

## DETC04/MECH-57473

### LINE BASED COLLISION DETECTION OF CYLINDRICAL RIGID BODIES

**John S. Ketchel**

Robotics and Spatial Systems Laboratory  
Mechanical and Aerospace Engineering Department  
Florida Institute of Technology  
Melbourne, Florida 32901  
Email: jketchel@fit.edu

**Pierre M. Larochelle\***

Robotics and Spatial Systems Laboratory  
Mechanical and Aerospace Engineering Department  
Florida Institute of Technology  
Melbourne, Florida 32901  
Email: pierrel@fit.edu

#### ABSTRACT

This paper presents a novel methodology for detecting collisions of cylindrically shaped rigid bodies moving in three dimensions. This algorithm uses line geometry and dual number algebra to exploit the geometry of cylindrical objects to facilitate the detection of collisions. First, the rigid bodies are modelled with infinite cylinders and a necessary condition for collision is evaluated. If the necessary condition is not satisfied then the two bodies do not collide. If the necessary condition is satisfied then a collision between the bodies may occur and we proceed to the next stage of the algorithm. In the second stage the bodies are modelled with finite cylinders and a definitive necessary and sufficient collision detection algorithm is employed. The result is a straight-forward and efficient means of detecting collisions of cylindrically shaped bodies moving in three dimensions. This methodology has applications in spatial mechanism design, robot motion planning, and workspace analyses of parallel kinematic machines such as Stewart-Gough platforms. A case study examining a spatial 4C mechanism for self collisions is included.

#### 1 INTRODUCTION

In this paper we present an algorithm for determining quantitatively if two bodies moving in three dimensional space collide. The methodology presented consists of two stages. In the first, infinite length cylinders are used to model the objects, then line geometry is used to determine if the cylinders intersect. If these

infinite cylinders do not intersect then the two bodies do not collide and no further testing is required. If the two infinite cylinders do intersect then further testing is necessary. We proceed to the second stage where cylinders of finite length are used to model the objects and they are tested to determine quantitatively if they collide.

Collision detection is vital for real world implementation of three dimensional mechanical systems such as robots, mechanisms, parallel kinematic machines, and linkages. Collision detection assists in motion planning, digital prototyping and motion simulation of the system. For motion planning applications collision detection can be used to verify that the planned motion of the system is collision-free with respect to the working environment and self-collisions. The methodology presented here enables the user to model the system and determine *without risking hardware* if there is a possible collision. Three dimensional mechanical systems can be difficult and expensive to develop hence such modelling and testing of the system in the early stages of design may save time and reduce costs. Often, complex systems are digitally prototyped and simulated. These simulations are improved by including motion planning and collision detection.

The methodology presented here is general and can be used to detect collisions between any rigid bodies moving in three dimensions provided that the bodies are predominantly cylindrical in shape. We focus upon such bodies because this shape is commonly found in industrial robots, parallel kinematic machines (e.g. Stewart-Gough platforms), and spatial mechanisms. Our primary motivation for this work comes from our efforts to

---

\*Address all correspondence to this author.

advance the state of the art in spatial mechanism design. Spatial mechanisms are closed kinematic chains consisting of rigid links connected by cylindrical(C), revolute(R), or prismatic(P) joints. Traditionally, the links of these mechanisms are cylindrical in shape. Recently, there have been some significant efforts made to address the challenge of designing useful spatial mechanisms. In [Larochelle, 1998] a Burmester Theory based computer-aided design program for spatial 4C mechanisms was reported. Efforts were made to address circuit and branch defects in [Larochelle, 2000]. Approximate motion synthesis was addressed in [Larochelle, 1994] and [Dees, 2001]. The exploration of utilizing virtual reality techniques to address the inherent visualization and interaction challenges was reported in [Kihonge et al, 2002].

The goal here was to facilitate the design of spatial 4C mechanisms by assisting in the selection of a mechanism that is free of collisions, including self-collisions. For collision detection, cylinders are used to model all objects that need to be analyzed. Cylinders are useful modelling tools since most three dimensional mechanical systems consist of some combination of prismatic, revolute, and cylindrical joints. With respect to the specific case of spatial 4C mechanisms, traditionally the links are cylindrical in shape. For the first stage of testing, infinite cylinders are used to model the links. This allows us to employ line geometry to yield a fast and efficient means of determining if a collision is possible. If the infinite cylinders do intersect then the actual finite links may in fact collide. Therefore, we proceed to the second stage of the collision detection algorithm where finite cylinders are used to model the links. Then these cylinders are tested for possible collisions.

The paper proceeds as follows. First, the distance calculations between infinite cylinders then finite length cylinders are presented. The necessary kinematic analysis of the spatial 4C mechanism are performed. Next, utilizing the distance calculations and the results of the spatial 4C analysis, we determine if a collision occurs for a spatial 4C mechanism. Finally, a case study for the self collision detection of a spatial 4C mechanism is presented.

## 1.1 Related Works

Zsombor-Murray [Zsombor-Murray, 1992] presents the visualization of the shortest distance between two lines in space. His constructive geometry and algebraic solutions to the problem motivated the work reported here. [Xavier, 2000] correctly states that failure to detect a collision is less acceptable than false positives, which can be further checked and that for the sake of speed exact or accurate collision detection is often sacrificed. Collisions are unacceptable and therefore being able to detect and avoid them is of vital interest. A great amount of work has been done on the collision detection problem. Many models and methods have been proposed to solve this important problem from

simple two dimensional models [Wu, Mayne, 2000] to highly complex models such as [Johnson, Vance, 2001], [Patoglu, Gillespie, 2000] and [Xavier, 2000]. These general methods are computationally intensive when compared to the algorithm presented here.

Although spatial 4C mechanisms are capable of spatial motion, i.e. motion in three dimensions, their motion is constrained to a complex three dimensional surface so traditional methods of path planning (e.g. [Belta, Kumar, 2000]) do not apply. Similarly, singular configurations can be easily identified during the testing phase and can be avoided during implementation [Mi, Yang, Abdel-Malek, Jay, 2002]. Path verification [Stocco, 2001] and reachable space methods [Fuhrmann, Schomer, 2001] do not apply for the self collision problem since the envelope of the mechanism does not take into account self collisions. There are several methods available to calculate the data necessary for the two step analysis that is proposed in this paper. The relatively low number of points and vectors necessary used do not warrant using vector bundles [Serre, Riviere, Duong, Ortuzar, 2001]. In future work, the use of dual vectors per [Fischer, 1994] for generating the necessary geometric data may be implemented.

## 2 Distance Calculation

### 2.1 Infinite Cylinder Testing

Here we use infinite and finite cylinders to model rigid bodies in three dimensions. Initially, each object is modelled by a cylinder of infinite length and finite radius. Infinite cylinders are simple models to check for collisions since they can be represented by a line with a radius. The shortest distance between two lines in space is along a line that intersects both lines and is also mutually orthogonal (common normal). An advantage of using cylinders is that their common normal line has a finite line segment between the two cylinders and by subtracting the two cylinders radii from the segment, possible collisions can be detected. If the distance between the two cylinders is less than the sum of their two radii then the infinite cylinders have collided. Hence, if the actual finite cylindrical objects have collided it is *necessary* that the minimum distance between their associated infinite cylinders is less than the sum of their radii.

The major axis of an infinite cylinder is a line. Here, we use Plücker coordinates and dual vectors to represent these lines in space. Plücker coordinates define a line by its unit directional vector and moment. Moreover, when convenient, we employ dual vector algebra to operate on lines. The Plücker coordinates of a line can be generated from two points on the line or from a point and direction vector (see Fig. 1). For example line  $S_1$  can be defined by points  $\vec{c}$  and  $\vec{f}$  or point  $\vec{c}$  direction vector  $\vec{s}$ . Likewise, line  $S_2$  can be defined by points  $\vec{d}$  and  $\vec{g}$  or point  $\vec{d}$  and direction vector  $\vec{w}$  (see eq. 1 and 2).

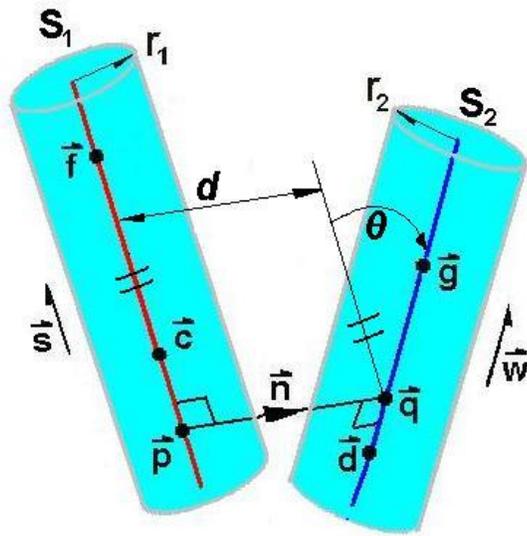


Figure 1. Infinite Cylinder Testing

$$S_1 = \left( \frac{\vec{f} - \vec{c}}{\|\vec{f} - \vec{c}\|}, \vec{c} \times \frac{\vec{f} - \vec{c}}{\|\vec{f} - \vec{c}\|} \right) \quad (1)$$

$$= (\vec{s}, \vec{c} \times \vec{s})$$

$$S_2 = \left( \frac{\vec{g} - \vec{d}}{\|\vec{g} - \vec{d}\|}, \vec{d} \times \frac{\vec{g} - \vec{d}}{\|\vec{g} - \vec{d}\|} \right) \quad (2)$$

$$= (\vec{w}, \vec{d} \times \vec{w})$$

We use the dual vector representation of the lines and dual vector algebra as follows [Fischer, 1999] and [McCarthy, 2000] where  $\epsilon^2=0$ .

$$\hat{S}_1 = (\vec{s}, \vec{c} \times \vec{s}) \quad (3)$$

$$= (a, a^0)$$

$$= a + \epsilon a^0$$

$$\hat{S}_2 = (\vec{w}, \vec{d} \times \vec{w}) \quad (4)$$

$$= (b, b^0)$$

$$= b + \epsilon b^0$$

Line dot product:

$$\hat{S}_1 \cdot \hat{S}_2 = (a, a^0) \cdot (b, b^0) \quad (5)$$

$$= (a \cdot b, a \cdot b^0 + b \cdot a^0)$$

$$= a \cdot b + \epsilon (a \cdot b^0 + b \cdot a^0)$$

$$= \cos \theta - \epsilon d \sin \theta$$

$$= \cos \hat{\theta}$$

Line cross product:

$$\hat{S}_1 \times \hat{S}_2 = (a, a^0) \times (b, b^0) \quad (6)$$

$$= (a \times b, a \times b^0 + a^0 \times b)$$

$$= a \times b + \epsilon (a \times b^0 + a^0 \times b)$$

$$= (\sin \theta + \epsilon d \cos \theta) \hat{N}$$

$$= \sin \hat{\theta} \hat{N}$$

where  $\hat{N}$  is the common normal line to  $\hat{S}_1$  and  $\hat{S}_2$ .

The above operations are important for calculating the distance  $d$  and the angle  $\theta$  between two lines. The resultant dual number of the dot product of two dual vectors yields the angle and distance between the two lines as long as they are not parallel to each other (see eq. 5). If the  $d \sin \theta$  term is not equal to zero, then the lines do not intersect ( $d \neq 0$ ) and are not parallel ( $\sin \theta \neq 0$ ). If  $d \sin \theta$  is equal to zero and  $\cos \theta$  term does not equal one, then the lines intersect ( $d=0$ ) and are not parallel.

If the  $\cos \theta$  term of the dot product is equal to 1 the vectors are parallel and another method of determining the distance between the two lines is necessary. If the lines are parallel, the resultant dual vector of the cross product of the lines yields the vector between the two lines (see eq. 6). If the  $d \cos(\theta)$  term is zero then the lines are identical and the distance between them is zero.

This is a fast method of determining the distance between the two lines and if a possible collision has occurred. By subtracting the radii of the two cylinders from the distance it can be determined if further testing of the cylinders is necessary. If the resulting distance is greater than the two radii then no collision is possible regardless of the length of the finite cylinders and the next cylinder pair can be tested. If the result is not greater than the two radii then a collision *may have* occurred and a finite cylinder model is used in the next stage of the collision detection algorithm.

## 2.2 Finite Cylinder Testing

If a possible collision has been detected by the infinite cylinder test then further testing is required to determine if an actual collision has occurred. The model is modified from cylinders of infinite length to finite length. This changes the base approach from testing lines to testing line segments. The same idea applies that the shortest distance between the lines is their common normal but the point where the common normal intersects the

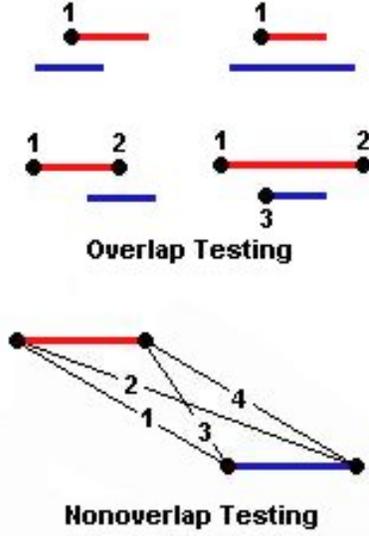


Figure 2. Parallel Testing

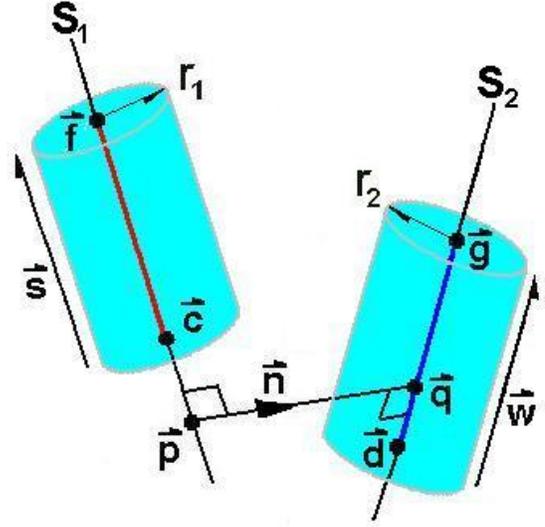


Figure 3. Finite Cylinder Testing

cylinders becomes important. Each intersection point lies either on the segment or off an end of the segment. If the point is off the segment then the nearest endpoint of the segment is used for testing since it is closest to the other segment. This is done for both segments and the distance between the two points is calculated and then compared to the two radii to determine if a collision has occurred.

From the initial testing it has already been determined if the lines are parallel. If they are parallel then there are two general cases possible: the segments overlap in some manner or there is no overlap. The equation for a plane can be determined that is perpendicular to the lines and passes through one endpoint utilizing planar geometry. The other line can then be tested to see at what point along its length it intersects the plane. This point needs to be tested to see if it is on the line segment or not. This is repeated for up to two other endpoints (a fourth endpoint being redundant) to test for overlapping (see Fig. 2). If no overlapping occurs of the parallel segments then the shortest distance will be between one of the four combinations of the two segments endpoints.

If the lines are not parallel then a hybrid Plücker based method is used. The line segments are described by an endpoint of the line segment and a *non*-unit direction vector (see Fig. 3). The common normal line  $N$  intersects the lines  $S_1$  and  $S_2$  at points  $\vec{p}$  and  $\vec{q}$  respectively. Parametric equations for points  $\vec{p}$  and  $\vec{q}$  of lines  $S_1$  and  $S_2$  can be calculated (see eq. 7).

$$\begin{aligned}\vec{p} &= \vec{c} + t_1 \vec{s} \\ \vec{q} &= \vec{d} + t_2 \vec{w}\end{aligned}\quad (7)$$

$$\begin{aligned}t_1 &= \frac{[(\vec{d} - \vec{c}) \times \vec{w}] \cdot \vec{n}}{\vec{n} \cdot \vec{n}} \\ t_2 &= \frac{[(\vec{d} - \vec{c}) \times \vec{s}] \cdot \vec{n}}{\vec{n} \cdot \vec{n}} \\ \vec{n} &= \vec{s} \times \vec{w}\end{aligned}$$

The points  $\vec{p}$  and  $\vec{q}$  are on the lines  $S_1$  and  $S_2$ . We must determine whether they are on the segments, before the segments or after the segments. The reason for not using unit directional vectors is now apparent. If  $t_1 \leq 0$  then  $\vec{p}$  lies at the start of the segment or earlier, so the start point is used to determine the distance. If  $t_1 \geq 1$  then  $\vec{p}$  lies at the end of the segment or later, so the end point is used. If  $0 < t_1 < 1$  then  $\vec{p}$  lies on the line segment so  $\vec{p}$  can be used to calculate the distance. The above holds true for  $S_2$ ,  $t_2$  and  $\vec{q}$ . This determines the two points to be used for the distance calculation presented above.

### 3 THE SPATIAL 4C MECHANISM

A spatial 4C mechanism has four cylindrical joints, each joint permitting relative rotation and translation along a line (see Fig. 4). The frame's axes are color coded red, green, and blue to correspond with the local XYZ axes. The link parameters that define the mechanism are listed in Tbl. 1 and the joint variables are defined in Tbl. 2.

The spatial 4C mechanism may be viewed as a combination of two CC dyads. The driving CC dyad has four independent joint variables, referred to as  $\theta$ ,  $d_1$ ,  $\phi$  and  $c_1$ . The driven dyad also has four independent joint variables,  $\psi$ ,  $d_2$ ,  $\delta$  and  $c_2$ . When

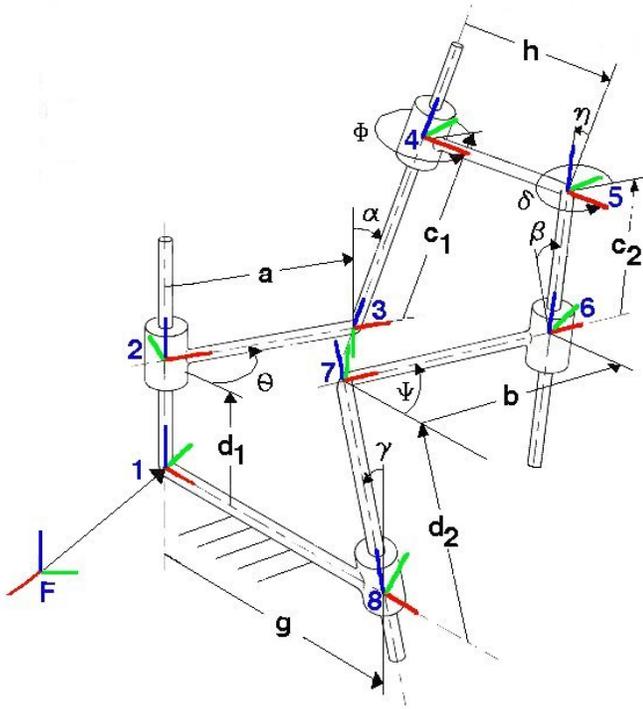


Figure 4. 4C Spatial Mechanism

Table 1. Common Normal & Link Parameters of the 4C Mechanism

Link	Dual Angle	Twist	Length
Driving	$\hat{\alpha}$	$\alpha$	$a$
Coupler	$\hat{\eta}$	$\eta$	$h$
Driven	$\hat{\beta}$	$\beta$	$b$
Fixed	$\hat{\gamma}$	$\gamma$	$g$

adjoined by the coupler link, the two dyads form a closed chain spatial 4C mechanism with two degrees of freedom. We chose  $\theta$  and  $d_1$  to be the independent joint variables. Note that  $\phi$  and  $c_1$  as well as the driven dyad's joint variables are now explicit functions of  $\theta$  and  $d_1$  and these functions are found below.

### 3.1 Spatial 4C Mechanism Analysis

We now present the equations that define the relative movement for the links of a spatial 4C mechanism given its physical dimensions and the input variables,  $\theta$  and  $d_1$ . The closed chain vector loop equations were solved to yield the following equations, [Larochelle, 1998] and [Duffy, 1980].

The coupler angle  $\phi$  is a function of the input angle  $\theta$  (see eq. 8).

Table 2. Moving Axes & Joint Variables of the 4C Mechanism

Joint Axis	Dual Angle	Rotation	Translation
Fixed	$\hat{\theta}$	$\theta$	$d_1$
Driving	$\hat{\phi}$	$\phi$	$c_1$
Coupler	$\hat{\delta}$	$\delta$	$c_2$
Driven	$\hat{\psi}$	$\psi$	$d_2$

$$\phi(\theta) = \arctan\left(\frac{B}{A}\right) \pm \arccos\left(\frac{C}{\sqrt{A^2 + B^2}}\right) \quad (8)$$

$$A = \sin(\eta) \sin(\gamma) \cos(\alpha) \cos(\theta) - \sin(\alpha) \sin(\eta) \cos(\gamma)$$

$$B = -\sin(\eta) \sin(\gamma) \sin(\theta)$$

$$C = \cos(\beta) - \cos(\eta) \sin(\alpha) \sin(\gamma) \cos(\theta) - \cos(\alpha) \cos(\eta) \cos(\gamma)$$

Note that  $\phi$  has two solutions corresponding to the two assemblies or circuits of the mechanism.

The output angle  $\psi$  is a function of the input angle  $\theta$  and the coupler angle  $\phi$  (see eq. 9).

$$\psi(\theta, \phi) = \arctan\left(\frac{B}{A}\right) \quad (9)$$

$$A = \frac{1}{-\sin(\beta)} \left\{ \cos(\eta) (\cos(\alpha) \sin(\gamma) - \cos(\gamma) \cos(\theta) \sin(\alpha)) - \sin(\eta) \cos(\phi) (\cos(\alpha) \cos(\gamma) \cos(\theta) + \sin(\alpha) \sin(\gamma)) + \sin(\eta) \cos(\gamma) \sin(\phi) \sin(\theta) \right\}$$

$$B = \frac{1}{\sin(\beta)} \left\{ \cos(\eta) \sin(\alpha) \sin(\theta) + \sin(\eta) \cos(\theta) \sin(\phi) + \sin(\eta) \cos(\alpha) \cos(\phi) \sin(\theta) \right\}$$

The output coupler angle  $\delta$ , i.e. the angle between the coupler and driven crank is a function of the input angle  $\theta$  and the output angle  $\psi$  (see eq. 10).

$$\delta(\theta, \psi) = \arctan\left(\frac{B}{A}\right) \quad (10)$$

$$A = \frac{1}{\sin(\eta)} \left\{ \cos(\alpha) (\cos(\gamma) \sin(\beta) + \sin(\gamma) \cos(\beta) \cos(\psi)) \right\}$$

$$\begin{aligned}
& \cos(\beta) \sin(\gamma) \cos(\psi) - \\
& \sin(\alpha) \cos(\theta) (\cos(\beta) \cos(\gamma) \cos(\psi) - \\
& \sin(\beta) \sin(\gamma)) - \\
& \sin(\alpha) \cos(\beta) \sin(\theta) \sin(\psi) \} \\
B = & \frac{1}{-\sin(\eta)} \left\{ \cos(\alpha) \sin(\gamma) \sin(\psi) + \right. \\
& \sin(\alpha) \sin(\theta) \cos(\psi) - \\
& \left. \sin(\alpha) \cos(\gamma) \cos(\theta) \sin(\psi) \right\}
\end{aligned}$$

The driving coupler translation  $c_1$ , the translation along the driving axis, is a function of  $\theta$ ,  $\psi$ ,  $\delta$  and the input translation  $d_1$  (see eq. 11).

$$\begin{aligned}
c_1(\theta, \psi, \delta, d_1) &= \frac{A}{B} \quad (11) \\
A &= d_1 \sin(\gamma) \sin(\psi) + a \cos(\theta) \cos(\psi) \\
&\quad + a \cos(\gamma) \sin(\theta) \sin(\psi) + h \cos(\delta) \\
&\quad - b - g \cos(\psi) \\
B &= \sin(\eta) \sin(\delta)
\end{aligned}$$

The driven coupler translation  $c_2$ , the translation along the driven axis, is a function of  $\theta$ ,  $\phi$ ,  $\psi$  and the driving coupler translation  $c_1$  (see eq. 12).

$$\begin{aligned}
c_2(\theta, \phi, \psi, c_1) &= \frac{A}{B} \quad (12) \\
A &= h \cos(\phi) \cos(\theta) + c_1 \sin(\alpha) \sin(\theta) + a \cos(\theta) \\
&\quad - h \cos(\alpha) \sin(\phi) \sin(\theta) - g - b \cos(\psi) \\
B &= \sin(\beta) \sin(\psi)
\end{aligned}$$

Finally, the translation along the driven axis  $d_2$  is a function of  $\theta$ ,  $\phi$ ,  $\psi$ , the driving coupler translation  $c_1$  and the driven coupler translation  $c_2$  (see eq. 13).

$$\begin{aligned}
d_2(\theta, \phi, \psi, c_1, c_2) &= \frac{A}{B} \quad (13) \\
A &= h \cos(\phi) \sin(\theta) - c_1 \cos(\theta) \sin(\alpha) \\
&\quad + a \sin(\theta) + h \cos(\alpha) \cos(\theta) \sin(\phi) \\
&\quad - b \cos(\gamma) \sin(\psi) + c_2 \cos(\beta) \sin(\gamma) \\
&\quad + c_2 \cos(\gamma) \cos(\psi) \sin(\beta) \\
B &= -\sin(\gamma)
\end{aligned}$$

## 4 Mechanism Collision Testing

### 4.1 Part 1: Analyzing Mechanism Via Points

Our implementation of the collision detection algorithm presented here requires a set of via points, the constant parameters

( $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\eta$ ,  $a$ ,  $b$ ,  $g$ ,  $h$ ) and the radii of each of the links of a spatial 4C mechanism. The set of via points contains the input values for  $\theta$ ,  $d_1$ , number of incremental steps to the next via point, and which solution to use for  $\phi$ . From this given information several tests can be performed to see if the mechanism is unsatisfactory.

Each via point's theta value can be tested to make sure that it is within the allowable motion range of the mechanism. The allowable motion range can be calculated using the link twist angles of the mechanism (see eq. 14) [Murray and Larochelle, 1998].

$$\begin{aligned}
C_1 &= \frac{\cos(\eta - \beta) - \cos(\alpha) \cos(\gamma)}{\sin(\alpha) \sin(\gamma)} \quad (14) \\
C_2 &= \frac{\cos(\eta + \beta) - \cos(\alpha) \cos(\gamma)}{\sin(\alpha) \sin(\gamma)} \\
&\quad -1 < C_1, C_2 < 1
\end{aligned}$$

This gives four possible cases of solutions sets: see Fig. 5.

Case 1. Neither  $C_1$  nor  $C_2$  are within the allowable range so the input link is capable of full rotation.

Case 2. Only  $\theta_1 = \arccos(C_1)$  exists, then the input link rocks across  $\pi$  from  $\pm\theta_1$ .

Case 3. Only  $\theta_2 = \arccos(C_2)$  exists, then the input link rocks across 0 from  $\pm\theta_2$ .

Case 4. Both  $\theta_1 = \arccos(C_1)$  and  $\theta_2 = \arccos(C_2)$  are within the allowable range. The input link can rock in two ranges, between  $\theta_1$  and  $\theta_2$  and  $-\theta_1$  and  $-\theta_2$  not passing through 0 or  $\pi$ .

Using the above equations the allowable ranges for  $\theta$  can be determined. If any of the via point's  $\theta$  values are not in the allowable range, the mechanism is not satisfactory.

In addition, each via point's  $d_1$  can be checked for a sign change or if it approaches zero. Currently, the spatial 4C mechanisms designed by SPADES and VRSpatial use the common normal to connect each link's collar to its axis. If the sign of  $d_1$  changes or if it nears zero a collision will occur between the driving link's collar and the fixed link's common normal.

Similarly, the  $\phi$  solution set can be inspected. The  $\phi$  solution set is passed in as part of the via points.  $\phi$  has two solution sets and the remaining calculations are based on only one of the sets for the mechanism to be acceptable. If the set changes, the mechanism changes circuits and/or moves through a singular configuration.

### 4.2 Part 2: Determining Maximum Length of each link's moving axis

The links of a spatial 4C mechanism can be modelled by a closed chain of eight line segments defined by twelve points (see

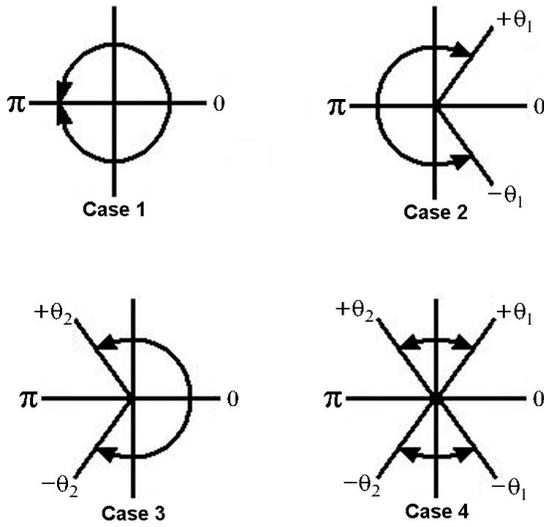


Figure 5. Input Theta Range

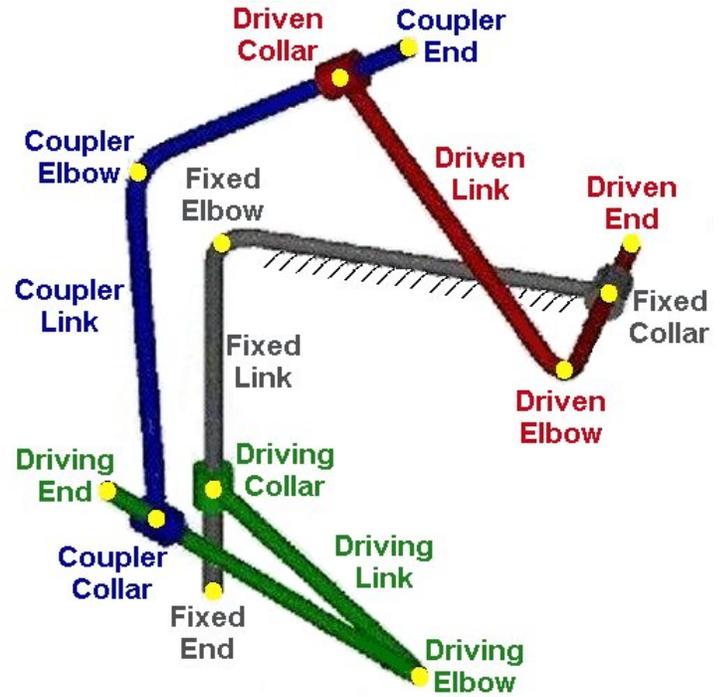


Figure 6. Point Designation

Fig. 6). Each of the four links are described by three points: one at the center of the link's *collar*, the *elbow* at the intersection of the link's common normal with its axis, and the *end* at the opposite end of its axis. Although each link's collar is co-linear with the previous link's axis, separate points are required for it and for the end of the moving axis.

We assume that each link axis is rigid and that its length will be sized accordingly. Hence, it is necessary to find the maximum length of each link's axis for the desired motion. To do this we use linear interpolation of the  $\theta$  and  $d_1$  via points to yield a discretized representation of the desired motion. At each discrete point we perform a kinematic analysis of the mechanism via Section 3. Finally, the minimum and maximum values of  $c_1$ ,  $d_2$ , and  $c_2$  are identified. These lengths are used to define the lengths of the finite cylinders that are used to model the link axes for collision detection.

### 4.3 Part 3: Infinite Line (Cylinder) Generation

The first step in testing the mechanism for a possible collision is generating the Plücker coordinates of its axes. We assign right-handed frames to the mechanism that translate and rotate along and about only the local X and Z axes, see Fig. 4). The frames are attached at the intersection of the link segments and are aligned such that either its X or Z axis is collinear with the link's direction vector. This kinematic analysis uses standard homogeneous transformations that are translations or rotations with respect to a single local axis (X or Z). Each homogeneous transformation contains the point on the line ( $\vec{p}$ ) and its direction vec-

tor (either  $\vec{X}$  or  $\vec{Z}$ ).

$${}^{i-1}T = \left[ \begin{array}{ccc|c} \vec{x} & \vec{y} & \vec{z} & \vec{p} \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (15)$$

The closed chain mechanism is modelled as two open kinematic chains fixed at frame 1 that are linked by the coupler's common normal segment  $h$ . Note that the distance between  ${}^1_4\vec{p}$

Table 3. Kinematic Analysis of Points

Frame	Axis	Rotation	Translation
${}^1_2T$	z	$\theta$	$d_1$
${}^2_3T$	x	$\alpha$	a
${}^3_4T$	z	$\phi$	$c_1$
${}^1_8T$	x	$\gamma$	g
${}^8_7T$	z	$\psi$	$d_2$
${}^7_6T$	x	$\beta$	b
${}^6_5T$	z	$\delta$	$c_2$

and  $\frac{1}{5}\vec{p}$  is  $h$ ) (see eq. 16). The Plücker coordinates of the moving axes are obtained from the matrices  ${}^1_4T$  and  ${}^1_5T$ :

$$\begin{aligned} {}^1_4T &= {}^1_2T_3^2T_4^3T & (16) \\ &= Z(\theta, d_1)X(\alpha, a)Z(\phi, c_1) \\ {}^1_5T &= {}^1_8T_7^8T_6^7T_5^6T \\ &= X(\gamma, g)Z(\psi, d_2)X(\beta, b)Z(\delta, c_2) \end{aligned}$$

where,

$$[X(\theta, x)] = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[Z(\theta, z)] = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

#### 4.4 Part 4: Finite Line (Cylinder) Generation

The next step in testing the mechanism is determining the location of all of the mechanism's points. For each linear interpolation of  $\theta$  and  $d_1$  of the mechanism, the points can be generated by using the above kinematic analyses. The end of each link has to be generated differently for each chain (see Fig. 7). For the driving chain we substitute  $d_{1max}$  for  $d_1$  and  $c_{1max}$  for  $c_1$  to obtain the three dimensional coordinates of the end point (see eq. 17). However, the driven chain measures the translations in the opposite direction- from collar to elbow instead of elbow to collar. This results in the end point being in the opposite direction of the elbow relative to the collar (except when the translation is at its maximum/minimum). This makes the translation of the point, for example,  $c_2 - c_{2max}$  (see eq. 18).

$$DrivingEnd = {}^1_{4max}T = Z(\theta, d_1)X(\alpha, a)Z(\phi, c_{1max}) \quad (17)$$

$$CouplerEnd = {}^1_{5max}T = X(\gamma, g)Z(\psi, d_2)X(\beta, b)Z(\delta, c_2 - c_{2max}) \quad (18)$$

For each incremental step the twelve points can be calculated and the eight line segments generated. The segments are numbered starting with the fixed link's axis, proceeding around the closed chain, and ending with the fixed link's common normal (see Tbl. 4). This yields the set of line segments for the incremental step that can then be tested to see if a collision has occurred.

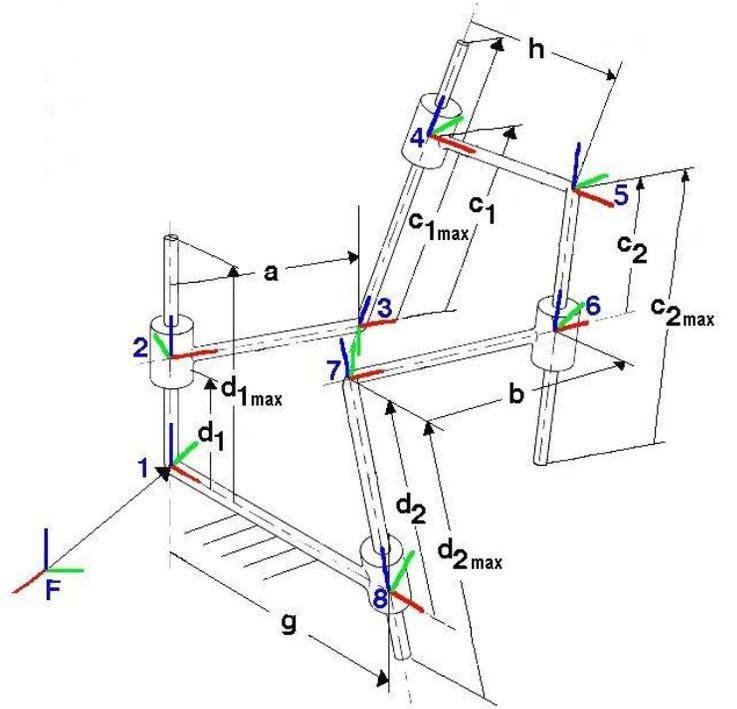


Figure 7. Link Length definitions

Table 4. Segment Designations of the Spatial 4C Mechanism

Segment No.	Start Point	End Point
1	Fixed Elbow	Fixed End
2	Driving Collar	Driving Elbow
3	Driving Elbow	Driving End
4	Coupler Collar	Coupler Elbow
5	Coupler Elbow	Coupler End
6	Driven Collar	Driven Elbow
7	Driven Elbow	Driven End
8	Fixed Collar	Fixed Elbow

## 5 Cylinder Testing Logic

It is necessary that for each incremental step that all of the cylinders are tested for collisions. Since speed is important for the calculations, we look to reduce the number of tests that must be run for each incremental step. First, cylinders can not collide with themselves since the links are rigid. Also, testing cylinder 1 for a collision with cylinder 5 is redundant to checking cylinder 5 to cylinder 1. This greatly reduces the number of tests required. The number can be further reduced by observing the design of

Table 5. Case Study - Link Parameters

<i>Twist (degrees)</i>	<i>Length (unit)</i>
$\alpha = 70$	$a = 10$
$\beta = 120$	$b = 8$
$\gamma = 110$	$g = 19$
$\eta = 55$	$h = 15$

Table 6. Case Study - Motion Input

$\theta$ (degrees)	$d_1$ (unit)	Increments	$\pm\phi$
78	100	50	+
105	140	40	+
130	110	60	+
100	80	30	+
95	100	50	+

Table 7. Case Study - Translation Output

<i>Translation</i>	<i>Min (unit)</i>	<i>Max (unit)</i>
$d_1$	80.000	140.000
$c_1$	-19.029	-108.032
$d_2$	-104.658	-174.518
$c_2$	-91.520	-138.356

Table 8. Case Study - Results

<i>Distance (unit)</i>	<i>Seg. No.</i>	<i>Seg. No.</i>	$\theta$ (degree)	$d_1$ (unit)
4.00	5	7	122.00	102.00

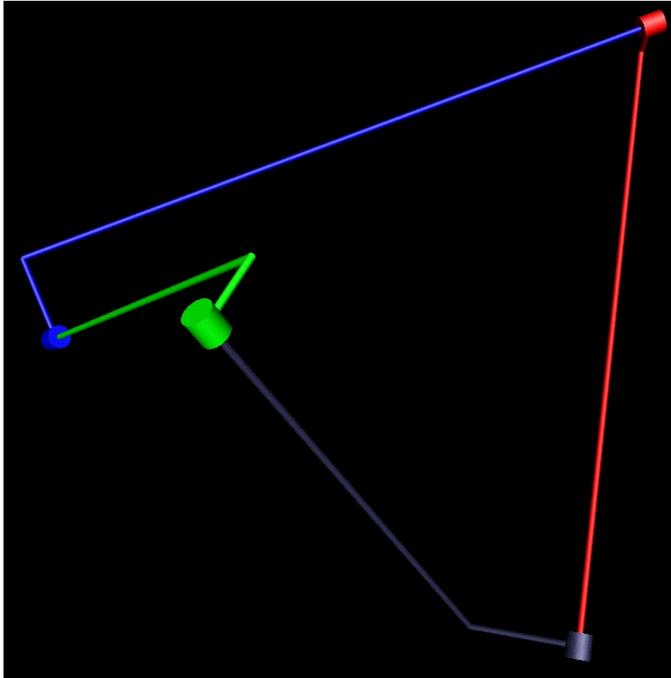


Figure 8. Case Study: 4C Mechanism

the mechanism. In a spatial 4C mechanism it is not possible for a cylinder to collide with the cylinders that are adjacent to it. This reduces the number of possible cylinder combinations in a spatial 4C mechanism to twenty.

## 6 Case Study

To demonstrate the methodology presented in this paper we used the spatial 4C mechanism described in Table 5. The mechanisms fixed link is grey, driving link green, driven link red and coupler link blue (see Fig. 8). Each link's common normal was assigned a radii of one unit and each axis a radii of two. The axis was modelled as having a larger radii to account for the collar that has to translate and rotate about the links' axis.

A set of via points (see Tbl. 6) was then entered for the mechanism and some initial testing was performed. From the link twist values the allowable range of  $\theta$  was calculated

(see eq. 14) and both  $C_1$  and  $C_2$  exist (case 4). This made the allowable  $\theta$  ranges  $52.33 \leftrightarrow 174.68$  and  $-52.33 \leftrightarrow -174.68$  (see eq. 19).

$$C_1 = 0.61107 \implies \theta_1 = 52.33 \quad (19)$$

$$C_2 = -0.99569 \implies \theta_2 = 174.68$$

After calculating the allowable  $\theta$  range each of the via points was tested to make sure that they were within the same allowable range.

The next step in testing the mechanism is to determine the global translational minimums and maximums (see Tbl.7). Inspection of the mechanism's translations shows that there were no sign changes in the individual translations and that none of them approach zero. After this preliminary testing and data acquisition, collision detection was performed. The result is that there was no collision detected. The closest that the mechanism came to a collision occurred when segments five and seven were only 4 units apart at  $\theta = 122$  and  $d_1 = 102$  (see Tbl. 8). Figure 9 shows the mechanism when it is in its minimum distance configuration.

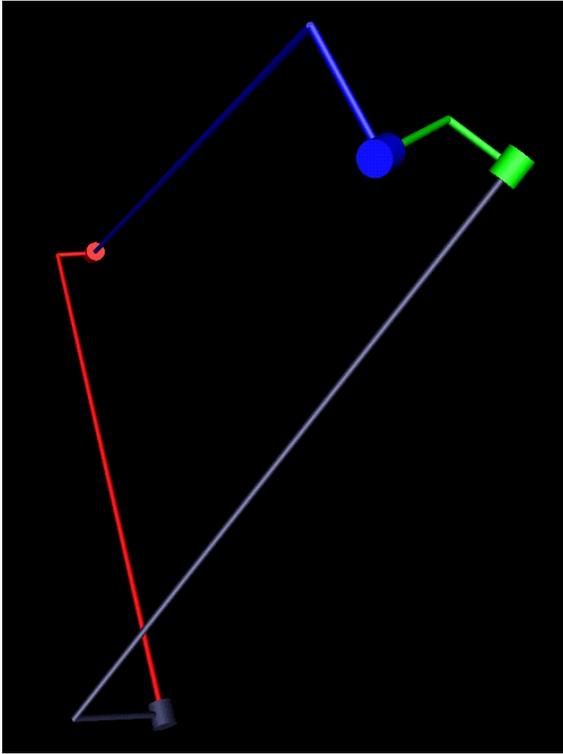


Figure 9. Case Study: Minimum Distance Configuration

## 7 Future Work

Currently, we are exploring methods to analytically determine the minimum and maximum translations along the joint axes of a spatial 4C mechanism. Additional are working toward implementing this work into our current spatial mechanism design programs; VRSpatial [Kihonge et al, 2002] and SPADES [Larochelle, 1998].

## 8 CONCLUSIONS

In this paper we have presented a novel methodology for detecting collisions of cylindrically shaped rigid bodies moving in three dimensions. This algorithm uses line geometry and dual number algebra to exploit the geometry of cylindrical objects to facilitate the detection of collisions. First, the rigid bodies are modelled with infinite cylinders and an efficient necessary condition for collision is evaluated. If the necessary condition is not satisfied then the two bodies do not collide. If the necessary condition is satisfied then a collision between the bodies may occur and we proceed to the next stage of the algorithm. In the second stage the bodies are modelled with finite cylinders and a definitive necessary and sufficient collision detection algorithm is employed. The result is a straight-forward and efficient means of detecting collisions of cylindrically shaped bodies moving in three dimensions. This methodology has applications in spatial

mechanism design, robot motion planning, and workspace analyses of parallel kinematic machines such as Stewart-Gough platforms. A case study examining a spatial 4C mechanism for self collisions was included.

## 9 ACKNOWLEDGMENT

This work was made possible by NSF Grant No. #9816611. The collaborations with and contributions of Prof. Vance and her student Denis Dorozhkin of Iowa State University are gratefully acknowledged.

## REFERENCES

- [Agius, Larochelle, 2002] Agius, A., and Larochelle, P. (2002), SPASUR: Interactive visualization of the coupler surfaces of the spatial 4C mechanism, *Proceedings of DETC/DAC: The 2002 ASME Design Engineering Technical Conference and Computers and Information Conference*.
- [Belta, Kumar, 2000] Belta, Calin and Kumar, Vijay (2000), An efficient, geometric approach to rigid body motion interpolation, *Proceedings of DETC'00: The ASME 2000 Design Engineering Technical Conferences and Computers and Information Conference*, DET2000/MECH-14216.
- [Dees, 2001] Dees, S.L. (2001), *Spatial mechanism design using an svd-based distance metric*, Master's Thesis, Florida Institute of Technology.
- [Duffy, 1980] Duffy, J. (1980), *Analysis of mechanisms and robot manipulators*, John Wiley & Sons.
- [Fischer, 1999] Fischer, Ian, (1999), *Dual-number methods in kinematics, statics and dynamics*, CRC Press LLC.
- [Fischer, 1994] Fischer, Ian (1994), *Novel methods in the displacement analysis of spatial mechanisms*, ASME.
- [Fuhrmann, Schomer, 2001] Fuhrmann, Artur and Schomer, Elmar (2001), A general method for computing the reachable space of mechanisms, *Proceedings of DETC/DAC: The 2001 ASME Design Engineering Technical Conference and Computers and Information Conference*, DET2001/DAC-21057.
- [Hanchak, Kashani, Murray] Hanchak, Michael, Kashani, Reza and Murray, Andrew, *Dynamics and control of a 4C mechanism*.
- [Johnson, Vance, 2001] Johnson, Tom and Vance, Judy (2001), The use of the VOXMAP pointshell method of collision detection in virtual assembly methods planning, *Proceedings of DETC/DAC: The 2001 ASME Design Engineering Technical Conference and Computers and Information Conference*, DET2001/DAC-21137.
- [Kihonge et al, 2002] Kihonge, J., Vance, J. and Larochelle, P. (2002), *Spatial Mechanism Design in Virtual Reality with Networking*, ASME *Journal of Mechanical Design* (in press).
- [Larochelle, 2000] Larochelle, P. (2000), *Circuit and branch rectification of the spatial 4c mechanism*, *Proceedings of*

- the ASME Design Engineering Technical Conferences, Baltimore, USA.
- [Larochelle, 1998] Larochelle, P. (1998), Spades: software for synthesizing spatial 4c mechanisms, *Proceedings of the ASME Design Engineering Technical Conferences*, Atlanta, USA.
- [Larochelle, 1994] Larochelle, P. (1994), *Design of cooperating robots and spatial mechanisms*, PhD Dissertation, University of California, Irvine.
- [McCarthy, 2000] McCarthy, J.M., (2000), Geometric design of linkages, Springer-Verlag New York.
- [Mi, Yang, Abdel-Malek, Jay, 2002] Mi, Z., Yang, J., Abdel-Malek, K. and Jay, L. (2002), Planning for kinematically smooth manipulator Trajectories, *Proceedings of DETC'02: ASME 2002 Design Engineering Technical Conferences and Computers and Information Conference*, DET2002/MECH-34325.
- [Murray and Larochelle, 1998] Murray, A.P., and Larochelle, P. (1998), A classification scheme for planar 4r, spherical 4r, and spatial rccc linkages to facilitate computer animation, *Proceedings of the ASME Design Engineering Technical Conferences*, Atlanta, USA.
- [Patoglu, Gillespie, 2000] Patoglu, Volkan and Gillespie, R. (2000), Extremal distance maintenance for parametric curves and surfaces, *Proceedings of the 2002 IEEE International conference on robotics & automation*.
- [Serre, Riviere, Duong, Ortuzar, 2001] Serre, P., Riviere, A., Duong, A. and Ortuzar, A. (2001), Analysis of a geometric specification, *Proceedings of DETC/DAC: The 2001 ASME Design Engineering Technical Conference and Computers and Information Conference*, DET2001/DAC-21123.
- [Stocco, 2001] Stocco, Leo (2001), Path verification for unstructured environments and medical applications, *Proceedings of DETC/DAC: The 2001 ASME Design Engineering Technical Conference and Computers and Information Conference*, DET2001/DAC-21128.
- [Wu, Mayne, 2000] Wu, C. and Mayne, R. (2000), A geometric based method for distance-to-contact calculations in two dimensions, *Proceedings of DETC'00: The ASME 2000 Design Engineering Technical Conferences and Computers and Information Conference*, DET2000/MECH-14128.
- [Xavier, 2000] Xavier, Patrick (2000), Implicit convex-hull distance of finite-screw-swept volumes, *Proceedings of the 2002 IEEE International conference on robotics & automation*.
- [Zsombor-Murray, 1992] Zsombor-Murray, P.J. (1992), Spatial visualization and the shortest distance between two lines problem, *Proceedings of the 5th ASEE International Conference ECGDG*. Melbourne, Australia.