Automatic Generation of Kinematic Joint Model from 3D Assembly Model for Mechanical Product Design

Xue-liang HUANG*, Yan CHENG and Lei WANG

Institute of Systems Engineering, CAEP, Mianyang Sichuan 621999, China

*Corresponding author

Keywords: Assembly constraint, Kinematic joint, Constraint graph, Equivalent constraint substitution.

Abstract. To achieve the seamless integration of 3D assembly design and kinematics and dynamics analysis for the development of mechanical products, an automatic generation method of kinematic joint model from 3D assembly model is proposed, which can extract the same mechanism information from different assembly constraint models of the same mechanical product. Firstly, an assembly constraint graph was constructed to represent the 3D assembly constraint model, and an equivalent constraint substitution method was employed to simplify the assembly constraint graph for extracting its intrinsic kinematic closed-chains. Secondly, the screw theory was introduced to recognize the kinematic joint induced by a combination of assembly constraints between two adjacent parts, so that the assembly constraint graph could be converted to kinematic joint graph. Subsequently, several identification rules of sub-assemblies which consist of several adjacent parts was established, and the sub-assemblies in the geometric constraint model were identified by using the analysis of degree of freedom, geometric contact analysis and graph search algorithm. In such a way, an assembly constraint model could be converted to a kinematic joint model automatically. Finally, two typical examples have been given to validate the correctness and effectiveness of the proposed method.

Introduction

Automatic generation of kinematic joint model plays an important role in the seamless integration of computer-aided assembly design and kinematics and dynamics analysis [1], which can be also used in the fields such as virtual simulation, assembly sequence planing, and 3D assembly model classification and retrieval, etc. In the past decades, a lot of effort has been made to achieve the automatic generation of kinematic joint model, and many scholars have done a lot of in-depth study on assembly feature recognition [2-3], kinematic joint recognition [1,4-6], automatic detection of fasteners [7], etc.

However, the current research might ignore an important phenomenon that the cycle structure of the adjacent relation graph of parts derived from the actual assembly design of mechanical product, is always inconsistent to its intrinsic kinematic joint graph. Meanwhile, the assembly constraints imposed by different designers when they assemble the same product model are usually different, so there exist different assembly constraint models for an assembly of mechanical product. It means that assembly constraint graph derived from the actual assembly design is also inconsistent to the intrinsic kinematic joint graph of the assembly model. For instance, the assembly constraint models illustrated in Figure1(b), Figure1(c) and Figure1(d) are produced by three designers when they assemble the product model depicted in Figure1(a). Although these Assembly Constraint Graphs (ACG) illustrated in Figure1 are derived from the same assembly model, their cycle structures are completely different and inconsistent to the intrinsic kinematic joint graph. If the inconsistency can not be eliminated, the kinematic joint models generated from the assembly constraint models of the same mechanical product are diversity and can not correctly reflect the characteristics of mechanism inside the mechanical product.
In order to solve the problem, this paper put forward a method which makes good use of the equivalent constraint substitution to simplify assembly constraint graph, and can generate the same kinematic joint models from multiple equivalent assembly constraint models with different ACGs.

**Simplification of Assembly Constraint Model**

The diversity of assembly constraint graphs stems from the equivalence of assembly constraint systems. The existence of cycle structures in assembly constraint graph does not mean that there are corresponding kinematic closed-chains in the assembly model of mechanical product. As shown in Figure 2, the ACG I shown in Figure 1(b) has four basic cycles, namely \( l_{11}, l_{12}, l_{13} \) and \( l_{14} \). The ACG II shown in Figure 1(c) has three basic cycles, namely \( l_{21}, l_{22} \) and \( l_{23} \). The ACG III shown in Fig 1(d) has two basic cycles, namely \( l_{31} \) and \( l_{32} \). In fact, the assembly model shown in Figure 1(a) contains only one kinematic closed-chain (actually a four-bar linkage) consisting of parts \( B_1, B_4, B_8 \) and \( B_5 \) connected by three revolute joints and one cylindrical joint. Obviously, some basic cycles in the ACGs shown in Figure 1 are not kinematic closed-chains. However, in the ACGs shown in Figure 1(b), 1(c) and 1(d), the motion spaces of the parts \( B_1, B_4, B_8 \) and \( B_5 \) are completely equivalent. In order to generate the correct kinematic joint model from assembly constraint model, the assembly constraint model should be simplified to eliminate the pseudo kinematic closed-chains.

Due to the equivalence of assembly constraints, the cycle structure of ACG can be changed by adding a new assembly constraint to replace an original assembly constraint equivalently. Then, by using the equivalent substitution of assembly constraints to eliminate the pseudo kinematic
closed-chain, the equivalent assembly constraint model which has the same cycle structure as the corresponding kinematic joint model can be obtained.

To achieve the simplification of ACG, we introduce a weighted Adjacent Relation Graph (ARG) of minimum basic cycles $G_i = (V_i, E_i, W_i)$ to represent the cycle structure of ACG. In the ARG, a vertex represents a minimum basic cycle in ACG, and an edge represents the edge-adjacent relation between two minimum basic cycles sharing the same edges in ACG. The weight of a vertex $v_i$ is equal to $m + a*n$, where $m$ is equal to the least number of assembly constraints in all non-communal edges of the basic cycle represented by the vertex $v_i$, and $a$ is a constant which can be taken the value 0.01 in this paper. The weight of an edge is equal to the number of communal edges between two basic cycles. Then, an algorithm for simplifying the ACG can be described as below.

**Algorithm 1. Simplify ACG.**

The input: an ACG $G$. 
The output: an equivalent ACG $G_e$.

Step1: Decompose the input ACG $G_s$ into a series of bi-connected subgraphs.

Step2: For each subgraph $G_i$ which contains a single cycle, process the subgraph with equivalent constraint substitution method\(^8\).

Step3: For each subgraph $G_j$ which contains multiple cycles, compute its Minimum Cycle Basis (MCB)\(^9\), and construct an ARG of these cycles in MCB. Then, repeat using equivalent constraint substitution method to deal with the basic cycle of subgraph $G_j$ whose vertex’s weight is the smallest in the ARG, until all basic cycles of subgraph $G_j$ have been processed.

Step4: Return the equivalent $G_e$, and the algorithm terminates.

For the convenience of the following statement, we use $CoiLL(B_i, B_j)$ to represent a coaxial constraint between two parts $B_i$ and $B_j$, and use $CoiFF(B_i, B_j)$ to represent a coplanar constraint between two parts $B_i$ and $B_j$.

While simplifying the ACGs illustrated in the Figure 1 with the above algorithm, the evolution of these ACGs and their ARGs can be found in Figure 3, and the detail process is as follows.

1) For the ACG I shown in Figure 1(b), the ARG of its four basic cycles $l_{11}, l_{12}, l_{13}$ and $l_{14}$ can be found in Figure 2. According to the degree of vertex in ARG, the basic cycles $l_{12}, l_{13}$ and $l_{14}$ should be handled firstly, and the three coaxial constraint $CoiLL(B_2, B_4), CoiLL(B_3, B_5)$ and $CoiLL(B_4, B_7)$ can be substituted by $CoiLL(B_1, B_3), CoiLL(B_1, B_5), CoiLL(B_4, B_8)$ to break basic cycles $l_{12}, l_{13}$ and $l_{14}$. Then, the basic cycle $l_{11}$ can be reduced by substituting $CoiLL(B_5, B_6)$ with $CoiLL(B_5, B_8)$.

2) For the ACG II shown in Figure 1(c), the ARG of its three basic cycles $l_{21}, l_{22}$ and $l_{23}$ can be found in Figure 2. According to the degree of vertex in ARG, the basic cycles $l_{22}$ and $l_{23}$ should be handled firstly, and these cycles can be broken by replacing $CoiLL(B_2, B_4), CoiLL(B_3, B_5)$ with $CoiLL(B_1, B_4), CoiLL(B_1, B_5)$ equivalently. Then, the basic cycle $l_{21}$ can be reduced by replacing $CoiLL(B_5, B_6), CoiFF(B_4, B_7)$ and $CoiLL(B_4, B_7)$ with $CoiLL(B_5, B_8), CoiFF(B_4, B_8)$ and $CoiLL(B_4, B_8)$.

3) For the ACG III shown in Figure 1(d), the ARG of its two basic cycles $l_{31}$ and $l_{32}$ can be found in Figure 2. According to the degree of vertex in ARG, the basic cycles $l_{31}$ should be handled firstly, and it can be broken by replacing $CoiLL(B_2, B_4)$, with $CoiLL(B_1, B_4)$ equivalently. Then, the basic cycle $l_{31}$ can be reduced by replacing $CoiLL(B_5, B_6), CoiFF(B_4, B_7), CoiLL(B_4, B_7), CoiFF(B_5, B_5)$ and $CoiLL(B_3, B_5)$ with $CoiLL(B_5, B_8), CoiFF(B_4, B_8), CoiLL(B_4, B_8), CoiFF(B_4, B_8)$ and $CoiLL(B_1, B_5)$ equivalently.
Identification of Kinematic Joint and Sub-assembly

Identification of Kinematic Joint from Assembly Constraints

In the assembly constraint model, there are usually several assembly constraints between two parts. To realize the conversion of the assembly constraint model to the kinematic joint model, it is necessary to identify the kinematic joint from these assembly constraints between two parts. Screw theory is a powerful mathematical tool for the analysis of kinematics of the mechanism, and can be used to define assembly constraint and kinematic joint.\textsuperscript{[10]} Once the twist matrix representation of each type of assembly constraints has been modeled, the infinitesimal relative motion and constraint that results from arbitrary combinations of assembly constraints can be calculated by using the intersection algorithm of twist matrices. Given a disturbance of relative position or posture between two parts in accordance with their infinitesimal relative motion, the satisfaction of assembly constraints between two parts can be analyzed to determine the kinematic joint corresponding to a
combination of assembly constraints. More detail description about the identification of kinematic joints from arbitrary combination of assembly constraints can be found in literature[6].

Identification of Sub-assembly in Assembly Model

In the assembly model of mechanical product, not all parts have to be involved as an independent component in kinematics and dynamics analysis. The mechanical joint elements such as bolts, nuts, washers, screws, locating pins, locating keys and rivets should be combined with their adjacent parts as a sub-assembly to participate in kinematics or dynamics analysis. Given a standard parts library, the above mechanical joint elements can be recognized by using 3D model retrieval algorithm. Then the following rules can be given to identify sub-assemblies.

**Rule 1.** If two or more adjacent parts (or sub-assemblies) have zero degrees of freedom of relative motion, then these parts (or sub-assemblies) are fully constrained and should be merged into a sub-assembly.

**Rule 2.** If the part $B_i$ is a threaded connector, and the set of the threaded connectors adjacent to $B_i$ is $S_i$, and the set of the parts (or sub-assemblies) adjacent to the threaded connectors in the set $S_i$ is $S_j$, and the set of the threaded connector adjacent to the parts in the set $S_i$ is $S_k$, then all the parts (or sub-assemblies) in the set $S_i \cup S_j \cup S_k$ should be merged into a sub-assembly.

**Rule 3.** If the part $B_i$ is a locating pin, and the set of the parts (or sub-assemblies) adjacent to $B_i$ is $S_i$, the parts (or sub-assemblies) in the set $S_i$ and $B_i$ should be merged into a sub-assembly. If $B_i$ is a pin shaft, and $B_j$ is a adjacent part (or sub-assembly) of $B_i$, then they should be merged into a sub-assembly.

**Rule 4.** If the part $B_i$ is a key, and the set of the parts (or sub-assemblies) adjacent to $B_i$ is $S_i$, then all the parts in $S_i$ and $B_i$ should be merged into a sub-assembly.

**Rule 5.** If the part $B_i$ is a rivet, the set of the part (or sub-assembly) adjacent to $B_i$ is $S_i$, and the set of the rivets adjacent to the parts in the set $S_i$ is $S_j$, then all the parts (or sub-assemblies) in the set $S_i \cup S_j$ should be merged into a sub-assembly.

**Rule 6.** If two adjacent parts $B_i$ and $B_j$ are interference fits, then they should be merged into a sub-assembly.

According to the rules 1 to 6, the analysis of degree of freedom, the analysis of geometric contact and the graph search algorithm can be used to identify all sub-assemblies in the assembly constraint model.

By using the above method, eight kinematic joints and two sub-assemblies in the assembly constraint model shown in Figure 3 can be identified, and the generated kinematic joint model is shown in Figure 4.

![Figure 4. Recognition of the kinematic joints and sub-assemblies.](image-url)
Examples

In this section, two typical examples in practical assembly design are analyzed to verify the correctness and effectiveness of the proposed method.

Figure 5(a) illustrates an assembly geometric model of the clamping mechanism, whose original ACG is shown in Figure 5(b). In the MCB of the original ACG, there are five basic cycles, such as \( l_1 = \{ B_1, B_2, B_3 \}, \, l_2 = \{ B_1, B_3, B_6, B_7, B_8 \}, \, l_3 = \{ B_3, B_4, B_6 \}, \, l_4 = \{ B_3, B_4, B_5 \} \) and \( l_5 = \{ B_3, B_5, B_7, B_6 \} \). The ARG of such five basic cycles is illustrated in Figure 5(f).

After simplifying the ACG illustrated in Figure 5(b) with the proposed algorithm, we can obtain an equivalent ACG showed in Figure 5(c), which can be transformed into the simplified kinematic joint graph depicted in Figure 5(e) by using kinematic joint recognition and sub-assembly recognition.

Figure 6(a) illustrates an assembly geometric model of the transmission mechanism, whose original ACG is shown in Figure 6(b). In the MCB of the original ACG, there are eleven basic cycles, such as \( l_1 = \{ B_1, D_3, E_3, C_{12}, D_6, B_2, D_3, C_{11}, E_2, D_2 \}, \, l_2 = \{ B_1, D_1, E_1, C_{10}, D_4, B_2, D_5, C_{11}, E_2, D_2 \}, \, l_3 = \{ B_1, C_3, D_3 \}, \, l_4 = \{ B_1, C_1, D_1 \}, \, l_5 = \{ D_3, C_9, E_3 \}, \, l_6 = \{ D_1, C_7, E_1 \}, \, l_7 = \{ B_2, C_6, D_6 \}, \, l_8 = \{ B_2, C_4, D_4 \}, \, l_9 = \{ B_1, C_2, D_2 \}, \, l_{10} = \{ D_2, C_8, E_2 \} \) and \( l_{11} = \{ B_2, C_5, D_3 \} \). The ARG of such eleven basic cycles is illustrated in Figure 6(f).

After simplifying the ACG illustrated in Figure 6(b) with the proposed algorithm, we can obtain an equivalent ACG showed in Figure 6(c), which can be transformed into the simplified kinematic joint graph depicted in Figure 6(e) by using kinematic joint recognition and sub-assembly recognition.
**Conclusion**

In this paper, an automatic generation method of kinematic joint model from assembly constraint model of mechanical product is proposed, which can eliminate the diversity of assembly constraint models and extract the intrinsic kinematic closed-chains. So that, the proposed method can overcome the drawback of existing methods that generate different kinematic joint models from multiple equivalent assembly constraint models derived from the same mechanical product, and can generate the kinematic joint model which is conform to the essential kinematic characteristics of mechanical product. The proposed method is not only beneficial for the seamlessly integration between variational assembly design and kinematics and dynamics analysis, but also beneficial for 3D assembly model classification and retrieval based on kinematic characteristics.

**Acknowledgement**

This research was financially supported by the National Science Foundation of China.(Grant No. 51505446).

**References**

[1] C.J. Wan, T.J Tan, Z.Y. Liu, et al. Automatic recognition and extraction of mechanism information for virtual prototyping, China Mechanical Engineering, 15.18 (2004) 1630-1634.

[2] Sung, Raymond CW, Jonathan R. Corney, Doug ER Clark. Automatic assembly feature recognition and disassembly sequence generation, ASME Journal of Computing and Information Science in Engineering, 1.4 (2001) 291-299.
[3] K. Sambhoos, K. Bahattin, N. Rakesh. Extracting assembly mating graphs for assembly variant design, ASME Journal of Computing and Information Science in Engineering, 9.3 (2009) 034501.

[4] W.L. Xu, S.Y. Zhang, G.D Yi, et al. Motion constraint recognition and its application oriented to multibody simulation, Chinese Journal of Mechanical Engineering, 44.6 (2008) 137-142.

[5] Z.X. Zhang, J.H. Liu, R.X. Ning. Automatic recognition method of kinematic pair in virtual assembly environment, Computer Integrated Manufacturing Systems, 17.1 (2011) 62-68.

[6] X.L. Huang, L.P. Chen, B.X. Wang. Solving assembly constraint closed-loops with projection transformation method, Journal of Computer-Aided Design and Computer Graphics, 22.12 (2010) 2138-2146.

[7] N. Rafibakhsh, W.F. Huang, and M.I. Campbell. Automatic detection of fasteners from tessellated mechanical assembly models, ASME Journal of Computing and Information Science in Engineering, 18.1 (2018) 011005.

[8] X.L. Huang, B.X. Wang, L.P. Chen, et al. Equivalence analysis of 3D geometric constraint systems, Journal of Software, 22.5 (2011) 1106-1120.

[9] T. Kavitha, Ch. Liebchen, K. Mehlhorn, et al. Cycle bases in graphs characterization, algorithms, complexity, and applications, Computer Science Review, 3.4 (2009) 199-243.

[10] J.D. Adams, D.E. Whitney. Application of screw theory to constraint analysis of mechanical joined by features, ASME Journal of Mechanical Design, 123.1 (2001) 26-31.