

# TECHNICAL RESEARCH REPORT

## Enumeration of Epicyclic-Type Automatic Transmission Gear Trains

*by G. Chatterjee and L-W. Tsai*

**T.R. 95-91**



*Sponsored by  
the National Science Foundation  
Engineering Research Center Program,  
the University of Maryland,  
Harvard University,  
and Industry*



---

# Enumeration of Epicyclic-Type Automatic Transmission Gear Trains

Goutam Chatterjee and Lung-Wen Tsai  
University of Maryland

Reprinted from: **Transmission and Driveline Developments**  
(SP-1032)



The appearance of the ISSN code at the bottom of this page indicates SAE's consent that copies of the paper may be made for personal or internal use of specific clients. This consent is given on the condition, however, that the copier pay a \$5.00 per article copy fee through the Copyright Clearance Center, Inc. Operations Center, 222 Rosewood Drive, Danvers, MA 01923 for copying beyond that permitted by Sections 107 or 108 of the U.S. Copyright Law. This consent does not extend to other kinds of copying such as copying for general distribution, for advertising or promotional purposes, for creating new collective works, or for resale.

SAE routinely stocks printed papers for a period of three years following date of publication. Direct your orders to SAE Customer Sales and Satisfaction Department.

Quantity reprint rates can be obtained from the Customer Sales and Satisfaction Department.

To request permission to reprint a technical paper or permission to use copyrighted SAE publications in other works, contact the SAE Publications Group.



**GLOBAL MOBILITY DATABASE**

*All SAE papers, standards, and selected books are abstracted and indexed in the Global Mobility Database.*

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

**ISSN 0148-7191**

**Copyright 1994 Society of Automotive Engineers, Inc.**

Positions and opinions advanced in this paper are those of the author(s) and not necessarily those of SAE. The author is solely responsible for the content of the paper. A process is available by which discussions will be printed with the paper if it is published in SAE transactions. For permission to publish this paper in full or in part, contact the SAE Publications Group.

Persons wishing to submit papers to be considered for presentation or publication through SAE should send the manuscript or a 300 word abstract of a proposed manuscript to: Secretary, Engineering Activity Board, SAE.

**Printed in USA**

90-1203C/PG



# Enumeration of Epicyclic-Type Automatic Transmission Gear Trains

Goutam Chatterjee and Lung-Wen Tsai  
University of Maryland

## Abstract

An automotive transmission maintains a proper equilibrium between the power and torque produced by an engine and those demanded by the drive wheels. Most automatic transmissions employ some kind of epicyclic gear mechanisms to achieve the above purpose. The first step in the design process of such a mechanism involves finding a configuration that provides a set of desired speed ratios, and meets other dynamic and kinematic requirements. In this work, the kinematic structural characteristics of epicyclic gear mechanisms have been identified, and a methodology is formulated to systematically enumerate all possible configurations of such mechanisms. This is achieved by defining a canonical graph to represent the mechanisms. Graphs of mechanisms with up to ten links have been generated using this methodology.

## 1 Introduction

The power from the engine crankshaft of an automobile is transferred to the drive wheel through a transmission unit, a final reduction unit, and a differential. The ratio of the speed of the input shaft (which brings power from the engine) to that of the output shaft of a transmission unit is called the *speed ratio* or the *reduction ratio*. Most of the automatic transmission units use epicyclic gear trains\* (EGTs) to achieve the desired reduction ratios. Some of them use one degree-of-freedom (dof) EGTs. Others use fractionated mechanisms, each fraction of which is a one-dof EGT. Tsai, et al. [14] have identified some of the structural characteristics required by an EGT to qualify for automatic transmissions, and have shown that of the many non-isomorphic graphs of the six-link one-dof EGTs, only six graphs can be used in automatic transmission gear boxes.

The EGT used in an automatic transmission is supported by bearings housed in the casing. This results in

\*A few use countershaft type of arrangement that is typical of manual transmission. A hybrid type using a combination of both is also possible.

a two-dof fractionated mechanisms. Henceforth, we will call the mechanism formed by an EGT and the casing of an automatic transmission gear box an *Epicyclic Gear Mechanism* (EGM).

## 2 Graph Representation of an EGT

The graph representation of an EGT has been described by Buchsbaum and Freudenstein in their pioneering paper [2]. The characteristics associated with the graph of an EGT have also been derived in the same paper and have been termed *fundamental characteristics*.

### 2.1 Isomorphism

Two graphs are isomorphic if there exists a one-to-one correspondence between their vertices and edges which preserves the incidence and labeling. The adjacency matrices of two isomorphic graphs can be different depending on the numbering of their vertices.

A reliable method for identifying isomorphism is to develop a unique code for each graph, such that two isomorphic graphs have the same code while two non-isomorphic graphs have different codes. This usually involves finding a way of uniquely numbering the vertices. The *degree code* formulated in [12] may be cited as an example employing this idea.

### 2.2 Pseudo-Isomorphism

Since an edge in a graph represents a joint between two links, only binary joints can be represented in a graph. Thus, a ternary joint is represented as two binary joints, a quaternary joints by three binary joints, and so on. This, however, creates a problem in uniquely representing a mechanism. For example, the mechanism shown in Fig. 1(a) can be reconfigured into the mechanism shown in Fig. 1(c) by rearranging the revolute joints among its coaxial links. Though these two mechanisms appear to





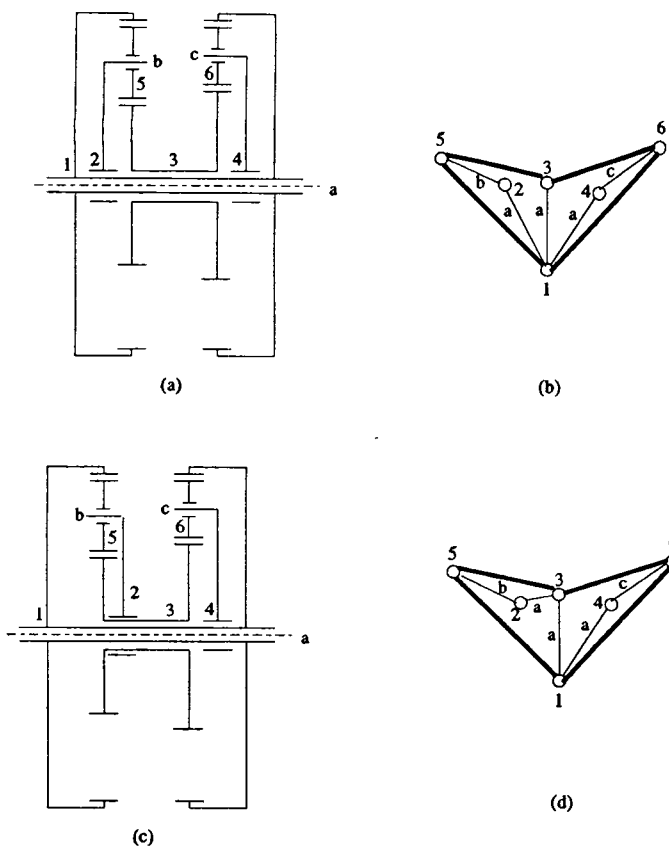


Figure 1 EGTs: (a, c) functional representations, (b, d) corresponding graph representations.

be structurally non-isomorphic [10], they are kinematically equivalent, and for the purpose of structural synthesis are considered the same. The graphs of the two mechanisms (Figs. 1(b) and (d) respectively) are mathematically non-isomorphic. The graph in Fig. 1(d) can be formed from the graph of Fig. 1(b), if the thin edge joining the vertices 1 and 2 is replaced by a thin edge of the same label joining vertices 2 and 3. This method of creating mathematically non-isomorphic graphs representing kinematically equivalent EGTs by replacing a thin edge by another thin edge of the same label is known as *vertex selection* and such mathematically non-isomorphic graphs are called *pseudoisomorphic graphs* [15]. The problem of pseudoisomorphism can be averted by imposing some rules that result in unique arrangement of the edges of the same label. Such a graph is called a *canonical graph* [13].

### 3 Canonical Graph Representation

An EGM typically consists of a one-dof EGT supported by the casing on one axis. Only the links that are connected to the casing by coaxial revolute joints can be used as input, output or fixed links [14].

The casing of an EGM is a unique link in its kinematic structure. Therefore, in the canonical graph representation of an EGM, the vertex representing the casing will be marked as the root of the graph. Recall that the coaxial

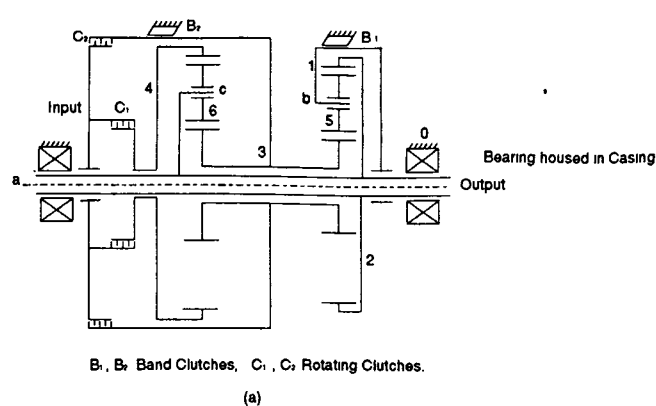


Figure 2 EGM employing Simpson gear set: (a) functional representation, (b) conventional graph representation, (c) canonical graph representation.

joints in a mechanism can be rearranged without affecting the mechanism functionally [13]. Among various arrangements of the coaxial joints there exists a unique configuration such that all the thin edged paths originating from the root and ending at all the other vertices will have distinct edge labels. This unique graph representation is called the canonical graph representation. Using canonical graph representation, the vertices can be divided into several levels. The casing is denoted as the *ground level* vertex. A vertex that is connected by only one thin edge to the root is defined as a *first level* vertex. A vertex that is connected to the root by two thin edges is defined as the *second level* vertex and so on. Each vertex at any particular level is connected to exactly one vertex at the immediate preceding level by a thin edge. All thin edges having the same label\* must be incident to one common lower level vertex. Henceforth, all the vertices at a particular level that are connected to a lower level vertex by thin edges of the same label, will be referred to as members of a *family*. The problem of pseudo-isomorphism does not arise once the canonical graph is defined. The canonical graph representation of the Simpson gear set shown in Fig. 2(a), is shown in Fig. 2(c). The second level vertices represent the planet gears.

### 4 Structural Characteristics

The canonical graph by virtue of its definition has one special feature, i.e.

\*Note the difference between *label* and *level*. A *label* denotes the location of the axis of a link in space while the word *level* denotes the location of a link in the kinematic chain relative to the casing.



C1: All the thin edges of the same label should be incident to a common lower level vertex.

All the fundamental characteristics of a graph of an EGT as described in [2] apply to the canonical graph of an EGM. These are described in section 4.1 under the heading *General Characteristics*. An EGM, besides possessing the characteristics of an EGT, also has its own specific characteristics because it is a mechanism that performs some special functions. These characteristics and their expressions in a canonical graph representation are discussed in Sections 4.2 through 4.4.

## 4.1 General Characteristics

The canonical graph of an EGM has no articulation point. It possesses the following characteristics :

C2: If there are  $n$  vertices in the canonical graph of an EGM then it must have  $n - 1$  thin edges and  $n - 3$  geared edges, since an EGM is a two-dof mechanism.

C3: The subgraph formed by removing the geared edges is a tree.

C4: A geared edge can *only* be incident with one of the following pairs of vertices.

- (a) Two vertices at the same level that are connected to the same lower level vertex by thin edges of different labels.
- (b) Two vertices at adjacent levels, that are connected by a path of exactly three thin edges having two different labels.
- (c) Two vertices one at level  $k$  and another at level  $k - 2$ , if there is a path of exactly two thin edges between them. For every vertex at level  $k$  there is only one vertex at level  $k - 2$ , to which it can be connected by a geared edge.

## 4.2 Coaxial Links

The first level vertices in the canonical graph of an EGM represent links that are connected to the casing by coaxial revolute joints. None of the links of an EGM should be connected to the casing by a gear joint\*. Therefore,

C5: The first level vertices are connected to the root by thin edges of the same label.

C6: No geared edge can be incident to the root.

In an EGM one of the coaxial links is permanently designated as the output. The desired reduction ratios are obtained by changing the input and the fixed links. Also, it is always possible to achieve a direct drive by locking all the links in the EGT together such that they rotate as a single link. Thus if  $N_e$  links of an EGT are coaxial, then it is possible to get  $(N_e - 1)(N_e - 2) + 1$  number of speed reductions. Therefore,

\*Note that it is possible to have one of the links of an EGT permanently fixed to the casing. This, however, will reduce the flexibility of obtaining more speed ratios from the gear train.

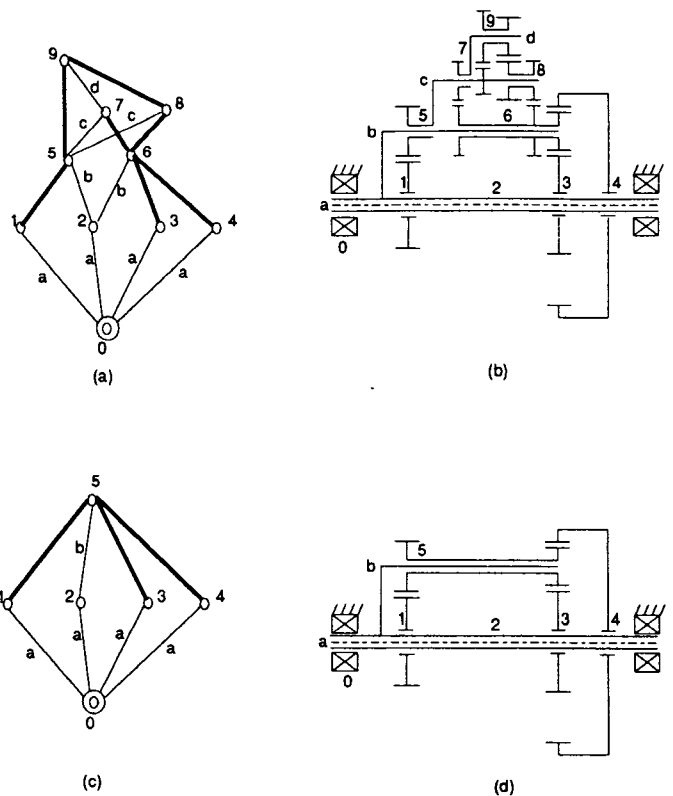


Figure 3: (a) A canonical graph containing a locked chain, (b) functional representation of the mechanism, (c) graph obtained by removal of locked chain, (d) functional representation of the resulting mechanism.

C7: If  $N_r$  is the number of required speed reductions (including reverse), then the number of first level vertices  $N_e$  in the canonical graph must satisfy the equation

$$N_r \leq (N_e - 1)(N_e - 2) + 1 \quad (1)$$

## 4.3 Locked Chains

A set of links forming a part of a mechanism is said to be locked if they undergo no relative motion when the mechanism is in operation and, hence, can be replaced by one link without altering the functional characteristics of the mechanism.

Consider the canonical graph of an EGM shown in Fig. 3(a) that satisfies all the fundamental characteristics described in [2]. However, the subgraph formed by removing vertices 0, 1, 2, 3, and 4 from the graph represents a kinematically locked chain. This is because the subgraph has 5 vertices but 4 geared edges. Fig. 3(c) shows a graph formed by replacing vertices 5, 6, 7, 8, and 9 in Fig. 3(a) by a vertex 5, that will perform the same functions.

C8: An EGM contains a locked chain if there exists a subgraph of  $p$  vertices in the graph of the EGM such that

- (a) the transfer vertex of each geared edge in the subgraph lies in the subgraph, and



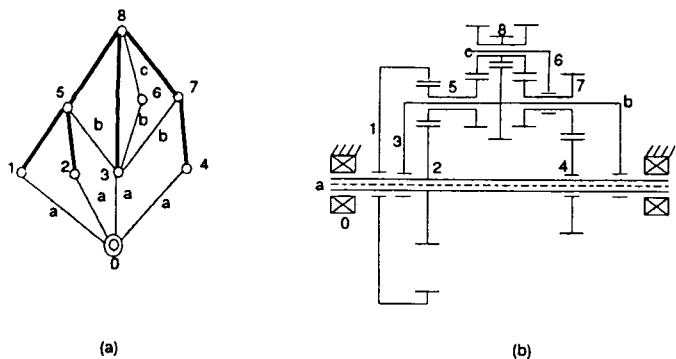


Figure 4 Canonical graph of an EGM with a binary carrier represented by vertex no. 6.

(b) the number of geared edges in the subgraph is more than  $p - 2$ .

A methodology is described in Section 7.2 that prevents the generation of EGM with locked chains during the enumeration process.

#### 4.4 Redundant Links

The other desired feature of a viable mechanism is that it should not have any *redundant* link. A link is said to be redundant, if it is never used as an input, output or a fixed link, and the removal of such a link does not change the degrees of freedom of the EGT. Such a link will not carry power during the operation of the mechanism at any of its reductions.

A one-dof EGT communicating, that is, giving and taking power, with the external environment requires at least an input, an output and a reaction (fixed) link. Thus, to function effectively it requires at least three ports of communication with the external environment. Similarly, a two-dof EGT requires at least four ports of communication, a three-dof EGT requires five ports of communication, and so on.

In the canonical graph representation of an EGM if there exists a subgraph that represents an  $n$ -dof EGT, then it must have at least  $n+2$  ports of communication with the external environment. Otherwise, those links that can not interact with the external environment would be redundant. The external environment includes both the rest of the gear train and the outside world. Two things are to be noted here.

1. The subgraph of a canonical graph represents an EGT if and only if the carrier of any gear pair within the subgraph is also a member of the subgraph. For example, the subgraph formed from the canonical graph of Fig. 4(a) by deleting vertices 0, 1, 2, 4, and 6 does not represent an EGT.
2. Some of the ports of communication at a first glance may not be obvious. For example, in the graph shown in Fig. 4(a), let vertex 1 represent the input link, vertex 2 the fixed link, and vertex 4 the output link. Consider the subgraph formed by deleting vertices 0,

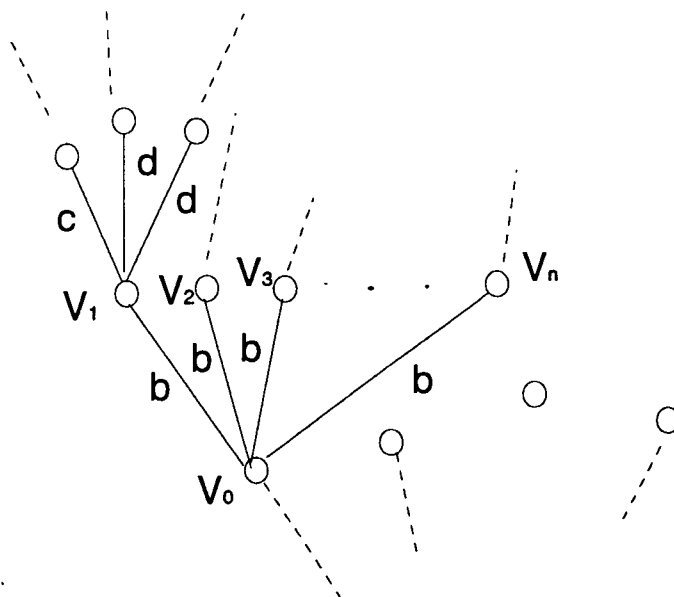


Figure 5 A tree obtained by removing the geared edges from a canonical graph

1, 2, 4, 5, and 7. It appears as if the subgraph has only two ports of communication, namely 3 and 8. However, the carrier represented by vertex 6 is also a port of communication. This is because vertex 6 is the transfer vertex of the geared pair connecting vertices 7 and 8, and vertex 7 is external to the subgraph. Therefore, the subgraph has actually three ports of communication.

From the above observations we will derive several conditions that the canonical graph of an EGM must satisfy. These conditions are necessary, but not sufficient. However, these conditions drastically reduce the number of graphs with redundant links during the enumeration process. The remaining few can be weeded out by inspection or by a methodology described later.

##### 4.4.1 Number of Branches

Consider the branches of a tree of a canonical graph emanating from vertex  $V_0$  as shown in Fig. 5. Let the level immediately above and arising from  $V_0$  contain  $n$  vertices  $V_1, V_2, \dots, V_n$  that belong to one family. Let the branches emanating from these vertices have  $N_1, N_2, \dots, N_n$  vertices, respectively. Now vertex  $V_1$ , and the branches emanating from it form a subgraph that represents an EGT, because a geared edge joining any two vertex of the subgraph will have its transfer vertex contained in the subgraph. Let all such subgraphs be named  $G_1, G_2, \dots, G_n$ , and let them have  $F_1, F_2, \dots, F_n$  dof, respectively. Then  $G_i$  will have  $N_i - F_i$  geared edges and will require at least  $F_i + 2$  ports of communication. One port of communication is the vertex  $V_i$  itself. The rest of the ports of communication will communicate through gear edges with any of the vertices  $V_0$  to  $V_n$  other than  $V_i$ . A geared edge coming from any port of  $G_i$  cannot connect to any vertex other than  $V_0$  to  $V_n$  of the EGM.



Next consider the subgraph  $G_0$  formed by vertices  $V_0$  to  $V_n$  and the branches emanating from  $V_1$  to  $V_n$ . The minimum number of geared edges that  $G_0$  should have in order that none of the links represented by vertices in the subgraphs  $G_1, G_2 \dots G_n$ , is rendered redundant can be calculated by summing the number of geared edges in each subgraph, and the number of geared edges required by each subgraph to maintain the minimum number of ports of communication. Thus the number of geared edges in  $G_0$  should be at least

$$\sum_{i=1}^n (N_i + 1 - F_i - 1) + \sum_{i=1}^n (F_i + 1) = \sum_{i=1}^n N_i + n \quad (2)$$

The number of geared edges in a subgraph representing an EGT must be less than the number of vertices by 2 or more, otherwise it will be locked. Since  $G_0$  has  $\sum_{i=1}^n N_i + n + 1$  vertices, this condition can not be satisfied for the above case. Therefore, not all the members of a family can give rise to branches. If one of the members does not give rise to any branch, then the number of vertices in  $G_0$  will be  $\sum_{i=1}^{n-1} N_i + n + 1$  and the minimum number of geared edges required to prevent redundancy will be at least  $\sum_{i=1}^{n-1} N_i + n - 1$ . Thus, if one of the members does not give rise to any branch, the above condition is satisfied. Also, if more than one member of a family does not give rise to any branch the above condition can be satisfied.

Let  $V_1, V_2 \dots V_n$  represent the first level vertices, and vertex  $V_0$  represents the root of the EGM. Since an EGM has exactly two-dof, the number of geared edges must be less than the number of vertices by 3. Applying this condition and following the above logic one can prove that there must be at least two vertices in the first level which should not give rise to any branch.

Thus, we get the following two conditions.

C9: For those vertices located at the higher levels, there must be at least one member in a family that does not give rise to any branch.

C10: There must be at least two vertices in the first level that do not give rise to any branch.

As a special case of Condition 1, a vertex cannot give rise to any branch if it is the only vertex in a family.

#### 4.4.2 Number of Incident Edges

The subgraph formed by vertices 3, 6, and 4 in Fig. 6 represents a one-dof EGT. Hence it should have at least three ports of communication. But it has only two ports of communication, namely vertices 3 and 4, and the removal of vertex 6 does not change the dof of the mechanism. Therefore, vertex 6 represents a link that is redundant and can be removed. This is generally true for any link that is not used as the input, output or fixed link and is connected by only one revolute joint and one gear joint. Thus,

C11: If a vertex is not located at the first level and is incident by only one thin edge, then it must be incident by at least two geared edges.

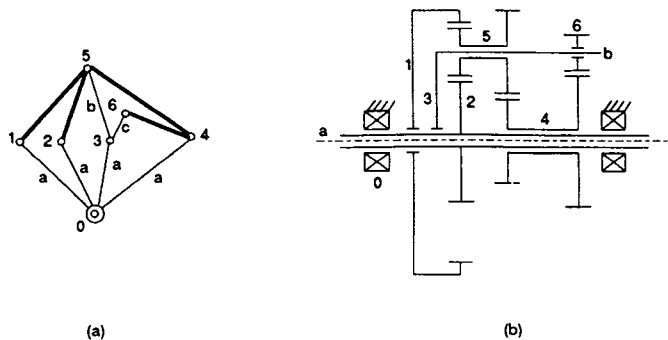


Figure 6: (a) A canonical graph containing a vertex (vertex no. 6) representing a redundant link, (b) functional representation of the mechanism.

C12: If a vertex is located at the first level and is incident by only one thin edge, it must be incident by at least one geared edge.

#### 4.4.3 Floating Carrier

Consider the graph shown in Fig. 7(a). The subgraph formed by vertices 3, 5, 6, 7 and 8 represents a two-dof EGT. However, it has only three ports of communication, namely 3, 5, and 7. Therefore, the links represented by vertices 6 and 8 are redundant. Note that the vertex 6 represents a binary vertex that is not connected to any other vertex by a geared edge. According to Condition C9, a binary vertex of this kind can only occur at the penultimate level of a branch. If the vertex at the higher level that is connected to the binary vertex is incident by two geared edges, then there are two possible ways of connecting them. These two ways are shown in Figs. 7(a) and (c). In both these cases the binary vertex, and the higher level vertex that is incident to it are redundant. In Fig. 7(c) links represented by vertices 5 and 7 are redundant because the subgraph formed by vertices 1, 5, 6 and 7 represents a one-dof EGT that has only two ports of communication. Thus,

C13: If a binary vertex in a tree is not at the first level then of the two vertices that it connects, the higher level vertex must be incident by more than two geared edges.

Fig. 4 shows the graph of an EGM that has a binary vertex, but no redundant links. In this case the higher level vertex (vertex No. 8) that is connected to the binary vertex 6 is incident by three geared edges.

## 5 Graph Enumeration

Most combinatorial enumeration procedures that require the enumeration of all possible solutions satisfying certain constraints, are done through the process of *generation and testing*. The procedure is thus divided into two parts [5]: a generator of all possible solutions and a tester that selects only those solutions that meet the constraints.





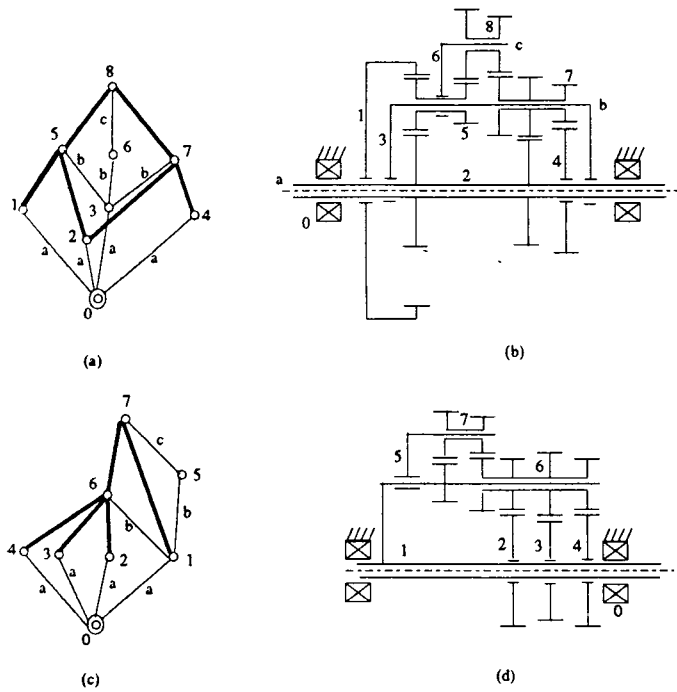


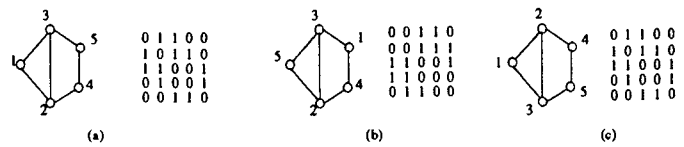
Figure 7: (a, c) Canonical graphs containing vertices (No. 6 & 8 in (a) and No. 5 & 7 in (c)) representing redundant links, (b, d) functional representation of the mechanisms.

The enumeration procedure described below uses a hierarchical generation and testing technique. A part of the canonical graph is generated at each step, and a test is carried out to prune out those solutions that will not give rise to canonical graphs with the desired characteristics.

The following observations are made before formulating an efficient enumeration procedure.

1. The canonical graph has a unique vertex - the root, with reference to which other vertices are divided into several levels. Thus, there is already some arrangement among the vertices. One can therefore think of obtaining a unique arrangement of vertices by adding some rules. This can then be used to develop a unique code for each graph that will serve as a reliable tool for an isomorphism test.
2. The definition of the canonical graph requires the edge labels to be distributed in a particular way. This forbids the generation of pseudo-isomorphic graph.
3. Most of the characteristics described in Section 4 are applicable to the tree of a canonical graph, rather than the canonical graph as a whole.
4. Characteristic C4, which prescribes the allowable geared edge connections, presumes the existence of a labeled tree.

Therefore, it follows that an efficient enumeration procedure can be achieved if it is divided into two phases. In the first phase labeled trees that will give rise to admissible canonical graphs are enumerated. In the second phase geared edges are added to these trees to create the canon-



Note: The matrix beside each graph is the link to link adjacency matrix.

Figure 8: Graphs in (a) and (b) are isomorphic but not automorphic, Graphs in (a) and (c) are automorphic.

ical graphs. Each of these phases comprises of various steps, that are described in Sections 6 and 7, respectively.

## 6 Enumeration of Trees

The characteristics C1, C2, C5 to C10 are used to formulate the procedure for enumerating trees with  $n$  vertices.

**Step 1.** From C7 calculate the minimum number of coaxial links required. This gives the minimum number of vertices that a tree should have at the first level.

**Step 2.** Distribute the remaining vertices into various levels. While making such a distribution, remember that all levels other than the highest level must have at least 2 vertices in order to satisfy C9.

**Step 3.** Divide the vertices at each level into families (see Section 3). At all levels other than the highest level there must be at least one family with two or more members in order to satisfy C9. One must note that the distribution of vertices into families is same as the problem of partitioning of integers. An algorithm to this end can be easily developed by using the concept of generating functions, explained in the book by Liu [9].

**Step 4.** Start adding the vertices of the first level. According to C5 all the vertices at the first level should be connected to the root with thin edges of the same label.

**Step 5.** Next connect the vertices of the second level to the first level vertices. According to C10, two of the vertices at the first level should not give rise to any branch. The definition of a canonical graph requires the members of the same family to be connected to the same lower level vertex by edges of the same label. Therefore, while adding the second-level vertices, add one family at a time. Start with the family that has the largest number of members. If a family of vertices is added in all possible ways a lot of isomorphic graphs would be generated. To reduce the number of isomorphic graphs we digress and introduce the concept of graph automorphism and similar vertices.

### 6.1 Graph Automorphism

Consider the graph shown in Fig. 8. The edges of the graph are unlabeled and its vertices are numbered. If we permute the numbering of the vertices, isomorphic graphs



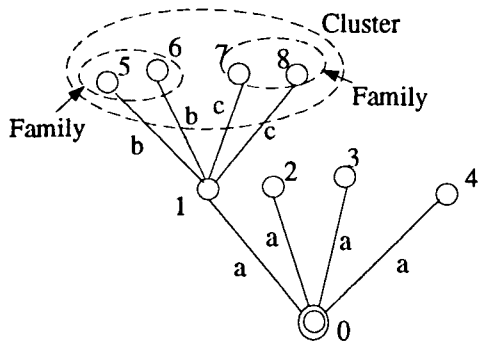


Figure 9 A tree showing similar vertices.

are produced. Most of these isomorphic graphs have their corresponding vertices numbered differently. However, some specific permutations produce graphs whose corresponding vertices bear the same number as the original one. These graphs are called *automorphic graphs*. For example, if we number the vertices 1, 2, 3, 4, and 5 of the graph in Fig. 8(a) as 1, 3, 2, 5, and 4, the resulting graph as shown in Fig. 8(c) is automorphic. The permutation in this case is denoted by  $(1)(2,3)(4,5)$ . Elements 2 and 3 are said to form a cycle of length 2, and element 1 a cycle of length 1. All such permutations that produce automorphic graphs form a group [9]. Each of these permutations is referred to as a *member* of the group. The application of any of the members from the group won't alter the adjacency matrix of the graph in any way. If two vertices  $p$  and  $q$  are contained in the same cycle of any member of such a permutation group, then vertices  $p$  and  $q$  are said to be similar [16]. Thus, one can divide the vertices of a graph into classes by putting the similar vertices together. If one wants to add a particular property to any vertex, then there can be one choice from a class of similar vertices since any choice is as good as the other.

Some very simple rules are prescribed here to identify some of the similar vertices. These rules won't prevent the generation of isomorphic graphs completely, but will reduce their number drastically. These rules are

**S1:** Vertices belonging to the same family and incident by only one thin edge and no other edges are similar. For example, vertices 2, 3 and 4 in Fig. 9 are similar.

**S2:** Families that have the same number of members and are connected to the same lower level vertex form a *cluster*. If none of the vertices in a cluster is incident by more than one thin edge, then all the vertices in the cluster are similar. For example, vertices 5, 6, 7 and 8 in Fig. 9 are similar. If a vertex is incident by more than one edge, then the family of vertices to which that vertex belongs becomes dissimilar, while the rest of the families in the cluster remain similar.

Whenever a choice for a vertex is to be made for adding a vertex, or a family of vertices, choose only one vertex out of a class of similar vertices.

The addition of vertices at the higher levels should proceed in the same way as the second level vertices. The

Table 1 Distribution of vertices of the graphs representing 9-link EGMs into levels and families: (a) level distribution, (b) family distribution.

Level I	7	6	5	5	4	4	4
Level II	1	2	3	2	4	3	2
Level III				1		1	2

(a)

Levels	Distribution of vertices into levels	Distribution of vertices into families
I	4	4
II	2	2
III	2	2
		1,1

(b)

only difference is that C9 is applicable instead of C10, i.e., at least one member in each family of vertices should not give rise to any branches. A test to eliminate isomorphic graphs is carried out after completing the addition of vertices at each level.

The example given below demonstrates the above enumeration method.

In this example, trees that can give rise to admissible canonical graphs which represent 9-link EGMs, capable of providing four speed reductions, are enumerated.

**Step 1.** There are nine links. The minimum number of coaxial links required is four. Therefore, the first level should have at least four vertices.

**Step 2.** The remaining vertices can be divided into levels as shown in Table 1(a).

**Step 3.** The vertices at each level are further divided into families. One of the distribution is chosen from Table 1(a) to demonstrate this step. The distribution of vertices at each level into families for the chosen distribution is shown in Table 1(b).

**Step 4.** This step and the next one are demonstrated by choosing one of the two distribution of vertices into families from Table 1(b). The distribution chosen is shown in Fig. 10(a). Fig. 10(b) and (c) shows the tree and the corresponding adjacency matrix formed after addition of the first level vertices. The adjacency matrix has all its elements zero except for those in the rows or columns corresponding to the root.

**Step 5.** In the second level there is only one family with two members. There is only one choice of vertex in the first level to which this family can be connected since all the vertices at the first level are similar according to S1. The resulting tree is shown in Fig. 10(d). In terms of the adjacency matrix this would mean the addition of two rows and columns to the matrix of Fig. 10(c). The resulting matrix is



Level	0	I	II	III
Family	1	4	2	1

(a)

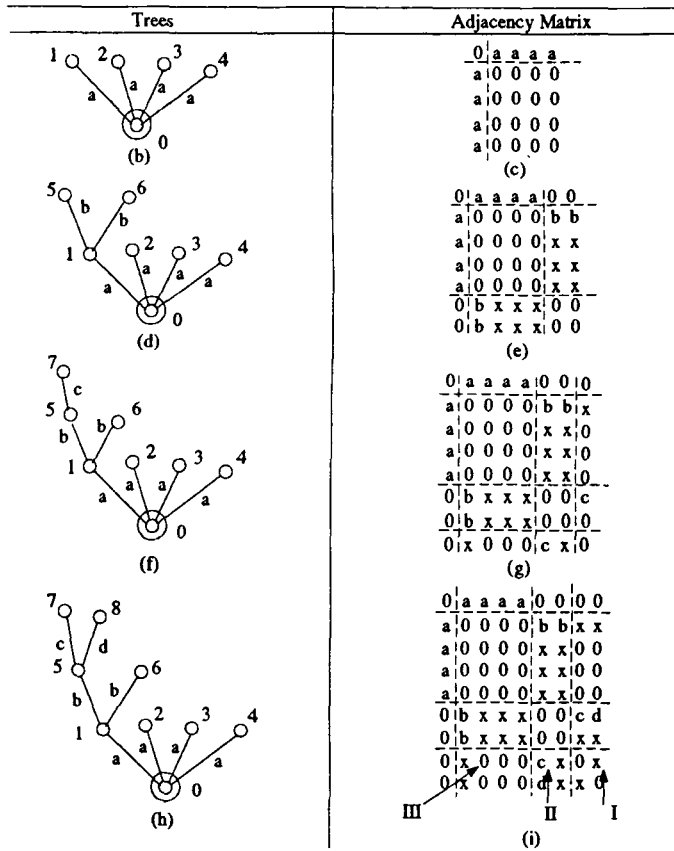
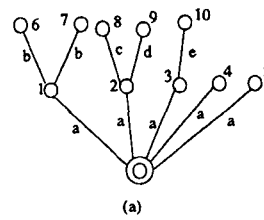


Figure 10 Enumeration of trees for a given family and level distribution: (a) level and family distribution, (b) and (c) addition of first level vertices, (d) and (e) addition of second level vertices, (f) - (i) addition of third level vertices.

shown in Fig. 10(e). All the elements of the new rows and columns are zero except for element number\* 1. However, some of the zero labeled elements will be relabeled  $g$  when geared edges are added. Potential geared edge connections can be found from C4. The corresponding elements, instead of being set to zero, are therefore set to  $x$  to facilitate the process of geared edges addition in the second phase. The elements  $A_{56}$  and  $A_{65}$  are set to zero because there can be no geared edge between vertices of the same family according to C4(a). All other elements are set to  $x$  since they satisfy C4(b). This completes the addition of vertices at the second level.

The addition of the vertices at the third level is illustrated in Fig. 10(f) to 10(i). The dashed lines across the adjacency matrices divide them into various sub-matrices, each containing information about a particular type of interactions. For example, sub-matrix I in Fig. 10(i) represents the interaction within

\*We are following the convention of C programming language in indexing the elements of rows and columns. The indexing starts with the number 0.



Level	Family	Cluster	Priority
0	(0)	(0)	(0)
I	(1,2,3,4,5)	(1,2,3,4,5)	(1)(2)(3)(4,5)
II	(6,7)(8)(9)(10)	(6,7)(8,9)(10)	(6,7)(8,9)(10)

(b)

Figure 11 A tree having its vertices numbered according to their priority.

the third level vertices, submatrix II the interaction between the third and second level vertices, and submatrix III the interaction between the third and first level vertices. Whether an element of the submatrices I, II, and III can be converted into  $x$  or not can be determined by applying C4(a), C4(b), and C4(c), respectively.

Before ending this section on enumeration of trees we note that the generator generates only those trees that have the desired characteristics. The tester has only to identify the isomorphic graphs whose number has been reduced by incorporating some rules in the generator.

## 6.2 Isomorphism

The issue of developing unique code for identifying isomorphic graphs has been addressed by many authors ([1], [12]). Most of these papers dealt with graphs whose edges are not labeled. The labeling of edges of a graph has both its advantages and disadvantages. On one hand, it divides the vertices into classes that we have already named as families and, therefore, introduces some amount of ordering among the vertices. On the other hand, since the labels are arbitrary they have to be permuted in all possible ways in order to detect isomorphism. Some papers have presented graph representations [11] that obviate the need to explicitly represent the labels of the revolute edges in the adjacency matrices. This paper achieves the above objective by proposing four simple rules.

- I1 Vertices at the lower levels should have higher priority than those at higher levels. For example, vertices 1, 2, 3, 4, and 5 in Fig. 11 have higher priority than vertices 6, 7, 8, 9, and 10. Hereafter, whenever we say that a vertex has a higher priority than another it means that the former is numbered lower than the latter.
- I2 Members of a family such as vertices 1, 2, 3, 4, and 5 in Fig. 11, should be consecutively numbered.
- I3 All members of the families that belong to the same level and have the same number of members should be consecutively numbered. For example, vertices 8, 9, and 10 in Fig. 11 are consecutively numbered.
- I4 Families that have more members, have higher priority than those having less. Vertices 6 and 7 in Fig. 11 have higher priority than vertices 8, 9 or 10.



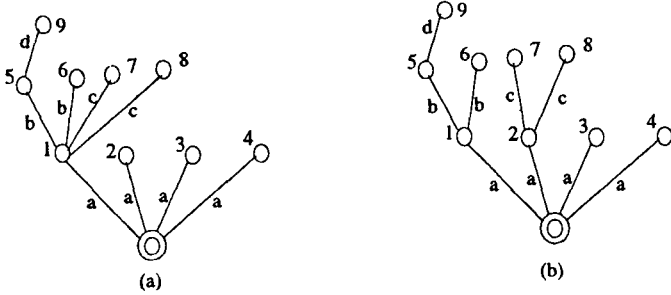


Figure 12 Trees having their vertices numbered in order of their priority.

If we number the vertices following the above rules, then there is no need to explicitly label the edges. One can uniquely determine the labels of the edges from the level and family distributions.

Next we propose a set of rules that decide the priorities of vertices within a family and the priority of a family of vertices over another having the same number of members. These rules should be applied successively in the order stated below and should not alter the priorities already decided by the application of previous rules. These rules are:

- I5** A vertex in a family precedes another vertex if it gives rise to more families with a higher number of members. For example, in Fig. 11 vertex 1 precedes vertex 2, which in turn precedes vertex 3.
- I6** To determine the order of families having the same number of members, the vertices of the families are compared. If the highest priority vertex of family *A* gives rise to more families with a higher number of members than that of family *B*, then *A* precedes *B*. If the precedence cannot be determined by comparing the highest priority vertices of the two families then the next priority vertices are compared. For example, vertices 5 and 6 precede vertices 7 and 8 in Fig. 12(a).

To develop the code all labels of the thin edges are replaced by 1 in the adjacency matrix. Then, the vertices are permuted to maximize the number formed by concatenating the elements of the upper triangular matrix (of the adjacency matrix), starting from the leftmost element of the topmost row and moving along row by row downwards. This maximized number, along with the level and family distribution gives a unique code for isomorphic trees. Since the permutation of similar vertices does not change the adjacency matrix, only dissimilar vertices of a family that have the same priority are permuted. Families that have the same priority are permuted en bloc.

The isomorphism test is applied after completing the addition of vertices at each level. Rules I1 to I4 are applied to arrange the vertices in the level that has just been added, whereas rules I5 and I6 are applied to arrange the vertices in the immediately preceding level.

## 7 Enumeration of EGMs

The adjacency matrices of the trees that have been enumerated until now have some of their elements labeled *x*. The addition of geared edges means relabeling some of these *x*'s as *g*'s, and the rest as zeroes. We describe below an algorithm to find the transfer vertex associated with each gear pair, and study the interaction among fundamental circuits formed by the addition of geared edges.

### 7.1 Locating the Transfer Vertex

Consider the tree and its adjacency matrix shown in Fig. 13. Suppose, a geared edge is added between vertices 7 and 6. To find the associated transfer vertex scan the row corresponding to the higher level vertex (vertex 7 in this case). If both the vertices are at the same level, then scan the row corresponding to any of the vertices. The column number corresponding to the first non-zero element that is not an '*x*' or a '*g*' gives the number of the transfer vertex. In case of the above example element no. 5 of row 7 gives the number of the transfer vertex. A fundamental circuit is characterized by the two end vertices of a geared edge and the associated transfer vertex. Therefore, once the transfer vertex is known, the fundamental circuit is in effect known.

### 7.2 Interaction Among Fundamental Circuits

The three vertices that characterize a fundamental circuit form a simple one-dof EGT with three links. In order to keep track of the interactions among fundamental circuits we construct a matrix whose column number corresponds to the vertex number. In the first row of this matrix we mark the elements that correspond to the characteristic vertices of the first fundamental circuit by a label, say 1. For example, if we connect the vertices 7 and 8 of the tree shown in Fig. 13(b) by a geared edge, then the matrix will take the form shown in Fig 14(a).

If the next fundamental circuit formed shares two of its vertices with the existing one then the two fundamental circuits will constitute a subgraph that represents a one-dof EGT. In the first row of the above matrix we label the element corresponding to the non-common vertex. For example, in Fig. 13(g), due to the fundamental circuit formed by the addition of the geared edge between vertices 6 and 8 the first row of the above matrix is modified as shown in Fig. 14(b). However, if the two fundamental circuits have one vertex in common we add a new row to the above matrix and label the elements corresponding to the characteristic vertices of the newly formed fundamental circuit. Thus, each row of the matrix corresponds to a subgraph that represents a one-dof EGT.

In general if we consider a graph in which *k* geared edges have been added, then the fundamental circuit formed due to the addition of the *k* + 1 geared edge can have the following relationships with any of the existing subgraphs.

1. It can have one of its vertices in common with a subgraph representing a one-dof EGT. In this case add





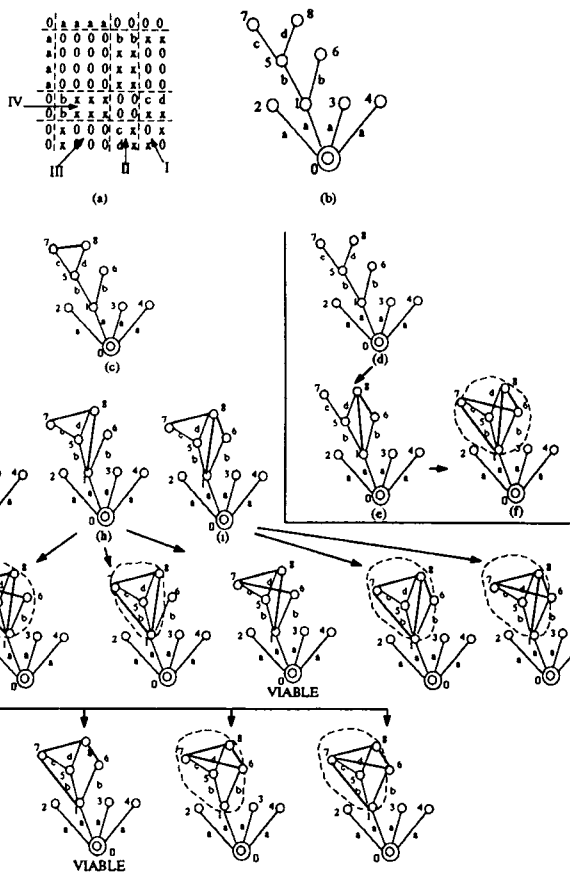


Figure 13 Step by step enumeration of EGMs from a tree - Part I.

a new row to the matrix, and label the elements corresponding to the characteristic vertices of the newly formed fundamental circuit.

- It can have two of its vertices in common with a subgraph representing a one-dof EGT. In this case, modify the row corresponding to the existing subgraph by labeling the element that corresponds to the non-common vertex of the fundamental circuit that has just been added. If the modified subgraph now shares two of its vertices with another existing subgraph combine the two corresponding rows into one, since the two subgraph together represent a one-dof EGT.

Repeat this process until no two subgraph have more than one vertex in common.

A new fundamental circuit formed cannot have three of its vertices in common with a subgraph representing a one-dof EGT, otherwise the mechanism will be locked.

The above observations will be used to keep a track of the subgraphs that are being formed, and to prevent the occurrence of locked chains.

### 7.3 Similar Edges

Consider the tree shown in Fig. 13. The similar vertices in the tree can be identified by applying S1 and S2 as defined in Section 6.1. Because of the similarity among

Vertices	1	2	3	4	5	6	7	8
					1		1	1

(a)

Vertices	1	2	3	4	5	6	7	8
					1	1	1	1

(b)

Figure 14 Matrices to keep track of the interaction among fundamental circuits.

vertices some of the candidate geared edges (represented by label  $x$  in the adjacency matrix) are similar. Some of these sets of similar geared edges can be identified by the application of the rule given below.

- S3** When several geared edges connect a common vertex to a set of similar vertices, they form a similar edge set.

### 7.4 Addition of Geared Edges

The method of adding geared edges can be formulated as follows.

**Step I.** First add geared edges connecting vertices at the highest level. For every geared edge that is being added check whether the addition of the geared edge results in a locked chain by the method described above. If it does, then set the label  $x$  that corresponds to the geared edge in the adjacency matrix to zero.

**Step II.** Next add geared edges from the highest level to the lower levels. Before doing this calculate the minimum number of geared edges to be added to each of the vertices at the highest level from C11. If the highest level is the second level, then the minimum number of geared edges to be incident on a first level vertex as given in C12 should also be taken into account. Care should be taken that the total number of geared edges to be added does not exceed that given by C2. As before, check for locked chains for every geared edge added.

**Step III.** Repeat steps I and II for the next lower level vertices, i.e, the vertices that are at one level immediately below the highest level.

Repeat step III until the second level is reached.

The methodology for addition of geared edges is illustrated in Fig. 13. Geared edges are added to the tree (Fig. 13(b)) starting from the highest level and continuing downward.

At the end of enumeration procedure three EGMs are formed. They are shown in Fig. 15(d), (e), and (f). The one shown in Fig. 15(d) has redundant links. This is because the subgraph formed by vertices 1, 5, 6, 7 and 8



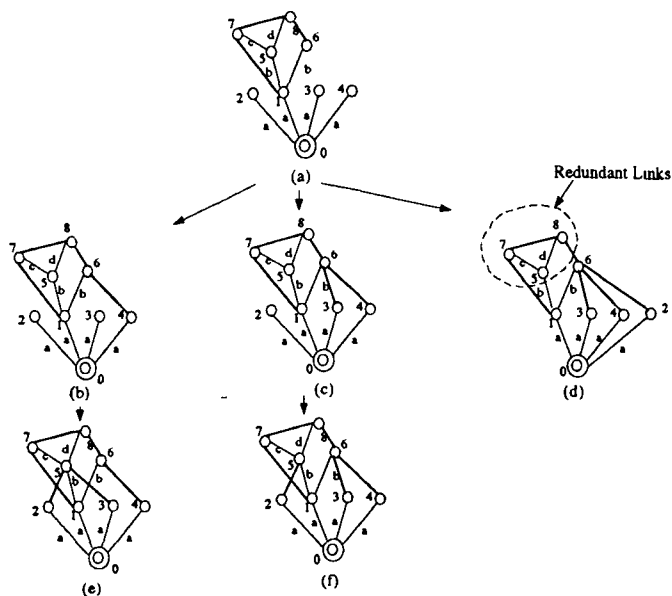


Figure 15 Step by step enumeration of EGMs from a tree - Part II.

represents a one-dof EGT that has only two ports of communication, i.e., vertices 1 and 6. The generation of such graphs can be prevented if we ensure that every subgraph (formed at the end of step II) that represents an  $n$ -dof EGT has  $n + 2$  ports of communication. The subgraphs and their dof can be obtained from the matrix that has been developed to prevent the occurrence of locked chains. However, such a verification is not required when geared edges are added to connect the highest level vertices to the lower level vertices, since conformation to C11 and C12 ensures that there will be no redundant links.

#### 7.4.1 Isomorphism

A test to identify isomorphic graphs is performed at the end of each step. To do this we extend the procedure described in Section. 6.2. The rules to identify the priority of vertices are given below. They, however, should not alter the priority set by rules I1 to I6. Also, the rules should be applied in the order given below and should not alter the arrangement set by the previous rules.

- I7 The vertex that is connected to vertices at two levels above it with more geared edges has the highest priority.
- I8 Among vertices of same priority in a family, the vertex that is connected to vertices at one level above it with more geared edges is given higher priority.
- I9 Among vertices of same priority in a family, the vertex that is connected to vertices at the same level with more geared edges is given higher priority.
- I10 Among vertices of same priority in a family, the vertex that is connected to vertices at lower levels with more geared edges is given higher priority.

To develop a code for the graph, the  $g$ 's in the adjacency matrix are replaced by 2's. Then, the vertices are per-

Table 2 Number of graphs enumerated for EGMs with up to 10 links.

No. of links	No. of graphs	No. of graphs with vertices only upto the second level.
6	1	1
7	7	7
8	22	20
9	157	128
10	899	620

mutated to maximize the number formed by concatenating the elements of the upper triangular matrix as described in Section 6.2.

## 8 Results and Discussions

The results are tabulated in Table 2. It has been mentioned that the first level vertices of a canonical graph represent the potential input, output and fixed links while the second level vertices represent the planet gears. A review of the work of Larew [7], Levai [8], Gott [4], and Tsai, et al. [14] had not revealed a single automatic transmission gear box having a link located on the third or higher levels. Since no physical reason could be found for this observation, it was not considered a structural characteristic of such gearboxes. However, Table 2 also lists the number of graphs having vertices only upto the second level. The graphs of 8-link EGMs, are shown in Fig 16. The graphs and adjacency matrices of all the EGMs with upto 9-links are documented in [3].

There is only one graph for 6-link EGMs\*, which is in agreement with the result given in [14]. There are 7 graphs for 7-link EGMs, which is one more than that given in the same paper. This is because that paper has excluded those graphs in which the geared edges form a closed loop. The verification for completeness of the set of graphs enumerated for 8-link EGMs has been accomplished in an indirect way. From the set of graphs of 7-link EGTs generated by Kim and Kwak [6], those that qualify for automatic transmissions were selected. A total of 20 such graphs were extracted from their paper which is less than the 22 given in Table. 2. The reason is that there are exactly 2 graphs of 8-link EGMs (Figs. 16(f) and 16(q)) that have geared edges forming a loop, and those graphs cannot be generated by the method of Kim and Kwak.

## 9 Conclusions

The structural characteristics of epicyclic gear mechanisms (EGMs) that are commonly used in automatic transmissions to obtain various speed ratios have been identified from the view point of kinematics. A canonical graph representation for this type of mechanisms has been defined. A methodology to systematically enumerate

\*An  $n$ -link EGM contains a  $(n - 1)$  link EGT and the casing of a transmission.



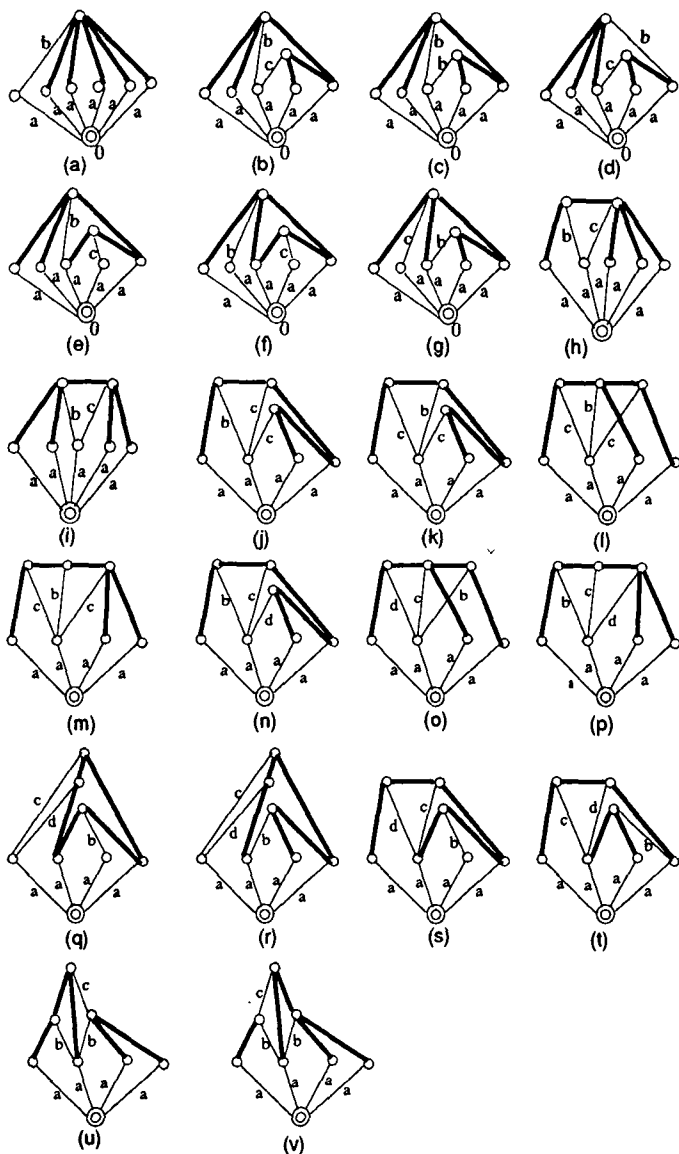


Figure 16: Graphs of 8-link EGMs.

these graphs has been developed and illustrated through various examples. Graphs of EGMs with up to 10-link have been enumerated using this method.

A methodology for automatically sketching the functional schematics of an EGM from its graph representation has been developed. It will be presented in the ASME 1994 Mechanisms Conference. It is hoped that this work will provide a basis for the design of future automatic transmissions.

## References

[1] Ambedkar, A.G., and Agrawal, V.P., 1987, "Canonical Numbering of Kinematic Chains and Isomorphism Problem: MIN Code," *Journal of Mechanism and Machine Theory*, Vol. 22, No. 5, pp. 453-461.

[2] Buchsbaum, F., and Freudenstein, F., 1970, "Synthesis of Kinematic Structure of Geared Kinematic

Chains and Other Mechanisms," *Journal of Mechanisms and Machine Theory*, Vol. 5, pp. 357-392.

[3] Chatterjee, G., 1993, *Enumeration and Automatic Sketching of Epicyclic Type Automatic Transmission Gear Trains*, Master's Thesis, University of Maryland, College Park, Maryland.

[4] Gott, P. G., 1991, *Changing Gears*, Society of Automotive Engineers, Inc., Warrendale, Pennsylvania.

[5] Hayes-Roth, F., Waterman, D.A., and Lenat, D.B., 1983, *Building Expert Systems*, Addison-Wesley, Reading, Massachusetts.

[6] Kim, J.T., and Kwak, B.M., 1990, "Application of Edge Permutation Group to Structural Synthesis of Epicyclic Gear Trains," *Journal of Mechanisms and Machine Theory*, Vol. 25, No. 5, pp. 563-574.

[7] Larew, W. B., 1966, *Automatic Transmissions*, Chilton Book Company, Radnor, Pennsylvania.

[8] Levai, Z., 1964, *Theory of Epicyclic Gears and Epicyclic Change-Speed Gears*, Doctoral Dissertation, Technical University of Building, Civil and Transportation Engineering, Budapest.

[9] Liu, C.T., *Introduction to Combinatorial Mathematics*, McGraw-Hill, New York.

[10] Mruthyunjaya, T.S., and Ravisankar, R., 1985, "Computerized Synthesis of the Structure of Geared Kinematic Chains," *Journal of Mechanisms and Machine Theory*, Vol. 20, pp. 367-387.

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

[12] Tang, C.S. and Liu, T., 1993, "Degree Code - A New Mechanisms Identifier," *ASME Journal of Mechanical Design*, Vol. 115, pp. 627-630.

[13] Tsai, L.W., 1988, "The Kinematics of Spatial Robotic Bevel-Gear Trains," *IEEE Journal of Robotics and Automation*, Vol. 4, pp. 150-156.

[14] Tsai, L.W., Maki, E.R., Liu, T., and Kapil, N.G., 1988, "The Categorization of Planetary Gear Trains for Automatic Transmissions According to Kinematic Topology," SAE Technical Paper # 885602, *Proceeding of the XXII FISTA Congress and Exposition*, Vol. 1., pp. 1513-1512.

[15] Tsai, L.W., and Lin, C.C. 1989, "The Creation of Non-Fractionated Two-Degree-of-Freedom Epicyclic Gear Trains," *ASME J. of Mechanisms, Transmissions, and Automation in Design*, Vol. 111, pp. 524-529.

[16] Yan, H.S., and Hwang, Y.W., 1991, "The Specialization of Mechanisms," *Mechanisms and Machine Theory*, Vol. 26, pp. 541-551.



---

# Enumeration of Epicyclic-Type Automatic Transmission Gear Trains

**Goutam Chatterjee and Lung-Wen Tsai**  
University of Maryland

Reprinted from: **Transmission and Driveline Developments**  
(SP-1032)

The appearance of the ISSN code at the bottom of this page indicates SAE's consent that copies of the paper may be made for personal or internal use of specific clients. This consent is given on the condition, however, that the copier pay a \$5.00 per article copy fee through the Copyright Clearance Center, Inc. Operations Center, 222 Rosewood Drive, Danvers, MA 01923 for copying beyond that permitted by Sections 107 or 108 of the U.S. Copyright Law. This consent does not extend to other kinds of copying such as copying for general distribution, for advertising or promotional purposes, for creating new collective works, or for resale.

SAE routinely stocks printed papers for a period of three years following date of publication. Direct your orders to SAE Customer Sales and Satisfaction Department.

Quantity reprint rates can be obtained from the Customer Sales and Satisfaction Department.

To request permission to reprint a technical paper or permission to use copyrighted SAE publications in other works, contact the SAE Publications Group.



**GLOBAL MOBILITY DATABASE**

*All SAE papers, standards, and selected books are abstracted and indexed in the Global Mobility Database.*

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

**ISSN 0148-7191**

**Copyright 1994 Society of Automotive Engineers, Inc.**

Positions and opinions advanced in this paper are those of the author(s) and not necessarily those of SAE. The author is solely responsible for the content of the paper. A process is available by which discussions will be printed with the paper if it is published in SAE transactions. For permission to publish this paper in full or in part, contact the SAE Publications Group.

Persons wishing to submit papers to be considered for presentation or publication through SAE should send the manuscript or a 300 word abstract of a proposed manuscript to: Secretary, Engineering Activity Board, SAE.



# Enumeration of Epicyclic-Type Automatic Transmission Gear Trains

Goutam Chatterjee and Lung-Wen Tsai  
University of Maryland

## Abstract

An automotive transmission maintains a proper equilibrium between the power and torque produced by an engine and those demanded by the drive wheels. Most automatic transmissions employ some kind of epicyclic gear mechanisms to achieve the above purpose. The first step in the design process of such a mechanism involves finding a configuration that provides a set of desired speed ratios, and meets other dynamic and kinematic requirements. In this work, the kinematic structural characteristics of epicyclic gear mechanisms have been identified, and a methodology is formulated to systematically enumerate all possible configurations of such mechanisms. This is achieved by defining a canonical graph to represent the mechanisms. Graphs of mechanisms with up to ten links have been generated using this methodology.

## 1 Introduction

The power from the engine crankshaft of an automobile is transferred to the drive wheel through a transmission unit, a final reduction unit, and a differential. The ratio of the speed of the input shaft (which brings power from the engine) to that of the output shaft of a transmission unit is called the *speed ratio* or the *reduction ratio*. Most of the automatic transmission units use epicyclic gear trains\* (EGTs) to achieve the desired reduction ratios. Some of them use one degree-of-freedom (dof) EGTs. Others use fractionated mechanisms, each fraction of which is a one-dof EGT. Tsai, et al. [14] have identified some of the structural characteristics required by an EGT to qualify for automatic transmissions, and have shown that of the many non-isomorphic graphs of the six-link one-dof EGTs, only six graphs can be used in automatic transmission gear boxes.

The EGT used in an automatic transmission is supported by bearings housed in the casing. This results in

\*A few use countershaft type of arrangement that is typical of manual transmission. A hybrid type using a combination of both is also possible.

a two-dof fractionated mechanisms. Henceforth, we will call the mechanism formed by an EGT and the casing of an automatic transmission gear box an *Epicyclic Gear Mechanism* (EGM).

## 2 Graph Representation of an EGT

The graph representation of an EGT has been described by Buchsbaum and Freudenstein in their pioneering paper [2]. The characteristics associated with the graph of an EGT have also been derived in the same paper and have been termed *fundamental characteristics*.

### 2.1 Isomorphism

Two graphs are isomorphic if there exists a one-to-one correspondence between their vertices and edges which preserves the incidence and labeling. The adjacency matrices of two isomorphic graphs can be different depending on the numbering of their vertices.

A reliable method for identifying isomorphism is to develop a unique code for each graph, such that two isomorphic graphs have the same code while two non-isomorphic graphs have different codes. This usually involves finding a way of uniquely numbering the vertices. The *degree code* formulated in [12] may be cited as an example employing this idea.

### 2.2 Pseudo-Isomorphism

Since an edge in a graph represents a joint between two links, only binary joints can be represented in a graph. Thus, a ternary joint is represented as two binary joints, a quaternary joints by three binary joints, and so on. This, however, creates a problem in uniquely representing a mechanism. For example, the mechanism shown in Fig. 1(a) can be reconfigured into the mechanism shown in Fig. 1(c) by rearranging the revolute joints among its coaxial links. Though these two mechanisms appear to

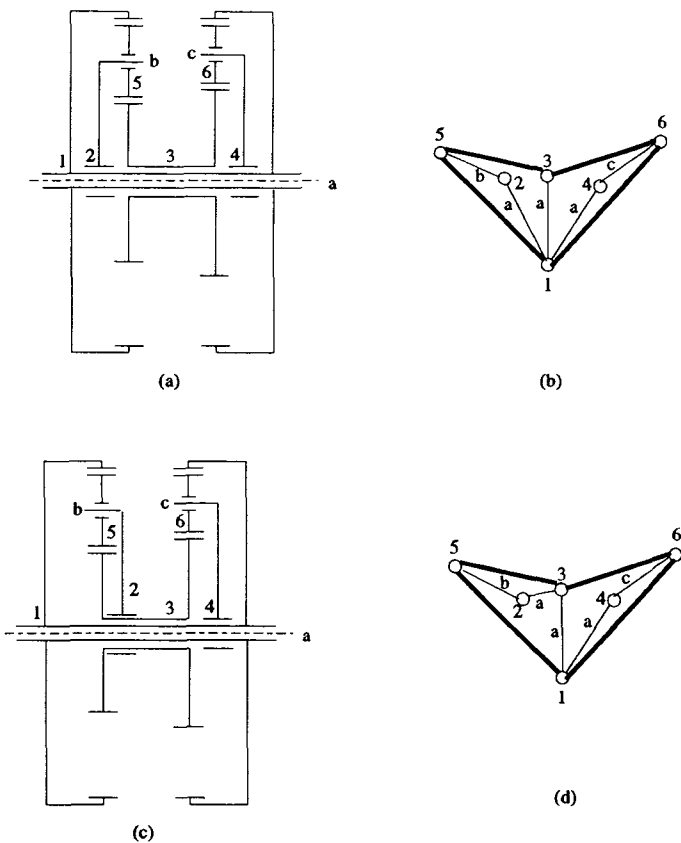


Figure 1 EGTs: (a, c) functional representations, (b, d) corresponding graph representations.

be structurally non-isomorphic [10], they are kinematically equivalent, and for the purpose of structural synthesis are considered the same. The graphs of the two mechanisms (Figs. 1(b) and (d) respectively) are mathematically non-isomorphic. The graph in Fig. 1(d) can be formed from the graph of Fig. 1(b), if the thin edge joining the vertices 1 and 2 is replaced by a thin edge of the same label joining vertices 2 and 3. This method of creating mathematically non-isomorphic graphs representing kinematically equivalent EGTs by replacing a thin edge by another thin edge of the same label is known as *vertex selection* and such mathematically non-isomorphic graphs are called *pseudoisomorphic* graphs [15]. The problem of pseudoisomorphism can be averted by imposing some rules that result in unique arrangement of the edges of the same label. Such a graph is called a *canonical graph* [13].

### 3 Canonical Graph Representation

An EGM typically consists of a one-dof EGT supported by the casing on one axis. Only the links that are connected to the casing by coaxial revolute joints can be used as input, output or fixed links [14].

The casing of an EGM is a unique link in its kinematic structure. Therefore, in the canonical graph representation of an EGM, the vertex representing the casing will be marked as the root of the graph. Recall that the coaxial

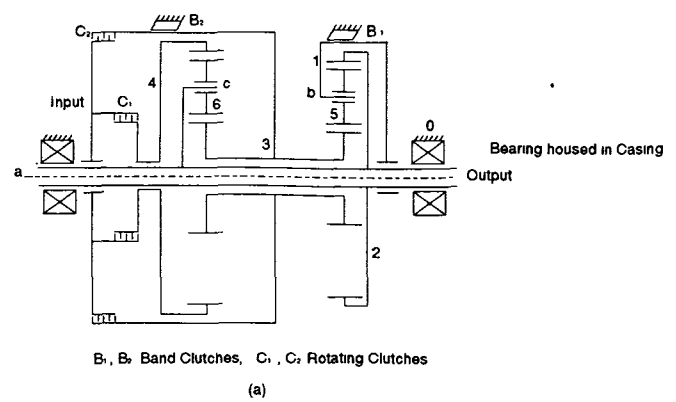


Figure 2 EGM employing Simpson gear set: (a) functional representation, (b) conventional graph representation, (c) canonical graph representation.

joints in a mechanism can be rearranged without affecting the mechanism functionally [13]. Among various arrangements of the coaxial joints there exists a unique configuration such that all the thin edged paths originating from the root and ending at all the other vertices will have distinct edge labels. This unique graph representation is called the canonical graph representation. Using canonical graph representation, the vertices can be divided into several levels. The casing is denoted as the *ground* level vertex. A vertex that is connected by only one thin edge to the root is defined as a *first* level vertex. A vertex that is connected to the root by two thin edges is defined as the *second* level vertex and so on. Each vertex at any particular level is connected to exactly one vertex at the immediate preceding level by a thin edge. All thin edges having the same label\* must be incident to one common lower level vertex. Henceforth, all the vertices at a particular level that are connected to a lower level vertex by thin edges of the same label, will be referred to as members of a *family*. The problem of pseudo-isomorphism does not arise once the canonical graph is defined. The canonical graph representation of the Simpson gear set shown in Fig. 2(a), is shown in Fig. 2(c). The second level vertices represent the planet gears.

### 4 Structural Characteristics

The canonical graph by virtue of its definition has one special feature, i.e.

\*Note the difference between *label* and *level*. A *label* denotes the location of the axis of a link in space while the word *level* denotes the location of a link in the kinematic chain relative to the casing.

C1: All the thin edges of the same label should be incident to a common lower level vertex.

All the fundamental characteristics of a graph of an EGT as described in [2] apply to the canonical graph of an EGM. These are described in section 4.1 under the heading *General Characteristics*. An EGM, besides possessing the characteristics of an EGT, also has its own specific characteristics because it is a mechanism that performs some special functions. These characteristics and their expressions in a canonical graph representation are discussed in Sections 4.2 through 4.4.

#### 4.1 General Characteristics

The canonical graph of an EGM has no articulation point. It possesses the following characteristics :

C2: If there are  $n$  vertices in the canonical graph of an EGM then it must have  $n - 1$  thin edges and  $n - 3$  geared edges, since an EGM is a two-dof mechanism.

C3: The subgraph formed by removing the geared edges is a tree.

C4: A geared edge can *only* be incident with one of the following pairs of vertices.

- (a) Two vertices at the same level that are connected to the same lower level vertex by thin edges of different labels.
- (b) Two vertices at adjacent levels, that are connected by a path of exactly three thin edges having two different labels.
- (c) Two vertices one at level  $k$  and another at level  $k - 2$ , if there is a path of exactly two thin edges between them. For every vertex at level  $k$  there is only one vertex at level  $k - 2$ , to which it can be connected by a geared edge.

#### 4.2 Coaxial Links

The first level vertices in the canonical graph of an EGM represent links that are connected to the casing by coaxial revolute joints. None of the links of an EGM should be connected to the casing by a gear joint\*. Therefore,

C5: The first level vertices are connected to the root by thin edges of the same label.

C6: No geared edge can be incident to the root.

In an EGM one of the coaxial links is permanently designated as the output. The desired reduction ratios are obtained by changing the input and the fixed links. Also, it is always possible to achieve a direct drive by locking all the links in the EGT together such that they rotate as a single link. Thus if  $N_e$  links of an EGT are coaxial, then it is possible to get  $(N_e - 1)(N_e - 2) + 1$  number of speed reductions. Therefore,

\*Note that it is possible to have one of the links of an EGT permanently fixed to the casing. This, however, will reduce the flexibility of obtaining more speed ratios from the gear train.

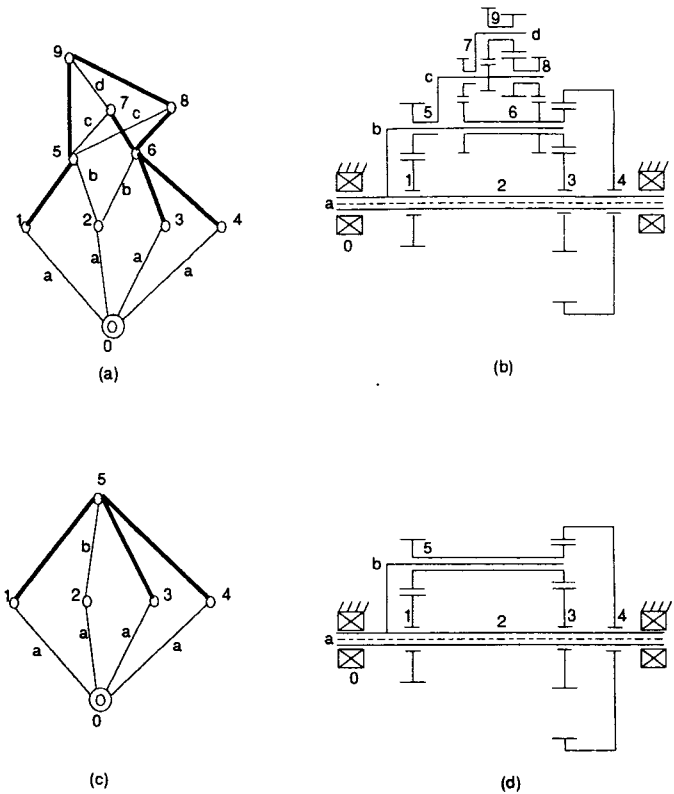


Figure 3: (a) A canonical graph containing a locked chain, (b) functional representation of the mechanism, (c) graph obtained by removal of locked chain, (d) functional representation of the resulting mechanism.

C7: If  $N_r$  is the number of required speed reductions (including reverse), then the number of first level vertices  $N_e$  in the canonical graph must satisfy the equation

$$N_r \leq (N_e - 1)(N_e - 2) + 1 \quad (1)$$

#### 4.3 Locked Chains

A set of links forming a part of a mechanism is said to be locked if they undergo no relative motion when the mechanism is in operation and, hence, can be replaced by one link without altering the functional characteristics of the mechanism.

Consider the canonical graph of an EGM shown in Fig. 3(a) that satisfies all the fundamental characteristics described in [2]. However, the subgraph formed by removing vertices 0, 1, 2, 3, and 4 from the graph represents a kinematically locked chain. This is because the subgraph has 5 vertices but 4 geared edges. Fig. 3(c) shows a graph formed by replacing vertices 5, 6, 7, 8, and 9 in Fig. 3(a) by a vertex 5, that will perform the same functions.

C8: An EGM contains a locked chain if there exists a subgraph of  $p$  vertices in the graph of the EGM such that

- (a) the transfer vertex of each geared edge in the subgraph lies in the subgraph, and

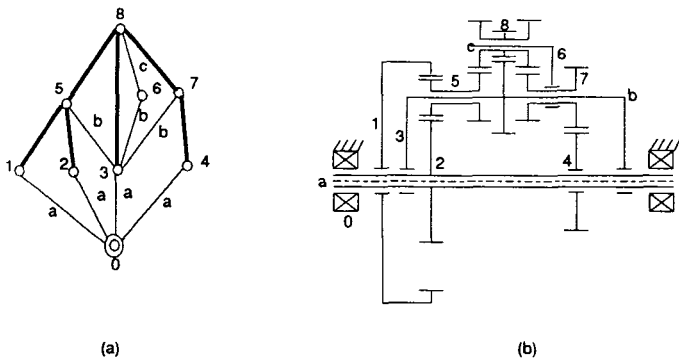


Figure 4 Canonical graph of an EGM with a binary carrier represented by vertex no. 6.

(b) the number of geared edges in the subgraph is more than  $p - 2$ .

A methodology is described in Section 7.2 that prevents the generation of EGM with locked chains during the enumeration process.

#### 4.4 Redundant Links

The other desired feature of a viable mechanism is that it should not have any *redundant* link. A link is said to be redundant, if it is never used as an input, output or a fixed link, and the removal of such a link does not change the degrees of freedom of the EGT. Such a link will not carry power during the operation of the mechanism at any of its reductions.

A one-dof EGT communicating, that is, giving and taking power, with the external environment requires at least an input, an output and a reaction (fixed) link. Thus, to function effectively it requires at least three ports of communication with the external environment. Similarly, a two-dof EGT requires at least four ports of communication, a three-dof EGT requires five ports of communication, and so on.

In the canonical graph representation of an EGM if there exists a subgraph that represents an  $n$ -dof EGT, then it must have at least  $n+2$  ports of communication with the external environment. Otherwise, those links that can not interact with the external environment would be redundant. The external environment includes both the rest of the gear train and the outside world. Two things are to be noted here.

1. The subgraph of a canonical graph represents an EGT if and only if the carrier of any gear pair within the subgraph is also a member of the subgraph. For example, the subgraph formed from the canonical graph of Fig. 4(a) by deleting vertices 0, 1, 2, 4, and 6 does not represent an EGT.
2. Some of the ports of communication at a first glance may not be obvious. For example, in the graph shown in Fig. 4(a), let vertex 1 represent the input link, vertex 2 the fixed link, and vertex 4 the output link. Consider the subgraph formed by deleting vertices 0,

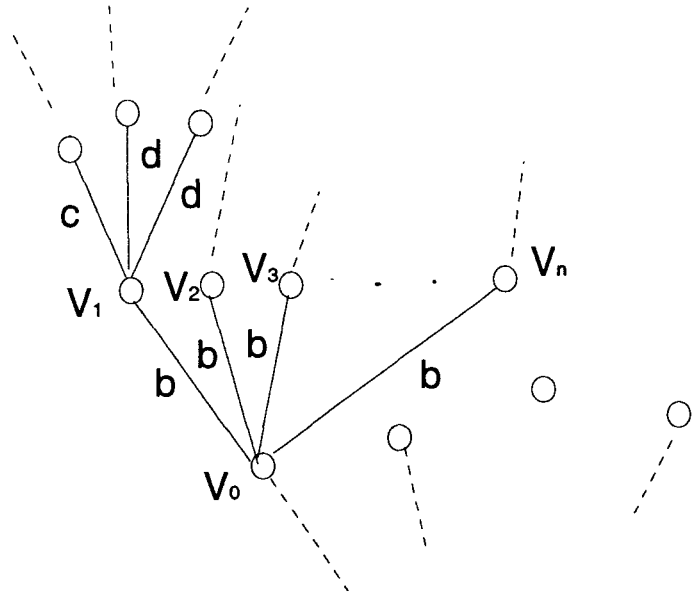


Figure 5 A tree obtained by removing the geared edges from a canonical graph

1, 2, 4, 5, and 7. It appears as if the subgraph has only two ports of communication, namely 3 and 8. However, the carrier represented by vertex 6 is also a port of communication. This is because vertex 6 is the transfer vertex of the geared pair connecting vertices 7 and 8, and vertex 7 is external to the subgraph. Therefore, the subgraph has actually three ports of communication.

From the above observations we will derive several conditions that the canonical graph of an EGM must satisfy. These conditions are necessary, but not sufficient. However, these conditions drastically reduce the number of graphs with redundant links during the enumeration process. The remaining few can be weeded out by inspection or by a methodology described later.

##### 4.4.1 Number of Branches

Consider the branches of a tree of a canonical graph emanating from vertex  $V_0$  as shown in Fig. 5. Let the level immediately above and arising from  $V_0$  contain  $n$  vertices  $V_1, V_2, \dots, V_n$  that belong to one family. Let the branches emanating from these vertices have  $N_1, N_2 \dots N_n$  vertices, respectively. Now vertex  $V_1$ , and the branches emanating from it form a subgraph that represents an EGT, because a geared edge joining any two vertex of the subgraph will have its transfer vertex contained in the subgraph. Let all such subgraphs be named  $G_1, G_2 \dots G_n$ , and let them have  $F_1, F_2 \dots F_n$  dof, respectively. Then  $G_i$  will have  $N_i - F_i$  geared edges and will require at least  $F_i + 2$  ports of communication. One port of communication is the vertex  $V_i$  itself. The rest of the ports of communication will communicate through gear edges with any of the vertices  $V_0$  to  $V_n$  other than  $V_i$ . A geared edge coming from any port of  $G_i$  cannot connect to any vertex other than  $V_0$  to  $V_n$  of the EGM.

Next consider the subgraph  $G_0$  formed by vertices  $V_0$  to  $V_n$  and the branches emanating from  $V_1$  to  $V_n$ . The minimum number of geared edges that  $G_0$  should have in order that none of the links represented by vertices in the subgraphs  $G_1, G_2 \dots G_n$ , is rendered redundant can be calculated by summing the number of geared edges in each subgraph, and the number of geared edges required by each subgraph to maintain the minimum number of ports of communication. Thus the number of geared edges in  $G_0$  should be at least

$$\sum_{i=1}^n (N_i + 1 - F_i - 1) + \sum_{i=1}^n (F_i + 1) = \sum_{i=1}^n N_i + n \quad (2)$$

The number of geared edges in a subgraph representing an EGT must be less than the number of vertices by 2 or more, otherwise it will be locked. Since  $G_0$  has  $\sum_{i=1}^n N_i + n + 1$  vertices, this condition can not be satisfied for the above case. Therefore, not all the members of a family can give rise to branches. If one of the members does not give rise to any branch, then the number of vertices in  $G_0$  will be  $\sum_{i=1}^{n-1} N_i + n + 1$  and the minimum number of geared edges required to prevent redundancy will be at least  $\sum_{i=1}^{n-1} N_i + n - 1$ . Thus, if one of the members does not give rise to any branch, the above condition is satisfied. Also, if more than one member of a family does not give rise to any branch the above condition can be satisfied.

Let  $V_1, V_2 \dots V_n$  represent the first level vertices, and vertex  $V_0$  represents the root of the EGM. Since an EGM has exactly two-dof, the number of geared edges must be less than the number of vertices by 3. Applying this condition and following the above logic one can prove that there must be at least two vertices in the first level which should not give rise to any branch.

Thus, we get the following two conditions.

C9: For those vertices located at the higher levels, there must be at least one member in a family that does not give rise to any branch.

C10: There must be at least two vertices in the first level that do not give rise to any branch.

As a special case of Condition 1, a vertex cannot give rise to any branch if it is the only vertex in a family.

#### 4.4.2 Number of Incident Edges

The subgraph formed by vertices 3, 6, and 4 in Fig. 6 represents a one-dof EGT. Hence it should have at least three ports of communication. But it has only two ports of communication, namely vertices 3 and 4, and the removal of vertex 6 does not change the dof of the mechanism. Therefore, vertex 6 represents a link that is redundant and can be removed. This is generally true for any link that is not used as the input, output or fixed link and is connected by only one revolute joint and one gear joint. Thus,

C11: If a vertex is not located at the first level and is incident by only one thin edge, then it must be incident by at least two geared edges.

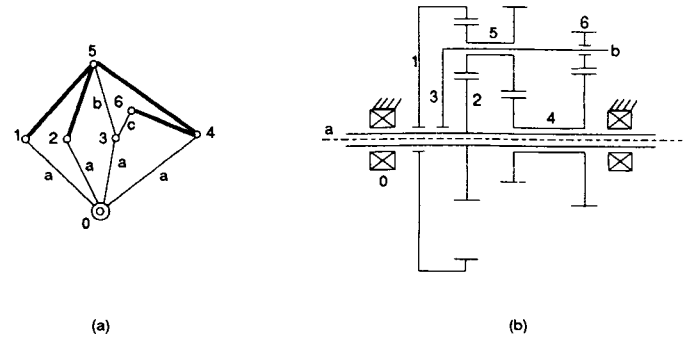


Figure 6: (a) A canonical graph containing a vertex (vertex no. 6) representing a redundant link, (b) functional representation of the mechanism.

C12: If a vertex is located at the first level and is incident by only one thin edge, it must be incident by at least one geared edge.

#### 4.4.3 Floating Carrier

Consider the graph shown in Fig. 7(a). The subgraph formed by vertices 3, 5, 6, 7 and 8 represents a two-dof EGT. However, it has only three ports of communication, namely 3, 5, and 7. Therefore, the links represented by vertices 6 and 8 are redundant. Note that the vertex 6 represents a binary vertex that is not connected to any other vertex by a geared edge. According to Condition C9, a binary vertex of this kind can only occur at the penultimate level of a branch. If the vertex at the higher level that is connected to the binary vertex is incident by two geared edges, then there are two possible ways of connecting them. These two ways are shown in Figs. 7(a) and (c). In both these cases the binary vertex, and the higher level vertex that is incident to it are redundant. In Fig. 7(c) links represented by vertices 5 and 7 are redundant because the subgraph formed by vertices 1, 5, 6 and 7 represents a one-dof EGT that has only two ports of communication. Thus,

C13: If a binary vertex in a tree is not at the first level then of the two vertices that it connects, the higher level vertex must be incident by more than two geared edges.

Fig. 4 shows the graph of an EGM that has a binary vertex, but no redundant links. In this case the higher level vertex (vertex No. 8) that is connected to the binary vertex 6 is incident by three geared edges.

## 5 Graph Enumeration

Most combinatorial enumeration procedures that require the enumeration of all possible solutions satisfying certain constraints, are done through the process of *generation* and *testing*. The procedure is thus divided into two parts [5]: a generator of all possible solutions and a tester that selects only those solutions that meet the constraints.

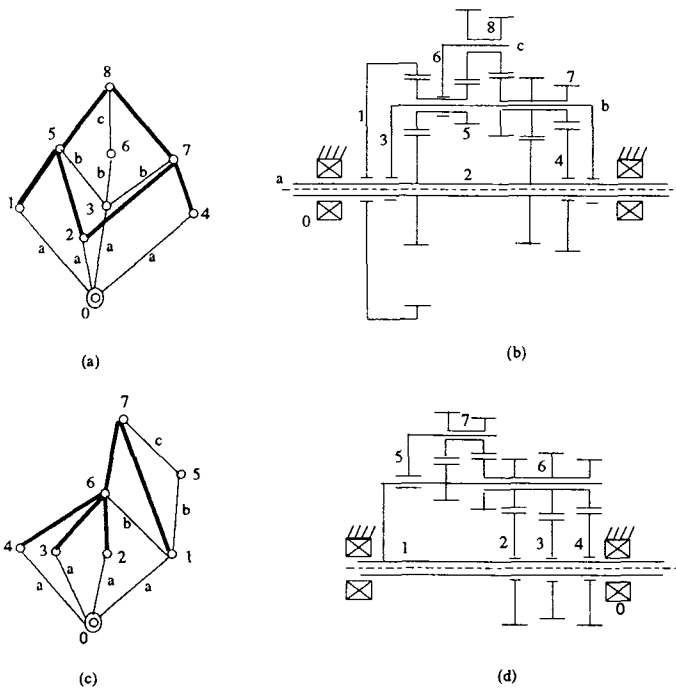


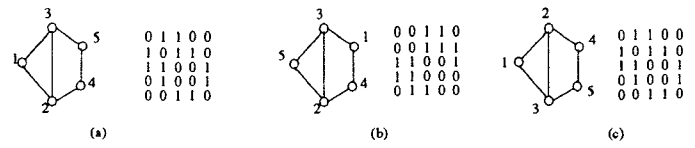
Figure 7: (a, c) Canonical graphs containing vertices (No. 6 & 8 in (a) and No. 5 & 7 in (c)) representing redundant links, (b, d) functional representation of the mechanisms.

The enumeration procedure described below uses a hierarchical generation and testing technique. A part of the canonical graph is generated at each step, and a test is carried out to prune out those solutions that will not give rise to canonical graphs with the desired characteristics.

The following observations are made before formulating an efficient enumeration procedure.

1. The canonical graph has a unique vertex - the root, with reference to which other vertices are divided into several levels. Thus, there is already some arrangement among the vertices. One can therefore think of obtaining a unique arrangement of vertices by adding some rules. This can then be used to develop a unique code for each graph that will serve as a reliable tool for an isomorphism test.
2. The definition of the canonical graph requires the edge labels to be distributed in a particular way. This forbids the generation of pseudo-isomorphic graph.
3. Most of the characteristics described in Section 4 are applicable to the tree of a canonical graph, rather than the canonical graph as a whole.
4. Characteristic C4, which prescribes the allowable geared edge connections, presumes the existence of a labeled tree.

Therefore, it follows that an efficient enumeration procedure can be achieved if it is divided into two phases. In the first phase labeled trees that will give rise to admissible canonical graphs are enumerated. In the second phase geared edges are added to these trees to create the canon-



Note. The matrix beside each graph is the link to link adjacency matrix

Figure 8 Graphs in (a) and (b) are isomorphic but not automorphic, Graphs in (a) and (c) are automorphic.

ical graphs. Each of these phases comprises of various steps, that are described in Sections 6 and 7, respectively.

## 6 Enumeration of Trees

The characteristics C1, C2, C5 to C10 are used to formulate the procedure for enumerating trees with  $n$  vertices.

**Step 1.** From C7 calculate the minimum number of coaxial links required. This gives the minimum number of vertices that a tree should have at the first level.

**Step 2.** Distribute the remaining vertices into various levels. While making such a distribution, remember that all levels other than the highest level must have at least 2 vertices in order to satisfy C9.

**Step 3.** Divide the vertices at each level into families ( see Section 3). At all levels other than the highest level there must be at least one family with two or more members in order to satisfy C9. One must note that the distribution of vertices into families is same as the problem of partitioning of integers. An algorithm to this end can be easily developed by using the concept of generating functions, explained in the book by Liu [9].

**Step 4.** Start adding the vertices of the first level. According to C5 all the vertices at the first level should be connected to the root with thin edges of the same label.

**Step 5.** Next connect the vertices of the second level to the first level vertices. According to C10, two of the vertices at the first level should not give rise to any branch. The definition of a canonical graph requires the members of the same family to be connected to the same lower level vertex by edges of the same label. Therefore, while adding the second-level vertices, add one family at a time. Start with the family that has the largest number of members. If a family of vertices is added in all possible ways a lot of isomorphic graphs would be generated. To reduce the number of isomorphic graphs we digress and introduce the concept of graph automorphism and similar vertices.

### 6.1 Graph Automorphism

Consider the graph shown in Fig. 8. The edges of the graph are unlabeled and its vertices are numbered. If we permute the numbering of the vertices, isomorphic graphs

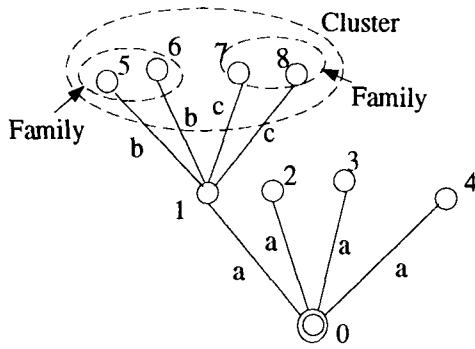


Figure 9 A tree showing similar vertices.

are produced. Most of these isomorphic graphs have their corresponding vertices numbered differently. However, some specific permutations produce graphs whose corresponding vertices bear the same number as the original one. These graphs are called *automorphic graphs*. For example, if we number the vertices 1, 2, 3, 4, and 5 of the graph in Fig. 8(a) as 1, 3, 2, 5, and 4, the resulting graph as shown in Fig. 8(c) is automorphic. The permutation in this case is denoted by  $(1)(2,3)(4,5)$ . Elements 2 and 3 are said to form a cycle of length 2, and element 1 a cycle of length 1. All such permutations that produce automorphic graphs form a group [9]. Each of these permutations is referred to as a *member* of the group. The application of any of the members from the group won't alter the adjacency matrix of the graph in any way. If two vertices  $p$  and  $q$  are contained in the same cycle of any member of such a permutation group, then vertices  $p$  and  $q$  are said to be similar [16]. Thus, one can divide the vertices of a graph into classes by putting the similar vertices together. If one wants to add a particular property to any vertex, then there can be one choice from a class of similar vertices since any choice is as good as the other.

Some very simple rules are prescribed here to identify some of the similar vertices. These rules won't prevent the generation of isomorphic graphs completely, but will reduce their number drastically. These rules are

**S1:** Vertices belonging to the same family and incident by only one thin edge and no other edges are similar. For example, vertices 2, 3 and 4 in Fig. 9 are similar.

**S2:** Families that have the same number of members and are connected to the same lower level vertex form a *cluster*. If none of the vertices in a cluster is incident by more than one thin edge, then all the vertices in the cluster are similar. For example, vertices 5, 6, 7 and 8 in Fig. 9 are similar. If a vertex is incident by more than one edge, then the family of vertices to which that vertex belongs becomes dissimilar, while the rest of the families in the cluster remain similar.

Whenever a choice for a vertex is to be made for adding a vertex, or a family of vertices, choose only one vertex out of a class of similar vertices.

The addition of vertices at the higher levels should proceed in the same way as the second level vertices. The

Table 1 Distribution of vertices of the graphs representing 9-link EGMs into levels and families: (a) level distribution, (b) family distribution.

Level I	7	6	5	5	4	4	4
Level II	1	2	3	2	4	3	2
Level III				1		1	2

(a)

Levels	Distribution of vertices into levels	Distribution of vertices into families
I	4	4
II	2	2
III	2	2
		1,1

(b)

only difference is that C9 is applicable instead of C10, i.e., at least one member in each family of vertices should not give rise to any branches. A test to eliminate isomorphic graphs is carried out after completing the addition of vertices at each level.

The example given below demonstrates the above enumeration method.

In this example, trees that can give rise to admissible canonical graphs which represent 9-link EGMs, capable of providing four speed reductions, are enumerated.

**Step 1.** There are nine links. The minimum number of coaxial links required is four. Therefore, the first level should have at least four vertices.

**Step 2.** The remaining vertices can be divided into levels as shown in Table 1(a).

**Step 3.** The vertices at each level are further divided into families. One of the distribution is chosen from Table 1(a) to demonstrate this step. The distribution of vertices at each level into families for the chosen distribution is shown in Table 1(b).

**Step 4.** This step and the next one are demonstrated by choosing one of the two distribution of vertices into families from Table 1(b). The distribution chosen is shown in Fig. 10(a). Fig. 10(b) and (c) shows the tree and the corresponding adjacency matrix formed after addition of the first level vertices. The adjacency matrix has all its elements zero except for those in the rows or columns corresponding to the root.

**Step 5.** In the second level there is only one family with two members. There is only one choice of vertex in the first level to which this family can be connected since all the vertices at the first level are similar according to S1. The resulting tree is shown in Fig. 10(d). In terms of the adjacency matrix this would mean the addition of two rows and columns to the matrix of Fig. 10(c). The resulting matrix is

Level	0	I	II	III
Family	1	4	2	1

(a)

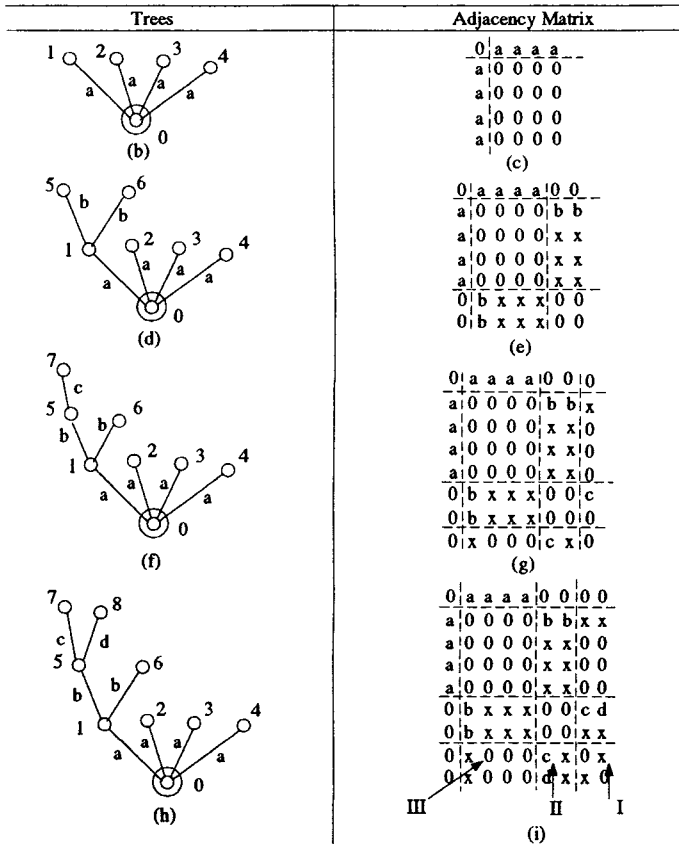
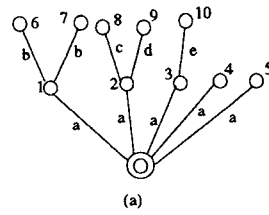


Figure 10 Enumeration of trees for a given family and level distribution: (a) level and family distribution, (b) and (c) addition of first level vertices, (d) and (e) addition of second level vertices, (f) - (i) addition of third level vertices.

shown in Fig. 10(e). All the elements of the new rows and columns are zero except for element number\* 1. However, some of the zero labeled elements will be relabeled  $g$  when geared edges are added. Potential geared edge connections can be found from C4. The corresponding elements, instead of being set to zero, are therefore set to  $x$  to facilitate the process of geared edges addition in the second phase. The elements  $A_{56}$  and  $A_{65}$  are set to zero because there can be no geared edge between vertices of the same family according to C4(a). All other elements are set to  $x$  since they satisfy C4(b). This completes the addition of vertices at the second level.

The addition of the vertices at the third level is illustrated in Fig. 10(f) to 10(i). The dashed lines across the adjacency matrices divide them into various sub-matrices, each containing information about a particular type of interactions. For example, submatrix I in Fig. 10(i) represents the interaction within



Level	Family	Cluster	Priority
0	(0)	(0)	(0)
I	(1,2,3,4,5)	(1,2,3,4,5)	(1)(2)(3)(4,5)
II	(6,7)(8)(9)(10)	(6,7)(8,9)(10)	(6,7)(8,9)(10)

Figure 11 A tree having its vertices numbered according to their priority.

the third level vertices, submatrix II the interaction between the third and second level vertices, and submatrix III the interaction between the third and first level vertices. Whether an element of the submatrices I, II, and III can be converted into  $x$  or not can be determined by applying C4(a), C4(b), and C4(c), respectively.

Before ending this section on enumeration of trees we note that the generator generates only those trees that have the desired characteristics. The tester has only to identify the isomorphic graphs whose number has been reduced by incorporating some rules in the generator.

## 6.2 Isomorphism

The issue of developing unique code for identifying isomorphic graphs has been addressed by many authors ([1], [12]). Most of these papers dealt with graphs whose edges are not labeled. The labeling of edges of a graph has both its advantages and disadvantages. On one hand, it divides the vertices into classes that we have already named as families and, therefore, introduces some amount of ordering among the vertices. On the other hand, since the labels are arbitrary they have to be permuted in all possible ways in order to detect isomorphism. Some papers have presented graph representations [11] that obviate the need to explicitly represent the labels of the revolute edges in the adjacency matrices. This paper achieves the above objective by proposing four simple rules.

- I1 Vertices at the lower levels should have higher priority than those at higher levels. For example, vertices 1, 2, 3, 4, and 5 in Fig. 11 have higher priority than vertices 6, 7, 8, 9, and 10. Hereafter, whenever we say that a vertex has a higher priority than another it means that the former is numbered lower than the latter.
- I2 Members of a family such as vertices 1, 2, 3, 4, and 5 in Fig. 11, should be consecutively numbered.
- I3 All members of the families that belong to the same level and have the same number of members should be consecutively numbered. For example, vertices 8, 9, and 10 in Fig. 11 are consecutively numbered.
- I4 Families that have more members, have higher priority than those having less. Vertices 6 and 7 in Fig. 11 have higher priority than vertices 8, 9 or 10.

\*We are following the convention of C programming language in indexing the elements of rows and columns. The indexing starts with the number 0.



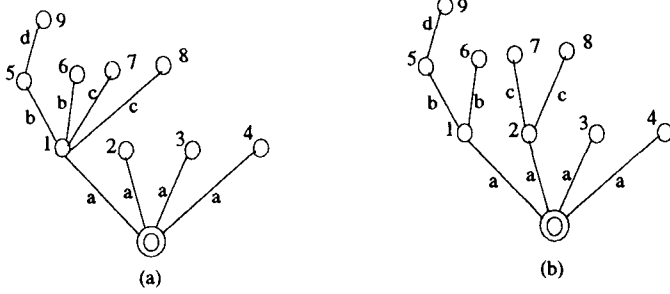


Figure 12 Trees having their vertices numbered in order of their priority.

If we number the vertices following the above rules, then there is no need to explicitly label the edges. One can uniquely determine the labels of the edges from the level and family distributions.

Next we propose a set of rules that decide the priorities of vertices within a family and the priority of a family of vertices over another having the same number of members. These rules should be applied successively in the order stated below and should not alter the priorities already decided by the application of previous rules. These rules are:

- I5 A vertex in a family precedes another vertex if it gives rise to more families with a higher number of members. For example, in Fig. 11 vertex 1 precedes vertex 2, which in turn precedes vertex 3.
- I6 To determine the order of families having the same number of members, the vertices of the families are compared. If the highest priority vertex of family *A* gives rise to more families with a higher number of members than that of family *B*, then *A* precedes *B*. If the precedence cannot be determined by comparing the highest priority vertices of the two families then the next priority vertices are compared. For example, vertices 5 and 6 precede vertices 7 and 8 in Fig. 12(a).

To develop the code all labels of the thin edges are replaced by 1 in the adjacency matrix. Then, the vertices are permuted to maximize the number formed by concatenating the elements of the upper triangular matrix (of the adjacency matrix), starting from the leftmost element of the topmost row and moving along row by row downwards. This maximized number, along with the level and family distribution gives a unique code for isomorphic trees. Since the permutation of similar vertices does not change the adjacency matrix, only dissimilar vertices of a family that have the same priority are permuted. Families that have the same priority are permuted en bloc.

The isomorphism test is applied after completing the addition of vertices at each level. Rules I1 to I4 are applied to arrange the vertices in the level that has just been added, whereas rules I5 and I6 are applied to arrange the vertices in the immediately preceding level.

## 7 Enumeration of EGMs

The adjacency matrices of the trees that have been enumerated until now have some of their elements labeled *x*. The addition of geared edges means relabeling some of these *x*'s as *g*'s, and the rest as zeroes. We describe below an algorithm to find the transfer vertex associated with each gear pair, and study the interaction among fundamental circuits formed by the addition of geared edges.

### 7.1 Locating the Transfer Vertex

Consider the tree and its adjacency matrix shown in Fig. 13. Suppose, a geared edge is added between vertices 7 and 6. To find the associated transfer vertex scan the row corresponding to the higher level vertex (vertex 7 in this case). If both the vertices are at the same level, then scan the row corresponding to any of the vertices. The column number corresponding to the first non-zero element that is not an 'x' or a 'g' gives the number of the transfer vertex. In case of the above example element no. 5 of row 7 gives the number of the transfer vertex. A fundamental circuit is characterized by the two end vertices of a geared edge and the associated transfer vertex. Therefore, once the transfer vertex is known, the fundamental circuit is in effect known.

### 7.2 Interaction Among Fundamental Circuits

The three vertices that characterize a fundamental circuit form a simple one-dof EGT with three links. In order to keep track of the interactions among fundamental circuits we construct a matrix whose column number corresponds to the vertex number. In the first row of this matrix we mark the elements that correspond to the characteristic vertices of the first fundamental circuit by a label, say 1. For example, if we connect the vertices 7 and 8 of the tree shown in Fig. 13(b) by a geared edge, then the matrix will take the form shown in Fig 14(a).

If the next fundamental circuit formed shares two of its vertices with the existing one then the two fundamental circuits will constitute a subgraph that represents a one-dof EGT. In the first row of the above matrix we label the element corresponding to the non-common vertex. For example, in Fig. 13(g), due to the fundamental circuit formed by the addition of the geared edge between vertices 6 and 8 the first row of the above matrix is modified as shown in Fig. 14(b). However, if the two fundamental circuits have one vertex in common we add a new row to the above matrix and label the elements corresponding to the characteristic vertices of the newly formed fundamental circuit. Thus, each row of the matrix corresponds to a subgraph that represents a one-dof EGT.

In general if we consider a graph in which *k* geared edges have been added, then the fundamental circuit formed due to the addition of the *k* + 1 geared edge can have the following relationships with any of the existing subgraphs.

1. It can have one of its vertices in common with a subgraph representing a one-dof EGT. In this case add

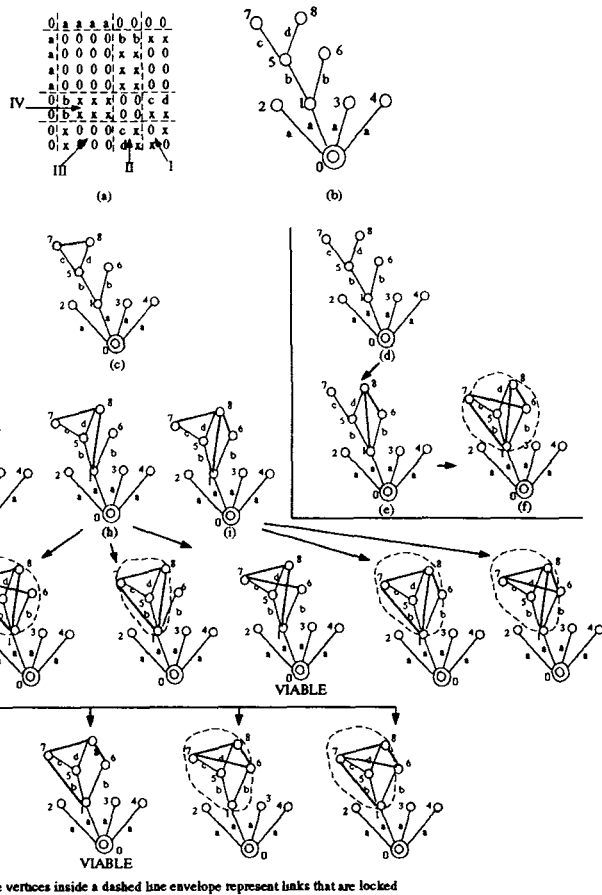


Figure 13 Step by step enumeration of EGMs from a tree - Part I.

a new row to the matrix, and label the elements corresponding to the characteristic vertices of the newly formed fundamental circuit.

- It can have two of its vertices in common with a subgraph representing a one-dof EGT. In this case, modify the row corresponding to the existing subgraph by labeling the element that corresponds to the non-common vertex of the fundamental circuit that has just been added. If the modified subgraph now shares two of its vertices with another existing subgraph combine the two corresponding rows into one, since the two subgraph together represent a one-dof EGT.

Repeat this process until no two subgraph have more than one vertex in common.

A new fundamental circuit formed cannot have three of its vertices in common with a subgraph representing a one-dof EGT, otherwise the mechanism will be locked.

The above observations will be used to keep a track of the subgraphs that are being formed, and to prevent the occurrence of locked chains.

### 7.3 Similar Edges

Consider the tree shown in Fig. 13. The similar vertices in the tree can be identified by applying S1 and S2 as defined in Section 6.1. Because of the similarity among

Vertices	1	2	3	4	5	6	7	8
					1		1	1

(a)

Vertices	1	2	3	4	5	6	7	8
					1	1	1	1

(b)

Figure 14 Matrices to keep track of the interaction among fundamental circuits.

vertices some of the candidate geared edges (represented by label  $x$  in the adjacency matrix) are similar. Some of these sets of similar geared edges can be identified by the application of the rule given below.

- S3** When several geared edges connect a common vertex to a set of similar vertices, they form a similar edge set.

### 7.4 Addition of Geared Edges

The method of adding geared edges can be formulated as follows.

**Step I.** First add geared edges connecting vertices at the highest level. For every geared edge that is being added check whether the addition of the geared edge results in a locked chain by the method described above. If it does, then set the label  $x$  that corresponds to the geared edge in the adjacency matrix to zero.

**Step II.** Next add geared edges from the highest level to the lower levels. Before doing this calculate the minimum number of geared edges to be added to each of the vertices at the highest level from C11. If the highest level is the second level, then the minimum number of geared edges to be incident on a first level vertex as given in C12 should also be taken into account. Care should be taken that the total number of geared edges to be added does not exceed that given by C2. As before, check for locked chains for every geared edge added.

**Step III.** Repeat steps I and II for the next lower level vertices, i.e, the vertices that are at one level immediately below the highest level.

Repeat step III until the second level is reached.

The methodology for addition of geared edges is illustrated in Fig. 13. Geared edges are added to the tree (Fig. 13(b)) starting from the highest level and continuing downward.

At the end of enumeration procedure three EGMs are formed. They are shown in Fig. 15(d), (e), and (f). The one shown in Fig. 15(d) has redundant links. This is because the subgraph formed by vertices 1, 5, 6, 7 and 8

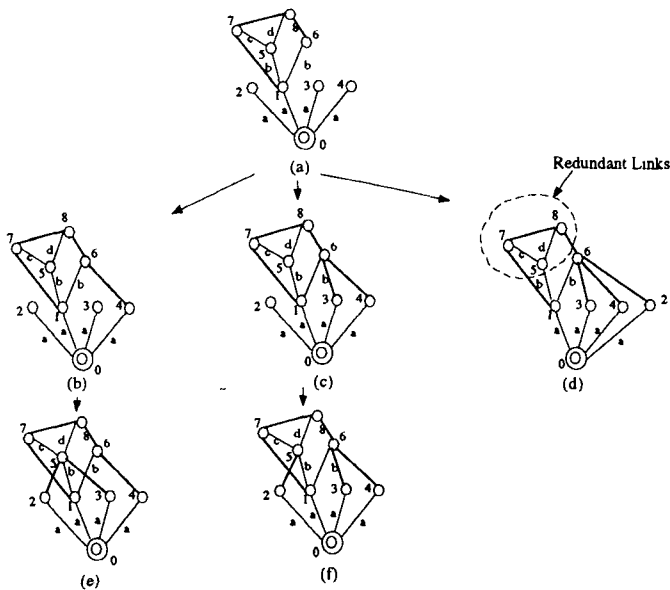


Figure 15 Step by step enumeration of EGMs from a tree - Part II.

represents a one-dof EGT that has only two ports of communication, i.e., vertices 1 and 6. The generation of such graphs can be prevented if we ensure that every subgraph (formed at the end of step II) that represents an  $n$ -dof EGT has  $n + 2$  ports of communication. The subgraphs and their dof can be obtained from the matrix that has been developed to prevent the occurrence of locked chains. However, such a verification is not required when geared edges are added to connect the highest level vertices to the lower level vertices, since conformation to C11 and C12 ensures that there will be no redundant links.

#### 7.4.1 Isomorphism

A test to identify isomorphic graphs is performed at the end of each step. To do this we extend the procedure described in Section. 6.2. The rules to identify the priority of vertices are given below. They, however, should not alter the priority set by rules I1 to I6. Also, the rules should be applied in the order given below and should not alter the arrangement set by the previous rules.

- I7** The vertex that is connected to vertices at two levels above it with more geared edges has the highest priority.
- I8** Among vertices of same priority in a family, the vertex that is connected to vertices at one level above it with more geared edges is given higher priority.
- I9** Among vertices of same priority in a family, the vertex that is connected to vertices at the same level with more geared edges is given higher priority.
- I10** Among vertices of same priority in a family, the vertex that is connected to vertices at lower levels with more geared edges is given higher priority.

To develop a code for the graph, the  $g$ 's in the adjacency matrix are replaced by 2's. Then, the vertices are per-

Table 2 Number of graphs enumerated for EGMs with up to 10 links.

No. of links	No. of graphs	No. of graphs with vertices only upto the second level.
6	1	1
7	7	7
8	22	20
9	157	128
10	899	620

mutated to maximize the number formed by concatenating the elements of the upper triangular matrix as described in Section 6.2.

## 8 Results and Discussions

The results are tabulated in Table 2. It has been mentioned that the first level vertices of a canonical graph represent the potential input, output and fixed links while the second level vertices represent the planet gears. A review of the work of Larew [7], Levai [8], Gott [4], and Tsai, et al. [14] had not revealed a single automatic transmission gear box having a link located on the third or higher levels. Since no physical reason could be found for this observation, it was not considered a structural characteristic of such gearboxes. However, Table 2 also lists the number of graphs having vertices only upto the second level. The graphs of 8-link EGMs, are shown in Fig 16. The graphs and adjacency matrices of all the EGMs with upto 9-links are documented in [3].

There is only one graph for 6-link EGMs\*, which is in agreement with the result given in [14]. There are 7 graphs for 7-link EGMs, which is one more than that given in the same paper. This is because that paper has excluded those graphs in which the geared edges form a closed loop. The verification for completeness of the set of graphs enumerated for 8-link EGMs has been accomplished in an indirect way. From the set of graphs of 7-link EGTs generated by Kim and Kwak [6], those that qualify for automatic transmissions were selected. A total of 20 such graphs were extracted from their paper which is less than the 22 given in Table 2. The reason is that there are exactly 2 graphs of 8-link EGMs (Figs. 16(f) and 16(q)) that have geared edges forming a loop, and those graphs cannot be generated by the method of Kim and Kwak.

## 9 Conclusions

The structural characteristics of epicyclic gear mechanisms (EGMs) that are commonly used in automatic transmissions to obtain various speed ratios have been identified from the view point of kinematics. A canonical graph representation for this type of mechanisms has been defined. A methodology to systematically enumerate

\*An  $n$ -link EGM contains a  $(n - 1)$  link EGT and the casing of a transmission.

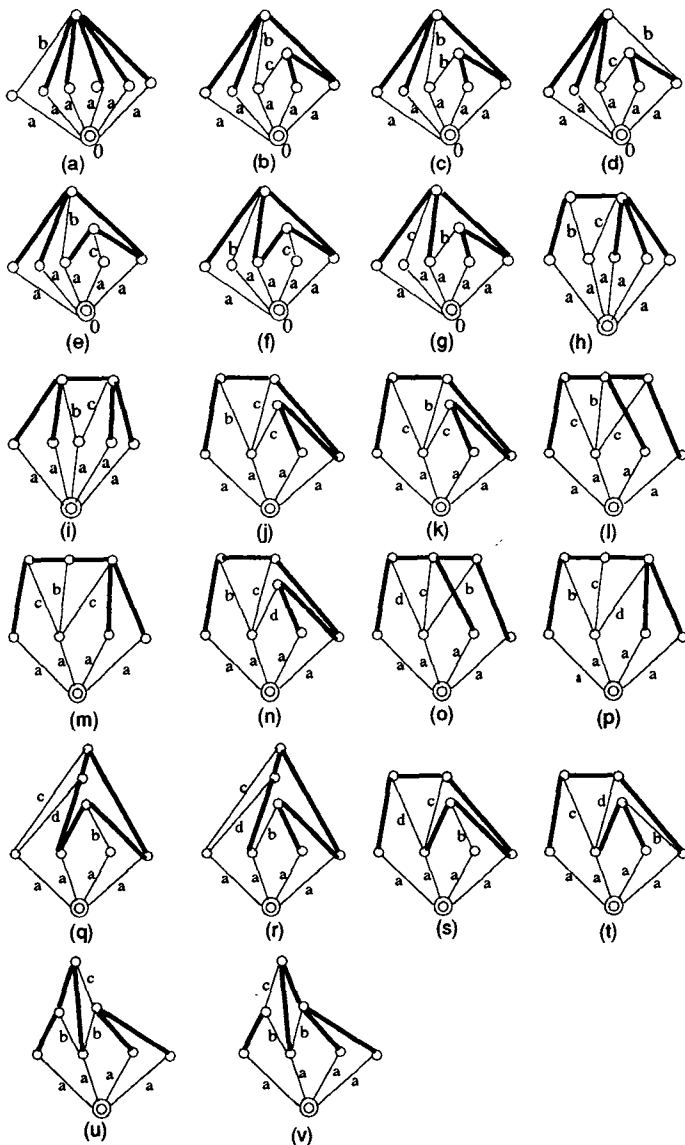


Figure 16 Graphs of 8-link EGMs.

these graphs has been developed and illustrated through various examples. Graphs of EGMs with up to 10-link have been enumerated using this method.

A methodology for automatically sketching the functional schematics of an EGM from its graph representation has been developed. It will be presented in the ASME 1994 Mechanisms Conference. It is hoped that this work will provide a basis for the design of future automatic transmissions.

## References

[1] Ambedkar, A.G., and Agrawal, V.P., 1987, "Canonical Numbering of Kinematic Chains and Isomorphism Problem: MIN Code," *Journal of Mechanism and Machine Theory*, Vol. 22, No. 5, pp. 453-461.

[2] Buchsbaum, F., and Freudenstein, F., 1970, "Synthesis of Kinematic Structure of Geared Kinematic

Chains and Other Mechanisms," *Journal of Mechanisms and Machine Theory*, Vol. 5, pp. 357-392.

[3] Chatterjee, G., 1993, *Enumeration and Automatic Sketching of Epicyclic Type Automatic Transmission Gear Trains*, Master's Thesis, University of Maryland, College Park, Maryland.

[4] Gott, P. G., 1991, *Changing Gears*, Society of Automotive Engineers, Inc., Warrendale, Pennsylvania.

[5] Hayes-Roth, F., Waterman, D.A., and Lenat, D.B., 1983, *Building Expert Systems*, Addison-Wesley, Reading, Massachusetts.

[6] Kim, J.T., and Kwak, B.M., 1990, "Application of Edge Permutation Group to Structural Synthesis of Epicyclic Gear Trains," *Journal of Mechanisms and Machine Theory*, Vol. 25, No. 5, pp. 563-574.

[7] Larew, W. B., 1966, *Automatic Transmissions*, Chilton Book Company, Radnor, Pennsylvania.

[8] Levai, Z., 1964, *Theory of Epicyclic Gears and Epicyclic Change-Speed Gears*, Doctoral Dissertation, Technical University of Building, Civil and Transportation Engineering, Budapest.

[9] Liu, C.T., *Introduction to Combinatorial Mathematics*, McGraw-Hill, New York.

[10] Mruthyunjaya, T.S., and Ravisankar, R., 1985, "Computerized Synthesis of the Structure of Geared Kinematic Chains," *Journal of Mechanisms and Machine Theory*, Vol. 20, pp. 367-387.

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

[12] Tang, C.S. and Liu, T., 1993, "Degree Code - A New Mechanisms Identifier," *ASME Journal of Mechanical Design*, Vol. 115, pp. 627-630.

[13] Tsai, L.W., 1988, "The Kinematics of Spatial Robotic Bevel-Gear Trains," *IEEE Journal of Robotics and Automation*, Vol. 4, pp. 150-156.

[14] Tsai, L.W., Maki, E.R., Liu, T., and Kapil, N.G., 1988, "The Categorization of Planetary Gear Trains for Automatic Transmissions According to Kinematic Topology," SAE Technical Paper # 885602, *Proceeding of the XXII FISTA Congress and Exposition*, Vol. 1., pp. 1513-1512.

[15] Tsai, L.W., and Lin, C.C. 1989, "The Creation of Non-Fractionated Two-Degree-of-Freedom Epicyclic Gear Trains," *ASME J. of Mechanisms, Transmissions, and Automation in Design*, Vol. 111, pp. 524-529.

[16] Yan, H.S., and Hwang, Y.W., 1991, "The Specialization of Mechanisms," *Mechanisms and Machine Theory*, Vol. 26, pp. 541-551.



