Screw Motions for the Animation and Analysis of Mechanical Assemblies*

Jay J. KIM** and Jaroslaw R. ROSSIGNAC**

CAD systems offer a variety of techniques for designing and rendering models of static 3D objects and even of mechanisms, but relatively few tools exist for interactively specifying arbitrary movements of rigid bodies through space. Such tools are essential, not only for artistic animation, but also, for planning and demonstrating assembly and disassembly procedure of manufactured products. Although, any two positions may be interpolated by an infinity of motions, we chose the screw motion for its relative generality and its computational advantages. The paper covers original algorithms for computing the screw motions from interpolated positions and the envelopes of swept regions generated during the motions. We will also discuss why the formulation of a rigid motion in terms of a sequence of screw motions is more advantageous than other more general motion formulations for computing envelopes of swept regions and for predicting collisions.

Key Words: Animation, Screw Motions, Envelopes of Swept Regions, Collisions, Assemblies

1. Introduction

A rigid body motion is a continuous mapping from the time domain to a set of positions. To relieve the designers from the burden of specifying this mapping in abstract mathematical terms, combinations of simple rigid body motion primitives, such as linear translations or constant axis rotations, are often used. These simple motions are planar and thus ill-suited for approximating arbitrary motions in 3D-space. Instead, we use the screw motion primitive, a special combination of linear translations and constant axis rotations, which has a simple geometric representation that can be automatically and unambiguously computed from the starting and ending positions of the moving body. Since interactive graphic techniques exist for positioning solids in

space, they provide an intuitive and easy to use method for designing screw motions that interpolate a sequence of user-specified positions.

Screw motion primitives considerably extend the commonly used rotations and translations, but retain their representational conciseness and specific geometric properties. Each piecewise-screw motion is specified in terms of a sequence of key positions of the object that it interpolates at user controlled time values. The resulting piecewise-screw motion is unique and coordinate system independent. Also, the motion is independent of the manner in which the key positions were specified. Furthermore, when the piecewise-screw formulation is used, only the final positions of the object at selected time values affect its motion. Such flexibility is important for interactive applications⁽⁸⁾. Yet it is not supported by many popular motion modelers, where each key position is characterized by three Euler angles and three translation coordinates and where the motions are derived by interpolating each one of these six values independently. With screw motion editing facility users can concentrate on what they see and where the objects are as time is varied; they do not have to worry about

* Received 17th January, 2000

** School of Mechanical Engineering, Hanyang University, Seoul, Korea. E-mail: jaykim@email.hanyang.ac.kr

*** College of Computing, Room 241, Georgia Institute of Technology, 801 Atlantic Drive, NW, Atlanta, GA 30332-0280, USA

local coordinate systems or about the sequence of rotations and translations used to position the object in its key positions.

A motion is an extension of R^3 into the four-dimensional time-space domain. Thus, a moving object may be represented as a hypersolid in 4D. The projection of this hypersolid into the 3D space orthogonal to the time axis is the 3D region swept by the moving objects. Some CAD applications to robotics or NC machining use the boundary of the region swept by moving solids. Furthermore, visualizing the swept region provides important feedback during motion analysis. The boundary of the swept region is a subset of a surface swept by the silhouette edges and characteristic curves (precisely defined later) of the moving object. However, for an arbitrary motion, silhouette edges and characteristic curves change with time. For screw motions they are fixed on the object. Thus, the generation of the swept region for moving objects is considerably simplified if motions are restricted to be piecewise screws.

The rest of the paper is organized as follow. The second section begins with the brief review of the screw motion and describes the computation of screw parameters from two interpolated positions. The third section addresses the computation of swept region by an object in a screw motion. The last section shows an example with the implementation of an experimental program for designing and visualizing motions of assembly components.

2. Screw Motions

2.1 Background

The concept of a screw motion was developed in the 18th - 19th centuries and later extensively elaborated by Ball⁽²⁾ for applications to the kinematic and dynamic analysis. Recently, the screw theory has been rejuvenated in the context of spatial mechanism and robots^{(4),(14)} and also been applied to an automatic fixture planning system⁽³⁾.

A screw motion is a special combination of a rotation and a translation performed along the same axis, the screw axis, so that during the motion, the ratio between rotation and translation speed is constant. The characteristic of the screw motion is that the trajectory of any point on the object in the motion is a helix and the velocity vector of the point is constant with respect to the object's (moving) body coordinate system.

A screw axis can be specified with a unit vector $\mathbf{s}$, and an arbitrary point $\mathbf{p}_s$ on the axis. Therefore, additionally, if we specify a translational distance d and a rotational angle ϕ , a piecewise screw motion is completely defined with four screw parameters $[\mathbf{s}, \mathbf{p}_s,$

$d, \phi]$. Note that pure translation and rotation are considered as the special cases of screw motions and simply represented $\phi=0$ and $d=0$ respectively.

2.2 Computing screw parameters

In the area of kinematics and robotics, several methods have been developed for computing screw parameters from displacement matrices^{(5),(12)}. A variation of these methods was used in computer graphics and animation^{(7),(11)} to convert 3×3 rotation matrices to quaternion. The methods described in^{(5),(11),(12)} use the elements of a displacement matrix to compute screw parameters and thus must perform the matrix inversion or unnecessary calculations. Here, we propose a different approach, which derives the screw parameters directly from the position matrices. This is based on geometric interpolations which considerably reduce the treatment of special cases.

Now let us derive the screw parameters $[\mathbf{s}, \mathbf{p}_s, d, \phi]$ for given initial and final positions $\mathbf{P}_1(\mathbf{i}_1, \mathbf{j}_1, \mathbf{k}_1, \mathbf{o}_1)$ and $\mathbf{P}_2(\mathbf{i}_2, \mathbf{j}_2, \mathbf{k}_2, \mathbf{o}_2)$ shown in Fig. 1. The first step is to compute the screw axis $\mathbf{s}$. Consider the variations of the orientation shown in Fig. 2 and the variation of the translation. These variations are defined as,

Fig. 1 Initial and final positions of the moving body

Fig. 2 Variations of the orientations

Fig. 3 The end point of the vector i_1 moving around a circle with radius $\|s \times i_1\|$

$$\begin{bmatrix} \Delta i & \Delta j & \Delta k & \Delta o \end{bmatrix} = \begin{bmatrix} i_2 - i_1 & j_2 - j_1 & k_2 - k_1 & o_2 - o_1 \end{bmatrix} \quad (1)$$

The cross products of the variations of the orientation are:

$$c_{ij} = \Delta i \times \Delta j, \quad c_{jk} = \Delta j \times \Delta k, \quad c_{ki} = \Delta k \times \Delta i \quad (2)$$

Since around the screw axis these variations are occurred, the screw axis s has the same direction with one of the non-zero cross products of the variations (for the sake of simple discussion throughout the rest of this paper we assume that the screw motion is not a pure translation). If there is any rotation, at least, one of the cross products in Eq. (2) is a non-zero. Therefore, simply to choose it from Eq. (2) will give, for example, suppose c_{ij} is a non-zero vector, the screw axis as

$$s = \frac{c_{ij}}{\|c_{ij}\|} \quad (3)$$

The next step is to compute a minimal rotation angle ϕ . Again suppose c_{ij} is a non-zero vector. Then, Fig. 3 shows that the end of i_1 moves on the circle with radius $\|s \times i_1\|$. From the geometrical relationship provided by Fig. 3, ϕ can be evaluated as

$$\|\phi\| = 2 \arcsin \left(\frac{\|\Delta i\|}{2\|s \times i_1\|} \right) \quad 0 < \|\phi\| \leq \pi \quad (4)$$

where the sign of ϕ is plus if $(s \times i_1) \cdot \Delta i$ is greater than zero, otherwise minus.

After computing s and ϕ , to compute a fixed point p_s , let us first define a plane with normal vector s and passing through the origin of the reference coordinate frame $O-XYZ$ (see Fig. 4(a)). Then in the direction of s , project o_1 and o_2 onto the plane. From the geometry of Fig. 4(a), these projected points can be obtained as

$$q_1 = o_1 - (o_1 \cdot s)s \quad q_2 = o_2 - (o_2 \cdot s)s \quad (5)$$

Refer to Fig. 4(b), a view of Fig. 4(a) in the direction of the screw axis s . Since p_s is an arbitrary point on the screw axis, we simply take it as the center point of a circular arc passing from q_1 to q_2 , and with subtending angle of ϕ . Then from the geometrical relationship, we obtain

(a) The projection of o_1 and o_2 on the plane with the normal s and passing through the origin O

(b) View of Fig. 5(a) from the direction s

Fig. 4 Computing a fixed point on the screw axis

$$p_s = \frac{q_1 + q_2}{2} + \frac{s \times (q_2 - q_1)}{2 \tan(\phi/2)} \quad (6)$$

The last step is to compute translational distance d which can be simply evaluated from the dot product of Δo and s as follows:

$$d = \Delta o \cdot s \quad (7)$$

It should be emphasized that all the previous steps for computing screw parameters can be also applied to a displacement matrix:

$$D_{2/1} = P_1^{-1} P_2 = \begin{bmatrix} i_{2/1} & j_{2/1} & k_{2/1} & o_{2/1} \end{bmatrix} \quad (8)$$

In this case, the variations are defined as follows:

$$\begin{bmatrix} \Delta i & \Delta j & \Delta k & \Delta o \end{bmatrix} = \begin{bmatrix} i_{2/1} - i & j_{2/1} - j & k_{2/1} - k & o_{2/1} \end{bmatrix} \quad (9)$$

Note that the screw parameters computed from these variations are defined with respect to the body coordinate system $o_1-x_1y_1z_1$.

Figure 5(a) shows the trajectory of a moving object in a screw motion from initial to final positions. It displays a trajectory that we can more or less predict. For the same initial and final positions, Fig. 5(b) shows the trajectory of a motion computed with linear interpolations. In contrast to Fig. 5(a), the trajectory looks rather arbitrary because of the order of three Euler angles applied for rotations.

3. Envelopes

3.1 Background

One of the effective techniques to detect interference between moving objects is to consider the envelope or the swept volume generated by the objects.

(a) Screw motion

(b) Linear interpolation

Fig. 5 Screw motion versus linear interpolation

This technique has been also applied to the simulation of NC machining process and geometric modeling^{(6),(10),(13)} or similarly blending and offsetting operation for solid models^{(1),(9)}. In this paper, the boundary of swept volume is defined as the subset of the envelope. Self-intersection can be occurred for envelopes but not for the boundary of swept volume.

If a family of surfaces of an object can be expressed with an implicit function $f(x, y, z, t)=0$, where t is the family parameter such as time, then the theory of envelopes offers the mathematical method to compute the envelope of the object by solving the system of equations;

$$\begin{aligned} f(x, y, z, t) &= 0 \\ \frac{\partial f}{\partial t}(x, y, z, t) &= 0 \end{aligned} \quad (10)$$

However the application of the theory of envelope is quite limited because the system of equations can be only solved for the smooth surfaces represented with analytic functions such as cylindrical or spherical surfaces but not for the planar polyhedra.

In general, the envelope of an object undergoing any motion can be obtained by considering the boundary of an object only. For example, the envelope of planar polyhedra can be obtained by considering only the faces of the object. The technique we have developed to generate envelopes can be characterized as a procedural approach: each edge and face of an object are visited and to be determined if their subsets

Fig. 6 Intersection line and a characteristic curve break point

becomes an envelope. The subset of a boundary generating an envelope is defined as *envelope generators* or simply *generators*. If the generators come from faces or edges, they are defined as *characteristic curves* or *silhouette edges* respectively. Generators are solely computed from the velocity of a point on the moving object and the geometry of the object itself. For example, the velocity of any point on a generator is always orthogonal to the normal vector of the face, or tangent to the face, from where the point is taken.

For a motion whose velocity is not constant, the generators of an object is also changed dynamically with time. However for a piecewise screw motion, it is very attractive that the generators are fixed on the object during its motion because the velocity is constant with respect to its body coordinate system. Based on this observation and the velocity of generator's orthogonal condition, the following sections present the algorithm to compute generators of polyhedra in screw motion.

3.2 Characteristic curve and its break point

Characteristic curves are envelope generators composed of sets of points extracted from the faces of a moving object. In the case of polyhedra, characteristic curves have the shape of the straight lines segmented by particular points on the faces and by the edges of the faces. In this section we will focus on how to compute the straight lines and the particular points defined as *characteristic curve break points*. In the next section, we will discuss about which edges of the faces segment the straight line and at where. It should be noted that for mathematical simplicity, all the geometry in this and next sections are defined with respect to the body coordinate frame of which origin is located at p_s and the z -axis is parallel to the screw axis s .

Consider a planar face f_i , one of the faces of a moving polyhedron and having a normal vector n_i in 3D space. Then the equation of the plane p_i of the planar face f_i is expressed as

$$(xi + yj + zk) \cdot (n_{ix}i + n_{iy}j + n_{iz}k) = d_i \quad (11)$$

where $\mathbf{i}, \mathbf{j}$ and $\mathbf{k}$ are the unit vectors indicating the direction of the body coordinate axes x, y and z respectively and d_1 is the distance from the origin to the plane (see Fig. 6: note that for simplicity Fig. 6 does not show the face f_1 but just the plane p_1).

Utilizing the generator's orthogonal condition, first write the velocity of an arbitrary point $\mathbf{p}(x, y, z)$ on the plane p_1 undergoing screw motion $[\mathbf{s}, \mathbf{p}_s, d, \phi]$:

$$\mathbf{v}_p = (\mathbf{k} \times \mathbf{p} \times \rho \mathbf{k}) \frac{d\phi}{dt} \quad \text{or} \quad \mathbf{v}_p = (-y\mathbf{i} + x\mathbf{j} + \rho \mathbf{k}) \frac{d\phi}{dt} \quad (12)$$

where $\rho = d/\phi$, defined as a pitch, and where t is time variable. Then, combining Eq.(12) with the orthogonal condition results in

$$\mathbf{v}_p \cdot \mathbf{n}_1 = 0 \quad \text{or} \quad (-y\mathbf{i} + x\mathbf{j} + \rho \mathbf{k}) \cdot \frac{d\phi}{dt} \cdot (n_{1x}\mathbf{i} + n_{1y}\mathbf{j} + n_{1z}\mathbf{k}) = 0 \quad (13)$$

Since we have assumed that there is a rotation in screw motion, we can cancel $d\phi/dt$ and rewrite the orthogonal condition as

$$(x\mathbf{i} + y\mathbf{j} + \rho \mathbf{k}) \cdot \frac{(-n_{1y}\mathbf{i} + n_{1x}\mathbf{j})}{\sqrt{(n_{1x}^2 + n_{1y}^2)}} = \rho \frac{n_{1z}}{\sqrt{n_{1x}^2 + n_{1y}^2}} \quad \text{or} \quad (x\mathbf{i} + y\mathbf{j} + \rho \mathbf{k}) \cdot \mathbf{n}_v = d_v \quad (14)$$

which has the form of a plane equation. Let us call this plane p_v . The necessary condition for the point $\mathbf{p}$ to be on the characteristic curve is that the point must satisfy Eqs.(11) and (14) simultaneously, which means that the characteristic curve is a subset of the infinite length of a straight line created by the intersection of the two planes p_1 and p_v . Eq.(14) shows the case that the plane p_v cannot be defined when the magnitude of the vector $\mathbf{k} \times \mathbf{n}_1$ or $\sqrt{n_{1x}^2 + n_{1y}^2}$ is zero: the screw axis is normal to the face f_1 . In this case, the intersection line does not exist, which means there is no characteristic curve on the face f_1 .

The infinite length of the intersection line is divided into two zones by the characteristic curve break point $\mathbf{b}_c$: among two, one becomes an invalid zone because it stays inside the swept volume as shown in Fig. 7. $\mathbf{b}_c$ can be computed from the following two important geometrical properties of the plane p_v interpreted from Eq.(14).

1) Because $\mathbf{k}$ component of $\mathbf{n}_v$ is zero and $\mathbf{n}_1 \cdot \mathbf{n}_v = 0$, p_v is parallel with z -axis (screw axis) and perpendicular to the face f_1 .

2) If d_v is positive, then p_v is located at the distance d_v from the origin in the direction of $\mathbf{k} \times \mathbf{n}_1$ ($= -n_{1y}\mathbf{i} + n_{1x}\mathbf{j}$). If d_v is minus, located in the opposite direction.

These properties imply that, $\mathbf{b}_c$ is the point nearest from the z -axis. Using the geometry shown in Fig. 6, we obtain

Fig. 7 A valid zone (indicated by a solid line) of the intersection line

$$\mathbf{b}_c = d_v \frac{\mathbf{k} \times \mathbf{n}_1}{\|\mathbf{k} \times \mathbf{n}_1\|} + \lambda \mathbf{k} \quad (15)$$

where $\lambda \mathbf{k}$ is the cut point of the z -axis by the plane p_1 . If we set $x=0$ and $y=0$ in Eq.(11), we obtain $\lambda = d/n_{1z}$. From Eq.(15), we can see that, when $\mathbf{k} \times \mathbf{n}_1$ or $\sqrt{n_{1x}^2 + n_{1y}^2}$ is zero, the break point is not defined. Also, when $n_{1z}=0$: the screw axis is parallel with the face f_1 , the break point is not defined. In this case the entire length of the intersection line could be temporarily accepted as a valid zone and will be segmented or discarded during the computation of silhouette edges.

Figure 7 shows which portions of the intersection line divided by the break point becomes the valid zone that stays outside the swept volume: if, $\|\mathbf{k} \times \mathbf{n}_1\| > 0$, then the upper portion of the intersection line is the valid zone and otherwise, the lower portion.

In this section we have introduced an approach to compute the valid zone of the intersection line. However the valid zone is still infinity in one direction and needs to be further segmented during the computation of silhouette edges.

3.3 Silhouette edge and its break point

Silhouette edges are envelope generators composed of the subsets of the edges belonging to a moving object and can only come from concave edges because convex edges are submerged by a swept volume. In the case of polyhedra, same as characteristic curves, silhouette edges are also straight lines because the edges of the object are straight lines. If an edge is intersected by a characteristic curve, the edge is divided into two zones: among two, only one becomes a silhouette edge because the other zone is submerged by a swept volume. On the other hand, if an edge is not intersected by the characteristic curve, the whole or no portion of the edge becomes a silhouette edge for the same reason.

In order to compute the intersection between an edge and a characteristic curve, defined as a *silhouette edge break point*, refer to Fig. 8 (similar to Fig. 6 except that the face f_1 is displayed instead of the plane p_1). Let us pick the edge e_{12} and compute a silhouette

Fig. 8 Silhouette edge and its break point

Fig. 9 A 2D example of using the velocity vector and the face normal vectors to determine a silhouette edge

edge break point b_s . Using the vertices p_1 and p_2 of the edge e_{12} , the position vector of a break point can be expressed as

$$b_s = p_1 + \alpha(p - p_1) \quad (16)$$

where α is a parameter to be evaluated. Since b_s is also positioned on the intersection line, it must satisfy Eq.(13) as well. Therefore writing Eq.(13) in the forms of b_s gives

$$v_{bs} \times n_1 = 0 \quad \text{or} \quad (k \times b_s + \rho k) \cdot n_1 = 0 \quad (17)$$

Combining Eqs.(16) and (17) results in

$$\alpha = \frac{(k \times p_1 + \rho k) \cdot n_1}{(k \times (p_1 - p_2)) \cdot n_1} \quad (18)$$

In the case of $0 < \alpha < 1$, b_s exists on e_{12} and divides the edge into two, among which only one becomes a silhouette edge. In the case of $\alpha \geq 1$ or $\alpha \leq 0$, b_s does not exist on e_{12} . In this case the whole or no portion of the edge becomes silhouette edge. For either cases the following test will decide which portion of the edge becomes a silhouette edge. The test uses the unit normal vector n_2 of the face f_2 which is sharing the edge e_{12} with the face f_1 (see Fig. 8). The test is based on the fact that, for any portion to be a silhouette edge, the projections of n_1 and n_2 onto the velocity vector of an arbitrary point taken from the portion, must have an opposite direction each other (see Fig. 9). For example, let us pick the point p_1 as an arbitrary point on the edge. In the case of $(v_{p1} \cdot n_1)(v_{p1} \cdot n_2) \leq 0$, if $\alpha \geq 1$ or $\alpha \leq 0$, then the whole edge becomes a silhouette edge. In the same case, if

(a) Envelope of a swept volume for a cuboid undergoing a screw motion

(b) Characteristic curve (indicated by the arrow) and silhouette edges for the envelope of Fig. 10(a)

Fig. 10 An example of an envelope

$0 < \alpha < 1$, then the only portion to which p_1 belongs becomes a silhouette edge. Besides, in the case of $(v_{p1} \cdot n_1)(v_{p1} \cdot n_2) > 0$, the opposite is true.

It can be observed from Eq.(18) that there are two cases that b_s is not defined:

(1) The magnitude of $k \times (p_1 - p_2)$ is zero: the screw axis is parallel with e_{12} .

(2) $k \times (p_1 - p_2) \cdot n_1$ is zero: the screw axis is normal to the face f_1 .

For all the two cases, we can assume that $\alpha \geq 1$ or $\alpha \leq 0$ and can apply the same test described in Fig. 9 to decide if the whole portion of the edge becomes a silhouette edge or not.

So far we have explained an algorithm to compute a silhouette edge for one edge e_{12} of the face f_1 . Applying the same algorithm to all the edges of the face f_1 gives the set of the break points as shown in Fig. 8. If we create an infinite length of a line passing through any two points in the set, it is the same intersection line generated by the planes p_1 and p_v in Section 3.2.

Among the set of the break points lying on the intersection line as shown in Fig. 8, choose the points that passes the test shown in Fig. 7. Then, if we create a set of lines lying on the face by connecting these

Fig. 11 Assembly of a puzzle composed of six components. The xyz coordinate axes show key-frame position of each component. Helix curves display the assembly paths of each component undergoing screw motions

points, the set finally becomes the characteristic lines of the face f_1 , which is the last step of computing a characteristic line for one particular face.

Applying the algorithms explained in Section 3.2 and Section 3.3 to all the faces of a moving body will give a complete set of envelope generators. With this set we can generate an envelope, which is similar to the generation of ruled surfaces.

Figure 10 show an example of an envelope and the generators for a cuboid in motion. Note that each of the two edges in Fig. 10(b) is cut by the characteristic curve indicated by an arrow.

4. Conclusions

We have introduced an efficient algorithm to compute a screw motion from position matrices without matrix inversion. The algorithm takes a geometrical approach and therefore reduces the treatment of special cases significantly. We have also introduced a concise representation of region swept

by polyhedra in screw motion. It provides simple geometric constructions for computing the silhouette of the moving object, which we define as the set of points whose velocity is tangential to the object's boundary.

Figure 11 shows an assembly process for the puzzle composed of six components. The initial and final positions of each component are displayed in Fig. 11(a) and 11(d) respectively. The xyz coordinates axes in each figure shows the series of key-frame positions specified by a user with interactive position editing facility. The helix curves are the paths of each component in screw motion and are computed from two consecutive key-frame positions.

References

- (1) Abdel-Malek, K., Yeh, H.J. and Othaman, S., Swept Volumes: Void and Boundary Identification. *Computer-Aided Design*, Vol. 30, No. 13 (1987), pp. 1009-1018.

- (2) Ball, R., A Treatise on the Theory of Screws, (1900), Cambridge Univ. Press.
 - (3) Cutkosky, M. and Lee, S., Fixture Planning with Friction for Concurrent Product/Process Design, NSF Engineering Design Research Conference, UMASS, Amherst June, (1989).
 - (4) Ohwovoriole, M. and Roth, B., An Extension of Screw Theory, Transaction of ASME, Journal of Mechanical Design, Vol. 103 (1981), pp. 725-735.
 - (5) Paul, R., Robot Manipulators, (1982), MIT Press.
 - (6) Pegna, J., Variable Sweep Geometric Modeling, Ph.D Thesis, Stanford University, Stanford, CA, December, (1987).
 - (7) Pletinckx, D., Quaternion Calculus as a Basic Tools in Computer Graphics, The Visual Computer, Vol. 5 (1989), pp. 2-13.
 - (8) Roschel, O., Rational Motion Design — A Survey, Computer-Aided Design, Vol. 30, No. 3 (1998), pp. 169-178.
 - (9) Rossignac, J. and Requicha, A.A.G., Constant Radius Blending in Solid Modeling, Computers in Mechanical Engineering, July, (1984), pp. 65-73.
 - (10) Sambandan, K. and Wang, K.K., Five-axis Swept Volumes for Graphic NC Simulation and Verification, ASME Design Automation Conference at Montreal Canada, September 17-20, (1989).
 - (11) Shoemake, K., Animating Rotation with Quaternion Curves, Computer Graph (Proc. Siggraph 85), Vol. 19, No. 3 (1985), pp. 245-254.
 - (12) Suh, C. and Radcliffe, C., Kinematics & Mechanisms Design, (1978), John Wiley & Sons.
 - (13) Weld, J., Geometric Representation of Sweep Volumes with Application to Polyhedral Objects, Ph.D Thesis, Cornell University, Ithaca, New York, August (1987).
 - (14) Yang, A. and Freudenstein, F., Application of Dual-Number Quaternion Algebra to the Analysis of Spatial Mechanisms, Transaction of ASME Journal of Applied Mechanics, (1964).
-