

# Geometric Model Based Displacement Analysis System of Mechanisms\*

Naoki UCHIYAMA\*\*, Eiji ARAI\*\*\*  
and Masanori IGOSHI\*\*\*\*

This paper deals with a software system for the displacement analysis of arbitrary mechanisms, which has been discussed by many researchers. However, concerning the topological analysis of a mechanism and the automated generation of joint information, several problems remain unsolved. We propose more general methods than those previously reported in order to handle spatial constraints involving higher pairs by using solid models and to analyze the structure of the mechanism by considering the directions of multiple closed loops. Furthermore, the technique of mathematical programming is applied to the displacement analysis. The practical applicability of the developed system with the methods described in this paper is demonstrated through several examples.

**Key Words:** Computer-Aided Design (CAD), Computer-Aided Manufacturing (CAM), Design, Mechanism, Kinematic Simulation

## 1. Introduction

Several software systems for analyzing kinematic motion of mechanisms have been developed by many researchers. Sheth and Uicker developed a system called IMP (integrated mechanism program) by extending the D-H notation method<sup>(1),(2)</sup>. This system can analyze not only kinematic characteristics but also static and dynamic ones. It is, however, restricted to closed-loop mechanisms. Although Nagamatsu et al. have extended the IMP to arbitrary mechanisms including opened- and closed-loop mech-

anisms, it is not clear how the structures of these mechanisms<sup>(3)</sup> are analyzed. Kuroiwa et al. took note of this point and developed a method for structural analysis using a reachability matrix<sup>(4)</sup>. However, the directions of multiple closed loops were not the focus of their research. To analyze the kinematic motion of mechanisms numerically, we must orient multiple closed loops that share several joints along the same direction. If the closed loops are in different directions, we will have to change the sign of the value representing the displacement during calculation.

On the other hand, there still exists a problem of generating joint information. In most commercial software, system users need to input joint information, such as what type of joint is created between two or more parts, and which direction the parts can move. This is troublesome for the users because 3-D coordinate values of the geometry of mechanism must be dealt with. To automate this work, several methods for extracting joint information have already been presented. Arai and Iwata, for example, defined a table whose rows and columns denote the contact states between parts in order to classify a given contact state into one of the joint types<sup>(5)</sup>, such as

\* Received 13th March, 1995. Japanese original: Trans. Jpn. Soc. Mech. Eng., Vol. 60, No. 575, C (1994), p. 2468-2474 (Received 16th December, 1993)

\*\* Department of Mechanical Engineering, Toyohashi University of Technology, 1-1 Hibarigaoka, Tempaku, Toyohashi, Aichi 441-8122, Japan

\*\*\* Department of Manufacturing Science, Graduate School of Engineering, Osaka University, 2-1 Yamadaoka, Suita, Osaka 565-0871, Japan

\*\*\*\* Department of Precision Engineering, Tokyo Metropolitan University, 1-1 Minami-osawa, Hachioji, Tokyo 192-0364, Japan

prismatic pair or a revolute pair. Kim and Lee also developed a series of rules for identifying the joint type from the given contact states<sup>(6)</sup>. Nevertheless, both methods still leave the ambiguity for the case that the contact states between parts are very complicated. Moreover, these two methods cannot handle higher pairs.

In this paper, we first propose a method for generating joint information from a given geometric model of a mechanism based on the considerations mentioned above. Then, a structural analysis method of the mechanism is developed where multiple closed loops sharing several joints are extracted so that they lie along the same direction. A data structure that facilitates the calculation of transformation matrices for the parts of the mechanism, is developed. Lastly, a nonlinear function minimization method is applied to the displacement analysis of closed-loop mechanisms.

### 2. Assumptions

In this research, the following four conditions are assumed.

- (1) All parts of the mechanism have rigid bodies.
- (2) At least one part is fixed.
- (3) All parts of the mechanism consist of either a planar, cylindrical, conical or spherical surface. The

contact state of any two parts involves any of the four types of surfaces.

- (4) There is no change in the contact state between parts during motion of the mechanism.

### 3. Generation of Joint Information from Geometric Model

There are several methods for deriving the allowable translational and rotational movements that maintain the contact states between two parts. Those derivations are mostly accomplished by integrating the contact states. However, since only simple contact states are considered by those methods, it is possible that allowable movements cannot be calculated correctly. We propose a new method for deriving possible movements by dealing with all contact states separately.

In our method, local coordinate systems are defined for each of the contact states shown in Fig. 1. Table 1 lists the constraints used to create the local coordinate systems for all contact states. Letters under column "type" correspond to those in Fig. 1. Symbols  $f_i$ ,  $a_i$ ,  $c_i$  denote the  $i$ th ( $i=1, 2$ ) surface itself, the central axis of the surface (if the surface is cylindrical or conical), and the center of the surface (if the surface is spherical), respectively. Symbols  $n_i$ ,  $\alpha_i$ ,  $v_i$  denote the normal vector of the surface (if the

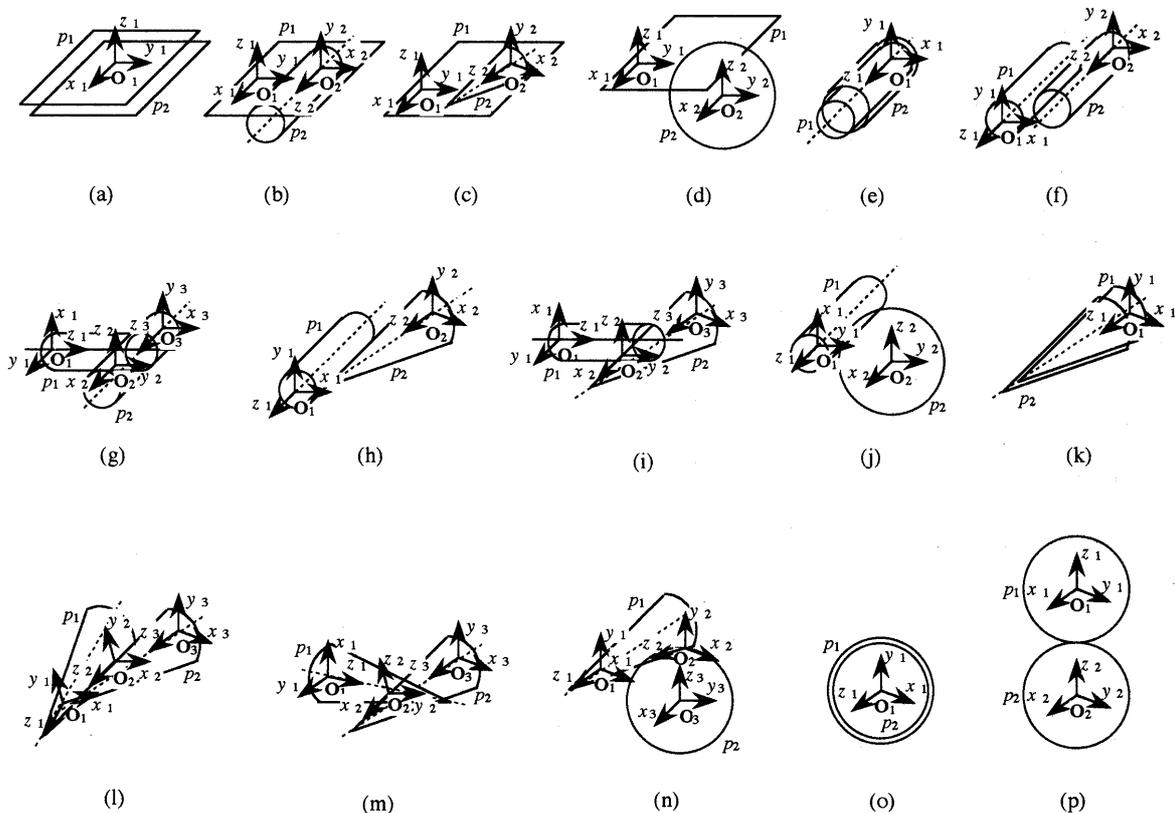


Fig. 1 Local coordinate systems defined for each contact state

Table 1 Constraints used to generate local coordinate systems

Type	Contact states			Constraints
	$f_1$	$f_2$	Contact region	
(a)	planar surf.	planar surf.		$z_1 // n_1, O_1$ on $f_1$
(b)	planar surf.	cylindrical surf.		$z_1 // n_1, O_1$ on $f_1, z_2 // a_2, O_2$ on $a_2$
(c)	planar surf.	conical surf.		$z_1 // n_1, O_1$ on $f_1, z_2 // a_2, O_2$ on $a_2$
(d)	planar surf.	spherical surf.		$z_1 // z_2 // n_1, O_1$ on $f_1, O_2$ on $c_2$
(e)	cylindrical surf.	cylindrical surf.	surface	$z_1 // a_1, O_1$ on $a_1$
(f)	cylindrical surf.	cylindrical surf.	line	$z_1 // a_1, O_1$ on $a_1, z_2 // a_2, O_2$ on $a_2$
(g)	cylindrical surf.	cylindrical surf.	point	$z_1 // a_1, O_1$ on $a_1, z_2 // m, O_2$ on $p, z_3 // a_2, O_3$ on $a_2$
(h)	cylindrical surf.	conical surf.	line	$z_1 // a_1, O_1$ on $a_1, z_2 // a_2, O_2$ on $a_2$
(i)	cylindrical surf.	conical surf.	point	$z_1 // a_1, O_1$ on $a_1, z_2 // m, O_2$ on $p, z_3 // a_2, O_3$ on $a_2$
(j)	cylindrical surf.	spherical surf.		$z_1 // a_1, O_1$ on $a_1, O_2$ on $c_2$
(k)	conical surf.	conical surf.	surface	$z_1 // a_1, O_1$ on $a_1$
(l)	conical surf.	conical surf.	line	$z_1 // a_1, O_1$ on $a_1, z_2 // l, O_2$ on $l, z_3 // a_2, O_3$ on $a_2$
(m)	conical surf.	conical surf.	point	$z_1 // a_1, O_1$ on $a_1, z_2 // m, O_2$ on $p, z_3 // a_2, O_3$ on $a_2$
(n)	conical surf.	spherical surf.		$z_1 // a_1, O_1$ on $a_1, z_2 // (v_1 - p), O_2$ on $p, O_3$ on $c_2$
(o)	spherical surf.	spherical surf.	surface	$O_1$ on $c_1$
(p)	spherical surf.	spherical surf.	point	$O_1$ on $c_1, O_2$ on $c_2, z_1 // z_2 // (O_1 - O_2)$

surface is planar), the directional vector of the axis of the surface (if the surface is cylindrical or conical), and the position vector of the vertex of the surface (if the surface is conical), respectively. Symbols  $m$  and  $n$ ,  $l$  and  $l$ , and  $p$  and  $p$ , denote the tangential plane to two contacting surfaces and its normal vector, the line of contact of two surfaces (if the contacting region becomes a line) and its direction vector, and the point of contact of two surfaces (if the contacting region becomes a point) and its position vector, respectively. Expression  $A // B$  means that vectors  $A$  and  $B$  are parallel, while  $A$  on  $B$  denotes that vector  $A$  is on  $B$ .

Table 2 shows the allowable movements of  $p_2$  relative to  $p_1$  for each of the contact states in Fig. 1, where symbol  $M_i$  denotes the matrix that makes the  $i$ th local coordinate system coincide with the world coordinate system. Symbols,  $T_z, T_\phi, T_{z\phi}, T_{xy\phi}, T_{\theta\phi}, T_{\theta\phi\phi}$  denote the matrices that represent the allowable translational and rotational movements as follows.

$$T_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_\phi = \begin{bmatrix} c\phi & -s\phi & 0 & 0 \\ s\phi & c\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{z\phi} = \begin{bmatrix} c\phi & -s\phi & 0 & 0 \\ s\phi & c\phi & 0 & 0 \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{xy\phi} = \begin{bmatrix} c\phi & -s\phi & 0 & x \\ s\phi & c\phi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

Table 2 Allowable movements for each contact state

Type	Allowable movements	Type	Allowable movements
(a)	$M_1^{-1}T_{xy\phi}M_1$	(i)	$M_1^{-1}T_\phi M_1 M_2^{-1}T_{xy\phi} M_2 M_3^{-1}T_\phi M_3$
(b)	$M_1^{-1}T_{xy\phi}M_1 M_2^{-1}T_\phi M_2$	(j)	$M_1^{-1}T_{z\phi}M_1 M_2^{-1}T_{\theta\phi\phi}M_2$
(c)	$M_1^{-1}T_{xy\phi}M_1 M_2^{-1}T_\phi M_2$	(k)	$M_1^{-1}T_\phi M_1$
(d)	$M_1^{-1}T_{xy\phi}M_1 M_2^{-1}T_{\theta\phi}M_2$	(l)	$M_1^{-1}T_\phi M_1 M_2^{-1}T_{z\phi} M_2 M_3^{-1}T_\phi M_3$
(e)	$M_1^{-1}T_{z\phi}M_1$	(m)	$M_1^{-1}T_\phi M_1 M_2^{-1}T_{xy\phi} M_2 M_3^{-1}T_\phi M_3$
(f)	$M_1^{-1}T_{z\phi}M_1 M_2^{-1}T_\phi M_2$	(n)	$M_1^{-1}T_\phi M_1 M_2^{-1}T_{z\phi} M_2 M_3^{-1}T_{\theta\phi\phi}M_3$
(g)	$M_1^{-1}T_\phi M_1 M_2^{-1}T_{xy\phi} M_2 M_3^{-1}T_\phi M_3$	(o)	$M_1^{-1}T_{\theta\phi\phi}M_1$
(h)	$M_1^{-1}T_{z\phi}M_1 M_2^{-1}T_\phi M_2$	(p)	$M_1^{-1}T_{\theta\phi}M_1 M_2^{-1}T_{\theta\phi\phi}M_2$

$$T_{\theta\phi} = \begin{bmatrix} c\phi & s\phi s\theta & s\phi c\theta & 0 \\ 0 & c\theta & -s\theta & 0 \\ -s\phi & c\phi s\theta & c\phi c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{\theta\phi\phi} = \begin{bmatrix} c\phi c\phi & c\phi s\phi s\theta - s\phi c\theta & c\phi s\phi c\theta + s\phi c\theta & 0 \\ s\phi c\phi & s\phi s\phi s\theta + c\phi c\theta & s\phi s\phi c\theta - c\phi s\theta & 0 \\ -s\phi & c\phi s\theta & c\phi c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

where  $x, y, z$  denote the translational displacement along, and  $\theta, \phi, \phi$  denote the rotational displacement about the three axes of the local coordinate system, and  $s = \sin$  and  $c = \cos$ .

The case that several contact states exist between two parts is dealt with as closed loops. For example, if there are two contact states between two parts, planar contact and cylindrical contact, as shown in Fig. 2, relative movements between these two parts must satisfy the following constraints:

$$J_1 J_2 = I, \quad J_1 = M_1^{-1} T_{xy\phi} M_1, \quad J_2 = M_2^{-1} T_{z\phi} M_2, \quad (1)$$

where  $I$  denotes the  $4 \times 4$  identity matrix, and  $J_1$  and  $J_2$  correspond to (a) and (e) in Table 2. In this way,

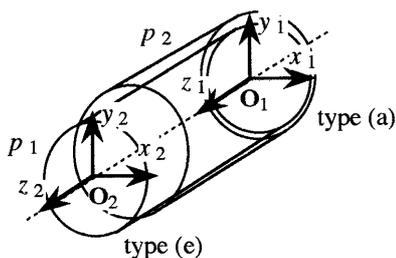


Fig. 2 Multiple contact states between two parts

our method can handle the case in which multiple contact states exist between two parts without the need for integrating the contact states.

4. Structural Analysis of Mechanisms

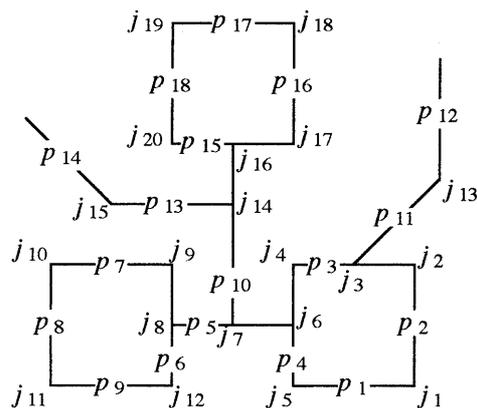
4.1 Extraction of closed loop considering its direction

Kuroiwa et al. have developed a method for analyzing structures of mechanisms by using the reachability matrix<sup>(4)</sup>. In their method, however, the directions of closed loops that share some common joints are not considered. If there exist several closed loops that share common joints, they must be extracted so that they lie along the same direction. Since the allowable movements between two parts are relative, a fixed part and a moving part of a common joint must coincide in multiple loops. In the following, a new method for analyzing the structure of mechanisms considering the directions of closed loops is presented.

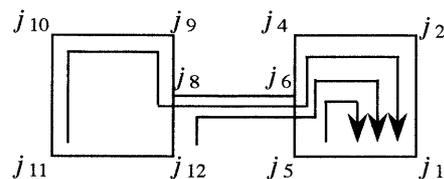
We classify the elements of the mechanism as follows.

- (1) Single closed loop
- (2) Multiple closed loop
- (3) Part not included in any loop

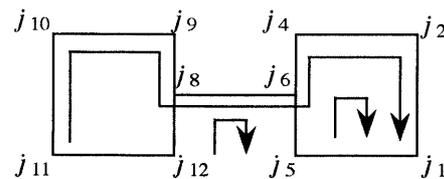
A single closed loop has no joint that is included in any other loops, while a multiple closed loop means several closed loops that share some joints. An incidence matrix<sup>(7)</sup> and a cycle matrix<sup>(7)</sup> are used to extract the single closed loop and the multiple closed loop. The model shown in Fig. 3(a), where  $p_i$  and  $j_k$  denote a part and a joint respectively, is used to explain the closed-loop extraction method. We assume that parts  $p_1$  and  $p_9$  are fixed. All fixed parts are recognized as one part. The incidence matrix for the model shown in Fig. 3(a) is represented by Eq. (2), where each row and each column denote part and joint, respectively.



(a)



(b)



(c)

Fig. 3 An example of mechanism

$$S = \begin{matrix} & j & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \begin{matrix} p \\ 1 \& 9 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \\ 17 \\ 18 \end{matrix} & \left[ \begin{matrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \right. \end{matrix}$$

$$\begin{matrix}
 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1
 \end{matrix} \quad (2)$$

For example, parts  $p_2$  and  $p_3$  are connected by  $j_2$  in Fig. 3(a) so that corresponding elements in the matrix become 1 and -1, respectively. In this example, the upper and lower rows are set as 1 and -1, respectively, for all columns. The cycle matrix can be obtained by the following calculation.

$$L = [-S_2^T(S_1^T)^{-1}I], \quad (3)$$

where matrix  $S_1$  is a full-rank sub matrix of a reduced matrix<sup>(7)</sup> of the incidence matrix.  $S_2$  is the remaining matrix of the reduced matrix. For example, in Eq. (2), the reduced matrix can be constructed by eliminating the first row, then the full-rank sub matrix can be constructed from the columns except for the 5th, 11th, 12th, and 20th rows. Accordingly,  $S_1$  and  $S_2$  are as shown in Eq. (4).

$$\begin{matrix}
 p & j & 1 & 2 & 3 & 4 & 6 & 7 & 8 & 9 & 10 \\
 2 & \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 3 & \begin{bmatrix} 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 4 & \begin{bmatrix} 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 5 & \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 \\
 6 & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\
 7 & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\
 8 & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\
 S_1=10 & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
 11 & \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 12 & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 13 & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 14 & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 15 & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 16 & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 17 & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 18 & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
 \end{bmatrix}
 \end{matrix}$$

$$\begin{matrix}
 13 & 14 & 15 & 16 & 17 & 18 & 19 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & -1 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & -1 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & -1 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & -1
 \end{matrix}$$

$$\begin{matrix}
 p & j & 5 & 11 & 12 & 20 \\
 2 & \begin{bmatrix} 0 & 0 & 0 & 0 \\
 3 & \begin{bmatrix} 0 & 0 & 0 & 0 \\
 4 & \begin{bmatrix} -1 & 0 & 0 & 0 \\
 5 & \begin{bmatrix} 0 & 0 & 0 & 0 \\
 6 & \begin{bmatrix} 0 & 0 & -1 & 0 \\
 7 & \begin{bmatrix} 0 & 0 & 0 & 0 \\
 8 & \begin{bmatrix} 0 & -1 & 0 & 0 \\
 S_2=10 & \begin{bmatrix} 0 & 0 & 0 & 0 \\
 11 & \begin{bmatrix} 0 & 0 & 0 & 0 \\
 12 & \begin{bmatrix} 0 & 0 & 0 & 0 \\
 13 & \begin{bmatrix} 0 & 0 & 0 & 0 \\
 14 & \begin{bmatrix} 0 & 0 & 0 & 0 \\
 15 & \begin{bmatrix} 0 & 0 & 0 & 1 \\
 16 & \begin{bmatrix} 0 & 0 & 0 & 0 \\
 17 & \begin{bmatrix} 0 & 0 & 0 & 0 \\
 18 & \begin{bmatrix} 0 & 0 & 0 & -1
 \end{bmatrix}
 \end{matrix}$$

A cycle matrix is obtained as shown in Eq. (5).

$$\begin{matrix}
 l & j & 1 & 2 & 3 & 4 & 6 & 7 & 8 & 9 & 10 & 13 & 14 \\
 1 & \begin{bmatrix} -1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 L=2 & \begin{bmatrix} -1 & -1 & 0 & -1 & -1 & 0 & -1 & -1 & -1 & 0 & 0 \\
 3 & \begin{bmatrix} -1 & -1 & 0 & -1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\
 4 & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 15 & 16 & 17 & 18 & 19 & 5 & 11 & 12 & 20 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 1
 \end{bmatrix}
 \end{matrix}$$

Each row and each column of the loop matrix represent, respectively, a loop and a joint of the mechanism. In Eq. (5), the first to third rows represent a multiple closed loop. Joints  $j_1$ ,  $j_2$  and  $j_4$  appear in the three loops, and  $j_6$  and  $j_8$  exist in both second and third rows. The fourth row represents a single closed loop because none of the joints in this row are included in the other rows.

The directions of the multiple closed loop must be arranged if some of the loops are not in the same direction. To accomplish this, a new matrix as shown in Eq. (6) is constructed for each multiple closed loop.

$$L_1 = \begin{matrix} & l & j & 1 & 2 & 4 & 6 & 8 & 9 & 10 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 5 \\ 11 \\ 12 \end{matrix} & \begin{bmatrix} -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{matrix} \quad (6)$$

This matrix represents the three loops shown in Fig. 3(b). If all non-zero row elements have the same sign for all columns, then all loops in the multiple closed loop lie along the same direction. For the case in Eq. (6), we need not change the directions. The following operations can be made to make the directions of loops coincide with each other.

(a) Change the signs of all elements in some rows of the matrix representing the multiple closed loop.

(b) Change the combination of some closed loops of the multiple closed loop.

The multiple closed loop shown in Fig. 3(b) can also be represented by an other combination of three loops as shown in Fig. 3(c). This is represented by the following matrix.

$$L_2 = \begin{matrix} & l & j & 1 & 2 & 4 & 6 & 8 & 9 & 10 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 5 \\ 11 \\ 12 \end{matrix} & \begin{bmatrix} -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \end{matrix} \quad (7)$$

Operation (a) means that the directions of some loops are turned to the other way. It is notable that the directions of all loops in Eq. (7) cannot be made the same by only operation (a). In this case, we must employ operation (b). Since the combination of the closed loops depends on the selection of  $S_1$  from the incidence matrix, we can make other combinations by changing  $S_1$ . This operation can be performed by the following calculation.

$$L_j = L_s^{-1} L_i, \quad (8)$$

where  $L_s$  denotes the full-rank sub matrix of a loop matrix  $L_i$ , and  $L_j$  means a loop matrix representing a new combination of the multiple closed loop. For example, Eq. (6) can be obtained from Eq. (8) by creating  $L_s$  from the last three columns of Eq. (7) as follows.

$$L_s = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}. \quad (9)$$

Using operations (a) and (b), multiple closed loops can be recognized by considering their directions. The proposed closed-loop extraction method can be summarized as follows.

(step 1) Make a loop matrix and extract all

multiple closed loops. Then, generate different matrices for each multiple closed loop. If no multiple closed loop exists, then exit. Otherwise, let  $L_i$  ( $i=1, \dots, n$ ) be each matrix and  $i \leftarrow 1$ . Go to (step 2).

(step 2) If  $i > n$ , then exit. Otherwise, go to (step 3).

(step 3) Decide whether all non-zero elements have the same sign for all columns. If yes, then  $i \leftarrow i + 1$  and go to (step 2). Otherwise, go to (step 4).

(step 4) Find the subset of rows for which all non-zero elements are the same by applying operation (a). If such subset is found, then  $i \leftarrow i + 1$  and go to (step 2). Otherwise, go to (step 5).

(step 5) Change the combination of loops constructing  $L_i$  by applying operation (b), then go to (step 4).

By this procedure, it is guaranteed that the directions of multiple closed loops become the same.

#### 4.2 Data structure of mechanism for displacement analysis

We propose the data structure of the mechanism that forms a tree in order to analyze the displacements simply. Figure 4 is a data structure of the mechanism shown in Fig. 3(a). Circles represent the single closed loop, the multiple closed loop or a part that is not included in any closed loop. Those circles are called structure element nodes. For example, node  $N_1$  represents a multiple closed loop that consists of three loops  $l_1-l_3$ , and node  $N_5$ , a single closed loop

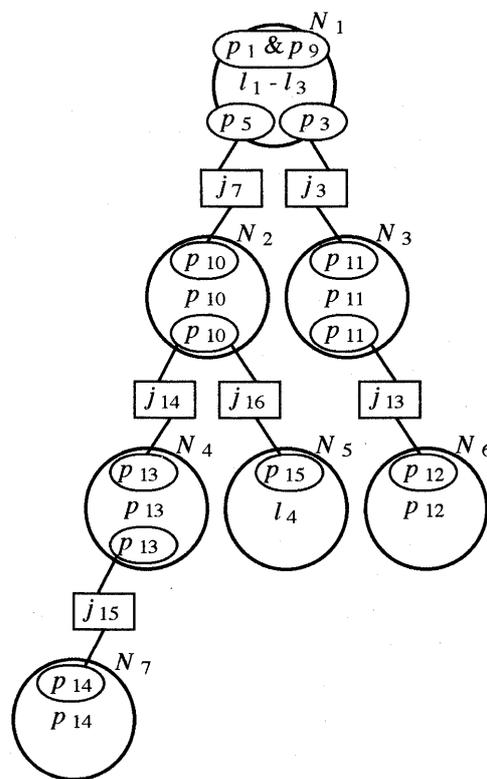


Fig. 4 Data for transformation matrix generation

Table 3 Loop data

Loop	Elements and thier order
$l_1$	$(p_1 \& p_9) - j_5 - p_4 - j_4 - p_3 - j_2 - p_2 - j_1$
$l_2$	$(p_1 \& p_9) - j_{11} - p_8 - j_{10} - p_7 - j_9 - p_6 - j_8 - p_5 - j_6 - p_4 - j_4 - p_3 - j_2 - p_2 - j_1$
$l_3$	$(p_1 \& p_9) - j_{12} - p_6 - j_8 - p_5 - j_6 - p_4 - j_4 - p_3 - j_2 - p_2 - j_1$
$l_4$	$p_{15} - j_{20} - p_{18} - j_{19} - p_{17} - j_{18} - p_{16} - j_{17}$

that consists of only one loop  $l_4$ . The other structure element nodes represent all parts that are not included in any closed loop. A joint adjacent to two different structure element nodes is represented by a block, and is called a joint node. Long circles included in each structure element node represent the part adjacent to the joint nodes. A long circle node shown in the upper portion of each structure element node is called an upper adjacent node, while that shown in the lower portion is called a lower adjacent node. The upper and lower adjacent nodes denote parts that are adjacent to those represented by their upper and lower nodes, respectively. For example, parts  $p_3$  and  $p_5$  in Fig. 3(a), the lower adjacent nodes of  $N_1$  in Fig. 4, are adjacent to parts  $p_{10}$  and  $p_{11}$ , the upper adjacent nodes of  $N_2$  and  $N_3$ , respectively. If the structure element node denotes a part, then the upper and lower adjacent nodes denote the same part. The root node of the tree structure denotes parts corresponding to the fixed parts, or the single closed loop or the multiple closed loop including the fixed parts. The upper adjacent node of the root node corresponds to the fixed parts. In each structure element node, the number of upper adjacent nodes is one if we regard the fixed parts as one part, whereas the number of lower adjacent nodes of all leaf nodes of the tree is zero.

The structure element node for the single closed loop or the multiple closed loop has pointers to other data called loop data as shown in Table 3, where  $l_1 - l_4$  correspond to those in Fig. 4. In each loop, parts and joints are connected alternately according to the loop direction. The first part of the loop data corresponds to the upper adjacent node of each structure element node. For example, the upper adjacent node of structure element node  $N_5$  in Fig. 4 corresponds to the first part of loop  $l_4$  in Table 3. The first loop of the structure element node must correspond to any loop that includes the upper adjacent node. For node  $N_1$ , all loops include the upper adjacent nodes,  $p_1$  and  $p_9$ ; therefore  $l_1$  is adopted as the first loop. The second loop must include at least one part of the first loop, and that part must be the first part of the second loop. Similarly the third loop must include at least one part that is in either the first or the second loop, and that part must be the first part of the third loop. The loop data of the multiple closed loop are generated accord-

ing to this procedure. Using this data structure, transformation matrices of all parts can be readily calculated.

## 5. Displacement Analysis of Mechanisms

### 5.1 Displacement calculation of joints of closed-loop mechanisms by nonlinear function minimization method

A nonlinear function minimization method is applied to the displacement calculation of the joints of the closed-loop mechanism. The movement of one multiple closed loop consisting of  $q$  loops is constrained by the following equation.

$$\mathbf{J}_1^h \cdots \mathbf{J}_{g(h)}^h = \mathbf{I}, \quad h=1, \dots, q, \quad (10)$$

where  $\mathbf{J}_i^h$  denotes the matrix representing the possible movements of the  $i$ th joint of the  $h$ th loop of the multiple closed loop. The symbol  $g(h)$  is the number of joints included in the  $h$ th loop. Elements  $e_{ij}$  on the left except for the fourth row in Eq.(10) must meet the following twelve equations from the right in Eq.(10).

$$\begin{cases} e_{ij}^h - 1 = 0, & \text{if } i=j \\ e_{ij}^h = 0, & \text{otherwise} \end{cases}, \quad i=1, \dots, 3, j=1, \dots, 4. \quad (11)$$

Equation (11) can be solved as the following nonlinear function minimization problem.

$$\min \sum_{h=1}^q \sum_{k=1}^{12} (f_k^h)^2, \quad (12)$$

where  $f_k^h$  denotes the  $k$ th equation of the twelve equations shown in Eq.(11). The solution of Eq.(11) exists if and only if the following condition is satisfied after minimizing Eq.(12).

$$\sum_{h=1}^q \sum_{k=1}^{12} (f_k^h)^2 = 0. \quad (13)$$

In the current implementation, we use the Davidon-Fletcher-Powell method<sup>(8)</sup> to solve Eq.(12).

### 5.2 Generation of transformation matrices

We explain how the transformation matrices for all parts of the mechanism are generated from the data structure mentioned in Section 4.2. The same mechanism shown in Fig. 3(a) and its data structure shown in Fig. 4 and Table 3, are employed to elucidate our method.

First, root node of the data structure is considered. If root node represents a single closed loop or a multiple closed loop, displacements of all joints included in the loop are calculated by the method described in Section 4.1. After the calculation, the transformation matrices for moving all parts included in the loop are generated. In this case, the transformation matrices for moving parts included in  $l_1$  are made according to the sequence of the first row in Table 1 as follows.

$$\mathbf{P}_{1\&9} = \mathbf{I}, \quad \mathbf{P}_4 = \mathbf{P}_{1\&9} \mathbf{J}_5, \quad \mathbf{P}_3 = \mathbf{P}_4 \mathbf{J}_4, \quad \mathbf{P}_2 = \mathbf{P}_3 \mathbf{J}_2,$$

where  $\mathbf{P}_i$  denotes the transformation matrix for

moving part  $p_i$ . Similarly, the transformation matrices for moving parts included in  $l_2$  can be obtained according to the sequence of the second row in Table 3.

$$P_8 = P_{1\&9}J_{11}, \quad P_7 = P_8J_{10}, \quad P_6 = P_7J_9, \quad P_5 = P_6J_8.$$

Then, loop  $l_3$  is considered. However, the transformation matrices for all parts included in loop  $l_3$  have already been calculated, so all calculations necessary for node  $N_1$  are done. Next, nodes  $N_2$  and  $N_3$  are focused on. The transformation matrices for parts of the upper adjacent nodes of  $N_2$  and  $N_3$  are calculated using the matrices for the part of the lower adjacent node of  $N_1$  and its adjacent joint nodes,

$$P_{10} = P_5J_7, \quad P_{11} = P_3J_3.$$

Since nodes  $N_2$  and  $N_3$  denote the parts, we shift to the lower nodes  $N_4, N_5$  and  $N_6$ . The transformation matrices for these nodes can be generated using those for the lower adjacent nodes of  $N_2$  and  $N_3$ , and their adjacent joint nodes.

$$P_{13} = P_{10}J_{14}, \quad P_{15} = P_{10}J_{16}, \quad P_{12} = P_{11}J_{13}.$$

Since node  $N_5$  represents a single closed loop, we must calculate the transformation matrices for parts included in it. If a displacement is given to any of the joints included in this single closed loop, the displacements for other joints of this loop must be calculated by the method explained in Section 5.1. Then, the transformation matrices are calculated according to the sequence of the fourth row in Table 3 as follows.

$$P_{18} = P_{15}J_{20}, \quad P_{17} = P_{18}J_{19}, \quad P_{16} = P_{17}J_{18}.$$

The transformation matrix for the upper adjacent node of  $N_7$  is generated in a similar manner.

$$P_{14} = P_{13}J_{15}.$$

In this way, we can obtain the transformation matrices of all parts of the given mechanism using the

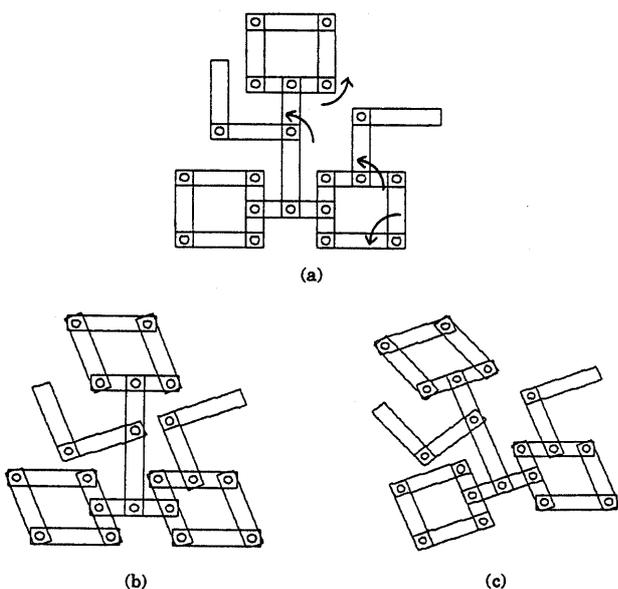


Fig. 5 Example 1

data structure as shown in Fig. 4 and Table 3.

### 6. Examples

The mechanism shown in Fig. 3(a) is modeled using a geometric method as shown in Fig. 5 on the computer. The proposed method is applied to this model. For simplicity, all joints of the mechanism are regarded as those of the cylindrical contacts shown in Fig. 1(e). We assume that parts  $p_1$  and  $p_9$  are fixed parts. Joint information is automatically generated from the geometric model by the method described in Section 2. The data structure shown in Fig. 4 is obtained by the method explained in Section 3. By inputting the displacements as shown by the arrows in Fig. 5(a) to joints  $j_1, j_3, j_{14}, j_{17}$  such that they rotate about each cylindrical axis, we can obtain the motion of the mechanism as shown in Fig. 5(b). Next, we designate only part  $p_1$  as the fixed part and input the same displacements; then, the mechanism is moved as shown in Fig. 5(c).

Fig. 6(a) is an example of a mechanism that has a contact state between planar and cylindrical surfaces. Figure 6(b) is its disassembled state. Table 4 shows all relations among parts, joints and contact states of the mechanism. From this table, we can see that there are three independent loops that share some common joints. Part  $p_3$  is designated as the fixed part. Loop data are generated as shown in Table 5 where all loops are in the same direction. By rotating part  $p_1$  about the cylindrical axis, as shown by an arrow in

Table 4 Structure of mechanism in Fig. 6

Joint No.	Part No.		Type of contact
	1	2	
1	1	2	(b)
2	1	3	(a)
3	1	3	(e)
4	2	3	(a)
5	2	3	(e)

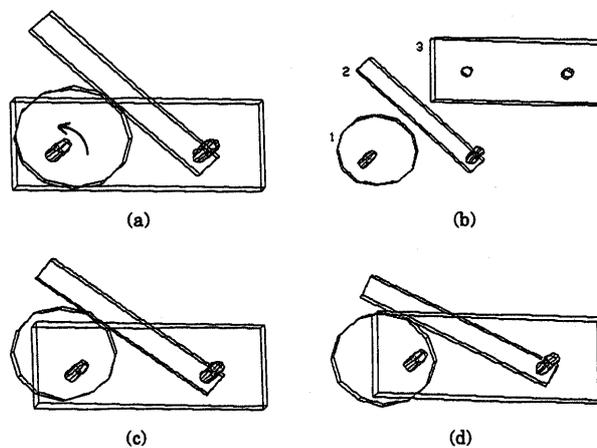


Fig. 6 Example 2

Table 5 Recognition result of loops for mechanism in Fig. 6

No.	Elements and their order
1	$p_3 - j_2 - p_1 - j_1 - p_2 - j_4 - p_3$
2	$p_1 - j_1 - p_2 - j_5 - p_3 - j_2 - p_1$
3	$p_3 - j_2 - p_1 - j_3 - p_3$

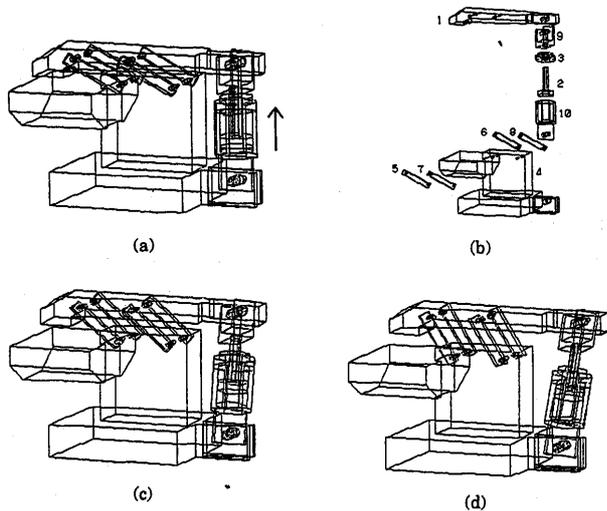


Fig. 7 Example 3

Fig. 6 (a), namely, by giving a displacement to joint  $j_3$ , we can obtain the movements shown in Figs. 6(c) and (d).

The final example shown in Fig. 7(a) is a very complex mechanism that consists of an air cylinder and a parallel linkage. Its disassembled state is shown in Fig. 7(b), and the relationship among parts, joints and contact states are shown in Table 6. There are seven independent loops sharing common joints in the mechanism, and they are recognized as shown in Table 7. All loops are in the same direction. Part  $p_4$  is designated as the fixed part. By translating part  $p_2$  along its cylindrical axis as shown by an arrow in Fig. 7(a), namely, by giving a displacement to joint  $j_3$ , the mechanism behaves as shown in Figs. 7(c) and (d).

7. Conclusions

In this paper, a new displacement analysis method for an arbitrary mechanism has been proposed. First, we have presented an automatic generation method of joint information from a geometric description of the mechanism. This method can handle a very complex joint consisting of multiple contact states including higher pairs. Then, a structure analysis method has been developed considering the direction of multiple closed loops that share some common joints. A data structure has also been proposed to easily generate the transformation matrices for all parts of the mechanism. Moreover, a nonlinear function minim-

Table 6 Structure of mechanism in Fig. 7

Joint No.	Part No.	Type of contact	Joint No.	Part No.	Type of contact
1	1	5	9	2	10
2	1	6	10	3	10
3	1	7	11	4	5
4	1	8	12	4	6
5	1	9	13	4	7
6	2	3	14	4	8
7	2	9	15	4	10
8	2	9	16	4	10

Table 7 Recognition result of loops for mechanism in Fig. 7

No.	Elements and their order
1	$p_4 - j_{12} - p_6 - j_2 - p_1 - j_1 - p_5 - j_{11} - p_4$
2	$p_1 - j_1 - p_5 - j_{11} - p_4 - j_{13} - p_7 - j_3 - p_1$
3	$p_1 - j_1 - p_5 - j_{11} - p_4 - j_{14} - p_8 - j_4 - p_1$
4	$p_1 - j_1 - p_5 - j_{11} - p_4 - j_{15} - p_{10} - j_9 - p_2 - j_7 - p_9 - j_5 - p_1$
5	$p_1 - j_1 - p_5 - j_{11} - p_4 - j_{16} - p_{10} - j_9 - p_2 - j_7 - p_9 - j_5 - p_1$
6	$p_{10} - j_9 - p_2 - j_6 - p_3 - j_{10} - p_{10}$
7	$p_2 - j_7 - p_9 - j_8 - p_2$

ization method has been applied to the displacement calculation of the closed loops. Finally, the effectiveness of our method has been demonstrated through several examples.

References

- (1) Sheth, P.N. and Uicker, J.J. Jr., A Generalized Symbolic Notation for Mechanisms, Trans., ASME, Ser. B, Vol. 94, No. 2 (1971), p. 102.
- (2) Sheth, P.N. and Uicker, J.J. Jr., IMP (Integrated Mechanisms Program), A Computer-Aided Design Analysis System for Mechanisms and Linkage, Trans. ASME, Ser. B., Vol. 94, No. 2 (1972), p. 454.
- (3) Nagamatsu, A., Saito, T. and Ichiyangi, T., Research on Simulation of the Robot (1st Report, Numerical Analysis of Displacement, Velocity and Acceleration), Trans. Jpn. Soc. Mech. Eng., (in Japanese), Vol. 51, No. 470, C (1985), p. 2668.
- (4) Kuroiwa, T., Motoe, A. and Hiraoka, K., Simulation System of Design for Link Mechanism (1st Report, Topological Analysis for Mechanism), Trans. Jpn. Soc. Mech. Eng., (in Japanese), Vol. 52, No. 484, C (1986), p. 3214.
- (5) Arai, E. and Iwata, K., Development of Integrated Product Model for CIM, Manufacturing Systems, Vol. 16, No. 3, (1987), p. 187.
- (6) Kim, S.H. and Lee, K., An Assembly Modelling System for Dynamic and Kinematic Analysis, Comput. Aided Des., Vol. 21, No. 1, (1989), p. 2.
- (7) Harary, F., Graph Theory, Addison Wesley, (1969).
- (8) Fletcher, R. and Powell, M. J. P., A Rapidly Convergent Descent Method for Minimization, Comp. J., Vol. 6, (1963), p. 163.