Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 1761968, 13 pages
http://dx.doi.org/10.1155/2016/1761968

Research Article

Recognition of Kinematic Joints of 3D Assembly Models Based on Reciprocal Screw Theory

Tao Xiong,[1] Liping Chen,[1] Jianwan Ding,[1] Yizhong Wu,[1] and Wenjie Hou[2]

[1] *National Engineering Research Center of CAD Software Information Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China*
[2] *Eaton (China) Investments Co. Ltd., Shanghai 200335, China*

Correspondence should be addressed to Jianwan Ding; dingjw@hust.edu.cn

Received 17 February 2016; Accepted 17 March 2016

Academic Editor: Chaudry Masood Khalique

Reciprocal screw theory is used to recognize the kinematic joints of assemblies restricted by arbitrary combinations of geometry constraints. Kinematic analysis is common for reaching a satisfactory design. If a machine is large and the incidence of redesign frequent is high, then it becomes imperative to have fast analysis-redesign-reanalysis cycles. This work addresses this problem by providing recognition technology for converting a 3D assembly model into a kinematic joint model, which is represented by a graph of parts with kinematic joints among them. The three basic components of the geometric constraints are described in terms of wrench, and it is thus easy to model each common assembly constraint. At the same time, several different types of kinematic joints in practice are presented in terms of twist. For the reciprocal product of a twist and wrench, which is equal to zero, the geometry constraints can be converted into the corresponding kinematic joints as a result. To eliminate completely the redundant components of different geometry constraints that act upon the same part, the specific operation of a matrix space is applied. This ability is useful in supporting the kinematic design of properly constrained assemblies in CAD systems.

1. Introduction

Currently, we can design a machine in terms of a top-down approach or a bottom-up approach in a CAD system. All parts of an assembly model are positioned using various geometry constraints so that they are in the correct relative positions and orientations. The 3D assembly models record the design parameters and some information that can be used for concept-level and detailed-level design. Its benefits are usually recognized only by those who design precision machines [1–3].

Any machine that consists of moving parts has to be designed to properly perform its kinematic functions. To analyze the kinematic property of assemblies, because of the lack of a kinematic description, many researchers have used screw theory. Applying screw theory to kinematics was first performed by Ball [4]. Hunt [5] identified 22 cases of Ball's reciprocal screw systems. References [6–8] applied screw theory to determine the degrees of freedom of any body in a mechanism. Eddie Baker [9] analyzed many types of complex mechanisms based on [6]. Mason and Salisbury [10] used screw theory to characterize the nature of contacts between robot gripper hands and objects. To analyze robotic multiple peg-in-hole, Fei and Zhao [11] described the contact forces by using the screw theory in three dimensions. Lee and Saitou [12] determined the dimensional integrity of robustness based on screw theory. Kim et al. [13] established a framework to estimate the reaction of constraints about a knee.

Assembly features encode geometric and functional relationships between parts that are widely used. Thus, many researchers perform kinematic analysis by using the assembly features. Konkar and Cutkosky [14] created screw system representations of assembly mating features. Adams et al. [15–17] used Konkar's algorithm and extended their work by defining extensible screw representations of some types of assembly features. Gerbino et al. [18, 19] proposed an algorithm to decompose the liaison diagram representing an assembly into simple paths, to which the application of the twist union or intersection algorithms may be easily applied. They also highlighted the complexity behind the analysis of overconstraints with respect to the motion analysis. Celentano [20] improved the algorithm presented in [18] and developed a tool to analyze the motions and constraints of CAD mechanisms. Su et al. [21] presented a theory-based approach for the analysis and synthesis of flexible joints. Dixon and Shah [22] provided a mechanism for user-defined custom assembly features by using an interactive system. After that, twist and wrench matrices could be extracted from face pair relations and used in kinematic and structural analysis.

Geometry constraints between two parts in an assembly may be arbitrary combinations. References [23–25] analyzed the state of a constraint of an assembly, mechanism, or fixture to see if it is properly constrained or whether it contains unwanted overconstraints. Rusli et al. [26, 27] addressed attachment-level design, in which decisions are made to establish the types, locations, and orientations of assembly features. Su et al. [28] analyzed the mobility of overconstrained linkages and compliant mechanisms. To provide a desired constrained motion for synthesizing a pattern of flexures, a new screw theory that addresses "line screws" and "line screw systems" was presented by Su and Tari [29].

According to [4–29], if the screw representations between any two parts of the assembly model are known, we can easily analyze the model's kinematic functions. References [4–13, 23–29] manually model the corresponding screw system. Thus, it is time consuming when the machine is large and complex. On the other hand, to determine the screw system by using the methods proposed in [14–22], all of the screw representations of different features that may be used first have to be modeled. This is awkward and resembles geometric reasoning [30]. If the incidence of redesign frequency is high, the screw system has to be rebuilt again and again, which has a great influence on the efficiency of the analysis procedure.

We hope to automatically and easily obtain the screw system of any assembly model. Thus, when we modify the assembly model during analysis-redesign-reanalysis cycles, the kinematic functions can be directly validated. For convenience, we adopt a graph of parts with twist among them to express the screw system. A screw system can be easily used for kinematic analysis, such as determining the degrees of freedom, estimating the reaction of constraints, and characterizing the nature of contacts between any two components.

In this paper, we propose a new method to automatically convert a 3D assembly model into a screw system based on reciprocal screw theory. The two fundamental concepts in the constraint-based approach, that is, freedom and constraint, can be represented mathematically by a twist and a wrench. Both the twist and kinematic joint represent motion; both the wrench and geometry constraint represent constraint. When a twist is given, we can obtain the corresponding wrench as the reciprocal product of a twist and wrench, which is equal to zero. Conversely, if we know a wrench, then the corresponding twist could be deduced as well. To avoid geometric reasoning, we model the three basic components of geometry constraints in terms of wrench. Thus, all common assembly constraints can be automatically deduced. For the redundant property of geometry constraints, the specific operation of a matrix space is applied. Finally, we achieve the graph of parts with kinematic joints among them with respect to an assembly model.

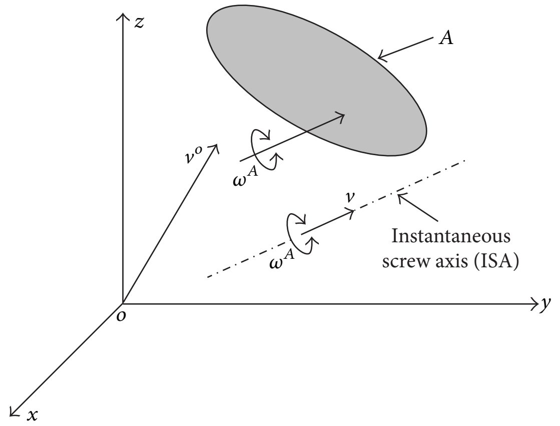


Figure 1: Twist.

1.1. Organization of the Paper. Section 2 reviews the twist representations of common kinematic joints. Section 3 analyzes the basic components of geometry constraints and models these components in terms of wrench. Section 4 shows how to recognize the corresponding kinematic joints of assemblies restricted by arbitrary combinations of geometry constraints. Section 5 contains several examples. Section 6 presents our conclusions.

2. Screw Models of Kinematic Joints

To recognize the corresponding kinematic joints from geometry constraints, a mathematical model must be developed that can sufficiently and simultaneously describe the properties of kinematic joints and geometry constraints. This section shows how to define kinematic joints using twist.

2.1. Twist. A twist [32, 33] takes the form

$$\mathbf{T} = \left[\omega_x \omega_y \omega_z;\ v_x v_y v_z\right]. \tag{1}$$

A twist is a screw that describes the instantaneous motion, to the first order, of a rigid body. This means that the motion described by a twist is allowed. The first triplet represents the angular velocity of the body with respect to a global reference frame. The second triplet represents the velocity, in the global reference frame, of that point on the body or its extension, which is instantaneously located at the origin of the global frame. Therefore, the unique line associated with the first triplet is the axis of rotation, and the unique point associated with the second triplet is the point on the body or its extension, which is, at that instant, located at the origin of the global reference frame. See Figure 1.

The ISA in Figure 1 is the axis introduced in Chasles Theorem. Both the twist and kinematic joint represent motion.

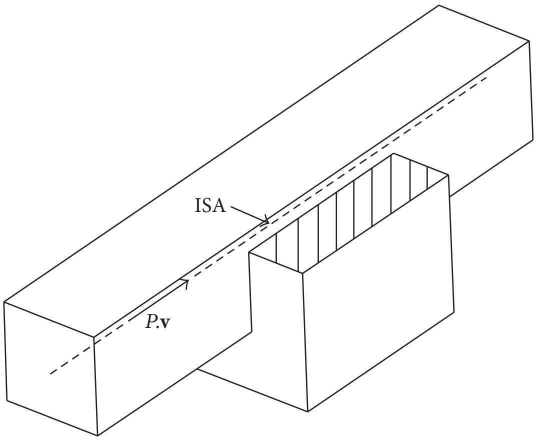


Figure 2: Prismatic joint.

Therefore, we can use a twist to characterize a component of the kinematic joint between two parts. Afterwards, the kinematic joint can be transformed into a corresponding twist matrix.

2.2. Twist Representations of Kinematic Joints. There are 6 types of kinematic joints in common: *prismatic joint*, *revolute joint*, *helical joint*, *cylindrical joint*, *spherical joint*, and *planar joint*. In this section, we use **s** to characterize the allowed motions of each kinematic joint and show their corresponding twist matrices.

2.2.1. Prismatic Joint. The *degrees of freedom* (dof) of a *prismatic joint* (P) is one. We use the unit vector $P.\mathbf{v} = (v_x, v_y, v_z)$ to characterize the allowed translation. See Figure 2.

Thus, $P.\mathbf{s}$ has the form

$$P.\mathbf{s} = P.\mathbf{v}. \tag{2}$$

The twist matrix representation of P is given by

$$\mathbf{T}^P = [\mathbf{0};\ \ P.\mathbf{s}]. \tag{3}$$

We use $\mathbf{0}$ instead of $[0, 0, 0]$ unless otherwise stated.

2.2.2. Revolute Joint. The dof of a *revolute joint* (R) is one. We use $R.\mathbf{L}$ to characterize the axis of R. The unit vector of $R.\mathbf{L}$ is characterized by $R.\mathbf{v} = (v_x, v_y, v_z)$. $R.\mathbf{P} = (P_x, P_y, P_z)$ is a single point on the $R.\mathbf{L}$. See Figure 3.

Thus, $R.\mathbf{L}$ has the form

$$R.\mathbf{L} = R.\mathbf{P} + a \cdot R.\mathbf{v}, \tag{4}$$

where a is a real number. $R.\mathbf{s}$ has the form

$$R.\mathbf{s} = R.\mathbf{P} \times R.\mathbf{v}. \tag{5}$$

Then, the twist matrix representation of R is given by

$$\mathbf{T}^R = [R.\mathbf{v};\ \ R.\mathbf{s}]. \tag{6}$$

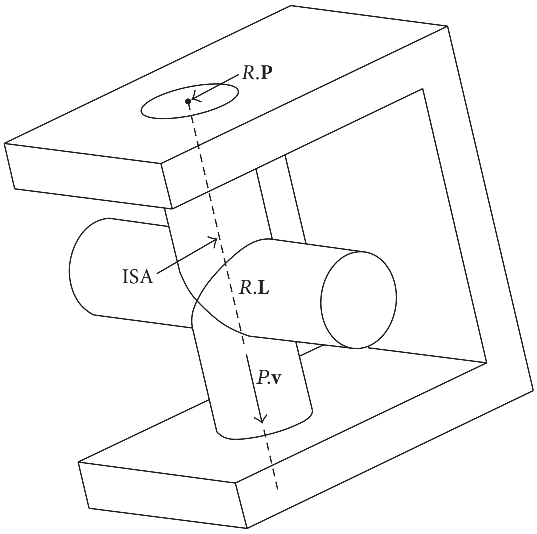


Figure 3: Revolute joint.

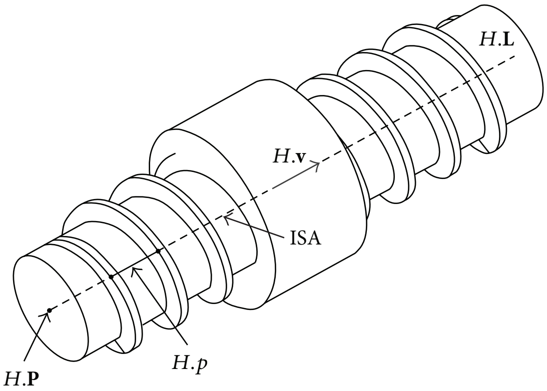


Figure 4: Helical joint.

2.2.3. Helical Joint. The dof of a *helical joint* (H) is one. We use $H.\mathbf{L}$ to characterize the axis of H. The unit vector of $H.\mathbf{L}$ is characterized by $H.\mathbf{v} = (v_x, v_y, v_z)$. $H.\mathbf{P} = (P_x, P_y, P_z)$ is a single point on the $H.\mathbf{L}$. See Figure 4.

Thus, $H.\mathbf{L}$ has the form

$$H.\mathbf{L} = H.\mathbf{P} + a \cdot H.\mathbf{v}, \tag{7}$$

where a is a real number. $H.\mathbf{s}$ has the form

$$H.\mathbf{s} = H.\mathbf{P} \times H.\mathbf{v} + H.p \cdot H.\mathbf{v}, \tag{8}$$

where $H.p$ is the pitch of H. Then, the twist matrix representation of H is given by

$$\mathbf{T}^H = [H.\mathbf{v};\ \ H.\mathbf{s}]. \tag{9}$$

2.2.4. Cylindrical Joint. The dof of a *cylindrical joint* (C) is two. We use $C.\mathbf{L}$ to characterize the axis of C. The unit vector of $C.\mathbf{L}$ is characterized by $C.\mathbf{v} = (v_x, v_y, v_z)$. $C.\mathbf{P} = (P_x, P_y, P_z)$ is a single point on the $C.\mathbf{L}$. See Figure 5.

Thus, $C.\mathbf{L}$ has the form

$$C.\mathbf{L} = C.\mathbf{P} + a \cdot C.\mathbf{v}, \tag{10}$$

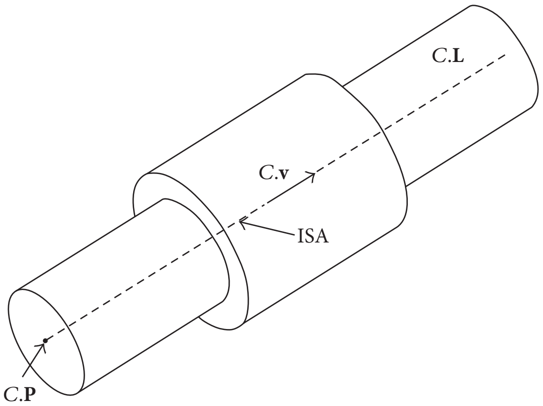


Figure 5: Cylindrical joint.

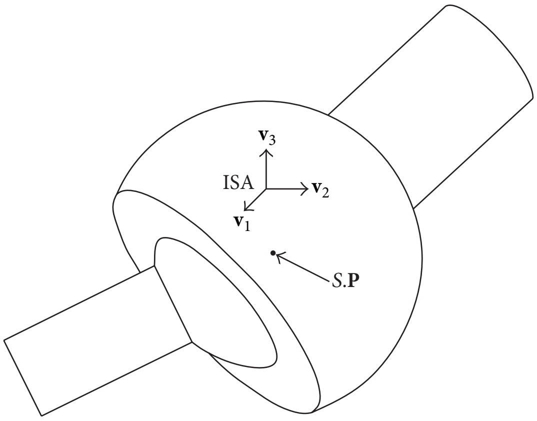


Figure 6: Spherical joint.

where a is a real number. $C.\mathbf{s}$ has two components, as shown in

$$\begin{aligned} C.\mathbf{s}_1 &= C.\mathbf{P} \times C.\mathbf{v}, \\ C.\mathbf{s}_2 &= C.\mathbf{v}. \end{aligned} \tag{11}$$

Therefore, the twist matrix representation of C is given by

$$\mathbf{T}^C = \begin{bmatrix} C.\mathbf{v}; & C.\mathbf{s}_1 \\ \mathbf{0}; & C.\mathbf{s}_2 \end{bmatrix}. \tag{12}$$

2.2.5. Spherical Joint. The dof of a *spherical joint* (S) is three. We use $S.\mathbf{P} = (P_x, P_y, P_z)$ to characterize the central point of S and define $\mathbf{v}_1 = (1, 0, 0)$, $\mathbf{v}_2 = (0, 1, 0)$, and $\mathbf{v}_3 = (0, 0, 1)$. See Figure 6.

Thus, $S.\mathbf{s}$ has three components, as shown in

$$\begin{aligned} S.\mathbf{s}_1 &= S.\mathbf{P} \times \mathbf{v}_1 = \left(0, P_z, -P_y\right), \\ S.\mathbf{s}_2 &= S.\mathbf{P} \times \mathbf{v}_2 = \left(-P_z, 0, P_x\right), \\ S.\mathbf{s}_3 &= S.\mathbf{P} \times \mathbf{v}_3 = \left(P_y, -P_x, 0\right). \end{aligned} \tag{13}$$

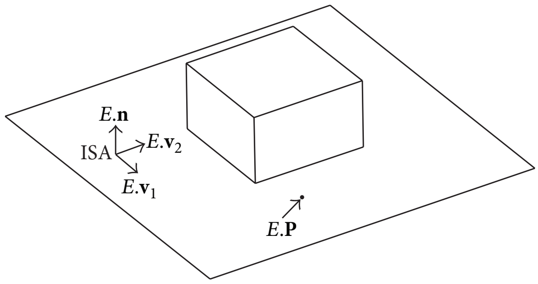


Figure 7: Planar joint.

Then, the twist matrix representation of S is given by

$$\mathbf{T}^S = \begin{bmatrix} \mathbf{v}_1; & S.\mathbf{s}_1 \\ \mathbf{v}_2; & S.\mathbf{s}_2 \\ \mathbf{v}_3; & S.\mathbf{s}_3 \end{bmatrix}. \tag{14}$$

2.2.6. Planar Joint. The dof of a *planar joint* (E) is three. The vectors $E.\mathbf{v}_1 = (\mathbf{v}_{x1}, \mathbf{v}_{y1}, \mathbf{v}_{z1})$ and $E.\mathbf{v}_2 = (\mathbf{v}_{x2}, \mathbf{v}_{y2}, \mathbf{v}_{z2})$ are orthogonal in the planar surface, and the unit vector $E.\mathbf{n}$ is the outer normal vector. $E.\mathbf{P} = (P_x, P_y, P_z)$ is a single point on the surface. See Figure 7.

Thus, $E.\mathbf{s}$ has three components, as shown in

$$\begin{aligned} E.\mathbf{s}_1 &= E.\mathbf{v}_1, \\ E.\mathbf{s}_2 &= E.\mathbf{v}_2, \\ E.\mathbf{s}_3 &= E.\mathbf{P} \times E.\mathbf{n}. \end{aligned} \tag{15}$$

Then, the twist matrix representation of E is given by

$$\mathbf{T}^E = \begin{bmatrix} \mathbf{0}; & E.\mathbf{s}_1 \\ \mathbf{0}; & E.\mathbf{s}_2 \\ E.\mathbf{n}; & E.\mathbf{s}_3 \end{bmatrix}. \tag{16}$$

3. Reciprocal Screw Models of Geometric Constraints

To transform arbitrary combinations of geometric constraints between two parts into the corresponding kinematic joint, we should use the same mathematical model with kinematic joints or a model that can be easily transformed into a twist. This section shows how to define geometry constraints using wrench.

3.1. Wrench. A wrench [32, 33] takes the form

$$\mathbf{W} = \left[f_x f_y f_z;\ m_x m_y m_z \right]. \tag{17}$$

A wrench is also a screw and describes the resultant force and moment of a force system acting on a rigid body. Any set of forces and couples applied to a body can be reduced to a set comprising a single force acting along a specific line in space and a pure couple acting in a plane perpendicular to that line. This means that the motion described by a wrench

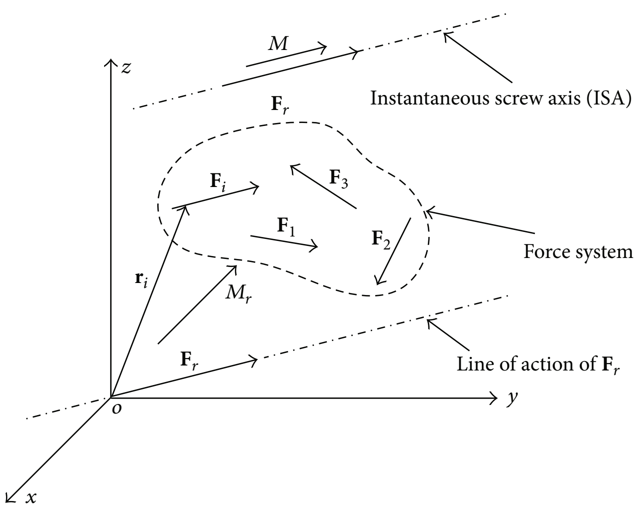


Figure 8: Wrench.

is forbidden. The first triplet describes the resultant force in a global reference frame. The second triplet represents the resultant moment of the force system about the origin of the global frame. See Figure 8.

Both a wrench and geometry constraint are structural constraints. Therefore, we can use a wrench to characterize a component of the geometry constraints between two parts. Afterwards, the geometry constraints can be transformed into a corresponding wrench matrix.

3.2. Basic Components of Geometric Constraints. When we describe a part in space, the orientation and position should be specified [34]. The orientation cannot be described in absolute terms. An orientation is given by a rotation from some known reference orientation. The amount of rotation is known as an angular displacement. In other words, describing an orientation is mathematically equivalent to describing an angular displacement. In the same way, when we specify the position of a part, we cannot do so in absolute terms; we must always do so within the context of a specific reference frame. Therefore, specifying a position is actually the same as specifying an amount of translation from a given reference point (usually the origin of some coordinate system).

In practice, there are many geometry constraints, such as *align* and *point on line*. We can use one or arbitrary combinations of geometric constraints to restrict the relative orientation and position between two parts. As we know, common geometry constraints, such as *coaxial constraints* and *spherical constraints*, usually have a high *degree of constraint* (doc). However, these geometry constraints are all combined by the three different types of basic constraints: *dot-1 constraint*, *dot-2 constraint*, and the *distance constraint* [35], which can be divided into two categories: the orientation part and the position part.

Consider a pair of rigid bodies, denoted as bodies i and j in Figure 9. Reference points $\mathbf{P}_i$ and $\mathbf{P}_j$, reference frames $\mathbf{f}_i$-$\mathbf{g}_i$-$\mathbf{h}_i$ and $\mathbf{f}_j$-$\mathbf{g}_j$-$\mathbf{h}_j$, and nonzero unit vectors $\mathbf{a}_i$ and $\mathbf{a}_j$ are fixed in bodies i and j, respectively. Vector $\mathbf{d}_{ij}$ connects $\mathbf{P}_i$ and $\mathbf{P}_j$ between bodies. The basic constraints are often characterized by such elements.

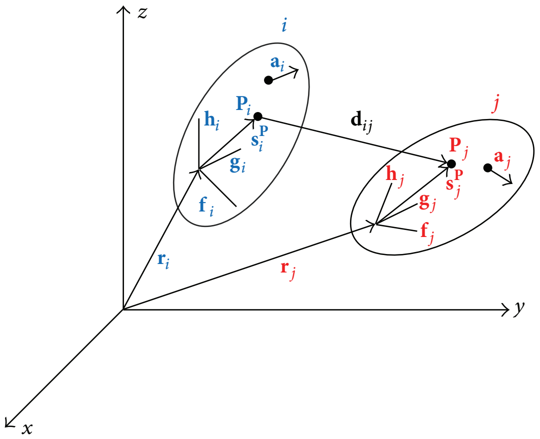


Figure 9: Vectors fixed in and between bodies.

3.2.1. Dot-1 Constraint. *Dot-1 constraint* restricts the relative orientation of a pair of bodies; that is,

$$\Phi^{d1}\left(\mathbf{a}_i,\mathbf{a}_j,C_1\right)\equiv \mathbf{a}_i^{\mathbf{T}}\mathbf{a}_j - C_1 = 0,\quad C_1\in(-1,1),\tag{18}$$

where $\operatorname{doc}(\Phi^{d1}) = 1$ and *dot-1 constraint* is symmetric with regard to bodies i and j. Note that *dot-1 constraint* would not be a basic constraint if $C_1 = \pm 1$. This means that the vectors $\mathbf{a}_i$ and $\mathbf{a}_j$ are parallel in the same or opposite directions. In this case, $\operatorname{doc}(\Phi^{d1}(\mathbf{a}_i,\mathbf{a}_j,\pm 1)) = 2$.

Writing the vectors $\mathbf{a}_i$ and $\mathbf{a}_j$ in terms of their respective body reference transformation matrices $\mathbf{A}_i$ and $\mathbf{A}_j$ and body-fixed constant vectors $\mathbf{a}_i'$ and $\mathbf{a}_j'$, where $\mathbf{a}_i = \mathbf{A}_i\mathbf{a}_i'$ and $\mathbf{a}_j = \mathbf{A}_j\mathbf{a}_j'$, (18) has the form

$$\Phi^{d1}\left(\mathbf{a}_i,\mathbf{a}_j,C_1\right)\equiv \mathbf{a}_i'^{\mathbf{T}}\mathbf{A}_i^{\mathbf{T}}\mathbf{A}_j\mathbf{a}_j' - C_1 = 0, \qquad C_1\in(-1,1),\tag{19}$$

where the transformation matrices $\mathbf{A}_i$ and $\mathbf{A}_j$ in (19) depend on the orientation generalized coordinates of bodies i and j, respectively.

As a result, the wrench representation for *dot-1 constraint* should also freeze the relative orientation of the vectors $\mathbf{a}_i$ and $\mathbf{a}_j$. Therefore, the wrench matrix is given by

$$\mathbf{W}^{d1} = \left(\mathbf{0};\ \mathbf{a}_i\times\mathbf{a}_j\right).\tag{20}$$

If we use the body reference transformation matrices, (20) has the form

$$\mathbf{W}^{d1} = \left(\mathbf{0};\ \left(\mathbf{a}_i'^{\mathbf{T}}\mathbf{A}_i^{\mathbf{T}}\right)\times\left(\mathbf{A}_j\mathbf{a}_j'\right)\right).\tag{21}$$

3.2.2. Dot-2 Constraint. *Dot-2 constraint* restricts the relative position of a pair of bodies; that is,

$$\Phi^{d2}\left(\mathbf{a}_i,\mathbf{d}_{ij},C_2\right)\equiv \mathbf{a}_i^{\mathbf{T}}\mathbf{d}_{ij} - C_2 = 0,\quad \mathbf{d}_{ij}\neq\mathbf{0},\tag{22}$$

where $\mathrm{doc}(\Phi^{d2}) = 1$, the same as that with *dot-1 constraint*, although *dot-2 constraint* is not symmetric with regard to bodies i and j. Analytically, it is important to recall that *dot-2 constraint* breaks down if $\mathbf{d}_{ij} = \mathbf{0}$. Write the vector $\mathbf{d}_{ij}$ as

$$\mathbf{d}_{ij} = \mathbf{r}_j + \mathbf{A}_j \mathbf{s}_j'^{\mathbf{P}} - \mathbf{r}_i - \mathbf{A}_i \mathbf{s}_i'^{\mathbf{P}}. \quad (23)$$

Equation (22) becomes

$$\Phi^{d2}\left(\mathbf{a}_i, \mathbf{d}_{ij}, C_2\right) \equiv \mathbf{a}_i'^{\mathbf{T}} \mathbf{A}_i^{\mathbf{T}} \left(\mathbf{r}_j + \mathbf{A}_j \mathbf{s}_j'^{\mathbf{P}} - \mathbf{r}_i - \mathbf{A}_i \mathbf{s}_i'^{\mathbf{P}}\right) - C_2 = 0. \quad (24)$$

As a result, the wrench representation for *dot-2 constraint* must freeze the relative translation along the vector $\mathbf{a}_i$. Therefore, the wrench matrix is given by

$$\mathbf{W}^{d2} = \left(\mathbf{a}_i;\ \mathbf{0}\right). \quad (25)$$

Equation (25) also has the form

$$\mathbf{W}^{d2} = \left(\mathbf{A}_i \mathbf{a}_i';\ \mathbf{0}\right). \quad (26)$$

3.2.3. Distance Constraint. In practice, we often require a pair of points on two bodies to coincide or maintain a determined distance; that is,

$$\Phi^{s}\left(P_i, P_j, C_3\right) \equiv \mathbf{d}_{ij}^{\mathbf{T}} \mathbf{d}_{ij} - C_3 \cdot C_3 = 0, \quad (27)$$

where $\mathrm{doc}(\Phi^{s})$ would be different depending on whether C_3 is zero. Therefore, we have

$$\begin{aligned} \mathrm{doc}\left(\Phi^{s}\right) &= 1 \quad \text{if } C_3 \neq 0, \\ \mathrm{doc}\left(\Phi^{s}\right) &= 3 \quad \text{if } C_3 = 0. \end{aligned} \quad (28)$$

As a result, the wrench representation for the *distance constraint* should have two types of expressions, as shown in (29) and (30):

$$\mathbf{W}^{s} = \left[\mathbf{d}_{ij};\ \mathbf{0}\right] \quad \text{if } C_3 \neq 0, \quad (29)$$

where the wrench representation for the *distance constraint* should freeze the relative translation along the vector $\mathbf{d}_{ij}$ if $C_3 \neq 0$. However, if $C_3 = 0$, the wrench representation has the form

$$\mathbf{W}^{s} = \begin{bmatrix} 1 & 0 & 0; & \mathbf{0} \\ 0 & 1 & 0; & \mathbf{0} \\ 0 & 0 & 1; & \mathbf{0} \end{bmatrix} \quad \text{if } C_3 = 0, \quad (30)$$

where the wrench, which is illustrated in (30), freezes all the relative translations along the axes x, y, and z.

3.3. Wrench Representations of Geometry Constraints. To introduce geometry constraints, we give Figure 10.

In Figure 10, reference points $\mathbf{P}_1$ and $\mathbf{P}_2$, reference frames $\mathbf{f}_1$-$\mathbf{g}_1$-$\mathbf{h}_1$ and $\mathbf{f}_2$-$\mathbf{g}_2$-$\mathbf{h}_2$, and central axes $\mathbf{L}_1$ and $\mathbf{L}_2$ are fixed in cylinders B_1 and B_2, respectively. Vector $\mathbf{d}_{12}$ connects $\mathbf{P}_1$ and $\mathbf{P}_2$ between the cylinders.

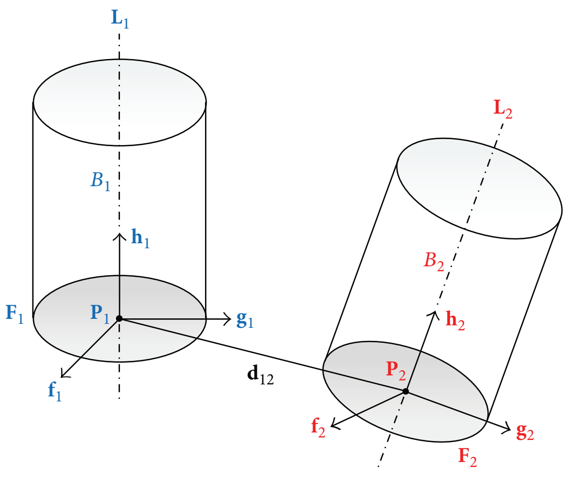


Figure 10: Instance of geometry constraints.

The reference point $\mathbf{P}_1$ has the form

$$\mathbf{P}_1 = \left(P_{x1}, P_{y1}, P_{z1}\right). \quad (31)$$

The central axis $\mathbf{L}_1$ has the form

$$\mathbf{L}_1 = \mathbf{P}_1 + a \cdot \mathbf{h}_1, \quad (32)$$

where a is a real number. The planar surface $\mathbf{F}_1$ has the form

$$\mathbf{F}_1 = \mathbf{P}_1 + b \cdot \mathbf{f}_1 + c \cdot \mathbf{g}_1, \quad (33)$$

where b and c are both real numbers. At the same time, the elements $\mathbf{P}_2$, $\mathbf{L}_2$, and $\mathbf{F}_2$ have the same forms. The following geometry constraints are often characterized by such elements.

We can divide all different types of geometry constraints, which have more than one doc, into six categories: *coaxial constraint*, *coplanar constraint*, *spherical constraint*, *align constraint*, *distance between line and surface*, and *point on line*. We have modeled the three basic components of geometry constraints in terms of wrench. Thus, the wrench models of these six types of geometry constraints can be automatically deduced.

3.3.1. Coaxial Constraint. We use $CoiLL(\mathbf{L}_1, \mathbf{L}_2)$ to denote the *coaxial constraint* of axes $\mathbf{L}_1$ and $\mathbf{L}_2$, as shown in Figure 11.

The doc of $CoiLL(\mathbf{L}_1, \mathbf{L}_2)$ is four, which restricts two relative orientations and two relative translations. In other words, it comprises two *dot-1 constraints* and two *dot-2 constraints*, as shown in

$$\Phi_1^{d1}\left(\mathbf{h}_1, \mathbf{f}_2, 0\right), \Phi_2^{d1}\left(\mathbf{h}_1, \mathbf{g}_2, 0\right), \Phi_3^{d2}\left(\mathbf{f}_1, \mathbf{d}_{12}, 0\right), \Phi_4^{d2}\left(\mathbf{g}_1, \mathbf{d}_{12}, 0\right). \quad (34)$$

Then, the wrench matrix of $CoiLL(\mathbf{L}_1, \mathbf{L}_2)$ is given by

$$\mathbf{W}_{CoiLL(\mathbf{L}_1,\mathbf{L}_2)} = \begin{bmatrix} \mathbf{0}; & \mathbf{h}_1 \times \mathbf{f}_2 \\ \mathbf{0}; & \mathbf{h}_1 \times \mathbf{g}_2 \\ \mathbf{f}_1; & \mathbf{0} \\ \mathbf{g}_1; & \mathbf{0} \end{bmatrix}. \quad (35)$$

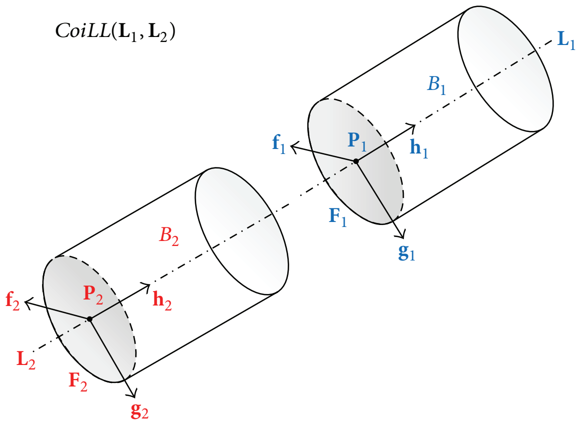


Figure 11: Coaxial constraint.

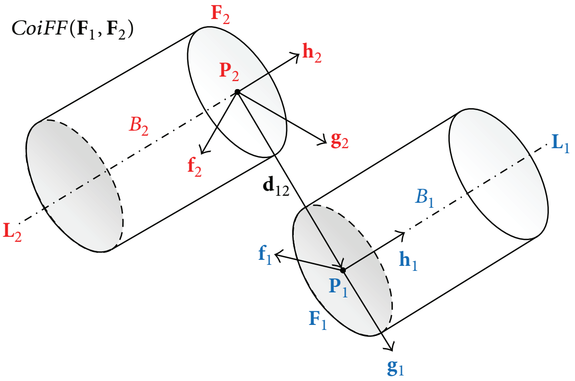


Figure 12: Coplanar constraint.

The corresponding kinematic joint of *coaxial constraint* is *cylindrical joint*.

3.3.2. Coplanar Constraint. We use *CoiFF*$(\mathbf{F}_1, \mathbf{F}_2)$ to denote the *coplanar constraint* of planar surfaces $\mathbf{F}_1$ and $\mathbf{F}_2$, as shown in Figure 12.

The doc of *CoiFF*$(\mathbf{F}_1, \mathbf{F}_2)$ is three, which restricts two relative orientations and one relative translation. In other words, it comprises two *dot-1 constraints* and one *dot-2 constraint*, as shown in

$$\Phi_1{}^{d1}\left(\mathbf{h}_1, \mathbf{f}_2, 0\right), \Phi_2{}^{d1}\left(\mathbf{h}_1, \mathbf{g}_2, 0\right), \Phi_3{}^{d2}\left(\mathbf{h}_1, \mathbf{d}_{12}, 0\right). \quad (36)$$

Then, the wrench matrix of *CoiFF*$(\mathbf{F}_1, \mathbf{F}_2)$ is given by

$$\mathbf{W}_{CoiFF(\mathbf{F}_1,\mathbf{F}_2)} = \begin{bmatrix} \mathbf{0}; & \mathbf{h}_1 \times \mathbf{f}_2 \\ \mathbf{0}; & \mathbf{h}_1 \times \mathbf{g}_2 \\ \mathbf{h}_1; & \mathbf{0} \end{bmatrix}. \quad (37)$$

The corresponding kinematic joint of *coplanar constraint* is *planar joint*.

3.3.3. Spherical Constraint. We use *CoiPP*$(\mathbf{P}_1, \mathbf{P}_2)$ to denote the *spherical constraint* of two points $\mathbf{P}_1$ and $\mathbf{P}_2$, as shown in Figure 13.

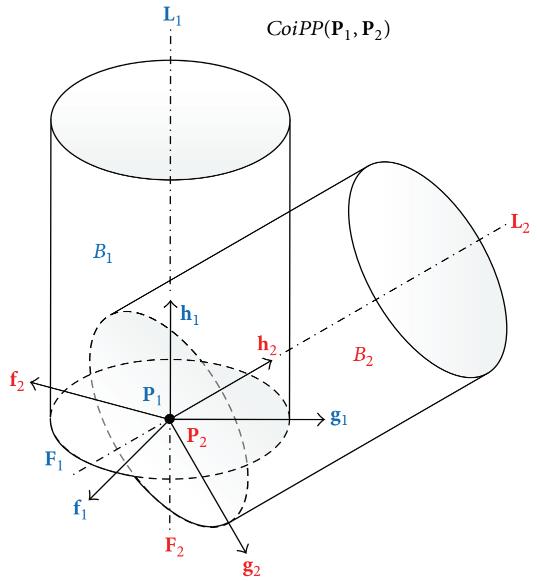


Figure 13: Spherical constraint.

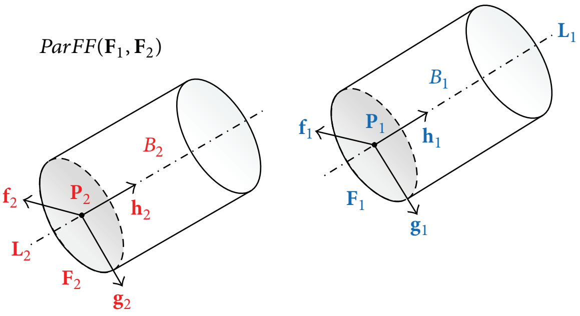


Figure 14: Align constraint.

The doc of *CoiPP*$(\mathbf{P}_1, \mathbf{P}_2)$ is three, which restricts three relative translations in three unrelated directions. In other words, it comprises one *distance constraint*, which has $C_3 = 0$ in Section 3.2.3, as shown in

$$\Phi^s\left(P_i, P_j, 0\right). \quad (38)$$

Then, the wrench matrix of *CoiPP*$(\mathbf{P}_1, \mathbf{P}_2)$ is given by

$$\mathbf{W}_{CoiPP(\mathbf{P}_1,\mathbf{P}_2)} = \begin{bmatrix} 1 & 0 & 0; & \mathbf{0} \\ 0 & 1 & 0; & \mathbf{0} \\ 0 & 0 & 1; & \mathbf{0} \end{bmatrix}. \quad (39)$$

The corresponding kinematic joint of *spherical constraint* is *spherical joint*.

3.3.4. Align Constraint. Align constraint (*ParFF*) is used to make two planar surfaces $\mathbf{F}_1$ and $\mathbf{F}_2$ parallel, as shown in Figure 14.

The doc of *ParFF*$(\mathbf{F}_1, \mathbf{F}_2)$ is two, which restricts two relative orientations. In other words, it comprises two *dot-1 constraints*, as shown in

$$\Phi_1{}^{d1}\left(\mathbf{h}_1, \mathbf{f}_2, 0\right), \Phi_2{}^{d1}\left(\mathbf{h}_1, \mathbf{g}_2, 0\right). \quad (40)$$

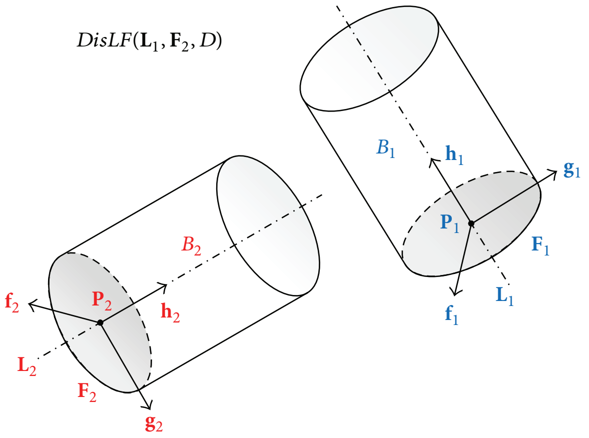


Figure 15: Distance between line and surface.

Then, the wrench matrix of $ParFF(\mathbf{F}_1, \mathbf{F}_2)$ is given by

$$\mathbf{W}_{ParFF(\mathbf{F}_1,\mathbf{F}_2)} = \begin{bmatrix} \mathbf{0}; & \mathbf{h}_1 \times \mathbf{f}_2 \\ \mathbf{0}; & \mathbf{h}_1 \times \mathbf{g}_2 \end{bmatrix}. \quad (41)$$

We cannot find a corresponding kinematic joint for *align constraint* among the common kinematic joints.

3.3.5. Distance between Line and Surface. We use $DisLF(\mathbf{L}_1, \mathbf{F}_2, D)$ to denote the axis $\mathbf{L}_1$ and the surface $\mathbf{F}_2$ that maintain a certain distance D, as shown in Figure 15.

The doc of $DisLF(\mathbf{L}_1, \mathbf{F}_2, D)$ is two, which restricts one relative orientation and one relative translation. In other words, it comprises one *dot-1 constraint* and one *dot-2 constraint*, as shown in

$$\Phi_1{}^{d1}\left(\mathbf{h}_1, \mathbf{h}_2, 0\right), \Phi_2{}^{d2}\left(\mathbf{h}_2, \mathbf{d}_{12}, D\right). \quad (42)$$

Then, the wrench matrix of $DisLF(\mathbf{L}_1, \mathbf{F}_2, D)$ is given by

$$\mathbf{W}_{DisLF(\mathbf{L}_1,\mathbf{F}_2,D)} = \begin{bmatrix} \mathbf{0}; & \mathbf{h}_1 \times \mathbf{h}_2 \\ \mathbf{h}_2; & \mathbf{0} \end{bmatrix}. \quad (43)$$

We also cannot find a corresponding kinematic joint for *distance between line and surface* among the common kinematic joints.

3.3.6. Point on Line. We use $CoiPL(\mathbf{P}_1, \mathbf{L}_2)$ to denote the point $\mathbf{P}_1$ on line $\mathbf{L}_2$, as shown in Figure 16.

The doc of $CoiPL(\mathbf{P}_1, \mathbf{L}_2)$ is two, which restricts two relative translations. In other words, it comprises two *dot-2 constraints*, as shown in

$$\Phi_1{}^{d2}\left(\mathbf{f}_2, \mathbf{d}_{12}, 0\right), \Phi_2{}^{d2}\left(\mathbf{g}_2, \mathbf{d}_{12}, 0\right). \quad (44)$$

Then, the wrench matrix of $CoiPL(\mathbf{P}_1, \mathbf{L}_2)$ is given by

$$\mathbf{W}_{DisLF(\mathbf{L}_1,\mathbf{F}_2,D)} = \begin{bmatrix} \mathbf{f}_2; & \mathbf{0} \\ \mathbf{g}_2; & \mathbf{0} \end{bmatrix}. \quad (45)$$

We still cannot find a corresponding kinematic joint for *point on line* among the common kinematic joints.

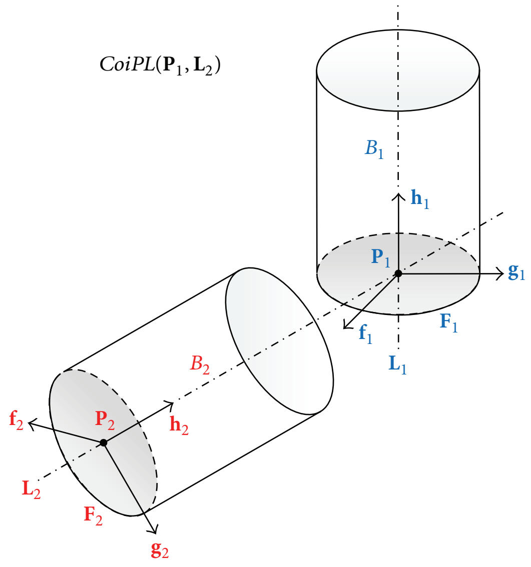


Figure 16: Point on line.

4. Recognition of Kinematic Joints

The key to recognizing kinematic joints in terms of geometry constraints is the reciprocity relation. We show how to define geometry constraints using wrench. The space of screws that defines the range of wrenches that can be exerted upon a body and the space of screws that defines the range of twists that the body can perform without breaking any contacts in the joints of the mechanism are mutually reciprocal screw spaces. When we know the wrenches acting upon a body, we can determine the motions it can perform. Conversely, if we know the motions a body is capable of performing, then the constraints that are exerted upon it can also be deduced.

4.1. Algorithm. We assume that A is a part in a 3D assembly model restricted by parts $B_1, B_2, \ldots, B_n$. The twist space of A is the space of screws reciprocal to the wrenches of contact acting upon it. The space of wrenches acting upon A is the union of all the individual wrenches acting upon it from the parts B_i. These individual wrenches are to be considered as acting alone. Algorithm 1 provides the twist space of A, denoted by $\mathbf{T}^A$.

The $Reciprocal(\mathbf{P})$ in Algorithm 1 means to exchange the first triplet and the second triplet of $\mathbf{P}$. For example, if $\mathbf{P} = [a, b, c;\ d, e, f]$, then we have $Reciprocal(\mathbf{P}) = [d, e, f;\ a, b, c]$. The recursive nature of the procedure requires the computation of the wrench space and twist space of each component. This proves to be computationally expensive because wrench space and twist space are defined in terms of the reciprocal of one another. The operation of deriving reciprocal screws involves computing the null space of matrices. The speed of the algorithm is linearly related to the number of B_i. That is, the algorithm is of the order $O(n)$.

4.2. Reciprocal Product. According to reciprocal screw theory, two screws (a twist and a wrench) are said to be mutually reciprocal if their reciprocal product is zero. The reciprocal

```
BEGIN
  FOR each part $B_i$, which restrict $A$.
    Convert all the geometry constraints between $A$ and $B_i$ into basic components $C_A^j$, $j = 1, \ldots, l$.
    Determine the wrenches $\mathbf{W}_A^k$, $k = 1, \ldots, m$ acting upon $A$ through $C_A^j$. We have $m$ not less than $l$.
    Collect all the wrenches $\mathbf{W}_A^k$ into a matrix $\mathbf{W}$.
  END FOR
  Compute the orthogonal matrix $\mathbf{Q}$ in terms of $\mathbf{W}$ by Gram-Schmidt orthogonalization [31].
  IF $Rank(\mathbf{Q}) = 6$ THEN
    Return "IMMOBILE".
  ELSE
    Compute the orthogonal complement matrix $\mathbf{P}$ of $\mathbf{Q}$.
  END IF
  $\mathbf{T}^A = Reciprocal(\mathbf{P})$.
END
```

Algorithm 1: Function twist space.

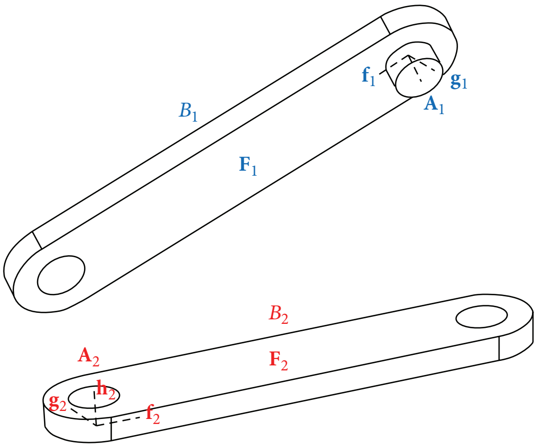


Figure 17: Redundant constraint.

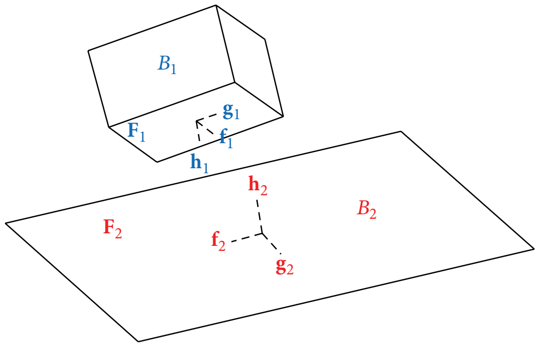


Figure 18: Recognition of planar joint.

product of a twist $\mathbf{T}_1$ and a wrench $\mathbf{W}_1$ thatisequal to zero is given by

$$\mathbf{T}_1 \circ \mathbf{W}_1 = \mathbf{T}_{1.1-3} \cdot \mathbf{W}_{2.4-6} + \mathbf{T}_{1.4-6} \cdot \mathbf{W}_{2.1-3} = [\mathbf{0};\ \mathbf{0}]. \quad (46)$$

Mathematically, because reciprocity imposes the relation $\mathbf{T}_1 \circ \mathbf{W}_1$, we can compute one of these quantities if the other is known. This allows us to compute the twists that a body may execute while in contact with other bodies without breaking the contacts.

4.3. Redundant Constraint. When more than one geometry constraint acts upon the same part, because some components of these constraints may be redundant, the resultant motion is the logical intersection of the individual wrench matrix that defines each constraint, for example, when we model a *revolute joint* R_{12} with two links: B_1 and B_2. The constraints used are $CoiLL(\mathbf{A}_1, \mathbf{A}_2)$ and $CoiFF(\mathbf{F}_1, \mathbf{F}_2)$. See Figure 17.

Also $\mathrm{dof}(R_{12}) = 1$. This means that $\mathrm{doc}(B_1, B_2) = 5$. However, we have $\mathrm{doc}(CoiLL(\mathbf{A}_1, \mathbf{A}_2)) = 4$ and $\mathrm{doc}(CoiFF(\mathbf{F}_1, \mathbf{F}_2)) = 3$. In other words, the dimension of the orthogonal wrench space, which comprises $\mathbf{W}_{CoiLL(\mathbf{A}_1,\mathbf{A}_2)}$ and $\mathbf{W}_{CoiFF(\mathbf{F}_1,\mathbf{F}_2)}$, is 5. Therefore, the motions that a body is capable of performing while under the action of multiple wrenches are the logical intersection of the motions it can perform under the action of each wrench acting alone. Thus, we can eliminate the redundant components of different geometry constraints that act upon the same part by using Gram-Schmidt orthogonalization in Algorithm 1.

5. Examples

In each of the examples below, different types of kinematic joints are recognized.

Example 1 (planar joint). Figure 18 shows a *planar joint* consisting of two parts, B_1 and B_2, and a *coplanar constraint*: $CoiFF(\mathbf{F}_1, \mathbf{F}_2)$.

Following the process outlined in Section 3.3.2, we find, based on the definition of the *coplanar constraint*, that the wrench matrix is

$$\mathbf{W}_{CoiFF(\mathbf{F}_1,\mathbf{F}_2)} = \begin{bmatrix} \mathbf{0}; & \mathbf{h}_1 \times \mathbf{f}_2 \\ \mathbf{0}; & \mathbf{h}_1 \times \mathbf{g}_2 \\ \mathbf{h}_1; & \mathbf{0} \end{bmatrix}, \quad (47)$$

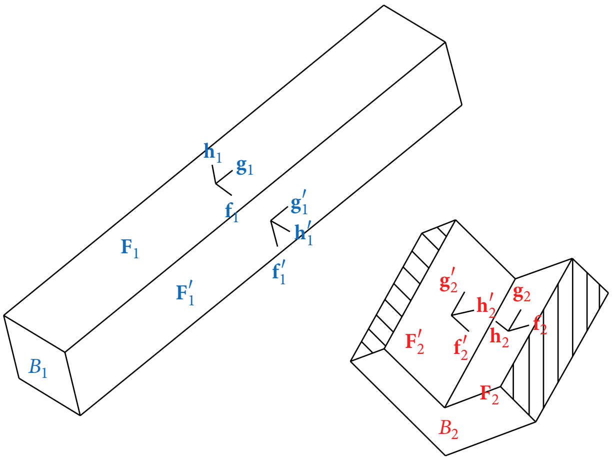


FIGURE 19: Recognition of prismatic joint.

where $\mathbf{W}_{CoiFF(\mathbf{F}_1,\mathbf{F}_2)}$ is already an orthogonal matrix. Then, the orthogonal complement matrix $\mathbf{P}_{CoiFF(\mathbf{F}_1,\mathbf{F}_2)}$ is

$$\mathbf{P}_{CoiFF(\mathbf{F}_1,\mathbf{F}_2)} = \begin{bmatrix} \mathbf{f}_1; & \mathbf{0} \\ \mathbf{g}_1; & \mathbf{0} \\ \mathbf{0}; & \mathbf{h}_1 \end{bmatrix}. \tag{48}$$

Therefore, the twist matrix $\mathbf{T}_{12}$ between B_1 and B_2 has the following form:

$$\mathbf{T}_{12} = Reciprocal\left(\mathbf{P}_{CoiFF(\mathbf{F}_1,\mathbf{F}_2)}\right) = \begin{bmatrix} \mathbf{0}; & \mathbf{f}_1 \\ \mathbf{0}; & \mathbf{g}_1 \\ \mathbf{h}_1; & \mathbf{0} \end{bmatrix}, \tag{49}$$

where $\mathbf{T}_{12}$ represents the motion of a *planar joint*, which is illustrated in Section 2.2.6. Moreover, the recognition process of a *cylindrical joint* or *spherical joint* is approximately the same as that of the *planar joint*.

Example 2 (prismatic joint). Figure 19 shows a *prismatic joint* consisting of two parts, B_1 and B_2, and two *coplanar constraints*: $CoiFF(\mathbf{F}_1, \mathbf{F}_2)$ and $CoiFF(\mathbf{F}_1', \mathbf{F}_2')$.

Following the process outlined in Example 1, we find that the wrench matrices between B_1 and B_2 are

$$\begin{aligned} \mathbf{W}_{CoiFF(\mathbf{F}_1,\mathbf{F}_2)} &= \begin{bmatrix} \mathbf{0}; & \mathbf{h}_1 \times \mathbf{f}_2 \\ \mathbf{0}; & \mathbf{h}_1 \times \mathbf{g}_2 \\ \mathbf{h}_1; & \mathbf{0} \end{bmatrix}, \\ \mathbf{W}_{CoiFF(\mathbf{F}_1',\mathbf{F}_2')} &= \begin{bmatrix} \mathbf{0}; & \mathbf{h}_1' \times \mathbf{f}_2' \\ \mathbf{0}; & \mathbf{h}_1' \times \mathbf{g}_2' \\ \mathbf{h}_1'; & \mathbf{0} \end{bmatrix}, \end{aligned} \tag{50}$$

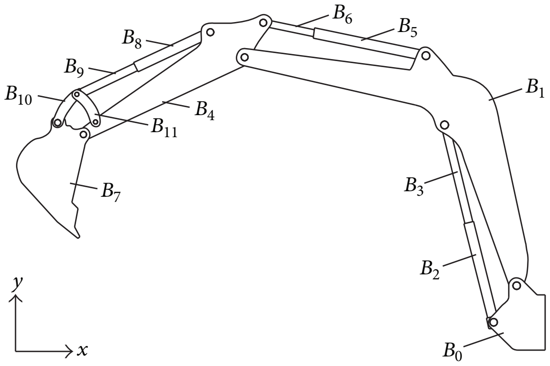


FIGURE 20: Hydraulic grab.

where $\mathbf{W}_{12} = \mathbf{W}_{CoiFF(\mathbf{F}_1,\mathbf{F}_2)} \cup \mathbf{W}_{CoiFF(\mathbf{F}_1',\mathbf{F}_2')}$. The orthogonal matrix $\mathbf{Q}_{12}$ of $\mathbf{W}_{12}$ is

$$\mathbf{Q}_{12} = \begin{bmatrix} \mathbf{0}; & \mathbf{h}_1 \times \mathbf{f}_2 \\ \mathbf{0}; & \mathbf{h}_1 \times \mathbf{g}_2 \\ \mathbf{0}; & \mathbf{h}_1' \times \mathbf{g}_2' \\ \mathbf{h}_1; & \mathbf{0} \\ \mathbf{h}_1'; & \mathbf{0} \end{bmatrix}. \tag{51}$$

The orthogonal complement matrix $\mathbf{P}_{12}$ of $\mathbf{Q}_{12}$ is

$$\mathbf{P}_{12} = [\mathbf{g}_1; \ \mathbf{0}]. \tag{52}$$

Therefore, the twist matrix $\mathbf{T}_{12}$ between B_1 and B_2 has the following form:

$$\mathbf{T}_{12} = Reciprocal\left(\mathbf{P}_{12}\right) = [\mathbf{0}; \ \mathbf{g}_1], \tag{53}$$

where $\mathbf{T}_{12}$ represents the motion of a *prismatic joint*, which is shown in Section 2.2.1. Moreover, the recognition process of a *revolute joint* is approximately the same as that of the *prismatic joint*.

Example 3 (hydraulic grab). Here, we present a project example. A hydraulic grab with three degrees of freedom in Cartesian coordinates is illustrated in Figure 20.

The hydraulic grab comprises 12 parts, B_0, B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, B_9, B_{10}, and B_{11}, 12 *coplanar constraints*, $CoiFF(B_0, B_2)$, $CoiFF(B_0, B_1)$, $CoiFF(B_1, B_3)$, $CoiFF(B_1, B_5)$, $CoiFF(B_1, B_4)$, $CoiFF(B_4, B_6)$, $CoiFF(B_4, B_{11})$, $CoiFF(B_4, B_7)$, $CoiFF(B_4, B_8)$, $CoiFF(B_9, B_{10})$, $CoiFF(B_7, B_{10})$, and $CoiFF(B_{10}, B_{11})$, and 15 *coaxial constraints*, $CoiLL(B_0, B_2)$, $CoiLL(B_0, B_1)$, $CoiLL(B_1, B_3)$, $CoiLL(B_2, B_3)$, $CoiLL(B_1, B_5)$, $CoiLL(B_1, B_4)$, $CoiLL(B_4, B_6)$, $CoiLL(B_5, B_6)$, $CoiLL(B_4, B_{11})$, $CoiLL(B_4, B_7)$, $CoiLL(B_4, B_8)$, $CoiLL(B_8, B_9)$, $CoiLL(B_9, B_{10})$, $CoiLL(B_7, B_{10})$, and $CoiLL(B_{10}, B_{11})$. See Figure 21.

Here, we present the processes of recognition among B_1, B_2, and B_3, as shown in Figure 22. The geometry constraint between B_2 and B_3 is $CoiLL(B_2, B_3)$. Following the process

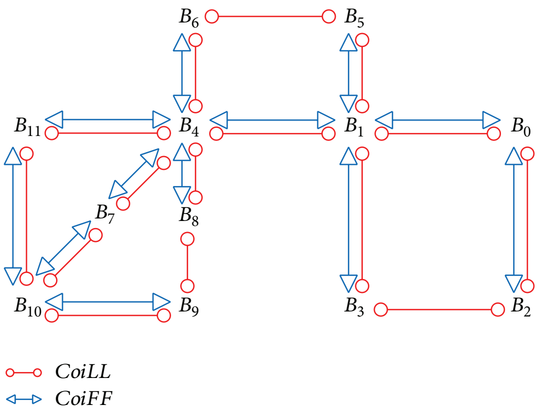


FIGURE 21: The constraint graph of a hydraulic grab.

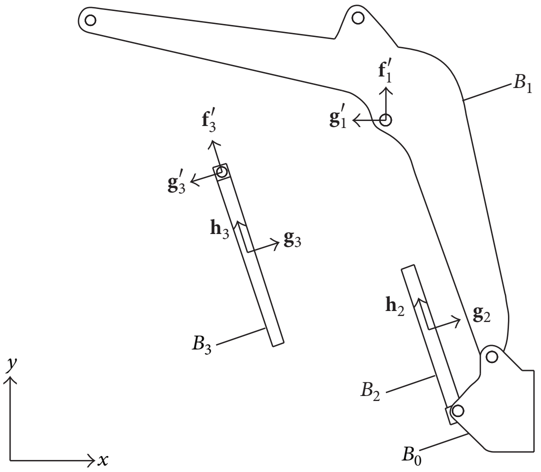


FIGURE 22: Parts B_1, B_2, and B_3 in a hydraulic grab.

outlined in Section 3.3.1, we find, based on the definition of the *coaxial constraint*, that the wrench matrix is

$$\mathbf{W}_{CoiLL(B_2,B_3)} = \begin{bmatrix} \mathbf{0}; & \mathbf{h}_2 \times \mathbf{f}_3 \\ \mathbf{0}; & \mathbf{h}_2 \times \mathbf{g}_3 \\ \mathbf{f}_2; & \mathbf{0} \\ \mathbf{g}_2; & \mathbf{0} \end{bmatrix}. \tag{54}$$

The orthogonal complement matrix is

$$\mathbf{P}_{CoiLL(B_2,B_3)} = \begin{bmatrix} \mathbf{0}; & \mathbf{h}_2 \\ \mathbf{h}_2; & \mathbf{0} \end{bmatrix}. \tag{55}$$

Then, the twist matrix $\mathbf{T}_{23}$ between B_2 and B_3 has the following form:

$$\mathbf{T}_{23} = Reciprocal\left(\mathbf{P}_{CoiLL(B_2,B_3)}\right) = \begin{bmatrix} \mathbf{h}_2; & \mathbf{0} \\ \mathbf{0}; & \mathbf{h}_2 \end{bmatrix}. \tag{56}$$

The geometry constraints between B_1 and B_3 are $CoiLL(B_1, B_3)$ and $CoiFF(B_1, B_3)$. Thus, the wrench matrices between B_1 and B_3 are

$$\mathbf{W}_{CoiLL(B_1,B_3)} = \begin{bmatrix} \mathbf{0}; & \mathbf{h}_3' \times \mathbf{f}_1' \\ \mathbf{0}; & \mathbf{h}_3' \times \mathbf{g}_1' \\ \mathbf{f}_3'; & \mathbf{0} \\ \mathbf{g}_3'; & \mathbf{0} \end{bmatrix},$$
$$\mathbf{W}_{CoiFF(B_1,B_3)} = \begin{bmatrix} \mathbf{0}; & \mathbf{h}_3' \times \mathbf{f}_1' \\ \mathbf{0}; & \mathbf{h}_3' \times \mathbf{g}_1' \\ \mathbf{h}_3'; & \mathbf{0} \end{bmatrix}, \tag{57}$$

where $\mathbf{W}_{13} = \mathbf{W}_{CoiLL(B_1,B_3)} \cup \mathbf{W}_{CoiFF(B_1,B_3)}$. The orthogonal matrix $\mathbf{Q}_{13}$ of $\mathbf{W}_{13}$ is given by

$$\mathbf{Q}_{13} = \begin{bmatrix} \mathbf{0}; & \mathbf{h}_3' \times \mathbf{f}_1' \\ \mathbf{0}; & \mathbf{h}_3' \times \mathbf{g}_1' \\ \mathbf{f}_3'; & \mathbf{0} \\ \mathbf{g}_3'; & \mathbf{0} \\ \mathbf{h}_3'; & \mathbf{0} \end{bmatrix}. \tag{58}$$

Then, the orthogonal complement matrix $\mathbf{P}_{13}$ of $\mathbf{Q}_{13}$ is

$$\mathbf{P}_{13} = \left[\mathbf{0};\ \mathbf{h}_3'\right]. \tag{59}$$

Therefore, the twist matrix $\mathbf{T}_{13}$ between B_1 and B_3 is

$$\mathbf{T}_{13} = Reciprocal\left(\mathbf{P}_{13}\right) = \left[\mathbf{h}_3';\ \mathbf{0}\right]. \tag{60}$$

In this paper, we propose a new method to automatically convert a 3D assembly model into a screw system based on reciprocal screw theory. According to the geometry constraints between any two parts in the hydraulic grab and Algorithm 1, we obtain the screw system, as shown in Figure 23.

As mentioned in [4–29], we can perform some different types of kinematic analyses using the screw system, such as determining the degrees of freedom of B_1 and B_4; that is,

$$\begin{aligned} \mathbf{T}_{B_1} &= \mathbf{T}_0^1 \cap \left(\mathbf{T}_0^2 \cup \mathbf{T}_2^3 \cup \mathbf{T}_1^3\right), \\ \mathbf{T}_{B_4} &= \left(\mathbf{T}_{B_1} \cup \mathbf{T}_1^4\right) \cap \left(\mathbf{T}_{B_1} \cup \mathbf{T}_1^5 \cup \mathbf{T}_5^6 \cup \mathbf{T}_4^6\right), \end{aligned} \tag{61}$$

where the resultant of the motions that the end-effector of a purely serial chain may perform due to the action of each joint acting alone is the union of that set and the resultant of the set of twists allowed separately by multiple contacts upon a body is the intersection of the individual twists. Thus, the degrees of freedom of B_1 are equal to the dimension of the orthogonal twist space of B_1, denoted by $\mathbf{T}_{B_1}$. In the same way, we can obtain the degrees of freedom of B_4 and any part in the hydraulic grab.

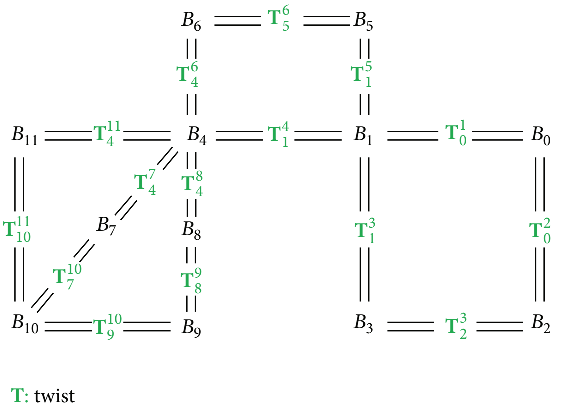


FIGURE 23: The screw system of a hydraulic grab.

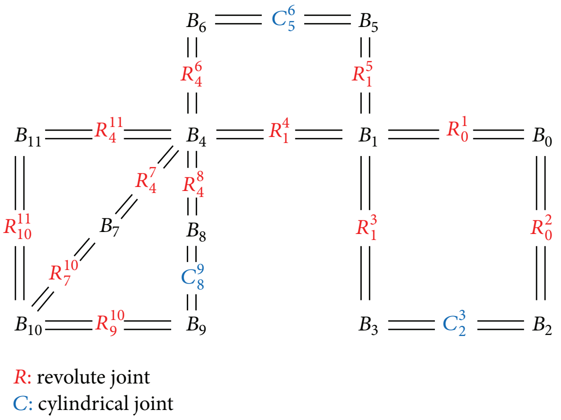


FIGURE 24: The kinematic joint model of a hydraulic grab.

According to Sections 2.2.2 and 2.2.4, the kinematic joints between B_2 and B_3 and B_1 and B_3 are the *cylindrical joint* and *revolute joint*. As a result, we can obtain the kinematic joint model. See Figure 24. If the corresponding kinematic joint between any two parts cannot be recognized, we can easily analyze the model's geometry constraints.

6. Conclusions

This paper investigates a method for determining the kinematic joints of assemblies restricted by arbitrary combinations of geometry constraints using reciprocal screw theory. We achieve the process of recognition in terms of the basic components of geometry constraints and kinematic joints. We describe the three basic components of geometric constraints in terms of wrench. At the same time, the specific operation of a matrix space is applied to eliminate the redundant components between different geometry constraints that act upon the same part. A user of this method can automatically and easily convert any assembly model into a screw system in any CAD system. Afterwards, the screw system can be easily used for validating kinematic functions, analyzing over- or underconstrained assembly configurations, and so on.

Additional Points

(i) We provide a new method for converting any 3D assembly model into a screw system based on reciprocal screw theory.

(ii) We describe the three basic components of geometric constraints in terms of wrench.

(iii) We eliminate the redundant components of different geometry constraints that act upon the same part by the specific operation of a matrix space.

(iv) We establish the maps between geometry constraints and kinematic joints as the reciprocal product of a twist and wrench, which is equal to zero.

(v) We present several application examples.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The authors would like to thank the Ministry of Science and Technology of the People's Republic of China (nos. 2011ZX02403-005, 2011CB706502, and 2013AA041301). This work is also supported by the National Natural Science Foundation of China (nos. 61370182 and 51175198).

References

[1] A. H. Slocum, *Precision Machine Design*, Prentice Hall, New York, NY, USA, 1991.

[2] M. Rajh, S. Glodež, J. Flašker, K. Gotlih, and T. Kostanjevec, "Design and analysis of an fMRI compatible haptic robot," *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 2, pp. 267–275, 2011.

[3] S. Awtar and G. Parmar, "Design of a large range XY nanopositioning system," *Journal of Mechanisms and Robotics*, vol. 5, no. 2, Article ID 021008, 2013.

[4] R. S. Ball, *A Treatise on the Theory of Screws*, Cambridge University Press, Cambridge, UK, 1998.

[5] K. H. Hunt, *Kinematic Geometry of Mechanisms*, Clarendon Press, Oxford, UK, 1978.

[6] K. J. Waldron, "The constraint analysis of mechanisms," *Journal of Mechanisms*, vol. 1, no. 2, pp. 101–114, 1966.

[7] R. Konkar, *Incremental kinematic analysis and symbolic synthesis of mechanisms [Ph.D. dissertation]*, Stanford University, Stanford, Calif, USA, 1993.

[8] J.-S. Zhao, Z.-J. Feng, and J.-X. Dong, "Computation of the configuration degree of freedom of a spatial parallel mechanism by using reciprocal screw theory," *Mechanism and Machine Theory*, vol. 41, no. 12, pp. 1486–1504, 2006.

[9] J. Eddie Baker, "Screw system algebra applied to special linkage configurations," *Mechanism and Machine Theory*, vol. 15, no. 4, pp. 255–265, 1980.

[10] M. T. Mason and J. K. Salisbury, *Robot Hands and the Mechanics of Manipulation*, MIT Press, Cambridge, Mass, USA, 1985.

[11] Y. Fei and X. Zhao, "An assembly process modeling and analysis for robotic multiple peg-in-hole," *Journal of Intelligent and Robotic Systems*, vol. 36, no. 2, pp. 175–189, 2003.

[12] B. Lee and K. Saitou, “Three-dimensional assembly synthesis for robust dimensional integrity based on screw theory,” *Journal of Mechanical Design*, vol. 128, no. 1, pp. 243–251, 2006.

[13] W. Kim, A. P. Veloso, D. Araújo, V. Vleck, and F. João, “An informational framework to predict reaction of constraints using a reciprocally connected knee model,” *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 18, no. 1, pp. 78–89, 2015.

[14] R. Konkar and M. Cutkosky, “Incremental kinematic analysis of mechanisms,” *Journal of Mechanical Design, Transactions of the ASME*, vol. 117, no. 4, pp. 589–596, 1995.

[15] J. D. Adams and D. E. Whitney, “Application of screw theory to constraint analysis of mechanical assemblies joined by features,” *Journal of Mechanical Design, Transactions of the ASME*, vol. 123, no. 1, pp. 26–32, 2001.

[16] J. D. Adams and S. Gerbino, “Application of screw theory to motion analysis of assemblies of rigid parts,” in *Proceedings of the IEEE International Symposium on Assembly and Task Planning (ISATP ’99)*, pp. 75–80, Porto, Portugal, July 1999.

[17] J. D. Adams, *Feature-based analysis of selective limited motion in assemblies [M.S. thesis]*, Mechanical Engineering Department, Massachusetts Institute of Technology, Cambridge, Mass, USA, 1998.

[18] S. Gerbino and F. Arrichiello, “How to investigate constraints and motions in assemblies by Screw Theory,” in *Proceedings of the CIRP International Seminar on Intelligent Computation in Manufacturing Engineering (CIRP ICME ’04)*, pp. 331–336, Sorrento, Italy, June-July 2004.

[19] P. Franciosa and S. Gerbino, “A CAD-based methodology for motion and constraint analysis according to screw theory,” in *Proceedings of the ASME International Mechanical Engineering Congress and Exposition (IMECE ’09)*, Paper no. IMECE2009-13159, pp. 287–296, Lake Buena Vista, Fla, USA, November 2009.

[20] M. Celentano, *Development of a CAD-based graphic environment for motion and constraint analysis according to Screw Theory [M.S. thesis in Mechanical Engineering]*, University of Naples—Federico II, Naples, Italy, 2007.

[21] H. J. Su, D. V. Dorozhkin, and J. M. Vance, “A screw theory approach for the conceptual design of flexible joints for compliant mechanisms,” *Journal of Mechanisms and Robotics*, vol. 1, no. 4, Article ID 041009, pp. 1–8, 2009.

[22] A. Dixon and J. J. Shah, “Assembly feature tutor and recognition algorithms based on mating face Pairs,” *Computer-Aided Design and Applications*, vol. 7, no. 3, pp. 319–333, 2010.

[23] D. E. Whitney, R. Mantripragada, J. D. Adams, and S. J. Rhee, “Designing assemblies,” *Research in Engineering Design*, vol. 11, no. 4, pp. 229–253, 1999.

[24] D. K. Smith, *Constraint analysis of assemblies using screw theory and tolerance sensitivities [Master of Science]*, Department of Mechanical Engineering, Brigham Young University, 2001.

[25] G. Shukla and D. E. Whitney, “The path method for analyzing mobility and constraint of mechanisms and assemblies,” *IEEE Transactions on Automation Science and Engineering*, vol. 2, no. 2, pp. 184–192, 2005.

[26] L. Rusli, A. Luscher, and J. Schmiedeler, “Analysis of constraint configurations in mechanical assembly via screw theory,” *Journal of Mechanical Design*, vol. 134, no. 2, Article ID 021006, pp. 1–12, 2012.

[27] L. Rusli, A. Luscher, and J. Schmiedeler, “Optimization of constraint location, orientation, and quantity in mechanical assembly,” *Journal of Mechanical Design*, vol. 135, no. 7, Article ID 071007, pp. 1–11, 2013.

[28] H. J. Su, L. Zhou, and Y. Zhang, “Mobility analysis and type synthesis with screw theory: from rigid body linkages to compliant mechanisms,” *Advances in Mechanisms, Robotics and Design Education and Research*, vol. 14, pp. 67–81, 2013.

[29] H.-J. Su and H. Tari, “On line screw systems and their application to flexure synthesis,” *Journal of Mechanisms and Robotics*, vol. 3, no. 1, Article ID 011009, 2011.

[30] D. C. Anderson and T. C. Chang, “Geometric reasoning in feature-based design and process planning,” *Computers and Graphics*, vol. 14, no. 2, pp. 225–235, 1990.

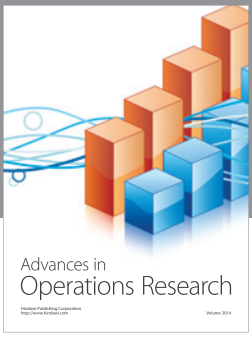
[31] J. Hefferon, *Linear Algebra*, 2014, http://joshua.smcvt.edu/linearalgebra/.

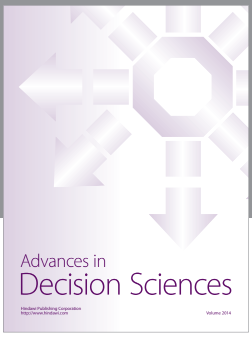
[32] J. Phillips, *Freedom in Machinery—Introducing Screw Theory*, vol. 1, Cambridge University Press, Cambridge, UK, 2007.

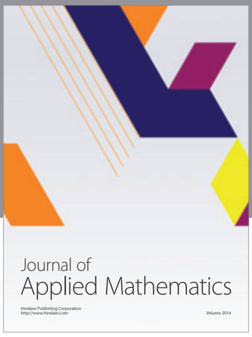
[33] Z. Huang, Q. Li, and H. Ding, “Basics of screw theory,” in *Theory of Parallel Mechanisms*, vol. 6, pp. 1–16, Springer, Dordrecht, The Netherlands, 2013.

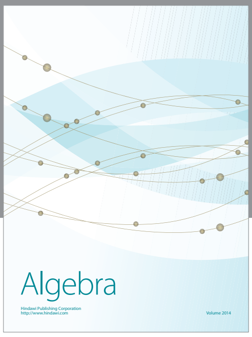
[34] F. Dunn and I. Parberry, *3D Math Primer for Graphics and Game Development*, CRC Press, Boca Raton, Fla, USA, 2nd edition, 2011.

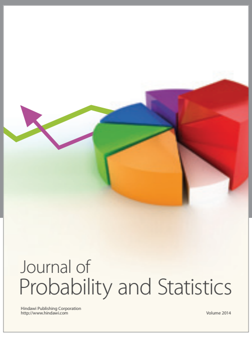
[35] E. J. Haug, *Computer Aided Kinematics and Dynamics of Mechanical Systems*, vol. 1 of *Allyn and Bacon Series in Engineering*, Basic Method, Boston, Mass, USA, 1989.


Advances in
Operations Research

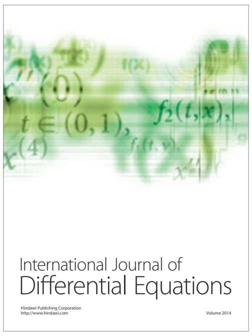

Advances in
Decision Sciences


Journal of
Applied Mathematics

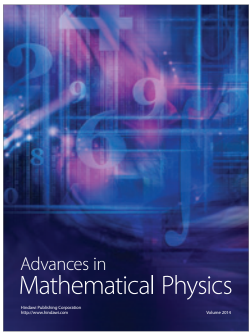

Algebra

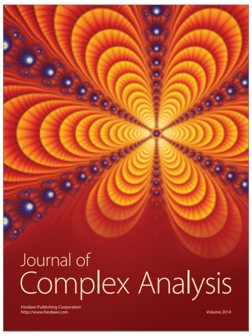

Journal of
Probability and Statistics

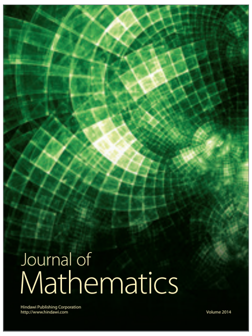
The Scientific
World Journal

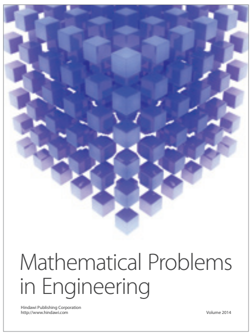

International Journal of
Differential Equations

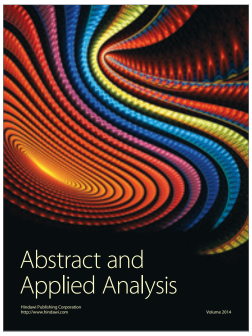
International Journal of
Combinatorics

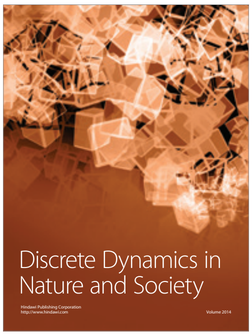

Advances in
Mathematical Physics

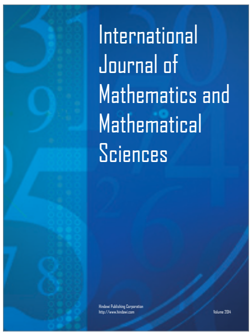

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

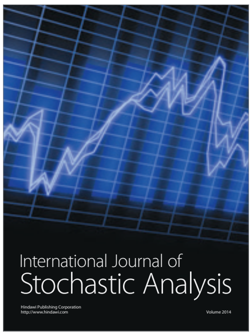

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization