THESIS REPORT
Master's Degree

SYSTEMS
RESEARCH
CENTER

Supported by the
National Science Foundation
Engineering Research Center
Program (NSFD CD 8803012),
the University of Maryland,
Harvard University,
and Industry

Optimization of Gear Ratios for Epicyclic Gear Train Transmission

by S.N. Mogalapalli
Advisors: E. Magrab and L-W. Tsai

M.S. 92-12

Abstract

Title of Thesis: OPTIMIZATION OF GEAR RATIOS FOR EPICYCLIC GEAR TRAIN TRANSMISSIONS

Name of degree candidate: Srinivas N. Mogalapalli

Degree and Year: Master of Science, 1992

Thesis directed by: Dr. Edward Magrab, Professor,
Department of Mechanical Engineering, and

Dr. Lung-Wen Tsai, Professor,
Department of Mechanical Engineering

An interactive design system has been developed for the automotive epicyclic gear trains that can provide at least three forward gears and one reverse gear. This user-friendly windowing system can access help files, display the functional representation of a mechanism, optimize the gear ratios for epicyclic gear trains and present the numerical results. The optimization procedure to find the optimum gear ratios and the corresponding number of gear teeth is the Augmented Lagrangian Multiplier Method, and can be applied to all gear trains having sets of three gears in which a ring gear is connected to a sun gear through a planetary gear. The procedure does not change for either different clutching sequences or number of clutches. The method is demonstrated for the Simpson and THM 440 gear trains. The gear teeth combinations are found such that they achieve the optimized gear ratios within less than 1% and satisfy the geometric constraints. An extension of the optimization procedure to the Ravigneaux gear train is also developed. PHIGS graphics functions are used in program routines to display the functional schematic of an epicylic gear train on the computer screen. These routines are written in such a manner that other types of gear combinations can be displayed by simply adding additional modules to represent these new gear elements.

OPTIMIZATION OF GEAR RATIOS FOR EPICYCLIC GEAR TRAIN TRANSMISSIONS

by

Srinivas N. Mogalapalli

Thesis submitted to the Faculty of the Graduate School
of The University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
1992

Advisory Committee:

Professor Edward Magrab, Chairman/Advisor
Professor Lung-Wen Tsai, Co-Advisor
Associate Professor Donald Barker

Dedicated

To that unknown person who is yet to lead me beyond the cycle of life and death

Acknowledgements

I would like to thank Dr. Magrab for his continuous support during the course of my thesis work. Dr. Magrab let me work independently but helped me stay on track and maintain perspective on issues rather than on details. I would like to thank Dr. Tsai for helpful insights on the subject every time I met him. Many times I wished that I had interacted with him more often. Again, I would like to thank Dr. Barker for serving on my advisory committee. Dr. Barker's initial help related to PHIGS is really appreciated.

My thanks go to numerous others who helped me in my thesis, including the Open Sun Lab staff, Mike McMahon, John Cugini of NIST, Rick Chimera and several others.

My stay in Maryland has been a very different experience. Financial support from the Department of Mechanical Engineering was essential. However, the weekly movie sessions, supportive roommates, friendly officemates, the beauty of the campus and several other factors have all made this period a much more enjoyable one.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Mechanisms are most frequently used to generate specific motion and forces. During the conceptual phase of the design of the mechanisms one attempts to satisfy a given set of functional requirements. Traditionally, the designer's intuition, creativity and experience have played a large role in the successful design of mechanisms.

To aid the designer in the design of mechanism and to minimize the dependence on the designer's experience a substantial amount of research has been done in the last few decades on the use of computers to systematically create mechanisms. This thesis is concerned with the class of epicyclic gear train (EGT) mechanisms which can be potentially used as Automotive gear trains.

In this chapter the objectives of the thesis are stated. The background leading up to this work is briefly summarized in Section 1.2. The last section explains the organization of the remaining chapters and the appendices.

1.1 Objectives

The objectives are to:

- Develop a methodology for the functional representation of epicyclic gear trains. This aids the design process in the stage of structure enumeration.
- Apply optimization techniques to find the gear ratios in automotive gear trains for a given set of reduction ratios. This is a part of the design and analysis stage of automotive mechanisms.
- Develop a friendly, window-based user interface for accessing the work done in the previous two stages. The window-based system helps the designer to perform his task effectively and efficiently.

1.2 Background

1.2.1 Graph Representation

Several methods for representing the topology of a mechanism exist. Graph representation of a mechanism is an abstract method that highlights the topology of a mechanism very clearly. A *graph* is a set of collection of vertices (points) in space and which are interconnected by a set of edges (lines). The graph of a kinematic chain with binary joints is a graph in which links correspond to vertices, joints to edges, and the joint connection of links to the edge connection of vertices. When more than one kind of joint is involved, the edges are labeled (or colored) according to the type of joint. For example, while representing a gear train, the gear pairs can be represented by heavy edges and the turning pairs can be represented by thin edges. The graph representation of the mechanisms shown in Figure 1.1 can therefore be represented as shown in Figure 1.2. Note that the graph for both mechanisms is the same, and hence, the correspondence is not unique. This is because the type of gear mesh in the graph represenation is not specified.

Several definitions for graphs have to understood before rules concerning these graphs are either derived or classified. Some of the relevant definitions can be found in [26], etc.

In order to automate the process of the systematic analysis of mechanisms it is necessary to have a symbolic representation that can be computerized. Matrix representation of mechanisms serves this purpose. Different types of *Adjacency matrices* which characterize a mechanism in terms of matrices can be found in reference [23]. For example, a link-to-link adjacency matrix for an n link mechanism is defined as an $n \times n$ matrix with elements given by

$$a(i,j) = \begin{cases} 1 & \text{if link } i \text{ is connected to link } j, \\ 0 & \text{otherwise (including } i = j) \end{cases}$$

1.2.2 Epicyclic Gear Trains

We confine ourselves to mechanisms and graphs that are known as the epicyclic gear trains. The creation of epicyclic gear trains using graph representation has been carried out by Buchsbaum and Freudenstein [2] initially, and several others subsequently [11, 13, 24]. Buchsbaum and Freudenstein [2] defined

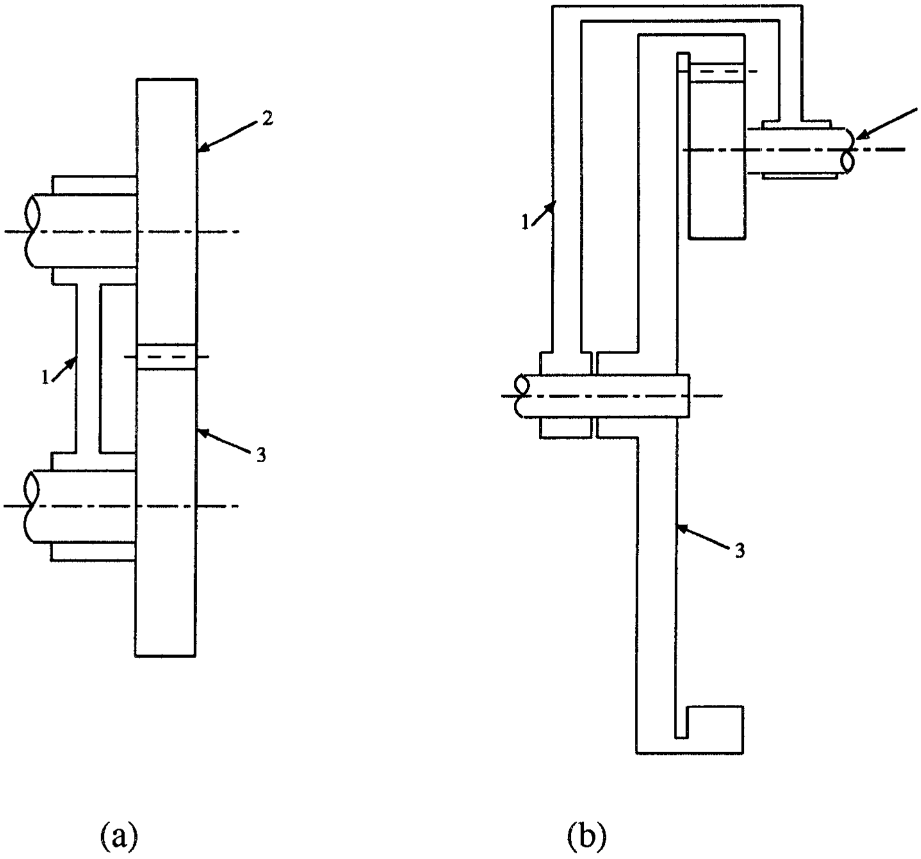


Figure 1.1: Functional representation of a mechanism: (a) an external gear mesh, (b) an internal gear mesh.

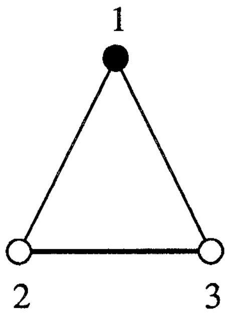


Figure 1.2: Graph representation of Figure 1.1

EGTs as geared kinematic chains that contain only revolute (R) and gear (G) pairs and which obey the following rules :

R1: The mechanism shall obey the general degree-of-freedom equation; that is, no special proportions are required to ensure the mobility of an epicyclic gear train, and its joints are binary and its graph planar.

R2: In the graph of an EGT there shall be no circuit that has either a zero or negative degree of freedom.

R3: The rotatability of all links shall be unlimited. Mechanisms with partial mobility or with partially locked structures shall be excluded.

R4: Each gear must have a turning pair on it's axis, and each link in a gear train must have at least one turning pair in order to maintain constant center distance between each gear pair.

Properties of EGTs with only one degree-of-freedom that can be concluded using this definition, the general degree of freedom equation for a mechanism, and results from graph theory can be found in [6].

1.2.3 Displacement Graphs and Displacement Equations

The *rotation graph* of an epicyclic chain is defined [6] as the graph obtained from the graph of the epicyclic chain by deleting the turning pair edges and the transfer vertices, and labeling each geared edge with the symbol for the associated transfer vertex. The rotation graph is obtained by joining the subgraphs at the common vertices and edges. Two rotations graphs are isomorphic if there is a one-to-one correspondence between their vertices, which preserves incidence and labeling. Figure 1.3(a) shows the graph representation of an EGT, and Figure 1.3(b) shows its rotation graph.

Let i, j denote two adjacent vertices of a geared edge in a rotation graph in which k is the associated transfer vertex. Let ω_i, ω_j, and ω_k denote the angular velocities of links i, j, and k, respectively. Let r_{ji} denote the gear ratio of the gear pair on links j and i, either positive or negative depending on whether the mesh is internal or external; that is, $\mid r_{ji} \mid = t_j/t_i$ where t_j and t_i denote the number of teeth of the gear pair on links j and i, respectively.

The links i, j, and k constitute a simple epicyclic gear train for which k is the *gear carrier*. The angular velocities of these elements are related by the linear equation

$$\omega_i - r_{ji}\omega_j + (r_{ji} - 1)\omega_k = 0 \tag{1.1}$$

For a one degree-of-freedom EGT with n links, there are $n-2$ geared edges. Equation (1.1) is written $n-2$ times, one for each of the basic EGTs. Since

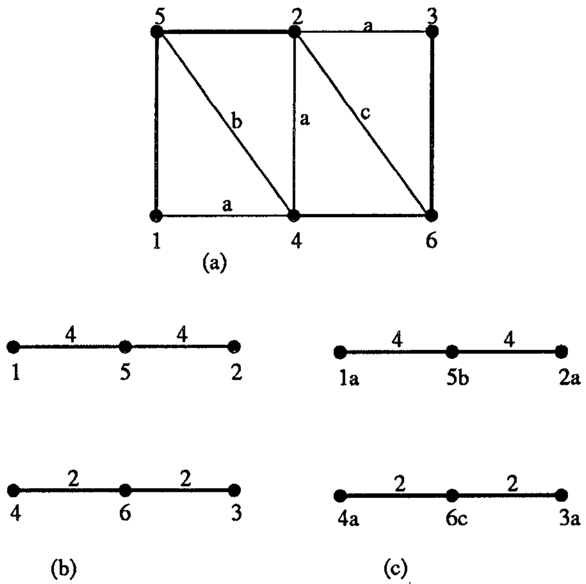


Figure 1.3: (a) Graph of an EGT (b) rotation graph of the graph in (a) (c) displacement graph of the graph in (b)

one of the n links is grounded and another is used as the input, there are $n-2$ unknown angular velocities. Hence, the $n-2$ equations uniquely determine the angular velocities of the EGT. Together, these equations comprise the *rotational displacement equations* of the epicyclic chain.

The rotational displacement equations for the graph in Figure 1.3 can be written as [24, 25]

$$[R][W] = [0]$$

where

$$R = \begin{bmatrix} 0 & -r_{25} & 0 & r_{25}-1 & 1 & 0 \\ -r_{15} & 0 & 0 & r_{15}-1 & 1 & 0 \\ 0 & r_{46}-1 & 0 & -r_{46} & 0 & 1 \\ 0 & r_{36}-1 & -r_{46} & 0 & 0 & 1 \end{bmatrix}$$

and

$$W = \begin{bmatrix} \omega_1, & \omega_2, & \omega_3, & \omega_4, & \omega_5, & \omega_6 \end{bmatrix}^T$$

If two epicyclic chains have isomorphic rotation graphs, their rotational displacement equations are isomorphic.

The *displacement graph* of an epicyclic chain can be obtained by labeling each vertex in a rotation graph according to the levels of the edge pair connecting it to the transfer vertex in the f−circuits [6] in which the vertex lies. If the vertex is associated with several levels, each level should be associated with the corresponding f−circuit. Figure 1.3(c) shows the displacement graph of the graph in Figure 1.3(a).

Two displacement graphs are said to be isomorphic if there is a one-to-one correspondence between their vertices and edges, which preserves incidence and labeling of edges and vertices.

A set of *linear displacement equations* analogous to rotational displacement equations can be obtained from the displacement graph of an epicyclic chain. Furthermore, if the displacement graphs of two epicyclic chains are isomorphic, their rotational and linear displacement equations are isomorphic.

1.2.4 Categorization of EGTs for Automatic Transmissions

Tsai[24] has shown that there are 80 non-isomorphic displacement graphs from which all of the six-link, one-degree-of-freedom gear trains can be constructed. Later Tsai, et al. [27] have shown that there are only seven non-isomorphic displacement graphs from which all epicyclic gear trains with up to six links and with four or more coaxial links can be derived.

For performing the kinematic structural analysis of gear trains found in automatic transmissions Tsai, et al. [27] have considered two issues: (a) the topology of gear trains used in the transmissions; and (b) the arrangement of the clutches. In order to identify the fundamental features, certain simplifying rules were considered.

Examining existing transmissions it was observed that only coaxial links of an EGT are used as the input, output, or clutched to the frame to obtain the desired speed ratio[1]. To provide a minimum of three forward speed ratios and one reverse, it was found that, there are

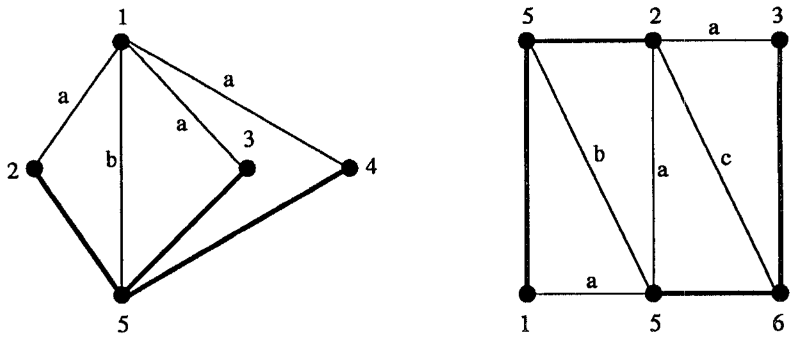
- One five-link chain with four coaxial links as shown in Figure 1.4(a).
- Six six-link chains with four or five coaxial links as shown in Figures 1.4(b) to 1.4(g)

Tsai, et al [27]. have also identified nineteen types of four-speed automatic transmissions, and these were classified as

1. Transmissions using one-degree-of-freedom gear trains with up to six links.
2. Transmissions using one-degree-of-freedom gear trains with more than six links.
3. Transmissions using two one-degree-of-freedom gear trains in series.

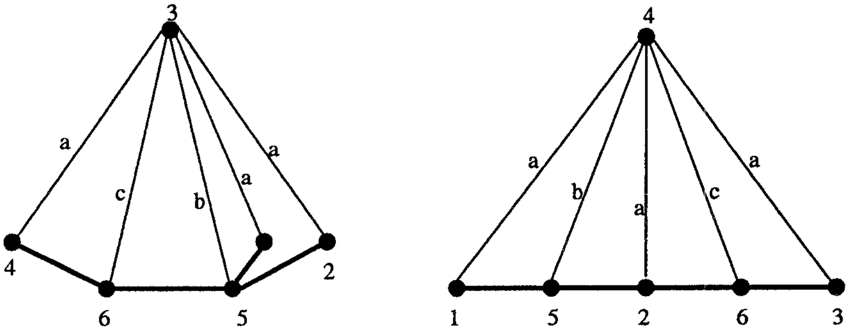
---

[1] *Speed ratio* is the ratio of the rotational speeds of the input link to the output link.



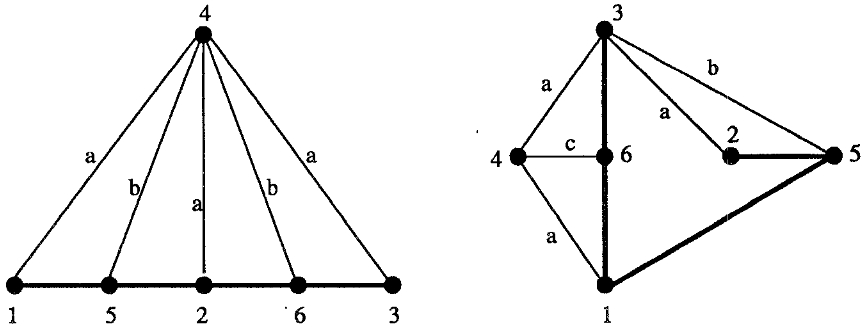
(a) Type 5301

(b) Type 6206-3



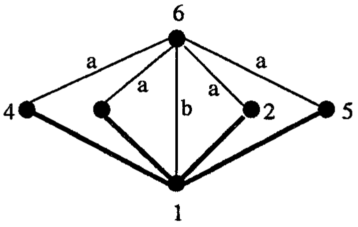
(c) Type 6401-9

(d) Type 6503-7



(e) Type 6503-8

(f) Type 6506-1



(g) Type 6601-4

Figure 1.4: Graphs of gear trains with six links or less and with four coaxial links.

Furthermore, it was shown that all practical six-link transmissions belong to the six non-isomorphic displacement graphs.

1.3 Organization of Thesis

Chapter 2 describes a methodology for displaying these gear trains on a computer screen. A graphics system called PHIGS is used for the graph representation and functional representation of EGTs. The methodology is demonstrated for two automotive gear trains: the Simspon gear train and the THM-440 gear train. Chapter 3 formulates the optimization problem to find the best gear ratios given a set of reduction ratios for a certain class of automotive gear trains. The results for two specific gear trains are discussed. The rotational displacement equations for the gear trains that are determined from the graph theory are used in the optimization problem. Chapter 4 discusses the development of a user friendly windowing system in Xview (a windowing tool kit in the Openwindows environment on the Sun operating system) for executing the optimization problem and for displaying the gear trains using buttons, pop-up windows, etc. Chapter 5 summarizes this work and suggests several extensions for future work.

Appendix A describes the program's logic with the help of flow charts. For the purposes of illustration, a listing of a module that generates the functional representation of an external gear is included. The optimization methodology is extended to a Ravigneaux automotive gear train in Appendix B.

Chapter 2

Functional Representation Using PHIGS

Functional representation refers to the conventional drawing in which the cross section of a mechanism with elements like shafts, gears, links are shown as such. To show the structural topology of a mechanism it is not necessary to represent exactly a scaled figure based on actual dimensions. However, if a mechanism has been subjected to dimensional synthesis based on stated requirements, it is much more appropriate and logical to display the functional representation as a scaled diagram. Whatever the scale may be, such a display aids the designer in having a more realistic view of his design.

In order to draw the cross section of a gear train, one has to know the dimensions of each of the elements in the gear train. For example, to draw the cross section of a simple external gear, we need to know its diameter (pitch circle diameter), thickness, bore size to accommodate the shaft about which it is rotating, and the dimensions of the hub, if it has one. In order to display elements with varying dimensions it is best to describe the various elements in terms of parameters. A display tool that displays the cross section of a gear train should be capable of interpreting these parameters.

This chapter deals with the methodology that has been developed for the display of the functional representation of a gear train. Both PHIGS and C have been used to obtain the display of various elements comprising the EGT. Section 2.1 gives a description of PHIGS. The next section compares PHIGS with other alternatives. Section 2.3 gives an overview of the program for display of the functional representation of any gear train. The last section displays some gear trains using the programming method described in Section 2.3.

2.1 PHIGS

PHIGS is an acronym for Programmer's Hierarchical Interactive Graphics System. PHIGS standard is an international graphics programming standard developed by the International Standards Organization (ISO) and the American National Standards Institute (ANSI). PHIGS is a graphics system composed of language-independent functions for programming applications that require computer generated images on raster-graphic or vector-graphic output devices. PHIGS allows for the creation, storage and dynamic modification of these images [5].

PHIGS is based on the concept of an abstract workstation. An abstract workstation is defined for an application program, which interprets the graphic attributes of the abstract workstation and translates them into device specific graphic attributes for a physical workstation. Actual workstations and devices differ from each other in hardware-specific ways. Hardware peculiarities are resolved through the use of an intermediate layer of software. Thus, the abstract workstation is device-independent. Output and attribute primitive elements are uniformly defined and organized for the abstract workstation. Some of the important features of PHIGS are:

- PHIGS is a 3D system. The user, however, can limit it to a 2D system using 2D features only. In order to display an image, the three dimensional information is transformed to two dimensional information through a transformation pipeline with the flexibility of a complete transformation and viewing system. The transformation system is described in more detail later in this section.

- PHIGS creation and editing of graphics data are defined by the application program in the database, independent of the actual display of the graphics on a display surface. The application program determines the display criteria: how, when, and where to display the graphic information.

- PHIGS uses structures to organize and control graphical data. Graphic images are created from structure networks. A structure is a group of elements combined to create an image component, such as a cube or a cylinder. A structure network can be composed of structures, basic image elements and/or other structure networks. Structures are stored in the centralized store structure (CSS), the workstation independent, conceptual storage area for structure and structure networks. When a structure is invoked, graphic primitives (such as a line) and graphic attributes (such as the color blue) combine to make a graphic image. Every structure has a structure identifier (ID). With PHIGS, primitive attributes are stored as elements of a structure. By changing an element, the structure is dynamically modified. The application program can change attributes without changing the data structure itself by dynamically modifying attribute elements. Furthermore, the hierarchical arrangement of structure in PHIGS permits subsequently linked structures to inherit the attributes of their predecessors.

To understand how an image of, say, a gear train with several elements is displayed, it is necessary to understand the coordinate transformations in PHIGS. The coordinate systems that are necessary for 3D representation on a 2D surface are:

- Modeling Coordinate System
- World Coordinate System
- Viewing Reference Coordinate System

- Normalized Projection (NP) Coordinate System

- Device Coordinate System

These five coordinate systems are discussed in the sections that follow.

2.1.1 Modeling Coordinate System

A large image usually has several subimages. Often it is more convenient to define subimages with respect to its own origin. Once an image is defined in relation to itself (the modeling coordinates), it can then be defined in relation to other images. For example, if a composite image of a face on a head is being defined, one can create an eye (along with other parts of the face) in modeling coordinates. The *modeling transformation* translates the coordinates of the eye so that a relationship can be created between the eye and the other parts of the defined. The ability to define images in modeling coordinates greatly aids the process of repeating subimages in an image. An eye, for example, defined in modeling coordinates can be placed twice on a face using two different modeling transformations.

2.1.2 World Coordinate System

The world coordinate system is used to express an image in relation to other images with respect to a common origin. Once images are defined in relation to other images, the world coordinates are translated into view reference coordinates that define the perspective from which the image is viewed.

2.1.3 View Reference Coordinate System

The view reference coordinate system is used to represent a 3D image in relation to other images and in relation the view perspective using a *projection reference point*, a *view reference point*, a *normal perspective view*, and a *view up vector*. These terms are defined in reference [29]. The view perspective is akin to viewing through a camera view lens. The viewing system parameters are analogous to the camera's position, rotation angle, zoom factor, aperture etc. that affect how the image appears. An additional feature (that is not found in cameras) is that before an image is translated into normalized projection coordinates, a *parallel* or *perspective* projection can be selected for the image.

2.1.4 Normalized Projection Coordinate System

The normalized projection coordinate system represents a 3D image that is 'flattened' for display on a workstation. Because some workstations can exhibit several windows, an image can be displayed in different perspectives, and is therefore mapped (conversion of view coordinates to normalized projection coordinates) for each of the displayed images.

2.1.5 Device Coordinate System

The device coordinate system represents a 3D image on a 2D display surface of a physical workstation. The image is mapped onto the workstation display surface by device-specific XY-coordinate values.

In order to display an image, the user program defines all the various subimages in their modeling coordinates and specifies the modeling transformation, world transformation and the viewing system parameters. If the image is now asked to be displayed, the image is processed through the transformation pipeline and then displayed onto the screen. All this and other PHIGS operations are achieved through function calls to PHIGS that are part of the user code. The process of displaying a simple image with one structure can be described as:

1. Open PHIGS and open workstation - PHIGS function calls can now be accessed.
2. Create Structure - This consists of three stages: (i) Open Structure, (ii) Construct Structure, and (iii) Close Structure. In general for a structure network, each structure could further be composed of several structures or structure networks. Also, transformation functions are specified here.
3. Post structure - The structure is posted onto the screen (displayed on the screen) by traversing the structure (or, in general, a structure network) after assigning attributes and transforming it through the transformation pipeline.
4. Close workstation and close PHIGS.

One more point to be noted about PHIGS is that PHIGS standards are currently being reviewed and extended. Different vendors offer extended options and function calls through PHIGS+ and PHIGS Extensions. The work presented here is limited to standard PHIGS functions and was tested on both VAX/VMS (DEC PHIGS) and the Sunsystem PEX (PHIGS with X Windows) [30]. To take care of the system differences, conditional statements have been included in the programs, and an additional argument specifying the system type must be provided at compile time.

2.2 PHIGS vs Alternatives

Two types of alternatives for drawing the images were considered: (i) graphics application developers like SunCGI [31]; (ii) commercial software graphics packages in a 'batch-mode' environment.

SunCGI has the following disadvantages for which it was not selected: (1) It is based on GKS - 2D and, hence, is limited to 2D. (2) The database is a linear list; segments are not linked or hierarchically ordered, and segments cannot invoke other segments. (3) Attributes are assigned when a primitive is defined. Attributes cannot be changed without changing the data structure. GKS - 3D is now currently available and has eliminated many of these problems.

The option of using the commercial software graphics package IDEAS was considered. IDEAS is a commercially available comprehensive solid modeling, drafting and finite element analysis package. IDEAS itself has been developed on PHIGS. IDEAS has an extremely powerful solid modeler. In order to generate images in IDEAS (say a gear train) IDEAS has to be opened when a display is needed and script files have to be run to generate images automatically. This approach has the following disadvantages: (1) Script files have to be created in the IDEAS's command language Ideal. These script files are basically text files containing commands and data as if the user were typing them step by step when using IDEAS interactively. These text files must be created from the information of the dimensional data of the elements of the gear train and the knowledge of how the various elements of the gear train are linked together. (2) There is a dependency on an external package. A package like IDEAS is very expensive and requires large amounts of memory and disk space. (3) Control of the program is lost to an external package. The user, also, is unlikely to be familiar with a complex program like IDEAS.

IDEAS, having been based on PHIGS, is commercially available on several computer systems. Other graphics packages are not as widely available on different computer systems. It should be noted that despite the disadvantages stated above, using a package like IDEAS is still a viable option. If long term goals are to display complex 3D figures, built in features in IDEAS for displaying, editing, viewing, etc. are much more convenient to use and outweigh the other disadvantages.

Because of the disadvantages of these alternatives and the advantages of PHIGS as mentioned in the previous section, PHIGS was selected as the display tool.

2.3 Graph and Gear Train Display

Currently, the representation has been limited to 2D cross sectional views of the gear trains. Towards developing a 3D display of gear train, a module for a 3D wireframe drawing of a cylinder with a bore has been written. This attempt was eventually abandonded in view of the amount of programming involved. However, PHIGS function calls are made in 3D to facilitate easy upgrading at a later stage. Information that is assumed to be known and the assumptions that have been made are mentioned where appropriate.

Given a gear train identification number [24], the objective is to display the graph showing the structural topology and the functional schematic of the gear train. To demonstrate the methodology, the task is restricted to displaying the functional schematic of two commercial gear trains and their corresponding graphs. It is noted again that there is no one-to-one correspondence between a graph and its functional schematic. (See for example Figure1.1 and Figure 1.2.)

Consider a PHIGS square display window with the length of its side equal to one. Any image that is displayed will be correspondingly scaled to fit in this window. Once the graph number is known, the graph is initially displayed in the upper left corner of the display window, and then the functional schematic is scaled to fit in the remaining area.

The turning pair adjacency matrix, gear pair adjacency matrix, and axes levels obtained for a given graph are assumed to be known. Ideally these should be generated from an algorithm based on the graph number selected.

Every graph is assumed to fit into a 1×1 unit square of side 1 in its modeling coordinates. These coordinates are assumed known and are obtained so that the resulting image is a planar connected graph (i.e., no crossing edges). Ideally, this should be connected to an algorithm that can generate coordinates from the adjacency matrices. Research is currently being carried out in the area of generating planar connected graphs without crossing edges [3, 12, 19]. Having obtained the vertex coordinates and the edge connections the graph is created. A scaling and translation transformation is applied to display the graph in the upper left corner of the display window.

To draw the functional representation of the EGT information concerning the number of elements, type of elements and the data for each element must be known. Each element (internal gear, external gear, etc.) is identified by a number. At present, four major modules have been written to create PHIGS structures for the following element types: internal gear, external gear, carrier and shaft. More modules for different element types can be easily added, and each module itself can be upgraded to represent an element with more details and complexity. The four element type modules are described below, with the internal gear module described in more detail.

2.3.1 Internal Gear Module

The internal gear is composed of the parameters shown in Figure 2.1(a) . The coordinate locations of the vertices that are numbered are described in terms of the parameters shown with respect to the local origin. Line 1–10 may or may not be drawn, depending on whether or not the shaft has a fixed or rotating center. The options that can be specified for drawing an internal gear are:

- orientation of the gear — the gear can be facing left or right.
- presence or absence of a shaft at its center
- when the shaft is present, whether it is inside or outside the internal gear; such a shaft usually acts as a carrier to some other element in the gear train.

These options are shown in Figure 2.1(b) and must be specified in the data file prior to drawing the element. If a hub is not present 'hub length' must be specified as half the arm thickness.

2.3.2 Multiple Gear Module

Only a multiple gear with external gears is considered. The multiple gear is shown in Figure 2.2. The coordinate locations of the vertices that are numbered are described in terms of the parameters shown with respect to the local origin. The multiple gear can be either a double or triple gear.

Orientation of a multiple gear is not an issue. The gears are all external and the gears have a turning pair at their centers.

2.3.3 External Gear Module

The external gear is shown in Figure 2.3. The coordinate locations of the vertices that are numbered are described in terms of the parameters shown with respect to the local origin. Option 1 is a fixed or rotating center for gear, and option 2 indicates whether the hub is to right or left of the local origin.

2.3.4 Carrier

The carrier is shown in Figure 2.4. The coordinate locations of the vertices that are numbered are described in terms of the parameters shown with respect to the origin. Option 1 gives the orientation of shaft number two, which can be either left or right with respect to the local origin. Rotating type joints are assumed at both ends of the carrier.

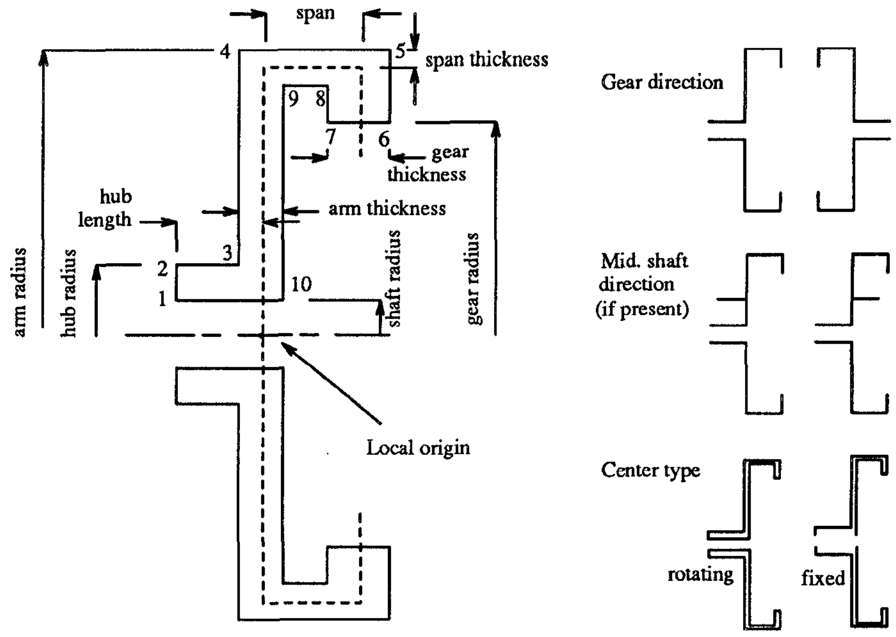
2.3.5 Shaft

A shaft is described by its diameter. A shaft can, however, be connected rigidly to any number of elements in the gear train. A typical schematic of a shaft may look like that shown in Figure 2.5(a). The local origin is placed at the center point, which is at the connection that is left-most on the shaft. The ends of a shaft are approximated as shown in Figure 2.5(b). By specifying the end type, the left and right ends of a shaft may or may not be drawn. For the purposes of the drawing, the connections are assumed to be one of the three types as shown in Figure 2.5(c).

Based on the gear train several appropriate data files are read to supply the relevant information of the various elements. Data files have the information concerning the element's parameters, their orientation, etc., specified in a certain order. These files would normally have been generated at an earlier stage say in the design process. Each element is described with repsect to its local coordinates. It is assumed that the relative location of each element is known with respect to the origin of a complete gear train.

2.4 Results and Discussion

Once all the relevant data are known, the program creates a structure network that is composed of several structures for the various elements in the gear train, with the specified transformations based on their relative locations. (The structure is only created, but not yet displayed.) The size of the gear train varies dynamically based on the current dimensional data. Since, irrespective of the size, this figure is scaled appropriately to fit on the screen, a scheme has been developed to keep track of the minimum and maximum xy-coordinates of each element, and an array of these values is used to scale the figure to fit in the window. This task is manageable for simple mechanisms because of the fact that the figure is limited to 2D. An added feature could allow the user to choose and vary the size of the display dynamically. An analogous autoscale



(a) (b)

Figure 2.1: Internal gear (a) parameters (b) options.

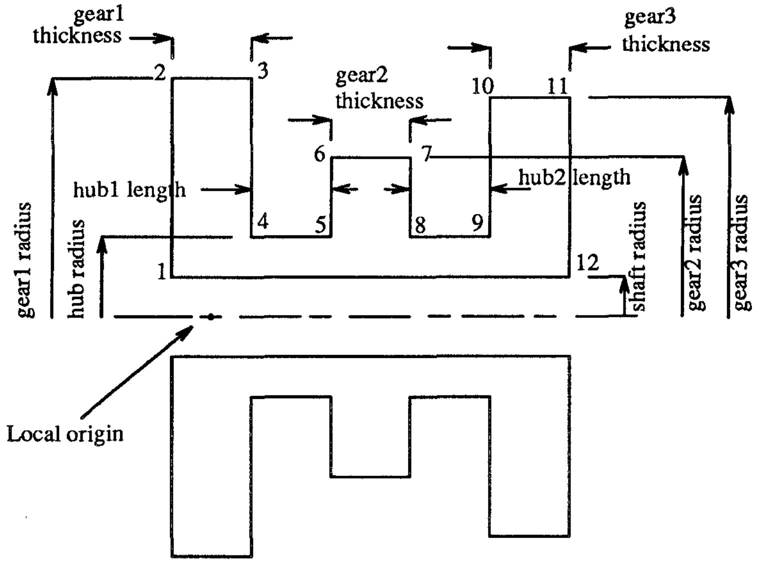


Figure 2.2: Multiple gear (a) parameters.

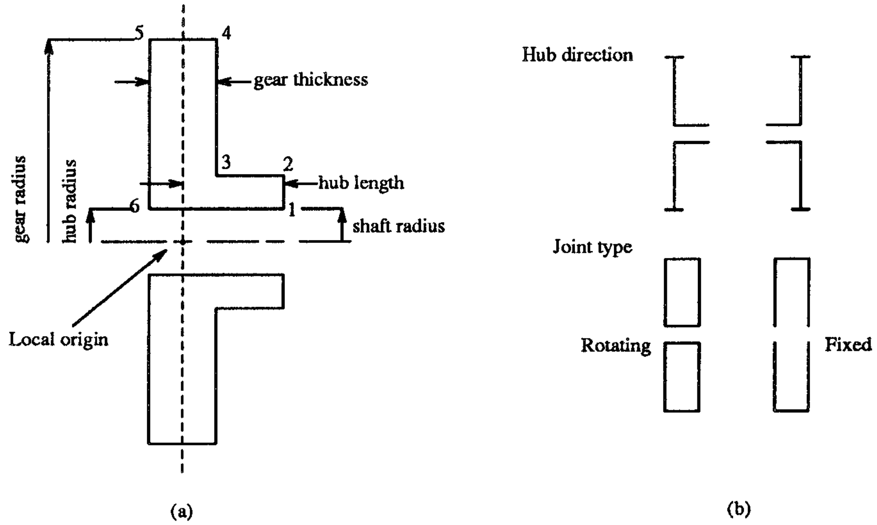


Figure 2.3: External Gear (a) parameters (b) options.

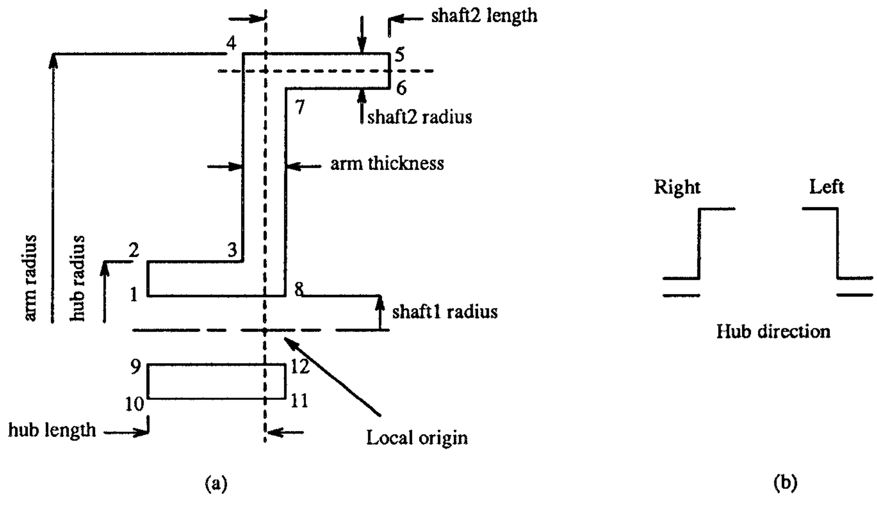


Figure 2.4: Carrier (a) parameters (b) options.

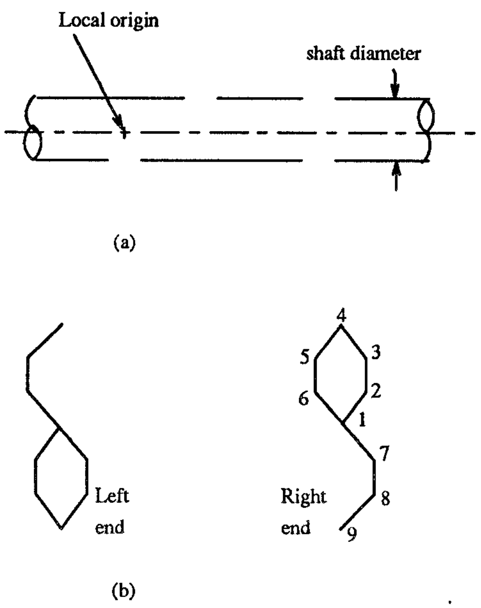


Figure 2.5: Shaft (a) typical cross-section (b) end approximation (c) connections.

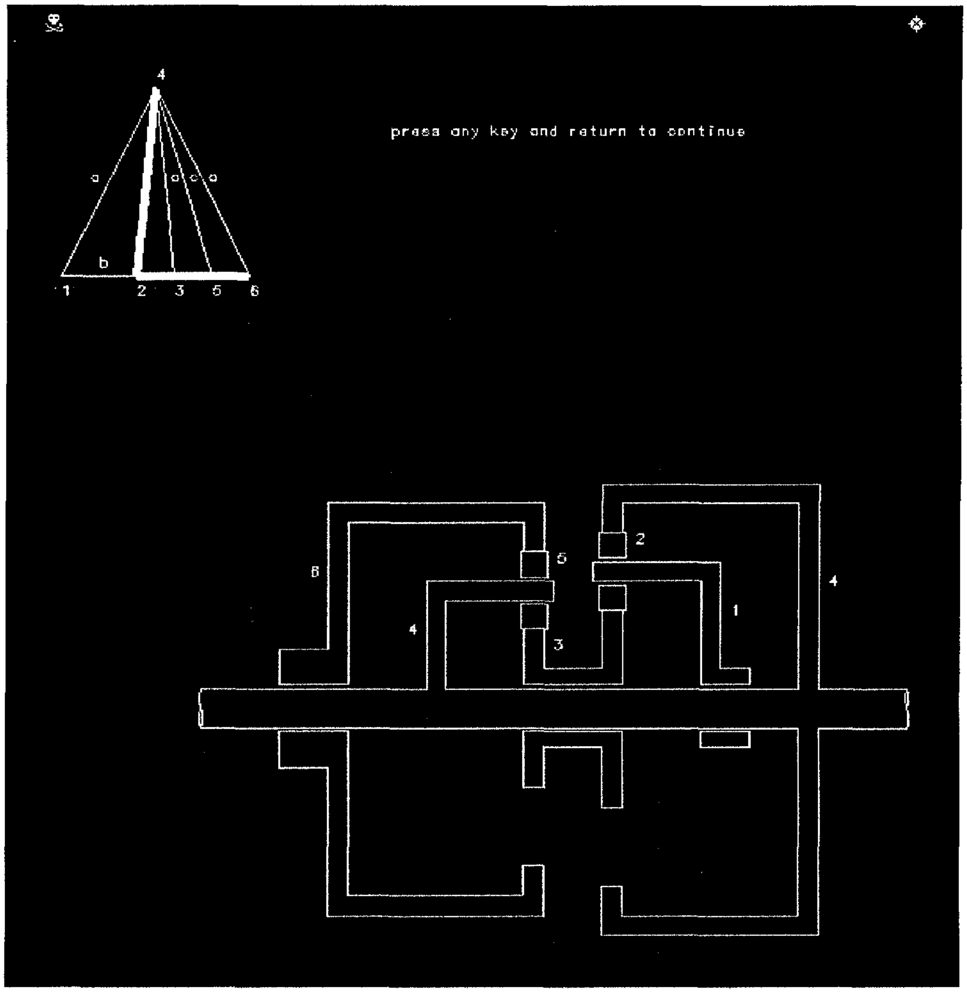
option might be necessary only in case of complex 3D images. Once the scale is found, the gear train structure is scaled and displayed onto the screen.

One important point to note is that all the text should be placed at the current location of the graphic elements, irrespective of the scaling and other transformations applied to the image. PHIGS fortunately provides the options for selecting the text parameters like height, offset, etc., in normalized projection coordinates with respect to some reference point that is known in the modeling coordinates.

To demonstrate the procedure, the Simpson gear train (Figure 24 in [27]) and the THM 440 gear train (Figure 2 in [27]) are selected. An optimization has also been applied to find gear ratios for these two gear trains in Section 3.1.

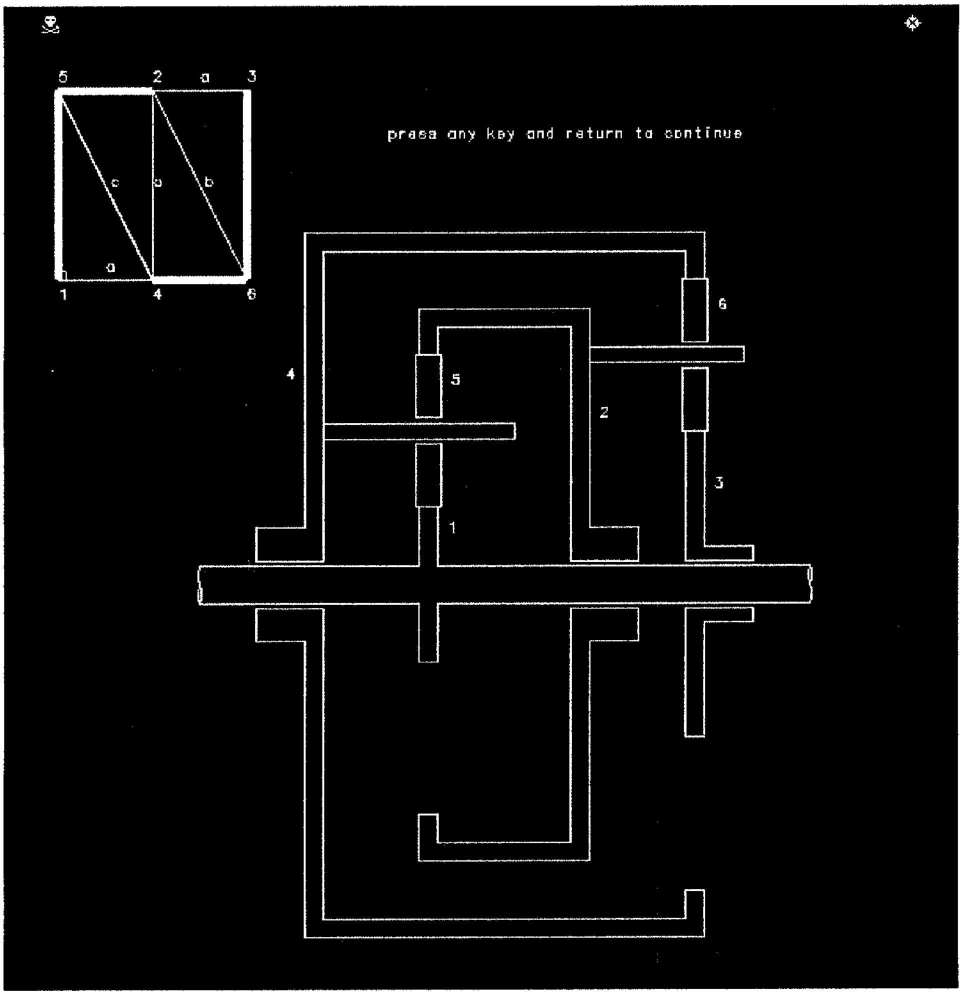
A data set has been created to test the display of these two gear trains and the results are shown in Figures 2.6 and 2.7. Appendix A, which has a description of the entire program developed for the current and subsequent chapters, includes flow charts for the creation of the functional schematic of the gear trains. It also includes the listing of the computer module that generates the functional representation of an external gear.

Some researchers [17] are currently working to generate functional representations, including 3-D shaded images, of various types of mechanisms based on the graph representation. Such images are, however, limited to external-external gear pair combinations, although equivalent mechanisms can be obtained using external-internal gear pair combinations. Although the approach here does not generate 3-D shaded images of the gear trains, it is not limited to gear trains containing only external gears and it gives a cross section of the gear train that can be understood by any designer.



press any key and return to continue

Figure 2.6: Functional representation of Simpson gear train



press any key and return to continue

Figure 2.7: Functional representation of THM 440 gear train

Chapter 3

Optimization of Gear Ratios for EGTs

In the systematic creation of mechanisms the functional evaluation stage deals with the selection of acceptable mechanisms (or graphs for the mechanisms) from the enumerated list of mechanisms, in order to satisfy the functional requirements. The next stage in the creation of mechanisms is its design and analysis.

In this stage dimensional synthesis techniques are used to determine the mechanism's proportions. Tsai, et al. [27] determined that there are only seven graphs (Figure 1.4) with six links or less for EGTs that can provide a minimum of three forward speed ratios and one reverse speed ratio. To achieve a specific set of reduction ratios the designer has to choose a gear train with specific internal-external gear pair combinations, a set of clutches that are to be operated in a chosen sequence, and a set of gear ratios that will provide a set of reduction ratios based on the rotational displacement equations. The classical approach of finding proper gear ratios has been to choose a gear train and a corresponding clutching sequence, and then to vary the gear ratios by trial and error until the best possible reduction ratios have been obtained. Note that the term gear ratio is used to denote the ratio of a gear pair in contact while the term reduction ratio is used to denote the speed ratio of the output link to the input link of a transmission mechanism.

In this chapter we describe an optimization methodology for determining the optimum gear ratios for a particular set of gear train. We demonstrate the methodology for two gear trains, each with a different number of reduction ratios. In both gear trains there are two sets of three gears each. Each set has a sun gear connected to a ring gear through a planetary gear. Next a clutching sequence is selected. The optimization formulation developed can be applied to all the variations of EGTs that can be obtained by changing the clutching sequences, clutch location and the number of clutches. For example, the optimization problem can be equally applied to the gear train in the Hydramatic THM 440-T4 (Figure 2 in [27]), the Ford Axod (Figure 3 in [27]) and the Hydramatic 700-R4, which differ only in their clutch locations or number of clutches.

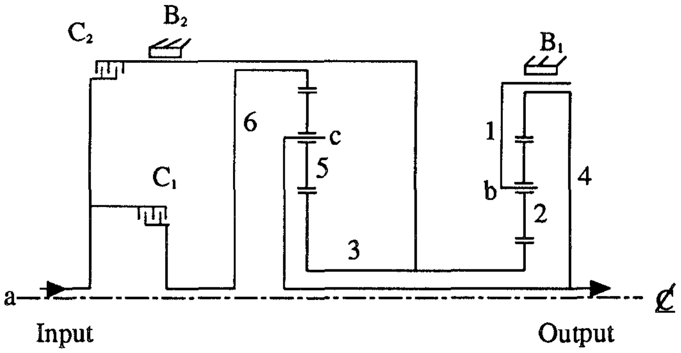
The optimization problem can also be set up for all the gear trains that fall in similar categories, which result from a systematic enumeration of EGT mechanisms resulting from the seven graphs using various combinations of internal-external, external-internal and, external-external pair types for each gear pair. The phrase similar category means the gear trains in which there are sets of three gears, each of which has a sun gear connected to a ring gear through a planetary gear.

The two specific gear trains for which the optimization problem is formulated are the Simpson gear set (Figure 24 in ref [27]) and the THM 440 gear set (Figure 2 in [27]). The clutching sequence, which includes the brake clutches, is assumed to be the same as those mentioned in [27]. The choice of the gear pairs (internal-external, external-external, external-internal) is also assumed to be the same. The identification numbers for the graphs corresponding to the gear trains have also been retained. The next section describes the way in which the problem is formulated. In the section that follows the procedure and algorithms that are used for the optimization procedure are described. In the last section of this chapter a discussion of the results obtained from the optimization method are presented.

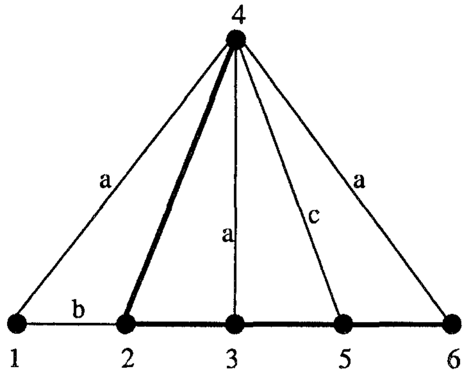
3.1 Problem Formulation

3.1.1 Formulation for Simpson Gear Train

Consider the stick diagram for the Simpson gear train shown in the Figure 3.1(a). The mechanism is symmetric with respect to the centerline and the lower half of the mechanism has been omitted from the drawing. Also in practise, usually more than one planetary gears are used. (Gears 2 and 5 are the planetary gears). Multiple planetary gears serve as parallel paths of power and increase the load capacity of a mechanism. However, the stick diagram



(a)



(b)

Gear	C_1	C_2	B_1	B_2
FIRST	X		X	
SECOND	X			X
THIRD	X	X		
REVERSE		X	X	

C - Input Clutch, B - Brake

(c)

Figure 3.1: Simpson gear train (a) functional schematic (b) graph no. 6506_1 (c) clutching sequence.

of the planetary gear train shows only one planetary gear, regardless of how many planetary gears are actually used in the design. The graph for the gear train is shown in Figure 3.1(b)[1] and the clutching sequence for the gear train is shown in the Figure 3.1(c). An 'X' mark in the clutching sequence indicates that the corresponding clutch has been applied. The Simpson gear train with the above clutching sequence is a very popular gear train, and can be found in Ford C3, Ford C5, Mercedes Benz, Toyota A40 and Nissan, to name a few. The rotational displacement equations for the graph representing the Simpson gear train are given as

$$[R][W] = [0] \tag{3.1}$$

where

$$R = \begin{bmatrix} r_{42}-1 & 1 & 0 & -r_{42} & 0 & 0 \\ r_{32}-1 & 1 & -r_{32} & 0 & 0 & 0 \\ 0 & 0 & -r_{35} & r_{35}-1 & 1 & 0 \\ 0 & 0 & 0 & r_{65}-1 & 1 & -r_{65} \end{bmatrix}$$

and

$$W = \begin{bmatrix} \omega_1, & \omega_2, & \omega_3, & \omega_4, & \omega_5, & \omega_6 \end{bmatrix}^T$$

As defined earlier $\mid r_{ji} \mid = t_j/t_i$, where t_j and t_i denote the number of teeth of the gear pair on links j and i, respectively. The quantity r_{ji} is either positive or negative according to whether the mesh is internal or external, and ω_i denotes the rotational speed of the link i. If the gear ratios are given, Eq. (3.1) represents a set of four simultaneous equations with six variables

---

[1] It can be shown that the graph of Figure 3.1(b) and the graph of Figure 1.4(f) are isomorphic.

$\omega_1 \ldots \omega_6$. If, two of these variables are specified from one of the rows of the clutching sequence shown in Figure 3.1(c), then we can solve for the remaining variables. That is, the rotational speeds of the remaining elements in the gear train are uniquely specified, including the output link. Note that the set of simultaneous equations to be solved depends on the two input variables, which in turn depend on clutch conditions. Assuming that the element clutched to the input link has a rotational speed of 1 rpm, and the elements for which a brake has been applied has a rotational speed of 0 rpm, then the rotational speed of the output element gives the reduction achieved by the gear train with the given clutch selection (or the reduction ratio). Usually, an reduction ratio is less than 1, but if the gear train functions as an overdrive it can be greater than 1. The reduction ratios can be expressed in terms of the gear ratios by solving the rotational displacement equations, using the clutching sequence table. We note that the output link shown in Figure 3.1(a) is to be connected to a final reduction gear set and then the differential gear in an automotive automatic transmission.

In practice, however, one wants to choose a set of gear ratios to achieve a set of reduction ratios. The conventional method has been to choose a set of gear ratios, solve for the reduction ratios based on the clutching sequence table, and then modify the gear ratios further in an iterative fashion until the required reduction ratios are achieved. At the end of this trial and error process the designer will not know whether the best possible reduction ratios with respect to the original requirements have been achieved.

To eliminate trial and error, we define an objective function F in terms of the reduction ratios R_k for the gear train, the minimization of which ensures that the best possible gear ratios are achieved, subject to certain constraints. Thus,

Minimize:

$$F(\mathbf{R}) \tag{3.2}$$

Subject to:

$$g_j(\mathbf{R}) \leq 0 \quad ; \quad j = 1, m \tag{3.3}$$

$$h_k(\mathbf{R}) = 0 \quad ; \quad k = 1, n \tag{3.4}$$

where g_j represent inequality constraints and h_k represent equality constraints. The objective function is

$$F(\mathbf{R}) = \sum_{i=1}^{4} (R_i - k_i)^2$$

where R_i is the ith reduction ratio parameter obtained by the minimization process and k_i is the ith desired reduction ratio. The reduction ratios $R_1 \ldots R_4$ depend on the clutch selection and the topology of the gear train. Solving the system of Eqs. (3.1) four times, each time for a different clutching sequence using the table shown in Figure 3.1(c), it can be shown that the reduction ratios are given as

$$R_1 = r_{65}r_{32}/(r_{65}r_{32} + (r_{42} - r_{32})r_{35}) \tag{3.5}$$

$$R_2 = r_{65}/(r_{65} - r_{35}) \tag{3.6}$$

$$R_3 = 1 \tag{3.7}$$

$$R_4 = r_{32}/r_{42} \tag{3.8}$$

Under certain clutch combinations the gear train may act as a rigid body, in which case the reduction ratio will be equal to one, and the designer will not be able to affect it by his selection of gear ratios. This is known as the direct drive and is chosen purposely to maximize the transmission efficiency. It should be realized that for an arbitrary combination of k_i there may not be a set of solutions for the gear ratios r_{ji} that satisfy Eqs. (3.5) to (3.8). In other words, the ideal minimum value of the cost function ($F(\mathbf{R}) \rightarrow 0$) may or may not be achievable.

The constraints under which the function $F(R)$ is minimized are now described. Let r_i, d_i, and t_i be the radius, diameter and number of teeth of gear element i. If the diametral pitch P of all the gears is the same, then $t_i = d_i/P$ and we have for the Simpson gear train $r_6 = r_3 + 2r_5$ or $t_6 = t_3 + 2t_5$, and, therefore $t_6/t_5 = t_3/t_5 + 2$. Thus,

$$r_{65} + r_{35} = 2 \tag{3.9}$$

and in a similar manner

$$r_{42} + r_{32} = 2 \tag{3.10}$$

Note that both r_{35} and r_{32} are negative numbers, since they correspond to external gear meshes.

A practical gear train cannot have too large or too small gear ratios. For the Simpson gear train we require the gear ratios to fall within the following limits :

$$\begin{array}{rcccr} 3.0 & \leq & r_{65} & \leq & 8.0 \\ -6.0 & \leq & r_{35} & \leq & -1.0 \\ 3.0 & \leq & r_{42} & \leq & 8.0 \\ -6.0 & \leq & r_{32} & \leq & -1.0 \end{array}$$

The sun gear is usually larger than the planet gear, and it accommodates the center shaft. Also, to avoid dynamic imbalance and to facilitate load sharing there are usually more than one planetary gears acting as parallel conduits of power transfer. Planet gears can thus be smaller; hence the limits on r_{65} and r_{35}. These limits are represented as variables in the optimization routine and can be changed if desired.

The optimization problem is now formally stated as:

Minimize

$$F(\mathbf{R}) = (R_1/k_1 - 1)^2 + (R_2/k_2 - 1)^2 + (R_3/k_3 - 1)^2 + (R_4/k_4 - 1)^2 \quad (3.11)$$

subject to the constraints

$$g_1 : \quad r_{65}/8.0 - 1 \quad (3.12)$$

$$g_2 : \quad 3.0/r_{65} - 1 \quad (3.13)$$

$$g_3 : \quad r_{35}/(-6.0) - 1 \quad (3.14)$$

$$g_4 : \quad -1.0/r_{35} - 1 \quad (3.15)$$

$$g_5 : \quad r_{42}/8.0 - 1 \quad (3.16)$$

$$g_6 : \quad 3.0/r_{42} - 1 \quad (3.17)$$

$$g_7: \quad r_{32}/(-6.0) - 1 \tag{3.18}$$

$$g_8: \quad -1.0/r_{32} - 1 \tag{3.19}$$

$$h_1: \quad (r_{65} + r_{35})/2 - 1 \tag{3.20}$$

$$h_2: \quad (r_{42} + r_{32})/2 - 1 \tag{3.21}$$

The quantities R_1, R_2, R_3, and R_4 are given in Eqs. (3.5)–(3.8). Note that all quantities have been scaled to have comparable magnitude before they are used, and that the inequality constraints with upper and lower limits have been split into two single-limit constraints.

3.1.2 Formulation for THM 440 Gear Train

The stick diagram for the General Motor's THM-440 gear train is shown in Figure 3.2(a). The graph for the gear train is shown in Figure 3.2(b) and the clutching sequence for the gear train is shown in the Figure 3.2(c).

The optimization problem can be similarly arrived at, as

Minimize

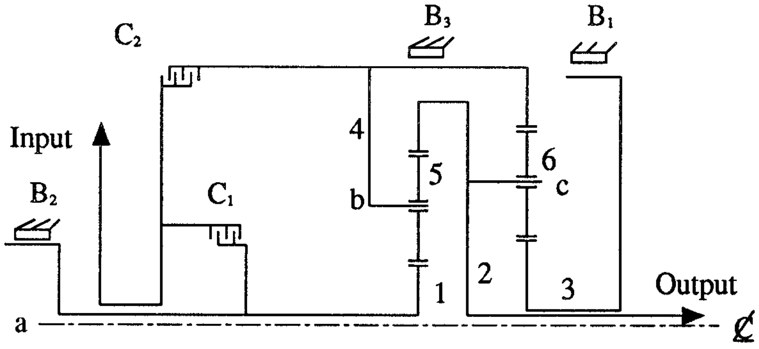
$$F(\mathbf{R}) = \sum_{i=1}^{5} (R_i/k_i - 1)^2 \tag{3.22}$$

where

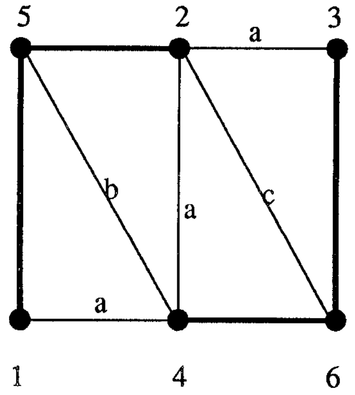
$$R_1 = r_{15}r_{46}/(r_{15}r_{46} + (r_{25} - r_{15})r_{36}) \tag{3.23}$$

$$R_2 = r_{46}/(r_{46} - r_{36}) \tag{3.24}$$

$$R_3 = 1 \tag{3.25}$$



(a)



(b)

	C_1	C_2	B_1	B_2	B_3
FIRST	X		X		
SECOND		X	X		
THIRD	X	X			
FOURTH		X		X	
REVERSE	X				X

C - Input Clutch, B - Brake

(c)

Figure 3.2: Hydra-Matic THM 440-T4 (a) functional schematic (b) graph 6206_3 (c) clutching sequence.

$$R_4 = 1 - r_{15}/r_{25} \tag{3.26}$$

$$R_5 = r_{15}/r_{25} \tag{3.27}$$

Replacing r_{65}, r_{35}, r_{42} and r_{32} by r_{25}, r_{15}, r_{46} and r_{36}, respectively, in the constraint Eqs. (3.12)–(3.21) gives the constraints equations for this problem.

3.2 Optimized Solution

The optimization problems formulated in the previous sections belong to a class of optimization problems where the objective function and/or the constraint functions are nonlinear. To solve this type of problem we will use one of the *Sequential Unconstrained Minimization Techniques* (SUMT). The general approach with SUMT is to minimize the objective function as an unconstrained function, while employing a penalty to limit constraint violations [28]. (Mathematically, the original objective function with constraints is converted to an equivalent pseudo-objective function without constraints.) Because of the way in which this penalty is imposed it often leads to numerically ill-conditioned problems. A method has been devised whereby only a moderate penalty is provided in the initial optimization stages, and this penalty is increased as the optimization progresses. This requires the solution of several unconstrained minimization problems to obtain the optimum constrained design. The algorithm for solving the unconstrained minimization problem is described in section 3.2.2.

The pseudo-objective function considered in SUMT is of the form

$$\Phi(\mathbf{R}, r_p) = F(\mathbf{R}) + r_p P(\mathbf{R})$$

where $F(\mathbf{R})$ is the original objective function and $P(\mathbf{R})$ is an imposed penalty function, the form of which depends on the SUMT being employed. The scalar r_p is a multiplier that determines the magnitude of the penalty, and is held constant for a complete unconstrained minimization. The subscript p is the unconstrained minimization number. A very extensive discussion on SUMT is provided by Fiacco and McCormick [9].

There are three important traditional approaches to minimize a constrained optimization problem using SUMT. The first, and the easiest one to incorporate into the design algorithm, is referred to as an exterior penalty function method because it penalizes the objective function only when the constraints are violated. An important observation to be made in this method is that as r_p is increased, the design approaches the true constrained optimum from the infeasible region, becoming feasible only in the limit as $r_p \to \infty$. This is often viewed as a disadvantage to this method. The second approach, known as the interior penalty function method, penalizes the objective function as the design approaches a constraint, but constraint violations are never allowed. The interior penalty function method has the advantage that it generates a sequence of improving feasible designs, but has the potential difficulty of having to deal with the discontinuity of $\Phi(\mathbf{R}, r_p)$ at the constraint boundaries. The third method, the extended interior penalty function method, incorporates the advantages of the first two methods.

3.2.1 The Augmented Lagrange Multiplier Method

While the classical SUMT methods are easy to program, the optimization process can be significantly improved by including the Lagrange multipliers into the algorithm. Indeed, Powell [14] notes that the use of SUMTs that do not include the lagrange multipliers is obsolete as a practical optimization tool. The method presented here is most often referred to as the Augmented Lagrange Multiplier(ALM) method [28], or simply as either the multiplier method or the primal-dual method. A general optimization problem with both equality and inequality constraints has the following pseudo-objective function in the ALM method

$$A(\mathbf{R}, \lambda, r_p) = F(\mathbf{R}) + \sum_{j=1}^{m}[\lambda_j \psi_j + r_p \psi_j^2] + \sum_{k=1}^{l}\{\lambda_{k+m} h_k(\mathbf{R}) + r_p[h_k(\mathbf{R})]^2\} \quad (3.28)$$

where

$$\psi_j = max\,[g_j(\mathbf{R}), -\lambda_j/2r_p]$$

The algorithm as described in [28] is given in Figure 3.3. The update formulas for the Lagrange multipliers are given as

$$\lambda_j^{p+1} = \lambda_j^p + 2r_p\left\{max\left[g_j(\mathbf{R}), -\lambda_j^p/2r_p\right]\right\} \quad j = 1, m$$

$$\lambda_{k+m}^{p+1} = \lambda_{k+m}^p + 2r_p h_k(\mathbf{R}) \quad k = 1, n$$

The update formula for r_p is given by

$$r_p^{p+1} = \gamma r_p^p$$

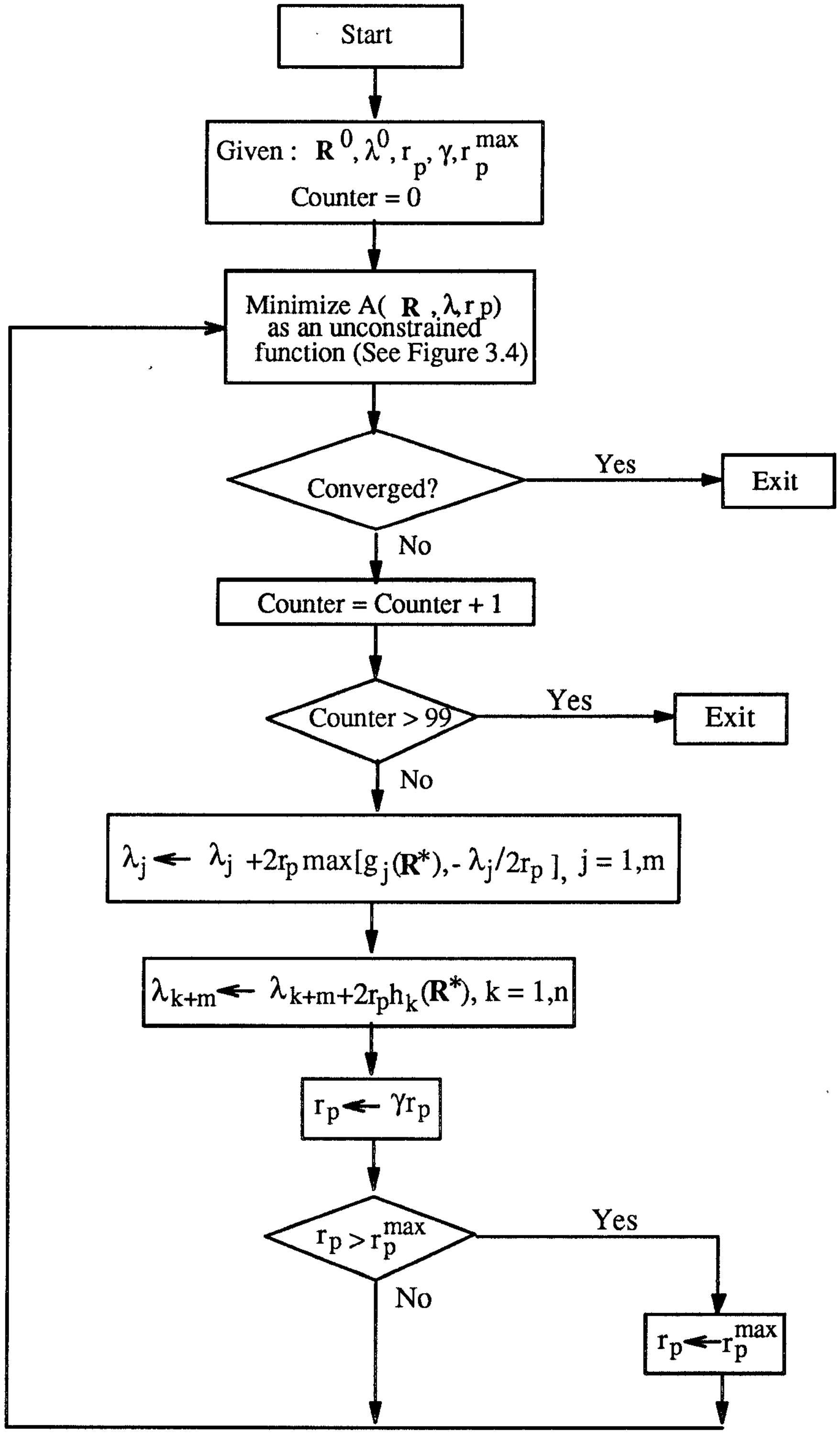

Start
Given : $\mathbf{R}^0, \lambda^0, r_p, \gamma, r_p^{max}$
Counter = 0
Minimize A(R , λ, r p) as an unconstrained function (See Figure 3.4)
Converged?
Yes
Exit
No
Counter = Counter + 1
Counter > 99
Yes
Exit
No
$\lambda_j \leftarrow \lambda_j + 2r_p \max[g_j(\mathbf{R}^*), -\lambda_j/2r_p],\ j = 1,m$
$\lambda_{k+m} \leftarrow \lambda_{k+m} + 2r_p h_k(\mathbf{R}^*),\ k = 1,n$
$r_p \leftarrow \gamma r_p$
$r_p > r_p^{max}$
Yes
No
$r_p \leftarrow r_p^{max}$

Figure 3.3: Algorithm for the augmented lagrangian method.

where $\gamma > 1$. However, r_p is limited a maximum of r_p^{max}.

The method has the following advantages.

1. It is relatively insensitive to the value of r_p. It is not necessary to increase r_p to ∞.
2. The precise satisfaction of $g(\mathbf{R}) = 0$ and $h_k(\mathbf{R})$ are possible.
3. Acceleration of convergence is achieved by updating the Lagrange multipliers.
4. The starting point may be either feasible or infeasible.
5. At the optimum, the value of $\lambda_j^* \neq 0$ will automatically identify the active constraint set.

The ALM method was used to solve the constrained optimization problems given by Eq. (3.11) and constraint Eqs. (3.12)–(3.21) for the Simpson gear train, and Eq. (3.22) and constraint Eqs. (3.12)–(3.21) for the THM 440 gear train. The number of iterations are limited to 99 for the ALM algorithm, to take take care of situations in which the algorithm is not able to converge because of constraint violations.

3.2.2 Conjugate Gradient Method

From Figure 3.3 and Eq. (3.28) it is seen that a means is needed to minimize the unconstrained optimization function that is generated every time in the iterative loop of the constrained optimization algorithm. The unconstrained optimization function can vary based on the current values of the design variables. For this function let $\mathbf{x}_{(k)}$ be the current iteration point for the design variables (for the Simpson gear train $\mathbf{x}_k = [x_{1(k)}, x_{2(k)}, x_{3(k)}, x_{4(k)}]^T$, where $x_{1(k)} = r_{65}$, $x_{2(k)} = r_{35}$, $x_{3(k)} = r_{42}$, and $x_{4(k)} = r_{32}$). Let $\mathbf{x}_{(k+1)}$ be the improved design point that is related to $\mathbf{x}_{(k)}$ by the relation $\mathbf{x}_{(k+1)} = \mathbf{x}_{(k)} + \triangle\mathbf{x}_{(k)}$. The change $\triangle\mathbf{x}_{(k)}$ can be expressed as

$$\triangle\, \mathbf{x}_{(k)} = \alpha_k \mathbf{d}_{(k)} \tag{3.29}$$

where $\mathbf{d}_{(k)}$ is a *desirable search direction* in the design space and α_k is a positive scalar called the step size in that direction. Thus the process of computing $\triangle\mathbf{x}_{(k)}$ involves two separate problems: finding the direction and determining the step length (scaling along the direction). Unconstrained optimization problems vary in the way the descent direction $\mathbf{d}_{(k)}$ is found, and several algorithms exist [8, 18, 28]. To solve the unconstrained minimization problem the conjugate gradient method [7] shown in Figure 3.4 is used. Assuming that the desirable direction of design change $\mathbf{d}_{(k)}$ has been found, the cost function at the iteration number $(k+1)$ is then given as

$$f(\mathbf{x}_{(k+1)}) = f(\mathbf{x}_{(k)} + \alpha_k \mathbf{d}_{(k)}) \tag{3.30}$$

Since now both $\mathbf{x}_{(k)}$ (the current design point) and $\mathbf{d}_{(k)}$ are known, $f(\mathbf{x}_{(k+1)})$ is a function of α_k only. The problem is reduced to a one-dimensional minimization problem of finding $\alpha* = \alpha_k$ such that $f(\alpha)$ is minimized. The algorithm

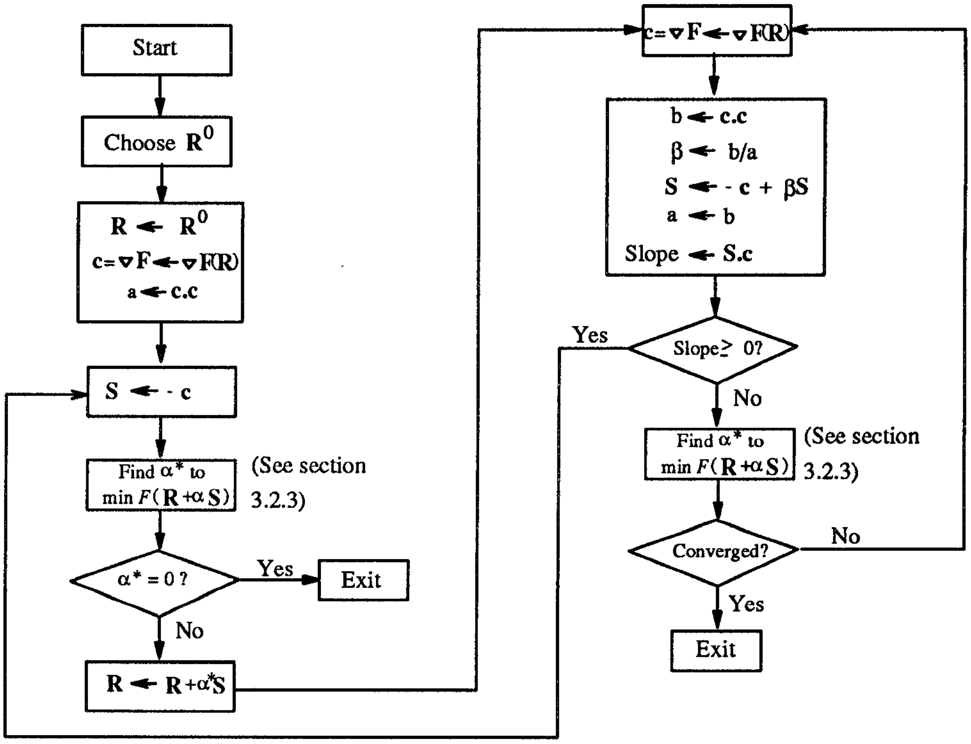


Figure 3.4: Algorithm for the Fletcher-Reeves conjugate direction method.

that is used to minimize the one-dimensional minimization problem, every time $\mathbf{d}_{(k)}$ is computed, is described in the next section.

3.2.3 Golden Section Search

Once the descent direction has been found using the conjugate gradient method, the *Golden Section Search* [8] is used to find $\alpha*$, which is the optimum α for every iteration of the unconstrained optimization. The basic idea of any one-dimensional search technique is to reduce successively the interval of uncertainty in which the minimum lies to a small acceptable value. The algorithm for the golden section search is as follows:

1. For a chosen small step size δ in α, let q be the smallest integer to satisfy

   $$f(\alpha_{q-1}) < f(\alpha_{q-2})$$

   and

   $$f(\alpha_{q-1}) < f(\alpha_q)$$

   where

   $$\alpha_q = \textstyle\sum_{j=0}^{q} \delta(1.618)^j; \quad q = 0, 1, 2, ...$$

   The upper and lower bounds on α^* are given, respectively, by

   $$\alpha_u \equiv \alpha_q = \sum_{j=0}^{q} \delta(1.618)^j$$

   $$\alpha_l \equiv \alpha_{q-2} = \sum_{j=0}^{q-2} \delta(1.618)^j$$

2. Compute $f(\alpha_a)$ and $f(\alpha_b)$, where $\alpha_a = \alpha_l + 0.382I$ and $\alpha_b = \alpha_l + 0.618I$, and $I = \alpha_u - \alpha_l$ is the interval of uncertainty.

3. Compare $f(\alpha_a)$ and $f(\alpha_b)$.

   (a) If $f(\alpha_a) < f(\alpha_b)$ then the optimum lies between α_l and α_b. The new limits for the reduced interval of uncertainty are $\alpha'_l = \alpha_l$ and $\alpha'_u = \alpha_b$. Let $\alpha'_b = \alpha_a$, $\alpha'_a = \alpha'_l + 0.382(\alpha'_u - \alpha'_l)$, compute $f(\alpha'_a)$ and go to step 4.

   (b) If $f(\alpha_a) > f(\alpha_b)$ then the optimum lies between α_a and α_u. The new limits for the reduced interval of uncertainty are $\alpha'_l = \alpha_a$ and $\alpha'_u = \alpha_u$. Let $\alpha'_a = \alpha_b$, $\alpha'_b = \alpha'_l + 0.618(\alpha'_u - \alpha'_l$, compute $f(\alpha'_b)$ and go to step 4.

(c) If $f(\alpha_a) = f(\alpha_b)$, let $\alpha_l = \alpha_a$ and $\alpha_u = \alpha_b$ and go to step 2.

4. If the new interval of uncertainty satisfies the convergence criterion, let $\alpha^* = (\alpha'_u + \alpha'_l)/2$ and stop. Otherwise, delete primes on α'_l, etc. and return to step 3.

3.3 Results and Discussion

The optimization program was written in C. The designer initially specifies the gear train type and stipulates the reduction ratios to be achieved. The optimization program starts at an arbitrary design point and at the end of its execution displays the optimized gear ratios. Results for sample inputs for the Simpson and THM-440 gear trains are shown in Tables 3.1 and 3.2.

To verify the procedure the results of Tsai [23] were used. Tsai has developed an algorithm for the automatic generation of the rotational displacement equations, if the adjacency matrices are given, and has demonstrated his algorithm for the angular velocity analysis of an EGT with any number of links. He started with a set of gear ratios for the Simpson gear train (actually, he started with the gear teeth numbers) and arrived at the reduction ratios and the angular velocity of other links in the gear train. Set # 1 of the Simpson gear train given in the Table 3.1 starts with the reduction ratios obtained by Tsai. As shown in Table 3.1 the optimization program obtains the same gear

Table 3.1: Table of optimized gear ratios for sample inputs for a Simpson gear train.

Set #	Desired Overall Reduction Ratio		Achieved Overall Reduction Ratio		F(**R**)	Optimized Gear Ratios	
1	k_1	0.3521	R_1	0.352189	1.68×10^{-7}	r_{65}	4.996
	k_2	0.6250	R_2	0.625115		r_{35}	2.996
	k_3	1.0000	R_3	1.000000		r_{42}	3.874
	k_4	-0.4839	R_4	-0.483897		r_{32}	1.874
2	k_1	0.3000	R_1	0.326227	4.92×10^{-2}	r_{65}	6.011
	k_2	0.7500	R_2	0.599792		r_{35}	4.011
	k_3	1.0000	R_3	1.000000		r_{42}	3.826
	k_4	-0.5000	R_4	-0.477251		r_{32}	1.826
3	k_1	0.4000	R_1	0.400004	3.76×10^{-8}	r_{65}	3.499
	k_2	0.7000	R_2	0.700006		r_{35}	1.499
	k_3	1.0000	R_3	1.000000		r_{42}	3.333
	k_4	-0.4000	R_4	-0.399994		r_{32}	1.333

ratios that was the input for Tsai's procedure.

It has been mentioned in Section 3.1 that Eqs. (3.5) to (3.8) for the Simpson gear train, and Eqs. (3.23) to (3.27) for the THM 440 gear train, have been obtained from their rotational displacement equations and clutching sequence tables. These equations are directly supplied to the optimization program. For each gear train, instead, it is possible to supply its rotational displacement equations (for the Simpson gear train they are given by Eq. (3.1)), the clutching sequence table and to specify the output link to the program and then have the program calculate the reduction ratios. Every time the value of $F(\mathbf{R})$ has to be calculated in the algorithm, the program would determine the matrix $[R]$ based on the current gear ratios and calculate the reduction

Table 3.2: Table of optimized gear ratios for sample inputs for a THM 440 gear train.

Run #	Desired Overall Reduction Ratio		Achieved Overall Reduction Ratio		F(**R**)	Optimized Gear Ratios	
1	k_1	0.3425	R_1	0.343011	1.00×10^{-5}	r_{25}	3.498
	k_2	0.6269	R_2	0.635171		r_{15}	1.498
	k_3	1.0000	R_3	1.000000		r_{46}	4.699
	k_4	1.4286	R_4	1.428328		r_{36}	2.699
	k_5	-0.4286	R_5	-0.428328			
2	k_1	0.4000	R_1	0.400547	4.16×10^{-3}	r_{25}	3.369
	k_2	0.7000	R_2	0.698090		r_{15}	1.369
	k_3	1.0000	R_3	1.000000		r_{46}	3.524
	k_4	1.5000	R_4	1.406424		r_{36}	1.524
	k_5	-0.5000	R_5	-0.406424			

ratios using the rotational displacement equations. Thus, the reduction ratios $R_1 \ldots R_4$ are found by solving the system of Eqs. 3.1 four times, each time for a different clutching sequence, shown in Figure 3.1(c), while optimizing the Simpson gear train. The optimization program has been extended to take rotational displacement equations, a clutching sequence table, and an output link number. This means that the optimization problem can be directly applied to all variations of the Simpson and THM 440 gear trains that differ only in their clutching sequence tables and/or clutch numbers. However, a set of simultaneous equations has to be solved several times (as many times as the number of reduction ratios), each time the program evaluates the error function $F(\mathbf{R})$. (The number of times the value of the error function is calculated is of the order of 10,000). This is at the cost of increasing the computational time. On a Sparc station this modification has resulted in an increase of total run time from less than a minute to around 7 minutes. Tsai's [23] algorithm can be incorporated as a further extension to determine the rotational displacement equations automatically when the adjacency matrices are supplied.

The design is far from complete at this stage. Once the gear ratios are obtained, further steps in the design's analysis are needed to determine the gear thickness, material, teeth number, the shaft diameters, etc., based on the power transmitted.

To relate the gear ratios to the gear teeth numbers, a program was written to generate a list of gear ratios for all combinations of gear teeth from 14 to 120. These ratios have been sorted and are used to determine the sets of gear teeth numbers that would give gear ratios to within ±1% of the optimized gear ratios. (The value of 1% was based on a sensitivity analysis that found the effect on the reduction ratios due to a perturbation of the gear ratios for the Simpson and the THM 440 gear trains considered).

In the Simpson gear train, r_{42} and r_{32} have element 2 as a common gear, and r_{35} and r_{65} have element 5 as a common gear. Similarly, for THM 440 gear train r_{25} and r_{15} have element 5 as a common gear and r_{46} and r_{36} have element 6 as a common gear. Also, the sum of the number of teeth of the sun gear and twice the number of teeth on the planet gear should equal the number of teeth on the ring gear. This is equivalent to all the constraints of the type specified by Eq. (3.20). The sets of gear teeth now are narrowed to

Table 3.3: Gear teeth values for obtaining optimized ratios r_{65}, r_{35}, r_{42}, and r_{32} within ±1%.

t_6	t_5	r_{65}	t_3	t_5	r_{35}	t_4	t_2	r_{42}	t_3	t_2	r_{32}
70	14	5.000	42	14	-3.000	66	17	3.882	32	17	-1.882
80	16	5.000	48	16	-3.000	62	16	3.875	30	16	-1.875
100	20	5.000	60	20	-3.000	58	15	3.867	28	15	-1.867
120	24	5.000	72	24	-3.000						
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

the set that can accommodate the same number of gear teeth for the common gear and can satisfy the geometry constraint given by Eq. (3.20). Carrying out this search to obtain the first set of optimized gear ratios in Table 3.1 for the Simpson gear train yields the following:

Tsai [23] started with the gear teeth number set $t_6 = 70$, $t_5 = 14$, $t_3 = 42$, $t_4 = 62$, $t_2 = 16$, and $t_3' = 30$. (Element 3 for the simpson gear train is a multiple gear). Referring to Table 3.3 it can be seen that other combinations are possible to give the reduction ratios to be exactly equal or very close to the values specified by set # 1 in Table 3.1.

The approach used to find the teeth number of the gears must be weighed against the changes that may have to be made during the gear dimensional analysis based on transmitted power. Reduction ratios that can be achieved may have to be modified with respect to the optimized values of the gear and reduction ratios. Hence, an iterative procedure between the values given in Table 3.3 and those required to satisfy the transmitted power may have to be developed in order for the process to be fully automated.

Chapter 4

User Interface for Developing EGTs

The trend for friendly user interfaces (UIs) for any computer package or utility has been unmistakably set. Almost everyone is familiar with the user friendly menus, pop-up windows, buttons, etc., that lead one through the application in the Apple-Macintosh systems. Most of the user's actions are performed in these applications by the 'click' and 'drag' options of a mouse button. Though Macintosh seems to have popularized user friendly interfaces, pioneering efforts in researching and developing the concept of visual or graphical user interfaces for the computer industry was introduced by Xerox. Such interfaces are now commonplace in different computer system applications; e.g., DEC Windows for VAX/VMS systems, MS Windows for IBM PCs, X-Windows in Sun systems, etc.

These interfaces are very easy to use. Even novices can find their way through difficult programs using 'Help' options or 'Messages' that seem to be appearing and disappearing at the right time. However, writing a simple and easy UI is rarely either simple or easy. It requires knowledge of the program's operating environment and the ability to grind out hundreds of lines of code.

The objective here was to develop an interface on the Sun system that would help a user access the optimization and functional representation display routines written for this EGT design environment (see Chapters 2 and 3). Several *widget sets* like Motif, Athena, Xview etc. are available that can be used on a Sun platform to develop an UI. *Widget sets* are used to manipulate (create, display, destroy, change properties, etc.) interface elements like a button or a text scrolling window. Widget sets are linked together to create UIs. This means that the way an interface element behaves should be decided before the programming code is written for that element. A small modification, like just changing the name of a button, in an interface element often mean a lot of changes to the code.

Very recently some application programs have started becoming available that help UI developers by providing high level function calls to create UI elements without the drudgery of widget set programming. The source code is generated by these application programs using one of the widget sets. The UI elements at this stage do not perform any actions. Once the code for the UI elements has been created, to perform specific actions it is merged with the code for which the UI is being developed, in order to perform specific actions. Application programs that help to create UI elements vary in the number of features that they can provide. Some of them create only the elements, and the linking of the various elements has to be done by manual coding. Others provide the option of specifying connections amongst the various elements. Once the code is created it is then compiled in the usual way. Compilation is often done with the help of 'make'[1] files to tackle the large number of files usually associated with a UI and its functionality.

We have used an interface development tool called OpenWindows Developer's Guide [15, 16] for creating the UI. The Developer's Guide (Devguide) operates in the OpenWindows application environment, a graphical user interface environment that operates on Sun workstations and uses the Xview widget set. Devguide is used to create UIs that conform to the OPENLOOK User Interface as it is implemented in OpenWindows.

The word "Guide" in the full name "OpenWindows Developer's Guide" is an acronym for Graphical User Interface Design Editor. Devguide can be used to assemble the elements of a UI by dragging the visual representation of the elements onto the workspace of the monitor screen and putting the elements together. Windows, control areas, panes, buttons menus, sliders and other OPENLOOK UI elements can be assembled in the UI being created and their properties modified. The elements can also be tested for their behavior while still in Devguide.

---

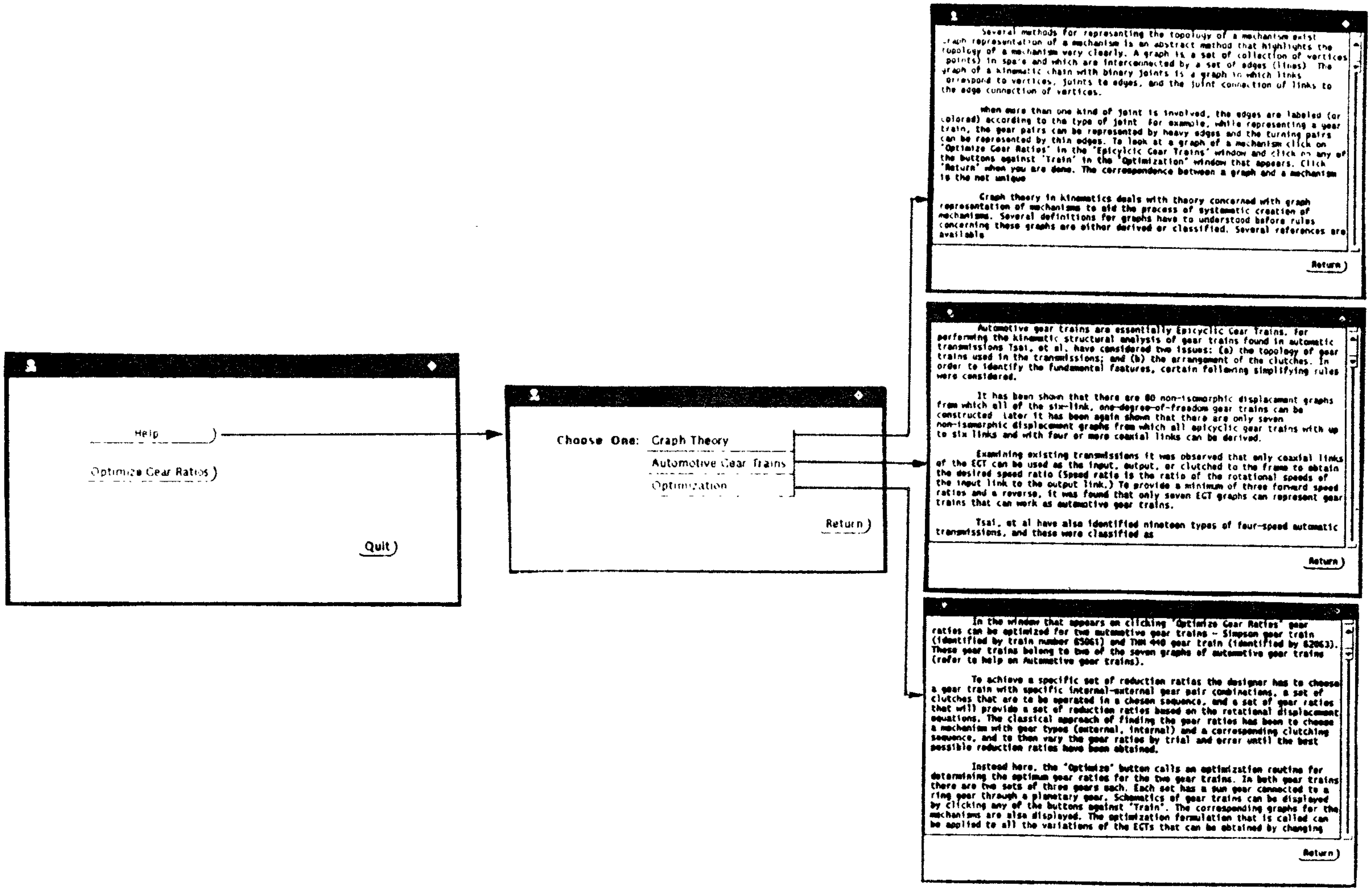
[1] For both the programmer and the computer, it is inefficient and costly to keep a moderate or large size program in one file that has to be recompiled repeatedly during program development. A much better strategy is to write the program in multiple files. The *make* [1] utility is then used to keep track of the source files and to provide convenient access to libraries and associated header files. It is a powerful utility always available on UNIX and often available in MS-DOS, where it is an add-on feature.

The process of creating UI elements using Devguide is not described here. Reference should be made to the OpenWindows Developer's Guide manual. This manual isn't for novice Sun users: it assumes that the interface developer has worked with Sun workstations and OpenWindows. The reader should be familiar with UNIX and SunOS, OpenWindows, and the OPENLOOK UI [20, 21, 22]. Furthermore, Xview and C programming are required to link the Devguide interfaces to application software [1, 4, 10].

4.1 Results and Discussion

UI windows developed for the present work are illustrated in Figures 4.1 to 4.3. The description of the working of the entire program is given in Appendix A. This section describes the functionality of the various UI elements in the windows of the Figures 4.1 to 4.3.

The initial execution of the program displays the left window shown in Figure 4.1. Clicking (with a mouse button) on the 'Quit' button exits the program. Clicking on the 'Help' button pops the middle 'Help' popup window shown in Figure 4.1 and clicking the 'Optimize Gear Ratios' button shows the bottom 'Optimize Gear Ratios' window shown in Figure 4.2(a). The 'Return' buttons in these and other windows hide the windows in which they are located. The user has three options in the help window (Figure 4.1). Clicking any of the help buttons display the corresponding help text windows as shown in the right of Figure 4.1. The help text windows are scrolling windows that can display



Several methods for representing the topology of a mechanism exist graph representation of a mechanism is an abstract method that highlights the topology of a mechanism very clearly. A graph is a set of collection of vertices points) in space and which are interconnected by a set of edges (lines) The graph of a kinematic chain with binary joints is a graph in which links orrespond to vertices, joints to edges, and the joint connection of links to the edge connection of vertices.

When more than one kind of joint is involved, the edges are labeled (or colored) according to the type of joint For example, while representing a gear train, the gear pairs can be represented by heavy edges and the turning pairs can be represented by thin edges. To look at a graph of a mechanism click on 'Optimize Gear Ratios' in the 'Epicyclic Gear Trains' window and click on any of the buttons against 'Train' in the 'Optimization' window that appears. Click 'Return' when you are done. The correspondence between a graph and a mechanism is the not unique

Graph theory in kinematics deals with theory concerned with graph representation of mechanisms to aid the process of systematic creation of mechanisms. Several definitions for graphs have to understood before rules concerning these graphs are either derived or classified. Several references are available

Automotive gear trains are essentially Epicyclic Gear Trains. For performing the kinematic structural analysis of gear trains found in automatic transmissions Tsai, et al. have considered two issues: (a) the topology of gear trains used in the transmissions; and (b) the arrangement of the clutches. In order to identify the fundamental features, certain following simplifying rules were considered.

It has been shown that there are 80 non-isomorphic displacement graphs from which all of the six-link, one-degree-of-freedom gear trains can be constructed later it has been again shown that there are only seven non-isomorphic displacement graphs from which all epicyclic gear trains with up to six links and with four or more coaxial links can be derived.

Examining existing transmissions it was observed that only coaxial links of the EGT can be used as the input, output, or clutched to the frame to obtain the desired speed ratio (Speed ratio is the ratio of the rotational speeds of the input link to the output link.) To provide a minimum of three forward speed ratios and a reverse, it was found that only seven EGT graphs can represent gear trains that can work as automotive gear trains.

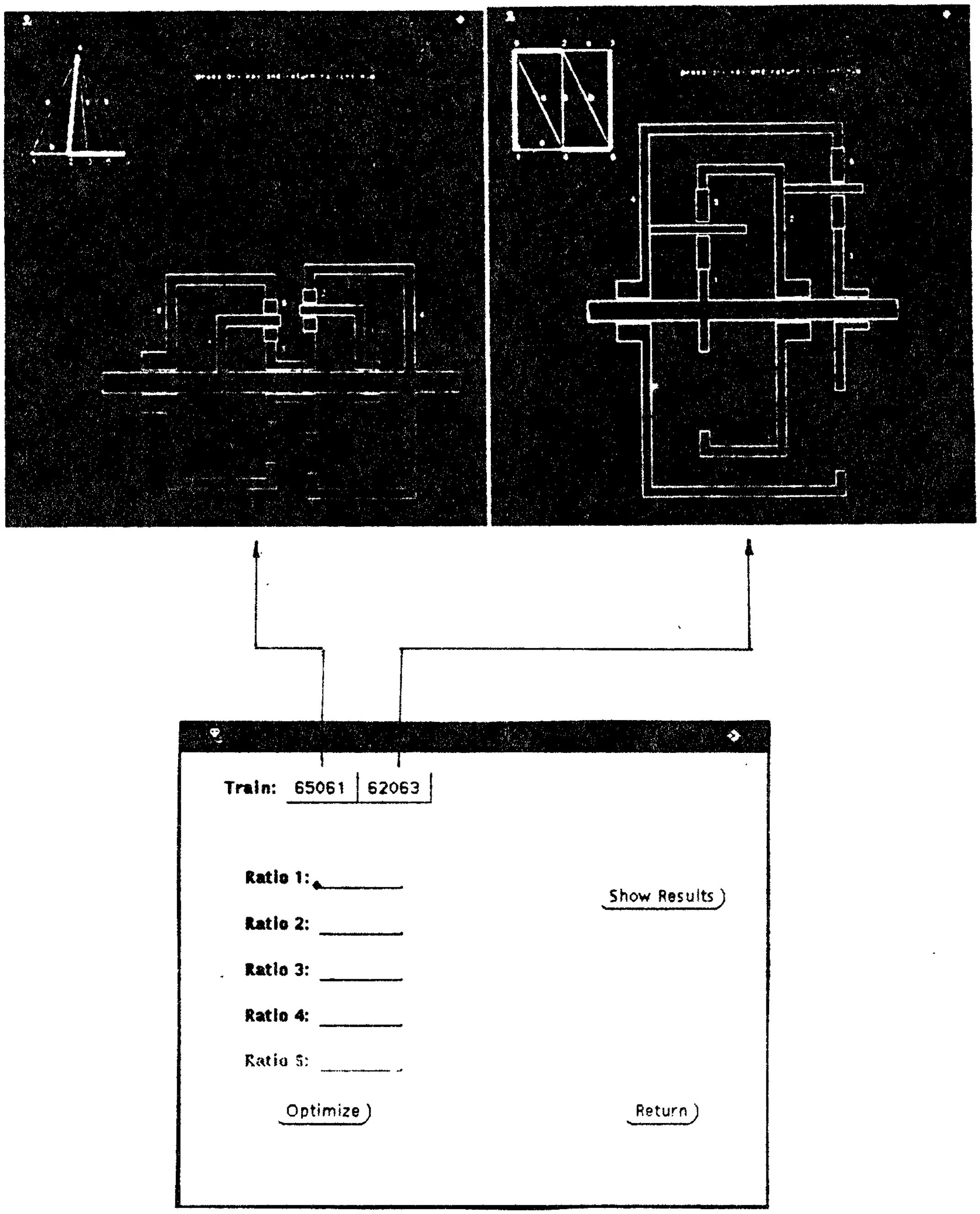
Tsai, et al have also identified nineteen types of four-speed automatic transmissions, and these were classified as

In the window that appears on clicking 'Optimize Gear Ratios' gear ratios can be optimized for two automotive gear trains - Simpson gear train (identified by train number 65061) and THM 440 gear train (identified by 62063). These gear trains belong to two of the seven graphs of automotive gear trains (refer to help on Automotive gear trains).

To achieve a specific set of reduction ratios the designer has to choose a gear train with specific internal-external gear pair combinations, a set of clutches that are to be operated in a chosen sequence, and a set of gear ratios that will provide a set of reduction ratios based on the rotational displacement equations. The classical approach of finding the gear ratios has been to choose a mechanism with gear types (external, internal) and a corresponding clutching sequence, and to then vary the gear ratios by trial and error until the best possible reduction ratios have been obtained.

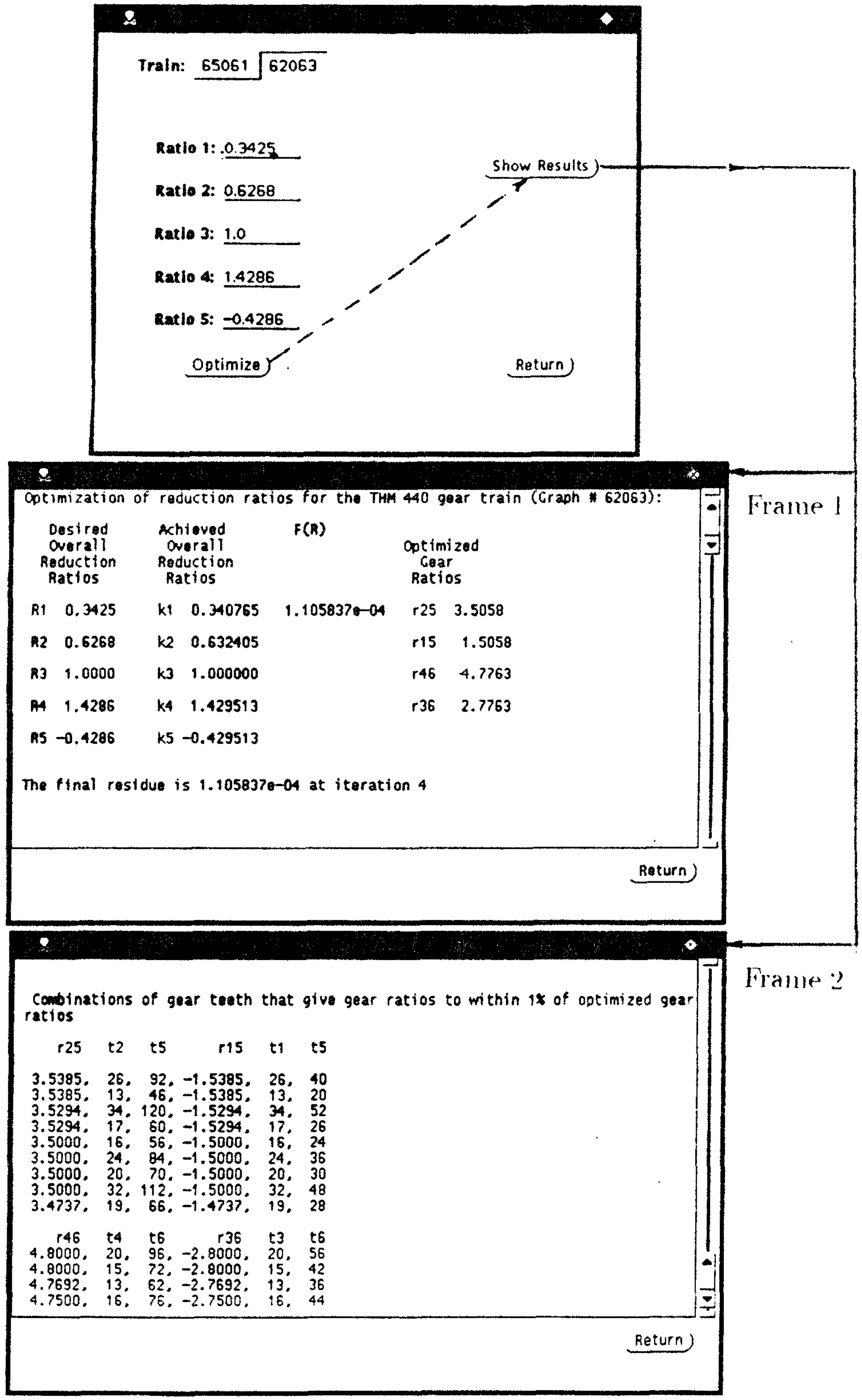
Instead here, the 'Optimize' button calls an optimization routine for determining the optimum gear ratios for the two gear trains. In both gear trains there are two sets of three gears each. Each set has a sun gear connected to a ring gear through a planetary gear. Schematics of gear trains can be displayed by clicking any of the buttons against 'Train'. The corresponding graphs for the mechanisms are also displayed. The optimization formulation that is called can be applied to all the variations of the EGTs that can be obtained by changing

Figure 4.1: UI windows to access help files



Train: 65061 | 62063

Figure 4.2: UI windows to display gear trains



Train: 65061 | 62063

Ratio 1: 0.3425

Ratio 2: 0.6268

Ratio 3: 1.0

Ratio 4: 1.4286

Ratio 5: -0.4286

```
Optimization of reduction ratios for the THM 440 gear train (Graph # 62063):

  Desired       Achieved        F(R)
  Overall        Overall                   Optimized
 Reduction     Reduction                     Gear
  Ratios         Ratios                     Ratios

 R1   0.3425    k1   0.340765  1.105837e-04  r25  3.5058

 R2   0.6268    k2   0.632405                 r15   1.5058

 R3   1.0000    k3   1.000000                 r46   4.7763

 R4   1.4286    k4   1.429513                 r36   2.7763

 R5  -0.4286    k5  -0.429513

The final residue is 1.105837e-04 at iteration 4
```

```
 Combinations of gear teeth that give gear ratios to within 1% of optimized gear
ratios

    r25    t2   t5       r15    t1   t5

 3.5385,  26,  92, -1.5385,  26,  40
 3.5385,  13,  46, -1.5385,  13,  20
 3.5294,  34, 120, -1.5294,  34,  52
 3.5294,  17,  60, -1.5294,  17,  26
 3.5000,  16,  56, -1.5000,  16,  24
 3.5000,  24,  84, -1.5000,  24,  36
 3.5000,  20,  70, -1.5000,  20,  30
 3.5000,  32, 112, -1.5000,  32,  48
 3.4737,  19,  66, -1.4737,  19,  28

    r46    t4   t6       r36    t3   t6
 4.8000,  20,  96, -2.8000,  20,  56
 4.8000,  15,  72, -2.8000,  15,  42
 4.7692,  13,  62, -2.7692,  13,  36
 4.7500,  16,  76, -2.7500,  16,  44
```

Figure 4.3: UI windows to optimize gear ratios and display results

large text files. In the ‘Optimize Gear Ratios’ window (Figure 4.2), the user has the option to choose either the Simpson or the THM-440 gear trains. Selecting any one of them displays the corresponding functional representations through a PHIGS window. These windows are shown in the top of Figure 4.2. The ‘Return’ button in the bottom window of Figure 4.2 also closes any PHIGS window if they are open. For optimization, the reduction ratios required are entered in the spaces provided, after which a clicking of the ‘Optimize’ button optimizes the gear ratios for the gear train selected. Furthermore, it also finds the sets of gear teeth as described in Section 3.3. Clicking the ‘Show Results’ displays the results in a text scrolling window (shown in two frames of Figure 4.3).

Chapter 5

Conclusions and Suggestions for Future Work

5.1 Conclusions

In this thesis we have concentrated on three issues related to automotive epicyclic gear trains.

First, a procedure for the functional and graph representation of EGT mechanisms on a computer screen using PHIGS has been developed and demonstrated for two existing automotive EGTs. A limited number of element modules (external gear, internal gear, multiple gear, carrier, and shaft) have been developed to display the EGTs. Virtually all mechanisms that are comprised of these elements in various combinations can be displayed using these modules. To broaden its application to other mechanisms comprising different elements, additional modules can be incorporated. There is no limitation to the nature or complexity of the element that can be displayed.

Second, the traditional trial and error approach of finding the gear ratios for an automotive gear train to obtain a set of reduction ratios has been eliminated by formulating the task as a constrained nonlinear optimization problem.

The optimization procedure was successfully applied to two automotive gear trains: the Simpson gear train and the THM-440 gear train. The optimization procedure finds those gear ratios for which the reduction ratios are closest to the original specification. Furthermore, the corresponding combination of gear teeth numbers that can satisfy these gear ratios to within 1% are also displayed. These gear teeth numbers also satisfy the geometric constraints.

Third, a user friendly user windows environment was developed to access the optimization results and EGT display. Several examples were given.

5.2 Future work

- Apply the optimization method to different clutching sequence tables and determine alternate ways of using these gear trains.
- Develop a methodology for the determination of all possible clutching sequences.
- Generalize the optimization procedure to make it applicable to other gear trains, such as the one that has an idler gear between the sun and planetary gear. See Appendix B.
- Perform the dimensional synthesis of mechanisms once the gear ratios have been found. Automation of dimensional synthesis could be based on issues like the power transmitted (force and torque analysis), material considerations, dimensional limitations, gear type, teeth number, etc.

Appendix A

Program Description

A.1 Program outline

The flow chart for using the program is described in Figures A.1, A.2, and A.3. Only the salient features of the program are described here. Some boxes in these figures are numbered to refer to the comments made here. When the program is executed the main window mentioned in Box # 1 of Figure A.1 is displayed on the screen. The window contain buttons and other user interface features that lead a user through the various options as shown in Figure A.1. Box # 2 of the flowchart is expanded into the flow chart shown in Figure A.2

The main data file specified by Box # 1 of the Figure A.2 provides the information on the number of elements. Following this is the data for each element—link number, element identity and the name of the file that contains the data for drawing the element. For each element a structure is created in PHIGS. Note that a link in the mechanism can comprise of more than one element. For example, Link # 4 of the Simpson gear train in Figure 2.6 is composed of a carrier, an internal gear and a shaft. A sample listing of this

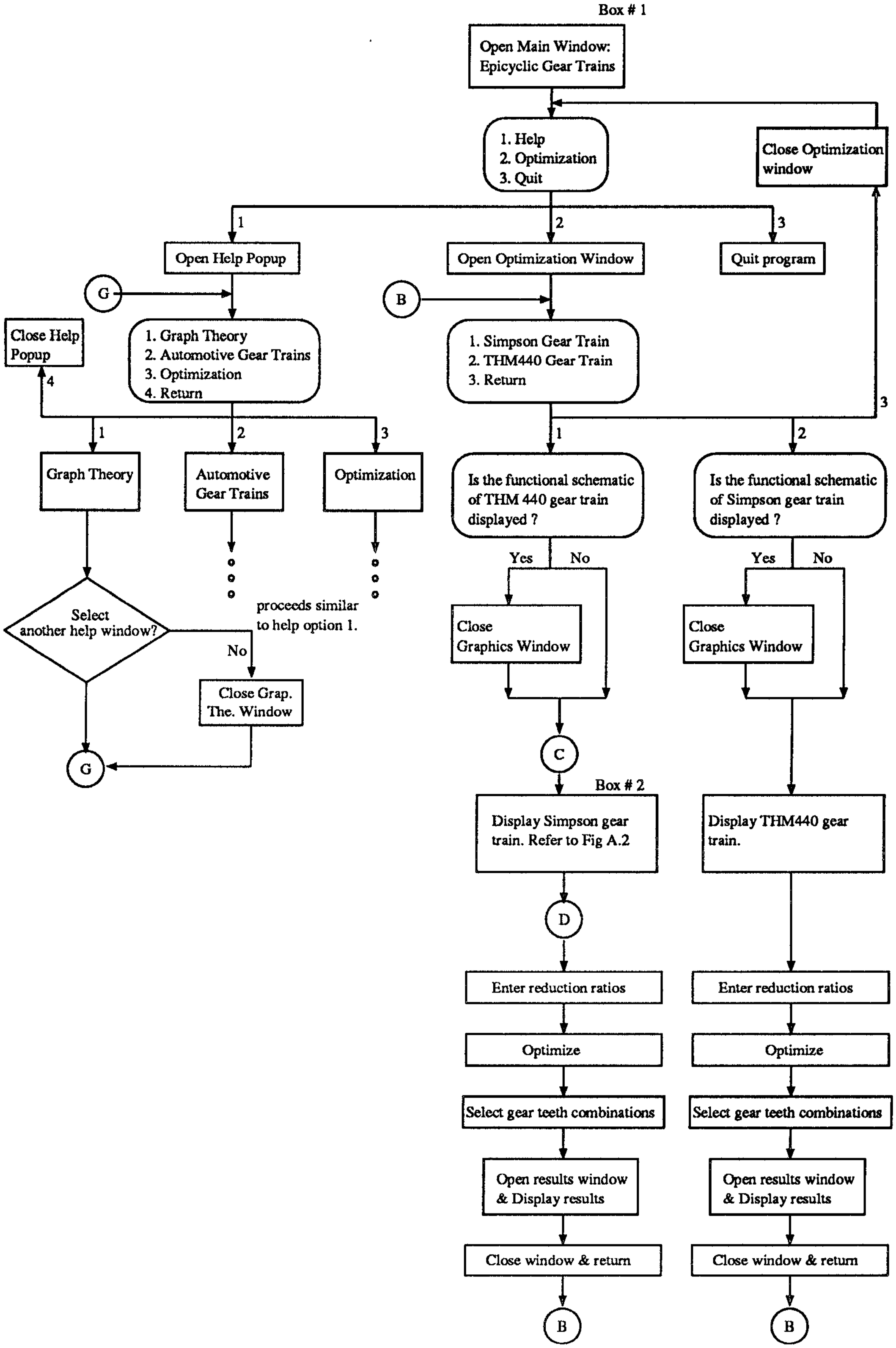

Box # 1
Open Main Window: Epicyclic Gear Trains
1. Help
2. Optimization
3. Quit
Close Optimization window
1
2
3
Open Help Popup
Open Optimization Window
Quit program
G
B
Close Help Popup
4
1. Graph Theory
2. Automotive Gear Trains
3. Optimization
4. Return
1. Simpson Gear Train
2. THM440 Gear Train
3. Return
1
2
3
Graph Theory
Automotive Gear Trains
Optimization
1
2
3
Is the functional schematic of THM 440 gear train displayed ?
Is the functional schematic of Simpson gear train displayed ?
Yes
No
Yes
No
Select another help window?
proceeds similar to help option 1.
No
Close Graphics Window
Close Graphics Window
Close Grap. The. Window
G
C
Box # 2
Display Simpson gear train. Refer to Fig A.2
Display THM440 gear train.
D
Enter reduction ratios
Enter reduction ratios
Optimize
Optimize
Select gear teeth combinations
Select gear teeth combinations
Open results window & Display results
Open results window & Display results
Close window & return
Close window & return
B
B

Figure A.1: Flow chart for the display and optimization program - Part I

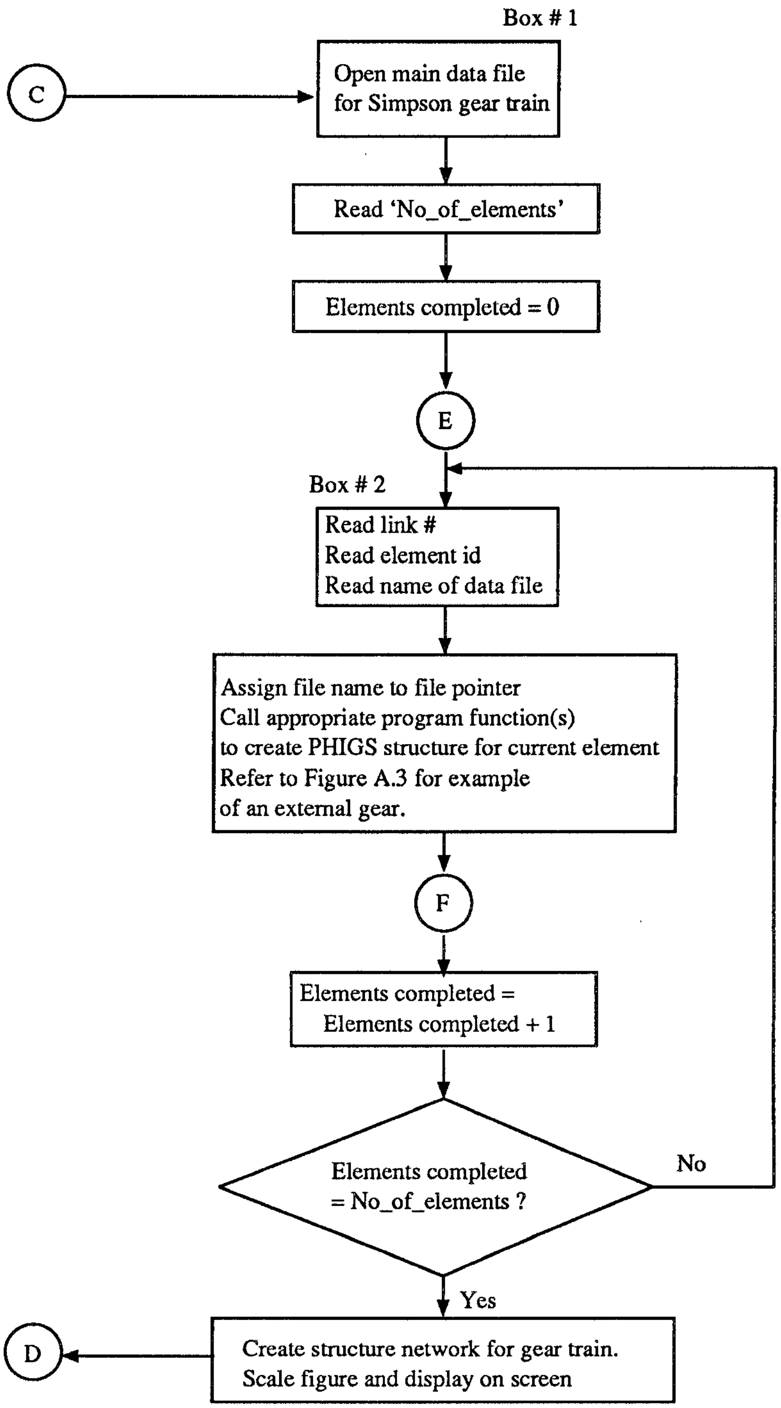


Figure A.2: Flow chart for the display and optimization program - Part II

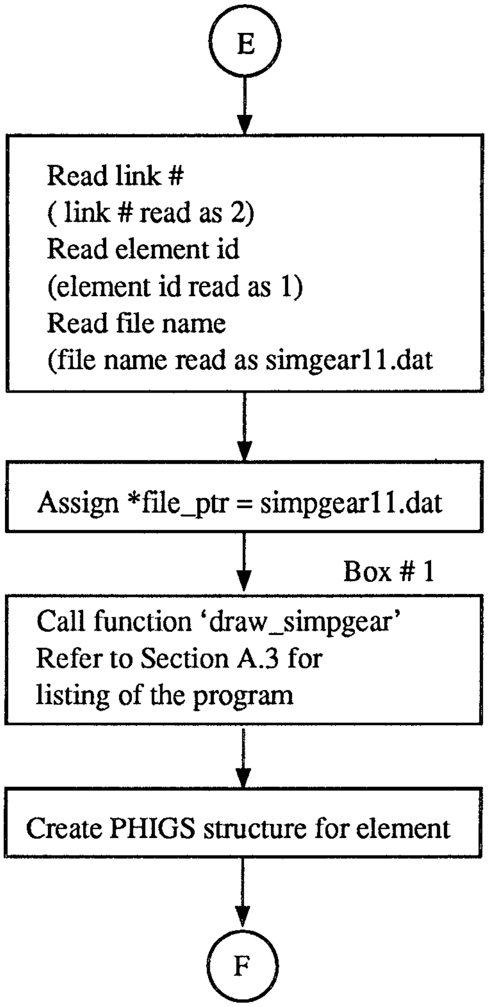


Figure A.3: Flow chart for the display and optimization program - Part III

data file is given in in Section A.2.

Box # 2 of Figure A.2, which pertains to the creation of the PHIGS structure of each element, is expanded in the flow chart shown in Figure A.3 for the specific case for the creation of a structure for an external gear. The data being read in the example are for the second element of the listing in Section A.2. The program listing for the function (subroutine) called in Box # 1 of Figure A.3 is given in Section A.3. The function that is called has the name 'draw_simpgear'. This function, in turn, calls the function 'find_pos_simpgear', which is also listed in Section A.3. Comments are included in the listing. The example data file that is read by this function for the PHIGS structure creation of this external gear is given in Section A.3.1.

A.2 Listing for main data file for Simpson gear train

```
8
1 4
carrier1_22.dat
2 1
simpgear22.dat
3 2
multgear21.dat
4 4
carrier1_21.dat
4 3
intgear22.dat
4 5
shaft21.dat
5 1
simpgear21.dat
6 3
intgear21.dat
```

```
/*
The order of the data is
number of elements
(for each element)
link number, element number, file name for data for the element}
*/
```

A.3 Listing for the PHIGS structure creation of an external gear

```
/* ---------------------------------- */
/* The function 'find_pos_simpgear' gets the local positions
of the simple gear based on the data in the file name associated
with the 'file_ptr'. */

/*      This function is called by 'draw_simpgear' function */

int find_pos_simpgear(vertex vert6[], FILE *file_ptr,
int *orient_ptr) {

/*      Variable declarations  */

        float   shftrad,hubrad1,hublnth1,gearrad1;
float gearthk1,clearance;
        int     temp2,orient_factor,link_type;
        ORIENT  orient_gear;

/*      Read information from data file about the orientation of
hub of the external gear. */

        fscanf(file_ptr,"%d",&temp2);
        orient_gear = (ORIENT)temp2;
        if(orient_gear == left) {
                orient_factor = -1;
                *orient_ptr = -1;
        }
        else {
                orient_factor = 1;
                *orient_ptr = 1;
        }
```

```
/*      Read information from data file whether the joint at the
center is of the fixed or the rotating type. Obtain information
of the sizes of the various parameters of the gear. */

        fscanf(file_ptr,"%d",&link_type);
        fscanf(file_ptr,"%f %f %f %f %f %f",&shftrad,&hubrad1,
            &hublnth1,&gearrad1,&gearthk1,&clearance);

/*      Based on the parameters find the locations of the
vertices for the external gear. The numbering of the
vertices for the external gear is shown in Figure 3.2.

        vert6[1].localx = orient_factor*(hublnth1);
        if(link_type == rot_pair)
                vert6[1].localy = shftrad + clearance;
        if(link_type == fix_pair)
                vert6[1].localy = shftrad;
        vert6[2].localx = vert6[1].localx;
        vert6[2].localy = hubrad1;
        vert6[3].localx = orient_factor*(gearthk1/2);
        vert6[3].localy = vert6[2].localy;
        vert6[4].localx = vert6[3].localx;
        vert6[4].localy = gearrad1;
        vert6[5].localx = orient_factor*(-gearthk1/2);
        vert6[5].localy = vert6[4].localy;
        vert6[6].localx = vert6[5].localx;
        vert6[6].localy = vert6[1].localy;
        return(link_type);
}
/* ----------------------------------- */


/* ----------------------------------- */
/* The function 'draw_simpgear' draws a simple gear. */
void draw_simpgear(char file_name[], int elem_id,
int stid, float loc_mxmn[]) {
#define maxpts 20

/*      Variable declarations. */

        int     i,j;
        int     link_type, orient_factor, *orient_ptr;
        vertex  vert6[maxpts];
        point3d points[maxpts];
        FILE    *file_ptr;

        float   ident_matrix[4][4] =   {
                {1,0,0,0},
                {0,1,0,0},
                {0,0,1,0},
                {0,0,0,1}
        };

        float   out_matrix[4][4];
        float   fixed_pt[3] =   {
                0.0,0.0,0.0             };
        float   scale_vector[3] = {
                1.0,1.0,1.0             };
        float   x_angle = 0.0, y_angle = 0.0, z_angle = 0.0;
        int     err_ind, comp_type = 2;
        float   trans_vector[3] = {
                0.6,0.6,0.0             };

        Ppoint  *text_posi;
        text_posi = (Ppoint *) malloc(sizeof(Ppoint));

/*      Open the data file. A typical example data file is
given at the end of the code. The element is described in
modeling coordinates. Read data file for the transformation
to be applied to move the element with respect to the
world coordinate system. */

        file_ptr(file_name,"r");
        fscanf(file_ptr,"%f %f",&trans_vector[0],
&trans_vector[1]);
        orient_ptr = &orient_factor;

/*      Obtain transformation matrix using PHIGS. */

        pcomposetran3(ident_matrix,fixed_pt,trans_vector,x_angle,
y_angle,z_angle,scale_vector,&err_ind,out_matrix);
        if(err_ind != 0)
                printf ("\n Error index is %d \n",err_ind);

/*      Call the function 'find\_pos\_simpgear' to find the
vertices of the simple gear in modeling coordinates. */
```

```
        link_type = find_pos_simpgear(vert6,file_ptr,orient_ptr);

/*      Find the location for placing the element number
for the element. */

        text_posi->x = vert6[3].localx;
        text_posi->y = (vert6[3].localy + vert6[4].localy)/2;
#ifdef PHIGS

/*      Open a structure to create a PHIGS structure. */

        popenstruct(stid);
        psetlinecolourind(elem_id%7+1);

/*      Apply the transformation matrix. */

        psetlocaltran3(out_matrix,comp_type);

/*      Draw the text for the element number. */

        draw_textelemid(elem_id, orient_ptr, text_posi);

/* Create the structure for the upper half of the
external gear. */

        for(i = 0; i < 6;  i++) {
                points[i].x = vert6[i+1].localx;
                points[i].y = vert6[i+1].localy;
                points[i].z = 0.0;
        }
        if(link_type == rot_pair) {
                points[i].x = vert6[1].localx;
                points[i].y = vert6[1].localy;
                points[i++].z = 0.0;
        }
        draw_polyline(i,points);

/* Keep track of the maximum and minimum modeling coordinates. */

        get_maxmin(i,points,loc_mxmn);

/*  Create the structure for the lower half of the
```

```
external gear. */

for(j = 0; j < i;  j++) {
                points[j].y = -points[j].y;
        }
        draw_polyline(i,points);

/* Keep track of the maximum and minimum modeling coordinates. */

        get_maxmin(i,points,loc_mxmn);
        for(j = 0; j < 4;  j++)
                printf("%f ",loc_mxmn[j]);
        printf("\n");

/* Keep track of the maximum and minimum world coordinates. */

        loc_mxmn[0] += trans_vector[0];
        loc_mxmn[1] += trans_vector[0];
        loc_mxmn[2] += trans_vector[1];
        loc_mxmn[3] += trans_vector[1];

/*      Close the structure for this element. A transformation
is later applied to this structure to fit the gear train onto
the PHIGS display window along with the structures for other
elements and then displayed onto the screen. */

        pclosestruct(stid);
#endif /* PHIGS */
}
/* ----------------------------------- */
```

A.3.1 Example data file for an External gear

```
0.0 0.0

1 0
0.025 0.05 0.075 0.2
0.025 0.005

/* The order of the data is:
xy-coordinates of translation of the element to
world coordinates..
```

```
orient_factor, link_type
shftrad, hubrad1, hublnth1, gearrad1,
gearthk1, clearance */
```

Appendix B

Extension of Optimization to the Ravigneaux gear train

Consider the Ravigneaux gear train in Figure B.1(a). (Figure 5 of [27]). Tsai et al. [27] have identified two practical automotive applications using this gear train. The graph is shown in Figure 1.4(b) and the clutching sequence table for the two applications is shown in Figure B.1(c). The objective function to be minimized is

$$F(\mathbf{R}) = \sum_{i=1}^{5}(R_i - k_i)^2 \tag{B.1}$$

where $R_1 \ldots R_4$ are given by

$$R_1 = r_{46}r_{65}/r_{25} \tag{B.2}$$

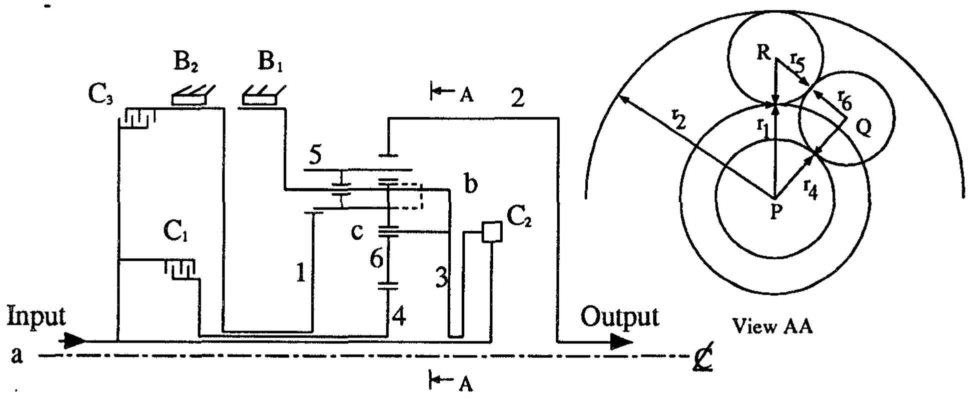
$$R_2 = (r_{25} - r_{15})r_{46}r_{65}/(r_{25}(r_{46}r_{65} - r_{15})) \tag{B.3}$$

$$R_3 = 1 \tag{B.4}$$

$$R_4 = 1 - r_{15}/r_{25} \tag{B.5}$$

$$R_5 = r_{15}/r_{25} \tag{B.6}$$

Some of the constraints differ from the optimization problems in Chapter 3.



(a)

	C_1	C_2	C_3	B_1	B_2
FIRST	X			X	
SECOND	X				X
THIRD	X	X			
FOURTH		X			X
REVERSE			X	X	

C - Input Clutch, B - Brake

(b)

Figure B.1: Ravigneaux Gear train (a) functional schematic (b) clutching sequence.

Similar to the derivation of Eq. (3.9), for gears 1, 2 and 5 we have

$$r_{25} + r_{15} = 2$$

Upper and lower and limits can be chosen for r_{25} and r_{15}. Other geometric constraints have to be derived to account for the presence of the idler gear. Let r_i be the radius of gear i. Then r_1 should be at least as large as r_4, i.e., $r_1 - r_4 \geq 0$, which can be equivalently represented as

$$r_{46} \leq r_{15}/r_{65}$$

The diameter of gear 6 should be at least as large as the difference of r_1 and r_4, i.e. $2r_6 \geq r_1 - r_4$, which can be equivalently represented as

$$2 \geq r_{15}/r_{65} + r_{46}$$

The diameter of gear 6 should not be very large and has to fit in the space between the sun and ring gears; therefore,

$$r_2 \geq r_4 + 2r_6 + (addendum\ of\ 2\ gears) + space \tag{B.7}$$

If the space is considered some arbitrary multiple of the addendum, $space = k(addendum)$, where k is some reasonable value we have that

$$2 - r_{15} \geq r_{46}r_{65} - 2r_{65} + (2+k)(addendum)/r_5 \tag{B.8}$$

Let us assume that the addendum is equal to the $1/P$, where P is the diametral pitch, and $k = 1$. Then

$$K = (2+k)(addendum)/r_5 = 2(2+1)/t_5 \tag{B.9}$$

The upper limit for K can be found from the minimum value of t_5.

For a given value of r_{15}, if either r_{46} or r_{65} is assumed to lie between some specified limits, then the upper and lower limits of the remaining variable is constrained by the Eqs. (B.7)–(B.9). For example, gear 4 will usually be larger than gear 6, and this can be judiciously used to set the limits of r_{46}. The constraints can be stated as follows, after choosing the upper and lower limits for some of the variables,

$$g_1 : \ r_{25}/8.0 - 1 \tag{B.10}$$

$$g_2 : \ 3.0/r_{25} - 1 \tag{B.11}$$

$$g_3 : \ r_{15}/(-6.0) - 1 \tag{B.12}$$

$$g_4 : \ -1.0/r_{15} - 1 \tag{B.13}$$

$$g_5 : \ r_{46}/(-2.5) - 1 \tag{B.14}$$

$$g_6 : \ -1.0/r_{46} - 1 \tag{B.15}$$

$$g_7 : \ 1/2(r_{15}/r_{65} + r_{46}) - 1 \tag{B.16}$$

$$g_8 : \ (r_{15} - 2 - 2r_{65} + r_{46}r_{65})/K + 1 \tag{B.17}$$

$$g_9 : \ r_{15}/r_{46}r_{65} + 1 \tag{B.18}$$

$$h_1 : \ (r_{25} + r_{15})/2 - 1 \tag{B.19}$$

From Eqs. (B.2) – (B.6) it can be seen that the effect of the gear ratios r_{46} and r_{65} on the reduction ratios is always through the product $r_{46}r_{65}$, which in turn can be considered as a single variable. Theoretically, this means that there are an infinite number of optimum solutions for r_{46} and r_{65} that result in the same reduction ratios. Referring to Figure B.1(b) this can be interpreted geometrically as varying the angle θ in the triangle PQR. Applying the cosine law for triangle PQR we have

$$(r_5 + r_6)^2 = (r_4 + r_6)^2 + (r_5 + r_1)^2 - 2(r_4 + r_6)(r_5 + r_1)cos\theta$$

In terms of the gear ratios this can be shown to be

$$cos\theta = \{r_{46}r_{65}^2(2 - r_{46}) + r_{15}(2 - r_{15}) + 2r_{65}\}/\{2r_{65}(1 - r_{15})(1 - r_{46})\}$$

A designer can use the above equation to find the value of θ, provided he has the values of the gear ratios. This equation can also be used as a constraint equation in the optimization problem if the designer wants to limit the angle θ to lie within a certain range.

Bibliography

[1] Al, K. and Ira, P., *A Book on C*, The Benjamin/Cummings Publishing Company, Inc., California, 1990.

[2] Buchbaum, F. and Freudenstein, F., "Synthesis of Kinematic Structure of Geared Kinematic Chains and other Mechanisms", *Journal of Mechanism and Machine Theory*, Vol. 5, 1970, pp. 357–392.

[3] Chieng, W.H. and Hoeltzel, D.A., "A Combinatorial Approach for the Automatic Sketching of Planar Kinematic Chains and Epicyclic Gear Trains", *ASME Trends & Development in Mechanisms, Machines & Robotics*, Vol. 1, 1988, pp. 79–90.

[4] Heller, D, *Xview Programming Manual : An OPENLOOK Toolkit for X11*, Sebastapool, California, 1990.

[5] *DEC PHIGS Reference Manuals*, Version 2.0 or higher, Digital Equipment Corporation, Marynard, Massachusetts, 1989.

[6] Freudenstein, F., "An Application of Boolean Algebra to the Motion of Epicyclic Drives", *ASME Journal of Engineering for Industry*, Vol. 93, 1971, pp. 176–182.

[7] Imai, K., *Configuration Optimization of Trusses by the Multiplier Method*, Ph.D. Thesis, University of California, Los Angeles, 1978.

[8] Arora S.J., *Introduction to Optimum Design*, McGraw-Hill, New York, 1989.

[9] Fiacco, A.V. and McCormick, G.P., *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley & Sons, New York, 1968.

[10] Kochan, S. G., *Programming in C*, CBS Publisher's & Distributors, Delhi, India, 1991.

[11] Mruthyunjaya, T.S., "A computerized methodology for structural synthesis of kinematic chains, Part I—Formulation", *Mechanism and Machine Theory*, Vol. 19, 1984, pp. 487–495.

[12] Olson, D.G., Thompson, T.R., Riley, D.R., and Erdman, A.G., "An Algorithm for Automatic Sketching of Planar Kinematic Chains", *ASME Journal of Mechanisms, Transmissions and Automation in Design*, Vol. 107, No. 1, 1985, pp. 106–111.

[13] Olson, D.G., Erdman, A.G., and Riley, D.R., "Topological Analysis of Single-Degree-of-Freedom Planetary Gear Trains", *ASME Journal of Mechanical Design*, Vol. 113, Mar., 1991, pp. 10–16.

[14] Powell, M.J.D., "Optimization Algorithms in 1979", *Committee on Algorithms Newsletter*, No. 5, Mathematical Programming Society, Feb., 1981, pp. 2–16.

[15] *OpenWindows Developer's Guide 3.0: User's Guide*, SunSoft - A Sun Microsystems Company, Mountainview, California, 1991.

[16] *OpenWindows Developer's Guide: Programmer's Guide to the Xview Toolkit Code Generator*, SunSoft - A Sun Microsystems Company, Mountainview, California, 1991.

[17] Ramakrishnan, B., Olson, D.G., "An Efficient Approach for Generating Solid-Shaded Images of Planetary Gear Trains", *Proceedings of the 1989 ASME International Computers in Engineering Conference*, Vol. 1, 1989, pp. 111–116.

[18] Reklaitis, G.V., Ravindran, A., and Ragsdell, L.M., *Engineering Optimization*, John Wiley & Sons, New York, 1983.

[19] Sohn, W. and Freudenstein, F., "An Application of Dual Graphs to the Automatic Generation of Kinematic Structures of Mechanisms", *ASME Journal of Mechanisms, Transmissions and Automation in Design*, Vol. 108, 1986, pp. 392–402.

[20] *The SunOS Reference Manual*, Revision A, Sun Microsystems, Inc., Mountainview, California, 1990.

[21] *The OpenWindows Version 2 User's Guide*, Sun Microsystems, Mountainview, California, Inc.

[22] *The OPENLOOK UI Style Guide*, Sun Microsystems, Inc., Mountainview, California, 1989.

[23] Tsai, L.W., "An Algorithm for the Kinematic Analysis of Epicyclic Gear Trains", *Sixth Applied Mechanisms Conference*, Kansas City, 1985.

[24] Tsai, L.W., "An Application of the Linkage Characteristic Polynomial to the Topological Synthesis of Epicyclic Gear Trains", *ASME Journal of Mechanisms, Transmissions and Automation in Design*, Vol. 109, No. 3, 1987, pp. 329–336.

[25] Tsai, L.W., "The Kinematics of Spatial Robotic Bevel gear Trains", *IEEE Journal of Robotics and Automation*, Vol. 4, No. 2, Apr., 1988, pp. 150–155.

[26] Tsai, L.W., *System Design of Mechanisms - Lecture notes for the course ENME 808A*, University of Maryland, College Park, 1992.

[27] Tsai, L.W., Maki, E.R., Liu, T, and Kapil, N.G., "The Categorization of Planetary Gear Trains for Automatic Transmissions according to Kinematic Topology", SAE Technical paper # 885062, *Proceedings of the XXII FISITA, Congress and Exposition*, Vol. 1, 1988, pp. 1513–1521.

[28] Vanderplaats G.N., *Numerical Optimization Techniques for Engineering Design: With Applications*, McGraw-Hill, New York, 1984.

[29] Watt, A., *Fundamentals of Computer Graphics*, Addison-Wesley, 1989.

[30] *PEX-SI Graphics Library Manual Pages (on-line)*, Version 5.1 or higher, XConsortium, 1991.

[31] *SunCGI Reference Manual*, Revision A, Sun Microsystems, Inc., Mountainview, California, 1988.