COMPUTERS IN ENGINEERING 1985

VOLUME TWO

- Finite Element Methods
- Expert Systems
- Simulation
- Education

COMPUTERS IN ENGINEERING 1985

Proceedings of the 1985 ASME International
Computers in Engineering Conference and Exhibition
August 4–8, 1985
Boston, Massachusetts

sponsored by
The Computer Engineering Division, ASME

Co-Editors
R. Raghavan
S. M. Rohde

Associate Editors
K. Ahluwalia
J. Lester
E. Patton
R. Rosenberg
T. Shoup
D. Whitney

VOLUME TWO

- Finite Element Methods
- Expert Systems
- Simulation
- Education

THE AMERICAN SOCIETY OF MECHANICAL ENGINEERS
United Engineering Center 345 East 47th Street New York, N.Y. 10017

Library of Congress Catalog Card Number 82-072306

FOREWORD

The papers in these three volumes were presented at the 1985 International Computers in Engineering Conference and Exhibit held in Boston, Massachusetts during August 1985. Since its beginning in 1980, the ASME Computers in Engineering conference has become a major international event, bringing together experts from industry, government, and academia concerned with the ever evolving role of the computer in engineering.

This year's theme was "Expert Systems: A New Dimension in Computer Engineering", and the technical content of the conference focused on areas related to research, development and applications of computers in mechanical engineering with emphasis on expert systems. The scope of the conference included robotics, computer aided design and manufacturing, and computer aided engineering, as well as computers in education, simulation, concurrent computations, software and hardware design, optimization, graphics, and modeling. This year 65 sessions were held including panel and tutorial sessions, 264 papers were contributed and participants came from 25 countries. Keynote speakers were Marvin Denicoff, Raj Reddy, H. Bloom, and Nien-hua Chao.

This year's technical program was put together by the following individuals who deserve special credit for the excellent job that they did: K. S. Ahluwalia, J. Lester, E. M. Patton, R. C. Rosenberg, T. Shoup, D. E. Whitney. In addition we would like to acknowledge the efforts of the International Program Chairman – A. Seirig, the Exhibits Chairman – D. Dietrich, and the ASME staff for their support and help.

On behalf of the Computer Engineering Division and the Conference Committee, we thank the speakers, session organizers, chairpeople, and the authors for their contributions. We look forward with anticipation to the new challenges in the years ahead.

R. Raghavan, Conference Chairman
S. M. Rohde, Conference Technical Program Chairman
Co-chairmen, Editorial Committee

CONTENTS

A COMPUTER-GRAPHICS BASED APPROACH TO MODELING COMPLEX PLANETARY GEAR TRAINS

L. J. Gutkowski
Crown Zellerbach, Inc.
Central Research
Camas, Washington

G. L. Kinzel
Department of Mechanical Engineering
The Ohio State University
Columbus, Ohio

ABSTRACT

An interactive, graphics-based approach is presented for modelling general planetary gear trains with nonparallel axes. The algorithm presented develops a graph for the gear train which is "built" in a interactive manner on a graphics terminal. The governing equations are developed automatically and solved. The solution developed gives the linear coefficients which relate the input velocities and torques to those at the output members. Two examples are presented.

INTRODUCTION

A planetary gear train (PGT) is one in which at least one gear rotation axis moves relative to the frame. Such gear trains permit large velocity ratios in a small package, and multidegree of freedom inputs can be easily accomodated. These features make PGTs ideal for applications such as automatic transmissions and robotic wrists. Unfortunately, complex PGTs are difficult to analyze, especially by hand; however, the analytical techniques for analysis are well adapted to the digital computer.

A general technique for the kinematic analysis of parallel axis planetary gear trains was presented by Freudenstein and Yang [1]. The method involved representing a PGT by a "graph" whose topology described the PGT's kinematic structure, and this graph could be used to develop kinematic equations that represented both the planetary and nonplanetary portions of the PGT. Day, Akeel, and Gutkowski [2] have extended this technique to the more general case of nonparallel axis PGTs (i.e., those possessing bevel gears).

This paper describes a graphics oriented computer program that was written to automate the PGT kinematic analysis. The analyst may simply draw (in a piecewise fashion) a schematic of a PGT on a graphics terminal using several basic elements. The program then uses the procedures set forth by Freudenstein, et. al. and Day, et. al. to kinematically analyze the gear train. It is not necessary to have the kinematic insight that is required to use techniques such as the Tabular Method [3], the Formula Method [4], or a building block method described by Hanson [5]. Very complicated assemblies with many degrees of freedom are easily modeled and analyzed.

REVIEW OF ANALYSIS PROCEDURE

Every gear mesh in a PGT is a form of the Fundamental Gear Mesh (Fig. 1). The fundamental unit consists of the two meshing gears and a link (carrier) that maintains the mesh center distance. Four different forms of this fundamental mesh exist and are based on the parallelism conditions between the $\hat{i}$, $\hat{j}$, and $\hat{k}$ rotation axes. By reducing a large PGT to a collection of fundamental gear meshes, a simultaneous set of linear kinematic equations can be developed.

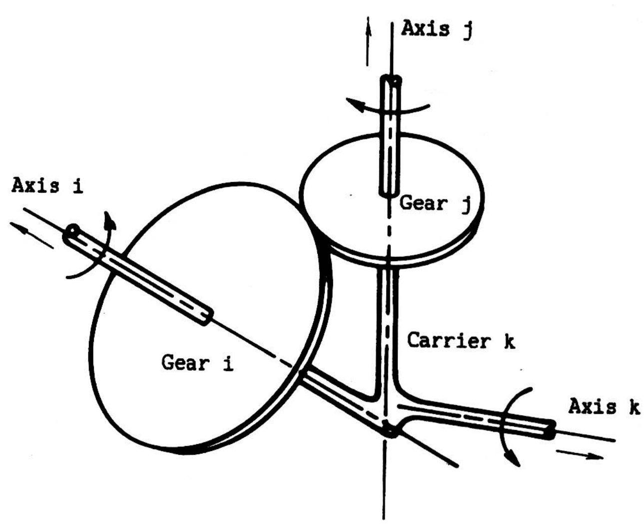


Figure 1: Fundamental gear mesh assembly.

The graphing procedure described by Freudenstein and Yang provides a "cookbook" method for finding the constituent fundamental meshes of a PGT. Once the fundamental mesh information is determined (i.e., the gears involved, the carrier, and their respective rotation axes), the kinematic relations are determined easily.

The graphing technique is shown in Fig. 2 for the Humpage reduction gear. The steps used in drawing the graph are as follows: (1) Number each link of the PGT. Do not include kinematically redundant links such as multiple planets in parallel. (2) Label the axes of turning pairs with lower case letters. (3) Represent each link by a numbered point corresponding to the link numbers of step 1. These points are called vertices. (4) If two gears (links) mesh with each other, draw a solid line (geared edge) between the corresponding vertices laid down in step 3. (5) If two links are connected by means of a turning joint, draw a dashed line (turning edge) between the corresponding vertices. The turning edge should then be labeled according to its axis label (from Step 2).

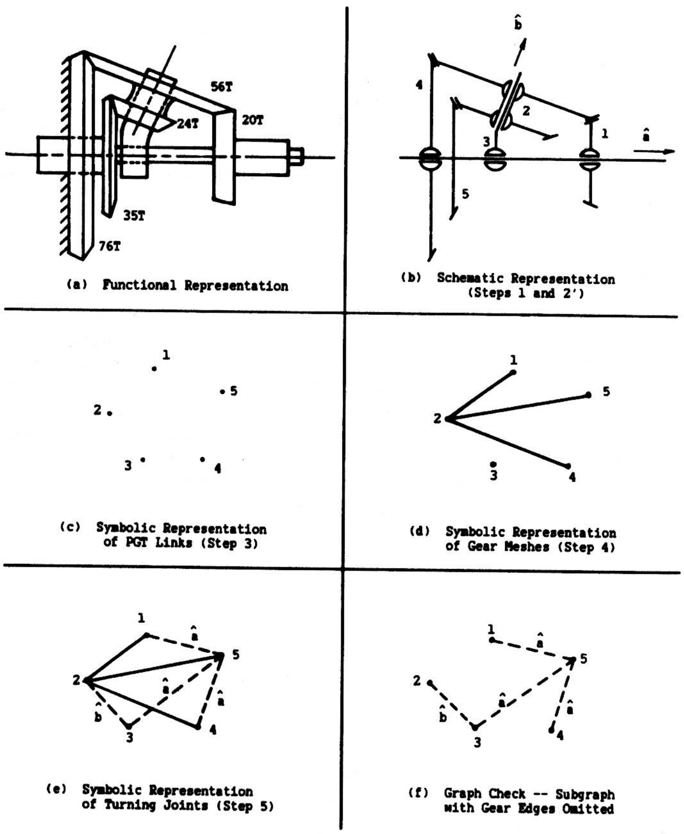


Figure 2: Graphing procedure for Humpage reduction gear.

In order to handle nonparallel axis PGTs, step 2 of the graphing method is modified as follows: (2') Define the orientation of turning pair axes with lower case unit vectors ($\hat{a}$, $\hat{b}$, $\hat{c}$, ...).

If a PGT graph is drawn correctly, each geared edge will form a unique closed path with other turning edges. This closed path, called a fundamental circuit (or f-circuit) will give useful information regarding the two gears connected by the geared edge. Each f-circuit consists of an open loop of turning edges which is closed by one geared edge. There should be an f-circuit in the graph for each gear mesh in the PGT. The f-circuits for Figure 2e are listed in Table 1.

Table 1: F-circuit work table with transfer vertices.

	GEARED EDGE	TURNING EDGES	TRANSFER VERTEX
Circuit I:	(1,2)	(2,3) (3,5) (5,1)	3
II:	(5,2)	(2,3) (3,5)	3
III:	(4,2)	(2,3) (3,5) (5,4)	3

Having established the f-circuits for a graph, the "transfer" vertex for each circuit must be determined. This vertex represents a link on which two different turning axes occur relative to the given mesh (i.e., the transfer vertex is the carrier). By traversing the turning edges of an f-circuit, the transfer vertex will be found where the turning axis label changes. For instance, if the first part of an f-circuit had turning edges labeled by axis $\hat{d}$ and the last part was labeled by axis $\hat{p}$, the transfer vertex is the vertex at which the axis changes from $\hat{d}$ to $\hat{p}$. The transfer vertices for Figure 2e are also shown in Table 1.

Kinematic equations can be developed quickly for the gear train at hand using the f-circuit information. These equations are essentially based on the kinematic properties of the fundamental gear mesh subassembly (i.e., two gears and a carrier). There are four different fundamental mesh configurations possible based on the parallelism of the $\hat{i}$, $\hat{j}$, and $\hat{k}$ turning axes. A governing kinematic equation can be written for each of these cases as follows [6]:

$$ {}^{1}W_i - N_{ji}\,{}^{k}W_j \mp W_k = 0 \qquad i \parallel k,\ j \nparallel k \qquad (1)$$

$$ {}^{k}W_i - N_{ji}\,{}^{1}W_j \pm N_{ji}\,W_k = 0 \qquad i \nparallel k,\ j \parallel k \qquad (2)$$

$$ {}^{1}W_i - N_{ji}\,{}^{1}W_j + (\pm N_{ji} \mp 1)\,{}^{1}W_k = 0 \qquad i \parallel k,\ j \parallel k \qquad (3)$$

$$ {}^{k}W_i - N_{ji}\,{}^{k}W_j = 0 \qquad i \nparallel k,\ j \parallel k \qquad (4)$$

i,j = gear indices

k = carrier index

${}^{b}W_a$ = angular speed of link a relative to link b

N_{ji} =±[teeth on gear j]/[teeth on gear i]; (positive if positive rotation of j causes positive rotation of i)

Assembly and Solution of the Kinematic Equations

For a given PGT, the appropriate equation (from Eqn (1)-(4)) will be written for each gear mesh (i.e., each f-circuit). If Jg represents the number of gear meshes, there will be Jg equations developed. If there are L links in the system there will be L angular velocity variables in the equations. These equations can be assembled in the form:

$$[\ S\ ]\{\ \Omega\ \} = \{\ 0\ \} \qquad (5)$$

where

$[\ S\ ]$ = the system kinematic matrix; dimension Jg x L

$\{\ \Omega\ \}$ = the column vector of angular velocities; dimension L x 1

$\{\ 0\ \}$ = the null or zero column vector; dimension Jg x 1.

To uniquely solve this system of equations, (L - Jg) variables must be specified (in general L > Jg). Hence the number of degrees of freedom for the system is

$$F = L - Jg \qquad (6)$$

The angular velocities for the input links can be classified as independent while those for the remaining links are dependent. The column vector of angular velocities is partitioned into independent (Ωi) and dependent (Ωd) sections. Hence, Equation (5) becomes:

$$\left[\ [\ A\ ] \,|\, [\ -b\ ]\ \right] \left\{ \begin{array}{c} \Omega d \\ \hline \Omega i \end{array} \right\} = \{\ 0\ \} \qquad (7)$$

where

$[\ A\ ]$ = the system matrix partition containing the dependent variable coefficients; dimension Jg x Jg

$[\ -b\ ]$ = the system matrix partition containing the independent variable coefficients; dimension Jg x F ,

or by separating the independent and dependent variables to different sides of the equation, one obtains:

$$[\ A\ ]\{\ \Omega d\ \} = [\ b\ ]\{\ \Omega i\ \}. \qquad (8)$$

The dependent variables can then be determined in terms of the independent variables using:

$$\{\ \Omega d\ \} = [\ A\ ]^{-1}[\ b\ ]\{\ \Omega i\ \}. \qquad (9)$$

Torque Equation Development and Solution

It is now possible to derive a relation among the independent and dependent torques if we assume there are no energy losses in the system. The controlling or independent links will be termed inputs, and the controlled or dependent links will be outputs.

An appropriate energy principle employed in this analysis is:

$$\sum \vec{T} \bullet \vec{W} = 0 \qquad (10)$$

where

$\vec{T}$ = externally applied torque about component turning axis

$\vec{W}$ = angular velocity of component about its turning axis.

The torque on a given component is assumed to act about the component's rotation axis (as was its angular velocity). Since this makes the external torque and angular velocity vectors parallel on a given component, the terms in (10) may be taken as signed scalars -- positive if the vector quantity is directed in the positive direction of the component's rotation axis unit vector.

Applying Eq. (10) to a multiple input/output system, one may write:

$$\{\ to\ \}^T \{\ \Omega d\ \} = \{\ ti\ \}^T \{\ \Omega i\ \} \qquad (11)$$

where

$\{\ to\ \}$ = column vector of output external torques acting on dependent variables $\{\ \Omega d\ \}$; dimension Jg x 1

$\{\ ti\ \}$ = column vector of input torques required on the input shafts with speeds $\{\ \Omega i\ \}$; dimension F x 1 .

Combining (9) and (11) yields:

$$\{\ to\ \}^T [\ A\ ]^{-1} [\ b\ ]\{\ \Omega i\ \} = \{\ ti\ \}^T \{\ \Omega i\ \}. \qquad (12)$$

Hence,

$$\{\ to\ \}^T [\ A\ ]^{-1} [\ b\ ] = \{\ ti\ \}^T, \qquad (13)$$

or solving for the input torques as a function of the output torques:

$$\{\ ti\ \} = \left[[\ A\ ]^{-1} [\ b\ ] \right] \{\ to\ \} \qquad (14)$$

where $[\ A\ ]$ and $[\ b\ ]$ have been determined from the kinematic analysis.

PROGRAM DESCRIPTION

GBUILD (Gear Train Model Builder) is an interactive graphics computer program designed to allow the user to draw schematic representations of planetary gear trains [6]. GBUILD is capable of analyzing the schematics to yield the necessary information for each fundamental circuit in the gear train. This f-circuit data can then be used by a solution program called GSOLVE to determine input/output kinematic and torque characteristics of the train.

GBUILD is similar in nature to a finite element graphic preprocessor. The elements (gears, shafts, and carriers) have nodes that may be connected via joints (geared, welded, or turning) to form a gear train schematic. Connectivity information for the gear train models is much more involved than that of a finite element preprocessor, however, due to the various types of joint possibilities between the elements. Geared, turning, and welded joints all have different kinematic ramifications thus making it necessary to specify more than just what elements are connected together -- it must also be recorded how they are connected together.

Gear Train Elements

The four elements used by GBUILD involve a number of parameters that are defined by the user with either the cross hairs or the keyboard. These parameters will result in the automatic placement (and numbering) of nodes on the elements. Elements may be connected to other elements at these nodal locations. Certain types of nodes (classified by where they occur on an element) are not always compatible with all the available joint types. If the user tries to join nodes of two elements together and they are not compatible with the type of joint used, an error message will be generated.

The External Gear Element (Figure 3a) in its most general form represents a beveled external gear. When first created, the external gear element has a pitch cone angle of 0 degrees. If this gear is meshed with another gear, each of their pitch cone angles are automatically adjusted to the proper bevel based on their respective shaft angles.

The parameters that define an external gear are its center of rotation, shaft angle, pitch radius, and the automatically defined pitch cone angle. Upon the creation of an external gear, three nodes and their X,Y locations are automatically defined -- one at the gear center and one at each end of its pitch diameter.

The Internal Gear Element (Figure 3b) is very similar to the external gear. The gear center, shaft angle, and pitch radius must be defined by the user. The pitch cone angle defaults to 0 degrees but will be adjusted to suit other gears joined to it. An additional parameter that the user must define for the internal gear, however, is that of the internal gear "depth" (Figure 3b). As with the external gear, the internal gear is set up with three nodes -- one at the gear center and one at each end of its pitch diameter.

The Shaft Element (Figure 3c) is intended for use where two links must turn about each other (i.e., via a revolute joint). This element is created by first defining the location of one of its endpoints and the shaft angle.

The Carrier Element (Figure 3d) is used as a rigid link "spacer" in the PGT schematics. A number of them may be combined together with shafts using welded joints to form large, complicated carrier links. Shafts may be used to perform the same function as carriers. The carrier is simpler to define, however, because there is no shaft angle associated with it. It is defined only by its endpoints. Because it has no shaft angle, the carrier may not have any element spin on it (i.e., it cannot act as a shaft for a gear).

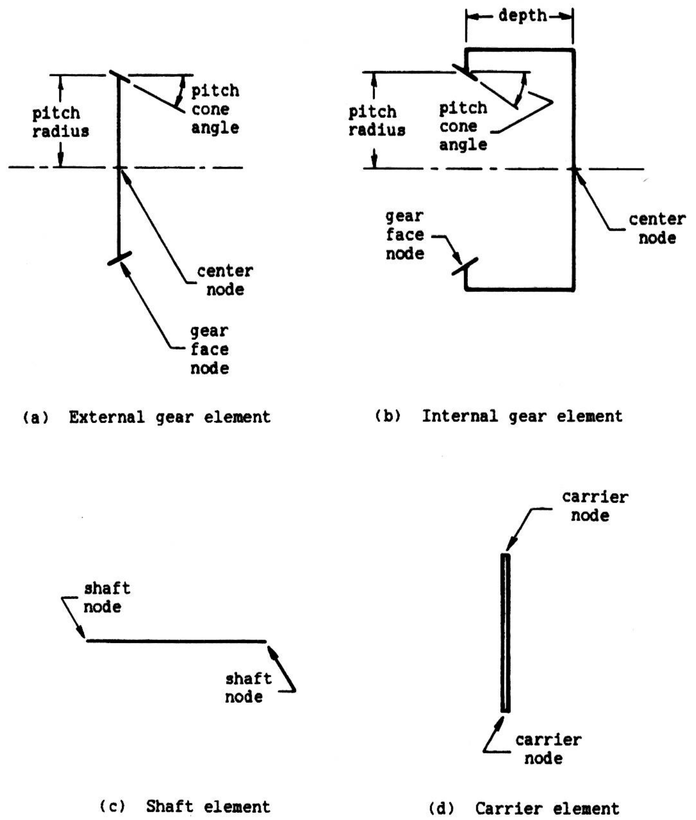


Figure 3: Gear train elements available in GBUILD.

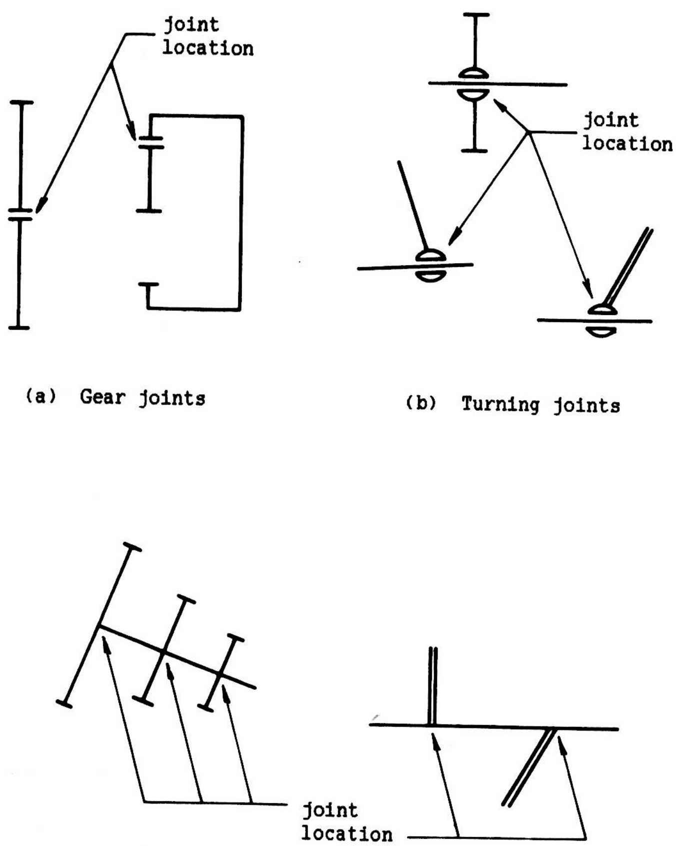
Gear Train Joint Types

GBUILD allows the user to connect the available elements together with three different types of joints. Listed below are the three types of joints, their properties, and rules for where they may be used.

The Geared Joint is used to create a gear mesh between two gears. Only nodes that reside at the gear faces of gear elements are eligible for this type of joint. When a geared joint is made, the two gear elements will be moved together on the screen (after redraw) so that the gear faces joined will appear to be in mesh (Figure 4a). Also, their pitch cone angles will automatically be adjusted so that their pitch cones roll on each other with a common apex.

The Turning Joint is a revolute joint. Its representation is shown in Figure 4b. It can be used to connect the center node of a gear to a shaft so that the gear spins on the shaft. The turning joint can also be used to connect shafts together such that one shaft spins around the other. A carrier may also be connected to a shaft in this manner.

The Welded Joint does not permit relative motion between the members joined. Its existence is necessary if gear trains involving compound gears or complex carrier assemblies are to be analyzed. All node types except gear face nodes may be involved in a weld joint. The existence of a weld joint in a schematic is represented merely by the fact that the welded nodes become coincident on the drawing (Figure 4c).



(a) Gear joints

(b) Turning joints

(c) Welded joints

Figure 4: Gear train joints available in GBUILD.

Determining F-Circuit Information

Each f-circuit is made up of three links -- two links (gears) joined by a gear mesh and one link called the transfer vertex (carrier). Although other information such as link rotation angles, gear tooth numbers, and mesh signs are needed for each f-circuit, the most difficult task lies in finding the three necessary links. Locating the two links involved in the gear mesh is a simple chore. GBUILD indexes through all of its joint information until it finds a gear mesh, and then it looks for the rest of the data to complete that f-circuit. Hence, the fact that GBUILD is developing a given f-circuit at all is due to the known existence of a gear mesh. The only real problem in completing an f-circuit, therefore, is to determine what link serves as the transfer vertex.

When using the PGT graphs, one must find a unique closed path of turning edges that starts and stops at a gear edge for each gear mesh in the gear train. This is used to develop the f-circuit information. Each turning edge in that closed path represents two links being joined by a revolute joint. As such, each turning edge has a turning axis associated with it. The vertex link will be indicated when a change in turning axis is found as the turning edge loop is traversed. The uniqueness of the path as well as the change in turning axis are the properties used by GBUILD to find the vertex link for a given gear mesh.

Using element shaft angle and connectivity information, GBUILD sets up a "table" (internally) that cross lists all turning joints and their axes by the links they join. For instance, Table 2 is set up internally for the train in Figure 5. The unit vector represents the turning axis for each joint . This table is useful for finding the closed path of turning edges for a given gear mesh.

Table 2: Cross listing of links by turning joint connection (lower case letters are turning axis labels)

Link i	Links Connected to Link i by a Turning Joint
1	4-$\hat{a}$
2	4-$\hat{b}$
3	4-$\hat{a}$
4	1-$\hat{a}$, 2-$\hat{b}$, 3-$\hat{a}$

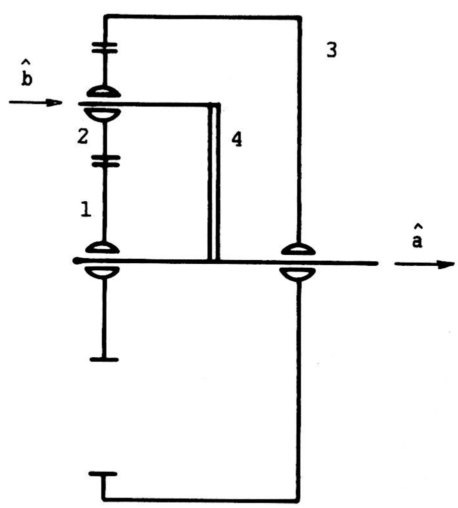


Figure 5: Schematic for a simple planetary gear train.

Suppose the path involving the gear mesh between links 2 and 3 of Figure 5 is needed. A tree can be drawn as shown in Figure 6. The idea is to start with the link number of one of the gears in mesh and then find the path down the tree leading to the link number of the other gear in mesh. This will give the closed path around the f-circuit of that mesh. Using this concept, link 2 (one of the gears) is put on level 1 of the tree. From the second line of Table 2, it is determined that link 4 is joined to link 2 by a turning joint. Thus link 4 is placed on level 2 of Figure 6. Link 4 is not the link number of the other gear in mesh so another level is added. Referring to line 4 of Table 2, it is seen that links 1, 2, and 3 are joined to link 4. Hence, level 3 of Figure 6 is filled with all three of these links.

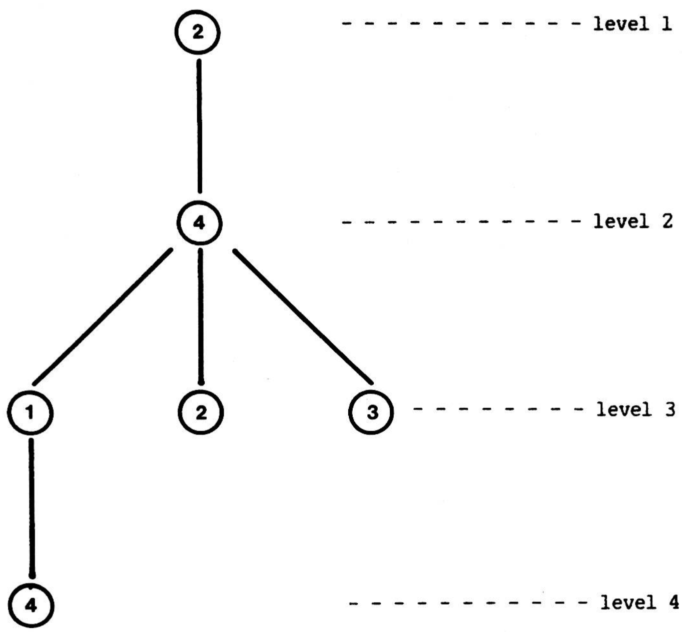


Figure 6: Internal tree structure developed by GBUILD used to find f-circuit paths.

Going from left to right in level 3, it is seen that link 1 is not the number of the link that link 2 is meshed with so this branch is extended to level 4. According to line 1 of Table 2, link 4 is placed on level 4 of the tree. Notice that link 4 in level 4 turns out to be a repeat of a link number higher up in the tree. Hence, this path is not possible for the f-circuit.

Therefore, the process is backed up to the next entry in level 3 -- this is link 2. But link 2 also is a repeat of a link number higher up in the tree. Thus, this path is not possible either. Proceeding to the next entry in level 3 yields link 3. This is the link number of the gear in mesh with link 2. Hence, the path has been found for the f-circuit corresponding to the gear mesh between links 2 and 3 (viz., link 2, link 4, link 3). This same technique can be used to find the f-circuits for the other gear joints. Each time a new tree has to be drawn where the first level will contain the link number of one of the gears in mesh.

Since such a simple example is depicted here, only three links are involved in the circuit path. Hence, by default, link 4 has to be the transfer vertex link. It is possible, however, that a tree may yield more than three links in the path, thus leaving more than one candidate for the vertex link. Therefore, once GBUILD finds the link path around a given f-circuit, it must next compare the turning axis of each turning edge, in order around the path, to the axis before it. Once the axis changes, the link number at which the axis change took place must be the vertex.

GBUILD will always proceed as far down the tree as it can go until it repeats a link number that occurs higher up in the tree, and it will do this in a "left to right" manner. If a dead end is encountered (i.e., a repeated link number), GBUILD will back up a level (or levels) and try the next branch. If GBUILD has gone down all of the branches and never finds the link number corresponding to the second gear in mesh for the given f-circuit, an error message will be displayed.

Once the f-circuit links are determined for each gear mesh in a gear train, GBUILD must now add the additional information such as link rotation axes, gear tooth numbers, and mesh signs. The rotation axis and tooth number information is obtained by merely looking them up in a data file. The mesh signs, however, require more logic to determine them.

Because the gear faces default to a pitch cone angle of 0 degrees when they are created, the modified pitch cone angles on beveled gears will have either a positive or a negative sign associated with them. This corresponds to a counter clockwise or a clockwise rotation of the gear face relative to its hub to simulate a beveled effect for the schematic elements. As noted before, the gear faces (or pitch cone angles) are modified automatically when two gears are joined by a gear mesh.

A negative pitch cone angle causes the pitch cone apex to point in the direction of the gear's shaft angle (i.e., the positive rotation axis). Hence, it turns out that if the pitch cone angles of meshing gears are of the same sign, the mesh sign will be negative. This implies that a positive rotation of one of the gears results in a negative rotation of the other gear. If the pitch cone angles are of opposite sign, the mesh is positive. In the situation where the cone angles are zero, then the gear shaft angles must be compared directly. If the shaft angles are the same, the mesh sign is negative; if they are 180 degrees apart, the mesh sign is positive.

The same technique is applied to internal/external meshes. In this case, however, if the cone angles are of the same sign, the mesh sign is positive; angles of opposite sign imply a positive mesh. In essence, conditions that cause a certain mesh sign for an external/external mesh will cause the opposite mesh sign for an external/internal mesh.

PROGRAM GENERAL STRUCTURE

The organization of GBUILD was designed with both the experienced and inexperienced user in mind. For those relatively unfamiliar with the use of a program, it is advantageous for the program to provide extensive prompting. However, the experienced user quickly becomes annoyed by excessive text on the screen when he already knows in advance what data to enter. To accomodate both of these extremes in operator skill (as well as those in between), a command/menu is used.

The general structure is one in which a main or master menu controls certain general modes of program functions (i.e., creating, editing, defining, viewing, etc.). The user may put the program into any one of these main modes after which appropriate subfunctions are then accessible (i.e., creating elements, creating joints, view translation, view magnification, etc.). All of the directive commands necessary to operate GBUILD are one or two letter abbreviations symbolic of what the command does. The commands can be strung together (by embedding spaces or blanks between them) on one entry line. If insufficient commands are entered, the program will prompt the user for more information. Note that this prompting occurs only if the user does not enter enough commands for a particular command sequence. Hence, the experienced user is capable of bypassing practically all of the program prompts.

Main Menu Modes

The main menu modes are accessed by typing a slash (/) followed by the appropriate symbolic letter. The main menu modes available are:

/C - Create	/V - View
/E - Edit	/U - Utility
/P - Parameter	/D - Define

In order to obtain this option list on the screen, one must type /O. Once GBUILD is put into one of the above modes, it will stay in that mode until a new "/something" is entered. The screen will always indicate the current mode that the program is in by writing the appropriate symbolic letter at the beginning of the next command line. The entire list of commands currently available is shown in Figure 7.

Examples

The Humpage reduction gear that was analyzed in [2] was modeled using GBUILD (Fig. 8). The gear tooth numbers are listed in the upper left hand corner. The symbol legend is in the lower right hand corner of Fig.8. Links 1 and 4 were chosen to be the independent links (see lower left hand corner). GBUILD analyzed the schematic to determine the f-circuit information (Fig. 9a) for this PGT. GSOLVE was then used to determine the input/output kinematic and torque relations (Fig. 9b,c). From Fig. 9b it is seen, for example, that

$$W3 = 0.20833 * W1 + 0.79167 * W4 \qquad (15)$$

The torque information is given in Fig. 9c. If known torques were required about links 2,3, and 5, the required input torques would be given by

$$T1 = -0.28274 * T2 + 0.20833 * T3 + 0.01446 * T5 \qquad (16)$$
$$T4 = 0.28274 * T2 + 0.79167 * T3 + 0.98554 * T5 \qquad (17)$$

An example of a 6-dof PGT for an automatic transmission is modeled in Fig. 10. The results are shown in Fig. 11a-c. Attempting to analyze this PGT by hand demonstrates the power of the computer programs.

```
/C - Create
   E - Element
      EG - External Gear
      IG - Internal Gear
      SH - Shaft
      CA - Carrier
   J - Joint
      GJ - Gear Joint
      TJ - Turning Joint
      WJ - Weld Joint
   D - Draw
----------------------------------
/P - Parameter
   L - Label
      N - Nodes
      E - Elements
      L - Links
      G - Gears
      A - Axes
      T - Teeth
   T - Tags
      N - Nodes
      E - Elements
   D - Draw
----------------------------------
/V - View
   M - Magnify
   Z - Zoom
   T - Translate
   C - Center
   P - Point Grid
   D - Draw
----------------------------------
/E - Edit
   MO - Modify
      SP - Subassembly Position
   PU - Purge
      J - Joint
      E - Element
   LI - List
      SB - Subassembly
      LK - Links
   LA - Label
      N - Nodes
      E - Elements
   D - Draw
----------------------------------
/D - Define
   LK - Link
   TE - Teeth
   IN - Independent Link
   AX - Link Rotation Axis
   D  - Draw
----------------------------------
/U - Utility
   A - Analyze Geometry
   L - List Dump
   E - Exit Program
   B - Begin Program
   S - Save Model
   R - Read Model
   D - Draw
```

Figure 7: GBUILD menu options.

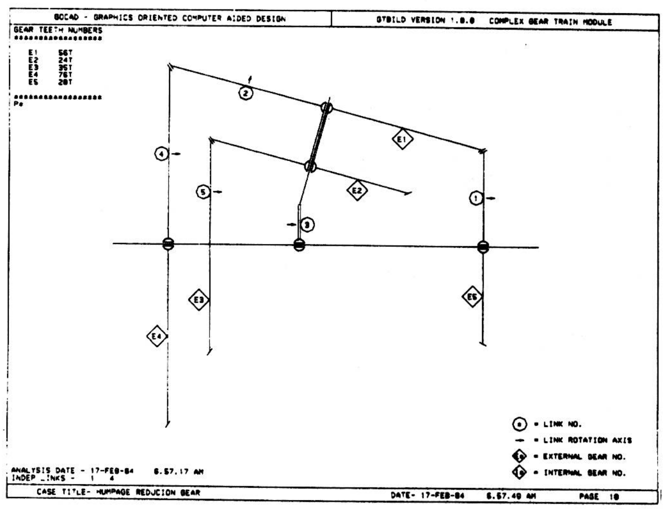


Figure 8: GBUILD schematic of Humpage reduction gear.

--------- COMPLEX PLANETARY GEAR TRAIN KINEMATIC ANALYSIS ----------
SOLUTION FROM GSOLVE VERSION 1.0.0 OHIO STATE UNIVERSITY FEB 1984

SOLUTION TITLE - HUMPGE GEAR REDUCTION

DATE OF SOLUTION - 17-FEB-84 6.58.44 AM

MODEL TITLE - EXAMPLE OF NONPARALLEL AXIS PGT

DATE OF GEOMETRY ANALYSIS - 17-FEB-84 6.57.17 AM

**** E C H O O F I N P U T D A T A ****

LINKS = 5 GEAR JOINTS = 3 DOF = 2

INDEPENDENT LINKS = 1 4

MESH NO	LINKS I	LINKS J	LINKS K	ANGLE I	ANGLE J	ANGLE K	TEETH I	TEETH J	MESH SIGN
1	5	2	3	0.0	75.0	0.0	35	24	1.
2	4	2	3	0.0	75.0	0.0	76	56	1.
3	1	2	3	0.0	75.0	0.0	20	56	-1.

Figure 9a: GSOLVE results for Humpage reduction gear; echo of input data.

--------- COMPLEX PLANETARY GEAR TRAIN KINEMATIC ANALYSIS ----------
SOLUTION FROM GSOLVE VERSION 1.0.0 OHIO STATE UNIVERSITY FEB 1984

SOLUTION TITLE - HUMPGE GEAR REDUCTION

DATE OF SOLUTION - 17-FEB-84 6.58.44 AM

MODEL TITLE - EXAMPLE OF NONPARALLEL AXIS PGT

DATE OF GEOMETRY ANALYSIS - 17-FEB-84 6.57.17 AM

**** V E L O C I T Y R E S U L T S ****

DEPENDENT SPEEDS	COEFFICIENTS FOR INPUT SPEEDS C 1	C 4
W 2	-0.28274	0.28274
W 3	0.20833	0.79167
W 5	0.01446	0.98554

ANGULAR SPEED OF LINK X ABOUT ITS ROTATION AXIS IS -

WX = SUMMATION OF (CI * WI),

FOR I = 1, 4

Figure 9b: GSOLVE results for Humpage reduction gear; kinematic results.

--------- COMPLEX PLANETARY GEAR TRAIN KINEMATIC ANALYSIS ----------
SOLUTION FROM GSOLVE VERSION 1.0.0 OHIO STATE UNIVERSITY FEB 1984

SOLUTION TITLE - HUMPGE GEAR REDUCTION

DATE OF SOLUTION - 17-FEB-84 6.58.44 AM

MODEL TITLE - EXAMPLE OF NONPARALLEL AXIS PGT

DATE OF GEOMETRY ANALYSIS - 17-FEB-84 6.57.17 AM

**** T O R Q U E R E S U L T S ****

INPUT TORQUES	COEFFICIENTS FOR OUTPUT EXTERNAL TORQUES K 2	K 3	K 5
T 1	-0.28274	0.20833	0.01446
T 4	0.28274	0.79167	0.98554

REQUIRED TORQUE ON LINK X ABOUT ITS ROTATION AXIS IS -

TX = SUMMATION OF (KI * TI),

FOR I = 2, 3, 5

Figure 9c: GSOLVE results for Humpage reduction gear; torsion results.

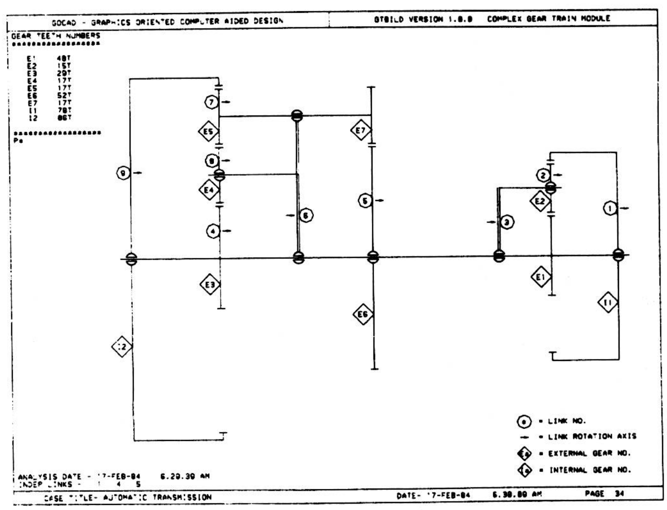


Figure 10: GBUILD schematic of complex PGT.

--------- COMPLEX PLANETARY GEAR TRAIN KINEMATIC ANALYSIS ----------
SOLUTION FROM GSOLVE VERSION 1.0.0 OHIO STATE UNIVERSITY FEB 1984

SOLUTION TITLE - ANALYSIS OF TRANSAXLE IN FIRST GEAR

DATE OF SOLUTION - 17-FEB-84 6.31.49 AM

MODEL TITLE - FIRST GEAR

DATE OF GEOMETRY ANALYSIS - 17-FEB-84 6.30.45 AM

**** E C H O O F I N P U T D A T A ****

LINKS = 9 GEAR JOINTS = 6 DOF = 3

INDEPENDENT LINKS = 1 4 5

MESH	LINKS			ANGLE			TEETH		MESH
NO	I	J	K	I	J	K	I	J	SIGN
1	9	7	6	0.0	0.0	0.0	86	17	1.
2	8	4	6	0.0	0.0	0.0	17	29	-1.
3	7	8	6	0.0	0.0	0.0	17	17	-1.
4	7	5	6	0.0	0.0	0.0	17	52	-1.
5	1	2	3	0.0	0.0	0.0	78	15	1.
6	2	4	3	0.0	0.0	0.0	15	48	-1.

Figure 11a: GSOLVE results for complex PGT; echo of input data.

--------- COMPLEX PLANETARY GEAR TRAIN KINEMATIC ANALYSIS ----------
SOLUTION FROM GSOLVE VERSION 1.0.0 OHIO STATE UNIVERSITY FEB 1984

SOLUTION TITLE - ANALYSIS OF TRANSAXLE IN FIRST GEAR

DATE OF SOLUTION - 17-FEB-84 6.31.49 AM

MODEL TITLE - FIRST GEAR

DATE OF GEOMETRY ANALYSIS - 17-FEB-84 6.30.45 AM

**** V E L O C I T Y R E S U L T S ****

DEPENDENT SPEEDS	COEFFICIENTS FOR INPUT SPEEDS C 1	C 4	C 5
W 2	2.60000	-1.60000	0.00000
W 3	0.61905	0.38095	0.00000
W 6	0.00000	0.35802	0.64198
W 7	0.00000	1.45316	-0.45316
W 8	0.00000	-0.73711	1.73711
W 9	0.00000	0.57450	0.42550

ANGULAR SPEED OF LINK X ABOUT ITS ROTATION AXIS IS -

WX = SUMMATION OF (CI * WI),

FOR I = 1, 4, 5

Figure 11b: GSOLVE results for complex PGT; kinematic results.

--------- COMPLEX PLANETARY GEAR TRAIN KINEMATIC ANALYSIS ----------
SOLUTION FROM GSOLVE VERSION 1.0.0 OHIO STATE UNIVERSITY FEB 1984

SOLUTION TITLE - ANALYSIS OF TRANSAXLE IN FIRST GEAR

DATE OF SOLUTION - 17-FEB-84 6.31.49 AM

MODEL TITLE - FIRST GEAR

DATE OF GEOMETRY ANALYSIS - 17-FEB-84 6.30.45 AM

**** T O R Q U E R E S U L T S ****

INPUT TORQUES	COEFFICIENTS FOR OUTPUT EXTERNAL TORQUES K 2	K 3	K 6	K 7
T 1	2.60000	0.61905	0.00000	0.00000
T 4	-1.60000	0.38095	0.35802	1.45316
T 5	0.00000	0.00000	0.64198	-0.45316

INPUT TORQUES	COEFFICIENTS FOR OUTPUT EXTERNAL TORQUES K 8	K 9
T 1	0.00000	0.00000
T 4	-0.73711	0.57450
T 5	1.73711	0.42550

REQUIRED TORQUE ON LINK X ABOUT ITS ROTATION AXIS IS -

TX = SUMMATION OF (KI * TI), FOR I = 2, 3, 6, 7, 8, 9

Figure 11c: GSOLVE results for complex PGT; torsion results.

REFERENCES

1. Freudenstein, F., and Yang, A. T., "Kinematics and Statics of a Coupled Epicyclic Spur-Gear Train," Mechanisms and Machine Theory, Vol 7., 1972, pp. 263-275.

2. Day, C. P., Akeel, H. A.,and Gutkowski, L. J., "Kinematic Design and Analysis of Coupled Planetary Bevel-Gear Trains", ASME Journal of Mechanical Design.

3. Mabie, H. H., and Ocvirk, F. W., Mechanisms and Dynamics of Machinery, John Wiley and Sons, Inc., New York, 1975.

4. Shigley, J. E., and Uicker, J. J., Theory of Machines and Mechanisms, McGraw-Hill, New York, 1980.

5. Hanson, R. A., "Calculation of Planetary Gear Speeds Using a Building Block Approach," John Deere Industrial Equipment Division - Product Engineering Center, Dubuque, Iowa.

6. Gutkowski, L. J., "A Kinematic Analysis Procedure for General Planetary Gear Trains", Master's Thesis, The Ohio State University, Columbus, Ohio, June 1984.

7. Wakerly, J. F., Microcomputer Architecture and Programming, John Wiley and Sons, Inc., New York, 1981.