An Algorithm for Determining the Symmetry of Kinematic Chains*

Jeong Tae KIM** and Byung Man KWAK**

A computer algorithm for determining the symmetry of a kinematic chain is developed. The vertex set of the graph generated from the kinematic chain is partitioned into classes of vertices according to certain structural properties in graph theory. Permutations of vertices in each class are selected if they can map the set of edges into the original set of edges corresponding to the class. They are then concatenated with those from other classes. All the permutations that remain after checking the automorphism become the vertex-induced group of the graph, which represents the symmetry of the corresponding kinematic chain.

Key Words: Mechanism, Kinematic Chain, Graph Theory, Symmetry, Vertex-Induced Group, Automorphism

1. Introduction

The symmetry information of kinematic chains is useful in enumerating kinematic chains and generating distinct mechanisms from them^{(1),(2)}. Recently, Tuttle et al.^{(3),(4)} proposed an enumeration algorithm which uses the symmetry information of kinematic chains in order to avoid the isomorphism problem. This algorithm uses an exhaustive method to determine the symmetry information of kinematic chains at the beginning of enumeration, and updates the information during the enumeration.

There can be two ways of representing the symmetry of kinematic chains. The first is to show that certain links are symmetric. This is related to the study of finding the set of links which have the same topological position in kinematic chains. Kim and Kwak⁽⁵⁾ recently suggested a heuristic algorithm to find these symmetric links and joints of kinematic

chains. The second is to describe the type of symmetry of kinematic chains. In this case, permutational relations of the labels of the links are used to identify the symmetry of kinematic chains. No systematic research on this approach has been reported although Freudenstein⁽⁶⁾ already discussed the usefulness of permutations in describing the symmetry information.

This paper describes an algorithm by which the symmetry information of kinematic chains is represented in the form of permutations of the labels of links.

2. Symmetry of Kinematic Chains

2.1 Graph representation of kinematic chains

Graphs have been utilized for representing the structure of kinematic chains. A kinematic chain is described as a graph in which links correspond to vertices, joints to edges, and the joint connection of links corresponds to the edge connection of vertices. It is assumed that the n links of the kinematic chain are numbered or labeled with integers $1, 2, \dots, n$. The vertices of the corresponding graph are also labeled with the n integers. In this case, the graph is said to be vertex-labeled. For example, a kinematic chain and

* Received 27th November, 1990

** Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, P. O. Box # 150, Cheongryang, Seoul 130-650, Korea

its graph are drawn in Fig. 1.

In this study, an edge list is used to represent a graph because it is more compact and convenient than a matrix representation for describing the structure of the graph^{(1),(2)}. Each edge is represented by an unordered pair of vertices which are connected by the edge. The edge list of the graph in Fig. 1(b) is [(1 2) (1 3) (1 4) (2 5) (3 5) (4 6) (5 6)].

2.2 Similar vertices and edges of a graph

As mentioned earlier, one of the two ways of showing the symmetry relation in a kinematic chain is to describe symmetric links and joints of the kinematic chain. If links [joints] have topologically the same position in the structure of the kinematic chain, they are said to be symmetric. In the graph of the kinematic chain, the vertices which correspond to the symmetric links are said to be similar. Edges corresponding to the symmetric joints are also similar. For example, the similar vertices of the graph in Fig. 1(b) are described as a list: [(1 5) (2 3) (4 6)]. This list means that vertices 1 and 5 are similar, vertices 2 and 3 are similar, and so on. The similar edges of the graph are also described as a list: [(1 2) (1 3) (2 5) (3 5)][(1 4) (5 6)]. Four edges (1 2), (1 3), (2 5), and (3 5) are similar, as are edges (1 4) and (5 6).

2.3 Vertex-induced group of a graph

An alternative way to represent the symmetry relation in a kinematic chain is to describe the type of symmetry. This kind of symmetry is usually described as permutations of the labels of links. It is related to the vertex mappings of the corresponding graph. The vertex mapping is described by a permutation of vertices. If a vertex mapping maps the graph into itself, it is called an automorphism of the graph. For example, Fig. 2 shows that a permutation of vertices maps a graph into another graph. Because the two graphs in the figure have the same edge list, they are the same graph. Consequently, the permutation of vertices is an automorphism of the graph in Fig. 2(a).

To describe the symmetry of a graph, all of the automorphisms of the graph should be found. The set of all automorphisms is called the vertex-induced

group of the graph, or simply the group of the graph⁽⁶⁾. The vertex-induced group of the graph in Fig. 2(a) is shown in Table 1. The automorphic mapping corresponding to the third row in the permuted sequences denotes the vertex mapping: $1 \rightarrow 5$, $2 \rightarrow 2$, $3 \rightarrow 3$, $4 \rightarrow 6$, $5 \rightarrow 1$, and $6 \rightarrow 4$.

If the edges of the graph are labeled rather than the vertices, there exists a set of edge permutations which maps the graph into itself. This is called the edge-induced group of the graph. For graphs in which at most one edge connects two distinct vertices, the vertex-induced group and edge-induced group have a 1:1 correspondence. For example, Table 2 shows the edge permutation corresponding to the vertex permutation in Fig. 2.

This kind of symmetry is useful in enumerating kinematic chains and selecting distinct ground links of kinematic chains, as shown by Tuttle et al.^{(3),(4)}.

The purpose of this study is to propose an algo-

Table 1 Set of automorphisms of the graph in Fig. 1(b)

original sequence	1	2	3	4	5	6
permuted sequences	1	2	3	4	5	6
	1	3	2	4	5	6
	5	2	3	6	1	4
	5	3	2	6	1	4

Table 2 Edge permutation corresponding to the vertex permutation in Fig. 2

original edges	permuted edges
(1 2)	(2 5)
(1 3)	(3 5)
(1 4)	(5 6)
(2 5)	(1 2)
(3 5)	(1 3)
(4 6)	(4 6)
(5 6)	(1 4)

Fig. 1 Kinematic chain and its graph representation

Fig. 2 Vertex permutation mapping from the graph in (a) into the graph in (c)

ithm which can find the vertex-induced group of a graph corresponding to a kinematic chain.

3. Finding the Vertex-Induced Group

A simple method of finding the vertex-induced group of the graph corresponding to a kinematic chain is to list it exhaustively. It requires $(n!)$ permutations of vertices, labelings of vertices, and comparisons of edge lists when there are n vertices. The algorithm proposed to overcome this inefficiency is composed of three parts. The first is to partition the vertex set into classes of vertices. Next is to find the feasible permutations of vertices in each class. The last is to concatenate the feasible permutations for all classes and find the vertex-induced group of the graph.

3.1 Vertex classification

To improve the brute force method, the vertex set is partitioned into disjoint classes of vertices, $V_1, V_2, \dots, V_m$, which contain $p_1, p_2, \dots, p_m$ vertices, respectively. Then permutations are restricted to those of vertices in the same class. This approach reduces the number of permutations from $(n!)$ to $(p_1!)(p_2!)\dots(p_m!)$. This vertex classification is done using the invariant property in graph theory.

A simple property of a vertex is the degree of the vertex, that is, the number of edges incident at the vertex. If some vertices have the same degree, they are collected in a class. Then the vertices in this class cannot be mapped into those in another class because the corresponding vertex mappings cannot map the graph into itself. For example, the vertex set of the graph in Fig. 3 is partitioned into $[(1\ 2\ 8\ 9)(3\ 4\ 5\ 6\ 7)]$ according to the degree of the vertex. It can be shown that the vertex mappings between vertices in distinct classes cannot be allowed. For example, if a vertex mapping is $1 \rightarrow 3, 3 \rightarrow 1, 2 \rightarrow 4$, and $4 \rightarrow 2$, then edge $(1\ 2)$ is mapped into $(3\ 4)$. Because edge $(3\ 4)$ does not exist in the original edge list, this mapping cannot be an automorphism.

In this example, the number of permutations to be

edge list : $((1\ 2)(1\ 3)(1\ 4)(2\ 5)(2\ 6)$
 $(3\ 8)(4\ 8)(5\ 9)(6\ 7)(7\ 9)(8\ 9))$

Fig. 3 Graph for vertex classification

generated is reduced from $(9!) = 362\ 880$ to $(4!)(5!) = 2880$.

To refine the classification, a sequence of the degrees of vertices adjacent to a vertex can be used. The vertices which have the same sequence are collected in a class. For example, vertices of the graph in Fig. 3 are partitioned into $[(6\ 7)(3\ 4\ 5)(1\ 2\ 8\ 9)]$. Because vertices 6 and 7 have the same sequence $(2\ 3)$, they become a class. Vertices 3, 4, and 5 are also collected in another class because they have the same sequence $(3\ 3)$. In this case, the number of permutations is reduced to $(2!)(3!)(4!) = 288$.

The finest partitioning of the vertex set is the automorphism partitioning^{(7),(8)}. It partitions the vertex set into classes of vertices that must be mapped to one another under any automorphism of the graph. In this case, vertices which are similar to one another are collected in a class. An algorithm for finding the similar vertices of a graph which has been studied by the authors⁽⁶⁾ is used in this study. The automorphism partitioning of vertices of the graph in Fig. 3 is $[(5)(1\ 8)(2\ 9)(3\ 4)(6\ 7)]$ by this algorithm. In this partitioning, the number of permutations is $(1!)(2!)(2!)(2!)(2!) = 16$.

3.2 Feasible permutations of vertices in a class

The purpose of this study is to find the set of vertex mappings which map the edge list of a graph into itself. If the vertex set of the graph is partitioned into classes of vertices by the methods in section 3.1, the vertex mapping between two vertices which are not in a class cannot be a part of an automorphism of the graph. Thus, the permutations of vertices in different classes are not considered.

Permutations of vertices in one class do not always map the edge list into itself. In order to eliminate these undesirable permutations, edges which connect vertices within the class are found. Then they are mapped by each permutation. If the edges are not mapped into edges in the edge list, the corresponding permutation is discarded as an infeasible case.

Although the vertex classification reduces undesirable permutations, the resulting number of permutations of vertices in a class can still be large. Therefore, it is better not to generate simultaneously all of the permutations of vertices in a class. Instead, permutations are generated one by one in the lexicographical order in this study. An algorithm to yield the next permutation of a given permutation in the lexicographical order is described as follows⁽⁹⁾:

- (a) Examine the given permutation, $(a_1\ a_2\ \dots\ a_n)$, element by element from right to left. Let a_m be the rightmost element such that $a_m < a_{m+1}$.
- (b) Examine the permutation element by element from right to left again. Let a_p denote the

rightmost element such that $a_m < a_p$.

(c) Interchange a_m and a_p .

(d) Interchange a_{m+1} and a_n , a_{m+2} and a_{n-1} , a_{m+3} and a_{n-2} and so on.

As an example, consider a permutation: (1 6 2 5 4 3). According to (a), the subscript m is equal to 3 because 2 is less than 5 in the given permutation. According to (b), the subscript p is 6 because 3 is greater than 2. By step (c), a_3 and a_6 are interchanged, that is, 2 and 3 should be exchanged. The resulting permutation becomes (1 6 3 5 4 2). Finally, according to (d), $a_4 (=5)$, is interchanged with $a_6 (=2)$, obtaining (1 6 3 2 4 5) as the next permutation.

For example, a class of vertices 1, 2, 5, and 6 of the graph in Fig. 4 is considered. The edge list of the graph is [(1 2) (1 3) (2 4) (3 4) (3 5) (4 6) (5 6)]. The edges which connect vertices in the class are (1 2) and (5 6). The permutations generated in the lexicographic order are shown in Table 3. Some of them do not map the edges into edges in the edge list. For example, the third permutation maps them into (1 5) and (2 6) which do not belong to the edge list. The feasible permutations are underlined by dashed lines in Table 3.

3.3 Concatenation of feasible permutations

An automorphism of a graph is a permutation of all vertices of the graph. To generate the set of all automorphisms, feasible permutations for a class of vertices should be concatenated with those for other classes. However, the permutations which are made by concatenating feasible permutations for two classes are not always feasible. To check the feasibility of the newly generated permutations, a two-step algorithm is devised. The first step is to find edges which connect vertices in one class with those in another class. Then each of the new permutations is tested to determine whether it maps the edges into edges on this edge list. If it does not, it is not feasible.

Next, after feasible permutations for the two classes are concatenated, the two classes are merged into one class for further combinations with other remaining classes. This idea is described in the next section.

For example, feasible permutations for two classes, (1 2 5 6) and (3 4), of the graph in Fig. 4 are concatenated and tested. In Table 3, eight feasible permutations of vertices 1, 2, 5, and 6 are specified. Feasible permutations of vertices 3 and 4 are mappings $(3 \rightarrow 3, 4 \rightarrow 4)$ and $(3 \rightarrow 4, 4 \rightarrow 3)$. The newly concatenated permutations are described in Table 4. Among 16 permutations, only 4 permutations are feasible. They are underlined by dashed lines in the table. Because the merged class contains all of the vertices, these 4 permutations constitute a vertex-

induced group of the graph.

3.4 Algorithm of finding a vertex-induced group of a graph

An edge list E and a vertex set $V = (1\ 2\ 3 \cdots n)$ of a graph are given.

Step (1) Initialization

(a) **active_e_set** $\leftarrow E$

(b) $i \leftarrow 1$

(c) **passed_v_set** $\leftarrow ()$

Table 3 Permutations of vertices 1, 2, 5, and 6 of the graph in Fig. 4 (feasible permutations are underlined by dashed lines)

No.	1 2 5 6	No.	1 2 5 6
1	<u>1 2 5 6</u>	13	5 1 2 6
2	<u>1 2 6 5</u>	14	5 1 6 2
3	1 5 2 6	16	5 2 1 6
4	1 5 6 2	16	5 2 6 1
5	1 6 2 5	17	<u>5 6 1 2</u>
6	1 6 5 2	18	<u>5 6 2 1</u>
7	<u>2 1 5 6</u>	19	6 1 2 5
8	<u>2 1 6 5</u>	20	6 1 5 2
9	2 5 1 6	21	6 2 1 5
10	2 5 6 1	22	6 2 5 1
11	2 6 1 5	23	<u>6 5 1 2</u>
12	2 6 5 1	24	<u>6 5 2 1</u>

Table 4 Concatenated permutations of feasible permutations for two classes (1 2 5 6) and (3 4) of the graph in Fig. 4 (permutations underlined by dashed lines are feasible for combined class)

1 2 3 4 5 6	1 2 3 4 5 6
<u>1 2 3 4 5 6</u>	1 2 4 3 5 6
1 2 3 4 6 5	1 2 4 3 6 5
2 1 3 4 5 6	2 1 4 3 5 6
2 1 3 4 6 5	<u>2 1 4 3 6 5</u>
<u>5 6 3 4 1 2</u>	5 6 4 3 1 2
5 6 3 4 2 1	5 6 4 3 2 1
6 5 3 4 1 2	6 5 4 3 1 2
6 5 3 4 2 1	<u>6 5 4 3 2 1</u>

Fig. 4 Graph for finding feasible permutations of vertices

(d) **perm_set** $\leftarrow$ ()

Step (2) Vertex classification

Using the methods in section 3.1, partition the vertex set V into disjoint subsets $V_1, V_2, \dots, V_m$ in which $|V_1| \leq |V_2| \leq \dots \leq |V_m|$.

Step (3) Consider vertices in V_i

(a) Assume V_i is a list $(a_1 a_2 \dots a_k)$ in which $a_1 < a_2 < \dots < a_k$.

(b) **new_perm_set** $\leftarrow$ ()

Step (4) Find edges related to the vertices of V_i .

(a) **in_e_set** $\leftarrow$ the set of edges in **active_e_set** which connects vertices in V_i .

(b) **out_e_set** $\leftarrow$ the set of edges in **active_e_set** connecting vertices in **passed_v_set** with those in V_i .

Step (5) Generate starting permutation of the vertices of V_i .

(a) **perm** $\leftarrow$ $(a_1 a_2 a_3 \dots a_k)$

Step (6) Mapping of edges in **in_e_set**

(a) If **in_e_set** is the empty set, go to step (7).

(b) If all edges in **in_e_set** are mapped into edges in **in_e_set** by **perm**, then go to step (7). If not, go to step (8).

Step (7) Mapping of edges in **out_e_set**

(a) Concatenate **perm** with each vertex mapping in **perm_set**.

(b) If the concatenated vertex mapping maps all edges in **out_e_set** into edges in **out_e_set**, or if **out_e_set** is the empty set, then add the vertex mapping to **new_perm_set**.

Step (8) Generate the next lexicographical permutation of **perm.**

(a) **perm** $\leftarrow$ the next permutation of **perm**

(b) If **perm** is the starting permutation, then go to step (9). If not, go to step (6).

Step (9) Update variables

(a) **perm_set** $\leftarrow$ **new_perm_set**

(b) Add the vertices in V_i to **passed_v_set**.

(c) Remove the edges in **in_e_set** and **out_e_set** from **active_e_set**.

(d) $i \leftarrow i + 1$. If $i \leq m$, then go to step (3). Otherwise, go to step (10).

Step (10) **perm_set is the vertex-induced group of the given graph.**

4. Examples

Two graphs are considered as examples. The first example is the graph in Fig. 3. The vertex-induced group of the graph is shown in Table 5. It is described as permutations of vertices. The second one is the graph of the kinematic chain in Fig. 5. The vertex-induced group of the graph is given in Table 6.

5. Conclusions

We have developed a system that determines the symmetry of kinematic chains in the form of permutations of vertices of corresponding graphs. The conclusions are as follows.

- (1) Two kinds of symmetry of kinematic chains are defined. The first is related to the symmetry of links and joints. The second is to the type of symmetry of kinematic chains.
- (2) The type of symmetry of kinematic chains is determined in this study in terms of the vertex-induced group of graphs.
- (3) To reduce the number of undesirable infeasible permutations, the vertex set is partitioned into several classes. Then only permutations of vertices in the same class need be considered.

Acknowledgements

The authors gratefully acknowledge the financial

Table 5 Vertex-induced group of the graph in Fig. 3

1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9
1	2	4	3	5	6	7	8	9
8	9	3	4	5	7	6	1	2
8	9	4	3	5	7	6	1	2

Table 6 Vertex-induced group of the graph in Fig. 5

1	2	3	4	5	6	7	8	9	10	11	12
1	2	3	4	5	6	7	8	9	10	11	12
1	8	3	4	12	6	7	2	9	10	11	5
6	5	10	9	2	1	11	12	4	3	7	8
6	12	10	9	8	1	11	5	4	3	7	2

Fig. 5 Kinematic chain and its graph for finding symmetry

support of the Ministry of Science and Technology, Korea, and the cooperative relationship with the Advanced Design Method Laboratory of the Ohio State University, U.S.A., through a joint research project agreement.

References

- (1) Pugh, J. T., Discussion-An Application of the Linkage Characteristic Polynomial to the Topological Synthesis of Epicyclic Gear Trains, *Trans. ASME, J. Mech. Transm. Autom. Des.*, Vol. 109, No. 3 (1987), p. 337.
- (2) Pugh, J. T., Discussion-An Application of Dual Graphs to the Automatic Generation of the Kinematic Structures of Mechanisms, *Trans. ASME, J. Mech. Transm. Autom. Des.*, Vol. 109, No. 4 (1987), p. 490.
- (3) Tuttle, E. R., Peterson, S. W. and Titus, J. E., Further Application of Group Theory to the Enumeration and Structural Analysis of Basic Kinematic Chains, *Trans. ASME, J. Mech. Transm. Autom. Des.*, Vol. 111, No. 4 (1989), p. 494.
- (4) Tuttle, E. R., Peterson, S. W. and Titus, J. E., Enumeration of Basic Kinematic Chains Using the Theory of Finite Groups, *Trans. ASME, J. Mech. Transm. Autom. Des.*, Vol. 111, No. 4 (1989), p. 498.
- (5) Kim, J. T. and Kwak, B. M., An Algorithm of Topological Ordering for Unique Representation of Graphs, Accepted for publication in *Trans. ASME, J. Mech. Des.* (1991).
- (6) Freudenstein, F., The Basic Concepts of Polya's Theory of Enumeration, with Application to the Structural Classification of Mechanisms, *Jnl. Mechanisms*, Vol. 3 (1967), p. 275.
- (7) Read, R. C. and Corneil, D. G., The Graph Isomorphism Disease, *J. Graph Theory*, Vol. 1 (1977), p. 339.
- (8) Gould, G., *Graph Theory*, Benjamin/Cummings Publishing Co., Menlo Park, California (1988), p. 258.
- (9) Liu, C. L., *Elements of Discrete Mathematics* (2nd ed.), McGraw-Hill, New York (1987), p. 100.