## References

1 Rosenauer, N., and Willis, A. H., *Kinematics of Mechanisms*, Associated General Publications, 1953, pp. 252–253.

2 Hinkle, R. T., *Kinematics of Machines*, 2nd ed., Prentice-Hall, 1960, p. 325.

3 Crossley, F. R. E., "A Contribution to Gruebler's Theory in the Number Synthesis of Plane Mechanisms," ASME *Journal of Engineering for Industry*, Vol. 86, 1964, pp. 1–8.

4 Sanger, D. J., "The Structure of Kinematic Chains Which Contain Links of Maximum Degree," *Mechanism and Machine Theory*, Vol. 9, 1974, pp. 115–121.

5 Paul, B., *Kinematics and Dynamics of Planar Machinery*, Prentice-Hall, 1979, pp. 284–285.

6 Mayourian, M., and Freudenstein, F., "The Development of an Atlas of the Kinematic Structures of Mechanisms," ASME Paper No. 84-DET-21.

7 Harary, F., *Graph Theory*, Addison-Wesley, 1969.


## Discussion

### On the Maximum Value of the Maximum Degree of Kinematic Chains

**Lung-Wen Tsai**[1]. The authors are to be congratulated for the derivation of the equations for the maximum value of the maximum degree of kinematic chains.

The discusser, however, would like to point out that the same results can also be derived by applying the well-known Euler's formula which states that $L=e-v+2$ for planar graphs, where $L$ denotes the number of faces or loops, including the peripheral face, $e$ denotes the number of edges and $v$ denotes the number of vertices. Since the degree of a vertex in a planar graph equals the number of loops passing through that vertex, the maximum possible degree of a vertex occurs when all the loops pass through that vertex. Hence,

$$d=e-v+2 \tag{1}$$

or, in terms of kinematic chains, we have

$$M=J-N+2 \tag{2}$$

where $d$ denotes the maximum degree of a vertex, $M$ the maximum number of joints on a link, $J$ the number of simple joints, and $N$ the number of links in a kinematic chain. This condition was pointed out earlier by Buchsbaum and Freudenstein in 1970 [1].

However, we should also keep in mind that a vertex can have at most $(v-1)$ incident edges in a graph with $v$ vertices. For kinematic chains, this implies that the value of $M$ in equation (2) should never exceed $N-1$.

Substituting $M=N-1$ into equation (2), we obtain $J=2N-3$. Hence, the following conditions should be imposed:

$$M=N-1, \qquad \text{for } J>2N-3 \tag{3}$$

Thus, we have arrived at the same conditions derived by Yan and Harary.

Furthermore, if we limit ourself to those joints with up to two degrees of freedom for planar mechanisms, and with up to three degrees of freedom for spatial mechanisms, then for all mechanisms with positive degree-of-freedom [2],

$$J\leq 2N-3 \tag{4}$$

Hence, for practical purposes, only equation (2) is sufficient.

The following special cases are worth mentioning:

(1) For one degree-of-freedom planar linkages with binary joints, the degree-of-freedom equation reduces to:

$$J=3N/2-2 \tag{5}$$

Substituting equation (5) into (2) yields

$$M=N/2 \tag{6}$$

This is in complete agreement with the result given by Rosenauer and Willis (listed as reference 5 on the paper).

(2) For one-degree-of-freedom epicyclic-gear trains, the number of turning-pair edges equals the number of links minus one, and the number of geared edges equals the number of turning-pair edges minus one. Therefore, the degree-of-freedom equation reduces to

$$J=2N-3 \tag{7}$$

Substituting equation (7) into (2) yields

$$M=N-1 \tag{8}$$

Hence, it is always possible to construct an $N$-link, one degree-of-freedom epicyclic-gear train with the maximum permissible number of joints on a link.

It is a pleasure to commend the authors for their fine work.

[1]Associate Professor, Department of Mechanical Engineering, University of Maryland, College Park, Maryland.


## References

1 Buchsbaum, F., and Freudenstein, F., "Synthesis of Kinematic Structure of Geared Kinematic Chains and Other Mechanisms," *Journal of Mechanisms and Machine Theory*, Vol. 5, 1970, pp. 367–392.

2 Mayourian, M., and Freudenstein, F., "The Development of an Atlas of the Kinematic Structures of Mechanisms," ASME Paper No. 84-DET-21, 1984.


## Authors' Closure

The authors would like to thank Professor L. W. Tsai for his discussion.

It is clear that the expression for the maximum value of the maximum degree of planar kinematic chains with revolute pairs and with positive number of degrees of freedom can be derived based on Euler's formula as pointed out by Professor Tsai. This fact was also derived early by Paul [5] and Mayourian [6] as indicated in the introduction of the paper.

In our approach, we identified that a "block" in graph theory is corresponding to a "kinematic chain" in kinematics. Here, a kinematic chain refers to a generalized kinematic chain which is connected, closed, without any cut link, and with simple generalized joints only. Based on the concept of block, we derived the generalized mathematical expressions for the maximum value of the maximum degree of generalized kinematic chains. We hope this approach can stimulate more studies on the application of the theory of blocks to kinematic number synthesis.

## Discussion

### An Application of Dual Graphs to the Automatic Generation of the Kinematic Structures of Mechanisms, Vol. 108, pp. 392–398 (86-DET-1) by W. J. Sohn and F. Freudenstein

**Jerry T. Pugh**[1]. The authors introduce the idea of using

[1]Manager Mechanical Engineering, Instrument Research and Development, Ames Division Miles Laboratories, 430 S. Beiger Street, Mishawaka, IN 46544.

the dual of the conventional graph as the starting point in the process of enumerating mechanism structures. The vertices of a dual graph correspond to the loops (faces) in the conventional graph. The edges of the dual graph correspond to loops which have one or more common edges. The authors indicate the number of common edges between two loops with a label on the corresponding edge of the dual graph. In this way, they eliminate multiple edges from the dual graph. The technique is similar to contraction of an arc to an edge in the conventional graph.

The authors give algorithms for enumerating the vertex-vertex incidence matrices of labeled dual graphs with up to five vertices. They then give an algorithm for enumerating the vertex-vertex incidence matrices of the conventional mechanism graphs.

They use characteristic polynomials to recognize and reject isomorphic conventional graphs. They note that conventional graphs generated from nonisomorphic unlabeled dual graphs cannot be isomorphic. And, that dual graphs must have the same unordered set of labels to be capable of generating isomorphic conventional graphs. They use these two rules to check for instances where the characteristic polynomials incorrectly indicate isomorphism. They report three errors for the over 1000 conventional graphs they generated.

The authors demonstrate the power of their procedure by generating 80 three-degree-of-freedom linkages with up to four loops and 726 five-loop, two-degree-of-freedom linkages they believe to be new.

The authors are to be congratulated for their work and on receiving a best paper award at the mechanisms conference. I enjoyed studying this paper and hearing the presentation given by Mr. Sohn at the conference. I look forward to the publication of his doctoral dissertation. I am especially impressed that the enumeration procedures used to generate the new structures were written in the BASIC programming language and implemented on a personal computer.

In their work, the authors use the vertex-vertex incidence matrix to represent a graph. This is the usual approach. It is quite natural to operate on arrays when using conventional computing languages like BASIC or FORTRAN. I would like to take this opportunity to suggest the use of an alternate representation which could prove useful. Specifically, I suggest representing the various relations of a graph with lists and using a list processing computer language like LISP to perform operations on these lists.

The features and relations of a graph can be easily represented with lists. One obvious example is that a graph itself can be represented as a list of adjacent vertices. The graph of Fig. 1*b*) can, for example, be represented as:

EDGES [(1 2)(2 3)(3 4)(4 5)(5 1)(3 6)(6 7)(7 1)]

As another example, consider the possibility of representing a symmetry of a graph as an equivalence relation [18] in which the vertices of the graph are partitioned into subsets of symmetric vertices. That is, a symmetry can be represented as a list of lists. For example, the symmetry of the vertices of the graph of Fig. 1(*b*) with respect to vertex 2 could be represented as:

SYMMETRY [(2)(1 3)(4 5 6 7)]

In general, a graph can have a number of symmetries, each of which could be represented in like manner. The symmetry of a graph might be useful in a number of ways. As an example, consider taking advantage of the known symmetry of a graph in order to avoid generating isomorphic colored graphs when enumerating different mechanism types from a given structure.

Although a list representation could be used with a conventional computational computer language, it is more convenient to use a language specifically designed for operation on lists. LISP is such a language. LISP is the principle language of the Artificial Intelligence (AI) community in this country. The acronym LISP comes from the words LISt Processing. Fundamental to LISP are word-like objects called atoms. Groups of atoms are used to form lists. Atoms and lists together are called symbolic expressions. The language is designed to manipulate these symbolic expressions [34].

It can prove useful to start small when beginning to work with LISP. Unfortunately, the LISP language is memory intensive and is not well-suited to use on a microcomputer. There are, however, at least two good ways to begin using a list processing language on a microcomputer. There is a modern LISP-like language called SCHEME designed specifically for the microcomputer environment. And, an excellent text is available [28]. A version of SCHEME is available from Texas Instruments, Inc. [32].

Another easy way to get started is to use the LOGO programming language. The LOGO language contains most of the essential list processing features for a minimal implementation of LISP [31]. In addition, the LOGO environment contains powerful Turtle graphic tools. A number of implementations of LOGO are available for microcomputers [29].

Although the LOGO language is most well-known for its Turtle graphics and most elementary texts concentrate on this feature, two more advanced texts [30 and 33] illustrate the powerful list processing features of LOGO.

The use of list representation may provide new insight into the problems associated with mechanism structure. Also, lists seem to me to be a more compact and versatile representation than incidence matrices. Finally, since LISP is the principal language of the AI community, techniques developed in LISP will be more compatible with their recent developments in the area of Expert Systems.

## References


28 Abelson, Harold, and Sussman, Gerald J., 1985, *Structure and Interpretation of Computer Programs*, The MIT Press, McGraw-Hill Book Company, New York.

29 Bridger, M., 1984, "Four LOGOs for the IBM-PC," *BYTE*, McGraw-Hill, Vol. 9, No. 8, pp. 287-301.

30 Harvey, B., 1985, *Computer Science LOGO Style*, Intermediate Programming, The MIT Press, Cambridge, Mass.

31 Hasemer, T., 1984, *Looking at LISP, A Guide to Bringing the Power of the Leading Artificial Intelligence Language to Microcomputers*, Addison-Wesley Publishing Company, Inc., Reading, Mass.

32 PC Scheme User's Guide, 1985, Texas Instruments Inc.

33 Thornburg, David D., 1986, *Beyond Turtle Graphics, Further Explorations of Logo*, Addison-Wesley Publishing Company, Reading, Mass.

34 Winston, Patrick H., and Horn, Berthold K. P., 1981, *LISP*, Addison-Wesley Publishing Company, Inc., Reading, Mass.


## Authors' Closure

The authors are grateful to Mr. Jerry Pugh for his thoughtful comments. As pointed out by the discusser the symmetry properties of a graph constitute a basic graph characteristic. For the computer generation of graphs, the latter are usually represented by various matrices of incidence. If, however, one wishes to enumerate graphs having specified structural characteristics (as is often required in mechanisms applications), it is necessary to eliminate isomorphic structures. In graph theory the number of isomorphic structures can be determined with the aid of Polya theory, which requires

knowledge of the symmetry properties of graphs (the group of automorphisms). The theory is elegant, but difficult to computerize. In addition, it is limited in scope, i.e., it yields the number of structures of given type, but it does not list the individual structures.

The listing of the symmetry properties of a graph can, therefore, provide important insights relative to their structural similarities and differences, as pointed out by the discusser. Languages such as LISP are certainly useful in this connection.

The algorithms developed in our paper were developed so as to require a minimum of isomorphism testing, for which the characteristic-polynomical test is basically adequate. In all graph investigations, however, the symmetry properties of graphs are basic to their understanding and the discusser's emphasis on these is well taken.