the genomes of the population are trained and tested then sorted into the order of descending fitness. The fitness value of the genome is determined from the network output to a series of test cases applied after training. The lower the average error between the target and the network output the fitter the genome, which in turn determines the genomes probability of reproduction. Each genomes fitness rating relative, to the rest of the population, is proportional to its probability of reproduction. Thus fit genomes relative to the population are given a greater chance of transferring their genes to the next generation, while unfit genomes are gradually removed. The fitness value is therefore taken as the reciprocal of the network error, thus ensuring the fitter genomes get a greater probability of reproduction.

Once the fitness ratings of the population have been determined the genomes reproduce to form the majority of the next generation. The reproduction process, termed crossover, splices two parent genomes to produce a single offspring. The splicing process takes randomly sized segments of one parent and combines it with another segment taken from the other parent, as illustrated in Figure 2.This process is repeated until the child genome is of the same length as the parent. Crossover accounts for approximately 98% of the next generation. The remainder of the next generation comprises of a randomly selection from the current population which pass unaltered through to the next generation.

0 1 1 0 0 1 0 1 0 1 1 1 0 1 0 1 0 0 1 0
0 1 1 0 1 1 0 1 0 1 0 0 1 0 1 0 1 0 0 1

0 1 1 0 1 1 0 1 0 1 1 1 0 0 0 1 0 0 1 1

Figure 2. Multi-Point, Random Crossover

Once the next generation is established a mutation operator is applied to the population with a probability of 5%. The mutation operator applied alters the encoded information contained within the complete genome as opposed to the typical bit mutation operator, Gen and Cheng (1997). Total mutation produces a new member to the population allowing the search to rapidly ingress into new areas of the search space without extending the population size, enabling the total area covered by the search to adaptively increase as the process progresses, Wakelam (1998). Mutation also helps prevent localised minimums limiting the scope of the search. As the affects of the mutation can be dramatic upon the search the probability of mutation is set low to limit disruption once convergence commences upon a solution.

Once crossover and mutation have been performed the new generation is complete and ready for the new fitness ratings to be determined. After the final generation is complete the network corresponding to the fittest genome becomes the resultant network from the GEN-NEU training aid, ready for application. The topology, transfer function and connective weights are recorded.

The number of generations that the search process performs is set to a finite amount instead of using the convergence of the population upon a single solution. This is due to the effect of the random initialisation of network connections at the beginning of training, which can lead to variations in performance that can disrupt convergence. Therefore the fixed length search prevents excessive computational expense for a small increase in network performance.

3. Encoding Information for the Genetic Algorithm

The information relating to the factors that affect the performance of a network during training are encoded into the genes within the genome in binary form. An example is shown in Figure 3. The decoded values correspond to a sigmoid transfer function, 19 elements in the first hidden layer and 9 in the second and a training factor of 5.

Binary coding has been used due to advantages over others, such as Gray coding for two reasons. Firstly, the type of coding will have little effect on the performance of the GA, because the genome comprises multiple genes whose information do not relate directly to one another. Secondly crossover is performed at random points during reproduction allowing genes to be spliced, causing a significant change in the encoded value.

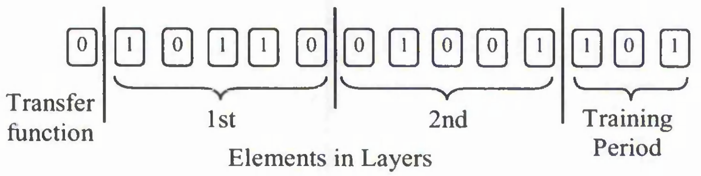


Figure 3. Combination of Genes to Form Genome

The length of the training period for different applications has a direct influence on the performance of the network. Therefore integration of the training period into the GA optimisation approach will aid successful training and reduce user interaction with the training of the network. However the encoding of the length of the training period into the genome required important consideration.

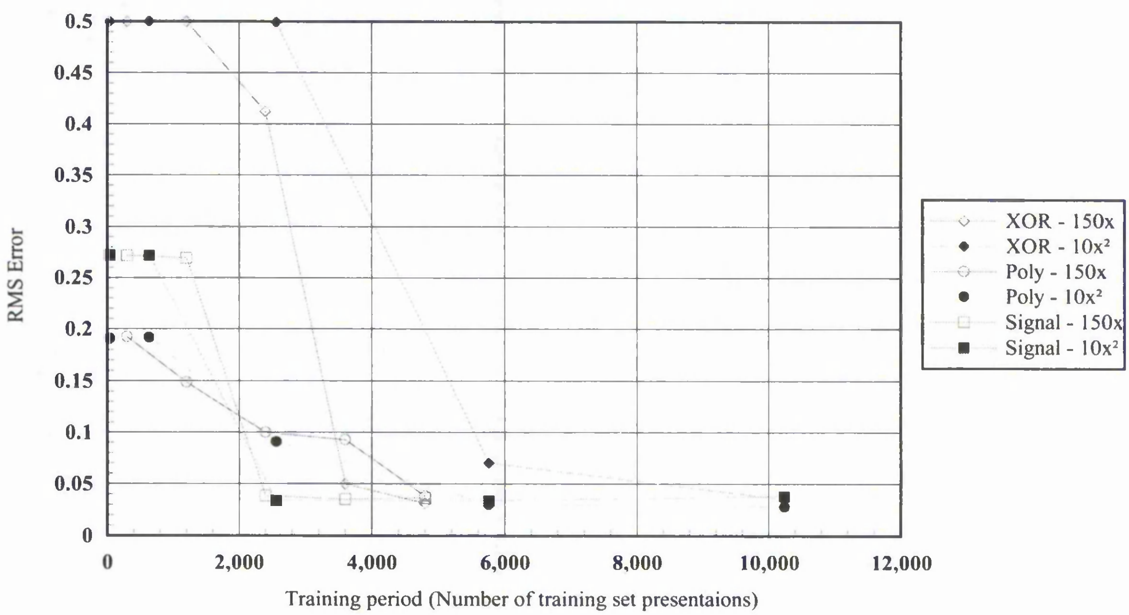


Figure 4. Effects of Training Period

The training period has been encoded in binary form as for the topology, from 0 to 31. The value from the gene is pre-processed before training to obtain a realistic value. Figure. 4 shows the effect that the training period has upon the performance of a network and two per-process functions, $150x$ and $10x^2$, using the range of values available from the genome.

The effect of the training period and the pre-process encoding functions were tested for the three test cases used later in this paper, XOR, a polynomial (poly) and a feedback control loop (signal). The pre-process encoding functions obtain similar results, where x is the value encoded in the gene (0-31). However these results are form training of relatively simple problems, however these functions provided training periods in excess of the levels require. This is illustrated by the leveling of the network error. More complex problems may require extensive training presentations, which the $10x^2$ function can offer, therefor this function has been integrated into the GEN-NEU program.

4. Validation of GEN-NEU Approach

To ensure that the GEN-NEU approach and program performs the optimisation process correctly and aids training, a series of validation tests have been performed, checking the BP technique, the GA process and the network performance achievable. These tests cover a variety of different applications including: classification, mathematical modelling and signal processing.

4.1. Validation of Backpropagation Network Performance

Validation of the network consisted of the exclusive OR (XOR) benchmark test. Classification relates to many applications where information appears to be in a random fashion, displaying no apparent order. The purpose of the network is thus to sort the information into classes or groups. The XOR problem has been presented to the network to prove that the network can classify data requiring more than one hyperplane.

The GEN-NEU program successfully trained a network to model the XOR problem, returning the network weights and structure information necessary to duplicate the network. The resultant network comprised of two inputs, a single hidden layer containing two processing elements (PEs) and a single output PE, (2-2-1), trained for 5000 presentations of the training data using the sigmoid transfer function. The trained network achieved an RMS error of 0.021 from the target values which after rounding to the nearest integer was removed.

To ensure that the training process was performed correctly a network was trained under the same conditions as those determined by the GEN-NEU program using the commercial ANN development tool, Professional II. The resultant RMS error was 0.024, thus indicating that the training process had no errors. The small variation in results is accredited to the random initialisation of the connection weights.

4.2. Validation of GA Approach

Validation of the GA approach consists of running the GEN-NEU program to optimise the network topology and transfer function when presented with the XOR problem. The topologies and transfer functions contained within the genomes of each generation are represented in Figures 5 and 6. These figures give different perspectives of the optimisation process. Figure 5 illustrates the topology of the ANNs, while Figure 6 illustrate the transfer function of the PEs (where 1 = sigmoid and 0 = *tanh*).

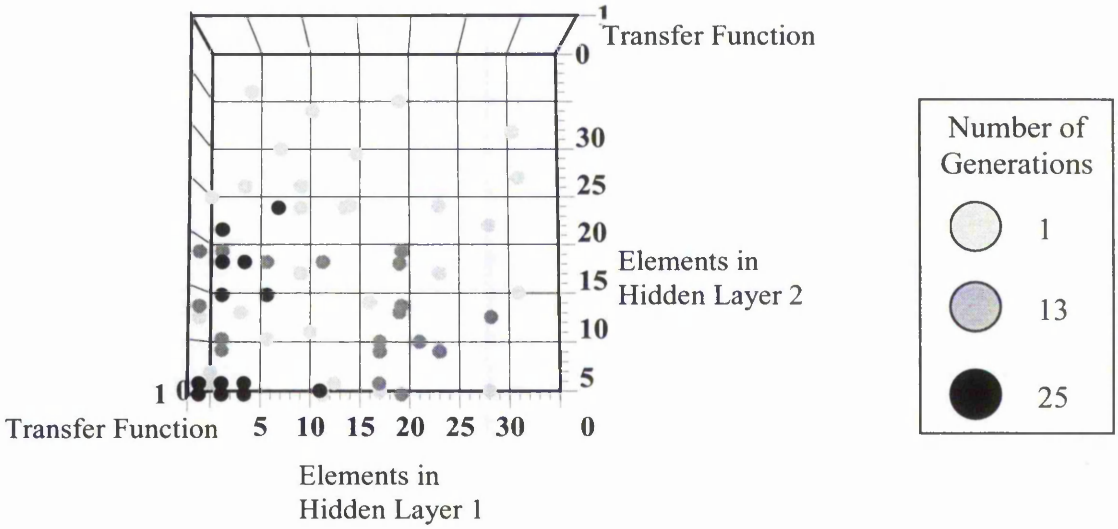


Figure 5. Convergence of Elements in Layers

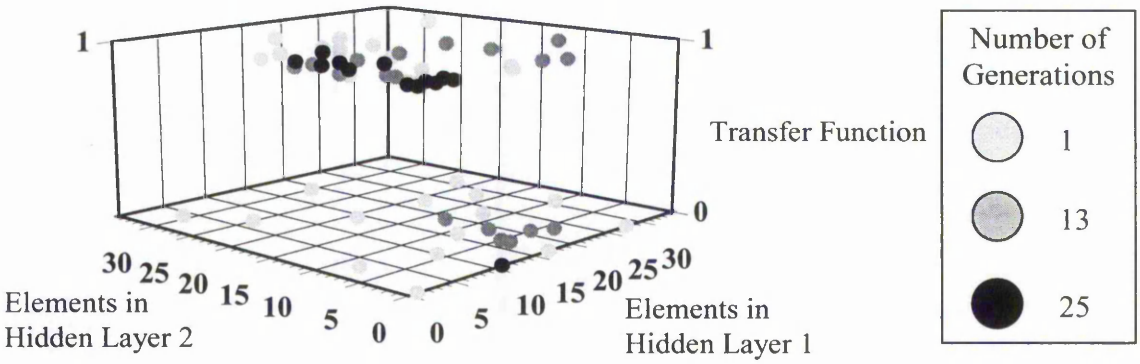


Figure 6. Convergence of Transfer Function

It can be seen from the localising of shaded points in Figures 5 and 6 that as the generations increase the search begins to converge towards the optimum network. The initial random scatter, shown by the light gray points, migrates from a low density concentration to a higher density at the end of the process. As the populations converge, multiple genomes generate the same network configuration and thus form coincident points on the graphs. This accounts for the apparent reduction in points as the generations increase. These figures demonstrate the GAs ability to converge upon a solution.

4.3. Performance of GEN-NEU Approach

The performance of the GEN-NEU approach has been tested using two ANN applications, modelling a mathematical equation and signal processing. These applications require different qualities of the ANN, therefore providing a means of establishing application independence.

Equation Modelling

The ability of the network to follow the contour of a line demonstrates a networks ability to model mathematical functions. A practical application of this feature would be for the control of a robot arms movement. For the purpose of validation, the polynomial $y = x^3 + 4x^2 - \frac{1}{x}$ was represented by the network. The resultant network from the GEN-NEU program was capable of modelling the function with an RMS error of 0.0025. This was achieved with a 1-7-3-1 topology, a *tanh* transfer function and a training period of 7680 iterations of the training set. The progress of the GEN-NEU program for the training of this network is illustrated in Figure 7. From the path of the RMS error for the population it can be seen that the trend of the training aid is to improve performance. Erratic fluctuations in the performance can be attributed to the introduction of new unfit genomes to the search population.

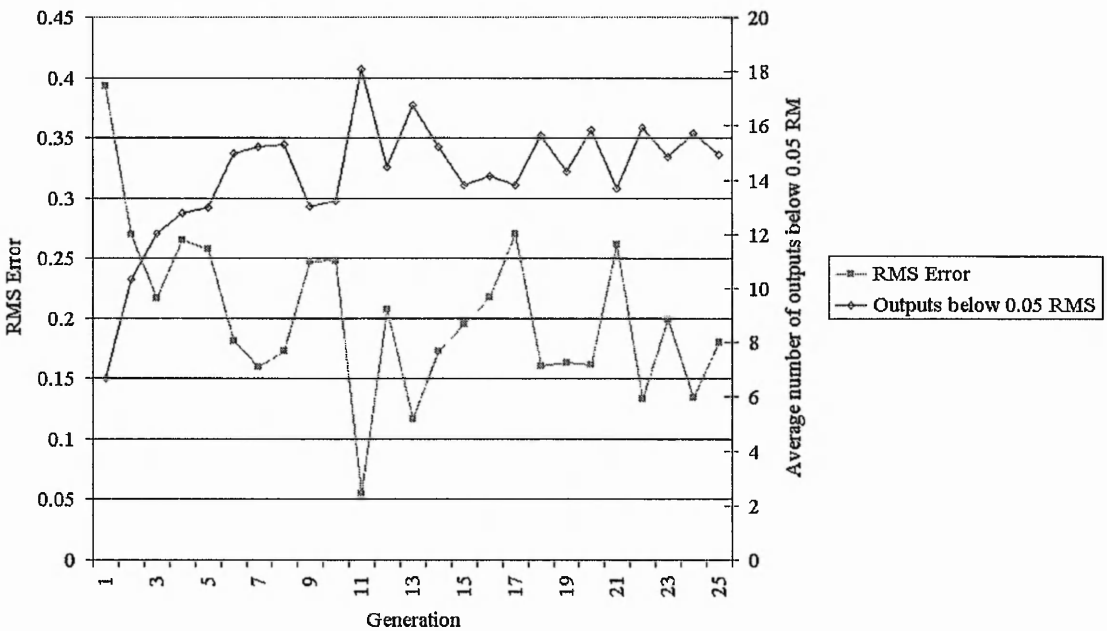


Figure 7. Average Performance of Networks During Training (Equation Modelling)

Signal Processing- Prediction

The capability of the network to predict values is another frequently used feature of ANNs, where an output is required from a region the network has not previously encountered, between training areas. Prediction was tested by applying Lapedes and Farbers (1987) signal processing problem, $x(t+1) = 4x(t) \cdot (1.0 - x(t))$. This application requires an output based upon the effects of the previous output, as in the case of a feedback loop.

The resultant network from the GEN-NEU program comprised of a 1-4-20-1 topology, using *tanh* transfer functions and trained for 7680 iterations of the training set. This training combination produced an RMS error of 0.0386 to the test data.

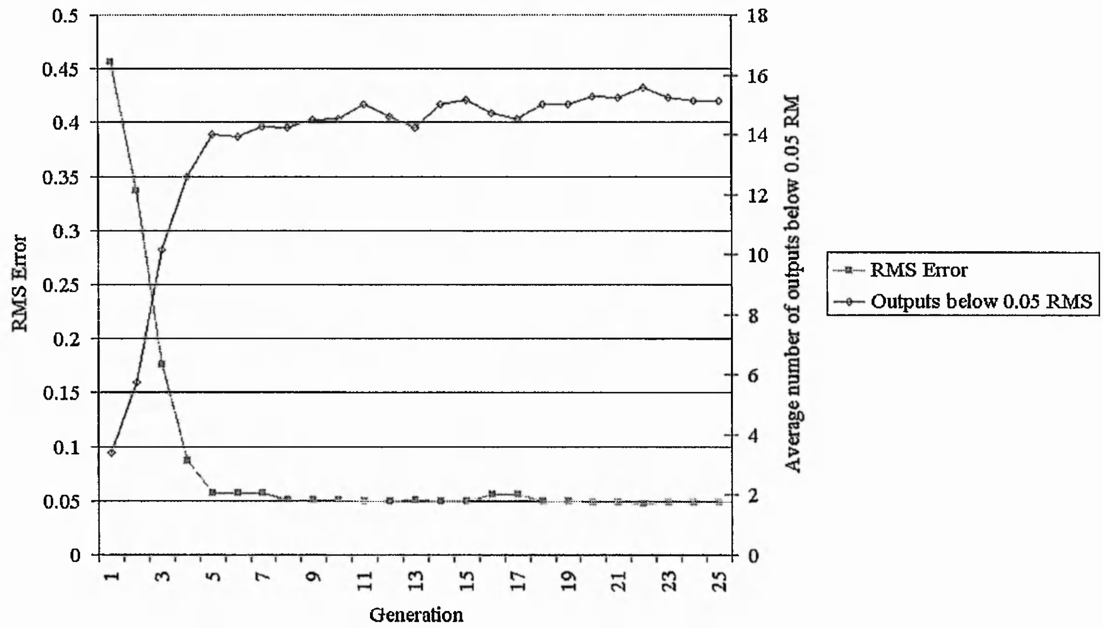


Figure 8. Average Performance of Networks During Training (Signal Processing)

As the training factors begin to converge into the region of the solution the average performance of the networks within the populations also begins to improve. This trend is evident in Figures 7 and 8 from the number of test RMS errors below 0.05. Thus the average network performance increases as the configurations of training factors that produce low performance networks are removed from the search, while the remainder combine to improve performance.

5. Conclusions

The GEN-NEU approach to aid the training of BP networks has proven to be capable of achieving successful results for a number of applications and input/output applications. As a result, the laborious process of trial and error and the need for rules to determine a successful network configuration has been removed from the training process.

The inclusion of the training period into the GAs search has the added beneficial effect of preventing over training. Over training decreases the networks ability to predict and generalise outputs based upon the training data. As deviation from the test data increases when the network loses its ability to generalise, the fitness of the genomes causing this phenomenon will decrease. The result of the decreased fitness is to reduce the occurrence of this training period in the search.

Although the training data is not included within the GA the effect of the amount of training data is compensated for by both the training period and network topology. As the training period is flexible and both small and large network topologies can be generated a combination of these factors are adjusted to model the data.

GEN-NEU is a general approach, which can be used with variations of the BP training technique and other supervised ANNs that require adjustment of several indirectly related factors.

References

B. Amirikian and H. Nishimura, What Size Network is Good for Generalisation of a Specific Task of Interest ?, *Neural Networks*, **Vol. 7**, No. 2 (1994), 321-329.

T. P. Caudell and C. P.Dolan, Parametric connectivity: Training a constrained network using genetic algorithms, Presented at the *Proceedings of the Third International Conference on Genetic Algorithms and Their Applications*, (1987).

M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design*, (1997), John Wiley & Sons, Inc. New York.

A. Lapedes and R. Farber, Non-linear signal processing using neural networks: Prediction and System Modelling, Los Alamos National Laboratory report LA-UR-87-2662, 1987,

V. Maniezzo, Genetic Evolution of the Topology and Weight Distribution of Neural Networks, *IEEE Transactions on Neural Networks*, **Vol. 5** No. 1 (1994), 39-53.

G. F. Miller, P. M. Todd and S. U. Hedge, Designing Neural Networks, *Neural Networks*, **Vol. 4** (1991), 53-60.

Z. Wang, C. Di Massimo, M. T. Tham and A. J. Morris, A Procedure for Determining the Topology of Multilayer Feedforward Neural Networks, *Neural Networks*, **Vol. 7** No. 2 (1994), 291-300.

M. Wakelam, Intelligent Hybrid Approach for Integrated Design, Ph.D. Thesis. The Nottingham Trent University, England, 1998.