An Analytical Method to Detect Collision between Cylinders Using Dual Number Algebra*

Rajeevlochana G. Chittawadigi and Subir K. Saha

Abstract— **Cylinder, a common geometric entity has a discontinuity at the joining of cylindrical surface and circular-disks. Hence, collision detection between two cylinders in space is a difficult task and few have reported formulations to solve it. In this paper, a novel analytical methodology is proposed to detect collision or intersection between two cylinders. The configuration, i.e., position and orientation, between the cylinders was represented using the four Denavit-Hartenberg (DH) parameters plus two extra parameters. Dual Number Algebra was used to extract these six parameters. Tests involved in collision detection between the cylinders were between the lines and rectangles in a plane, thus considerably simplifying the problem of collision detection. As an illustration, an one-DOF arm modeled as a cylinder with cylindrical shaped obstacles were modeled and tested for their collisions. The results were validated with an analytical method available in the literature and a commercial software.**

I. Introduction

The term 'cylinder' typically refers to the volume bound by a right-circular cylindrical surface and two circular disks as end-caps. Along with spheres, cuboids, cones, etc., cylinders are one of the main geometric primitives used in Computer Graphics, Computer Aided Geometric Design and related areas. In applications related to Robotics, medical and surgical simulations, simulation of various processes, intersection of different geometric objects in the environment needs to be determined for realistic simulation, which is referred as 'Collision Detection'.

Collision detection, by itself is a very large field and researchers have proposed different formulations for geometric objects of various shapes [1]. To simplify the process, typically, geometric objects (shapes) are enclosed inside simpler bounding volumes such as spheres, Axis Aligned Bounding Box (AABB), Oriented Bounding Box (OBB), etc. and a broad phase test is done between the bounding volumes. If these volumes intersect, narrow phase testing is done between the actual geometry (shape) [2]. Cylinder, though a very commonly used geometric entity, is usually not considered as a bounding volume because collision detection between two cylinders is reported [3] to be expensive. Instead, a capsule (cylinders with hemi-spherical ends) has been suggested to be used as bounding volumes. Note that the latter tests are quite simple.

* This research was funded by BARC/ BRNS, Mumbai, India, under the project, "Setting up of a Programme for Autonomous Robotics" at IIT Delhi, India.

R. G. Chittawadigi is an M.S. (Research) student in the Department of Mechanical Engineering, Indian Institute of Technology Delhi, New Delhi, India (e-mail: rajeevlochan.iitd@gmail.com).

S. K. Saha is a professor in the Department of Mechanical Engineering, Indian Institute of Technology Delhi, New Delhi, India (phone: +91 11 26591135; fax: +91 11 26582058; e-mail: saha@mech.iitd.ac.in).

Exact collision detection between two cylinders in the 3D Cartesian space, either for broad phase or narrow phase testing has been reported by few. Eberly [3] considered Separating Axis Tests (SAT) to determine overlap of two cylinders along certain directions (each of the cylinder axes, a vector connecting the centers of the cylinders and vectors on a plane perpendicular to each of cylinder axes and passing through their respective centers). It also required optimization of a function over a spherical surface. A method was proposed by Ketchel and Larochelle [4] using line geometry to detect if two cylinders intersect, by determining the common normal between the cylinder axes and the points where the normal intersects the axes. Based on the condition of these points lying inside respective cylinders, different tests were proposed. However, detailed logical derivation was not reported. Choi [5] reported cylinder as a Composite Quadric Model (CQM), where cylindrical surface (quadric) is bounded by a circular edge and plane on either side, collectively referred as boundary elements. As CQMs are semi-algebraic entities, firstly, algebraic methods were used to compute contact points between all possible pairs of boundary elements. Secondly, the contact points were verified if they lay on the actual CQMs. Biermann et al. [6] used optimization techniques to determine the distance between different entities of cylinders (cylindrical surface, circular edge and circular face). A concept of Minkowski Portal Refinement (MPR) was used in XenoCollide [7] to represent convex shapes, including cylinder, using a support function and collision between two cylinders was reported using an iterative technique. XenoCollide has been implemented in libccd [8], an open-source collision library for convex bodies. Kodam et al. [9] used cylinder-cylinder intersection in Discrete Element Method (DEM) to simulate the dynamics of particulate systems. They recognized six contact scenarios between bands (cylindrical surfaces), faces (circular faces) and edges (circular edges) and gave analytical expressions to determine the occurrence and type of contact. However, the axisymmetric nature of cylinders was not exploited by them. Guo et al. [10] reported limitations with methodologies proposed by Kodam et al. [9] in certain configurations of cylinders and proposed several additional tests and modifications to be incorporated for accurate tests between finite cylinders. The cylinder-cylinder collision module in Teikitu Gaming System [11] also uses separating axis tests but it is based on heuristic solutions and no formal derivation is available in open literature. An analytical formulation was reported in [12] to find the proximity of two cylinders in 3D space. Each cylinder comprised of four geometric primitives and shortest distance between five combinations of these primitives was used to conclude if two cylinders collide or not. However, four of these were solved in closed form, whereas the fifth combination required solution of an eighth-order polynomial equation.

In this paper, a novel analytical methodology is proposed to detect collision between two finite cylinders in 3D space. It exploits the axisymmetric nature of the cylinders to simplify the tests which has not been used, to the best knowledge of the authors, in the reported literature. The reminder of the paper is arranged as follows: Concepts of Dual Number Algebra and Denavit-Hartenberg (DH) [13] parameters are defined in Section II, which are used in Section III for the derivation of proposed methodology. As an illustration, collision between a rotating arm and few static cylindrical objects was tested in Section IV. The results were verified with those obtained using the algorithm of [4] and the collision detection module of Autodesk Inventor software.

II. Dual Number Algebra

A dual number or a dual vector is represented as a sum of a real part and a dual part. The latter begins with the dual entity ε, which is nilpotent, i.e., $\varepsilon^2 = 0$ [14]. Dual Number Algebra is often used in the field of displacement analysis, kinematic synthesis and dynamic analysis of spatial mechanisms as it produces concise and compact notations [15-16]. The same when extended to vectors is known as Dual Vector Algebra, which was used by the authors in [17] to derive an analytical methodology to extract DH parameters of a serial robot from its CAD model. Here, the concept is improvised and extended to test collision between cylinders.

A dual number ($\hat{a}$) is represented by

$$\hat{a} = a + \varepsilon a^* \tag{1}$$

where a and a^* are real scalar number and corresponding dual number, respectively, whereas $\varepsilon^2 = 0$. Similarly, a dual vector ($\hat{\mathbf{a}}$) is expressed as

$$\hat{\mathbf{a}} = \mathbf{a} + \varepsilon \mathbf{a}^* = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^{\mathrm{T}} + \varepsilon \begin{bmatrix} a_x^* & a_y^* & a_z^* \end{bmatrix}^{\mathrm{T}} \tag{2}$$

where $\mathbf{a}$ is called the resultant vector and $\mathbf{a}^*$ is called the moment vector, which are related as

$$\mathbf{a}^* = \mathbf{p} \times \mathbf{a} \tag{3}$$

In (3), $\mathbf{p}$ is the position vector of any point P on the line shown in Fig. 1(a). Operations on dual vectors are performed similar to those on Cartesian vectors. For example, dot- and cross-products of two dual vectors, also referred to as line dot- and cross-products [14], are defined by

$$\begin{aligned} \hat{\mathbf{a}}_1 . \hat{\mathbf{a}}_2 &= (\mathbf{a}_1 + \varepsilon \mathbf{a}_1^*) . (\mathbf{a}_2 + \varepsilon \mathbf{a}_2^*) \\ &= \mathbf{a}_1 . \mathbf{a}_2 + \boldsymbol{\varepsilon} (\mathbf{a}_1 . \mathbf{a}_2^* + \mathbf{a}_1^* . \mathbf{a}_2) \end{aligned} \tag{4}$$

$$\begin{aligned} \hat{\mathbf{a}}_1 \times \hat{\mathbf{a}}_2 &= (\mathbf{a}_1 + \varepsilon \mathbf{a}_1^*) \times (\mathbf{a}_2 + \varepsilon \mathbf{a}_2^*) \\ &= \mathbf{a}_1 \times \mathbf{a}_2 + \varepsilon (\mathbf{a}_1 \times \mathbf{a}_2^* + \mathbf{a}_1^* \times \mathbf{a}_2) \end{aligned} \tag{5}$$

In the same way, the Euclidean norm of a dual vector $\hat{\mathbf{a}}$ is given by

$$\|\hat{\mathbf{a}}\| = \|\mathbf{a}\| + \frac{\boldsymbol{\varepsilon}\,(\mathbf{a}.\mathbf{a}^*)}{\|\mathbf{a}\|} \tag{6}$$

where $\|\hat{\mathbf{a}}\|$ and $\|\mathbf{a}\|$ are the Euclidean norms of the dual vector $\hat{\mathbf{a}}$ and vector $\mathbf{a}$, respectively. Dual unit vector $\hat{\mathbf{e}}$ is defined as a dual vector whose Euclidean norm is equal to unity [18], i.e.,

$$\hat{\mathbf{e}} = \frac{\hat{\boldsymbol{a}}}{\|\hat{\boldsymbol{a}}\|} = \frac{\boldsymbol{a}}{\|\,\boldsymbol{a}\|} + \frac{\boldsymbol{\varepsilon}\,(\boldsymbol{a} \times \boldsymbol{a}^*) \times \boldsymbol{a}}{\|\boldsymbol{a}\|^3} \tag{7}$$

A. Dual Angle

The concept of dual angle is now introduced which is used to represent the relative displacement and orientation between two lines in space [14]. The dual angle ($\hat{\alpha}$), as illustrated in Fig. 1(b), is defined as

$$\hat{\alpha} = \alpha + \varepsilon \alpha^* \tag{8}$$

where α is the projected angle between the lines and $\alpha^*$is the shortest distance between them. If $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$ are considered dual unit vectors representing two lines and $\hat{\mathbf{n}}$ is the dual unit vector along common normal, the dual angle between them is determined using

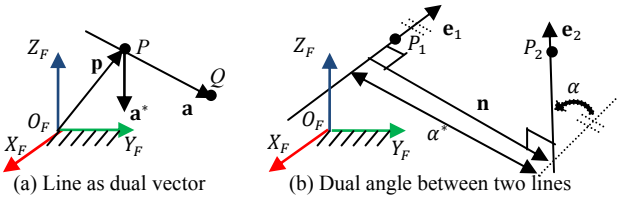
$$\cos \hat{\alpha} = \hat{\mathbf{e}}_1 . \hat{\mathbf{e}}_2 = x + \varepsilon x^* \tag{9}$$

$$\sin \hat{\alpha} = (\hat{\mathbf{e}}_1 \times \hat{\mathbf{e}}_2) . \hat{\mathbf{n}} = y + \varepsilon y^* \tag{10}$$

where $x, x^*,$ y and y^* are real numbers and the dual angle is

$$\begin{aligned} \hat{\alpha} &= \arctan2(\sin \hat{\alpha}, \cos \hat{\alpha}) \\ &= \arctan2(y, x) + \frac{\varepsilon [x y^* - x^* y]}{(x^2 + y^2)} \end{aligned} \tag{11}$$

Using (8) and (11), the relative orientation (α) and distance between the lines (α^*) can be determined. Note that, for parallel lines, no unique common normal exists and a method used in [17] should be followed.



(a) Line as dual vector (b) Dual angle between two lines

Figure 1. Line representation using Dual Number Algebra

B. DH Parameters Based Transformation

To represent a coordinate frame with respect to another, six coordinates or parameters are required, of which three correspond to the relative position (e.g., position vector) and remaining three for their relative orientation (e.g., Euler angles). Melchiorri [19] reported that four Denavit-Hartenberg (DH) [13] parameters, typically used in geometric modeling of serial robots, and additional two parameters can also be used as the required six parameters, referred here as six DH (SDH) parameters. As illustrated in Fig. 2, the four DH parameters between frames $X_1Y_1Z_1$ and $X_1^{IV}Y_1^{IV}Z_1^{IV}$, relates a line (Z_1^{IV}) relative to frame $X_1Y_1Z_1$. The extra parameter c is required to locate the origin of frame $X_2Y_2Z_2$ along Z_1^{IV} and parameter φ to orient X_1^{V} (parallel to X_1^{IV}) with X_2. The six DH parameters are described in Table I.

TABLE I. Descriptions of six DH parameters

Parameters	Description
b	Distance between X_1 and X_1^{I} along Z_1
θ	Angle between X_1^{I} and X_1^{II} about Z_1^{I}
a	Distance between Z_1^{II} and Z_1^{III} along X_1^{II}
α	Angle between Z_1^{III} and Z_1^{IV} about X_1^{III}
c	Distance between X_1^{IV} and X_1^{V} along Z_1^{IV}
φ	Angle between X_1^{V} and X_2 along Z_1^{V}

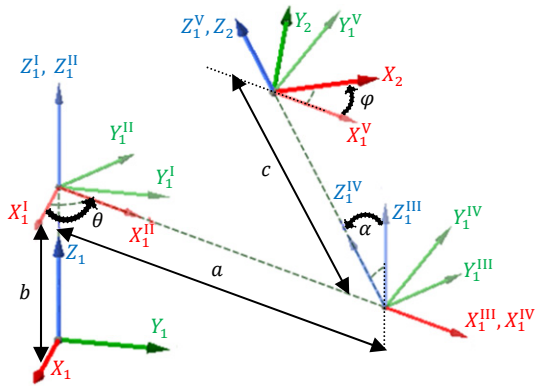


Figure 2. Six DH parameters between two coordinate frames

Given the homogeneous transformation matrix of Frame 2 ($X_2Y_2Z_2$) with respect to Frame 1 ($X_1Y_1Z_1$), which is denoted as $[\mathbf{T}_2]_1$, the common normal (i.e., $X_1^{II} = X_1^{IV}$) between Z_1 and Z_2 is determined. The corresponding SDH parameters can then be found as three dual angles between dual vectors mentioned in Table II and by using (8-11).

TABLE II. SIX DH PARAMETERS AS DUAL ANGLES

Parameters	Dual Angle	Dual Vectors
θ and b	$\hat{\theta}$	$\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_1^{II}$
α and a	$\hat{\alpha}$	$\hat{\mathbf{z}}_1^{II}$ and $\hat{\mathbf{z}}_1^{IV}$
φ and c	$\hat{\varphi}$	$\hat{\mathbf{x}}_1^{IV}$ and $\hat{\mathbf{x}}_2$

III. COLLISION BETWEEN CYLINDERS

A novel analytical method is proposed here to detect collision between two cylinders using the SDH parameters given in Table I. Referring to Fig. 3, a coordinate frame (F_1) is attached to the center of the cylinder C_1, whose Z axis (Z_1) is along the axis of the cylinder. Similarly, F_2 is attached to cylinder C_2. For any transformation between them, corresponding SDH parameters are determined using (8-11).

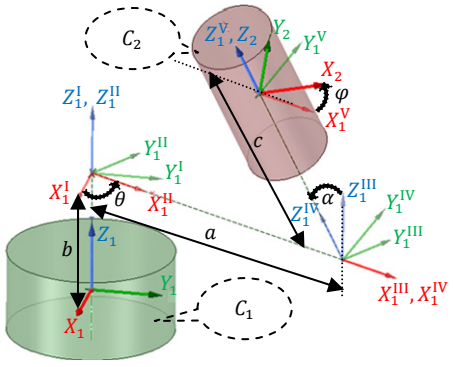


Figure 3. Six DH parameters: Sequence b, θ, a, α , c and φ

In this formulation, a modified sequence of transformations is used. The angular parameter θ is moved to the beginning of the series of transformations and distance parameter a, is moved towards the end, before angle parameter φ. This is done to take the advantage of cylinders being axisymmetric and to facilitate projected rectangles tests in the proposed formulation. Note that the transformation between F_1 and F_2 is still the same, as shown in Fig. 4. Since

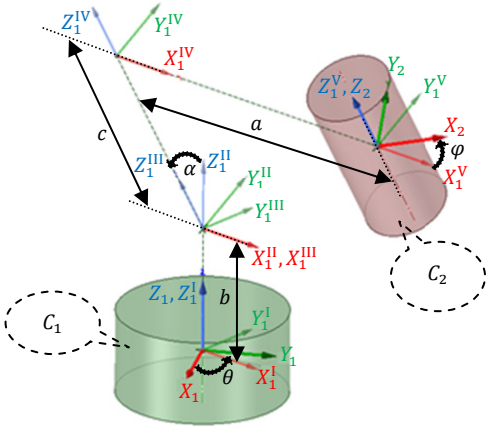


Figure 4. Six DH parameters: Sequence θ, b, α , c, a and φ

cylinders are axisymmetric, θ and φ do not play any role in deciding whether two cylinders intersect or not. Hence, only four parameters, namely b, a, α and c, between the frames $X_1^IY_1^IZ_1^I$ and $X_1^VY_1^VZ_1^V$ are sufficient. For the sake of convenience, Cartesian frame $X_1^IY_1^IZ_1^I$ henceforth is referred as F_A and $X_1^VY_1^VZ_1^V$ as F_B. Note that all vector equations in the formulation are referred in these frames.

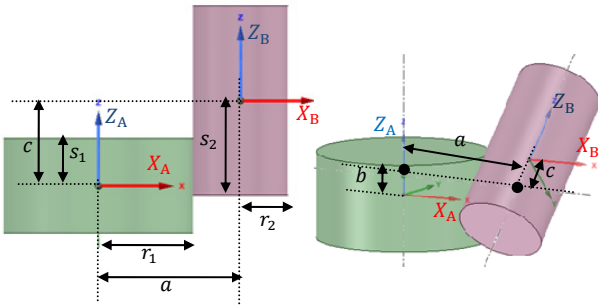
A. Methodology

As an input to the proposed methodology, transformation ($[\mathbf{T}_2]_1$) of F_2 with respect to F_1 is required. Further the radii (r_1 and r_2) and half of the heights of the cylinders, referred here as half-extents (s_1 and s_2), are needed. Given two cylinders, one can find the SDH parameters using the Dual Number Algebra presented in Section II. To detect collision, intersection of cylinders of infinite length is tested first by checking if the separation distance (a) is greater than the sum of radii (r_1+r_2). If so, the cylinders do not intersect and there is no need of further tests. Else, further tests on finite cylinders need to be performed. For finite cylinders, the following strategies are undertaken:

- If the cylinders are parallel (i.e., $\alpha = 0°$ or $180°$ and $b = 0$, $c \in \mathbb{R}$), as shown in Fig. 5(a), the sum of half-extents (s_1 and s_2) is checked with axial distance between the centers of the cylinders, i.e., the value of c. If the sum is greater or equal to the value of c, the cylinders intersect. If the sum is less, the two cylinders do not intersect.

- For non-parallel cylinders, the points on cylinder axes where the common normal (X_1^{II}) intersects are tested. If these points are within respective cylinders (i.e., $|b| \le s_1$ and $|c| \le s_2$), then the two cylinders intersect, as illustrated in Fig. 5(b). This is similar to "On-On Test" reported in [4]. Here it is referred as Non Parallel Test (NPT).

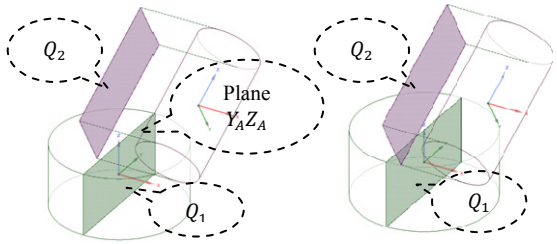
  If either or both common normal points is/are outside the respective cylinders, C_1 and C_2 are projected onto Y_AZ_A plane resulting in two rectangles Q_1 and Q_2, respectively, as shown in Fig. 6. A simple Separating Axis Test [2] is performed between the rectangles and if they do not intersect (Fig. 6(a)), the cylinders do not intersect and no further test is required. The coordinates of the vertices of the projected rectangles



(a) Parallel Test (b) Non Parallel Test

Figure 5. Finite cylinder tests

are computed as explained later. If the projected rectangles intersect, as shown in Fig. 6(b), the two cylinders at hand may or may not intersect, which is determined by the Vertex Edge Test explained below.



(a) Non-intersecting rectangles (b) Intersecting rectangles

Figure 6. Projection Test between rectangles

B. *Vertex Edge Test*

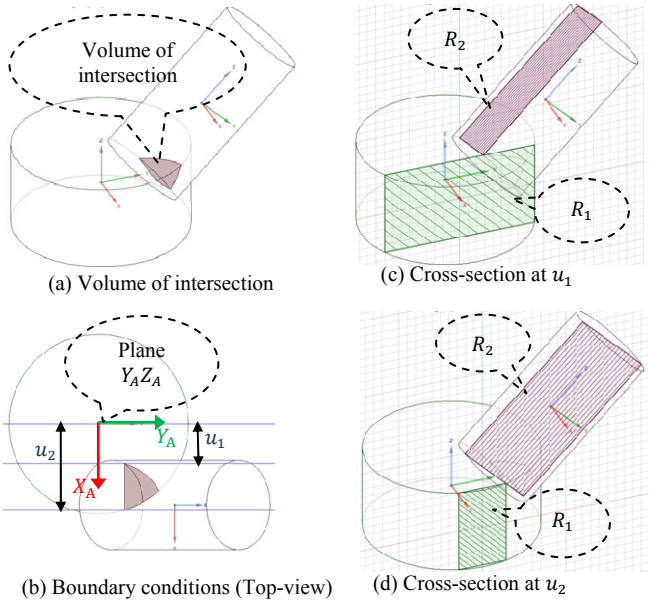
First, a hypothesis is proposed which will be verified later.

Hypothesis: If two cylinders are colliding, a volume of intersection exists between them which can be determined using numerical integration techniques [20]. In the proposed methodology, this step is avoided by considering only the boundary conditions of the volume of intersection along X_A. At these boundary conditions, the cross-sections of the C_1 and C_2 on a plane parallel to Y_AZ_A result in rectangles R_1 and R_2, which would be touching each other.

For example, in Fig. 7(a), the volume of intersection of two intersecting cylinders is illustrated, for which the boundary conditions at u_1 and u_2 from Y_AZ_A along X_A in the top-view shown in Fig. 7(b). The cross-section of the cylinders at boundary condition u_1 is shown in Fig. 7(c). Note that the two rectangles are touching. Similarly rectangles are shown touching for boundary condition u_2 of Fig. 7(d).

Proof: The homogeneous transformation matrix ($[\mathbf{T}_B]_A$) representing F_B with respect to F_A is determined as

$$[\mathbf{T}_B]_A = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & \cos\alpha & -\sin\alpha & -c\sin\alpha \\ 0 & \sin\alpha & \cos\alpha & b + c\cos\alpha \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$



(a) Volume of intersection (c) Cross-section at u_1

(b) Boundary conditions (Top-view) (d) Cross-section at u_2

Figure 7. Boundary conditions of Vertex Edge Test

For the cross-section plane at a distance of u from Y_AZ_A as shown in Fig. 8, the expressions of the coordinates of vertices of R_1 in F_A, are

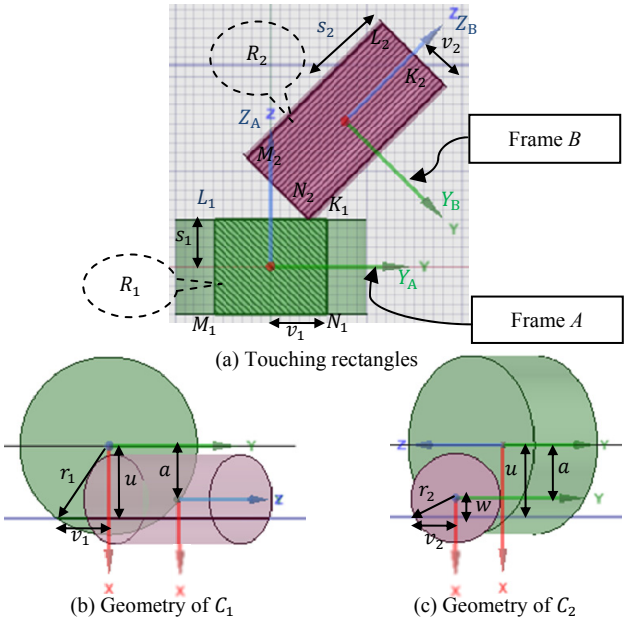
$$[\mathbf{k}_1]_A = [u \quad v_1 \quad s_1]^T \quad (13)$$

$$[\mathbf{l}_1]_A = [u \quad -v_1 \quad s_1]^T \quad (14)$$

$$[\mathbf{m}_1]_A = [u \quad -v_1 \quad -s_1]^T \quad (15)$$

$$[\mathbf{n}_1]_A = [u \quad v_1 \quad -s_1]^T \quad (16)$$

where v_1 and s_1 are half-breadth and half-extent, respectively, of R_1 as shown in Fig. 8(a). Similarly, vertices of R_2 in F_B and after transforming to F_B are given by



(a) Touching rectangles

(b) Geometry of C_1 (c) Geometry of C_2

Figure 8. Rectangles R_1 and R_2 at a boundary condition

$$[\mathbf{k}_2]_B = [w \quad v_2 \quad s_2]^{\mathrm{T}}\ ; \quad [\mathbf{k}_2]_A = [\mathbf{T}_B]_A[\mathbf{k}_2]_B \quad (17)$$

$$[\mathbf{l}_2]_B = [w \quad -v_2 \quad s_2]^{\mathrm{T}}\ ; \quad [\mathbf{l}_2]_A = [\mathbf{T}_B]_A[\mathbf{l}_2]_B \quad (18)$$

$$[\mathbf{m}_2]_B = [w \quad -v_2 \quad -s_2]^{\mathrm{T}}\ ; \quad [\mathbf{m}_2]_A = [\mathbf{T}_B]_A[\mathbf{m}_2]_B \quad (19)$$

$$[\mathbf{n}_2]_B = [w \quad v_2 \quad -s_2]^{\mathrm{T}}\ ; \quad [\mathbf{n}_2]_A = [\mathbf{T}_B]_A[\mathbf{n}_2]_B \quad (20)$$

where $w = u - a$, v_2 and s_2 are half-breadth and half-extent of R_2, respectively.

The condition for touching at the boundary exists if two rectangles just touch each other. It can be established by constraining a vertex of one rectangle to lie on an edge of the other. This can be derived as algebraic equation in u. The tests are done for 32 possible combinations of edges and vertices, of which two are illustrated here. Tests on remaining combinations are similar.

1) Vertex N_2 on Edge K_1L_1

The condition for vertex N_2 to lie on edge K_1L_1 is obtained by equating the Z components of vertex N_2 and K_1 in F_A using (13) and (20), as illustrated in Fig. 8(a), i.e.,

$$s_1 = v_2 \sin\alpha + (c - s_2)\cos\alpha + b \quad (21)$$

which is rearranged as

$$v_2 = \frac{s_1 - b - (c - s_2)\cos\alpha}{\sin\alpha} \quad (22)$$

Using the geometry of C_2, Fig. 8(c), a quadratic equation in u is formed, i.e.,

$$u = a \mp \sqrt{{r_2}^2 - {v_2}^2} \quad (23)$$

After solving (23), one or both of the real solutions of u are tested to lie in the interval defined by

$$\{u \in \mathbb{R} \mid a - \min(r_1, r_2) \le u \le \max(r_1, r_2)\} \quad (24)$$

For value(s) of u lying in the interval of (24), the Y component of N_2 is tested if it lies on the edge K_1L_1 and not beyond it. If so, it is concluded that the projected rectangles R_1 and R_2 intersect and hence the cylinders C_1 and C_2 intersect and tests for the remaining combinations of vertices and edges are not required.

2) Vertex N_2 on Edge L_1M_1

The condition for vertex N_2 to lie on edge L_1M_1, as shown in Fig. 9, is obtained by equating Y components of N_2 and L_1 using (14) and (20), i.e.,

$$-v_1 = v_2 \cos\alpha - (c - s_2)\sin\alpha \quad (25)$$

Which, unlike (21), has term of v_1. Substituting $f \equiv 1$, $g \equiv -\cos\alpha$ and $h \equiv -(c - s_2)\sin\alpha$ in (25), a general form is given by

$$f v_1 + g v_2 + h = 0 \quad (26)$$

From cylinders' geometry, Fig. 8(b-c), two equations are

$$u^2 + {v_1}^2 = {r_1}^2 \quad (27)$$

$$(a - u)^2 + {v_2}^2 = {r_2}^2 \quad (28)$$

Combining (26-28), a quartic equation in u is then obtained as follows:

$$\lambda_1 u^4 + \lambda_2 u^3 + \lambda_3 u^2 + \lambda_4 u + \lambda_5 = 0 \quad (29)$$

where the coefficients of the equation are

$$\lambda_1 = -f^4 + 2f^2g^2 - g^4 \quad (30)$$

$$\lambda_2 = -4af^2g^2 + 4ag^4 \quad (31)$$

$$\lambda_3 = 2r_1^2f^4 + 2a^2f^2g^2 - 2r_1^2f^2g^2 - 2r_1^2f^2g^2 - 6a^2g^4 + 2r_2^2g^4 - 2f^2h^2 - 2g^2h^2 \quad (32)$$

$$\lambda_4 = 4ar_1^2f^2g^2 + 4a^3g^4 - 4ar_2^2g^4 + 4ag^2h^2 \quad (33)$$

$$\lambda_5 = -r_1^4f^4 - 2a^2r_1^2f^2g^2 + 2r_1^2r_2^2f^2g^2 - a^4g^4 + 2a^2r_2^2g^4 - r_2^4g^4 + 2r_1^2f^2h^2 - 2a^2g^2h^2 + 2r_2^2g^2h^2 - h^4 \quad (34)$$

Equation (29) can be solved using the analytical technique reported in [21]. Note that only two solutions of u are real and geometrically feasible. If one or both of the real solutions of u lie(s) in the interval defined in (24), the Z coordinate of vertex N_2 is determined to find if N_2 lies on the edge L_1M_1 and not beyond it. Hence collision is concluded.

Note that by substituting u = 0 and a = 0 in (13-20), the coordinates of the projected rectangles Q_1 and Q_2, introduced in Fig. 6 can be determined.

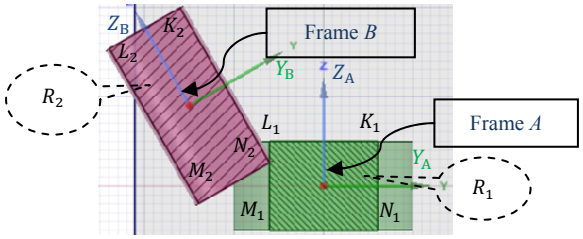


Figure 9. Condition for vertex N_2 to lie on edge L_1M_1

IV. Application

For broad phase collision tests between robot links and objects in the environment, Oriented Bounding Box (OBB) was used in [22], whereas Axis Aligned Bounding Box (AABB) was used in [23]. If they collide, actual geometric shapes of the robot links are then tested for collision. As most of the robot links are not rectangular blocks but cylindrical in shape, the broad phase tests are too conservative. Hence, to overcome this, the links are considered as cylinders (bounding volume), as also reported in the literature, e.g., [4].

In this paper, as a proof-of-concept, cylindrical shaped link for one-DOF arm and cylindrical obstacles were modeled in Autodesk Inventor software, as shown in Fig. 10. The proposed analytical method was implemented as an addin module inside the software using Visual C#. The joint was rotated programmatically and the collision tests between the arm and the obstacles were determined using the proposed methodology. The tests were also performed using the methodology in [4] and collision detection module of Autodesk Inventor. The results of the collision tests, i.e., the boolean (true or false) value, matched perfectly for each increment in joint angle, thus, validating the proposed methodology. Note that Autodesk Inventor uses a 3rd party commercial library by D-Cubed [24] for collision detection. No detailed mathematical formulation on D-Cubed is available in open literature. On the other hand, the authors of [4] used MATLAB environment for the implementation of their algorithm.

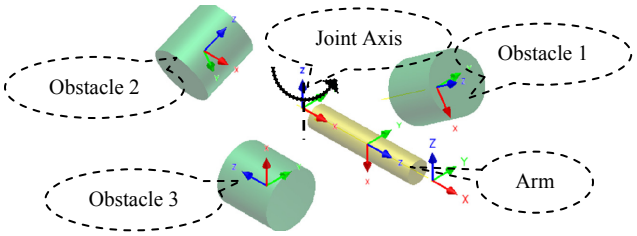


Figure 10. One-DOF arm with obstacles

For the proposed methodology, the sequence or the flow of the collision tests conducted between 'Arm' and 'Obstacle 1', for one complete rotation of the joint axis, with an increment of 1° is reported in Fig. 11. Note that majority of the rejection occurred in "Infinite Cylinders Tests". During the finite cylinder tests, the "Projection Test" also had high percentage of rejection due to which very few number of "Vertex Edge Test", which are relatively expensive, were carried out. Similar results were obtained for the tests with the other two obstacles having early exit from the tests.

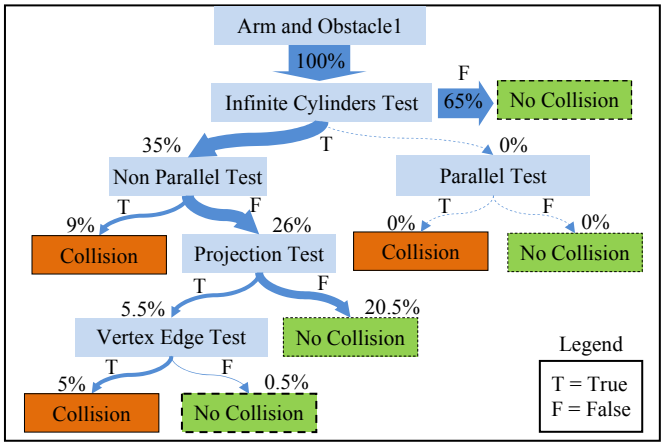


Figure 11. Flow of collision tests between 'Arm' and 'Obstacle 1'

In terms of efficiency, the proposed algorithm appears to have performed best among the three algorithms compared for the present task, as it took CPU[1] time of about 0.05-0.12 milliseconds (ms), whereas those using [4] and Autodesk Inventor took about 0.17-1.8 ms and 2-16 ms, respectively. Since the above computations were carried out in different software platforms, the CPU times may not convey the effectiveness of the theoretical formulations provided by different authors. However, they do reflect the efficiencies the way they were implemented by the authors. In future, more rigorous tests will be carried out for more complex tasks and using other cylinder-cylinder collision formulations reported in the literature in order to establish the robustness and efficiency of the proposed algorithm.

V. Conclusions

Cylinders are one of the most common geometric entities that are used in various fields of visualization but there are very few exact and accurate methods to detect collision or intersection between them. In this paper, an analytical methodology is proposed for the same. The methodology relies on planar geometries, thereby, making the collision tests fairly straightforward, unlike those reported in the literature which are based on either 3-dimensional spatial geometries or iterative procedure. Based on the CPU times of three different algorithms used to test the collision of the task shown in Fig. 10, the proposed algorithm has been proven to be accurate and most efficient.

[1] System Configuration: Windows XP 32 bit Operating System, Intel Core 2 Duo (3.0 GHz) processor and 4 GB RAM

References

[1] M. C. Lin and S. Gottschalk, "Collision detection between geometric models: A survey," in *Proc. IMA Conference on Mathematics of Surfaces*, vol. 1, pp. 602-608, 1998.

[2] C. Ericson, *Real-time collision detection*, Morgan Kaufmann, 2005.

[3] D. Eberly, *Intersection of Cylinders*, Technical Report, Magic Software, 2000.

[4] J. Ketchel and P. Larochelle, "Collision detection of cylindrical rigid bodies for motion planning," in *Proc. IEEE International Conference on Robotics and Automation*, pp. 1530-1535, 2006.

[5] Y. K. Choi, "Collision detection for ellipsoids and other quadrics," Ph.D. dissertation, Dept. Comput. Sci., Univ. of Hong Kong, 2008.

[6] D. Biermann, R. Joliet and T. Michelitsch, "Fast distance computation between cylinders for the design of mold temperature control systems," Technical Report, TU Dortmund University, 2008.

[7] G. Snethen, "Xenocollide: Complex collision made simple," *Game Programming Gems*, vol. 7, pp. 165-178, 2008.

[8] Libccd, http://libccd.danfis.cz, accessed in July 2013.

[9] M. Kodam, R. Bharadwaj, J. Curtis, B. Hancock and C. Wassgren, "Cylindrical object contact detection for use in discrete element method simulations. Part I–Contact detection algorithms," *Chemical Engineering Science*, vol. 65 no.22, pp. 5852-5862, 2010.

[10] Y. Guo, C. Wassgren, W. Ketterhagen, B. Hancock and J. Curtis, "Some computational considerations associated with discrete element modeling of cylindrical particles," *Powder Technology*, vol. 228, pp. 193-198, 2012.

[11] Teikitu Gaming System, http://www.andrewaye.com/, accessed in Sept. 2012.

[12] R. A. Srivatsan and S. Bandyopadhyay, "An analytical formulation for finding the proximity of two arbitrary cylinders in space," *Proc. International conference on Advances in Robotics*, 2013.

[13] J. Denavit and R. S. Hartenberg, "A Kinematic Notation for Lower-pair Mechanisms Based on Matrices," *ASME Journal of Applied Mechanisms*, vol. 22, no. 2, pp. 215–221, 1955.

[14] I. S. Fischer, *Dual-Number Methods in Kinematics, Statics and Dynamics*, CRC Press, 1998.

[15] S. Bandyopadhyay, "Analysis and Design of Spatial Manipulators: an Exact Algebraic Approach using Dual Numbers and Symbolic Computations," Ph.D. dissertation, Dept. Mech. Eng., Indian Institute of Science, Bangalore, India, 2006.

[16] E. Pennestrì and R. Stefanelli, "Linear Algebra and Numerical Algorithms using Dual Numbers," *Multibody System Dynamics*, vol. 18, no. 3, pp. 323-344, 2007.

[17] C. G. Rajeevlochana, S. K. Saha and S. Kumar, "Automatic Extraction of DH Parameters of Serial Manipulators using Line Geometry," in *CD Proc. The 2nd Joint International Conference on Multibody System Dynamics*, 2012.

[18] M. A. González-Palacios and J. Angeles, *Cam Synthesis*, Springer, 1993.

[19] C. Melchiorri, "Kinematic Model of Robot Manipulators," Online Lecture Notes, University of Bologna, accessed in Sept. 2012.

[20] Y. T. Lee and A. A. Requicha, "Algorithms for computing the volume and other integral properties of solids. I. known methods and open issues," *Commun. of the ACM*, vol. 25, no. 9, pp. 635-641, 1982.

[21] J. Schwarze, "Cubic and quartic roots," *Graphics Gems*, Academic Press Professional, Inc., 1990.

[22] T. Reichenbach and Z. Kovacic, "Derivation of Kinematic Parameters from a 3D Robot Model Used for Collision-free Path Planning," in *11th Mediterranean Conference on Control and Automation*, 2003.

[23] X. Yang, Y. Zhao, W. Wu and H. Wang, "Virtual Reality Based Robotics Learning System," in *Proc. IEEE International Conference on Automation and Logistics*, pp. 859-864, 2008.

[24] D-Cubed, www.siemens.com/D-Cubed, accessed in Sept. 2012.